A structural representation for understanding line-drawing images

J.Y. Ramel1, N. Vincent2, H. Emptoz3

1 Laboratoire ICTT, INSA de Lyon – Bât. 403, 20, av. Einstein, 69621 Villeurbanne Cedex, France; e-mail: ramel@gprhp.insa-lyon.fr
2 LI/E3i, Université F. Rabelais, 64, av. J. Portalis, 37200 Tours, France; e-mail: vincent@balzac.univ-tours.fr
3 Reconnaissance de Formes & Vision, INSA de Lyon – Bât. 403, 20, av. Einstein, 69621 Villeurbanne Cedex, France; e-mail: emptoz@rfv.insa-lyon.fr

Abstract. In this paper, we are concerned with the problem of finding a good and homogeneous representation to encode line-drawing documents (which may be handwritten). We propose a method in which the problems induced by a first-step skeletonization have been avoided. First, we vectorize the image, to get a fine description of the drawing, using only vectors and quadrilateral primitives. A structural graph is built with the primitives extracted from the initial line-drawing image. The objective is to manage attributes relative to elementary objects so as to provide a description of the spatial relationships (inclusion, junction, intersection, etc.) that exist between the graphics in the images. This is done with a representation that provides a global vision of the drawings. The capacity of the representation to evolve and to carry highly semantic information is also highlighted. Finally, we show how an architecture using this structural representation and a mechanism of perceptive cycles can lead to a high-quality interpretation of line drawings.

Keywords: Vectorization – Feature extraction – Structural representation – Line drawings

1 Introduction

The classical aim of vectorization is to convert raster images to a vector form. Vectorization is a fundamental task in the interpretation process of line drawings because it is used as a preliminary step. Further interpretation requires that the results give the finest possible description of the raster image. Thus, many vectorization techniques have been developed in various domains, and many methods are now available [1–4]. Most of these methods are adapted to thin shapes, but not to the other kinds of data that also need to be extracted (text, filled shapes, etc.). We think that most of the problems in document understanding are due to processing this way.

This paper tries to show that more robust representations than the usual vectorization can be defined to represent a drawing. Vectorization is often sufficient to store a line drawing, but some problems occur in interpretation.

This paper first describes a new vectorization algorithm that uses vectors and quadrilateral primitives to provide a fine description of the drawing. Then we show that, based on these primitives, a structural graph can give the essential information about relationships between the objects comprising the drawing.

We comment on results obtained for several images and show that this process leads to good representations of line drawings. Then, the mechanism of the evolution of this initial representation is briefly described and shows that the higher-level entities (curves, text, etc.) can be handled well in this structural representation.

2 Freehand line-drawings and vectorization techniques

2.1 Freehand line drawings

Each document has its own structure, chosen so that the information transmitted is easily understood and can be reused by the readers. Freeline drawings constitute a particular category: they are made up of fluctuating lines, solid areas, hatched areas, text, etc. Shapes to recognize in these drawings are generally polygons, lines, circles, and other elementary geometric objects. The variable quality of the lines requires setting up specific line-recovery and correction algorithms before or during the recognition phase. Moreover, research done in this field [5–7] emphasizes the difficulty of this task, even for a human reader. The classical techniques of vectorization, text localization, curve localization, and so on can only be used when some modifications, so that they take into account this fluctuating aspect of freehand lines. The vectorization algorithm that we propose allows the analysis of such drawings.
2.2 Well known vectorization techniques

Vectors are compact forms that describe the geometry of a graphical object with a small number of attribute values. For thin shapes, the median axis and the line length may be considered. Line drawings are essentially made of straight line segments and arc lines, so many automatic reading systems begin the analysis process with a skeletonization. Then, ensuing treatments are made easier, but this method induces problems because information such as line thickness is lost. After that, a more sophisticated structure than the pixel has to be chosen to encode the skeleton: critical points have to be selected. Most often, the approximation uses linear elements (segments) to vectorize the image [8]. Unfortunately, the control points (of the polygonal lines) do not always lead to a good initial shape, as can be seen in Fig. 1. Furthermore, skeletonization is not at all appropriate for the study of filled shapes.

Although an image of a line drawing essentially consists of lines and curves, there are other kinds of shapes, such as solid shapes and text zones. To process such graphics, it is essential to generate an appropriate representation (not only for thin lines). These problems are sometimes solved by vectorization methods based on opposite-contour matching [9] instead of using a skeleton or by combining information about a skeleton and knowledge about contours to improve the final result [10]. In this last paper, Tanigawa and colleagues present a precise line-detection method that uses both the contour and the skeleton to preserve both the line connectivity and the shape of the original line, but does not mention how this method treats shapes other than thin ones (filled shapes, text areas, etc.). Nevertheless, they provide some new results that can be applied successfully.

Shih and Kasturi [11] propose an interesting method to take care of solid regions. It is based on the Maximal Square Moving (MSM) algorithm, but it seems that it was not used to treat the large filled areas during the entire process. Furthermore, other problems (text representation, freehand drawings) still remain unsolved.

Furthermore a straight line segment is not always the most flexible feature for drawings that may contain curves. Nonrectilinear shapes that are numerous in hand-drawn entities are poorly represented, and problems occur when their relationships with other parts of the drawing (especially links between straight lines and curves – see Fig. 2) are to be stored. Few studies tackle the problem of the extraction and of the coding of curves [4]. For this, the Orthogonal Zig-Zag (OZZ) and Sparse Pixel Tracking (SPT) algorithms of Wenyin and Dori [12] seem to provide quite good solutions.

All these examples show that the bitmap image has to be simplified in order to build a representation of its contents. This transformation should have some properties that facilitate:

- The conservation of the information that is contained in the image
- The reduction of the data storage
- The simplification and the adaptability of the new representation space to further processing.

It is difficult to develop a representation that respects these three constraints. In practice, a compromise has to be reached in which the most important aspects of the problem are taken into account first.

A common remark that can be made about these vectorization methods is that they all need to segment text or solid shapes. This segmentation is often based on heuristical knowledge to separate the various kinds of shape layers constituting the drawing. Most of the time, this phase uses nonoptimal criteria, and furthermore, after this separation phase, each layer is processed separately by various methods. Of course, these methods are adapted to the kind of data that needs to be extracted, but in fact, various representations are needed to describe all the shapes that are present in the image, which differ a great deal. The comparison and the combination of the results from each layer is then a difficult task, and back-tracking is impossible without restarting the entire analysis.

3 A representation using vectors and quadrilaterals

The proposed method has been implemented on both a HP Unix workstation and a PC Pentium III (450 MHz, Windows 98). The interpretation process starts from the binary raster image of the drawing generally scanned at 300 dpi and is made up the following steps:

Vector_Construction();
Do
  Vector_Merging();
Until number of merging = 0;
Quadrilateral_Construction();
Do
  Quadrilateral_Sorting();
  Quadrilateral_Merging();
Until number of merging = 0;
Graph_construction();