

Adaptable Vectorisation System Based on Strategic Knowledge and XML Representation Use

Delalandre Mathieu¹, Saidali Youssouf¹, Ogier Jean-Marc², Trupin Eric¹

¹ PSI Laboratory, University of Rouen, 76 821 Mont Saint Aignan, France
E-mail: {mathieu.delalandre ; youssouf.saidali ; eric.trupin}@univ-rouen.fr

² L3I Laboratory, University of La Rochelle, 17042 La Rochelle, France
E-mail: jmogier@univ-lr.fr

Abstract

This paper presents a vectorisation system based on the use of *strategic knowledge*. We present here our first advancements concerning the experimentation of this system. This one is composed of two parts: a processing library and a graphics user interface. Our processing library is composed of image pre-processing and vectorisation tools. Our pre-processing tools allow to deal with noisy images. Our vectorisation tools are of high granularity level in order to permit to use them within a strategic approach. Our graphics user interface is used for the *strategic knowledge* acquisition and operationalisation. This graphics user interface permits to construct and execute scenarios, exploiting any processing of our library, according to the documents' contexts and the users' adopted strategies. A XML data representation is used, allowing an easy data manipulation. A scenario example is presented for symbol recognition on utility maps.

Keywords: pre-processing, vectorisation, strategic knowledge, graphics user interface, XML

1 Introduction

Vectorisation is used for different purposes like: graphics document recognition, handwriting recognition, and so on. Vectorisation systems use two main types of knowledge: *descriptive knowledge*, and *strategic knowledge*. The first one concerns the objects' descriptions into documents, and the second one concerns the chaining relations between image processing tools. The *descriptive knowledge* is usually employed, but not the *strategic knowledge*. This paper presents a vectorisation system based on the use of *strategic knowledge*. We present here our first advancements concerning the experimentation of this system. This one is composed of two parts: a processing library and a graphics user interface. Our processing library is composed of image pre-processing and vectorisation tools. Our pre-processing tools allow to deal with noisy images. Our vectorisation processing tools are of high granularity level in order to permit to use them within a strategic approach. Our graphics user interface is used for the *strategic knowledge* acquisition and operationalisation. This graphics user interface permits to construct and execute scenarios, exploiting any processing of our library, according to the documents' contexts and the users' adopted strategies. A XML data representation is used, allowing an easy data manipulation. In the following paper, we present in section (2), an overview on knowledge based vectorisation systems. In section (3), we present our image processing library. In section (4), we present our graphics user interface. In section (5), we present the XML use in system. In section (6), we present a scenario example for symbol recognition on utility maps. Finally, in section (7), we conclude.

2 Overview on Knowledge Based Vectorisation Systems

Vectorisation is a stage of document interpretation problem [16] that is used for different purposes like: graphics document recognition (technical document [39] [1] [41], map [28], symbol [22], and so on.), handwriting recognition (especially Chinese handwriting [12]), and so on. It is a well-known problem and many commercial applications exist [25]. From our point of view, a vectorisation system can be decomposed in two parts: a processing part (vectorisation) and a system part (the control).

Vectorisation extracts the vectorial data from document images [1]. These vectorial data correspond to mathematical object graphs composed of: vectors, circles, and curves. Vectorisation is a complex process that may rely on many different methods [9]. Some of them perform vectorisation in two steps [42]. The first one extracts object graphs. Various approaches may be used [21] like: contouring, skeletonisation, and run decomposition. The second one transforms object lists into mathematical object lists [34]. Some other methods perform directly vectorisation like: tracking methods [39], object segmentation methods [24], and mesh based methods [43]. Several overviews on vectorisation can be found in [1], [9], [21], [42], and [39].

The control system may use various approaches in order to supervise the vectorisation process. These approaches come from pattern recognition and artificial intelligence domains. Among them, let's cite: the knowledge based approaches [3] [14] [7], the multi-agent [15] or blackboard [1] systems, the perceptive approaches [47] [27], and so on. These systems are generally used for recognition and interpretation, but can have other applications like: learning [23], indexing [46], structuring data [44], and so on. So, systems deal with two main properties: the knowledge use [3] [14], and the automatic control of processings [47] [27] [1] [15]. The last ones systems deal currently with the both [1]. Several overviews on vectorisation systems can be found in [1], [41], and [28].

Like document interpretation systems, vectorisation systems use two main types of knowledge: *descriptive knowledge* and *strategic knowledge* [37]. The first one concerns the objects into documents and the relations between them. The second one concerns the image processing tools used to construct the objects and chaining relations between these tools. The descriptive knowledge is usually employed in vectorisation systems [32]. At the opposite, the strategic knowledge is less used, whereas there exist many common steps among vectorisation process [9]. In the following of this paper, we propose a strategic knowledge based vectorisation system. It is composed of a processing library and a graphics user interface for scenario construction and their execution.

3 Image Processing Library

3.1 Introduction

All our image processings are included in the PSI¹ Image Processing Library, which is composed of image pre-processings and vectorisation processings. This library is included in a complete document image recognition library, the PSI Library. A description of some library's processings can be found in [8], and a short presentation of the library in [7]. We use the pre-processing and vectorisation in order to construct pre-processing and vectorisation scenarios. The image pre-processings can be used on grey-level and binary images. We develop its content in section (3.2). The vectorisation processings are based on various approaches (skeletonisation, contouring, region and run decomposition, direct vectorisation). We have decomposed the classical vectorisation chain into granular processings (Table 1) in order to use them within a strategic approach. Our vectorisation processings are decomposed in three levels: an image level, a structured data level, and a boundary level between the image data and the structured data. We present each level in sections (3.3), (3.4), and (3.5).

Image	Skeletonisation Adaptation, Contouring, Skeletonisation
Boundary	Progressive Object Simplification, Pixel List Extraction, Direct Vectorisation, Direct Contouring, Run Decomposition, Region Decomposition
Structured Data	Interiority Degree Segmentation, Junction Reconstruction, Graph Construction, Polygonisation, Curve and circle fitting, Pruning, Merging, Smoothing, List Filtering

Table 1: data levels of vectorisation processings

¹ Perception Systems Information Laboratory: <http://www.univ-rouen.fr/psi/>

3.2 Image Pre-processing

The image pre-processing level is composed of different methods for noisy images. Firstly, we use grey-level filtering methods on scanned images like median filter and mean filter [30]. Next, we binarise our images. We use two standard algorithms for the automatic computation of binarisation thresholds: Otsu's method [29] and Kittler's method [17]. The first one is a histogram based method and the second one a clustering based method [35]. These two methods have been used for map pre-processing [18], in order to segment the network part from the cadastral part, according to the parts' grey-levels. The Figure 1 (a) gives an example of map (left) and segmented network (right). Then, we use two methods for noise reduction on obtained binary images. The first one is a connected component filtering method based on blob coloring algorithm [4], which uses automatic or pre-defined user surface threshold. The automatic threshold computation is based on maximum proportional ratio search of connected components' surfaces. The second one uses mathematical morphology operations like: dilatation, erosion, opening, and closing. These operations use classical masks 3*3 sized and of "+" "X" type. Finally, in order to test our pre-processing scenarios, we use distance computation functions between images (Hausdorff, Vinet, Kermad) [13]. The Figure 1 (b) gives an example of pre-processing scenario result with a scanned grey-level image (high) and the result binary image (low) after a median filtering, an Otsu's binarisation, an automatic connected component filtering, and an opening.

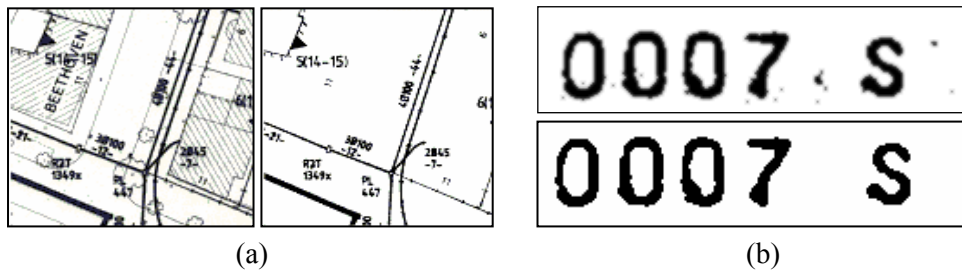


Figure 1 : (a) Otsu's binarisation (b) pre-processing scenario result

3.3 Vectorisation's Image Level

The vectorisation's image level uses classical image processings for the contouring and the skeletonisation. For the skeletonisation, we use two standard algorithms: Di Baja [10] and Taconet [40]. These skeletonisation algorithms are based on the medial axis transform [1]. For the contouring, we use a classical neighbouring test method with classical masks 3*3 sized and of "+" "X" type. The Figure 2 (a) gives an example of overlapped results of skeletonisation and contouring. The main problem of the skeletonisation is the noise which depends on the lines' thickness of images. In order to adapt images for the skeletonisation, we use the 3-4 distance transform of Di Baja [10] in combination with a classical image reducing tool. The Figure 2 (b) gives an example of the skeletonisation's impact without any reduction (middle) and with a reduction (right).

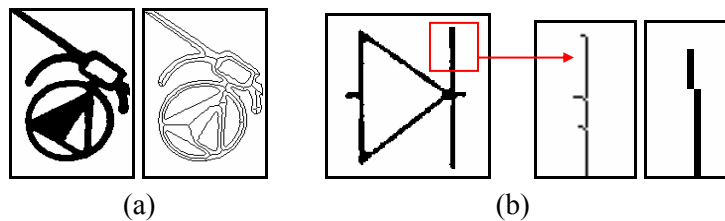


Figure 2: (a) skeletonisation and contouring (b) skeletonisation adaptation

3.4 Vectorisation's Boundary Level

3.4.1 Introduction

Our vectorisation's boundary level uses different methods in order to extract structured data from images. We present each method in the six next sections.

3.4.2 Direct Contouring

For our direct contouring, we use a processing based the blob coloring algorithm use [4]. During the connected component labelling, the internal and external shapes' contours are extracted. Then, the contours' pixels are structured into chains. In a following step, the inclusion relations between connected components are searched. With this application, we extract inclusion graphs which contain the external and the internal contour chains, and the inclusion relations between the connected components. This method gives global/local descriptions of image's shapes [9]. The Figure 3 (a) gives an example of raster representation of extracted external contours.

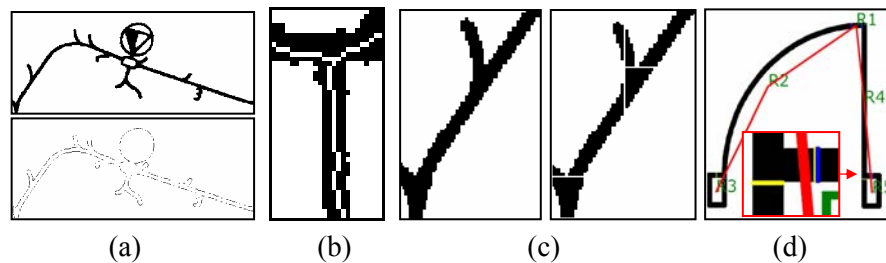


Figure 3: (a) direct contouring (b) direct vectorisation (c) run decomposition (d) region decomposition

3.4.3 Direct Vectorisation

This method has been used for the roads extraction on cadastral maps [26]. Firstly, image is analyzed to find an entry point. The following tracking process is two types: line tracking and junction tracking. The line tracking uses a point element for the line analysis. The point element advances into the line's middle according to the contours following. The displacement's length is proportional to the line's thickness. An equivalent method can be found in [38]. The junction tracking analyses the connection breaks of contours following. In this case, a circle including the junction is used to find the other junction's lines. The crossings between the lines and the circle are searched. From these data, a image's graph is updated and used to start the tracking of new lines. The Figure 3 (b) gives an example of raster representation of line and junction extraction.

3.4.4 Run Decomposition

This method is under realisation. In a first step the image is encoded into runs. Then, the runs are structured into run graphs [48]. In these graphs, the successive runs are represented by run chains and the 3-connected runs (or more) by junctions. The encoded runs can be horizontal and vertical [5]. From these run graphs, the contours and skeleton can be extracted [11]. The Figure 3 (c) gives an example of mixed raster representation of a horizontal and vertical run graph. The junction nodes are represented in white.

3.4.5 Region Decomposition

We use region decomposition method based on a wave aggregation. Firstly, the image is analyzed to find an entry point. Then, the method searches the neighbouring points. These points are labelled, aggregated, and stored into a wave object. Successively, the previous waves are used for the new aggregation processes. The wave breaking and stopping cases define the regions' boundaries. The boundaries are next used in order to

create entry waves for the new region search. From the labelled region map, a primary region graph is created. In a following step, this primary region graph is analysed in order to construct the line and junction regions. From these region graphs, the contours and skeleton can be extracted. Also, graphs' regions can be processed by statistical recognition methods [2]. The Figure 3 (d) gives an example of constructed primary region graph from a labelled region map. The zoomed zone shows the boundary lines (clear lines), and a started wave (dark line).

3.4.6 Pixel List Extraction

The pixel list extraction allows to convert the skeleton and contour images into structured data. This tool is based on the connected pixel destruction. Firstly, all 3-connected pixels are destroyed. After, we chain the connected pixels into pixel list. Each list is composed of 1-connected pixel (extremity) and 2-connected pixel. The Figure 4 (a) gives an example of pixel list extracted.

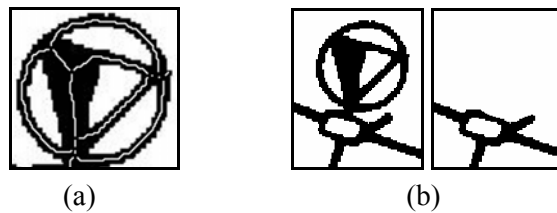


Figure 4: (a) pixel list extraction (b) object simplification

3.4.7 Object Simplification

The object simplification allows to exploit the structured data on images. We use this approach to simplify processed images during vectorisation scenarios [38]. Currently, this tool runs only with the circle objects. The Figure 4 (b) gives an example of circle object erasing result.

3.5 Vectorisation's Structured Data Level

3.5.1 Introduction

The structured data level is the central part of vectorisation scenario. Indeed, all the processings use the same input/output data type. Our data type represents geometrical object graph base composed of circles, curves, polylines, pixel lists, and junctions. In practice, we can call any processing in any order, but in theory, some processings depend on a minimum structuring degree of data (for example the circle fitting depends of polygonisation results). All boundary level's processings (see section 3.4) export their data in this type. The goal of structured data level's processings is to add semantic information to basic graphs obtained by the boundary level's processings. For that, we use different granular processings (see Table 1). We can decompose these processings in two types: lists processing and graphs processing. We present these two types in the two following sections.

3.5.2 Lists Processing

The lists processings are used for different purposes like interiority degree segmentation and mathematical approximation (vectors, circles, and curves).

For the interiority degree segmentation, we apply a thickness segmentation threshold based on a simple test of the thickness's variation. Information of the pixels' interiority degrees is obtained by the successive calls of skeletonisation/pixel list extraction tools. The Figure 5 (a) gives an example of interiority degree segmentation, with the original image (left), the graphics representation of pixel list before segmentation (middle) and after segmentation (right).

For the polygonisation (transformation of pixel lists into vector lists “polylines”), we use the “standard” Ramer’s method [33], with the “standard” split & merge Pavlidis’s method [31]. For each polyline, we compute some attributes, like the vectors’ thickness (from original pixel lists), the vectors’ lengths, and the angular relations between two consecutive vectors according to the polyline tracing. Our circle fitting algorithm is a basic tool only based on the test of angular and length relations inside a polyline. We use here a standard geometrical property: a circle can be approximated by a regular (in length and angle) polygon. The Figure 5 (b left) gives an example of polygonisation and circles fitting applied on a contours image. Recently, we have extended the geometrical object extraction with the standard Bernstein’s curve approximation [20] of pixel lists. The Figure 5 (b right) gives an example of Bernstein’s curve approximation.

Furthermore, we use some post-correction processings on the lists. In one hand we use a list filtering processing. The Figure 5 (c left-middle) gives an example of list filtering. The used threshold can be automatically computed, or pre-defined by the user. The automatic computation is based on maximum proportional ratio search of lists’ lengths. This processing decreases the complexity of junction detection method (section 3.5.3). In the other hand we use a list smoothing algorithm. The Figure 5 (c middle-right) gives an example of list smoothing. It is based on variation analysis of Freeman directions into the pixel lists. This processing increases the data qualities for the polygonisation and curve fitting.

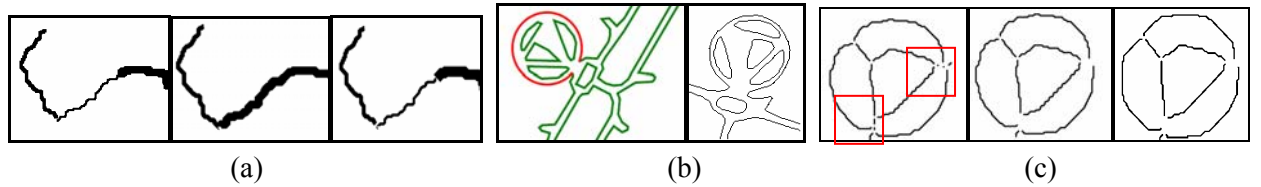


Figure 5: (a) interiority degree segmentation (b) mathematical approximation (c) pixel list correction

3.5.3 Graphs Processing

An important processing of structured data level is the junction detection. We base our approach on a junction reconstruction algorithm. This algorithm constructs all the connections between the objects’ extremities according to a Euclidian distance threshold. This threshold can be defined by the user or automatically computed. This automatic computation is based on maximum proportional ratio search of connections’ lengths. We analyze all the connections to find the inter-connections (group of joined extremities). Each inter-connection constitutes a junction. During this research, we forbid the inter-connections between two extremities of a same object.

With the information about junctions, we use a graph construction algorithm to construct the geometrical object graphs composed of: pixels, circles, curves, and polylines. We exploit the information about junctions for two standard processings in the vectorisation process: pruning and merging. The Figure 6 gives a junction reconstruction example (a, b) with successive pruning (b, c) and merging (c, d).

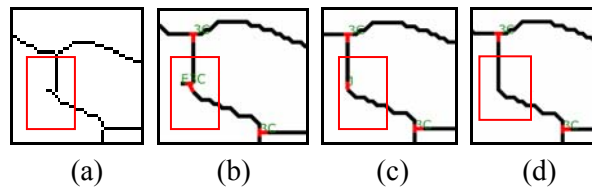


Figure 6: (a-b) junction reconstruction (b-c) pruning (c-d) merging

4 Vectorisation Scenarios Construction

We use a graphics user interface for the strategic knowledge acquisition and operationalisation: ACTI_VA² [36]. It enables to construct scenarios for document image recognition according to the documents' contexts (recognition goal, document type, and so on.), and the users' adopted strategies. In a first step, the user defines the context of the analyzed image (quality, type, and so on.) (Figure 7 (a)). A set of processing it is then proposed to the user for document image recognition. Each processing is realized from its dashboard "or processing panel" representing a scenario stage. The user oversees the scenario's construction, following permanently the results evolution. So, a state viewer shows the intermediate data graphics representations (image or structured data results) between each scenario stage (Figure 7 (b)). For an adaptable and evolving assistance, the user can at any time return to any previous stage in order to, modify the parameters setting, change the processing stage, seek a textual help, or display use examples. When the user reaches his purpose, he saves its scenario in a scenario base. During the scenario's construction, the user can search similar scenario examples in the scenario base with two research tools: a query language allowing to request the scenario base, and a graph-matching tool allowing to compare the scenarios' structures. ACTI_VA is based on the dynamic generation of panels (from an external description), using the TaskGuide package³. Currently, ACTI_VA exploits the PSI Library's processings (section 3), but it can be adapted to other libraries.

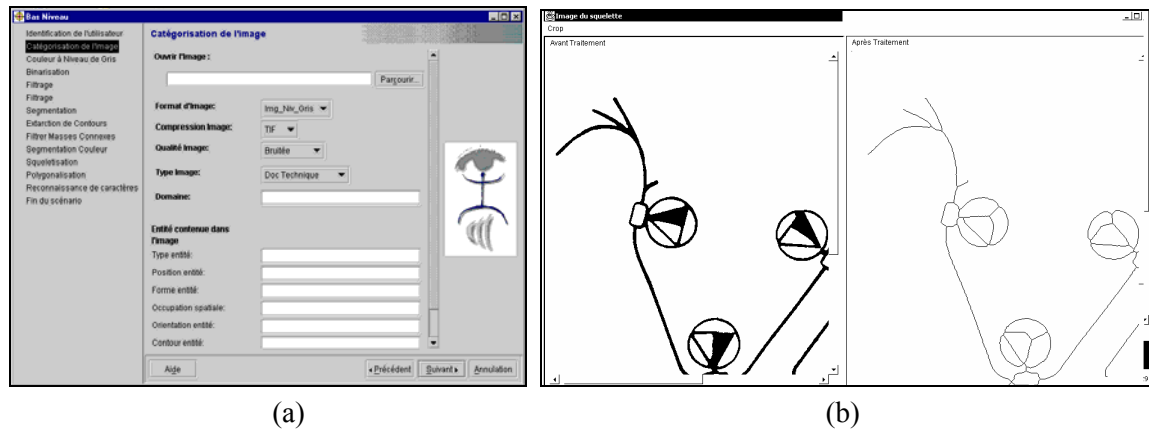


Figure 7 : (a) document's context acquisition (b) graphics representation of result

5 XML use in System

Our two system's parts (processing library and graphics user interface) use the XML⁴ language for the knowledge representation. The processing library uses it for the structured data output of processings. The graphics user interface uses it for the scenario storage into an XML base. The use of this data representation language offers several possibilities and as much as enhancement for our system [45]:

- XML is a meta-language because it is defined as a root language, which enables to define specialized sub-languages. We use SVG⁵ in the system for our data graphics representations. We also use XGMML⁶ for the graphs' description provided by some PSI Library's processings.

² in French: "Acquisition de Connaissances Traiteur d'Images pour leur VAlorisation"

³ TaskGuide: <http://www-3.ibm.com/>

⁴ eXtensible Markup Language: <http://www.w3.org/>

⁵ Scalable Vector Graphics

⁶ eXtensible Graph Markup and Modelling Language

- XML uses transforming processors. These processors easily transform the XML data streams with the help of XSLT⁷ scripts. This enables easy data communications between the processings on the one hand, and between the processings and our graphics user interface on the other hand. Furthermore, this enables to create the data graphics representations from XML output streams of processings.
- XML enables to request the XML file bases with XML-QL⁸. We use one of the candidate languages for XML-QL: Quilt⁹ [6]. We use Quilt for the management of scenario bases and the reconstruction of final XML documents (see section 6.5). The use of Quilt with XSLT enables to transform scenarios into graph base for the structural classifiers of the PSI Library [8] (see section 4).

6 Scenario Example

6.1 Introduction

This system is under experimentation, the objective is to test this system during the GREC's symbol recognition contest¹⁰. We present here a scenario example, for a recognition application on the FT's utility maps¹¹. Other applications on these utility maps have already been presented into some related papers [2] [8] [18]. We have two main components on these utility maps (Figure 8 (a)): the characters, and the graphics parts. We present here the graphics part recognition [18]. We report the reader to [2] for the description of character recognition. Also, we have two main components on these graphics parts (Figure 8 (a)): the symbols and the network. The symbols represent technical equipments permitting the connections on the network. We have three symbols' classes (Figure 8 (b)): room, CP1¹² and CP2. Our graphics recognition scenario may be decomposed into four steps: pre-processing, network's vectorisation, zone of interest research, and XML object reconstruction. We present each scenario's step in the four next sections.

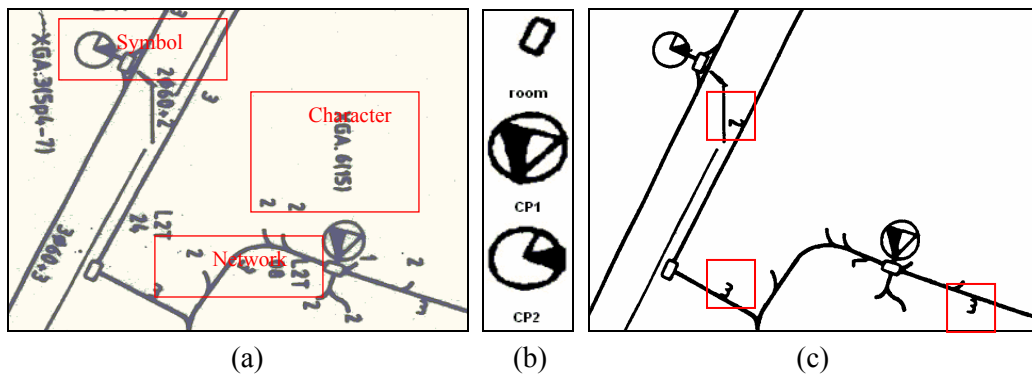


Figure 8: (a) FT's utility map (b) FT's symbols (c) pre-processing result

6.2 Pre-Processing

In a first step, we use a pre-processing scenario (see section (3.2)) on scanned images in order to, reduce the result noise of the acquisition stage, and to segment the characters and the graphics parts from FT's utility maps. We have applied successively a median filtering, an Otsu's binarisation, an automatic connected component filtering, and an opening. Thus, we have obtained clean network images (Figure 8 (c)). However, this pre-processing step cannot deal with the connected characters to the network (Figure 8 (c)'s rectangles).

⁷ eXtensible Stylesheet Transform Language

⁸ XML Query Language

⁹ <http://www.almaden.ibm.com/cs/people/chamberlin/quilt.html>

¹⁰ <http://www.cvc.uab.es/grec2003/contest.html>

¹¹ French Telecommunication operator: <http://www.rd.francetelecom.fr/>

¹² Concentration Point 1

6.3 Network's Vectorisation

Next, we use a vectorisation scenario in order to extract vectorial data corresponding to the network. Successively, we have applied a contouring, a pixel list extraction, a smoothing, a polygonisation, a split and merge method, and a circle fitting. The Figure 9 (a, and b) gives an example of network's vectorisation. The internal contours and the circle correspond to symbols (Figure 9 (b)), and the other vector lists correspond to the network (Figure 9 (a)).

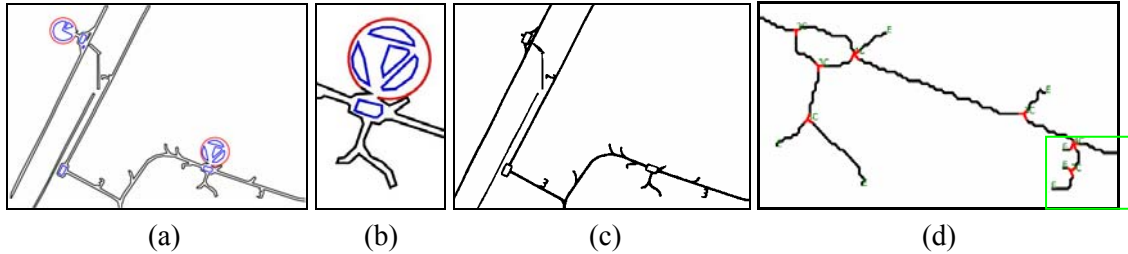


Figure 9 : (a) network's vectorisation (c) symbol's vectorisation
(c) circle simplification (d) interest zone research

6.4 Zone of Interest Research

In a third step, we use another vectorisation scenario in order to deal with the connected character research. Firstly, we use the vectorial data of the last step in order to reduce the complexity of processed images. We apply a circle object simplification (see section 3.4.7) with an automatic connected component filtering. The automatic connected component filtering erases the possible small connected components obtained after the circle simplification. The Figure 9 (c) gives an example of result after circle simplification. Next, we use the skeleton graph in order to research the zones of interest for the connected characters. We have applied successively a skeletonisation adaptation, a skeletonisation, a pixel list extraction, junction detection, a graph construction, and a pruning and merging. Finally, we use a graph tool [8] in order to search the graphs' parts corresponding to zones of interest. We construct graphs with the length information of pixel lists (two node labels: *short* and *long*), and search into the graph base the groups of connected node, which are labelled *short*. Figure 9 (d) gives an example of result. We have detected the zone of connected character "3".

6.5 XML Object Reconstruction

The two last steps give recognition results into XML format, stored into different XML files. These XML data are weakly structured: there are no links between them. In order to solve this problem, we use an XML object reconstruction step. This step organizes the different XML streams in order to reconstruct the document's objects. This reconstruction permits to extract the symbols from the vectorial data, and to structure the symbols and connected characters with the network. We use for that a reconstruction scenario based on XSLT and Quilt (see section 5). During this scenario, the internal contours and the circles are constructed as symbols, according to their inclusion relations. Then, the symbols' internal contour numbers define their classes. Indeed, these symbols can be recognized by their internal contours (or loops [8]): 4 for CP1, 2 for CP2, and 1 for room. Next, the symbols and the zones of interest are associated to their network's part. The zones of interest can be exploited later by statistical filtering methods [2]. The Figure 10 illustrates the XML reconstruction of network (a) and symbols (b).

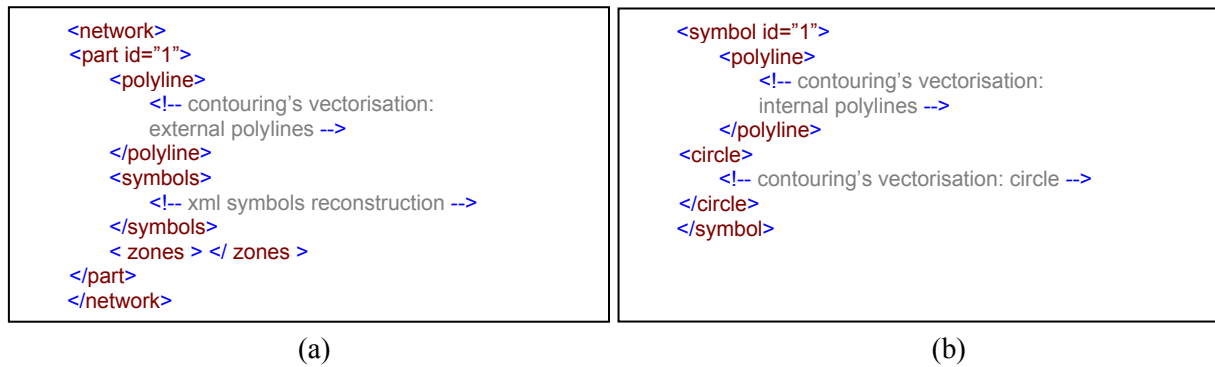


Figure 10: (a) XML network reconstruction (b) symbols' reconstruction

7 Conclusion

In this paper we have presented a vectorisation system based on strategic knowledge use. This system is under experimentation, but permits first experiments. Our processing library is composed of image pre-processing and vectorisation tools. Our pre-processing tools allow to deal with noisy images. Our vectorisation tools are of high granularity level in order to permit to use them within a strategic approach. The graphics user interface used permit to construct and execute different vectorisation scenarios. This permits to test different strategies according to the recognition goals, and to easily adapt the system to new applications. The image XML data representation is used allowing an easy data manipulation, for the graphics representation creations of results, and the XML object reconstruction.

8 Acknowledgement

The authors wish to thank Sébastien Adam (PSI Laboratory, Rouen University, France), and Mario Di Carlo (L3I Laboratory, La Rochelle University, France) for their contribution to this work.

9 References

- [1] S. Ablameyko, T.P. Pridmore, "Machine Interpretation of Line Drawing Images", *Springer-Verlag*, 2000.
- [2] S. Adam, J.M. Ogier, C. Cariou, J. Gardes, Y. Lecourtier, "Combination of Invariant Pattern Recognition Primitive on Technical Documents", *Graphics Recognition (GREC)*, 1999.
- [3] M. Ahmed, R. Kreidieh, "An Expert System for General Symbol Recognition", *Pattern Recognition (PR)*, 33 : 1975-1988, 2000.
- [4] A.K. Aggarwal, A.V. Kulkarni. A Sequential Approach to the Extraction of Shape Features. *Computer Graphics and Image Processing (CGIP)*, 6(6) : 538-557, 1977.
- [5] M. Burge, W.G. Kropatsh, "A Minimal Line Property Preserving Representation of Line Images", *Structural and Syntactical Pattern Recognition (SSPR)*, 1998.
- [6] D. Chamberlin, J. Robie, D. Florescu, "Quilt: An XML Query Language for Heterogeneous Data Sources", *Workshop on the Web and Databases (WebDB)*, 2000.
- [7] E. Clavier, G. Masini, M. Delalandre, M. Rigamonti, K. Tombre, J. Gardes "DocMining: A Cooperative Platform for Heterogeneous Document Interpretation According to User-Defined Scenarios", *Graphics Recognition (GREC)*, 2003.
- [8] M. Delalandre, S. Nicolas, E. Trupin, J.M. Ogier, "Symbols Recognition by Global-Local Structural Approaches, Based on the Scenarios Use, and with a XML Representation of Data", *International Conference on Document Analysis And Recognition (ICDAR)*, 2003.

- [9] M. Delalandre, E. Trupin, J.M. Ogier, "Local Structural Analysis: A Primer", *Graphics Recognition (GREC)*, 2003.
- [10] G.B Di Baja, "Well shaped, Stable, and Reversible Skeletons from the 3-4 Distance Transform", *Journal of Visual Communication and Image Representation*, 5(1):107-115, 1992.
- [11] K.C. Fan, W.H. Wu, "A Run Length Coding Based Approach to Stroke Extraction of Chinese Characters", *Pattern Recognition (PR)*, 33(11) : 1881-1895, 2000.
- [12] J. Fan, "Off-line Optical Character Recognition for Printed Chinese Character-A Survey", *Technical Report, University of Colombia, USA*, 2002.
- [13] V. Di Gesu, V. Starovoitov, "Distance Based Function for Image Comparison", *Pattern Recognition Letters (PRL)*, 20 : 207-214, 1999.
- [14] J.E. Den Hartog, "Knowledge Based Interpretation of Utility Maps", *Computer Vision and Image Understanding (CVIU)*, 63(1) : 105-117, 1996.
- [15] T.C. Henderson, L. Swaminathan, "Agent Based Engineering Drawing Analysis", *Symposium on Document Image Understanding Technology (SDIUT)*, 2003.
- [16] R. Kasturi, L. O'Gorman, V. Govindaraju, "Document Image Analysis: A Primer", *Sadhana*, 27(1) : 3-22, 2002.
- [17] J. Kittler, J. Illingworth, "Minimum Error Thresholding", *Pattern Recognition (PR)*, 19(1) : 41-47, 1986.
- [18] A. Lassaulzais, R. Mullot, J. Gardes, Y. Lecourtier, "Segmentation d'Infrastructures de Réseau Téléphonique", *Colloque International Francophone sur l'Ecrit et le Document (CIFED)*, 1998.
- [19] K.K. Lau, P.C. Yuen, Y.Y. Tang, "Stroke Extraction and Stroke Sequence Estimation On Signatures", *International Conference on Pattern Recognition (ICPR)*, 2002.
- [20] C. W. Liao, J. S. Huang, "Stroke Segmentation by Bernstein-Bezier Curve Fitting", *Pattern Recognition (PR)*, 23(5):475-484, 1990.
- [21] W. Liu, D. Dori, "From Raster to Vectors : Extracting Visual Information from Line Drawings", *Pattern Analysis and Applications (PAA)*, 2(2) : 10-21, 1999.
- [22] J. Lladós, E. Valveny, G. Sánchez, E. Martí, "Symbol Recognition : Current Advances an Perspectives", *Graphics Recognition (GREC)*, 2001.
- [23] Y. Luo, W. Liu, "Engineering Drawings Recognition Using a Case-based Approach", *International Conference on Document Analysis And Recognition (ICDAR)*, 2003.
- [24] J. Matas, C. Galambos, J. Kittler, "Progressive Probabilistic Hough Transform for Line Detection", *Computer Vision and Pattern Recognition (CVPR)*, 1999.
- [25] E.F El-Mejbri, H. Grabowski, H. Kunze, R.S. Lossack, A. Michelis, "A Contribution to the Reconstruction Process of Article Based Assembly Drawings", *Graphics Recognition (GREC)*, 2001.
- [26] J.M. Ogier, C. Olivier, Y. Lecourtier, "Extraction of Roads from Digitized Maps", *European Signal Processing Conference (EUSIPCO)*, 1992.
- [27] J.M. Ogier, R. Mullot, J. Labiche, Y. Lecourtier, "Semantic Coherency, the Basis of an Image Intepretation Device – Application to The Cadastral Map Interpretation", *Transactions on Systems, Man and Cybernetics, part B : Cybernetics (TSMCB)*, 30(2) : 322-338, 2000.
- [28] J.M. Ogier, S. Adam, A. Bessaid, H. Bechar, "Automatic Topographic Color Map Analysis", *System.Graphics Recognition (GREC)*, 2001.
- [29] N.Otsu, "A Threshold Selection Method from Gray-Level Histograms", *Transactions on Systems, Man and Cybernetics (TSMC)*, 9(1) : 62-66, 1979.
- [30] J.R. Parker, "Algorithms for Image Processing and Computer Vision", *Paperback editions*, 1996.
- [31] T. Pavlidis, S. L. Horowitz, "Segmentation of Plane Curves", *Transactions on Computers (TC)*, 23 : 860-870, 1974.
- [32] B. Pasternak, B. Neumann, "The Role of Taxonomy in Drawing Interpretation", *International Conference on Document Analysis And Recognition (ICDAR)*, 1995.

- [33] V. Ramer, "An Iterative Procedure for the Polygonal Approximation of Plane Curves", *Computer Vision Graphics and Image Processing*, 1(3) : 244-246, 1972.
- [34] P.L. Rosin, A.W. West, "Nonparametric Segmentation of Curves Into Various Representations", *Pattern Analysis and Machine Intelligence (PAMI)*, 17(12) : 1140-1153, 1995.
- [35] B. Sankur, M. Sezgin, "Image Thresholding Techniques : A Survey Over Categories", *Pattern Recognition (PR)*, under review.
- [36] Y. Saidali, N. Beaudoin, E. Trupin, M. Holzen, J. Labiche, "An Interactive Application for Knowledge Acquisition in Image Processing", *International Conference on Artificial Intelligence and Applications (IASTED)*, 2002.
- [37] Y. Saidali, S. Adam, J.M. Ogier, E. Trupin, J. Labiche, "Knowledge Representation and Acquisition for Engineering Document Analysis", *Graphics Recognition (GREC)*, 2003.
- [38] J. Song, F. Su, C. Tai, S. Cai, "An Object-Oriented Progressive-Simplification based Vectorisation System for Engineering Drawings: Model, Algorithm and Performance", *Pattern Analysis and Machine Intelligence (PAMI)*, 24(8):1048-1060, 2002.
- [39] J. Song, M.R. Lyu, M. Cai, and S. Cai, "Graphic Object Recognition from Binary Images: a Survey and an Integrated Paradigm", *Transactions on Systems, Man and Cybernetics, part C : Applications and Reviews (TSMCC)*, under review.
- [40] B. Taconet, A. Zahour, S. Zhang, A. Faure, "Deux Algorithmes de Squelettisation", *Reconnaissance Automatique de l'Ecriture (RAE)*, 1990.
- [41] K. Tombre, "Ten Years of Research in the Analysis of Graphics Documents, Achievements and Open Problems", *Image Processing and Image Understanding*, 1998.
- [42] K. Tombre, C. Ah-Soon, P. Dosch, G. Masini, S. Tabbone, "Stable and Robust Vectorisation : How to Make the Right Choices", *Graphics Recognition (GREC)*, 1999.
- [43] P. Vaxiviere, K. Tombre, "Knowledge Organisation and Interpretation Process in Engineering Drawing Interpretation", *Document Analysis Systems (DAS)*, 1994.
- [44] M. Weindorf, "Structure Based Interpretation of Unstructured Vector Maps", *Graphics Recognition (GREC)*, 2001.
- [45] K. William, M Brundage, P. Dengler, J. Gabriel, A. Hoskinson, M. Kay, T. Maxwell, M. Ochoa, J. Papa, M. Vanmane, "Professional XML DataBases", *WorkPress, publishing*, 2002.
- [46] M. Worring, B. Wielinga, F. Verster, A. Anjewierden, L. Todoran, S. Kabel, R. de Hoog, "Automatic Indexing of Text and Graphics in Technical Manuals", *International Conference on Multimedia and Expo (ICME)*, 2001.
- [47] S. Yoon, G. Kim, Y. Choi, Y. Lee, "New Paradigm for Segmentation and Recognition", *Graphics Recognition (GREC)*, 2001.
- [48] S. Di Zenzo, L. Cinque, S. Leviadi, "Run Based Algorithms for Binary Image Analysis and Processing", *Pattern Analysis and Machine Intelligence (PAMI)*, 18(1) : 83-89, 1996.