Ridge Extraction

V. Lacroix *Signal and Image Centre, Royal Military Academy,
1000 Bruxelles, Belgium
Vinciane.Lacroix@elec.rma.ac.be

April 20, 2001

Abstract

A new ridge operator is proposed. It is based on a flexible intensity model exploiting the rank of intensities in a local window around each pixel. A hierarchical method is then proposed to obtain a clean and connected output.

The complete method is used to extract rivers and roads in two SPOT images.

1 Introduction

Many computer vision tasks start from a simple image description usually obtained by a low-level feature extraction process: contours, ridges, salient points, and blobs are the most relevant characteristics.

"Ridge", "bar", "line" , "crest and valley line" are different terms for the same feature, that is, a curvilinear structure either brighter or darker than its surrounding.

Most ridge detectors implicitly or explicitly rely on a intensity model. Masks can be used as templates: in [5] and [2], the best fit provides the operator output and the characteristics of the template are assigned to the pixel. The results thus depend on how well the templates may represent a given line.

Other detectors use the edges as in [4] where the author defines road-centre-hypotheses as anti-parallel intensity edges. The line detection then relies on the edge detection process. Moreover, edges at the border of thin lines are in general too weak to be detected, especially in pairs.

Finally, some detectors use differential geometry, while modeling the image as a continuous function. Gradient or higher derivatives may be used. In [1] for example, a segmentation based on the gradient orientation guides the road detection. The line detector in [3] is based on the fact that the gradient vectors are pointing in opposite direction at each side of a line. In [6], an asymmetrical bar-shape line model is introduced and differential geometry is used to detect its centre. This category of operators has a scale parameter (σ), allowing various line widths extraction. However, if a low σ is chosen, much noise appears, while thin lines disappear for larger σ, if the feature is not salient enough.

The current ridge detector uses a flexible intensity model; it is based on the following assumption: if there is a dark (bright) ridge on a bright (dark) background in the local window, then, there exists an intensity threshold such that all the pixels below(above) the threshold form a connected component, without junction, having at each side pixels belonging to the background. Moreover the connected component may only be one or locally two pixels wide. This model is especially adapted to thin lines but may still represent the middle of wider lines as long as the intensity profile presents there a crest or valley shape.

Low-level operators like ridge or edge extraction usually provide a non satisfying output. Indeed, on the one hand, the output presents undesirable elements, while it lacks of important elements. The undesirable elements come from low contrasted features or from small and isolated elements, all satisfying the local model. The first ones could be removed using some threshold on the contrast, but an overall thresholding may reduce the global connectivity. This is somehow overcome by using hysteresis thresholding in a linking procedure: start on a high contrast value and allow to continue on a lower value. The only way to get rid of the second ones is to impose some constraints on the feature extend. Finally the missing elements come from parts that do not satisfy the model locally, thus a grouping strategy is necessary to extract them. The problem is to avoid to suppress the desirable elements and to link elements that correspond to the same physical object. Most linking procedures do not use the local model anymore, but only the local contrast, while most grouping procedures make use of geometrical con-

*This work was supported by the TELSAT programme of the Scientific, Technical and Cultural Affairs of the Prime Minister's Service (Belgian State)

straints also forgetting the underlying model.

Thus two necessary procedures are needed to provide a satisfying output: linking elements in features (i.e. joining all elements satisfying the model) and grouping features (i.e. introduce elements that do not satisfy the model). It is thus very difficult to compare two low-level operators unless these post-processing procedures are the same.

In this paper, we propose a new low-level ridge output. The linking and the grouping strategy are probably not original by themselves; they could surely be enhanced.

Definitions and Overview

A "pixel" and an "edgel" are respectively defined as a "PIcture ELement", and as an "EDge ELement"; by analogy, we define the new term "ridgel" as being a "RIdge ELement", that is, a pixel belonging to a ridge. A ridge is a set of at least three connected ridgels. A ridge is not allowed to contain a junction, except at its end. Finally, we define a "line" as a set of connected ridgels; it may thus contain junctions. The following analogy will help to understand the line network extraction process: consider ridgel as atom, ridge as molecules, and line as macroscopic matter. In the proposed method, ridgels are first identified by the Ridgel Detection procedure. Then, at each ridgels, the longest ridge centered on the ridgel is obtained by the Longest Local Centered Ridge procedure; the latter assigns a ridge value based on this length. The linking procedure joins the centers of the longest ridges in order to form lines. Finally end of lines are sought for continuation, allowing small gaps to be filled. These last two procedures are repeated until no end of line can be continued.

The next section presents the ridge model and the Ridgel Detection procedure. Section 3 describes the hierarchical method to obtain clean and continuous ridge features. Section 4 shows the results of the method applied to two Spot images. Conclusions and further suggestions are given in Section 5.

2 Ridge Model

2.1 Hypothesis

If a dark (bright) ridge is present on a bright (dark) background in the local window, there exists a threshold such that all the pixels below (above) the threshold form a one or at most two pixels wide line (i.e a connected component without any junction) while the other pixels belong to the background.

Note that the model tolerates asymmetries at each side of the line, as there is no restriction on the background variation. On the other hand, the hypothesis above excludes the ridges that have one dark background at one side and a bright background at the other side, thus avoiding the response of edges. Moreover, the model excludes landing of intensity larger than two pixels. Figure 1 shows examples of valid and non valid ridges, according to the above hypothesis.

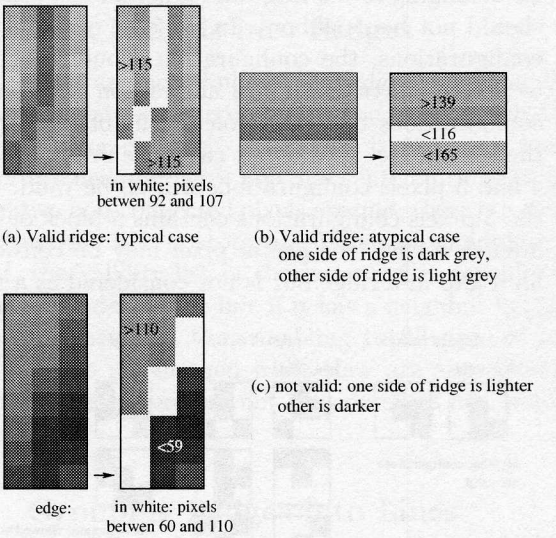


Figure 1: Examples of ridges satisfying or not satisfying the hypothesis

2.2 Ridgel detection

If the ridge assumption is correct, it should also be correct for the smallest local window (i.e. 3×3). The ridge should contain at least 3 pixels (if less, the pixels do not form a ridge) and at most 5 pixels (if more, the number of pixels belonging to the background becomes too small to be called "background").

Thus the aim of the Ridge Detection procedure is to extract the pixels that belong to a local ridge, and generate a "validity list" defined hereafter. The idea of the validity list is to obtain for each potential threshold the list of pixels participating in the local ridge.

The 3×3 surrounding window is analyzed, and the intensity values are sorted by increasing intensity. Let $s[i]$ with $i = 0, ...8$ be the sorted grey-values of the local window.

In the following, we will consider a dark ridge; generalization to bright ridge is straightforward.

As the ridge value contains from 3 to 5 pixels, the grey level g of the current pixel should satisfy the following necessary condition, called "radiometric condition",

to be a dark ridge:
let n ($n = 3, 4, 5$) be the number of pixels in the ridge, then g must be lower than $s[n]$; the background is then made of the values $s[n+1]$,..., $s[8]$.

Thus only the three thresholds $s[3], s[4]$ and $s[5]$ must be tested on each local window. In order to be a valid ridge a "topological condition" should be satisfied: the local configuration should not contain junctions. In the case of 3 pixels configuration, it means that the pixels belonging to the ridge and different from the cent-re should not be neighbors. In the case of 4 and 5 pixels configurations, the configuration should be compared to a look-up table. Fig 2 shows non valid and valid configurations depending on the numbers of pixels in the local ridge. In the best case, like in Figure 3, the 3, 4 and 5 pixels configurations will all be valid. Some of the 5 pixels configurations contains a block of 4 pixels; in such a case, the center pixel may be considered to fill a gap in a ridge but is not considered as a ridgel.

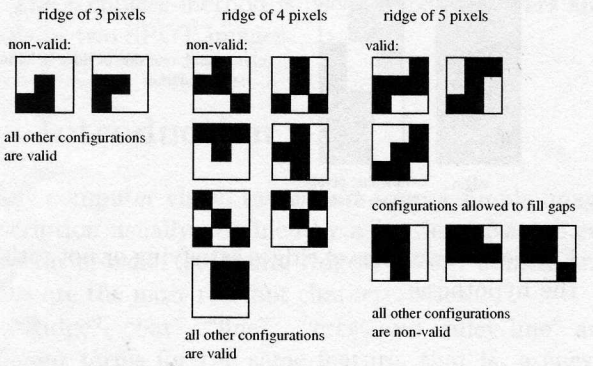


Figure 2: Valid and non-valid local configurations

Assume a process that would highlight the darkest element of the window. Let the highligthing threshold increase from $s[0]$ to t ($s[0] < t < s[1]$) . For some $t = s[i] (i \geq 2)$ a ridge may appear; then, increasing t may highlight more elements in the ridge, until it reaches some $t = s[j]$ where the highlighted pixels do not form a valid ridge anymore. The values of t such as the ridge configuration is valid — and the identity of the pixel — are kept in a "validity list". The last t_{max} is such that the ridge is not valid anymore. Let $[t_0 : id_0, t_1 : id_1, ..., t_{max} : id_{max}[$ be such a list. In the example shown in Figure 3, assuming the origin at the upper left corner, the validity list is is [17:(2,0), 17:(0,2), 17:(1,1),19:(1,0),23:(1,2),30:(0,1)[. Indeed, the first three pixels belonging to the ridge are (2,0),(0,2), and (1,1); they are highlighted when $t = 17$. The next pixel (1,0) is highlighted when t reaches 19 while the fifth pixel (1,0) is highlighted when t is 23. The ridge is not valid anymore when the pixel (0,1) is highlighted, that is, when t reaches 30.

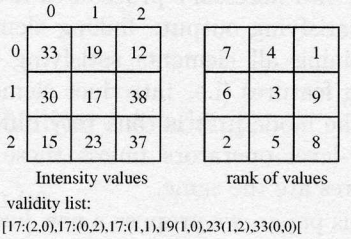


Figure 3: List of valid thresholds

Thus, at the end of the Ridge Detection procedure, at each potential ridgel, a sorted list of intensity values associated to pixel identification is obtained. It gives all the valid thresholds generating a valid ridge and the pixel participating to the local ridge.

3 Ridge following

The aim of this procedure is to obtain the network of lines present in the image. In a first procedure, the longest ridge centered on a ridgel is obtained. The ridgel are then sorted based on the length of the centered ridge. Then a following is performed in three steps. First, ridgels are linked to form ridges without junctions. Then ridges are linked into lines according to geometrical conditions. Finally, the neighborhood of end of lines are explored in order to extend them thus generating joining segments.

3.1 Longest Centered Ridge procedure

The aim of this procedure is to obtain for each ridgel, the longest ridge centered on the current pixel. The value of this length is kept as ridge value. In the current implementation, a procedural approach has been used though on a parallel computer, a systolic algorithm would be preferred. In the ridge detection procedure, all potential thresholds have been obtained. The aim of the current procedure is to find among these thresholds the one that would give the longest ridge centered on the current ridgel, exploring neighbors by and by. This is performed by modifying the current validity list while exploring neighboring pixels.

Then length is subdivided in two, one length per pixel side. At a given pixel, the length at side i will be 2 or 3 depending on the presence of a 4 or 8-neighbor. An example is shown in Figure 4.

At the beginning of the procedure, the pixel belonging to each side of the ridge are identified in the validity list. At each time the "ideal threshold" is the minimum of the list. The first side and the second side will be

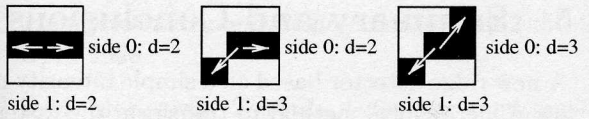


Figure 4: Ridge length

continued at odd time and even time respectively, while comparing the current validity list with the validity list of the pixels at each extreme. When the ridge cannot be continued at one of the side, the process stops.

The compatibility between validity list mainly consists in comparing thresholds. Assume a current pixel with validity list

$$[t_0 : id_0, t_1 : id_1, ..., t_{max} : id_{max}[.$$

At t=1, the validity list of the pixel at side 1 should be compared with the current one. Let

$$[t'_0 : id'_0, t'_1 : id'_1, ..., t'_{max} : id'_{max}[$$

be its validity list. The two lists are merged in one

$$[..., t'' : id, ..., t''_{max} : idmax, ...[$$

with $t'' = \max\{t1, t1'\}$ and $t''_{max} = \min\{t_{max}, t'_{max}\}$. If $t'' > t_0$, additional pixels that were not considered to belong to the ridge should now be checked for compatibility; their validity list should be merged accordingly. Then, only the thresholds between the new t'' and t''_{max} are kept in the validity list.

The length of the current side of the ridge is augmented by 2 or 3 if the considered threshold is associated to a 4-neighbors or 8-neighbor respectively. When the validity list is empty, the thresholds are not valid anymore and the ridge size is obtained. The grey value of a small sample displayed in Figure 5, the validity list and the ridge value at even time are listed below.

```
   |                  437
-----------------------------------------------
   | 141 121 117 137 153 151 143 134 134 144
   | 140 119 120 148 157 142 131 128 122 111
169| 127 124 114 124 121 120 117 111 122 127
   | 115 109 128 142 139 136 135 134 134 127
   | 149 144 158 169 161 154 144 141 131 124
```

Intensity value around (437,169)

```
at (437,169) time-distance-validity list:

t=0 d=4 [121:(438,169),121:(436,169),131:(438,168),135:(438,170)[
t=2 d=6 [121:(438,169),121:(436,169),128:(439,168),131:(438,168),[
t=4 d=8 [124:(437,169),124:(435,169),128:(439,168),131:(438,168),[
t=6 d=11 [124:(437,169),124:(435,169),128:(439,168),[
t=8 d=14 [124:(437,169),124:(435,169),124:(434,169),124:(434,168),
         124:(436,169),128:(439,168),[
```

The pixel at each extreme is stored so that at each centered ridge is characterized by its endpoints. When all the ridgels have be explored, they are sorted on the basis of their ridge size.

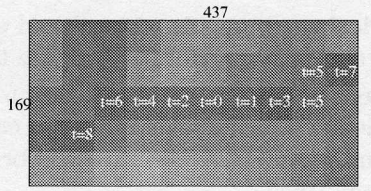


Figure 5: Example of Longest Ridge Value computation

3.2 Link Ridgels into Ridges

The ridgels are now linked into ridges, ignoring the thresholds values, starting from the pixels with the longest centered ridge. Each side is explored, looking for the neighbor with non null centered ridge value. When two non-connected pixels are met, there is a junction thus the line is stopped. When no candidate is found, the validity list of the last pixel is considered, and is appended to the line if it has a neighbor belonging to another line, thus enabling to fill gaps of one pixel. Pixels are marked while they are appended to the list so that there are not considered as new potential starts.

3.3 Grouping Ridges into Lines

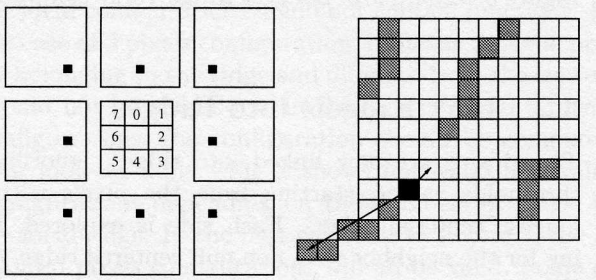
The ridges should now be linked according to geometric conditions. The process analyses ridges by and by, if they were not already appended to a line and while they are longer than RIDGE_SIZE. At both side of the ridge, candidates for continuation are extracted by looking at lines in the neighborhood. The ridge showing the best compatibility with the current ridge is retained. If two ridges have similar angles, a new approximation of the current ridge direction is obtained, and the ridge that is the most compatible is selected, taking the two different approximations into account. At this stage, a complete line is obtained as a list of ridges, themselves being a list of ridgels. The ridge where no candidates are found will be look further in the "Focus on end of line" procedure, if they do not lie on the image border.

3.4 Filling gaps: focus on the end of Lines

Thanks to the local direction of the label at the pixel of focus, a window of WSIZE×WSIZE placed in the continuation direction, is scanned, and, when a ridgel is met, if it belongs to a line, the compatibility between the lines is computed, and the link if performed in the case of agreement. If only ridgels are met, a compatibility test is performed using the longuest centered ridge at that pixel. Then, the ridgel that offers the best

geometrical conditions is selected and a line following is performed; the extremes of lines are then joined.

Figure 6 (a) shows how the window is placed according to the digital approximation of the current direction at the end of the ridge. An example for a direction= 1 is shown in (b).



(a) position of focussing window in function of end ridge direction

(b) example for direction 1

Figure 6: Focusing window

4 Applications

The method is applied to two part of the third band of SPOT image taken in Mozambique, shown in Figure 7. The image size is 512×512. Here are the parameters values:
$RIDGE\_SIZE = 20$
$WSIZE = 7$
Lower values of $RIDGE\_SIZE$ generate noisy lines. Results are shown in Figure 8 and Figure 9(a). Some parts of the lines present in the images are not extracted by the process. A closer look to the regions where this problem occurs shows that ridgels are detected in the neighborhood, but their longest centered ridge were not large enough to start a line there, or the process stopped because of a junction. A smarter following strategy should thus be implemented.

These results can be compared to the ones provided by the "detect-lines" algorithm [6] (special thanks to the author that made his code accessible), shown in Figure 8 and Figure 9(b). Two values of σ have been used. In the test with low σ, the hysteresis thresholds have been set in order to minimize the noise, which still remains important. Note that Steger method allows lines with an intensity profile having a landing at its center, thus generating more candidates. In the second test, the hysteresis thresholds are much smaller in order to be able to keep the road, but many parts are nevertheless lost in this process.

5 Summary and Conclusions

A new ridge detector based on a simple intensity model and a hierarchical method to transform the raster outputs into a network of vector lines has been proposed. The method to exploit the ridge output could probably be enhanced while using a following algorithm that considers a larger neighborhood, as one could predict by looking at the ridge output where the line following stopped.

The process may be summarized as follows. In the Ridge Detection procedure, all pixel belonging to a potential ridge are extracted and the thresholds generating these ridges are retained. In the Longest Centered Ridge procedure, the threshold providing the longest centered ridge is computed, and the size of that ridge is given as ridge output. The ridgels are then sorted according to the ridge size. The largest one are first considered as potential line start. When a junction is met the process stops. Then, ridgels are linked in ridges, ignoring threshold values; the linking stops when junctions are met. Ridges are then linked into lines using geometry only. When no neighboring ridge is found, the end is kept as a focus point. All focus points are then analyzed, and a search in a local window is performed to find non-contiguous candidates. When new candidates are found, the linking and the focus method are used until no focus points exist. No attempt is made to obtain the edges at each side of the line.

Here are the good points of the proposed method:

- large lines indeed extracted
- sorted line output
- precise location and details kept
- thin and low contrasted lines extracted
- low level of noise; noisy lines have the largest labels

References

[1] J.B. Burns, A.R. Hanson, and E.M. Riseman. Extracting straight lines. *IEEE Trans. on Pattern Anal. and Machine Intelligence*, pages 425–455, July 1986.

[2] M.A. Fischler, J.M. Tenenbaum, and H.C. Wolf. Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. *Computer Graphics and Image Processing*, pages 201–223, 1981.

[3] V. Lacroix and M. Acheroy. Feature-extraction using the constrained gradient. *ISPRS Journal of*

Photogrammetry and Remote Sensing, 53(2):85–94, April 1998.

[4] D.M. McKeown. *Towards Automatic Cartographic Feature Extraction*, volume F65, chapter Mapping and Spatial Modelling for Navigation, pages 149–180. Springer-Verlag, nato asi series edition, 1990.

[5] R. Nevatia and K.R. Babu. Linear feature extraction and description. *Computer Graphics and Image Processing*, 13:257–269, 1980.

[6] Carsten Steger. An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):113–125, February 1998.

[7] P. H. Winston. *Artificial Intelligence*. W. H. Freeman and Company, 1982.

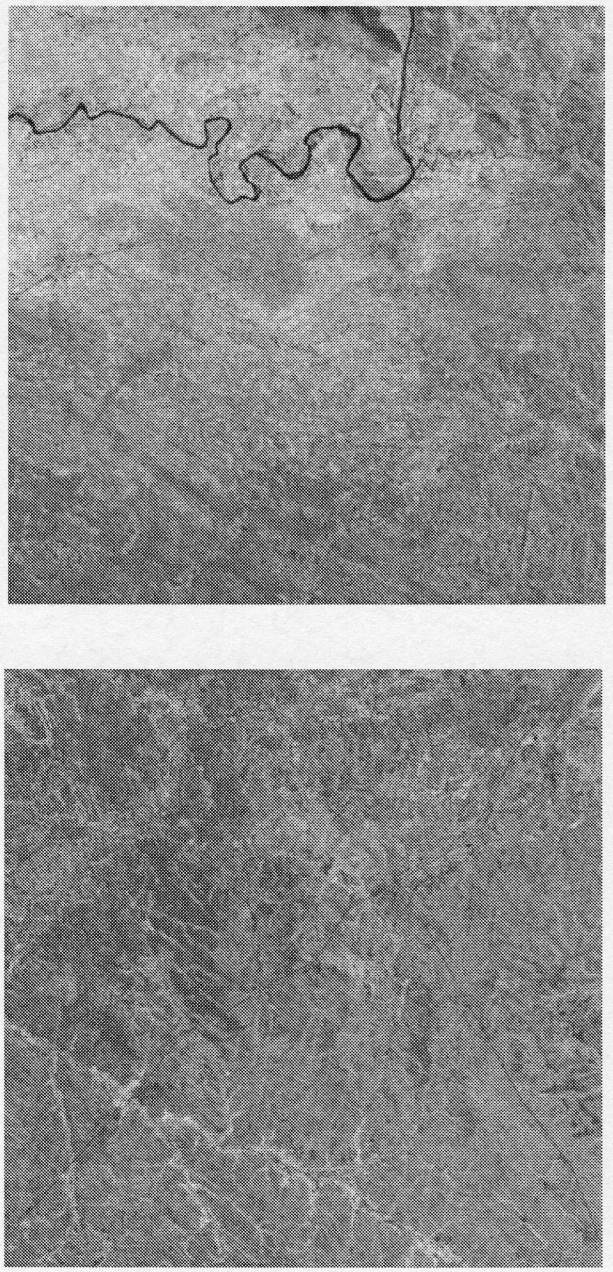
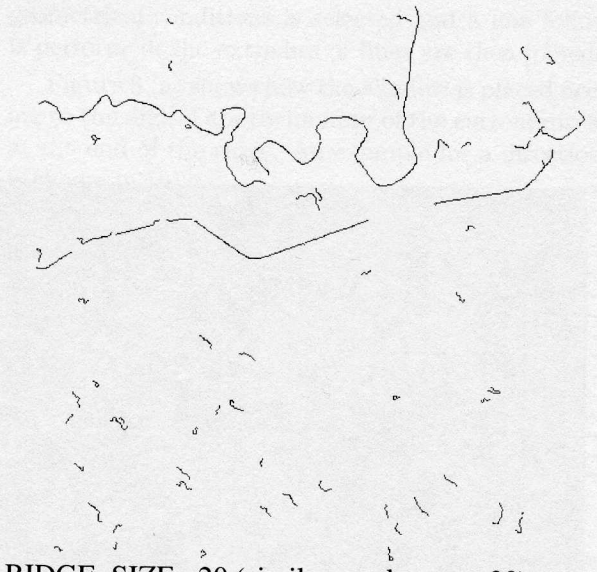
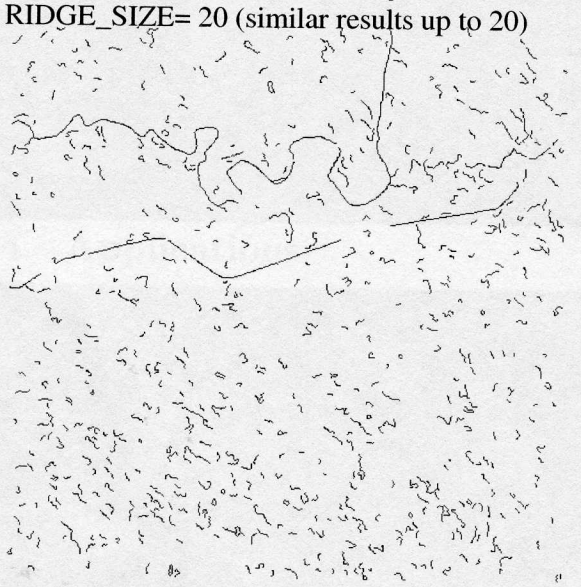


Figure 7: Parts of third band of a Spot Image

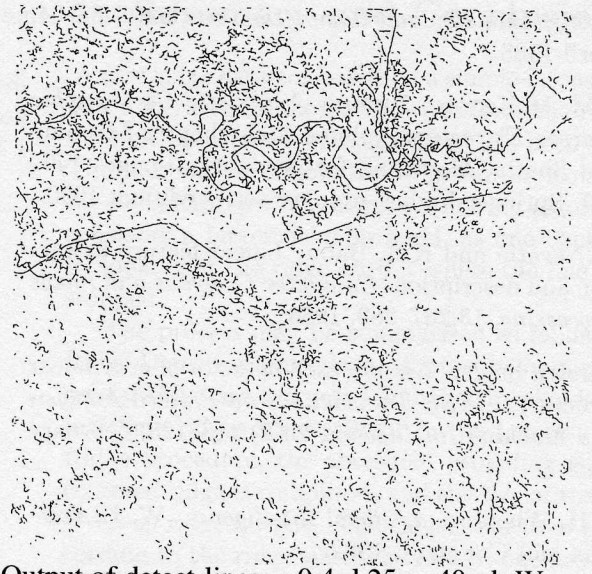


RIDGE_SIZE= 20 (similar results up to 20)

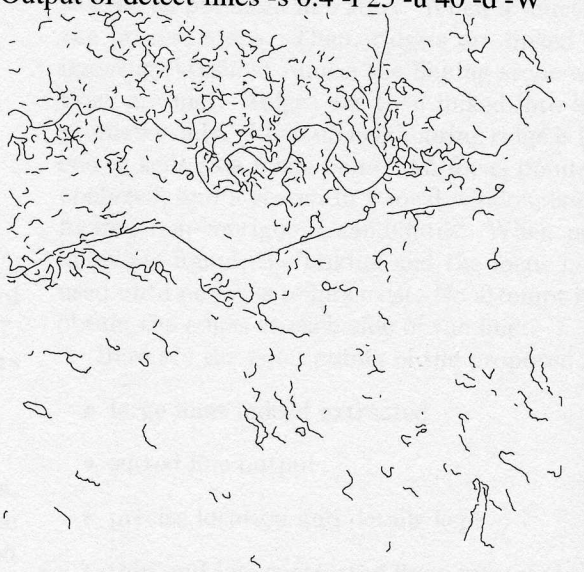


RIDGE_SIZE= 16

(a) Result of the proposed method



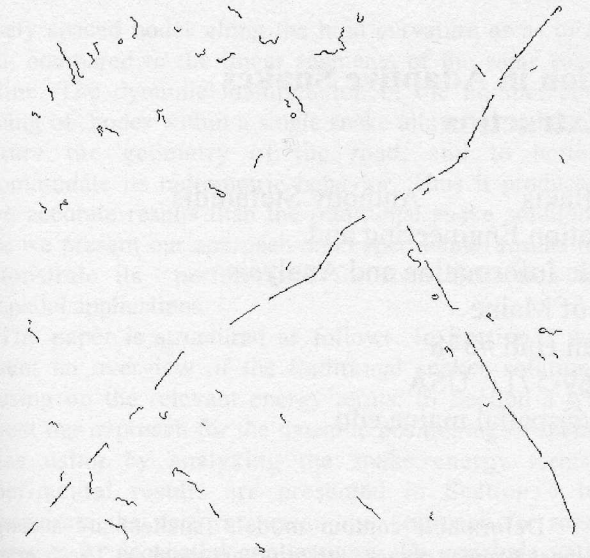
Output of detect-lines -s 0.4 -l 25 -u 40 -d -W



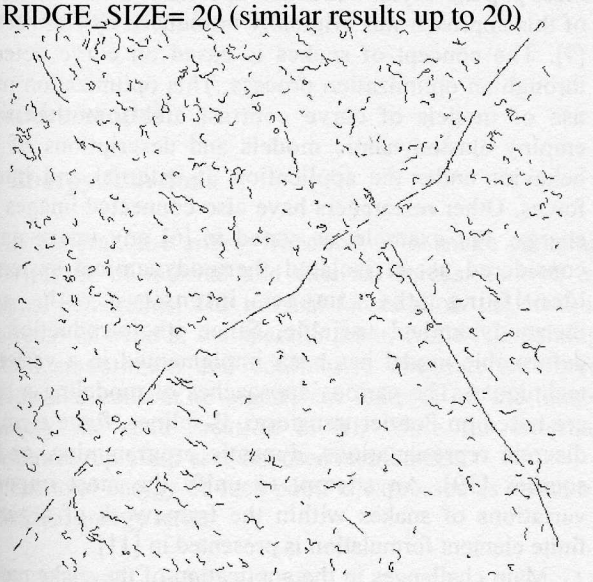
Output of detect-lines -s 0.8 -l 0.8 -u 20 -d -W

(b) Results of Steger's algorithm

Figure 8: Results on a Spot Image

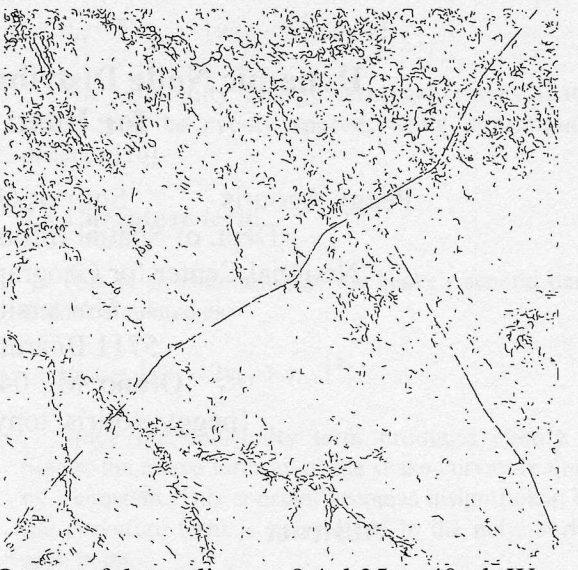


RIDGE_SIZE= 20 (similar results up to 20)

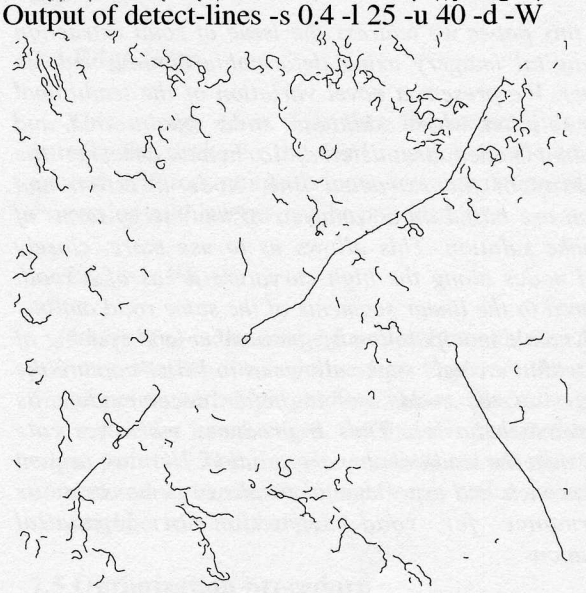


RIDGE_SIZE= 12

(a) Result of the proposed method



Output of detect-lines -s 0.4 -l 25 -u 40 -d -W



Output of detect-lines -s 0.8 -l 0.8 -u 20 -d -W

(b) Results of Steger's algorithm

Figure 9: Results on a Spot Image