Project Number:	FP6-516059
Document number:	
Document Title:	DR_FRESH_D2.2.4_Consideration on recognition for “dynamic-sized” equipment component
Document status:	
Date:	13/10/06
Availability:	

Abstract Consideration on recognition for “dynamic-sized” equipment component

Keyword List WP2, image segmentation

CONTENTS

1 INTRODUCTION

Symbol recognition is a field within graphics recognition to which a lot of efforts have already been devoted. However, a document analysis expert who is more familiar with OCR might rightfully wonder what exactly we call a symbol and how symbol recognition differs from basic character recognition.

In this report we will outline the method we propose to develop for the recognition of the equipment present in the drawings submitted to us within WP2 of the Fresh project. The method is based on thorough analysis of existing symbol recognition methods and algorithms; several excellent surveys are available on that matter.

2 DEFINITION OF THE PROBLEM

We consider graphical layers of documents where image segmentation has already been performed: characters and lines have been separated using the methods developed within tasks 2.1 and 2.2 (see deliverables D.2.2.1 and D.2.2.2). We therefore can assume that the images processed are relatively clean. As a consequence of common work by the partners to select the typical area of drawings which are representative of the data to be processed (see deliverable D.2.6.1), we have outlined the following features of the problem to be solved:

- There are about 500 different symbols to be recognized.
- Most of these symbols are compositions of simpler geometrical shapes. Symbol recognition should be able to deal with such *complex symbols* in order to be of practical use.

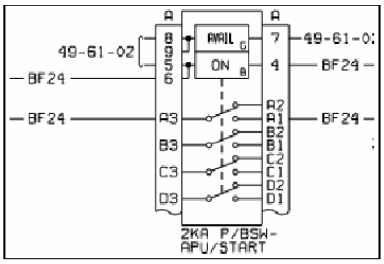

Figure 1: Examples of complex symbols.

- We should be able to discriminate between symbols which may differ not by their graphical shape, nor by their topology, but simply by the number of connectors or by the type of textual annotations.

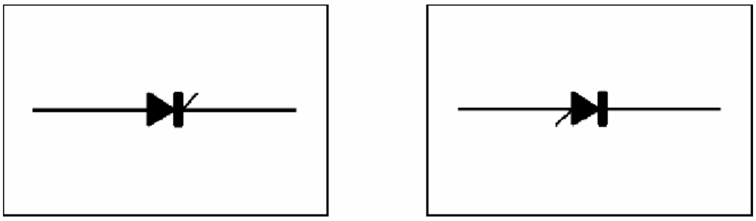

Figure 2: Example of very similar symbols.

In the present deliverable report, we will concentrate on the method to be used for the graphics recognition part, i.e. for recognizing the geometrical shapes making up a symbol. Ultimately, this will have to be combined with the result of text recognition, dealt with in task 2.3, and with the information yielded by segmentation (task 2.2) about connectors and wiring paths, in order to achieve full recognition of the dynamic-sized components.

3 ELEMENTS FOR THE CHOICE OF THE RECOGNITION METHOD

3.1 STRUCTURAL METHODS AND GRAPH MATCHING

Symbol recognition is especially well suited to using structural pattern recognition techniques, unlike usual statistical classification techniques. When dealing with specific families of symbols, techniques similar to OCR could be used; this is the case for symbols having all a loop or for music recognition. However these techniques have their own limitations, in terms of computational complexity and of discrimination power, and in the present case we have a family of symbols which is much too large and diverse for these techniques to be suitable. Actually, *graph matching techniques* are especially adapted to the specificities of symbol recognition.

Although simple, the basic idea of graph matching still suffers from a number of drawbacks:

1. Sensitive to errors and noise. To deal with this problem, error-tolerant subgraph isomorphism algorithms have been proposed.
2. Computational complexity of subgraph isomorphism methods. To deal with this, we need to use simple graphs, i.e. not to build the graphs by ``crude'' methods based on pixels or segments, but A lot of efforts have therefore been devoted to optimizing the matching process through continuous optimization or constraint propagation techniques to perform discrete or probabilistic relaxation.
3. The scaling of such structural methods to encompass a large number of candidate symbols

Therefore, we propose to integrate graph matching, which remains a very powerful tool for capturing the diversity of topological and geometrical relations, into a combined recognition method which includes the robust recognition of basic geometrical shapes; thus structural recognition will only be one part of the whole process, used where it is most suitable.

3.2 GENERIC VS. AD HOC RECOGNITION METHODS

As the project aims at delivering general-usage methods, we are aware that methods built on manually set thresholds are not suitable, as they would be hard to adapt to various kinds of documents and aircraft types. The general aim is to have as few thresholds as possible, and to master the setting of those which are unavoidable. *Generic methods* are well suited to this aim as they require few thresholds which can generally be automatically defined. Our first experiments were therefore with a generic raster-to-vector method developed in our group and which yields very good results for detecting arcs and segments in line drawings, based on the skeleton of the image and a robust statistical fitting method. Nevertheless, due to its genericity, small parts of the skeleton were ignored by the method in some cases.

The solution consists in going for a generic description of complex symbols, based on graphs connecting basic geometrical components, on one hand. On the other hand, for efficiency in the present application, the extraction of the basic geometrical components themselves will be done through an ad hoc method for the specific case dealt within this project, as will be explained later in this document. In this way, the general framework will be reusable in different contexts, whereas the operators for extracting basic shapes can be interchangeable, depending on the required specificities.

3.3 THE RIGHT INFORMATION IN THE RIGHT PLACE

Despite their limitations, structural recognition methods provide powerful tools for dealing with complex information. This stems from the large representational power of a graph, as a structure to capture pieces of information and the relationships between these pieces. Attributed relational graphs (ARG) are especially suitable for supporting the structural representation of symbols. But it is a challenge to *put the correct information into the graph*. A typical natural, but often simplistic, and sometimes even wrong way of proceeding is to use the result of some raster-to-vector process to build a graph where the vertices would be the vectors and the nodes the junctions between the vectors. This leads to representing a symbol as a set of graphical features and the spatial relations between these features, represented by relational attributes. Adding higher-level topological, geometrical and relational information to the nodes and vertices of the graph can open up new possibilities in recognition problems. When some pre-segmentation methods can divide the image to be analyzed into homogeneous regions, region adjacency graphs are a good candidate as they capture a lot of interesting information. When this is not possible, it may make sense to start with extracting simple graphical features which can be reliably found without prior segmentation: vectors, arcs, basic shapes, and to use a graph where these basic features are attributes of the nodes and the vertices convey information about topological and geometrical relationships between these features (inside, above, at-right-of, touching, etc.)

4 DECOMPOSITION INTO BASIC SHAPES

In this section we will describe the main elements of the decomposition method which has been designed and is currently under implementation, for task 2.4 of workpackage 2. The method is based on the analysis of the specificities of wiring diagrams, from the set selected for that purpose. The main considerations of this analysis are:

- most lines are horizontal or vertical;
- There are numerous rectangles;
- There are a number of small shapes (circles, triangles...)

Ultimately, we will have to perform graph matching to discriminate between various symbols.

4.1 RECTANGLES

Rectangular shapes are extracted using a robust and reliable operator. For this, templates of the four rectangle corners are searched for and matched to form rectangles.

Once all these rectangles are located, they are extracted to a sub-layer, which means that the pixels they cover are removed from the image layer to decrease processing time of the next steps as well as the possibility of miss-classification. The same process is undertaken with the other shapes. Figure 3 illustrates the extraction of rectangles on a drawing.

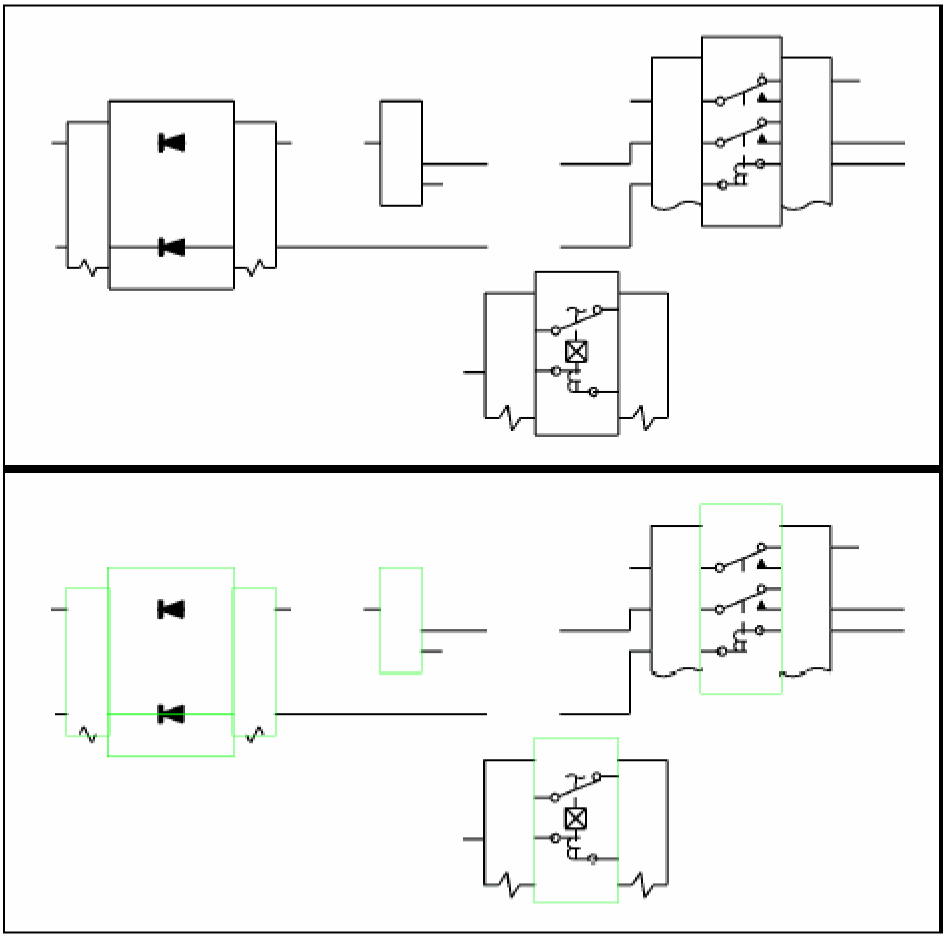

Figure 3: Example of rectangle extraction (original and the result in green).

4.2 SMALL OCCLUSIONS

A simple and efficient analysis of the white connected components allows us to extract to a specific sub-layer all small occlusions. Further compacity measures can lead to a discrimination between circular and angular shapes. When the white connected component is identified, the surrounding black component must be removed.
Figure 4 illustrates on an example how the occlusions are detected.

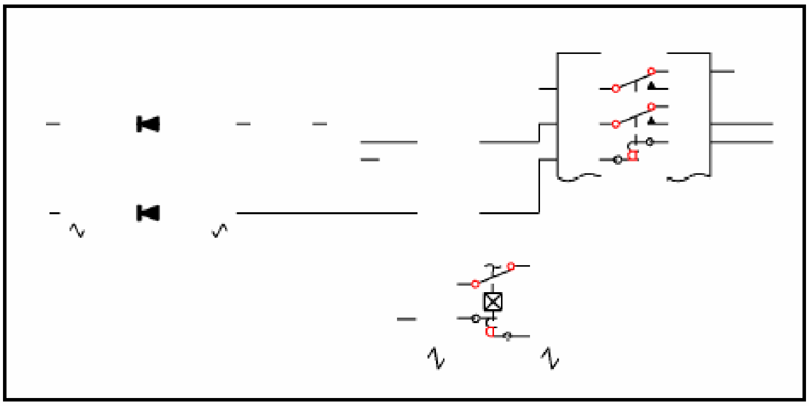

Figure 4: Example of small occlusions detection.

4.3 FILLED SHAPES

On the remainder, filled shapes are considered to be parts of the image which have a thickness larger than T_f pixels. The extraction of this sub-layer is based on simple mathematical morphology operators.

On the thick layer, the objects can be represented by their contours for further processing. Figure 5 illustrates the extraction of filled shapes on an example.

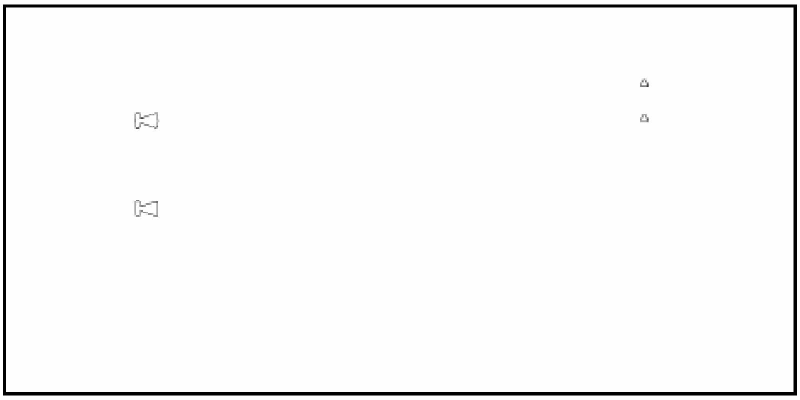

Figure 5: Example of filled shapes extraction.

4.4 CIRCULAR ARCS

On the remainder, after removal of all the previous objects, a skeleton is computed. To guarantee the precision of the skeleton, we advocate the use of chamfer distances, which come closer to approximating the Euclidean distance.

The skeleton is linked to form a graph where each node corresponds to the extremity of a curve. Each of these digital curves is further split into fuzzy segments with fixed orders. An algorithm under development analyzes these segments to provide hypotheses for arcs on one hand and segments on the other.

4.5 HORIZONTAL AND VERTICAL LINES, AND REMAINDER

Among the curves which are not labeled as arcs by the previous step, it is of interest to extract specifically the horizontal and vertical lines, even when they are partially interrupted. For this, we scan the image until we find a black pixel and track from there horizontally (resp. vertically) to check for long enough segments covering black pixels. The homogeneity of the thickness is also checked for. After this step, the remaining lines are represented by their polygonal approximation, as explained above. Figure 7 illustrates the extraction of horizontal and vertical lines, and the remaining lines, on an example.

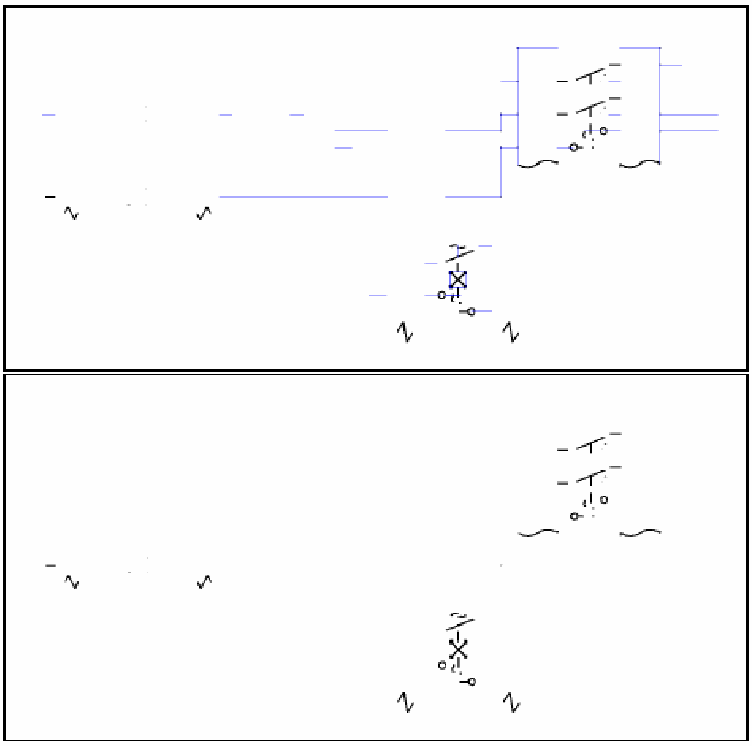

Figure 7: Example of horizontal and vertical lines extraction (the result in blue and the remainder).

5 SYMBOL RECOGNITION METHOD

During the previous decomposition step we have decomposed the symbol layer into sub-layers containing arcs and circles, rectangles, filled shapes, small occlusions, etc. These sub-layers are the basic primitives used to define the more informative representation associated to each symbol. For each symbol to be found in the database, a structural representation can be made of the way these primitives are assembled, introducing geometric rules and spatial relations into a relational graph. Due to the low number of possible features and of spatial and geometrical relations, the computational complexity should not be a problem in this graph matching method. Actually, the fact that the shapes correspond to a well-defined ``vocabulary'' should make it possible to use a matching method based on graph grammars. This will be investigated in the coming months.

Ultimately, the grammar will also have to take into account other pieces of information, of course, such as the annotations present in the text layer and the connection information from the vectorized connecting lines. It is obvious that ARG graph-based recognition methods require high time processing in broad applications. Nevertheless such methods can be suitable for our peculiar problem, due to the constraints and heuristics coming from the understanding of the documents to be processed. At last, adding some cost function between parts (e.g. small circles and arcs, which are difficult to compare) could also improve the approach by limiting the effects of miss-classifed elements.