# Tools for interactive map conversion and vectorization

Line Eikvil    Kjersti Aas    Hans Koren

Norwegian Computing Center
P.O.Box 114 Blindern, N-0314 Oslo, Norway

## Abstract

*The process of converting an analog map into structured digitized information requires several different operations, which are all time-consuming when performed manually. Strictly automatic processing is not always a possible solution, and an interactive approach can then be an alternative.*

*This paper describes a tool for map conversion, focusing on the functionality for extraction of line structures. An interactive approach is used as it gives the user an opportunity to survey the process, and utilize human knowledge. The methods are based on contour following, extracting centre points needed for accurate vector representation of the line during tracing.*

## 1 Introduction

The use of geographical information systems is increasing. For such a system to serve its purpose it is important to be able to easily acquire the necessary information. Often the information must be retrieved from paper-based maps. Effective ways for conversion of these maps for integration into the geographical information system are then needed.

Manual tracing of maps is a costly and tedious process. Automation is therefore desirable, and several commercial systems for raster-to-vector conversion are available. Most of these systems aim at digitizing the maps automatically. This is a good approach for maps of high quality, but for maps of less quality a great deal of manual operation is still necessary. To ensure correct results, even the results from digitizing of high quality maps are often manually checked against the original.

A different approach is to perform the digitizing interactively. This gives the user a possibility to control the results during digitizing, and human knowledge can be utilized as a supplement to the automatic techniques. This makes the system better suited for complicated maps.

A complete system for data conversion of maps should cover two functions; line structure extraction and symbol recognition, but in this paper we focus on line structure extraction only. Line structures are extracted through vectorization. Two approaches to vectorization are common; thinning techniques [?] and border following techniques [?].

The thinning approach can cause shape distortion at junctions, and it is also less suitable for an interactive approach. We have therefore chosen an approach where the median line is computed during contour following. A similar approach has also been used in [?]. Our method differs from the one described there as it detects corners and curvature during tracing, and adjusts the sampling frequency accordingly. The disadvantage of the contour approach is said to be its inability to distinguish between borders of lines and characters [?]. However, this problem is avoided through the semi-automatic approach.

## 2 System Overview

The process of converting the scanned raster map starts by extracting the line structures. The underlying raster of these lines can then be removed from the working layer. By the removal of these lines, the symbols in the map are segmented and can be recognized automatically. In this paper we concentrate on the part of the system used for data acquisition, and describe the raster-to-vector conversion of the line structures.

## 3 Interactive vectorization of lines

The digitizing starts by the operator indicating the start point and direction for the line. The line is then traced automatically, and gaps and junctions can also be automatically crossed. The tracing continues until the end of the line is reached or until an area of conflict is encountered. A conflict occurs when the digitizer is unable to find a continuation. The user can then guide the digitizer through the difficult area.

It can be useful for the operator to be able to see which lines that have been digitized. Therefore the underlying raster of the processed lines are stored. These lines can then be removed or marked with a different colour.

### 3.1 Tracing lines

The lines are digitized by tracing their contours (fig. ??), and keeping continually track of the positions on each contour. Along straight line segments, one point is fetched from each contour at the time. The next point fetched is always a neighbour of the current point.

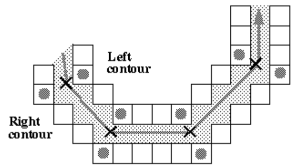


Figure 1: *Tracing the raster line. The filled circles indicate the contour points selected for computation of the representative points, which are marked with crosses.*

The midpoint between the current two contour points are computed at short intervals, and stored if needed to represent the line. The maximum distance between two representative points is predefined, but in areas where the direction of the line changes representative points are determined at shorter intervals. In addition, centres of corners and junctions and start and end points of gaps are stored.

Whenever a representative point is computed, the current points on each contour should be situated opposite of each other. That is, the line between the two points should be perpendicular to the raster line at this point. To obtain this the current points on the contours are moved independently when the line curves. Whether a line curves or not is decided by investigating whether the direction of the contours has recently changed. When curvature is detected, the contours are checked for corners.

## 3.2 Detecting corners

Curvature and corners are found by investigation of segments along the contours. The distance between the straight line corresponding to a contour segment and each point on this segment is computed (see fig. ??). If the distance to the point furthest away from this straight line is very small the line is assumed to be straight. Otherwise, the line is assumed to change direction at this point.

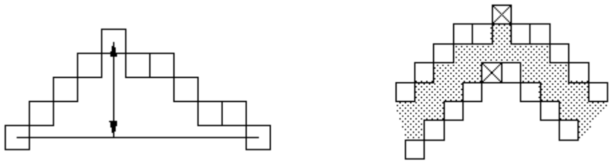

Figure 2: *Detecting changes in direction.*

We distinguish between changes in direction caused by corners and by curves, as this helps the determination of accurate representative points. The types of the corners will be evaluated, and for each outcome of this evaluation, different situations are assumed. For a true corner (fig. ??, left) the representative midpoint is found. For a junction (fig. ??, middle and right), the correct path through the junction is found.

## 3.3 Crossing junctions.

There are three different alternatives for automatic continuation through a junction; straight through the junction, along the leftmost path or along the rightmost path. One of these alternatives should be chosen by the user prior to the linefollowing, and may be changed for each new line.

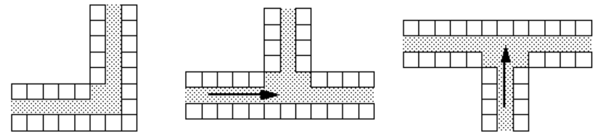

Figure 3: *Left: 1 concave and 1 convex corner, Middle: 1 concave corner, Right: 2 concave corners.*

### 3.3.1 Straight crossing

To continue straight through a junction corresponds to continuation in the current direction. The digitizer therefore determines the direction of the segment traced immediately before the junction. Moving tentatively a distance forward in this direction, a search for contour points on the line emerging on the other side of the junction is performed. If the digitizer is unable to find an appropriate path, the digitizer stops and asks for user interaction. In figure ?? the result of digitizing a utility map with several junctions, is shown.

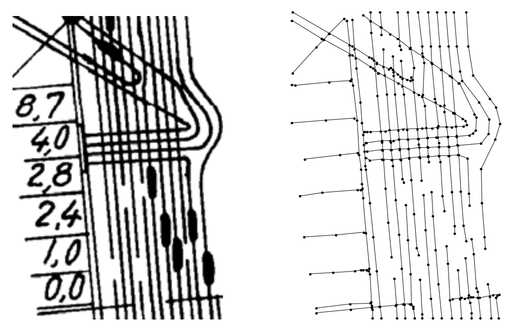


Figure 4: *Raster map (left) and resulting vectors (right) of a utility map.*

After passing a junction, reconstruction of the contours may be necessary depending on the type of junction. The junction is therefore classified into one of two categories. The first category includes junctions where lines meet and the second category junctions where the lines actually cross. Reconstruction is performed only in the first case.

### 3.3.2 Left or right turn

In this case one of the contours will be continuous through the junction. The idea is then to continue along this contour, passing the junction. When the junction is passed, the raster line is traversed to find a point on the opposite contour. Finally, the type of the junction is evaluated for restoration of the contour.

#### 3.3.3 Reconstruction of contour

To obtain the contours of the digitized raster, the course of the lines are stored as they are traced. At junctions, the contours are not defined and may therefore have to be reconstructed.

For junctions where lines cross (fig. ??, left), it is desirable to keep the crossing line intact. Therefore the underlying raster of these junctions should not be removed from the working layer. This can be obtained, if the contours are not reconstructed through the junction.

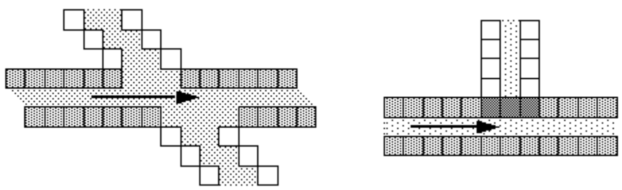

Figure 5: *Exact contours for crossing lines (left) and meeting lines (right).*

For junctions where lines meet (fig. ??, right), we assume that the outgoing lines which are not being followed, starts at the border of the current line. This border is then reconstructed. If the underlying raster is removed from the working layer, symbols connected to the line will be left undisturbed. This is useful for later recognition of these symbols.

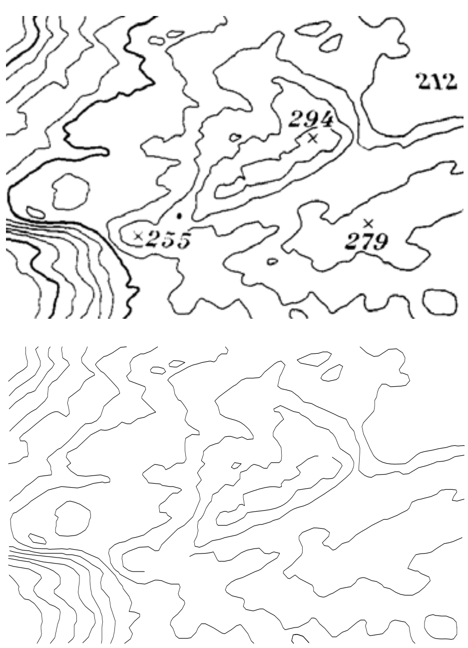


Figure 6: *Raster map (top) and resulting vectors (bottom) of a geographical map.*

### 3.4 Crossing gaps.

Gaps occur where lines are accidentally broken or in intended dashed lines. In the case of accidental gaps we assume that there is only one possible continuation, and that the continuation can be found along the current direction of the line. The gap is crossed by searching from the point at the end of the line within a sector around the current direction.

The search for a continuation in dashed lines is performed within a sector defined by an angle and a minimum and maximum length of a gap.

## 4 Digitizing of areas

The digitizing of an area is not interactive in the same way as the linefollowing. When the user has specified the area which should be digitized, the digitization is performed without user interaction. Two types of areas are defined:

- **A solid area** is a connected area of foreground pixels. Emerging lines connected to the area may or may not be a part of the area, depending on the current definition.
- **A linebounded area** is a connected area of background pixels delimited by a border of foreground pixels.

To start the digitizing, the user specifies a point within the area. The type of area is evaluated, and from the initial point a search for a startpoint on the contour is performed. The contour of the area is then traced with a different method for the two area types. When the tracing is complete, the extracted contour is checked to determine whether it circumscribes the initial point. If not, the contour does not belong to the correct area and the whole procedure is repeated. When the correct area is found, the desired representation of the area is extracted from the contour.

### 4.1 Tracing areas

The initial start point may be arbitrarily chosen within the area, and to start the digitizing, a point on the actual contour must be found. From the initial start point the area is traversed in one predefined direction at the time, until an edgepoint is found, or until a maximum number of attempts has been done.

#### 4.1.1 Tracing a solid area

The contour is traced starting in the obtained edge point, and adjacent points on the boundary of the area are fetched one by one. Some areas may have incoming lines which we want to omit (e.g. lakes with streams). Such lines must therefore be detected and crossed, and at certain intervals a check is performed to find out whether the digitizer is still on the contour of the actual object, or on the contour of an emerging line.

In the case of an incoming line, a point on the other edge of the line is found. A straight line between the previous contour point and this point is reconstructed and added to the contour, before the the tracing continues from the new contour point.

#### 4.1.2 Tracing a linebounded area

The methods used for tracing linebounded areas are similar to those presented for interactive vectorization. However, when junctions are encountered, the innermost path is always chosen. This means that incoming lines will be included during the tracing. However, these are removed at a later stage.

The handling of line ends also differ from that of the linefollower. As the assumption is that the area is closed, a line end should only occur if there is a gap in the contour or if an incoming line is encountered. In the first case the gap is crossed. In the latter case, nothing is done, which means that the tracing continues back along the incoming line (see fig. ??). Hence, the tracing does not stop at the end of a line, but at return to start.

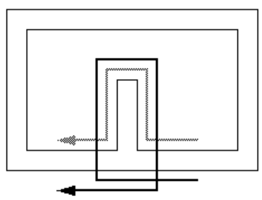

Figure 7: *Inner and outer contour along an incoming line.*

The earlier mentioned incoming lines, are removed when the whole area is traced. The idea is to exploit the fact that the incoming lines are connected to the true contour at junctions. The junction leading out to an incoming line will always be passed twice. By investigating all the junctions, the incoming lines can then be detected. A list of the positions of all the junctions, is therefore updated as the junctions are passed. The positions of the incoming lines are identified, and the part of the contour lying between the first and the second pass of a junction, is removed. In figure ?? the result of digitizing som linebounded areas are shown.

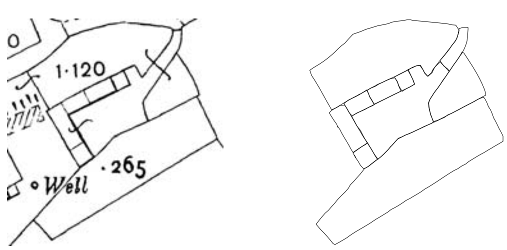


Figure 8: *Results tracing some linebounded areas.*

### 4.2 Verification of the area contour

When the tracing of an area is complete, it has to be verified that the extracted contour circumscribes the initial startpoint. The verification is performed imagining a straight line emerging from the initial startpoint. The number of times this line crosses the contour is then counted. If the number of crossings is odd, the initial startpoint is contained in the area found, otherwise it is not. If the area does not enclose the startpoint, the entire process is repeated searching for an edgepoint in a different direction.

## 5 Summary and conclusions

In this paper we have presented selected tools for map conversion included in a system providing a GIS toolset. These tools include interactive vectorizing of lines, digitizing of areas, filtering and text recognition. Only the first two options are described in this paper. A binary raster image is the basis for the map conversion. The lines are digitized using an approach based on contour tracing, and the representative points are extracted as the lines are followed. The exact contours of the raster lines are also extracted during linefollowing, for later removal or marking of the underlying raster.

The interactive approach gives the user an opportunity to survey the process. At any point the process may be interrupted and restarted. This makes the tools well suited for digitizing of map sheets with reduced quality or where different themes are superimposed on the same sheet. In addition to the information extracted directly from the map, it is often desirable to registrate additional information connected to the structures and objects that are digitized. With an interactive system, attributes from different sources can be included during digitizing, which is often convenient.

**Acknowledgement** The Norwegian Research Council have supported most of the work presented in this paper.

## References


[1] L. Lam, S-W. Lee & C. Y. Suen. *Thinning Methodologies – A Comprehensive Survey.* IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 14, No. 9, pp. 869–885, 1992.

[2] M. P. Martinez-Perez, J. Jimenez & J.L. Navalon. *A Thinning Algorithm Based on Contours.* Computer Vision and Image Processing 39, pp. 186-201, 1987.

[3] P. R. van Nieuwenhuizen, O. Kiewiet & W. F. Bronsvoort. *An integrated Line Tracking and Vectorization Algorithm* Proceedings Eurographics '94, Vol. 13, No. 3, pp. 349–359, 1994.

[4] T. Kaneko. *Line Structure Extraction from Line-Drawing Images.* Pattern Recognition, Vol. 25, No. 9, pp. 963–973, 1992.