IAPR Workshop on CV - Special Hardware and Industrial Applications OCT.12-14, 1988, Tokyo

A RASTER-TO-VECTOR-CONVERSION CONCEPT BASED ON INDUSTRIAL REQUIREMENTS

Rune Espelid
Bergen Scientific Centre, IBM

Allégaten 36
5007 Bergen, Norway

ABSTRACT

A raster-to-vector (r-t-v) conversion system that is to be of general interest to the industry must meet certain requirements. With IGES as the starting point, covering the CAD area, I discuss a number of such needs. A strategy for the design of a r-t-v system that matches the requirements listed is then proposed. I find it necessary to leave the pure skeleton based approach, but use skeletonal information as a model of the drawing, and classify the entities and extract their parameter data in a top-down approach. During this process the geometric features are extracted from image contour lines.

INTRODUCTION

Raster-to-Vector (r-t-v) conversion is a problem which is not widely considered in the literature, even though the subprocesses of skeletonizing and polygonizing raster patterns has a long research history. One reason for this is that the problem has been considered to be an engineering task of combining well described methods. However, existing r-t-v conversion systems are not widely accepted as useful tools in data capture. The results produced are of variable quality, and the manual verification and editing necessary as postprocesses make the benefits questionable.

Two surveys [1], [2] capture the history of computer processing of line images. The first one discusses a range of topics from preprocessing to entity recognition, but the problem of vectorizing is only treated as a problem of image quantizing and tracing. The other surveys a wide range of thinning algorithms, but the author is not satisfied with their quality or speed for practical purposes. Thinning-based r-t-v systems are described for example in [3] and [7]. The question of requirements of an r-t-v algorithm is on touched in [3], but what is actually specified are the requirements for a skeletonizing process.

However, a few attempts have been made to design a vectorizing system that is not based upon skeletonized patterns. Reference [5] describes a method where the vectors are formed by analysing the contours of the image pattern. A serious problem with this kind of approach, which is not discussed in the paper, is to guarantee that the topology of the input line-pattern is perceived.

In [6] the method is to do radial search and tracking to find the longest vector drawable on the line segment. Again, the problem will be to reproduce the topology of the line work, especially when the input pattern also contains solid figures, for example small arrowheads thinner than the upper limit for acceptable line thickness. In [4] Bley concentrates on creating a graph structure to represent the input pattern.

My approach has some similarities with the approach found in [4]. Like Bley, I create a graph structure to represent the topology of the image pattern. According to a geometrical condition, he merges and splits the black pixels into primary components, which are represented by the nodes in a certain type of line-adjacency graph. I map the topology more directly since the nodes of my graph represent the crossings of the linework, extracted from a skeletonized image.

The first part of this paper presents a list of requirements that a true raster-to-vector conversion system should meet. This list emerges from a discussion of what is required as output from a total drawing interpretation system. I require this system to be capable of supplying CAD databases with logical entities as they are defined in the Initial Graphics Exchange Specification (IGES) [8]. From this point of view the raster-to-vector conversion is the first and critical step in a complex process, where it is important to extract correctly, during vectorization, certain features of the input pattern. Furthermore, the system should not make premature decisions when eliminating noise and classifying entities. That is the task of higher-level interpretation modules which have the necessary overview of the context to draw the correct conclusions.

On the basis of the defined functionality of a raster-to-vector conversion system, a new strategy for

designing such a system is presented in the last part of the paper. The approach is based on combining skeletonizing and contour tracing. A connected skeleton provides the structure of the input data, while the contours give us shape attributes. Both classes of information are of critical importance to local and regional interpretation of the drawings. This system has the property of not making any connections that do not appear in the input patterns, as well as not making any artificial gaps. The system is capable of differentiating between thick lines and small filled-in areas, and the tendency to represent straight lines by a sequence of short vectors is avoided.

INDUSTRIAL REQUIREMENTS

My key issue is to determine what exactly is required from a r-t-v system in order to meet industrial needs. There are two dominant applications, the mapping area and the CAD area. I choose here to concentrate on the CAD area, claiming without further justification, that the fundamental problems of vectorizing maps are then covered at the same time. A general description of CAD formatted data is given by the IGES. A useful conversion system should produce data that match these descriptions.

It is a challenging task to try to develop a system that is able to extract from paper drawings all the logical information specified in IGES. A useful r-t-v system has to be designed with that goal in sight. In such a context I consider the r-t-v system as a first and critical step, taking us from low-level image processing to high-level image analysis. It is therefore important that the r-t-v conversion should not produce insufficient and inaccurate data.

The four main headlines of the IGES description are Data Form, Geometric Entities, Annotation Entities and Structure Entities. In addition to the twenty different form parameters, there are altogether fifty five specified entities. Each entity is described by an individual set of parameter data. From this description I have extracted what I call the **basic entities**. These are listed in Table 1, together with those parameters which determine their geometrical properties.

All other entities are either what I call **composite entities** (i.e. entities made up by combining basic entities), or they can be recognized from the same set of features as is needed to be extracted for the basic entities, e.g. splines can be recognized by analysing the same set of features as is needed for the circular arc. Leaders (arrows), planes and symbols are examples of composite entities. As far as 3D is concerned, that is also a problem of deducing logical information by analysing the combination of entities relevant in the context.

A general r-t-v system should be able to recognize the basic entities, with high quality parameter data, with some modifications for the note entity. I now discuss a list of requirements that the system has to meet in order to perform this task successfully.

ENTITY	PARAMETER / FORM DATA
Point	(x, y)
Line	(x_s, y_s), (x_t, y_t)
Circ. arc	(x_c, y_c), (x_s, y_s), (x_t, y_t)
Composite	Parameters of constituent curves
curve	
Note	Position, text font, size, orientation
	Line weight (thickness)
	Filled area

s=start point, t=termination point, c=center point

Table 1. The basic entities.

Find the interesting points: By interesting points I mean single points, end points, junction points, and separating points (the points separating the constituents of composite curves). This is an obvious requirement. However, the problem is to have them registered with the accuracy required. The separating points are the most problematic ones.

Register the line thickness: The line thickness is not only needed as parameter data describing the final entity, but it also enters into various subsequent recognition processes as an important feature. The line thickness should therefore be known for all local sections along the curve. It is not sufficient to measure a few samples along the curve, and then accept the mean value as the answer.

Register the curvature: The curvature along the line pattern is needed for the classification of entity types, e.g. straight lines, circular arcs and splines, and for the determination of the constituents of the composite curves.

Register filled-in areas: The geometrical shape of filled-in areas (black areas) must be extracted by the r-t-v system. This is required not only for the large ones, but also for small arrowheads and other black symbol parts.

Map the topology exactly: The basic entities constitute input data for the recognition of composed entities and structures. But extraction of this logical information is impossible without knowing the topology of the drawing. I have to know all connections and gaps. These should not be determined by thresholding the spatial distance between the basic entities. I require that no artificial gaps must be created, and no connections should be made that do not exist in the drawing.

Label text candidates: Recognition of text characters and strings is not the task of the r-t-v system. Text recognition is generally based on a set of well-described features that may be quite different from what is needed to recognize the geometric entities. It is therefore profitable to distinguish between text and geometry as early as possible in the automatic conversion process. I therefore prefer the r-t-v system to be capable of making an initial guess as to what are character candidates. Further more, it should support as much relevant information as possible to determine text font and size.

Support sufficient accuracy: Several factors influence the final accuracy of the vectorized data: the accuracy with which it was drawn, the stretching of the paper that has been scanned, the geometrical error in the scanner optics and discretization process, and the geometrical accuracy of the r-t-v system. In extreme situations the paper stretching can be measured in millimeters per metre, while the scanning error may be kept within the limit of 0.06 mm (a 400 dpi scanner). For some applications a sufficient requirement for the r-t-v process may be that it produces visually attractive results. That implies that the entity parameters must retain an accuracy, compared to the input drawing, of at worst 1/10 mm. On the other hand, calculating coordinate values with more significant digits than the scanner resolution holds is a waste of effort.

Limited usage of predefined parameters: It has been considered of importance in image processing applications to use a priori information, in order to reduce the computational complexity. The danger is that the systems may be made too specialized. The reliability of the results is also weakened when the operator has to answer a long sequence of questions. This is especially true when technical knowledge is required about how the implied methods in the r-t-v process depend on the parameters. Predefined parameters should therefore be directly related to technical terms of the application, and hence easy to understand by the user. Such parameters, that do not vary much from one class of drawings to another, are minimum and maximum line thickness, and minimum and maximum character height

THE PROPOSED CONCEPT

I now propose a strategy for the design of a r-t-v system that matches the requirements discussed. I find it necessary to leave the pure skeleton-based approach. The thinning methods are noise-sensitive, and the skeletons they produce are not symmetric, junctions and filled-in areas, in particular, are badly treated. Entity parameters calculated from the skeletons will therefore not meet my accuracy condition. On the other hand, a connected skeleton carries the important topology information of the drawing, and the underlying philosophy of my approach is to take advantage of that information. I use the skeletonal information as a model of the drawing, and classify the entities and extract their parameter data in a top-down approach. During this process the geometric features are extracted from the pre-thinned image contour lines. The principles of my approach are discussed in the following sections.

Contouring and partial thinning: Skeletons have little meaning for filled-in areas. Also, information about the existence of such areas is an important part of the model of the drawing. The maximum line thickness parameter is therefore used to stop the thinning process after a certain number of iterations. All lines are then skeletonized, while the black areas are somewhat distorted, but perceived by the system. Partial thinning combined with contour tracing of remaining areas was first introduced as a part of a feature extraction method for recognition of symbols [9], but appears to be of general interest for document-processing applications.

In order to perceive the connectivity, I require the thinning method to produce 8-connected skeletons.

In addition to a partially thinned skeleton, I also produce a contour image of the pre-thinned pattern. These contours are required to give us the necessary curvature and shape information.

The drawing-graph: A model of the drawing is generated by creating a graph structure, where the nodes represent the interesting points (and black areas), and the edges represent the curves and lines. This graph is called the *drawing-graph*. The important elements of this graph are shown in Table 2. The nodes and edges are established by applying a line-following algorithm to the skeleton and the borders of the remaining distorted areas.

NODES: Position, Degree, Shape pointers, Edge pointers.

EDGES: Contour-1 pointer , Contour-2 pointer, Node pointers.

CONTOURS: Chain code list 1, ... ,Chain code list n.

Table 2. The main elements of the drawing-graph.

However, the double set of contour information which describes each edge is extracted by applying a line-following algorithm to the pre-thinned contour image. If the node represents a black area, the shape pointers point to contour chain codes. The contours of the distorted black areas act as intermediate results, that are replaced by the corresponding pre-thinned contours as soon as the topology of the area is clear. Notice that the line thickness is implicitly available for every position of the linework.

This graph can be analyzed for connected components of certain sizes and complexities, and it offers the possibility of segmenting the drawing, e.g. isolated text candidates.

Cleaning the graph: The artifacts produced by the thinning process are well known. They can be grouped into three categories, see Figure 1. These effects have some characteristics that can easily be detected. They produces nodes of degree three, they have a free end or merge into another artifact (the X-effect), and their length is proportional to the local line thickness. The XYT-effects can therefore be cleaned from my graph description.

a) X-effect b) Y-effect c) T-effect

Figure 1. Artifacts produced by the thinning.

Even when partial thinning has been applied, some significant black shapes may have been removed. See Figure 2 for examples.

a) Arrowhead b) Small area c) Varying weight

Figure 2. Shapes destroyed by partial thinning.

A close analysis of the contours of the pre-thinned drawing is necessary to detect these shapes. They are represented by extra nodes inserted into the drawing-grapf.

Recognizing basic entities: The two contours associated with each edge of the drawing-graph describe the curve between the two nodes. This curve must first be analysed in order to determine whether it is a composite curve or not, and if it is, to determine the separation points. These points constitute new nodes in my graph.

Secondly, the entity type is determined by extracting curvature parameters, and at the same time its position and orientation can be calculated. When considering these computations, an important detail is to avoid including the end sections of the contours. The shape of these sections is influenced by the neighbouring lines and areas, and will therefore disturb the results. The contours of the (pre-thinned) filled-in areas are analysed in the same manner as the between-node curves.

The end points of the entities are determined in a separate iteration over the drawing-graph. For each node the intersection(s) of the incoming lines are calculated, thereby determining their end points. For nodes of degree one (free ends), the end point is determined as the point of intersection between the curve and the contour.

By my approach I have avoided all the problems related to the inaccuracy of the geometry of the skeletons. The intersections are given a geometrically correct position. The connectivity of the drawing, which was perceived in my graph, is now transferred back into the geometry.

Opposite to a skeletonizing and polygonizing approach, the starting hypothesis of my method consideres the whole between-node curve as one entity. Only if composite curves are detected this entity will be devided into more than one.

CONCLUSION

I have been able to identify a list of requirements that a general raster-to-vector conversion system should meet in order to be of interest to industry. Systems that base the recognition of geometric properties on polygonized skeletons will not be able to meet these requirements. Furthermore, the topology of the drawing appears to be indispensable in order to recognize composite entities.

I have found a concept and a combination of methods that have the potential to fulfil these requirements.

By my approach I have avoided all the problems related to the inaccuracy of the geometry of the skeletons, which have been a headache for many designers. The intersections and end points are given a geometrically correct position, and the connectivity of the recognized entities will match the connectivity of the input drawing.

The problem of ending up with several short vectors that describe one straight line has been a typical pitfall for skeleton-based approaches. That problem is minimized by my method, and for two reasons. These are, that I do not incorporate the misleading geometry of the skeleton, but also because my approach, as a starting hypothesis, considers the whole between-node curve as one entity.

References:

[1] H. Freeman, *Computer processing of line-drawing images*, Computing Surveys, vol. 6, No.1, pp.60-97, March 1974.

[2] R.W. Smith, *Computer processing of line images: A survey*, Pattern Recognition, vol. 20, No.1, pp.7-15, 1987.

[3] J.F. Harris, et.al., *A modular system for interpreting binary pixel representations of line-structured data*, in Proc. of the NATO Adv. Study Institute on Pattern Recognition Theory and Applications held at St. Annes's College, Oxford, March 29-April 10, 1981.

[4] H. Bley, *Segmentation and preprocessing of electrical schematics using picture graphs*, Computer Vision, Graphics, and Image Processing, 28, pp. 271-288, 1984.

[5] W. Pferd and K. Ramachandran, *Computer aided digitizing of engineering drawings*, Proc. IEEE Comsac, pp. 630-635, 1978.

[6] M. Ejiri, et.al., *Automatic recognition of design drawings and maps*, Proc. 7th ICPR, pp. 1296-1305, Montreal, Canada 1984.

[7] G. Woetzel, *A fast and economic scan-to-line conversion algorithm*, Gesellschaft für Matematik und Datenverarbeiten, Postfach 1240, D-52 St. Augustin 1, West-Germany.

[8] *Initial Graphics Exchange Specification (IGES)*, Version 3.0, April 1986, U.S. National Bureau of Standards (NEL), Geithersburg, MD 20899.

[9] E. Hansen and K.P. Villanger, *A combined thinning and contour tracing approach to the recognition of engineering drawing symbols*, The Int. Seminar on Symbol Recognition, Oslo, October 1985.