# A framework for knowledge-based map interpretation

Jurgen den Hartog

25 september 1995

# A framework for knowledge-based map interpretation

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof.ir. K.F. Wakker,
in het openbaar te verdedigen ten overstaan van een commissie,
door het College van Dekanen aangewezen,
op maandag 25 september 1995 te 10.30 uur

door

**Jurgen Edwin DEN HARTOG**

doctorandus informatica,
geboren te Rotterdam.

Dit proefschrift is goedgekeurd door de promotor Prof.dr.ir. E. Backer.

Toegevoegd promotor: Dr.ir. J.J. Gerbrands.

Samenstelling promotiecommissie:

Rector Magnificus, voorzitter
Prof.dr.ir. E. Backer, Technische Universiteit Delft, promotor
Dr.ir. J.J. Gerbrands, Technische Universiteit Delft, toegevoegd promotor
Prof.dr.ir. A.W.M. Smeulders, Universiteit van Amsterdam
Prof.dr. H. Koppelaar, Technische Universiteit Delft
Prof.dr.ir. M.G. Vosselman, Technische Universiteit Delft
Dr. A.M. Vossepoel, Technische Universiteit Delft
Dr.ir. T.K. ten Kate, TPD-TNO

This research is funded by the TNO Institute of Applied Physics and the European Community as a part of ESPRIT (ROCKI, EP-5376).

The cover artwork is drawn by Roland den Hartog.

CIP-DATA KONINKLIJKE BIBLIOTHEEK, DEN HAAG

den Hartog, J.E.

A framework for knowledge-based map interpretation /
Jurgen Edwin den Hartog. - [S.l. : s.n.]. - Ill.
Thesis Technische Universiteit Delft. - With ref. - With
summary in Dutch.
ISBN 90-9008476-2
Subject headings: knowledge-based systems /
map interpretation / image processing.

# Contents

# Introduction

# 1

# Introduction

In the industrialized world, there is an urgent need for solutions to drawing conversion. Drawing conversion is the process of converting paper maps and drawings into a digital format which captures all relevant information. The information in maps, for example, can be digitally stored in a dedicated database called a Geographic Information System (GIS) whereas electric diagrams can be represented in a CAD format. Considering the enormous amount of paper information and its importance to a variety of industrial branches, such as public utilities and the construction and transport industry, efficient conversion of maps is a topic of substantial interest to our society. Because both purpose and nature of the conversion process differ with each application, there is an urgent need for generally applicable techniques for (semi-) automatic drawing conversion. However, automatic drawing interpretation is not an easy task. A digital image consists of an array of pixels and, although simple for a human, it is not obvious how to extract the image contents from this format.

The goal of the research described in this thesis is the design and development of techniques which are capable of automatic drawing interpretation. Because of the multitude of map applications, the flexibility of these techniques is a very important aspect of our research. Moreover, the management of a company, such as a public utility, depends heavily on information about the infrastructure. The results of automatic interpretation should therefore be very reliable. To obtain a reliable interpretation it is necessary to utilize the available knowledge about the application. Since maps are drawn according to specific drawing rules using a limited set of symbols, for each type of map there is much knowledge available prior to the interpretation. Besides a priori knowledge about the map itself, there is also knowledge about applicable image processing functions and the interpretation strategy. For optimal interpretation results, it is vital to represent all types of knowledge. The knowledge representation format should therefore provide for simple but effective means to model and manipulate the knowledge. In addition to a knowledge representation formalism, a mechanism is required to control the interpretation process using all available knowledge. The interpretation of an image is composed of a large number of image processing steps. The control mechanism should be able to make inferences, based on the a priori knowledge and intermediate results, and guide the interpretation in promising directions by execution of appropriate image processing techniques with the proper parameter settings. In summary, the purpose of this study is the development of a flexible and generally applicable framework for knowledge-based map interpretation. To obtain a reliable and accurate description of the map contents, the framework should provide methods to model and to manipulate various kinds of knowledge as well as a reasoning mechanism to control the image processing steps.

The first two sections concentrate on some important aspects concerning the design of such a framework. The fundamentals in the design of a control mechanism for image interpretation are introduced in Section 1.1. Some basic aspects of knowledge representation are discussed in Section 1.2. Section 1.3 gives a detailed description of the application which has been chosen to evaluate the techniques and concepts developed during this study. The chapter is concluded with an outline of the thesis.

## 1.1 Image interpretation

Image interpretation is the process of understanding the contents of an image. In this process, a two or three dimensional array of gray values has to be transformed into a detailed high-level representation describing the individual components and their relationships in the image. In each

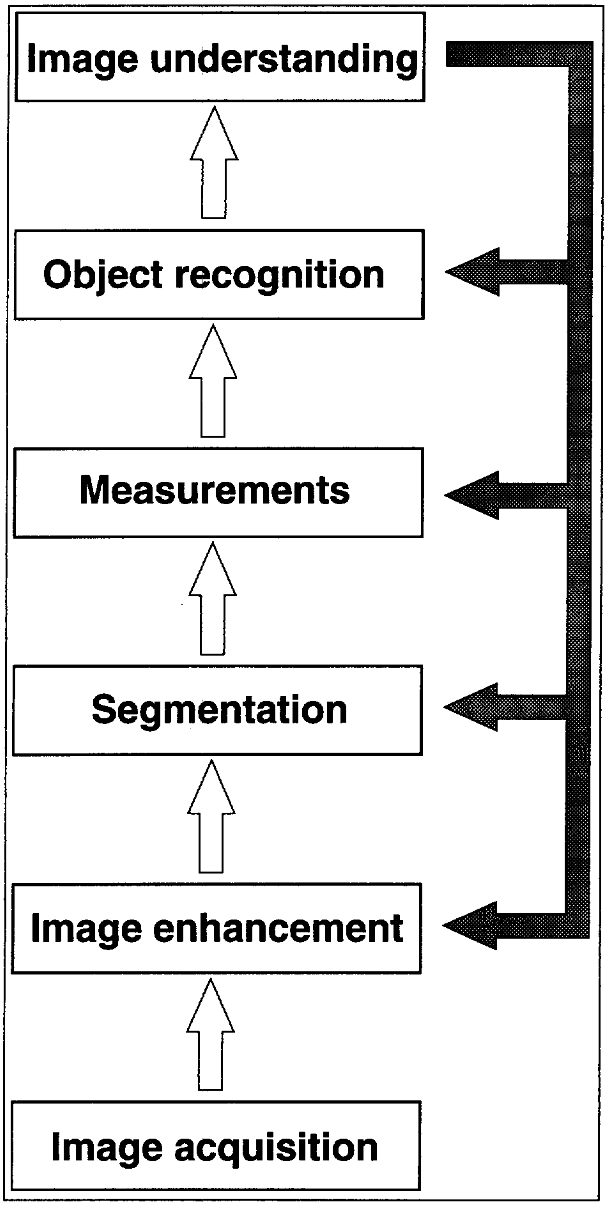



**Figure 1.1**: *In this image interpretation model the white arrows represent a strict bottom-up information flow, whereas the grey arrows denote the top-down control of the processing steps.*

step of the interpretation, the enormous amount of low-level pixel information has to be reduced, classified and reorganized into an increasingly complex data structure. Many operations are involved in this complex process and it is therefore necessary to design a control strategy to organize the combined effort of the image processing operations. The key question in the design of such a control strategy is whether the image data or a priori knowledge should control the interpretation process. The remainder of this section is devoted to this issue. For a general introduction to both image processing and machine vision, the reader is referred to *Digital image processing* by Gonzalez and Woods [2], *Computer and robot vision* by Haralick and Shapiro [3, 4], and *Image processing, analysis and machine vision* by Sonka, Hlavac and Boyle [6].

### 1.1.1 The traditional bottom-up strategy

In a bottom-up processing strategy, each step is based on the results of the previous step. Fig. 1.1 shows the conventional model in image interpretation as a sequence of image operations. The white arrows represent the bottom-up information flow between subsequent steps. The gray arrows represent top-down information flow, but they are left for the moment and will be discussed in Section 1.1.3.

All image processing begins with image acquisition by means of an input device such as a scanner or a camera. Following this step, the raw image data may be further processed in an enhancement step to facilitate the processing in succeeding steps. An example of enhancement is a sharpening operation to compensate for the blur introduced during the acquisition. The information in the (enhanced) image data is captured in an enormous number of pixels. Such a low-level representation is not suitable for direct automatic interpretation and more structure is added in the segmentation step where corresponding pixels are grouped together in regions. A simple example of segmentation is thresholding; each pixel is classified as either background or foreground, depending on whether its value is below or above a fixed threshold. In the measurement step, for each segmented region, one or more distinguishing features are calculated, e.g. area, perimeter, and the optical density of a region. Based on these features, an object label can be assigned to the regions in the recognition step. In the understanding step, the multiple recognitions are combined and verified to form higher-level objects and to obtain an understanding of the image.

This rather simple strategy is favorable if the processing steps are more or less independent of the contents of a specific image and if each step yields reliable results for succeeding processing steps. However, if the image data is noisy or ambiguous, the results of a processing step may contain errors. Due to the static nature of the bottom-up approach, subsequent steps may not be able to detect and handle these errors, thus causing the errors to propagate upwards along the processing path to the final understanding step.

For more complex applications, a single algorithm within each processing step may be insufficient. For example, in an industrial application where images contain several differently shaped machine parts, multiple recognition algorithms are required. In a medical image, both areas with poor contrast and high contrast may be found, each requiring a different segmentation algorithm. However, in a bottom-up approach, no model is available to determine when a specific algorithm should be used. As a consequence, within a processing step a brute-force approach may be inevitable where all available algorithms are employed, resulting in unnecessary processing and waste of resources. In the worst case, each processing step may needlessly generate large amounts of data which have to be processed by all algorithms in the succeeding step, which, in their turn, generate even more data. When an application becomes this complex, it is obvious that a need exists for an application model to guide the interpretation process and to select the appropriate algorithms when necessary.

### 1.1.2 A top-down strategy

A top-down control strategy is based on the hypothesize-and-test concept. In this concept, the image interpretation process is based on an internal model which describes what to expect in an image. From this model, hypotheses are generated. A complex hypothesis may be split into sub-hypotheses, which in their turn can be divided further until a hypothesis can be either accepted or rejected using only the image processing necessary for its verification. The model guides the interpretation process in promising directions through generation of the proper hypotheses. Thus, a top-down approach may use the computational resources more efficiently than a brute-force bottom-up strategy. A top-down strategy therefore seems most appropriate when dealing with complex applications or large amounts of data.

This does not mean that top-down control is always superior to bottom-up processing. The top-down strategy also has important disadvantages. Since the model describes what to expect in an image, the interpretation process only tries to verify the hypotheses issued from above. The interpretation therefore only yields good results if the image contains what the model describes. If the input data changes even slightly, a top-down approach may be unable to verify a single hypothesis. A bottom-up approach, however, uses fewer assumptions about the image data and is therefore more robust to changes in image contents. Furthermore, a top-down control strategy is very complex in its design and implementation, because knowledge about the application has to be modeled and represented, while requiring an inference mechanism to operate with the knowledge on all the processing levels. Finally, top-down control alone is never sufficient for image interpretation and some bottom-up processing will always be required. This can be explained as follows. A complex hypothesis is recursively divided into sub-hypotheses and this continues until the hypotheses can be verified directly with prior information. However, in the case of image interpretation there is no prior information about the exact location or the shape of objects. This information can only be extracted if at least some segmentation is performed in a bottom-up processing step. Therefore, instead of focusing on either a bottom-up or a top-down approach, it may be more practical to develop a strategy which integrates both.

### 1.1.3 A combined approach

Neither top-down nor bottom-up processing yields a complete solution to image interpretation. As discussed above, a pure top-down approach is complex and its use is limited to the restraints imposed by the internal model. A strict bottom-up strategy may result in a brute-force approach while errors propagate along with the processing. It seems sensible to combine both strategies to get the best of both worlds. In this scenario, a bottom-up strategy is favored for initial processing and when its simplicity is sufficient. Top-down processing is needed for efficient control and to obtain additional results when bottom-up processing is inadequate or incorrect.

Fig. 1.1 shows the general idea of an approach which allows bidirectional processing. The results of bottom-up processing ascend to higher levels along the white arrows, while top-down control actions are passed to lower levels through the gray arrows. The following example illustrates the concept of bidirectional control.

If the basic segmentation step yields adequate results in general, a bottom-up approach is most appropriate because it is simple. Following segmentation, the bottom-up processing proceeds with the measurement step and the object recognition step. The understanding step is then able to verify the recognized objects with its internal application model. When an inconsistency is detected between two outcomes, the model could supply a possible cause for this conflict, e.g. a poor segmentation which obstructs the proper recognition of an object. Based on the model, the interpretation process then returns to the segmentation level to use an alternative segmentation algorithm or another parameter setting for the image part under consideration. At this point, bottom-up processing continues with resegmentation, feature extraction and object recognition. The interpretation process then resumes with the understanding step again.

From this example, it may be clear that a combination of both bottom-up and top-down control is advantageous as it provides a method to correct results from a lower-level. In this scenario, the simplicity of bottom-up processing is used when adequate and top-down control is employed when needed. Therefore, the combined approach has been chosen as the underlying concept of the interpretation strategy proposed in this thesis.

## 1.2 Knowledge representation

Knowledge is one of the key issues in this study. In this section, a short overview of three of the most basic representation formalisms is given. For a more thorough discussion on this important subject, the reader is referred to the books on artificial intelligence by Rich and Winston [5, 7].

Before we focus on the basic representation paradigms, it is necessary to make a distinction between two types of knowledge first. The two types which are generally distinguished are *declarative knowledge* and *procedural knowledge*. The former type represents knowledge as a collection of facts, sometimes accompanied by a limited set of rules describing how to manipulate them. The latter type represents knowledge about how and in which order things should be done. Each of these types of knowledge has advantages and limitations, but in general it is agreed that both types are needed and most systems therefore employ a combination of both. Within the context of image interpretation, declarative knowledge may be used to describe how objects look like, while procedural knowledge may be more appropriate to describe when to search for a specific object and which algorithm should be used.

The remainder of this section is dedicated to three basic knowledge representation formalisms.

**Semantic nets.** A semantic net is a graph-like structure where the nodes represent objects and the links describe the relationships between them. Because both objects and relationships can be of any type, the semantic net is a very useful tool to model commonsense knowledge. By means of special relationships, i.e. the *Instance of* and *A kind of* relationships, it is possible that the properties of a conceptual object are inherited by its instances. Although semantic nets are appropriate for representation of declarative knowledge, they are less suited for representation of procedural knowledge. When considering their use for map interpretation, the nodes may describe the various object types in the map while the links represent the spatial relationships between them.

**Frames.** Frames may be regarded as complex semantic nets which typically represent instances of object types. However, frames are more structured than semantic nets as all important information about the object is concentrated in the so-called slots and fillers. Each slot describes an aspect of an object and may be filled by other frames, a default value, or procedural information. In the latter case, a procedure is associated with a specific slot by means of a mechanism called *procedural attachment*. These procedures describe what to do when a particular slot is filled with a value or how a value should be computed for a slot when needed. Thus, a major weakness of semantic nets, i.e. the difficulty to represent procedural knowledge, is at least partly solved by the frame representation. Within the context of map interpretation, a frame may represent an instance of a map object where its slots represent information about its features and spatial relationships with other frames. Procedural attachment may be used to execute an (image processing) procedure if the value of a specific feature is needed, or, to search for related objects if a conceptual object is instantiated.

**Rules.** In general, a rule can be viewed as a conditional statement which is written as

*if <condition> then <action>*

The condition part describes the applicability of the rule whereas the action part describes what to do when the condition is true. This concept is easy to understand and its intuitive use is appealing to many researchers. Moreover, a rule-based system is very appropriate to model a data-driven or bottom-up inference process. If the system is provided with new information, its behavior will change. Although the knowledge base is easy to extend with new rules, it is difficult for the knowledge engineer to anticipate all consequences. For example, a rule which seems appropriate for a specific inference, may also be applicable in other unexpected situations. Therefore, an unstructured rule base may easily become unpredictable, and some kind of ordering is required to guarantee that the interpretation will indeed converge to an acceptable outcome. A frequently encountered solution to this problem is the concept of meta-rules. Meta-rules are a special type of rules and provide extra information about the applicability of other rules, such as which rules are more important in a given situation and should be executed first.

According to Davis and King [1], a rule based approach is most suited for a domain with many independent states, when the control flow consists of a sequence of independent actions and the knowledge can be represented as a collection of independent chunks of knowledge. When regarding map interpretation, there is much a priori knowledge available about the structure and contents of the map. Since the structure of the map is used to guide the interpretation, neither the control flow nor the a priori knowledge is independent. For this reason, a rule based approach may be less appropriate. The nature of the knowledge to be represented, i.e. a network of objects and their spatial relationships, seems to be reflected most adequately by the semantic network and the frame representation. In this thesis, the knowledge representation which is used is a hybrid structure which integrates both aspects of semantic networks and frames, but for simplicity, it will be referred to as the semantic network.

## 1.3 The application

The conversion of the paper utility maps from the PNEM (a Dutch acronym for Provinciale Noordbrabantse Energiemaatschappij) was chosen as the application to test new techniques and methodologies developed within the context of our research. The purpose of this section is to introduce the problem and its current solution and to briefly describe a proposal for an (semi-) automatic solution.

### 1.3.1 Outline of the problem

The PNEM provides most of the province of Noord-Brabant with multiple services such as electricity, water, gas and cable television. For each service, information about position, structure and contents of the network infrastructure is drawn on paper maps. Basically, the maps contain the relative position of the pipelines and conduits with respect to distinctive landmarks such as the corners of houses. Further, the maps contain information about identification, type and covering of the network components. Since the maps also provide information about street names and house numbers, the location and contents of the (usually) underground network can be exactly reconstructed from the maps.

For efficient management of the infrastructure, the information in the maps has to be available in a GIS. Currently, new maps are drawn within the GIS-environment and the information is incorporated directly in the GIS. Despite the digital design of new maps, most existing information is only available on paper. Therefore, there is a great interest in techniques for efficient conversion of the paper maps.

### 1.3.2 Advantages of a digital description

It is desirable to make the information in the maps available in a GIS, even at the expense of millions of guilders in investments in the conversion process. The motivation for this process is threefold. First, there is a need for efficient retrieval of symbolic, numeric and spatial information. Manual retrieval of information from maps is, at the least, time consuming. If a query involves thousands of maps, e.g. *How many kilometers of cable are covered with this type of PVC in district X?*, manual processing of the maps may very well become a hopeless task. From this example, it may be clear that a digital high-level description in a GIS is vital for efficient management and maintenance of the network infrastructure.

A second reason for conversion is the efficient maintenance of the information in the maps. The infrastructure drawn on the maps is not static, and contents of the maps are often subject to change. It is therefore necessary to update the paper maps on a regular basis. Updating a paper map is a tedious and labor intensive process which requires erasing and redrawing parts of the map. However, when the maps are available in a GIS, manipulation of the information with the proper tools is easy and efficient while changes can be incorporated directly into the GIS.

Modeling of the network capacity is the third important advantage of a digital description. For example, if a public utility has to transport a large quantity of water and the optimal network route is required, a digital description of structure, position and capacity of the network is necessary.

### 1.3.3 Advantages of a knowledge-based approach

Earlier in this chapter, we explained the reasons to employ knowledge in the conversion process. The advantages of a knowledge-based approach for this specific application will now be discussed in more detail.

In the Netherlands there are about 40 to 50 public utilities, where each utility usually provides multiple services. Each utility is independent and has its own drawing conventions. Within a utility, different services are drawn on different types of maps using a slightly different symbol set. Even within a single service a further distinction can be made. For example, gas is transported to local distribution centers through a high pressure network, while for distribution to customers, a low pressure network is used. Again, the drawings of both types of network are based on slightly different drawing conventions. Moreover, it is possible that, over time, the drawing conventions have been adjusted to meet new requirements. As a consequence, in the Netherlands, and even for a single public utility, there is an enormous variety in types of utility maps. Because so many applications have to be considered, the flexibility of the interpretation system is most important. A knowledge-based approach may offer the flexibility and facilitate the change to another type of map. Most of the reprogramming of the application may be circumvented because it is sufficient to adjust the knowledge base. A knowledge-based approach may therefore considerably speed up development for the multitude of applications.

Due to years of intensive use in the field, the maps may be wrinkled and stained, while often parts have been erased and redrawn which has degraded the paper material even further. Nevertheless, an automatic interpretation system has to satisfy the high demands made upon its reliability. A system capable of converting the entire map but with an error rate of 10% is not practical, as all results have to be verified and corrected manually. However, a system which correctly converts 80% of the information in the maps leaving the remaining 20% to the operator could achieve a considerable speed-up of the conversion process. In this context, knowledge about the structure of the map and the appearance of objects could guide the automatic interpretation into promising and reliable directions and help to detect and reject inconsistent results. Knowledge about the causes of these inconsistencies and possible solutions could further increase the reliability. Even in the case of the rejected classifications, it could still be possible to generate a list of potential candidates to provide efficient support during manual conversion.

### 1.3.4 Conversion of the maps

In this section, some definitions on positioning are given first, followed by a discussion on existing conversion methods. The section is concluded with a proposal to integrate an automatic conversion method with the current conversion techniques.

#### 1.3.4.1 Definitions

- A position in **world coordinates** of an object describes its position in the real world with respect to a unique landmark. In the Netherlands, a widely used coordinate system is the *Rijksdriehoeksnet*, a triangulation network covering the Netherlands, where each node represents the exact location of an outstanding landmark such as a church tower.

- A position in **map coordinates** denotes the position of a map object with respect to the cross-hairs drawn on the border of the map.

- The **relative position** of a map object describes its position with respect to other map objects. In the

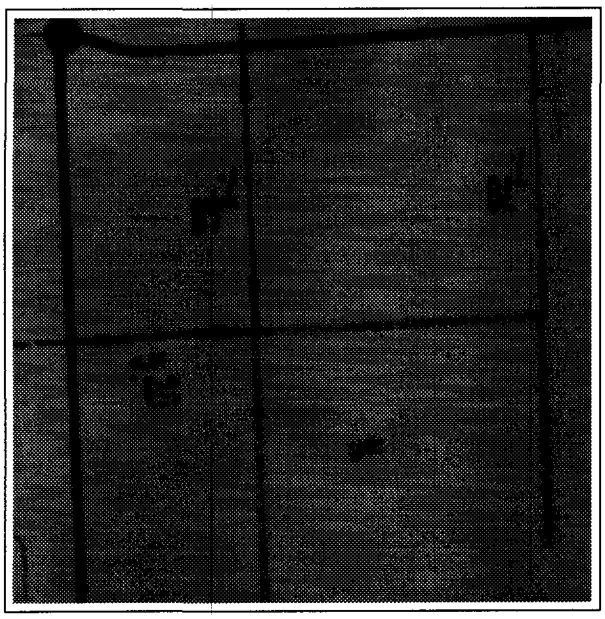


**Figure 1.2**: *This fragment of a PNEM utility map shows the relative positioning of the network with respect to the houses.*

PNEM application, the relative position is denoted with an arrow and a dimension. Fig 1.2 shows an example.

#### 1.3.4.2 Conversion methods

The major aim of the conversion process is to reconstruct the actual position of the network from the maps and to store it in world coordinates in a GIS. Since each map represents a rectangular part of a district, one might expect that a simple linear transformation should be sufficient to transform map coordinates into world coordinates. Unfortunately, the truth is more complicated. For the objects in the maps, a distinction has to be made between the relative position and the position in map coordinates. Although the relative positions in the maps contain few errors, the position in map coordinates may be very unreliable and therefore unsuitable for direct reconstruction. However, the Dutch cadaster is making the position (in world coordinates) of Dutch real estate available in a digital format. This format will be referred to as the digital topography. Thus, when the houses in the maps can be matched with the corresponding houses in the digital topography, it should be possible to reconstruct the precise position of the network from its relative position to the houses and their exact position in the digital topography.

The success of the conversion depends on the match between houses on the maps and the houses in the digital topography. In some cases, when the positions of the objects in map coordinates are inconsistent with the digital topography, this match is not trivial and requires much operator interaction. As a consequence, the quality of the maps, i.e. the consistency with the digital topography, is one of the main factors to determine the conversion speed as well as the conversion method. At the moment, in practice, four manual conversion methods are used.

**Digitization.** In this method the digital cadastral topography is plotted accurately on stretch-free sheets. Using the old maps, which contain the relative position of the network with respect to the houses, a draftsman redraws the accurate position of the network on the sheets. Following this step, the stretch-free sheet is fixed on a digitization table and the position of the network is traced with a digitizer. Since the new map is redrawn exactly to scale, the digitized map coordinates can be transformed directly into world coordinates.

Redrawing the maps on the stretch-free sheets is the bottle-neck of this approach and it makes it the slowest of all available conversion methods. Conversion of an A0 map may take up to several days, and, as a consequence, this method is used only for complicated situations when the maps deviate widely from the digital topography and only an operator can reconstruct the actual situation.

**Reconstruction.** In the reconstruction approach, the digital topography is drawn on a computer display. Using the analog maps, the draftsman now draws the position of the network on the display, in this way combining redrawing and digitization in a single step. This approach is approximately 40% more efficient than digitization. However, a disadvantage of this approach is that the draftsman has to divide his attention among the analog map, the computer display, and an ASCII terminal for the display of alphanumerical information. If the draftsman is confronted with many inconsistencies which require a lot of information from other sources, it is more efficient to use the previous method. Therefore, this approach is best suited if a limited number of analog maps are involved and the information in the maps is more or less consistent with the digital topography.

**Digital conversion.** In the first step of this approach, the draftsman directly traces the position of the network in the original maps with a digitizer. Only the relative position of the network with respect to the houses is expected to be accurate, while the positions of the houses in map coordinates are likely to contain mistakes. In a second step, the system therefore tries to warp the digitized coordinates on the digital topography. This complex and computationally expensive process will only be successful if the maps do not widely deviate from the digital topography, and this method is therefore not appropriate for inconsistent maps. However, if automatic warping is successful, digital conversion is about twice as fast as digitization.

**Improved reconstruction.** The fourth approach is similar to reconstruction. The main difference is that the paper maps are digitally scanned and the map image is plotted together with the digital topography and the alphanumerical information on the display of a workstation. This approach has the advantage that the attention of the operator no longer has to be divided among the maps and the display. The operator only has to concentrate on the workstation and he can work more efficiently and more reliably than in standard reconstruction. Improved reconstruction has a performance similar to digital conversion but its use in practice is limited by the high costs associated with the specialized workstations.

If all paper maps have to be converted manually, the entire conversion project will take several more years. Therefore, there is a need to add an automatic method to the existing conversion techniques. In an automatic conversion method, improved reconstruction has to be combined with digital conversion. In this case, the scanned map is plotted on the display. The operator then selects an area from the map and offers it to the computer system for conversion. The conversion results are shown on the screen and, if accepted by the operator, the map coordinates are warped onto the digital topography. When the system encounters a difficult part, it should reject it and leave it to be classified by the operator. In such a case, the system could still offer support by guiding the operator to rejected parts and suggest potential solutions when available. Similar to reconstruction and digital conversion, this method is not appropriate for very inconsistent maps. To increase the potential use of the automatic conversion, the current warping process has to be improved. This topic is not within the scope of this thesis and will be reported elsewhere. Nevertheless, if despite possible inconsistencies, it is possible to warp the houses in the map on the digital topography, a considerable speed-up can be achieved with a reliable (semi-) automatic conversion method.

## 1.4 A framework for knowledge-based map interpretation

As discussed in the previous sections, a framework for map interpretation should be able to combine both bottom-up and top-down processing. Chapters 2 and 3 of this thesis are dedicated to the bottom-up part of the interpretation process. In Chapter 2, a new low-level representation is proposed to describe a binary image in terms of its graphical primitives while Chapter 3 concentrates on a recognition strategy to classify these primitives. In Chapter 4, a knowledge-based approach to map interpretation is presented. In this approach, a top-down control strategy guides the interpretation process while bottom-up processing is used for object recognition. In Chapter 5, this concept is further refined with a top-down strategy to control the segmentation of the map image. The interpretation system is evaluated on the PNEM application. The applicability of top-down segmentation for aerial image interpretation is then illustrated by a case study in Chapter 6. Chapter 7, finally, discusses the concepts developed during this research and gives directions for future research.

# Bibliography

[1] R. Davis and J. King. *An overview of production systems*, pages 300–332. Machine intelligence 8. Ellis Horwood Limited, Chichester, Sussex, England, 1977. ISBN 85312-058-7.

[2] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Addision Wesley, Reading, Massachusetts, 1992. ISBN 0–201–50803–6.

[3] R.M. Haralick and L.G. Shapiro. *Computer and robot vision*, volume 1. Addison-Wesley, Reading, Massachusetts, 1992. ISBN 0-201-10877-1.

[4] R.M. Haralick and L.G. Shapiro. *Computer and robot vision*, volume 2. Addison-Wesley, Reading, Massachusetts, 1992. ISBN 0-201-56943-4.

[5] E. Rich. *Artificial Intelligence*. McGraw-Hill series in artificial intelligence. McGraw-Hill, 8th printing edition, 1988. ISBN 0–07–052261–8.

[6] M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis and machine vision*. Chapman Hall, Cambridge, Great Britain, 1st edition, 1993. ISBN 0-412-45570-6.

[7] P.H. Winston. *Artificial intelligence*. Addison-Wesley, Reading, Massachusetts, 2nd edition, July 1984. ISBN 0-201-08259-4.

# 2

# An alternative to vectorization: decomposing graphics into primitives

This chapter is based on the following publication:

J.E. den Hartog, T.K. ten Kate, J.J. Gerbrands, and G. van Antwerpen, An alternative to vectorization: decomposition of graphics into primitives. In *Proc. of the 3rd Annual Symposium on Document Analysis and Information Retrieval (Las Vegas)*, pages 263-274, April 1994.

# 2

# An alternative to vectorization: decomposing graphics into primitives


## Abstract

*In this chapter, a new method to describe graphics is proposed as an alternative to the approach of vectorization. Though vectorization has the advantage of reducing memory requirements as well as that of introducing an abstract representation of the image, it has the disadvantage of loss of morphological information and of introducing inaccuracies. In the new approach, graphics are described as a collection of primitives, which are obtained by decomposition of the skeleton of the graphics, followed by a reconstruction step in which each primitive is reconstructed from a skeleton fragment. The proposed method has the advantage of preserving morphological information of the reconstructed primitives thus considerably facilitating their recognition.*


**Keywords:** vectorization, graphics decomposition, graphical primitives, graphics recognition.

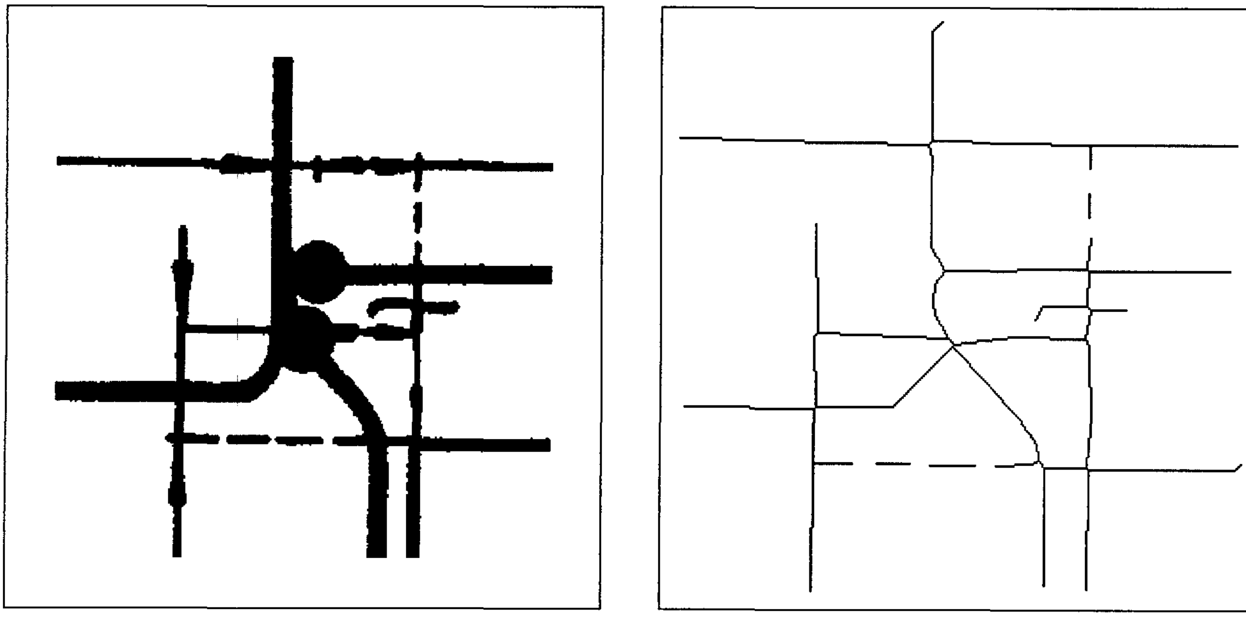
## 2.1 Introduction

Recent years have seen an increasing use of computers for engineering drawing applications. For many public utility organizations, digital information is the primary source for design, planning and maintenance. Despite the ongoing increase in the digital management of information, the major part of communication still takes place on paper. Paper however does not facilitate the management of documents. Electronic handling of these documents is required to allow for easy storage, retrieval, reproduction, exchange and editing. Even though most current work is directly stored in digital form, there are still an enormous number of paper-based drawings which need to be converted. Mere digitization of these drawings into a raster image by using a scanning device does not allow for efficient management by means of a geographic information system (GIS). As a consequence, digitization should be followed by subsequent analysis where structure and objects are recognized in order to arrive at a complete and compact description of the drawing.

Vectorization is an often encountered (first) step prior to the recognition of structure and objects in line drawings, e.g. [1, 2, 5, 8, 11, 12]. The advantages of vectorization are obvious, namely reduction of memory requirements and a more abstract description of the information in the image. Further, vectorization enables efficient spatial reasoning on the objects in the image.

However, in spite of the attractive simplicity, rigorously applying vectorization may introduce unwanted inaccuracies. Almost every technical drawing contains symbols and objects which cannot be described both efficiently and accurately through a vector representation. Vectorization of small or curved lines, for example, may result in a very coarse estimation which may obstruct proper recognition. Consider the drawing fragment in Fig. 2.1a with the corresponding vectorization in Fig. 2.1b. This fragment contains several graphical objects, many of which have a very specific appearance. Reducing these objects to a single line discards important morphological information, which complicates their proper recognition.

Even though vectorization is a useful method in the process of interpreting technical drawings, mere vectorization is not sufficient to describe all pixel information accurately. Therefore, vectorization should be combined with a method which preserves morphological object information. In this chapter, a new method is proposed which describes graphical objects in terms of their constructing primitives thus providing a more abstract description of the image while preserving morphological information.



(a) (b)

**Figure 2.1**: *In (a) a fragment of a utility map is shown together with the corresponding vectorization in (b).*

## 2.2 The algorithms

In this section, some definitions are given first, followed by a discussion of some important aspects of the Euclidean distance skeleton, which is an essential step in the decomposition of a connected component into its primitives. Next, a method is presented to decompose a component through breaking up its skeleton. Finally, the algorithm is explained to reconstruct a graphical primitive starting from a skeleton fragment.

### 2.2.1 Definitions

The formal definitions of some terms and image processing functions are introduced in this section.

- A primitive $P$ is a connected component which is a part of another connected component $C$, such that $P \subseteq C$.
- $disk(c,r)$, with center $c = (c_x, c_y)$, and radius $r$ is a collection of pixels $p = (p_x, p_y)$ with the property that $disk(c,r) = \{p \mid (c_x - p_x)^2 + (c_y - p_y)^2 \leq r^2\}$.
- The function $dist(B_1) = D$ calculates for each object pixel in binary image $B_1$ its pseudo-Euclidean distance to the background [3]. The distance is stored as an integer value in the corresponding pixel in the distance image $D$.
- The function $cdt(B_2, C) = D_c$ calculates for the binary image $B_2$ the constrained distance transform. For each object pixel in image $B_2$ the distance to the background is calculated, taking in account the preset values in the constraint image $C$. The constrained distance transform is introduced in Section 2.2.4.1.
- The function $skeleton(B_3) = S$ calculates the pseudo-Euclidean skeleton of the binary image $B_3$ and stores the result in binary image $S$ [3].
- The function $threshold(G,t) = B_4$ converts the grey value image $G$ into the binary image $B_4$, such that

$$\forall i,j \;\; B_4(i,j) = \begin{cases} 0 & \text{if } G(i,j) < t \\ 1 & \text{otherwise} \end{cases}$$

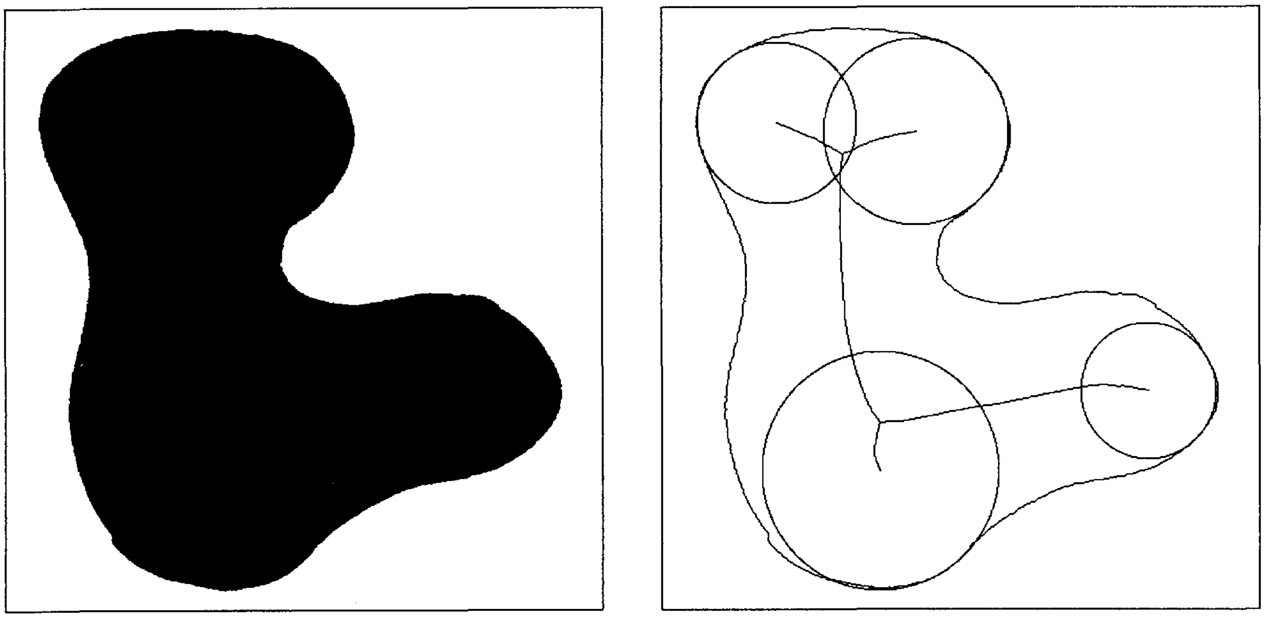
### 2.2.2 The pseudo-Euclidean distance skeleton

The algorithm to decompose a component into its primitives is based on the following perception of the Euclidean skeleton :

Given the function $dist(p)$ stating the distance from a pixel $p$ to the background, and $C$, a connected component. The Euclidean skeleton $ES$ of component $C$ can be regarded as the smallest set of connected pixels with the property that

$$\bigcup_{p \in ES} disk(p, dist(p)) = C \tag{2.1}$$

Thus, a skeleton of a component comprises a minimal set of connected pixels in such a way that the union of all pixels within the set of disks centered on the skeleton pixels, with radii equal to the distance from each skeleton pixel to the background, reconstructs the component [14]. The pixels within such a disk are said to be associated with the skeleton pixel at the center. The relationship between a component and its Euclidean distance skeleton is also depicted in Fig. 2.2.



(a) (b)

**Figure 2.2**: *The relationship is shown between the arbitrary component in (a) and its Euclidean distance skeleton in (b) together with four disks centered at the end points.*

As a consequence, each component can be reconstructed from its skeleton pixels, and the corresponding distances to the background. However, reconstructing only a fragment of the skeleton will result in a partial reconstruction of the component. Thus, decomposing the skeleton into fragments and propagating these fragments into partial reconstructions will effectively decompose the component.

### 2.2.3 Skeleton decomposition

The complexity of a component is reflected in its topology, and, as a consequence, also in its skeleton. Therefore, a decomposition of a component can be obtained through a topological decomposition of its skeleton. Consider the connected component in Fig. 2.3a. The skeleton of this component in Fig. 2.3b is decomposed into fragments, using its branch points as intersections. To prevent fragmentation of the component, small skeleton segments with a branch point at one end and an end point at the other end are eliminated.

Next, for each remaining skeleton fragment, all pixels associated with the fragment are reconstructed resulting in the basic primitives shown in Fig. 2.3c. To be able to show each reconstructed primitive, the primitives in Figs. 2.3c and 2.3d are displayed in exploded view.

The little blobs at the end of the primitives in Fig. 2.3c are artifacts originating from the reconstruction of the neighborhood of the branch point which lies inside the component. The correct decomposition is obtained in the following way. First, the distance to the background in the branch point is determined, resulting in a value $n$. Next, the first $n$ pixels of each skeleton fragment starting in the branch point are removed, resulting in the decomposition shown in Fig. 2.3d.

### 2.2.4 The reconstruction algorithm

In Section 2.2.2, it was argued that (part of) an object can be reconstructed from its skeleton by reconstructing all disks with the center at a skeleton point, and a radius equal to the distance from the center to the background. Reconstructing these disks one by one for each skeleton pixel would be computationally inefficient. An efficient algorithm to reconstruct an object from its skeleton using the constrained distance transform (CDT) was proposed in [4]. This algorithm is based on the opposite approach where, instead of generating the disks, for each pixel position in the image it is decided whether this pixel is part of the object or not. The outcome depends on its distance from the skeleton pixels *and* the radii of the disks centered at these skeleton pixels. The CDT provides an efficient method to determine for each pixel in the image its distance to the skeleton in proportion to the radii of these disks. Thresholding the constrained-distance transformed image will render the original object.

#### 2.2.4.1 The constrained distance transform

The CDT is adopted from the distance transformation. Conditions are added to the transformation which are specified through a grey value "constraint" image. Non-zero points within this image are interpreted as preset values for the distance transform values of the corresponding pixels, regardless of the result of the distance transform itself. Distances between two points are locally increased by any

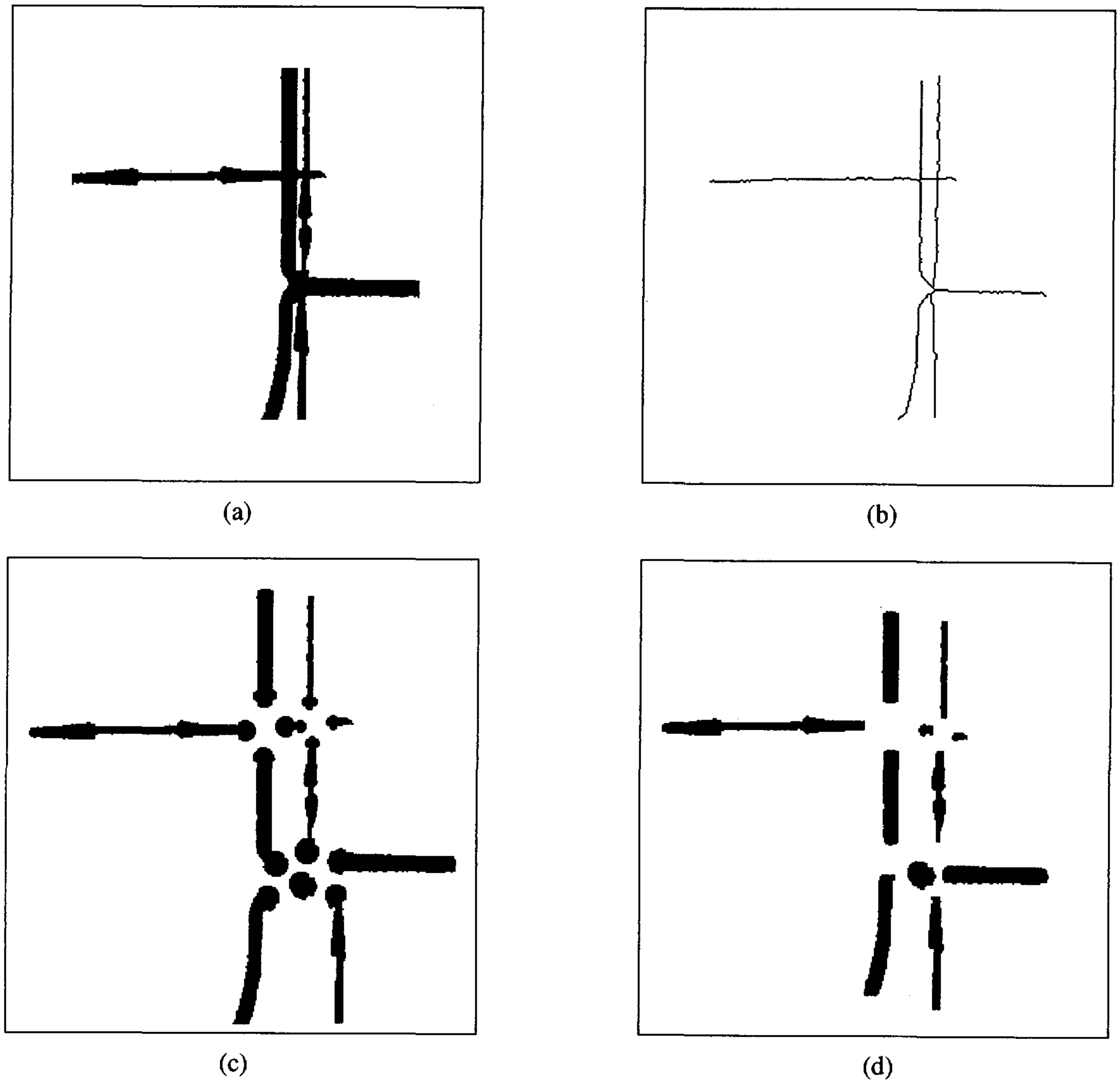



**Figure 2.3**: *The connected component in (a) is decomposed by propagating the fragments of its skeleton in (b) into the primitives in (c). The blobs at the end are removed, resulting in the enhanced primitives in (d).*

preset pixels on the shortest path between the two points.

Consider the example in Fig. 2.4a, in which the dark pixels denote object pixels while the two white pixels form the background. Figure 2.4b shows the distances of the object pixels to the two background pixels according to the standard chamfer 5x5 (5,7,11) distance transform [3] with integer values $5(=5\sqrt{1})$, $7(\approx 5\sqrt{2})$, and $11(\approx 5\sqrt{5})$ for horizontally and vertically connected neighbors, diagonally connected neighbors, and neighbors connected by the knights-move from chess. Next, conditions are added to this transform by means of the preset values in Fig. 2.4c. Note that value $-7$ is set to a background pixel, while value $-2$ is the preset value for one of the object pixels. Figure 2.4d shows the effects of these two preset pixels on the distance values.

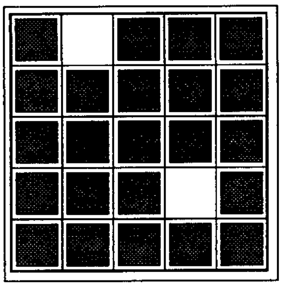
#### 2.2.4.2 An example reconstruction

Next, the reconstruction of an example component using the CDT is shown. For reasons of clarity, only the reconstruction of a very small and simple component is shown. Consider the component in Fig. 2.5a with its 5x5 (5,7,11) chamfer distance values in Fig. 2.5b. Note that the distance values of the skeleton pixels are printed in bold face.

The skeleton pixels and the corresponding distance values are used to calculate the input images for the CDT. The input comprises a binary image $B$ and a grey value constraint image $C$. Image $B$ and $C$ are shown in Figs. 2.5c and 2.5d respectively. The algorithm to calculate $B$ and $C$ is shown in pseudo-code:

```
D = distance(O);      // Fig. 2.5b
S = skeleton(O);      // Fig. 2.5b
B = ¬S;               // Fig. 2.5c
D = - D;
C = D ∧ S;            // Fig. 2.5d
```

Next, the original input image is reconstructed using the CDT:



(a)

| | | | | |
|---|---|---|---|---|
| 5 | | 5 | 10 | 15 |
| 7 | 5 | 7 | 10 | 11 |
| 11 | 10 | 7 | 5 | 7 |
| 15 | 10 | 5 | | 5 |
| 16 | 11 | 7 | 5 | 7 |

(b)

| | | | | |
|---|---|---|---|---|
| | | | | |
| | **-2** | | | |
| | | | | |
| | | | **-7** | |
| | | | | |

(c)

| | | | | |
|---|---|---|---|---|
| 5 | | 5 | 8 | 9 |
| 3 | **-2** | 3 | 3 | 4 |
| 5 | 3 | 0 | -2 | 0 |
| 8 | 3 | -2 | **-7** | -2 |
| 9 | 4 | 0 | -2 | 0 |

(d)

**Figure 2.4**: *The constrained distance transform is illustrated by the example object in (a) and the constraints in (c). The standard chamfer 5x5 (5,7,11) distance values are given in (b) while the constraint distances are presented in (d).*

```
Dc = cdt(B,C);              // Fig. 2.5e
R  = threshold(Dc , 0);
R  = ¬R;                    // Fig. 2.5f
```

The constrained distance transform applied to the input images $B$ and $C$ results in the output image $D_c$ depicted in Fig. 2.5e. All pixels in $D_c$ with value less than zero are object pixels. Thus, thresholding grey value image $D_c$ with threshold 0 into binary image $R$ will reconstruct the original object. Inversion of image $R$ will result in the reconstruction shown in Fig. 2.5f. The light grey pixels in Fig. 2.5f denote the location where the reconstruction failed a pixel, which is caused by small inaccuracies in the skeleton.

## 2.3 Discussion and results

### 2.3.1 Advantages

In this chapter, a new approach to describe graphical components with application to the automatic interpretation of technical drawings has been presented. Combined with the conventional vectorization approach, a more complete description is obtained which preserves morphological object information and allows for easy geometric reasoning as well.

Decomposition of graphics into the constructing primitives facilitates the automated interpretation of technical drawings. The attractiveness of this approach is that all primitives can be stored in separate bitmaps, which enables the application of image processing algorithms on individual primitives.

The ability to apply image processing to a single primitive opens the way to calculate attributes for individual primitives. A vector-based object description does not allow for the calculation of many useful attributes, e.g. area, perimeter, attributes based on the minimal enclosing rectangle [6, 9], and moment-based features [7, 10, 13]. Further, individual representation of primitives allows for their recognition by using template matching. Template matching in the original binary image will often yield erroneous results due to interference from neighboring primitives, while representation of each primitive in a separate bitmap does not have this disadvantage.

Another important advantage of the presented algorithm is that it is generically applicable to all binary drawing images. Though in theory no user-defined parameters are necessary to decompose the components, the practical implementation requires two parameters to prevent unnecessary decomposition and the fragmentation of connected components.

The first parameter determines whether a connected component in the image will be offered for decomposition. Only the graphical components which are constructed of several primitives, usually the large components, have to be decomposed. For the recognition of small objects, such as characters, decomposition is not desired. Therefore, an area threshold is applied to all components in the image, and only those components with a pixel area above this threshold will be offered for decomposition.

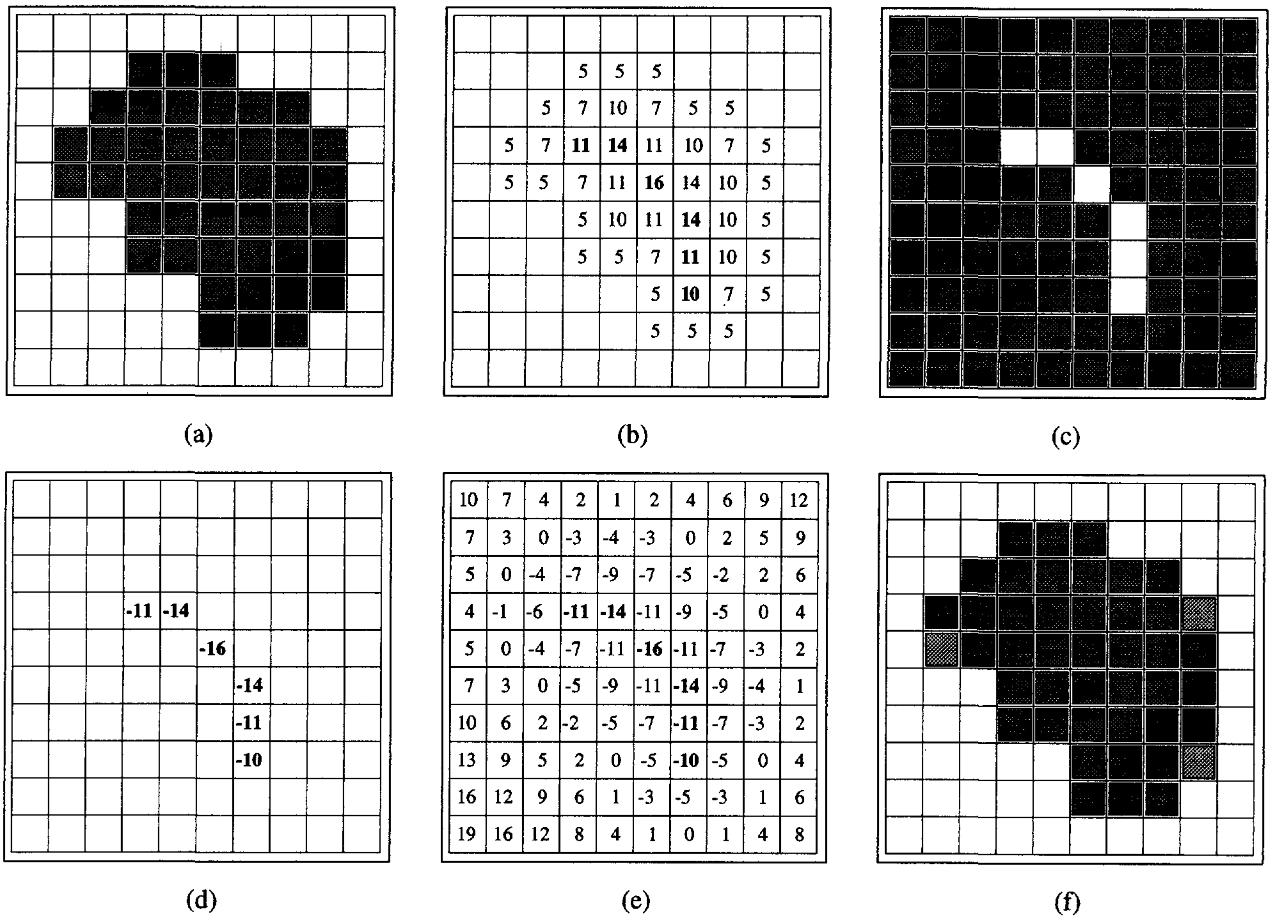
The second parameter determines which side branches of the skeleton will be removed. If no branches are removed, the algorithm will yield a very fragmented decomposition. If, on the other hand, long branches are removed, pixel information will be lost. Depending on the type of drawing and the scanning density, a threshold for the minimum length of side branches has to be determined.

### 2.3.2 Drawbacks

The main practical drawback of component decomposition is that the algorithm is based on two distance transformations. As a consequence, the algorithm requires more memory and more computations than a conventional vectorization algorithm.

To be able to make a comparison between vectorization and component decomposition, concerning computation time and memory requirements, three experiments were carried out on a dec5000/240 with 64 megabytes of internal memory. In the experiments, two vectorization algorithms and the component decomposition were tested on a CAD drawing with a size of 2900×2400 pixels. The example image given in Fig. 2.8a is a fragment of this drawing.

In the first experiment, the standard pseudo-Euclidean




(b)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | |
| | | | 5 | 5 | 5 | | | | |
| | | 5 | 7 | 10 | 7 | 5 | 5 | | |
| | 5 | 7 | **11** | **14** | 11 | 10 | 7 | 5 | |
| | 5 | 5 | 7 | 11 | **16** | 14 | 10 | 5 | |
| | | | 5 | 10 | 11 | **14** | 10 | 5 | |
| | | | 5 | 5 | 7 | **11** | 10 | 5 | |
| | | | | | 5 | **10** | 7 | 5 | |
| | | | | | 5 | 5 | 5 | | |
| | | | | | | | | | |

(d)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | **-11** | **-14** | | | | | |
| | | | | | **-16** | | | | |
| | | | | | | **-14** | | | |
| | | | | | | **-11** | | | |
| | | | | | | **-10** | | | |
| | | | | | | | | | |
| | | | | | | | | | |

(e)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 10 | 7 | 4 | 2 | 1 | 2 | 4 | 6 | 9 | 12 |
| 7 | 3 | 0 | -3 | -4 | -3 | 0 | 2 | 5 | 9 |
| 5 | 0 | -4 | -7 | -9 | -7 | -5 | -2 | 2 | 6 |
| 4 | -1 | -6 | **-11** | **-14** | -11 | -9 | -5 | 0 | 4 |
| 5 | 0 | -4 | -7 | -11 | **-16** | -11 | -7 | -3 | 2 |
| 7 | 3 | 0 | -5 | -9 | -11 | **-14** | -9 | -4 | 1 |
| 10 | 6 | 2 | -2 | -5 | -7 | **-11** | -7 | -3 | 2 |
| 13 | 9 | 5 | 2 | 0 | -5 | **-10** | -5 | 0 | 4 |
| 16 | 12 | 9 | 6 | 1 | -3 | -5 | -3 | 1 | 6 |
| 19 | 16 | 12 | 8 | 4 | 1 | 0 | 1 | 4 | 8 |

**Figure 2.5**: *The reconstruction algorithm is explained by the example object in (a). The distance values of the object pixels are given in (b) with the skeleton pixels printed in boldface. The images in (c) and (d) are input to the constrained distance transform resulting in the output image (e). Thresholding (e) will yield the reconstructed object in (f).*

distance skeleton was vectorized [3]. Because the pseudo-Euclidean distance image is required for the calculation of the skeleton, the vectorization process is computationally expensive. Calculation of the distance image, the skeleton, and the vectorization required 10 mega bytes of memory and 72.6 seconds of computer time, resulting in 2110 line segments.

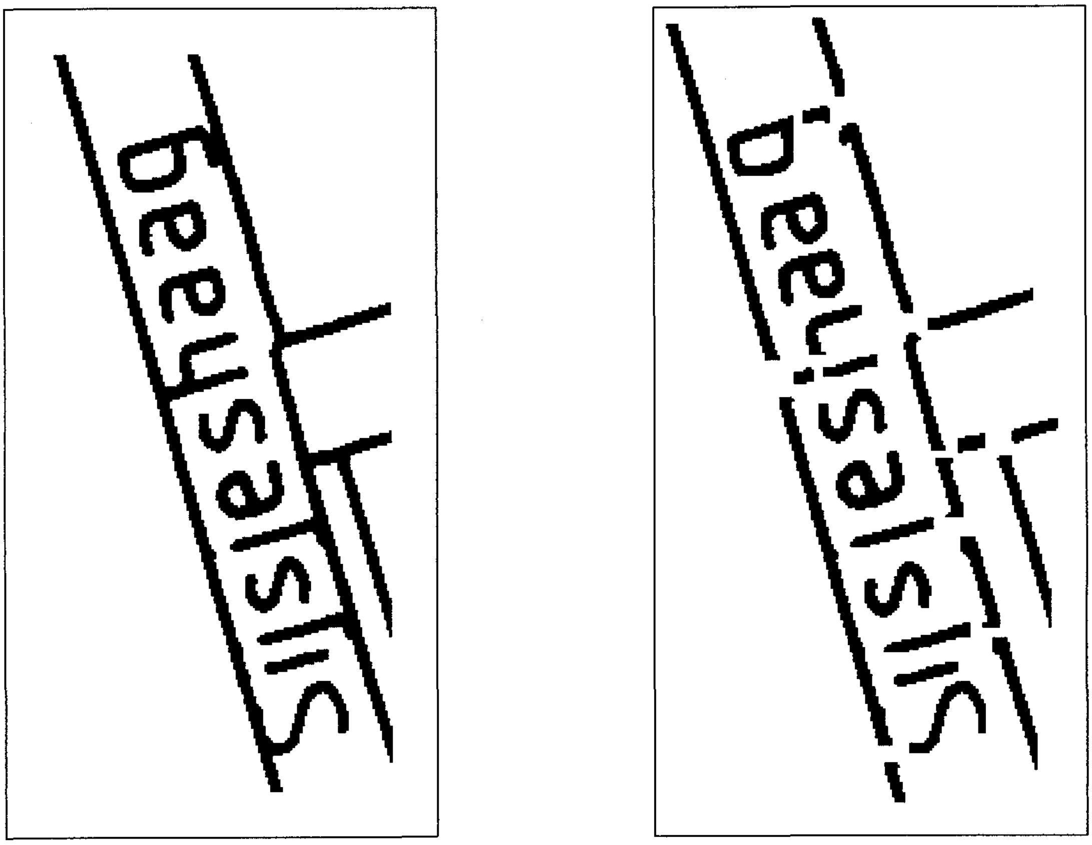
In the second experiment, the drawing was first skeletonized using an algorithm based on the repeated thinning of the components while preserving the connectivity of the remaining component pixels [7]. This type of skeletonization has the advantage of being computationally less expensive than calculation of the skeleton from the distance image, however, in some situations, the skeleton will be less accurate than the pseudo-Euclidean skeleton. Skeletonization and vectorization required maximally 2.5 mega bytes of memory and 48.2 seconds of computer time, while, due to a larger number of branch points in the skeleton, 2213 lines were extracted.

The component decomposition algorithm was tested in the third experiment. The entire decomposition costs 91.0 seconds and required a maximum of 17 mega bytes. After decomposition, 745 primitives were reconstructed.

From the experiments, it can be concluded that component decomposition will cost about twice the computation time needed for straightforward vectorization, while memory costs are considerably larger. If a more accurate vectorization, based on the Euclidean skeleton, is demanded, both time and memory requirements will increase drastically. In this case, the advantage of vectorization to component decomposition concerning computational costs and memory requirements will be considerably smaller.

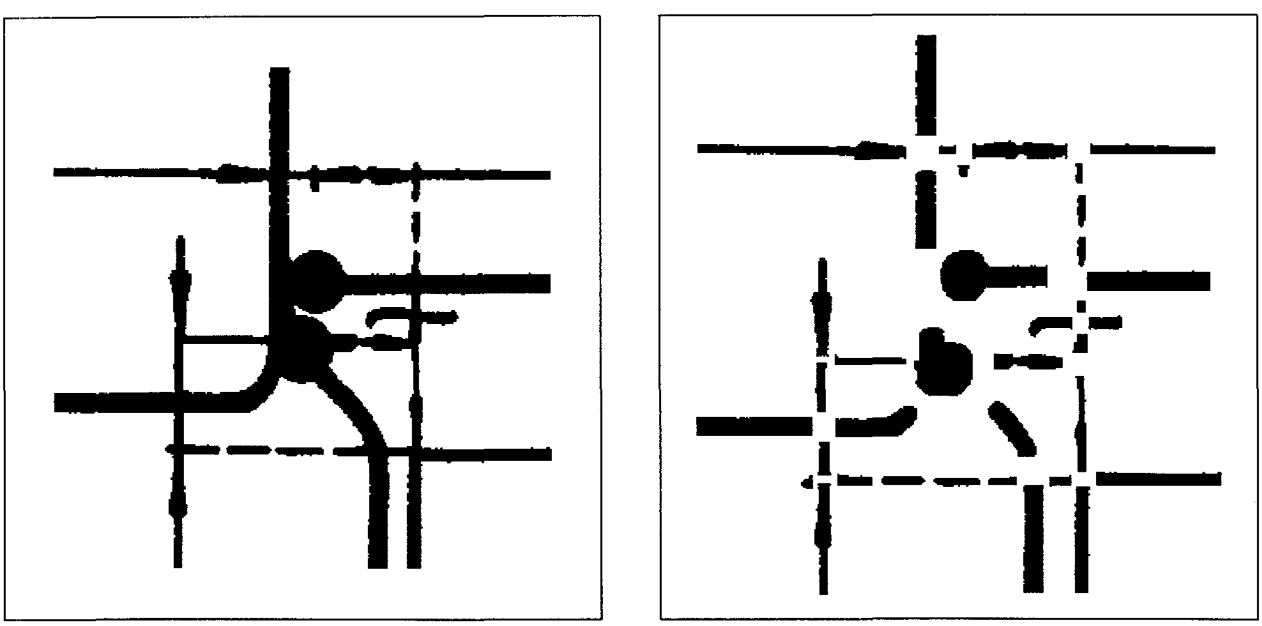
The second limitation of the decomposition algorithm is a consequence of its dependency of the skeleton on decomposition and reconstruction. For example, the occurrence of small holes in a binary object will have dramatic consequences for the skeleton, and therefore for the decomposition into primitives too. Furthermore, the extraction of meaningful primitives from an object as shown in Fig. 2.2a will also be difficult. However, these limitations are also encountered when vectorizing the skeleton.

Therefore, to compare component decomposition with vectorization is to weigh the extra computational costs and memory requirements against the various advantages described in Section 2.3.1.



(a) (b)

**Figure 2.6**: *In (a) several of the characters are touching graphics. The decomposition is shown in (b).*



(a) (b)

**Figure 2.7**: *The utility map in (a) containing several complex symbols is decomposed into the primitives in (b).*

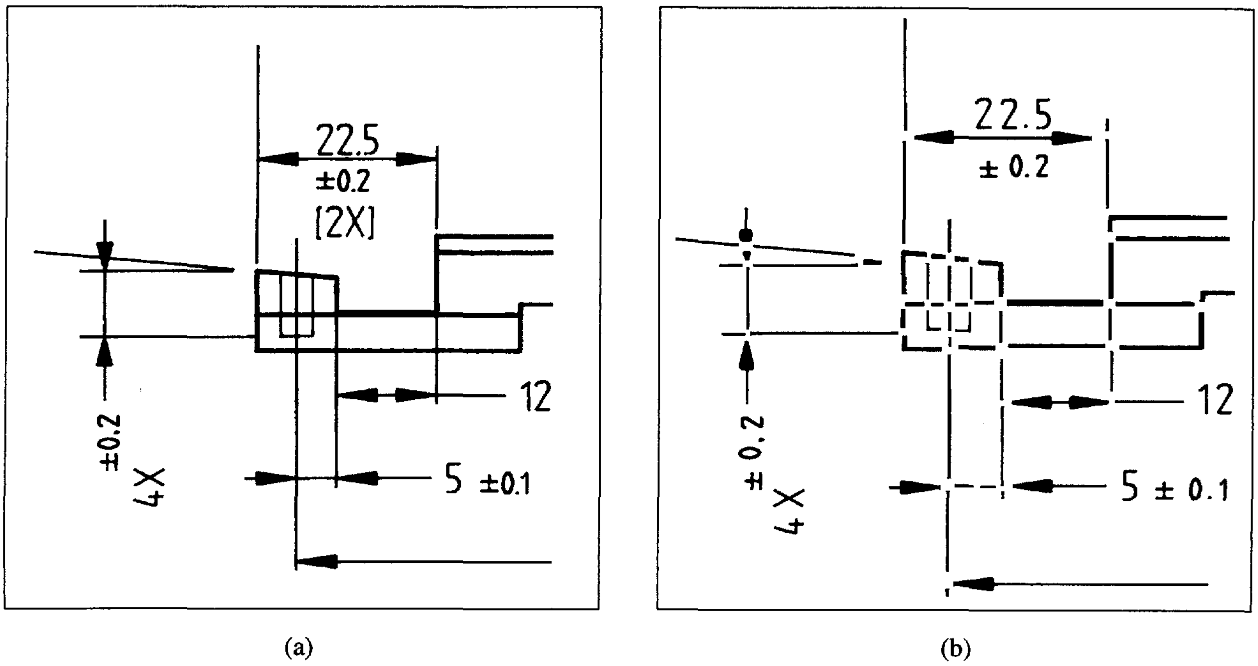



**Figure 2.8**: *The CAD drawing in (a) is decomposed into the primitives in (b).*

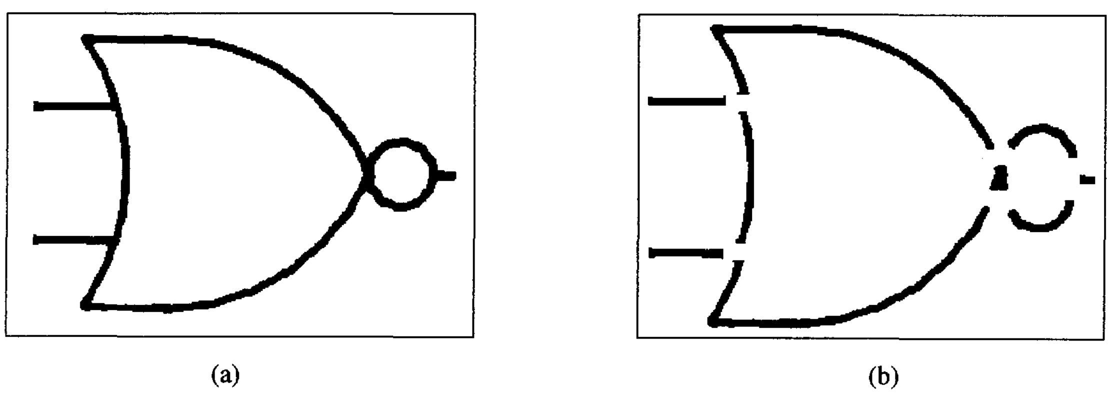



**Figure 2.9**: *The part of the electric diagram in (a) is decomposed into the primitives in (b).*

### 2.3.3 Application examples

The algorithm has been tested on several applications. From three applications, the resulting decomposition of a complex situation is shown. The fourth example shows the decomposition of a complex curved symbol.

The first example to be considered is shown in Fig. 2.6a where five characters of a street name are connected to the graphics. Figure 2.6b shows the resulting decomposition in exploded view. Decomposition of the unconnected characters is prevented by application of the area threshold mentioned in Section 2.3.1. After decomposition of the connected characters, it is in principle possible to classify the primitives either as part of the graphics or as part of a character. However, the problem of recognition of the characters reconstructed from these primitives still remains.

The example in Fig. 2.7a is identical to the fragment in Fig. 2.1. The size of the fragment in pixels is $300^2$ and the fragment is taken from a public utility map scanned at a density of 300 dpi. It contains several very irregular and interconnected symbols. As can be seen in Fig. 2.7b, the reconstruction of the thick circular primitives is not perfect. Even so, successful recognition of all individual primitives is, in principle possible, when based on simple attributes such as thickness, pixel area, and attributes calculated from the minimum enclosing rectangle.

Fig. 2.8a shows a part of a CAD drawing ($1024^2$, 400 dpi). As can be seen in Fig. 2.8b, a nearly perfect reconstruction of the arrows is possible, which will facilitate their recognition considerably. Even the arrow intersected by the line at the left of the image may be recognized from its two reconstructed primitives.

Figure 2.9, finally, shows the decomposition of a single complex curved symbol taken from an electric diagram (300×200, 400 dpi).

## 2.4 Conclusions

The proposed algorithm decomposes a graphical component in terms of its constructing primitives through decomposition of the skeleton, followed by a reconstruction step that is based on the constrained distance transform. Describing a component in terms of its primitives has the advantage that, at the expense of extra computation time and memory requirements, morphological information is retained, as opposed to a vector description. The main advantage of the described algorithm is that the recognition of graphical primitives is facilitated considerably as it offers the possibility to apply image processing techniques, e.g. template matching and the calculation of attributes, to individual primitives.

For this reason, we consider the decomposition of graphics combined with a vector description a very useful low-level method, which can be the basis of a system to interpret complex technical drawings with irregular symbols. Current research is concentrating on the further development of such a system.

### Acknowledgements

This research was partly funded by Esprit (ROCKI, EP5376). We also would like to thank the Provinciale Noordbrabantse Energiemaatschappij and the Dutch cadaster in the Netherlands for putting some sample drawings at our disposal.

# Bibliography

[1] D. Antoine. CIPLAN, a model-based system with original features for understanding French plats. In *First Int. Conf. on Document Analysis and Recognition (Saint Malo)*, volume 2, pages 647–655, October 1991.

[2] D. Benjamin, P. Forgues, E. Gulko, J.B. Massicotte, and C. Meubus. The use of high-level knowledge for enhanced entry of engineering drawings. In *9th IAPR Int. Conf. on Pattern Recognition (Rome)*, volume 1, pages 119–124, November 1988.

[3] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics and Image Processing*, 27:321–345, 1984.

[4] L. Dorst and P. Verbeek. The constrained distance transform: a pseudo-Euclidean recursive implementation of the Lee algorithm. In *Signal processing III: theories and applications*, pages 917–920, Amsterdam, 1986. North-Holland.

[5] C.S. Fahn, J-F. Wang, and J-Y. Lee. A topology-based component extractor for understanding electronic circuit diagrams. *Computer Vision, Graphics and Image Processing*, 44:119–138, 1988.

[6] H. Freeman and R. Shapira. Determining the minimum-area encasing rectangle for an arbitrary closed curve. *Communications of the ACM*, 18(7):409–413, 1975.

[7] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Addision Wesley, Reading, Massachusetts, 1992. ISBN 0–201–50803–6.

[8] K.J. Goodson and P.H. Lewis. A knowledge-based line recognition system. *Pattern Recognition Letters*, 11(4):295–304, 1990.

[9] F. Groen, P. Verbeek, N. de Jong, and J. Klumper. The smallest box around a package. *Pattern Recognition*, 14:173–178, 1981.

[10] M.K. Hu. Visual pattern recognition by moment invariants. *IRE Trans. Info. Theory*, IT-8:179–187, 1962.

[11] M. Ilg. Knowledge-based understanding of road maps and other line images. In *10th IAPR Int. Conf. on Pattern Recognition (Atlantic City)*, volume 1, June 1990.

[12] R. Kasturi, S.T. Bow, W. El-Masri, J.R. Gattiker, and U.B. Mokate. A system for interpretation of line drawings. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(10):978–992, 1990.

[13] A. Khotanzad and Y.H. Hong. Rotation invariant pattern recognition using Zernike moments. In *9th IAPR Int. Conf. on Pattern Recognition (Rome)*, pages 326–328, November 1988.

[14] U. Montanari. A method for obtaining skeletons using a quasi-Euclidean distance. *Journal of the ACM*, 15:600–624, 1968.

1234567

# Finding arrows in utility maps using a neural network

This chapter is based on the following publication:

J.E. den Hartog and T.K. ten Kate, Finding arrows in utility maps using a neural network. In *Proc. of the 12th IAPR Int. Conf. on Pattern Recognition (Jerusalem)*, pages 190-194, October 1994.

# 3

# Finding arrows in utility maps using a neural network


## Abstract

*In this chapter, a new technique is proposed for the reliable classification of poor quality arrows in hand-drawn utility maps. The classification uses a neural network which is trained to distinguish arrows from other line symbols. A line symbol is represented by a feature vector based on the pseudo-Euclidean distances along the skeleton. The classification is evaluated with an independent test set.*


**Keywords:** map interpretation, graphics recognition, symbol recognition, neural network.

## 3.1 Introduction

One of the most frequently encountered object types in technical drawings is the arrow. The reliable classification of arrows is very important to the entire interpretation process, because, in general, an arrow denotes a spatial relationship between two objects. Therefore, their classification allows for the identification of other objects and their spatial relationships.

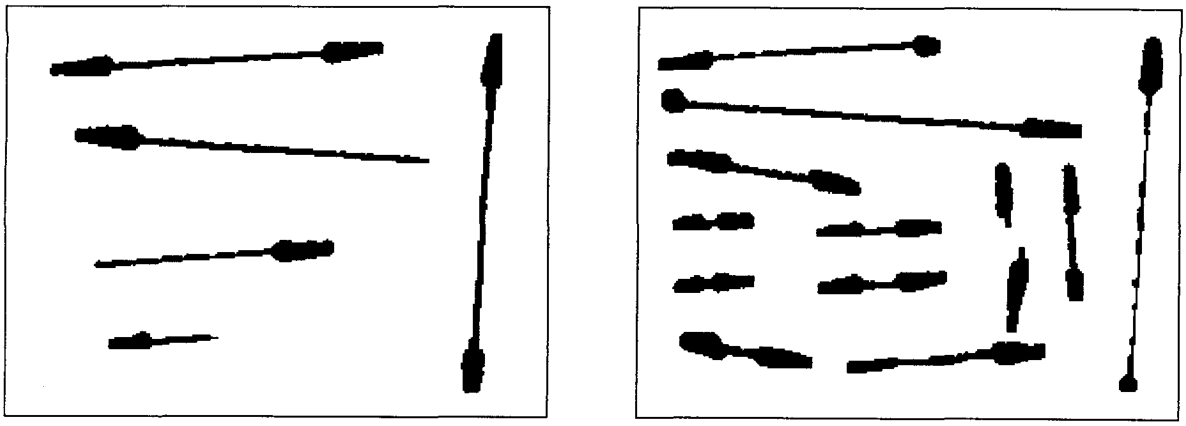
Our research concentrates on the automatic conversion of Dutch utility maps. In this chapter, we consider the problem of distinguishing arrows in these maps from other line symbols. Because all maps are drawn by hand, and many maps are in poor condition, the arrows do not have a regular shape. Ideally, the arrows should not be distorted and the arrowheads would be solid triangles. A few examples of these are shown in Fig. 3.1a. However, most arrows are irregular or distorted, and some examples are depicted in Fig. 3.1b. A consequence of the irregular shape of most arrows is the obstruction of their reliable recognition when using standard techniques such as template matching. Therefore, a classification method capable of handling irregularly shaped objects is required. In this chapter, we present a new technique which can be employed to recognize arrows, which is based on the pseudo-Euclidean distances along the skeleton used as a feature set for a neural network.

## 3.2 Properties of the Euclidean distance skeleton

### 3.2.1 Definitions

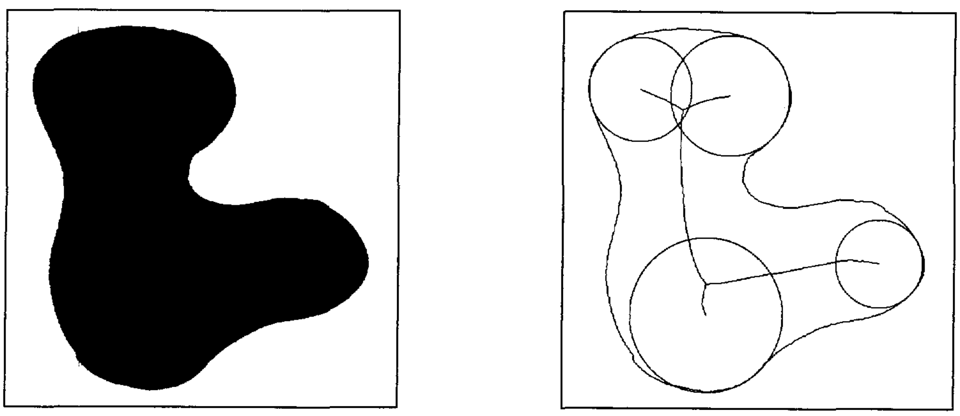
In this section, we introduce five formal definitions which are essential to describe the properties of the Euclidean distance skeleton.

- A component is a collection of topologically connected pixels.
- A skeleton segment $S'$ is a subset of a skeleton $S$, such that $S'$ is topologically connected and contains no branch points.
- A primitive is a component reconstructed from a skeleton segment.
- $disk(c, r)$, with center $c = (c_x, c_y)$, and radius $r$ is a collection of pixels $p = (p_x, p_y)$ with the property that $disk(c, r) = \{p \mid (c_x - p_x)^2 + (c_y - p_y)^2 \leq r^2\}$.
- The function $dist(B_1) = D$ calculates for each object pixel in binary image $B_1$ its pseudo-Euclidean distance to the nearest background pixel [2]. The distance is stored as an integer value in the corresponding pixel in the grey value distance image $D$.



(a) (b)

**Figure 3.1**: *Some good quality arrows are shown in (a), while (b) shows some irregular arrows.*



(a) (b)

**Figure 3.2**: *The relationship is shown between the arbitrary component in (a) and its Euclidean distance skeleton in (b) together with four disks centered at the end points.*

### 3.2.2 The pseudo-Euclidean distance skeleton

The recognition of the arrows is based on the following perception of the Euclidean skeleton:

Given the function *dist(p)* which states the distance from a pixel $p$ to the background, and $C$, a component. The Euclidean skeleton $ES$ of component $C$ can be regarded as the smallest set of connected pixels with the property that

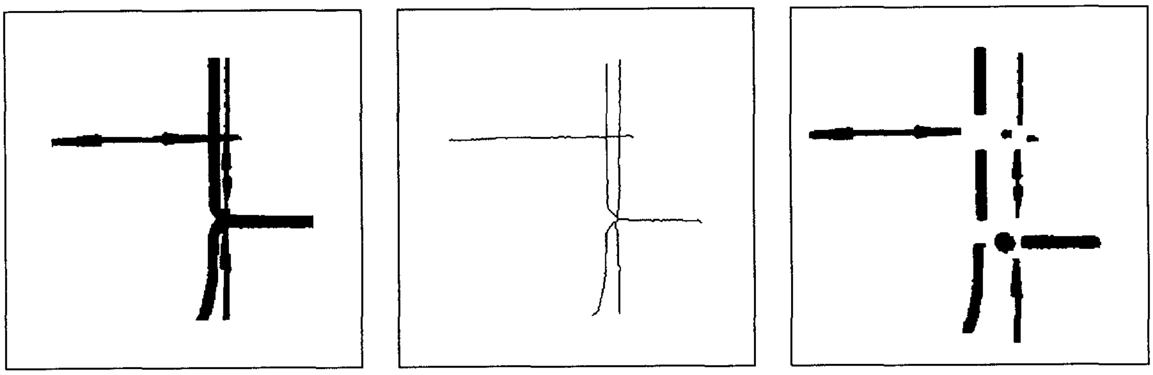
$$\bigcup_{p \in ES} disk(p, dist(p)) = C \tag{3.1}$$

Thus, a skeleton of a component comprises a minimal set of connected pixels such that the union of all pixels within the set of disks centered on the skeleton pixels, with radii equal to the distance from each skeleton pixel to the background, reconstructs the component [6]. The relationship between a component and its Euclidean distance skeleton is also depicted in Fig. 3.2.

### 3.2.3 Decomposition and reconstruction

Each component can be reconstructed from its skeleton pixels and the corresponding distances to the background. However, reconstruction of only a fragment of the skeleton will result in a partial reconstruction of the component. Thus, decomposition of the skeleton into fragments and the propagation of these fragments into partial reconstructions will effectively decompose the component. In [4] we described an algorithm to decompose graphical objects into their constructing primitives while preserving morphological information. In this section, we briefly outline this algorithm.

The algorithm decomposes the binary input image into primitives in two steps. In the first step, a distinction is made between large and small components in the image, based on the area of the components. After this separation step, each small component is stored as an individual image, while the large components, which usually consist of connected graphical symbols, are broken down into basic primitives in the second step. In this step, each large



(a) (b) (c)

**Figure 3.3**: *The connected component in (a) is decomposed by propagating the fragments of its skeleton in (b) into the primitives in (c) which are shown in* ***exploded view.***

object is skeletonized using the pseudo-Euclidean distance skeleton [2]. The skeleton is then searched for short segments which are removed to prevent fragmentation of the components in a later phase. Next, the skeleton is decomposed into its segments using the branch points as breaking points. The pixels in the original binary image which belong to a specific skeleton segment are reconstructed using the constrained distance transform [5]. Each skeleton segment is reconstructed in this way and the result is stored in a separate bitmap. An example of such a reconstruction is shown in Fig. 3.3.

## 3.3 Recognition strategy

If a primitive, such as a line symbol, can be reconstructed from its skeleton pixels and the corresponding distance values, then the skeleton and the distance values provide a compact but complete description of the line symbols which can be used for their classification. In general, the shape of an arrow is determined by a line which is connected to one or two triangularly shaped arrowheads. After skeletonization each arrow will be reduced to a single skeleton segment. Thus, each arrow can be described by **one** skeleton segment and the corresponding set of distance values. As the set of distance values, or distance profile, provides an accurate description of the object shape along the skeleton, it is feasible to train a neural network to distinguish arrows from other line symbols using the distance profile. An example arrow, its skeleton and some of its distance values are depicted in Fig. 3.4. The most prominent feature of an arrow is its (more or less) triangularly shaped head which is located at one or at both ends. To be able to discriminate between arrows and other line symbols, a subset of the distance values, i.e. the distance values taken from both ends of the skeleton fragment that represent the arrowheads, should be sufficient.

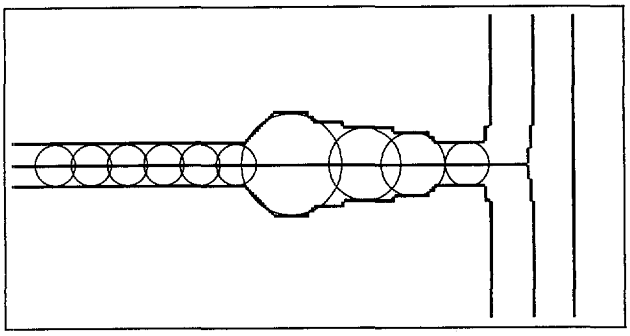


**Figure 3.4**: *An arrow is shown together with its skeleton and its distance profile.*

## 3.4 Experiments and results

### 3.4.1 Data acquisition

Three original utility maps were used for our experiments. From each map, two US-Legal (8.5"×14") sized parts were scanned in grey value at a density of 400 dpi. Next, each grey value image was automatically thresholded [7] to obtain a binary image. From the first two maps, line symbols were selected and randomly distributed over a training set and a test set. From the third map, which was drawn by an independent team of draftsmen, a second test set was selected. The utility maps comprised five different types of line symbols; pipeline segments, house segments, road segments, single-headed arrows, and double-headed arrows. Earlier experiments proved that with regular attributes such as length and average thickness, only the house segments and the road segments were misclassified as arrows and vice versa [3]. Therefore, the two arrow types, the house segments, and the road segments were manually selected to compose one training set and two independent test sets. For each set, the samples from the four classes were then distributed over an arrow class and a non-arrow class. The composition of the sets is given in Table 3.1.

As discussed in the previous section, the distance values taken from both ends, which represent the profile of the arrowheads, were used as a feature set for the neural net-

| | total | total arrows | total non-arrows | single arrows | double arrows | house segments | road segments |
|---|---|---|---|---|---|---|---|
| Training set | 235 | 119 | 116 | 64 | 55 | 79 | 37 |
| Test set #1 | 235 | 118 | 117 | 64 | 54 | 79 | 38 |
| Test set #2 | 254 | 138 | 116 | 98 | 40 | 66 | 50 |

**Table 3.1**: *The composition of the training and test sets.*

work. The number of distance values required to be able to distinguish between arrows and non-arrows depends on the maximum length of the arrow heads and the scanning density. A scanning density of 400 dpi and the maximum length of the arrowheads requires approximately 40 distance values to describe an arrowhead. From the skeleton, the first 40 and last 40 distance values were used. If a skeleton fragment contained less than 40 pixels, the remaining distance values were set to 0. The length of the skeleton fragment and the maximum distance value were added as additional features thus bringing the entire feature set to a total of 82 features.

### 3.4.2 Network training

A fully connected feedforward network was used for the experiments. The network was trained by using backpropagation with conjugate-gradient optimization [1]. One training set and two test sets were used. During training, after each 10 iterations, the network was evaluated by using the first test set. The training was continued until the network converged. Since the resulting network after convergence does not necessarily perform best on an independent test set, the network that performed best on the first set was selected and then evaluated by using the second test set.

| | classified as arrow | classified as non-arrow | correct % |
|---|---|---|---|
| arrow | 116 | 2 | 98.3 |
| non-arrow | 4 | 113 | 96.6 |

**Table 3.2**: *The optimal performance on the first test set with an overall performance of 97.5%.*

| | classified as arrow | classified as non-arrow | correct % |
|---|---|---|---|
| arrow | 127 | 11 | 92.0 |
| non-arrow | 0 | 116 | 100 |

**Table 3.3**: *The performance on the second test set with an overall performance of 95.7%.*

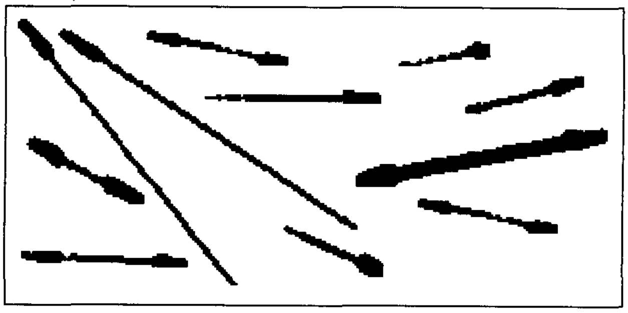


**Figure 3.5**: *The rejected arrows from the second test set.*

### 3.4.3 Results

Several experiments were performed by varying the number of hidden units and the initial random setting of the weights. In general, we observed that the performance of the network did not depend much on the number of hidden units, only the number of required iterations increased strongly with the number of hidden units. The best performance on the first test set was obtained when using 10 hidden units. After only 70 iterations, the network was able to classify 97.5% of the test samples correctly, while the performance on the training set was 100%. The confusion matrix is shown in Table 3.2. To be able to predict the classification performance in an operational interpretation system, this network was evaluated by using the second (independent) test set. In this experiment, the network correctly classified 95.7% of the objects. The confusion matrix for this experiment is presented in Table 3.3. This table shows that the network did not accept any road or house segments as arrows, and therefore the network can make a reliable distinction between arrows and other line symbols at the cost of 8% rejected arrows.

## 3.5 Discussion and Conclusions

In this chapter, a system has been proposed for the classification of irregular, poor quality arrows taken from hand-drawn Dutch public utility maps. The classification uses a neural network which is trained with a feature set based on the pseudo-Euclidean distances along the skeleton. From the experiments, we conclude that it is possible to reliably distinguish arrows from other line symbols. In earlier experiments [3] arrow recognition proved to be problematical when using features such as average width, length, aspect ratio, etc. While 14% of the arrows remained unclassified, still 5% of the objects recognized as arrows were classified incorrectly. The neural network, however, when evaluated by using the second test set, rejected 8% of the arrows, but no line symbols other than arrows were accepted.

The misclassified arrows from the second test set are shown in Fig. 3.5. The analysis of these arrows is not unambiguous. Some arrows are distorted, and one misclassification is caused by the use of an incorrect pen size, but others have a good appearance. This last category of misclassifications is most likely due to the small sample size behavior of the network.

An important advantage of the presented approach is that it should be applicable to many other drawing applications containing solid arrows. However, it is yet unclear how to use the current approach to classify symbols other than arrows, because arrows have the advantage that they can be recognized by the arrowhead. The arrowhead can be represented by a feature set composed of a limited set of pixels and their distance values. For symbols which do not have such a prominent feature, it is not trivial to design such a feature set. Therefore, our future research in this area will concentrate on further development of our approach for the classification of other line symbols and making the arrow classifier available in an operational map interpretation system.

### Acknowledgments

The maps used in this research were placed at our disposal by the Provinciale Noordbrabantse Energiemaatschappij (PNEM), a Dutch public utility. The financial support of Shell Nederland towards participation in this conference is gratefully acknowledged.

# Bibliography

[1] E. Barnard and R.A. Cole. A neural-net training program based on conjugate-gradient optimization. Technical Report CSE 89-014, Oregon Graduate Institute of Science and Technology, Beaverton, OR, July 1989.

[2] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics and Image Processing*, 27:321–345, 1984.

[3] J.E. den Hartog, T.K. ten Kate, and J.J. Gerbrands. Knowledge-based interpretation of public utility maps. accepted for publication in *Computer Vision and Image Understanding*, 1995.

[4] J.E. den Hartog, T.K. ten Kate, J.J. Gerbrands, and G. van Antwerpen. An alternative to vectorization: decomposition of graphics into primitives. In *Third Annual Symposium on Document Analysis and Information Retrieval (Las Vegas)*, pages 263–274, April 1994.

[5] L. Dorst and P. Verbeek. The constrained distance transform: a pseudo-Euclidean recursive implementation of the Lee algorithm. In *Signal processing III: theories and applications*, pages 917–920, Amsterdam, 1986. North-Holland.

[6] U. Montanari. A method for obtaining skeletons using a quasi-Euclidean distance. *Journal of the ACM*, 15:600–624, 1968.

[7] T.W. Ridler and S. Calvard. Picture thresholding using an iterative selection method. *IEEE Trans. Systems, Man and Cybernetics*, 8(8):630–632, August 1978.

1234567

# Knowledge-based interpretation of utility maps

This chapter is based on the following paper:

J.E. den Hartog, T.K. ten Kate and J.J. Gerbrands, Knowledge-based interpretation of public utility maps. Accepted for publication in *Computer Vision and Image Understanding*, 1995.

# 4

# Knowledge-based interpretation of utility maps


## Abstract

*In this chapter, a knowledge-based approach to the interpretation of public utility maps is presented. The interpretation is based on two types of a priori knowledge which are represented in a semantic network. The semantic network provides an efficient representation for reasoning with a priori knowledge and interpretation results. An interpretation system for maps based on the semantic network is described. In this system, a top-down control mechanism is integrated with a bottom-up object recognition strategy. The interpretation system has been tested on public utility maps, and the results are presented and evaluated.*


**Keywords:** map analysis, knowledge-based drawing interpretation, knowledge-based image analysis, graphics recognition.

## 4.1 Introduction

In recent years there has been an increasing use of computers in engineering applications. For many public utility organizations, digital information is the primary source for design, planning and maintenance. Despite the ongoing increase in the digital management of spatial information, the major part of communication is still based on paper maps. Paper, however, does not facilitate the management of spatial information. Electronic handling of this information is required to allow for easy storage, retrieval, reproduction, exchange and editing. Even though most current work is directly stored in digital form, an enormous amount of paper maps still exists which needs to be converted. Mere digitization by means of a scanning device is not sufficient to provide for efficient management by means of a geographic information system (GIS). As a consequence, digitization has to be succeeded by a step in which structure and objects are recognized. At the moment, three approaches to achieve an interpretation of structure and objects can be distinguished. In the first approach a human operator manually points out each object. Clear disadvantages of this approach are the many man years involved and therefore the high costs of the conversion process, while, due to the monotonous work, 2% to 3% of the interpretation contains errors. In the second approach, the digitized maps are vectorized. Vectorization combined with an efficient user interface requires fewer manual operations, thus achieving some speed-up of the tedious conversion process. In the third approach, the recognition of objects and structure is automated, and only in complex or conflicting situations is the operator involved. It is estimated that in the Netherlands alone, manual conversion of the maps at the cadaster, the telecommunication service and the public utilities could take about 10,000 man years. It should be clear that the third approach is the most promising method to accomplish a conversion of current analog maps within the near future and with a substantial reduction of the conversion costs. Therefore, there is a considerable interest in the research to explore new techniques for automatic conversion.

In the next section, current approaches and techniques for automatic map and drawing interpretation are discussed. Section 4.3 describes three concepts for the conversion of line drawing images into a high-level description. The results are presented and discussed in Section 4.4. In Section 4.5, the results are discussed. Section 4.6, finally, contains some concluding remarks and our suggestions for future research.

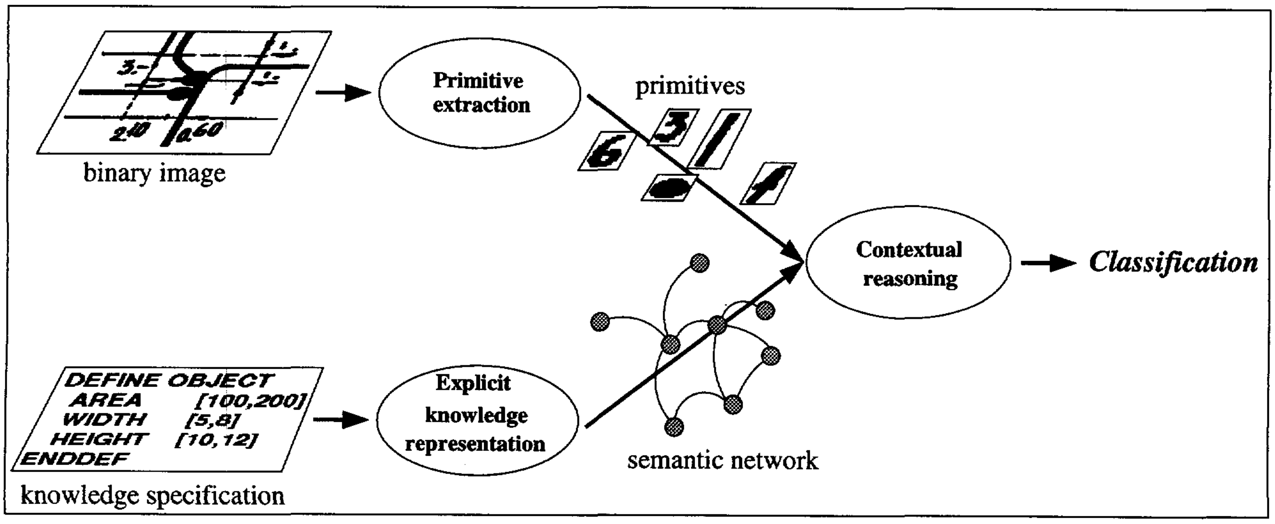



**Figure 4.1**: *The three basic concepts.*

## 4.2 Related work

In the past five years the knowledge-based interpretation of line drawings and other data, like angiograms and aerial images, has emerged as an important research topic. Though all these approaches are called "knowledge-based", the meaning of this term differs enormously. Knowledge can be implicit and hidden in the source code [7, 24, 28] or made more explicit by using rules [16, 33, 36]. It can be task specific [4, 7, 18, 35] or more general [12, 31, 33, 36]. And sometimes the knowledge is partly composed of formerly derived results, e.g. [16].

Despite the research efforts in knowledge-based interpretation, most of the research in the area of document analysis still concentrates on the design of specific algorithms rather than on image understanding [9, 13, 21, 23, 32, 37, 38]. Regarding line drawings, a common approach to convert a rasterized line drawing into an object description is to develop application specific algorithms to extract the objects from the image, and apply these algorithms in a static bottom-up sequence [1, 2, 3, 4, 5, 20, 25, 26, 30, 34]. An often encountered first step in such a sequence is separating the text symbols from the graphics, using features like the size and collinearity of the text [4, 20, 26]. Following this step, the text can be classified by an OCR module, and the remaining graphics are approximated by using vectorization. Graphical objects can then be extracted by grouping appropriate vectors [3, 4, 16, 18, 20]. Some knowledge-based approaches to group vectors into objects using a rule-based system have been described in the literature [16, 34]. However, rule-based reasoning is traditionally also bottom-up, and the main disadvantage of a bottom-up interpretation is the difficulty of guiding the interpretation process and to correct mistakes made at a lower level. Due to their bottom-up nature, many drawing interpretation systems have no method to guide the interpretation process.

An interesting approach to the knowledge-based interpretation of drawings is Anon, a schema-based system described by Joseph and Pridmore [19]. This system comprises a set of schemas which represent the entities to be recognized in the image. Each schema consists, among others, of a geometrical object description and a number of C-functions to interface with the image processing stages. The geometrical description has to be satisfied by the results of image processing before the instantiation of a schema class. In this concept, all declarative knowledge, i.e. the geometrical object specification, is represented by the program code, which requires reprogramming when the application is changed. Procedural knowledge in Anon consists of a set of 191 rules. An LR(1) grammar is used as an interpreter to limit the number of applicable rules. However, no mechanism like meta rules is described to order or select the applicable rules.

Work on automatic drawing interpretation has been reported by Pasternak and Neumann [27]. An adaptable drawing interpretation system named Adik is described which uses an explicit format to represent knowledge. However, only the declarative part of the knowledge is made explicit. A specification language is used to represent the knowledge needed to group geometrical entities, such as lines, into high level objects. The specification language used in Adik reduces the time to adapt the interpretation to other applications significantly. However, the procedural knowledge is not made explicit. In Adik, the interpretation sequence, i.e. the order in which object classes are instantiated, is determined automatically by the availability of the geometric components of the objects. An object is only instantiated if all components are detected while no control mechanism exists for generating hypotheses for the interpretation process.

From this overview, it may be concluded that relatively little research on knowledge to guide the interpretation of drawings has been reported. We propose to enhance the control of the interpretation process by utilizing task-specific knowledge. Combining the task-specific knowledge with derived results renders the possibility for a top-down approach in which goals are generated for the inter-

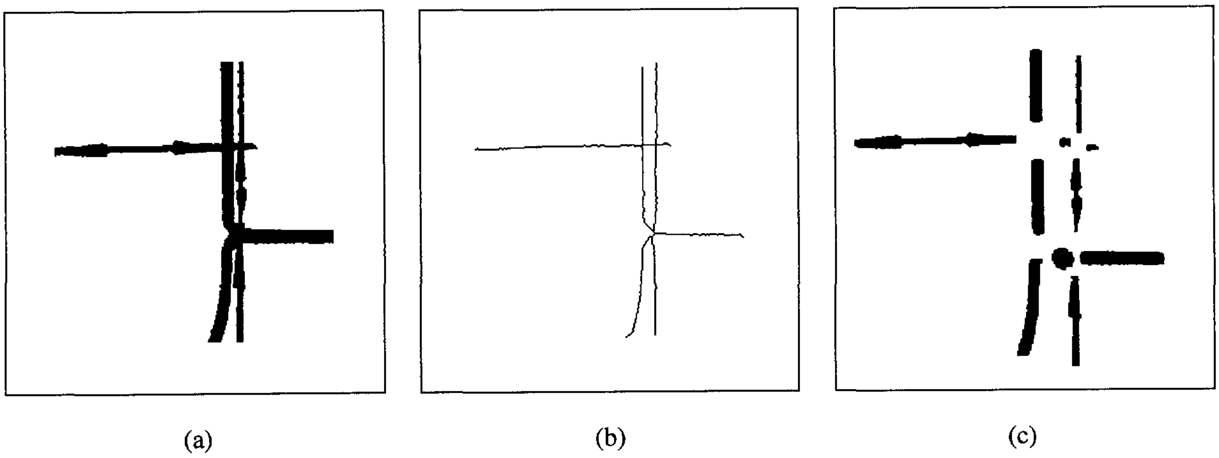



**Figure 4.2**: *The connected component in (a) is decomposed by propagating the fragments of its skeleton in (b) into the primitives in (c) which are shown in* ***exploded view.***

pretation process. Selecting the most promising goals first opens the way to control and guide the interpretation.

For technical drawings in particular, a knowledge-based approach to interpretation seems appropriate, as technical drawings are usually highly structured documents drawn according to explicit drawing conventions, using a limited and well known set of symbols. Using knowledge about the structure of the drawings, such as the spatial relationships between objects, appears to be useful for effectively steering the interpretation and detecting errors and inconsistencies in the interpretation. Providing a mechanism for control of the interpretation order of the drawing may therefore lead to a successful drawing interpretation system. In the next section we propose a first step towards the development of such a system.

## 4.3 Concepts

The aim of our research is to develop a system which allows for easy adaptation to other drawing applications. In this chapter, an initial approach to an application independent interpretation mechanism is proposed. Further, the use of an explicit representation of the knowledge is proposed to allow for the easy adjustment of the prior knowledge, thereby enhancing the flexibility of the system.

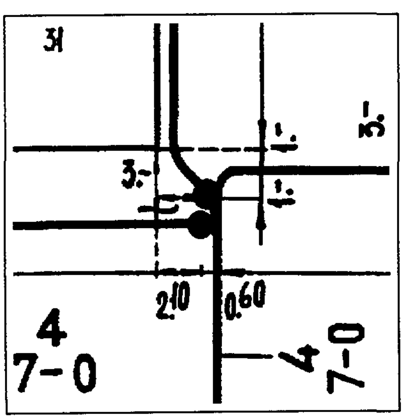



**Figure 4.3**: *A piece of a utility map.*

To avoid confusion about what is meant by knowledge, a definition to be used in the remainder of this chapter is given first. This definition should not be regarded as a universal definition of knowledge. The intention is to provide a clear and unambiguous definition of the knowledge used by the proposed system.

**Definition:** *Knowledge is the information known about the application in* **advance**, *which is used for the automatic interpretation and which is represented in an* **explicit** *way.*

As a first step towards a more generic interpretation of technical drawings, three basic concepts are proposed:

1. **Primitive extraction**. A representation of the binary image by its connected components.

2. **Explicit knowledge representation.** Separation of application-specific knowledge from the implementation using a dedicated knowledge specification language.

3. **Contextual reasoning**. A mechanism for both bottom-up and top-down interpretation of the image based on prior knowledge about document structure and appearance of map objects.

These three concepts and their mutual relationships are shown in Fig. 4.1. In this figure, the concepts are represented by the ellipses while the arrows represent information flow between the concepts. First, the binary input image is processed and the contents of the low-level bitmap are described as a set of primitive objects. The concept of primitive extraction is presented in Section 4.3.1.

The knowledge specification which describes all a priori knowledge about the application is represented by a semantic network where each node denotes a geometrical

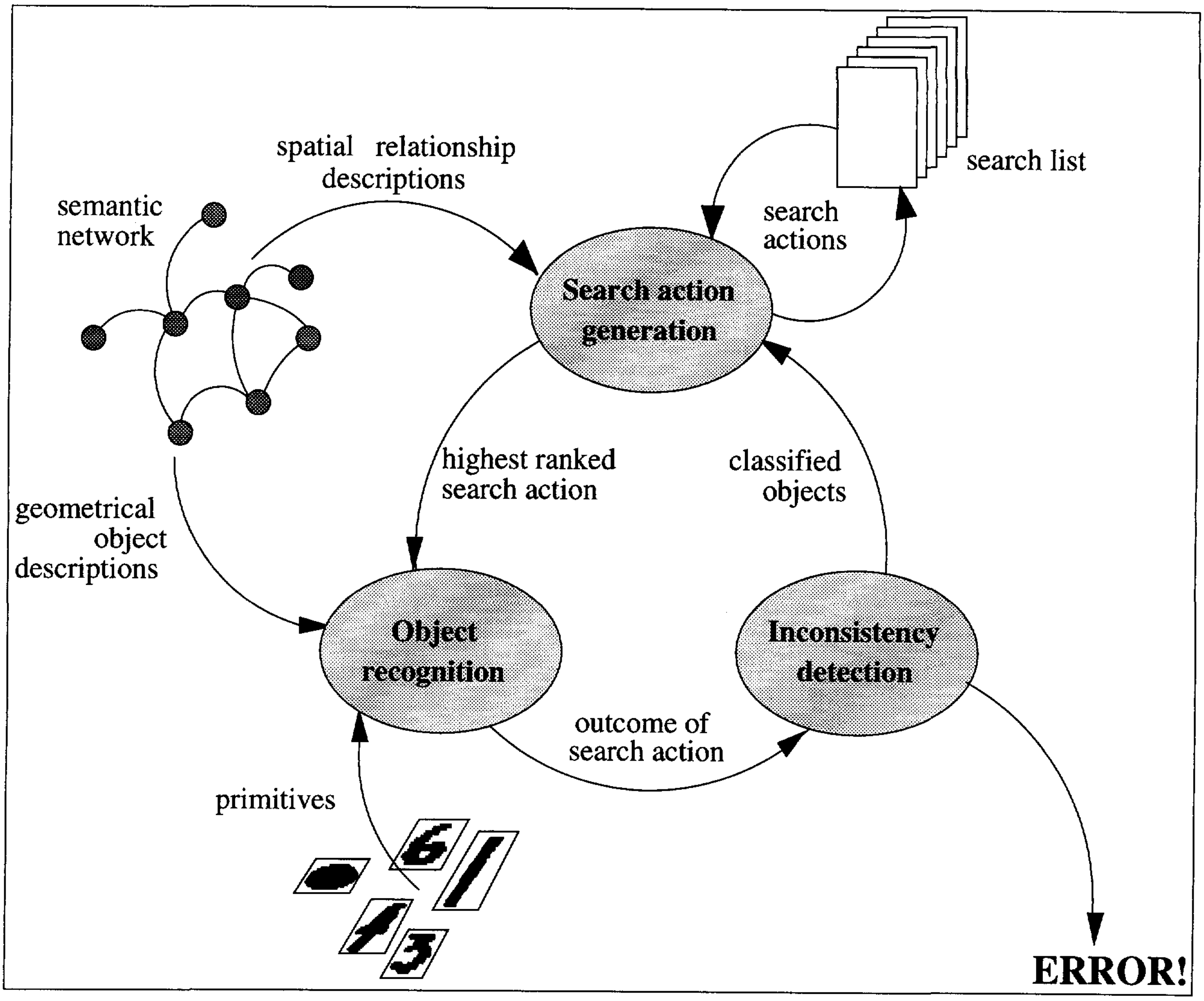



**Figure 4.4**: *The concept of contextual reasoning is shown as a cycle of three alternating processes.*

description of an object type which may occur in the image, while the links in the network construct spatial relationships between object types.

Both the extracted primitives and the semantic network are input for the interpretation process. The interpretation is based on the concept of contextual reasoning which combines a bottom-up approach to object recognition with a top-down approach to interpretation control. Object recognition as well as interpretation control require knowledge which is provided by the semantic network. The geometrical description represented by the nodes of the network is necessary to classify the primitives, while the spatial relationships are used to generate new goals.

For comprehensibility of the knowledge specification language, the concept of contextual reasoning is explained first in Section 4.3.2, followed by a discussion on explicit knowledge representation in Section 4.3.3.

### 4.3.1 Primitive extraction

As discussed in Section 4.2, skeleton vectorization is an often encountered step in the interpretation of maps, engineering drawings and other line drawings [2, 4, 11, 16, 18, 20]. Advantages of vectorization are obviously reduction of memory requirements and a more abstract description of the information in the image. Further, vectorization allows for efficient spatial reasoning on the objects in the image.

However, in spite of the attractive simplicity, rigorous vectorization may introduce unwanted inaccuracies. Almost every technical drawing contains symbols and objects which cannot be described both efficiently and accurately by means of a vector representation. The vectorization of small or curved lines for example, may result in a very coarse estimation which may obstruct proper recognition. Consider the drawing fragment in Fig. 4.3 which contains several graphical objects. Most of these objects have a very specific appearance. Reducing these objects to a single line discards important morphological information and this complicates their proper recognition.

Though vectorization is a useful technique in the process of interpreting technical drawings, mere vectorization is not sufficient to describe all pixel information accurately. Therefore, vectorization should be combined with a method which preserves morphological object information. We developed a method to decompose graphical ob-

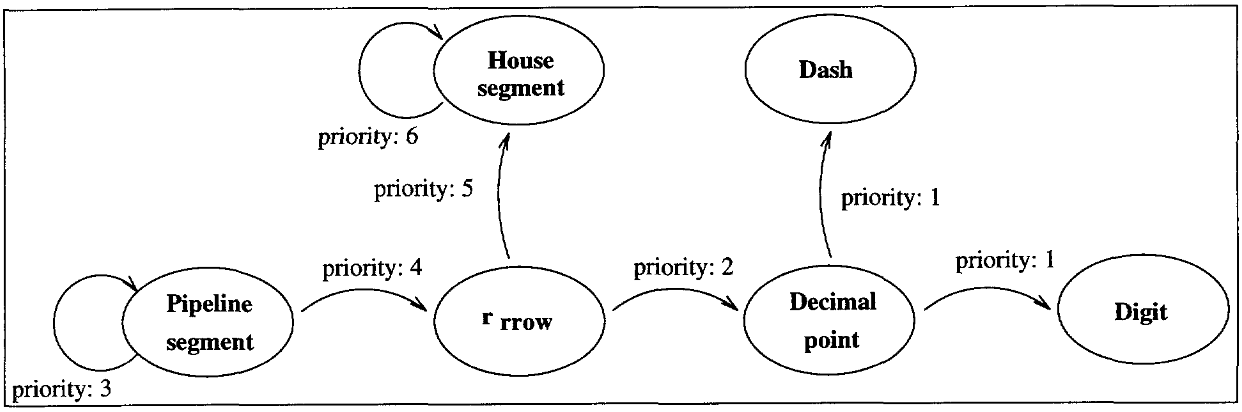



**Figure 4.5**: *An example of a semantic network.*

jects into their constructing primitives while preserving morphological information [8].

The algorithm decomposes the binary input image into primitives in two steps. First, a distinction is made between large and small connected components in the image, based on the area of the components. Following this separation step, each small component is stored in an individual image, while the large components, which usually consist of connected graphical symbols, are processed further. In the next step these remaining components are broken down into basic primitives.

In this step, each large object is skeletonized using the pseudo-Euclidean distance skeleton [6]. The skeleton is then searched for short segments which are removed to prevent fragmentation of the components in a later phase. Next, the skeleton is decomposed into its segments using the branch points as breaking points. The pixels in the original binary image belonging to a specific skeleton segment are reconstructed using the constrained distance transform [10]. Each skeleton segment is reconstructed in this way and the result is stored in a separate bitmap. An example of such a reconstruction is shown in Fig. 4.2.

Describing a component in terms of its primitives has the advantage that no morphological information is lost, as opposed to a vector description. Another advantage is that all primitives can be stored in separate bitmaps, which enables the application of image processing algorithms on individual primitives. This opens the way to calculate features for individual primitives using image processing. A vector-based object description which reduces primitives to single lines does not allow for the calculation of many useful features for individual primitives, e.g. area, perimeter, features based on the minimal enclosing rectangle [14, 17], and moment-based features [15, 22]. Furthermore, individual representation of primitives allows for their recognition by using template matching. Template matching in the original binary image will often yield erroneous results due to interference with neighboring primitives, while representation of each primitive in a separate bitmap does not have this disadvantage. Therefore, the described approach to low-level processing is a very important basis for the remaining interpretation process.

## 4.3.2 Contextual reasoning

### 4.3.2.1 Bottom-up versus top-down

The concept of contextual reasoning is based on alternating a top-down process, a bottom-up process and a process for inconsistency detection to obtain a reliable interpretation of the drawing in an efficient way. The control cycle of these three processes is shown in Fig. 4.4.

The bottom-up process receives a search action from a top-down process to search for a specified object type in a restricted area. A priori knowledge describing the geometry of the object type searched for is available to the bottom-up process. All primitives overlapping with the search area are matched with this geometric specification. If the features of the primitive correspond with the geometrical specification, the primitive is offered as a candidate for classification to the verification process. If no inconsistencies with former results are detected, the search results in an identification and a series of new search actions will be added to the search list.

The mechanism to generate new search actions is based on the perception that each object type is related to other object types. For example, an arrow usually depicts a distance between two objects, e.g. a house and a pipeline. Furthermore, related to an arrow is a dimensioning, a numerical representation of the depicted distance. Thus, detection of an object immediately generates expectations about other objects in its neighborhood, which are very suitable to generate new goals for the interpretation process. When an arrow is detected, search actions for a house, a pipeline, and the objects which make up the dimensioning are generated for a small region of interest (ROI) around the arrow. Each time a new object is detected, the top-down process collects all related object types and generates new search actions for the search list. The search list is sorted on the priorities associated with each search action. The priority for a search action is defined by the user who can assign priorities to the relationships between the objects. Each search action generated from a relationship is assigned the corresponding priority and inserted in the search list. The search action at the top of the search list is then distributed to the bottom-up pro-

| | |
|---|---|
| **Initial Search list:** | pipeline segment, prior. 3 |
| **Event #1:** | *pipeline segment found* |
| **Search list:** | pipeline segment, prior. 3<br>arrow, prior. 4 |
| **Event #2:** | *pipeline segment found* |
| **Search list:** | pipeline segment, prior. 3<br>arrow, prior. 4<br>arrow, prior. 4 |
| **Event #3:** | *pipeline segment NOT found* |
| **Search list:** | arrow, prior. 4<br>arrow, prior. 4 |
| **Event #4:** | *arrow found* |
| **Search list:** | decimal point, prior. 2<br>arrow, prior. 4<br>house segment, prior. 5 |
| **Event #5:** | *decimal point found* |
| **Search list:** | digit, prior. 1<br>dash, prior. 1<br>arrow, prior. 4<br>house segment, prior. 5 |
| **Event #6:** | *digit found* |
| **Search list:** | dash, prior. 1<br>arrow, prior. 4<br>house segment, prior. 5 |
| **Event #7:** | *dash found* |
| **Search list:** | arrow, prior. 4<br>house segment, prior. 5 |
| **Event #8:** | *arrow NOT found* |
| **Search list:** | house segment, prior. 5 |
| **Event #9:** | *house segment found* |
| **Search list:** | house segment, prior. 6 |
| **Event #10:** | *house segment found* |
| **Search list:** | house segment, prior. 6 |
| **Event #11:** | *house segment NOT found* |
| **Search list:** | EMPTY |

**Figure 4.6**: *The interpretation sequence.*

cess which tries to detect new objects in the assigned ROI in the image. Because a search for an object is based on contextual evidence, and the search area is restricted to a small and confined part of the image, the number of search operations as well as the number of incorrect object classifications will be reduced, thus rendering the interpretation more efficient and reliable than non-contextual straightforward approaches to object classification.

The object types and their spatial relationships construct a semantic network where the links make up the procedural part of the knowledge, while the nodes represent declarative knowledge. Each node in the network represents a description of the geometry of one object type, while each link represents a spatial relationship between two object types. A small example of a semantic network is shown in Fig. 4.5.

To gain further control of the top-down interpretation process, the user is allowed to assign priorities to specific search actions. In Fig. 4.5 the highest priority, i.e. priority 1, is assigned to the most important search actions. The priority mechanism is very useful to increase the reliability of the interpretation. By assigning highest priority to search actions for object types which are known to be recognized reliably, the number of misclassifications will further decrease.

#### 4.3.2.2 An example

Next, the process of contextual reasoning will be demonstrated with an example. Consider the drawing fragment in Fig. 4.7. In this very small part of a utility map, six object types and eight objects can be distinguished: two pipeline segments, one arrow, two house segments, a decimal point, a digit, and a dash. The events in Fig. 4.6 correspond with the numbers in Fig. 4.7. The interpretation sequence in Fig. 4.6 starts with an initial search action for a reliably recognized object type, in this example a pipeline segment. From this action **event #1** results: a pipeline segment is found. Next, the search list is updated using the relationships represented by the semantic network in Fig. 4.5. The search actions in the list are sorted on priority and the search action at the top of the list is processed first. Note that a smaller value denotes a higher priority, thus spatial relationships with priority value 1 will generate search actions with the highest priority. In the case of

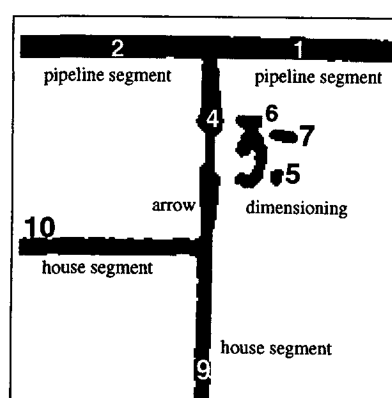



**Figure 4.7**: *An example image.*

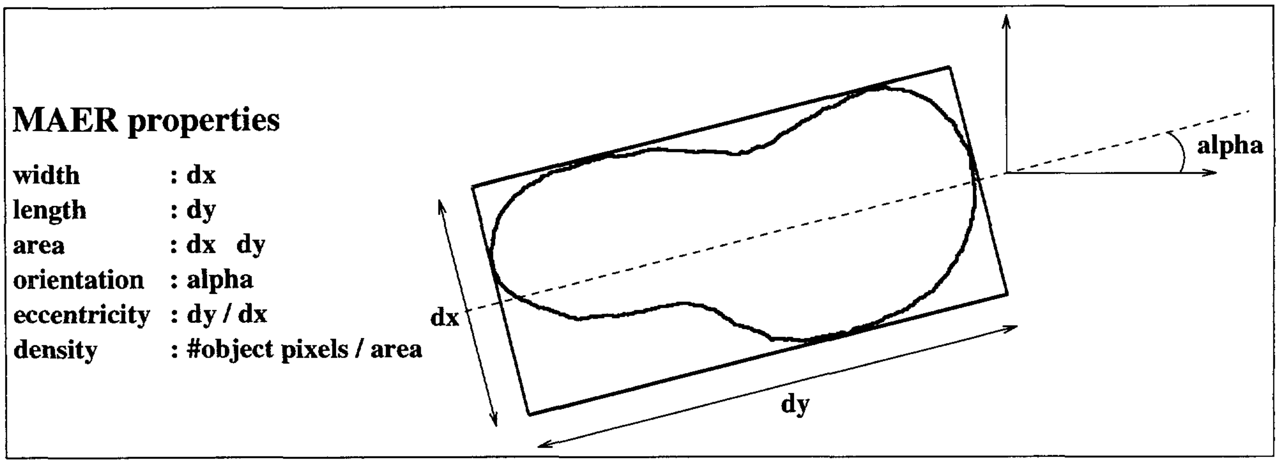



**Figure 4.8**: *An example object with its MAER.*

search actions with equal priority numbers, the order of the generated search actions is chosen arbitrarily. In this case, the detection of a pipeline segment results in search actions for two object types: a pipeline segment and an arrow. Because the search action for the pipeline segment has highest priority, this action is put at the top of the list. Again a pipeline segment is searched and found. Consequently, the search list is updated and two new search actions for a pipeline segment and an arrow are added to the list. Next, it appears that the search action in front of the list is unsuccessful: no pipeline segment can be detected. Therefore, no new actions are added to the search list and the interpretation continues with the next action to search for an arrow. The interpretation, i.e. processing of the search list, continues until the search list is empty (**event #11**).

#### 4.3.2.3 Detection of inconsistencies

Even when both object recognition and control of the interpretation process are excellent, it is still not realistic to assume perfect recognition of all objects. An A0-sized map (1 square meter or 1600 square inches) contains thousands of objects, and therefore it is more than likely that, due to noise or poor quality of the map, some objects will be misclassified.

Primarily, these misclassifications will increase the costs of the interpretation process because of the enormous problem of detection afterwards. Therefore, it is important to be able to detect misclassifications early, during interpretation. The automatic finding of errors is based on detecting inconsistencies, either between two or more results, or inconsistencies between results and a priori knowledge.

There are several options available to detect these inconsistencies. For example, each object could be classified by two independent classification algorithms. In the case of conflicting outcomes, a third classifier could make the final decision. To validate the OCR, recognized words could be matched with a dictionary of the application, while OCR of dimensionings could be validated using the length of the corresponding arrow.

Because the aim of the research is to develop a generically applicable system, research has concentrated on a generic method which is based on the detection of contextual inconsistencies. The method is based on the perception that three types of spatial relationships may be distinguished:

1. The **optional** relationship.
2. The **essential** relationship.
3. The **negative** relationship.

An **optional** relationship between two objects indicates that if the first object is found, it is likely to find the second in its vicinity. An **essential** relationship implies that if the first object is found, the related object **must** be found also. An example of an essential relationship is the arrow which is always accompanied by a dimensioning. The **negative** relationship, finally, is the opposite of the essential relationship. If this type of relationship is shared by two object instances, a misclassification has occurred. For example, a pipeline segment cannot have a distance to a house of less than 0.5 meters.

The essential and the negative relationships provide a *simple and generically applicable mechanism to detect inconsistencies* in the interpretation. Each time an object is found which shares one of these two relationships with other object types, these object types are searched first. If the outcome is inconsistent, either the help of an operator is invoked, or the inconsistent classification is rejected.

### 4.3.3 Explicit knowledge specification

The development of a system for knowledge-based image interpretation consists of a laborious cycle of design, implementation, testing and redesign. However, if the system implementation and the application knowledge are separated, the time needed to develop the interpretation system can be reduced significantly. Separating knowledge and the program code, by making the knowledge explicit in a special knowledge-representation format, enables flexible adaptation of knowledge. Adjusting knowledge at run time will bypass the tedious trajectory of recompilation and linking. Furthermore, knowledge represented in a specification language will offer much more insight in the behavior of the system than when obfuscated by statements of program code. Nevertheless, the major advantage of the ability to edit knowledge in an easy and flexible way is the potential gain concerning the effort to convert the system to another application. Tedious reprogramming can often be avoided because mainly the explicit task-specific knowledge has to be edited to suit the new application.

As discussed in the previous section, the reliability as well as the efficiency of the interpretation depend on the control of the interpretation. Therefore, especially flexible manipulation of procedural knowledge is important when developing an interpretation system.

In Section 4.3.2, a control mechanism for the interpretation was presented. The control, i.e. the generation of new goals for the search list and processing of the search list, is based on the semantic network constructed by the spatial relationships among the objects. In the remainder of this section, a method for the explicit representation of the semantic network is presented. Two types of objects within the network are distinguished at the moment:

1. Nodes, describing the geometry of the object types.
2. Arcs, describing the spatial relationships between the object types.

#### 4.3.3.1 Geometric object description

Currently, a geometric specification consists of a name, a set of numerical features, and for each feature a range of the allowed values. To describe each object type, a standard set of features is provided. Some important features are based on the minimum area enclosing rectangle (MAER) of the object, described in [14, 17]. The MAER of a primitive allows for the calculation of a set of rotation invariant features, but it also provides a method to estimate the orientation of an arbitrary primitive. An example primitive with its MAER and some features to be calculated from the MAER are shown in Fig. 4.8. The complete set of standard feature types is listed below.

- The minimum, maximum and average width along the skeleton.
- The length of the skeleton.
- The area of the object.
- The eccentricity, i.e. the ratio of the length of the major axis to the length of the minor axis of the MAER.
- The density, i.e. the ratio of the area of the primitive to the area of the MAER.
- The orientation of the major axis of the MAER to the x-axis.
- The width and height of the MAER.

It is not necessary to define all possible features for all objects. For example, for a digit, it is not very meaningful to define the features concerning the width along the skeleton. Therefore, only a specification of significant features is required. Some examples of a geometrical description are shown below.

```
DEFINE GEO_SPEC
   objname      Arrow
   max_width    [4.0, 7.5]
   avg_width    [1.4, 7.0]
   length       [60.0, 1000.0]
   area         [150, INF]
   eccentricity [10.0, INF]
ENDDEF
```

```
DEFINE GEO_SPEC
   objname      PIPELINE_SEGMENT
   max_width    [12.0, 16.0]
   avg_width    [12.0, 14.0]
   length       [60.0, INF]
   area         [200, INF]
ENDDEF
```

```
DEFINE GEO_SPEC
   objname      DECIMAL_POINT
   area         [5, 50]
   eccentricity [1.0, 1.25]
   maer_width   [2.0, 5.0]
   maer_height  [2.0, 5.0]
ENDDEF
```

The first geometrical description describes an arrow. In this case, an arrow is described by six geometrical features, and for each feature a range describing the minimum and maximum allowed value is given. For the length, the area, and the eccentricity only a minimum value is given. To provide optimal flexibility, the user can easily attach new features to an object type.

#### 4.3.3.2 Spatial relationship description

Similar to the geometrical object description, the spatial relationship also consists of a set of features describing the properties of the relationship between two objects. The following set of spatial features is provided by the current implementation:

- The type of the relationship, i.e., optional, essential, or negative.
- Two identifiers specifying the first object and the related object.
- The priority of the search action resulting from the relationship.

- The radius of the ROI to search for the related object.
- The angle between two objects based on the orientation of the MAERs.
- The angle between two objects based on the vectorization of the skeletons.
- A feature to specify whether the two objects should overlap.
- A feature to specify whether the first object should be inside the related object, and vice versa.

The knowledge descriptions of some of the spatial relationships in Fig. 4.5 are given below:

```
DEFINE REL_SPEC
   From      PipelineSegment
   To        Arrow
   Type      Optional
   Priority  4
   Radius    5
   VecAngle  [85.0, 95.0]
ENDDEF
```

```
DEFINE REL_SPEC
   From      PipelineSegment
   To        PipelineSegment
   Type      Optional
   Priority  3
   Radius    5
   VecAngle  [-5.0, 5.0]
ENDDEF
```

```
DEFINE REL_SPEC
   From      Arrow
   To        DecimalPoint
   Type      Essential
   Priority  2
   Radius    30
ENDDEF
```

```
DEFINE REL_SPEC
   From      DecimalPoint
   To        Digit
   Type      Essential
   Priority  1
   Radius    10
ENDDEF
```

Similar to the geometrical object specification, the spatial relationship description also provides a mechanism to expand the set of features.

## 4.4 Results

### 4.4.1 Experiments

The developed techniques were applied to a set of public utility plans. These plans are hand drawn to a scale of 1 to 500 and they represent the position of the pipeline system with respect to landmarks, for example the corner of a house.

Two A0-sized original plans were available for our experiments, but due to hardware limitations only US-Legal (8.5"×14") sized parts were taken from these maps. From each map, two parts were scanned in grey value at a density of 400 dpi, followed by an automated isodata-like thresholding algorithm [29] to obtain binary images.

From the four available binary images, two images (one from each utility map) were used to optimize the interpretation system while the remaining two images were used for testing and evaluation. There was no overlap between the images used to optimize the system and the images used to evaluate the system.

For an applicable interpretation system, it is very important to reduce the probability of errors. However, decreasing the probability of misclassifications will, in general, increase the probability of a reject. In the experiment, the spatial relationships between the object types were used to detect inconsistencies in the classification. In the case of a detected inconsistency, the classification was rejected to reduce the number of misclassifications. As a consequence, the number of rejects increased considerably.

All experiments were carried out on a SUN Sparcstation 10 equipped with 128 megabytes of memory. A part of one of the test images is shown in Fig. 4.9.

### 4.4.2 Evaluation

Regarding interpretation of the public utility plans, five main symbols have to be recognized automatically: pipeline segments, house segments, arrows and the dots and digits of the dimensionings. The dimensionings are either composed of a dot and three digits or a dot, a digit and a dash. In this evaluation of the experiments, the dashes and the digits were regarded as one symbol type.

For each object type, the recognition was manually categorized into one of the four following classes:

1. correct, the percentage of the correctly classified objects (with respect to the total number of objects in that specific class).
2. reject, the percentage of the objects which remained unclassified (with respect to the total number of objects in that specific class).
3. misclassification, the proportion of all the incorrectly classified objects (with respect to the total number of objects in that class).
4. false accepts, the proportion of objects from other classes which were accepted incorrectly (with respect

| | total | correct % | rejects % | misclass. % | false accepts % |
|---|---|---|---|---|---|
| pipeline m. | 3.4 m | 93.2 | 6.8 | 0.0 | 3.0 |
| pipeline # | 221 | 69.7 | 30.3 | 0.0 | 2.7 |
| arrows | 154 | 78.6 | 20.1 | 1.3 | 1.6 |
| houses m. | 4.2 m | 79.0 | 17.9 | 3.1 | 0.3 |
| houses # | 173 | 74.6 | 22.5 | 2.9 | 0.6 |
| dim. dots | 151 | 76.8 | 23.2 | 0.0 | 4.1 |
| dim. digits | 437 | 73.4 | 25.3 | 1.3 | 0.0 |

**Table 4.1**: *The classification results using contextual reasoning.*

| | classification | | | | | |
|---|---|---|---|---|---|---|
| type | pipeline % | arrow % | house % | dim. dots % | dim. digits % | unclassified % |
| pipeline m. | 93.2 | | | | | 6.8 |
| pipeline # | 69.7 | | | | | 30.3 |
| arrows | | 78.6 | 1.3 | | | 20.1 |
| houses m. | 2.3 | 0.8 | 79.0 | | | 17.9 |
| houses # | 2.3 | 0.6 | 74.6 | | | 22.5 |
| dim. dots | | | | 76.8 | | 23.2 |
| dim. digits | | 0.2 | | 1.1 | 73.4 | 25.3 |

**Table 4.2**: *The confusion matrix of the results using contextual reasoning.*

to the total number of accepts for this specific class). Note that generally a false accept will also be counted as a misclassification of an other object class.

An interpretation system should minimize the number of misclassifications and false accepts, because many incorrectly classified objects will make manual verification of *all* results necessary. Though rejects also require operator interaction, they are less severe shortcomings of the interpretation, because the operator only has to focus on the unclassified primitives instead of all classified primitives.

The interpretation is based on the classification of the primitives which are extracted from the binary image (Section 4.3.1), which makes it difficult to evaluate the recognition of all object types in a uniform way. In general, arrows, and the dots and digits from the dimensionings are represented by a single primitive. However, houses and pipelines are always represented by several primitives, and, as a consequence, it may happen that these objects are partly recognized. A partial recognition can be evaluated in two ways: by the proportion of the number of recognized primitives with respect to the total number of primitives which make up the object, or, by the proportion of the length of the recognized primitives with respect to the total length of the primitives composing the object. The former method of evaluation is related to the number of interactions to correct the result, whereas the latter method is related to the conversion throughput. Both evaluation methods are relevant measures of the performance of the system, depending on whether the interest is focused either on the number of corrective actions or on the unit length to be reclassified. For this reason, it is sensible from a practical point of view to provide both figures for the object types house and pipeline. The results of the recognition using contextual reasoning are presented in Table 4.1. The misclassifications are shown as a confusion matrix in Table 4.2.

In addition to the evaluation of the overall recognition performance it is equally important to evaluate the effect of the contextual reasoning mechanism. For this reason, a second experiment was carried out to evaluate global object recognition thereby leaving out knowledge about spatial relationships. In this experiment, a global search was made for each object type starting with the pipeline which is recognized best. After detection of the pipeline, the arrows, which are recognized second best, were searched, followed by the house segments, and the dimensioning dots and digits respectively. The results of this experiment are shown in Table 4.3.

Estimates of the public utility which provided the plans indicate that an operator using existing (manual) techniques, is capable of converting 200 meters of pipeline an hour (0.4 meter of drawn pipeline including arrows, dimensionings etc.). The two test images comprised approximately 3.4 meters of drawn pipeline, which is equivalent to 1700 meters of pipeline, or approximately one day's manual conversion. Automatic classification of the objects in these images, including segmentation and preprocessing, required about 30 minutes of computing time. One should bear in mind that this does not include manual

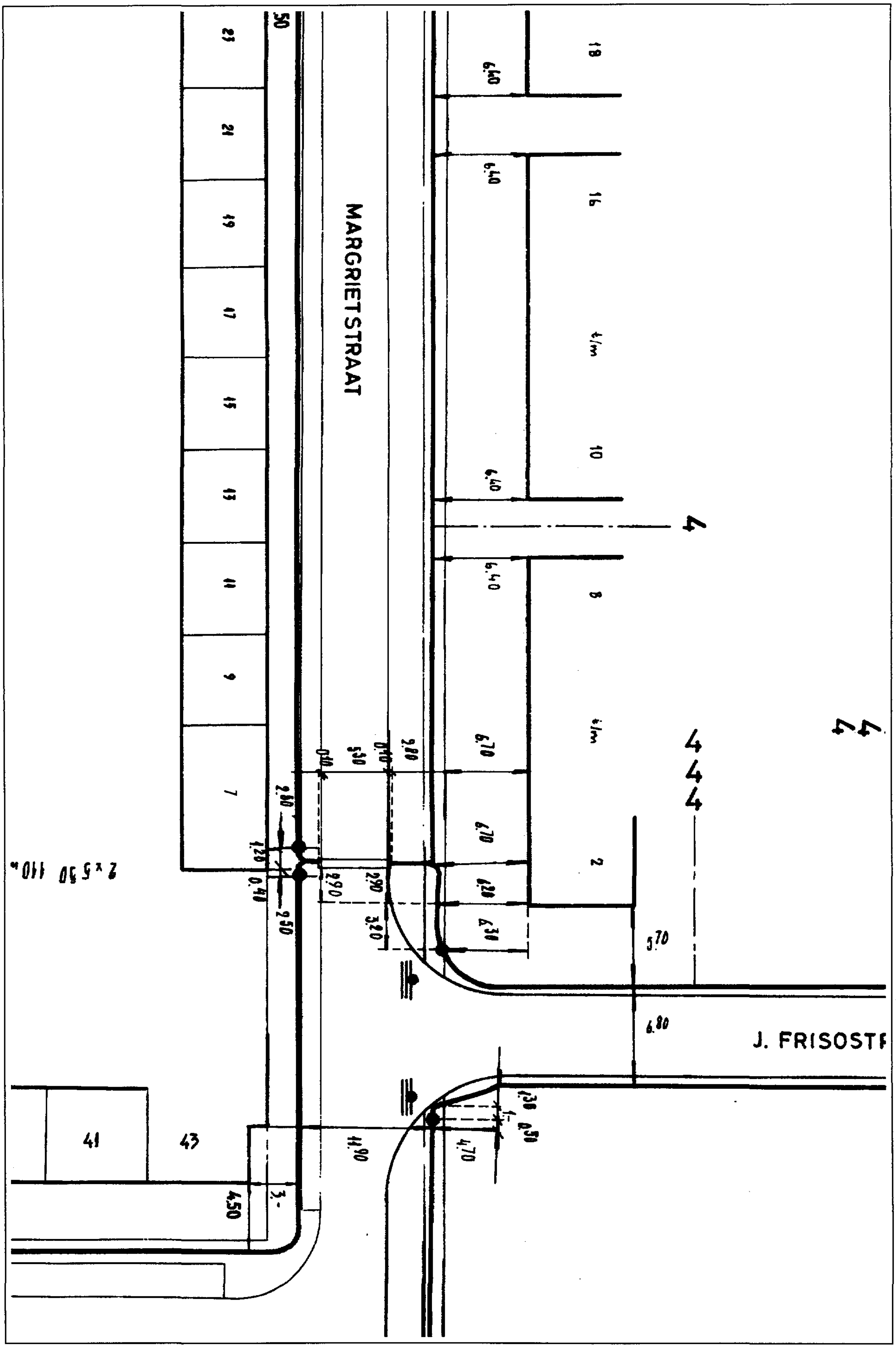



**Figure 4.9**: *A part of a test image*

| | total | correct % | rejects % | misclass. % | false accepts % |
|---|---|---|---|---|---|
| pipeline m. | 3.4 m | 93.2 | 6.8 | 0.0 | 3.0 |
| pipeline # | 221 | 69.7 | 30.3 | 0.0 | 2.7 |
| arrows | 154 | 85.9 | 13.5 | 0.6 | 5.0 |
| houses m. | 4.2 m | 85.0 | 12.2 | 2.9 | 6.8 |
| houses # | 173 | 82.7 | 15.6 | 1.7 | 15.9 |
| dim. dots | 151 | 94.3 | 5.7 | 0.0 | 20.4 |
| dim. digits | 437 | 93.2 | 5.5 | 1.3 | 54.2 |

**Table 4.3**: *The classification results using a global search.*

conversion of the rejected objects (about 20%) and OCR of the dimensionings.

Important aspects to consider are the causes of the rejects and the misclassifications. For rejects, the following causes can be distinguished:

1. The set of geometrical features was inadequate to recognize all objects of a specific type.
2. No related objects were detected in the context, and, consequently, no search action was made.
3. The contextual reasoning mechanism could not detect an essential relationship and rejected the classification.
4. Poor segmentation of the grey value image.

For the misclassifications three causes can be distinguished:

1. The set of geometrical features was inadequate to make a perfect distinction between all object classes.
2. Poor segmentation of the grey value image.
3. Violation of the drawing conventions by the draftsman.

Due to intra-class variance of the objects, it was not possible to classify all objects reliably with the current set of geometrical features. Approximately 70% of all rejected pipeline segments and a third of the unclassified arrows can be explained by an inadequate feature set, while this only accounts for 10% of the rejected dimensioning symbols.

Many of the unclassified objects were never searched as a consequence of a reject of related objects, while some correctly classified objects were still rejected because an essential relationship could not be detected. Lack of contextual evidence is the direct cause for approximately 75% of all unclassified dots and digits, and 30% of the arrows and house segments which remained unrecognized. From this it can be concluded that a large number of rejects is a consequence of the contextual reasoning mechanism. However, the results of leaving out the contextual reasoning mechanism are shown in Table 4.3. A global search for all objects reduces the number of rejects, but at the cost of a significant increase in the number of false accepts.

Though insufficient contextual evidence might be the direct cause for many rejects, this originates in approximately 40% of the cases from a poor global segmentation. Especially the segmentation of small objects, i.e. the dimensioning dots and digits, proved to be prone to error. Almost 7% of the dimensioning dots were not classified due to an inadequate segmentation, and as a consequence, many dimensioning digits remained unclassified.

When considering the misclassifications, three causes can be identified. Violation of the drawing conventions was only encountered in the experiment once, when some house segments were classified as pipeline segments. These misclassifications were caused by the use of an incorrect pen size by the draftsman. Very few misclassifications are caused by an incorrect segmentation (5% to 10% of all misclassifications) while all other misclassifications can be explained by limitations of the current feature set.

## 4.5 Discussion

Though work on a number of successful systems has been reported in the literature [4, 11, 16, 19, 26, 27, 34] no objective evaluation strategy has been described. As a consequence, it is very hard to make a quantitative comparison with other work. Therefore, we first concentrate on some qualitative aspects of the proposed system.

### 4.5.1 Advantages

The proposed system is based on three concepts, which offer a number of advantages over earlier systems. Describing the binary image in its constructing primitives offers an abstract representation of the image which supports the interpretation based on high-level spatial reasoning. Because low-level pixel information is preserved, the recognition of individual primitives is facilitated.

Another important advantage is the control mechanism based on spatial relationships between the objects. We conclude that this is a very flexible and efficient method to guide the interpretation and detect inconsistencies. Further, making both declarative and procedural knowledge available in an explicit format is a powerful concept which yields a significant reduction in the time and effort required to adapt the system to a specific application. In contrast with rule-based systems which usually require hundreds of rules, e.g. [19], the set of spatial relationships provides a compact knowledge description to guide the interpretation process.

### 4.5.2 Current limitations

In the current system, the input binary image is decomposed into its primitives. Following this step, the primitives are recognized by using knowledge about the geometrical appearance of object types and the spatial relationships between them. Though primitives are recognized within their context and are labeled as, for example, digits, arrows, house segments, it is difficult with the current knowledge specification to describe higher-level complex object types consisting of multiple simple objects.

An example of such a complex object type is the dimensioning which consists of several simple objects, i.e. a decimal point and two or three digits. A dimensioning and an arrow in their turn make up an even higher-level object indicating a specific distance between two other objects. The knowledge framework as presented in this chapter can describe the spatial relationships between primitives, but for better understanding of the image it is desirable to be able to describe the hierarchical relationships between objects. Furthermore, even though in our current application the number of spatial relationships is limited compared to rule-based systems, it is possible that for other application areas with more complex objects or larger sets of objects the number of spatial relationships in the knowledge specification will drastically increase. When the spatial relationships are organized in a hierarchical framework, unnecessary generation of search actions can be avoided, thus resulting in a more efficient interpretation. Therefore, it will be necessary to extend the current knowledge specification and contextual reasoning mechanism to allow for a hierarchical partitioning of the knowledge base.

Another limitation of the current system is its dependency on the quality of the binary image. The binary image is decomposed into primitives, but the output quality of the decomposition algorithm depends on the quality of the skeleton. For example, the occurrence of small holes in a binary object will have dramatic consequences for the skeleton, and therefore also for the decomposition into primitives. However, the same limitations are encountered with vectorization-based interpretation systems. Therefore, to improve the performance, it is important to concentrate on the development of a system which is able to use the gray level image when, due to a poor segmentation, the binary image is noisy or distorted.

## 4.6 Conclusions

From the experiments, we conclude that a fully automatic interpretation of these plans is not yet possible. However, an overall object recognition of approximately 80%, an error rate of about 2% and a reject rate of 20% make a considerable speed-up of the manual conversion process feasible, and part of our current research is therefore directed at developing a large-scale interpretation system based on the results of this research.

A poor segmentation and an inadequate feature set were identified as the two main causes for the rejects and misclassifications. It is therefore important to develop new geometrical features and to improve the segmentation to reduce the number of rejects and misclassifications. A better global segmentation can be obtained by enhancing the grey value image or by developing other segmentation techniques. However, contextual reasoning offers a mechanism to make predictions about the occurrence of objects in limited parts of the image. In the case of an inconsistent outcome, this information could be used to correct the local segmentation.

In the current implementation, the system can reject object classifications or invoke operator help in the case of inconsistencies. A next step towards a fully automated drawing interpretation system is to develop knowledge-based techniques to handle inconsistencies. Therefore, our future research will concentrate on new techniques for knowledge-based segmentation and inconsistency handling, and the representation of, as well as the reasoning with, hierarchical relationships.

### Acknowledgments

This research is part of ROCKI, Esprit project 5376, sponsored by the European Community. The maps used in this research were placed at our disposal by the public utility Provinciale Noordbrabantse Energiemaatschappij (PNEM). The assistance of Bernardus Holtrop with the tedious work evaluating the interpretation is gratefully acknowledged.

# Bibliography


[1] J. Alemany et al. Interpretation of paper-based maps. In *SPIE Appl. of Digital Image Processing X*, volume 829, pages 125–137, 1987.

[2] D. Antoine. CIPLAN, a model-based system with original features for understanding French plats. In *First Int. Conf. on Document Analysis and Recognition (Saint Malo)*, volume 2, pages 647–655, October 1991.

[3] J.F. Arias, C.P. Lai, S. Chandran, R. Kasturi, and A. Chhabra. Interpretation of telephone system manhole drawings. In *Second Int. Conf. on Document Analysis and Recognition (Tsukuba Science City)*, pages 365–368, October 1993.

[4] D. Benjamin, P. Forgues, E. Gulko, J.B. Massicotte, and C. Meubus. The use of high-level knowledge for enhanced entry of engineering drawings. In *9th IAPR Int. Conf. on Pattern Recognition (Rome)*, volume 1, pages 119–124, November 1988.

[5] L. Boatto, V. Consorti, M. Del Buono, S. Di Zenzo, V. Eramo, A. Esposito, F. Melcarne, M. Meucci, A. Morelli, M. Mosciatti, S. Scarci, and M. Tucci. An interpretation system for land register maps. *IEEE Computer*, 25(7):25–33, July 1992.

[6] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics and Image Processing*, 27:321–345, 1984.

[7] S. Colin and P. Vaxivière. Recognition and use of dimensioning in digitized industrial drawings. In *First Int. Conf. on Document Analysis and Recognition (Saint Malo)*, pages 161–169, 1991.

[8] J.E. den Hartog, T.K. ten Kate, J.J. Gerbrands, and G. van Antwerpen. An alternative to vectorization: decomposition of graphics into primitives. In *Third Annual Symposium on Document Analysis and Information Retrieval (Las Vegas)*, pages 263–274, April 1994.

[9] M.P. Deseilligny, H. le Men, and G. Stamon. Map understanding for GIS data capture: algorithms for road network graph reconstruction. In *Second Int. Conf. on Document Analysis and Recognition (Tsukuba Science City)*, pages 676–679, October 1994.

[10] L. Dorst and P. Verbeek. The constrained distance transform: a pseudo-Euclidean recursive implementation of the Lee algorithm. In *Signal processing III: theories and applications*, pages 917–920, Amsterdam, 1986. North-Holland.

[11] C.S. Fahn, J-F. Wang, and J-Y. Lee. A topology-based component extractor for understanding electronic circuit diagrams. *Computer Vision, Graphics and Image Processing*, 44:119–138, 1988.

[12] J.L. Fisher, S. Hinds, and D. D'Amato. A rule-based system for document image segmentation. In *10th IAPR Int. Conf. on Pattern Recognition (Atlantic City)*, pages 567–572, 1990.

[13] L.A. Fletcher et al. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(6):910–918, 1988.

[14] H. Freeman and R. Shapira. Determining the minimum-area encasing rectangle for an arbitrary closed curve. *Communications of the ACM*, 18(7):409–413, 1975.

[15] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Addision Wesley, Reading, Massachusetts, 1992. ISBN 0–201–50803–6.

[16] K.J. Goodson and P.H. Lewis. A knowledge-based line recognition system. *Pattern Recognition Letters*, 11(4):295–304, 1990.

[17] F. Groen, P. Verbeek, N. de Jong, and J. Klumper. The smallest box around a package. *Pattern Recognition*, 14:173–178, 1981.

[18] M. Ilg. Knowledge-based understanding of road maps and other line images. In *10th IAPR Int. Conf. on Pattern Recognition (Atlantic City)*, volume 1, June 1990.

[19] S.H. Joseph and T.P. Pridmore. Knowledge-directed interpretation of mechanical engineering drawings. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(9):928–940, September 1992.

[20] R. Kasturi, S.T. Bow, W. El-Masri, J.R. Gattiker, and U.B. Mokate. A system for interpretation of line drawings. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(10):978–992, 1990.

[21] R. Kasturi et al. A sytem for recognition and description of graphics. In *9th IAPR Int. Conf. on Pattern Recognition (Rome)*, pages 255–259, 1988.

[22] A. Khotanzad and Y.H. Hong. Rotation invariant pattern recognition using Zernike moments. In *9th IAPR Int. Conf. on Pattern Recognition (Rome)*, pages 326–328, November 1988.

[23] S.H. Kim, J.W. Suh, and J.H. Kim. Recognition of logic diagrams by identifying loops and rectilinear polylines. In *Second Int. Conf. on Document Analysis and Recognition (Tsukuba Science City)*, pages 349–352, October 1993.

[24] D. Madej. An intelligent map-to-CAD conversion system. In *First Int. Conf. on Document Analysis and Recognition (Saint Malo)*, pages 603–610, 1991.

[25] M.T. Musavi, M.V. Shirvalkar, E. Ramanathan, and A.R. Nekovei. A vision-based method to automate map processing. *Pattern Recognition*, 21(4):319–326, 1988.

[26] V. Nagasamy and N.A. Langrana. Engineering drawing processing and vectorization system. *Computer Vision, Graphics and Image Processing*, 49:379–397, 1990.

[27] B. Pasternak and B. Neumann. ADIK: an adaptable drawing interpretation kernel. In *Avignon 1993, 13th Int. Conf. on AI, Expert Systems and Natural Language (Avignon)*, volume 1, pages 531–540, May 1993.

[28] P. Puliti and G. Tascini. Interpreting technical drawings. *The Computer Journal*, 33(5), 1990.

[29] T.W. Ridler and S. Calvard. Picture thresholding using an iterative selection method. *IEEE Trans. Systems, Man and Cybernetics*, 8(8):630–632, August 1978.

[30] C. Shih and R. Kasturi. Extraction of graphic primitives from images of paper-based line drawings. In *Machine and Vision Applications 2*, pages 103–113, 1989.

[31] C. Smets, G. Verbeeck, P. Suetens, and A. Oosterlinck. A knowledge-based system for the delineation of blood vessels on substraction angiograms. *Pattern Recognition Letters*, 8:113–121, September 1988.

[32] S.N. Srihari et al. Document image analysis. In *8th IAPR Int. Conf. on Pattern Recognition (Paris)*, pages 434–436, 1986.

[33] S.A. Stansfield. ANGY, a rule-based expert system for automatic segmentation of coronary vessels from digital subtracted angiograms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(2):188–199, March 1986.

[34] S. Suzuki and T. Yamada. MARIS: map recognition input system. *Pattern Recognition*, 23(8):919–933, 1990.

[35] K. Tombre and P. Vaxivière. Structure, syntax and semantics in technical document recognition. In *First Int. Conf. on Document Analysis and Recognition (Saint Malo)*, pages 61–69, 1991.

[36] J. van Cleynenbreugel, F. Fierens, P. Suetens, and A. Oosterlinck. Knowledge-based improvement of automatic image interpretation for restricted scenes: two case studies. *Image and Vision Computing*, 6(4):238–246, Nov 1988.

[37] F.M. Wahl, K. Wong, and R. Casey. Block segmentation and text extraction in mixed text/image documents. *Computer Vision, Graphics and Image Processing*, 20:375–390, 1982.

[38] T. Xu and X. Lin. A new algorithm separating text strings from map images. In *Second Int. Conf. on Document Analysis and Recognition (Tsukuba Science City)*, pages 910–913, October 1993.

# Knowledge-based segmentation for automatic map interpretation

This chapter is based on the following paper:

J.E. den Hartog, T.K. ten Kate and J.J. Gerbrands, Knowledge-based segmentation for automatic map interpretation. Part of this chapter has been published in *Proc. of the Graphics Recognition Workshop, (Penn State), August 1995.* A full copy is currently under review.

# 5

# Knowledge-based segmentation for automatic map interpretation


## Abstract

*In this chapter, a knowledge-based framework for the top-down interpretation and segmentation of maps is presented. The interpretation is based on a priori knowledge about map objects, their mutual spatial relationships and potential segmentation problems. To reduce computational costs, a global segmentation is used when possible, but an applicable top-down segmentation strategy is chosen when errors in the global segmentation are detected. The interpretation system has been tested on utility maps and the experiments show that when a top-down resegmentation strategy is used to correct errors in the global segmentation, the recognition performance is improved significantly.*


**Keywords:** *map analysis, drawing interpretation, knowledge-based segmentation.*

## 5.1 Introduction

Our research concentrates on the automatic conversion of Dutch utility maps and in this chapter we consider the problem of obtaining a correct segmentation of the grey value map images. Over the past eight years a number of interpretation systems for line drawings have been described in literature. In most of the reported work, the segmentation process is assumed to be trivial as a binary scanning process is employed [21], or simple techniques such as thresholding are applied [19, 22]. Furthermore, many of the interpretation systems have no strategy to handle segmentation errors and therefore assume good quality binary images to be processed [1, 15, 26]. In some work, the problem of an imperfect preprocessing result is recognized. These systems approximate the graphics by using vectorization and try to solve local errors which might be due to the segmentation, such as broken lines, by using techniques based on merging collinear vectors separated by a small gap, e.g. [14, 18]. Though such an approach might be successful in more or less simple situations with carefully drawn high quality maps, the result might be less satisfactory for more complex applications. For example, due to years of intensive use the maps may be wrinkled and stained, while often parts have been erased and redrawn, leaving traces of ink on the linen material.

The most commonly used technique of thresholding introduces the problem of determining the optimal global threshold [20, 23]. If the threshold value is too low, small objects are lost, while a high threshold value results in noisy images and smearing of the objects. Often, there is no optimal threshold level which avoids smear and loss of objects. In such a case, a practical solution is to determine a sub-optimal threshold which minimizes both effects. Better results may be obtained by using local adaptive threshold techniques, e.g. [3, 8], but these techniques suffer from a variety of disadvantages. For example, usually a threshold is calculated from a window of interest. If the size is too small, these algorithms tend to emphasize noise or paper texture in regions without foreground pixels. In the case of larger windows, the algorithm becomes computationally expensive and it may have the same drawbacks as global threshold techniques.

An interesting approach to top-down interpretation and segmentation of drawing images is Anon, a schema-based system described by Joseph and Pridmore [13]. This system comprises a set of schemas which represent the entities to be recognized in the image. Each schema consists, among others, of a geometrical object description and a number of C-functions to interface to the image processing stages. The geometrical description has to be satisfied by the results of image processing before the instantiation of a schema class. This paper is one of the first to observe that there is a need for top-down segmentation strategies in drawing conversion. In Anon, all objects, without exception, are extracted by means of top-down segmentation. A typical A0-sized drawing (1 square meter or circa 1600 square inches) contains thousands of objects and if all these objects have to be extracted by means of top-down segmentation the system may become very slow and thus obstruct a practical solution.

Even if the initial global segmentation contains many errors, it still contains useful information which should not be discarded. Hence, a more realistic approach might be to develop a strategy which combines both bottom-up and top-down segmentation techniques. The computationally cheap global segmentation is used when possible, but specialized top-down segmentation techniques are utilized when needed.

In this chapter, a knowledge-based strategy is proposed which combines low-level bottom-up processing with top-down segmentation. The low-level preprocessing is discussed first. The top-down segmentation is embedded in a framework for contextual reasoning which is described in Sec. 5.3. In Sec. 5.4 the ideas underlying top-down segmentation are introduced, while the organization and representation of the knowledge is discussed in Sec. 5.5. The top-down segmentation is illustrated with two examples in Sec. 5.6. The experiments and the results are then given in Sec. 5.7. The advantages and limitations of our approach are discussed in Sec. 5.8 and in Sec. 5.9 our conclusions are given.

## 5.2 Preprocessing

Although the top-down segmentation strategy can correct errors made during bottom-up segmentation, the initial preprocessing is still important to the final interpretation result. The preprocessing to obtain a good low-level description of the image contents consists of the following steps:

1. Sharpening to enhance the blurred original image.
2. Binarization.
3. Removal of holes in the graphics.
4. Decomposition of the graphics into primitives.

### 5.2.1 Image sharpening

The use of standard linear filters such as the Laplacian operator to sharpen the blurred image (e.g. [10]) has the advantage of computational efficiency. Disadvantages of these algorithms are the tendency to amplify noise and the necessity of clipping or scaling to make the resultant pixels span the range $< 0, 255 >$. In this section, we give a brief

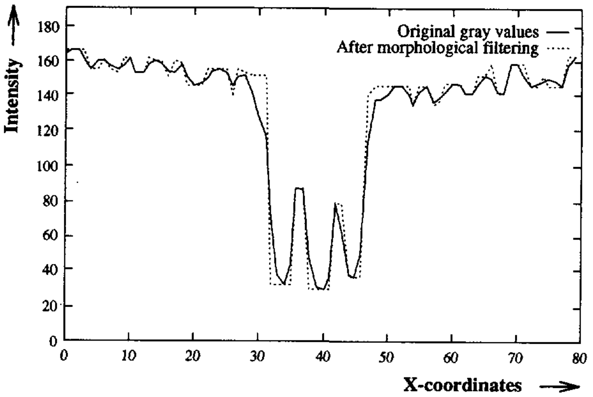



**Figure 5.1**: *The result of morphological filtering in the one-dimensional case is shown. The continuous line represents an 80 pixels wide horizontal scanline. The result after filtering is represented by the dashed line.*

description of an algorithm which uses standard grey value morphology which does not have these disadvantages.

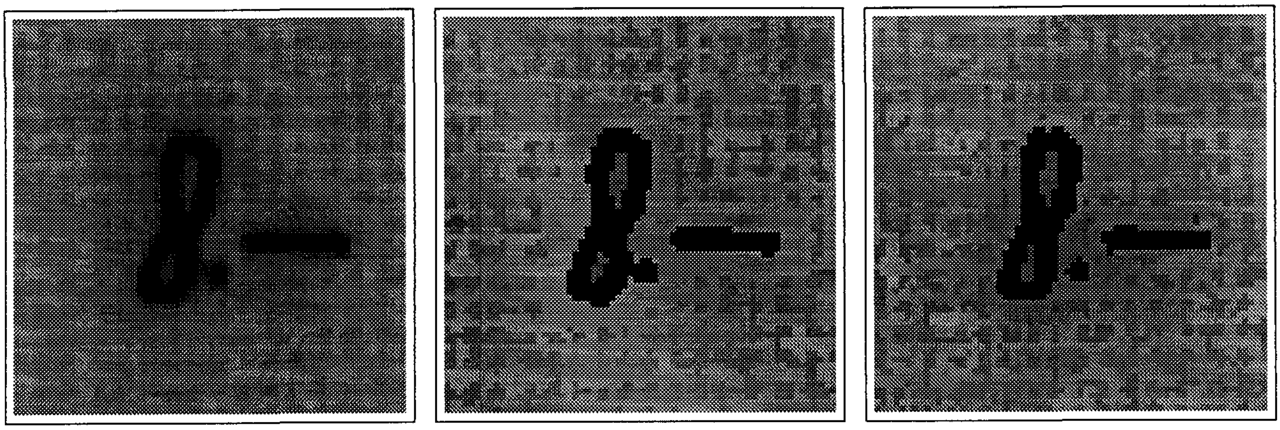
Grey value morphology is based on grey value dilations and erosions. In the case of a grey value erosion, for example, the image is scanned with a moving structuring element. The minimum value within the structuring element is calculated for each image position. This local minimum value is then stored in the pixel that corresponds with the center of the structuring element. For an extensive description of mathematical morphology the reader is referred to [24].

In the algorithm which was proposed first by Kramer [16], for each pixel the local grey value minimum and maximum within a structuring element is computed. The algorithm simply consists of replacing the grey value of each pixel by its local minimum or maximum, whichever of the two is closer in value. The result of the filter in the one-dimensional case is shown in Fig. 5.1.

Kramer showed that repeated filtering always converges, although usually many iterations are needed for complete convergence. However, after a small number of iterations, only few pixel values will change. For the utility maps only 3 iterations with a 3x3 window are sufficient for near-complete convergence. Further, it was found that filtering with a 3x3 sized window and a small number of iterations yielded better results than filtering once with a larger window. This is illustrated in Fig. 5.2. In Fig. 5.2a, a blurred dimension is shown. Applying the filter once with a 7x7 window results in a poor contrast between the main digit and the dot (Fig. 5.2b). The result with a 3x3 window after 3 iterations is shown in Fig. 5.2c.

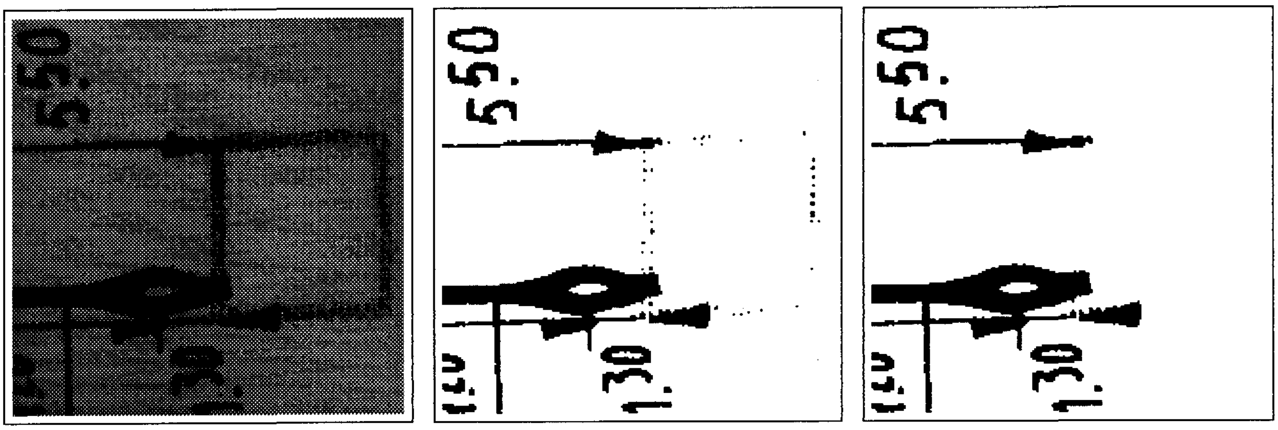
### 5.2.2 Thresholding with hysteresis

In the introduction, the advantages and disadvantages of local algorithms [3, 8] and global threshold algorithms [20, 23] were discussed. Another approach to obtain an initial segmentation is **thresholding with hysteresis** [4, 25]. In this method, a segmentation is obtained in two steps. In



(a) (b) (c)

**Figure 5.2**: *The small original $64^2$ map image in (a) is sharpened with the morphological filter with window size 7x7. The result in (b) shows that the main digit and the dot have become connected. Filtering with window size 3x3 results in (c) after three iterations.*



(a) (b) (c)

**Figure 5.3**: *The original $200^2$ map image in (a) is thresholded at a level for the optimal extraction of the graphics. The binary image in (b) shows that this may result in a noisy image. However, if the image is thresholded with hysteresis this results in the much less noisier image shown in (c).*

the first step, all pixels are classified as one of three possible categories by using a high and a low threshold. If a pixel has a grey value below the low threshold, it is classified as a definite object pixel while pixels with grey values above the high threshold are classified as definite background. The pixels with values between the high and the low threshold remain to be processed further in the second step.

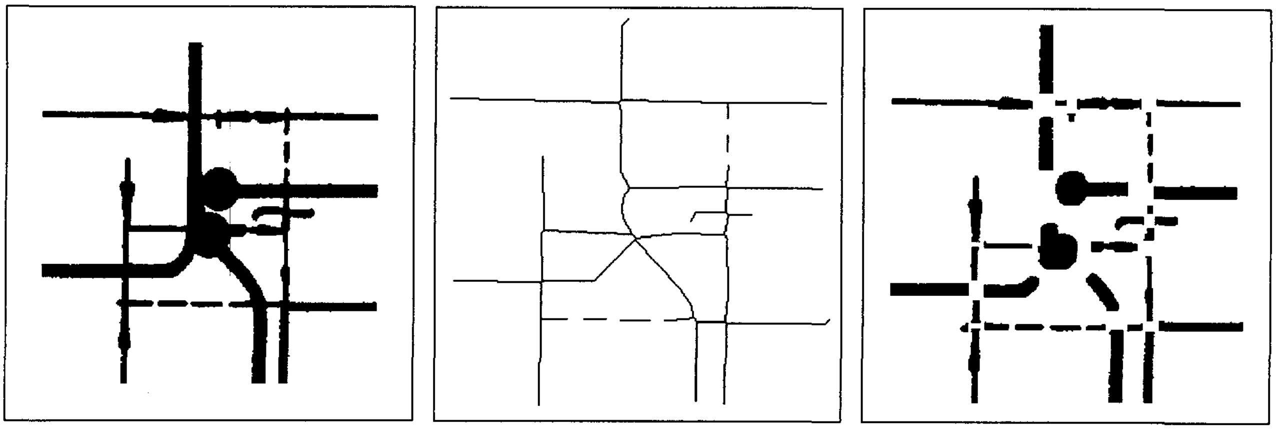
Though the remaining pixels usually are object pixels, they frequently correspond to noise or stains in the map. Only if they are connected to any of the definite object pixels are they considered to be object pixels too. Because the previous image sharpening step reduces the number of actual grey levels in the image, the thresholding step will be less sensitive to the selection of the threshold values. Fig. 5.3 shows an example of global thresholding and thresholding with hysteresis.

### 5.2.3 Hole removal

After thresholding the grey value image, the binary image is processed to remove small holes. First, based on the area, for each object a distinction is made between small objects such as characters and large objects such as the graphics. Holes are only likely to occur in small objects, e.g. digits such as a 6 or a 9. Therefore, small holes in the graphics can be regarded as noise and are removed.

### 5.2.4 Decomposition of the graphics into primitives

Skeleton vectorization is an often-encountered step in the interpretation of maps, engineering drawings and other line drawings [1, 2, 9, 11, 12, 15]. However, in spite of the attractive simplicity, rigorous vectorization may introduce unwanted inaccuracies. In Fig. 5.4a, a piece of graphics



(a) (b) (c)

**Figure 5.4**: *Vectorization of the graphics in (a) results in the binary image in (b). The reconstructed primitives are displayed in* ***exploded view*** *in (c).*

from a utility map is shown with its vectorization in Fig. 5.4b. Reducing the graphical objects to vectors discards the morphological information required to recognize the various objects. In [7] we described an algorithm which decomposes a binary image into its constructing primitives. In the remainder of this chapter, we refer to a primitive as the most basic image component that consists of a set of connected pixels without any meaning attached to it yet. Fig. 5.4c shows the resulting primitives. To be able to display individual primitives, the primitives are shown in the **exploded view**.

Describing graphics in terms of primitives has the advantage that no morphological information is lost, as opposed to a vector description. Another advantage is that all primitives can be stored in separate bitmaps, which enables the image processing of individual primitives without interference from the neighborhood in the original image. As a consequence, it is possible to calculate many useful features for individual primitives, which is not possible with a vector-based description. The primitive-based description clearly facilitates object recognition when compared to a vector description and it forms the basis for the remaining interpretation process.

## 5.3 Contextual reasoning

The strategy to combine both bottom-up and top-down image processing is implemented in a framework provided by a knowledge-based map interpretation system described in [6]. This system guides the interpretation by means of contextual reasoning.

The concept of contextual reasoning is based on alternating a top-down process to generate search actions, a bottom-up process to recognize objects and a process to verify the results. The interpretation cycle consisting of these three processes is shown as part of a larger process in Fig. 5.5. The bottom-up process receives from the top-down process a search action, i.e. a task to search for a specified object type in a restricted area. All primitives overlapping with the search area are matched with an a priori geometric specification consisting of a list of features and for each feature a range of allowed values. If the primitive corresponds with the geometrical specification, it is offered as a candidate for classification to the verification process. If no inconsistencies with former results are detected, the search results in an identification and new search actions are added to the search list.

The mechanism to generate new search actions is based on the perception that each object type is related to other object types. For example, in many applications a distance between two objects is depicted by an arrow and a dimension, a numerical representation of the depicted distance. Thus, detection of an object immediately generates expectations about other objects in its neighborhood, which are very suitable to generate new goals for the interpretation process.

For example, in the case of detection of an arrow, search tasks for related objects, such as the dimension, are generated for a small region of interest (ROI) around the arrow. Each time a new object is detected, the top-down process collects all related object types and generates new search tasks for the search list. The search task at the top of the search list is then distributed to the bottom-up process which tries to detect new objects in the assigned ROI. Because a search for an object is based on contextual evidence and the search area is restricted to a small and confined part of the image, both the number of search actions and the number of incorrect object classifications are reduced, thus rendering the interpretation both efficient and reliable.

## 5.4 Top-down segmentation

### 5.4.1 Inconsistency-based resegmentation

The concept of top-down segmentation is integrated within the contextual reasoning framework. During contextual reasoning, inconsistencies can be detected between recognized objects and prior knowledge. In

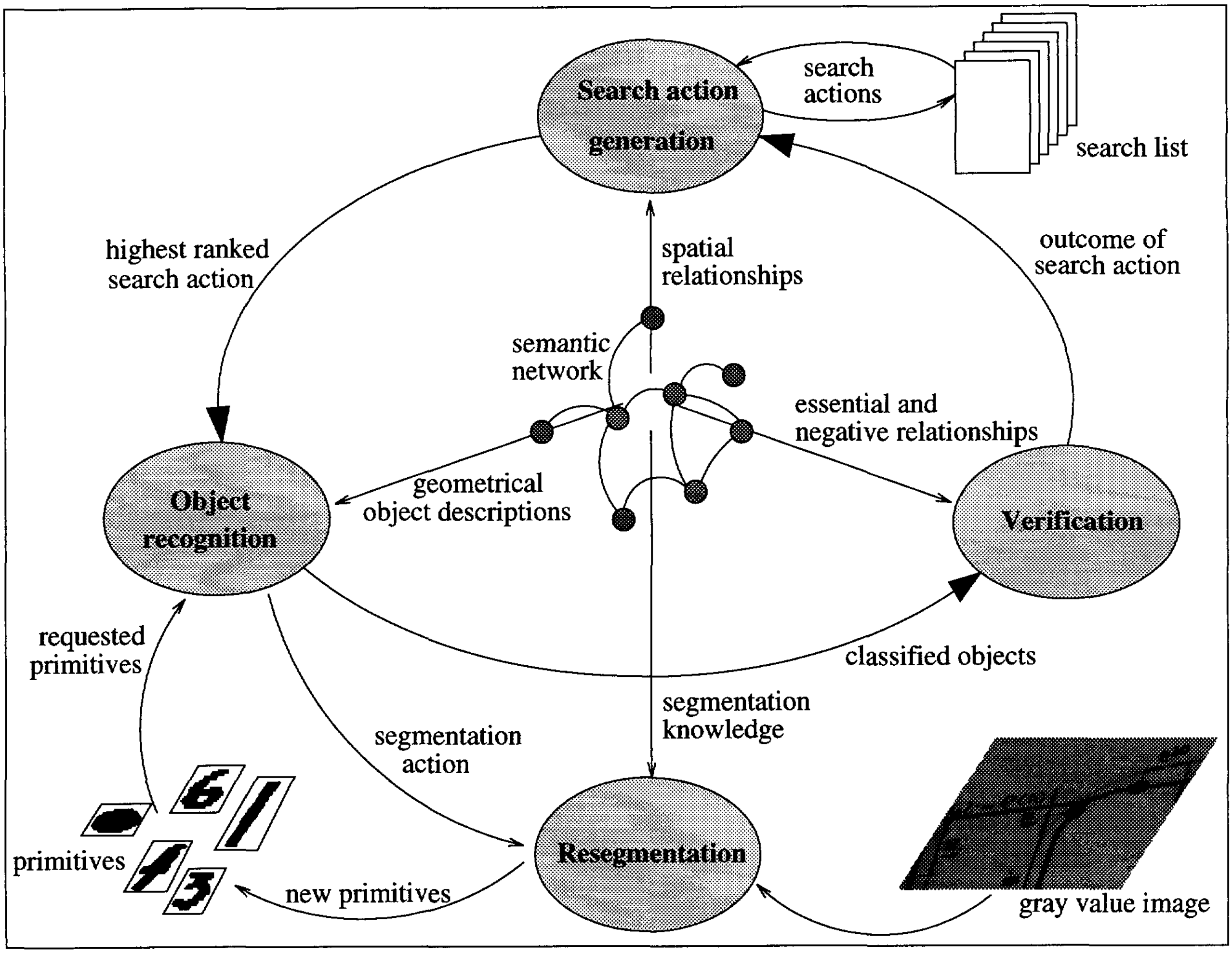



**Figure 5.5**: *The concept of contextual reasoning and resegmentation*

our application area we experienced that inconsistencies are often due to a poor initial segmentation. If knowledge about inconsistencies and potential segmentation problems is used during interpretation to improve the local segmentation, many inconsistencies may be solved automatically.

#### 5.4.1.1 Inconsistency detection

The utility maps considered in our research are 10 to 30 years old and have been used in field work. As a consequence, the maps are often wrinkled and stained and it is more than likely that, due to noise or poor segmentation, many objects are misclassified or rejected from classification. These misclassifications and rejects slow down the interpretation process considerably as the results from the automatic interpretation have to be verified manually. It is therefore important to be able to detect and solve misclassifications and rejects during the interpretation instead of afterwards.

Because the aim of the research is to develop a system with a broad application area, our research has concentrated on a method based on the detection of contextual inconsistencies between results and prior knowledge. The method is based on the perception that three types of spatial relationships between objects may be distinguished:

1. The **optional** relationship.
2. The **essential** relationship.
3. The **negative** relationship.

An optional relationship between two objects indicates that if the first object is found, it is likely that the second will be found in its vicinity. For our application, most of the relationships are optional, but a substantial part of the relationships is essential. An essential relationship implies that if the first object is found, the related object **must** be present too. An example of an essential relationship is the arrow which always has a dimension. The **negative** relationship is the opposite of the essential relationship. In this case two objects should not share a specific spatial relationship. For example, the distance between a conduit and a house should not be less than 0.5 meters.

#### 5.4.1.2 Inconsistency handling

The concept of inconsistency handling is based on knowledge about the causes of specific inconsistencies. For example, each house has a number. Thus, if a house has been found but the number cannot be detected, in general there are three possible causes for this inconsistency:

1. The classification of the house is wrong.
2. The draftsman did not write the number on the map.
3. The house number cannot be recognized due to poor segmentation.

Knowledge about these causes can help to solve the inconsistency without the assistance of the operator. To check whether the classification of the house is correct an alternative detection method is applied first. If the second technique confirms the classification of the house it is assumed that the segmentation of the house number is poor. In this case, the local segmentation could be improved by using specific algorithms with parameter settings tailored to an optimal segmentation of small objects such as digits. If resegmentation does not result in the recognition of the house number, this might be due to a violation of the drawing rules and the system should invoke operator assistance.

The interpretation cycle is shown in Fig. 5.5. Each filled ellipse represents a sub-process within the interpretation process. If, for a given search action, the outcome of object recognition is such that the verification process detects an inconsistency, a message is passed to the search action generation process. This process then searches in the knowledge base for a segmentation method to solve the inconsistency. If a method is available, it is passed to the object recognition process which sends the segmentation action to the resegmentation process. Next, the resegmentation process gathers the segmentation algorithm and the corresponding parameter settings from the semantic network and, as a next step, executes the segmentation action. After resegmentation, the resulting primitives are added to the existing list of primitives and the object recognition process then again executes the search action which led to the inconsistency. If the resegmentation was successful the inconsistency will no longer be detected. If the inconsistency is not solved, the classification causing the inconsistency is rejected and it remains to be classified by the operator afterwards.

### 5.4.2 Directly processing the grey value image

If it is known a priori that the initial segmentation cannot properly segment certain object types, it is inefficient to use the inconsistency mechanism for resegmentation. A better approach is to process the grey value image directly to find the objects. The interpretation process is therefore extended with the addition of grey value objects. When searching a grey value object, a segmentation method and a geometrical object description are passed to the object recognition process. Object recognition then passes the segmentation action to the resegmentation process and after resegmentation it tries to match the segmentation result with the object description.

## 5.5 Knowledge representation

Flexible manipulation of the knowledge is important when developing an interpretation system. For this purpose, an explicit knowledge representation language has been developed. In this section, the merits of explicit knowledge are discussed first and are followed by some relevant aspects of the language.

### 5.5.1 Explicit knowledge

Public utilities usually provide multiple services such as electricity, water, gas and cable television, and for each service a different type of map is used. Besides the multiple services, multiple networks are used for the transportation of a single service. For example, gas is transported to local distribution centers through a high pressure network, while for distribution to customers a low pressure network is used. All these types of networks are drawn on different maps. The possible variety in maps becomes wider when the drawing conventions change in time or when services are taken over from other public utilities.

Because many applications have to be considered even for a single public utility, flexibility of interpretation is one of the most important goals in our research. Although the symbols and structure of the maps are more or less similar for the various applications, it may be clear that an interpretation scheme for one type of map cannot be used directly for other maps. However, the concepts underlying the interpretation are shared by all map applications. If a priori knowledge about an application is separated from the implementation, it should be relatively easy to tailor an interpretation system to other applications.

In our system, the knowledge has been separated from the implementation by means of a knowledge specification language. All a priori knowledge about the application is read from a file at run time and converted into an internal data structure. This data structure is a hybrid structure which is referred to in this chapter as the semantic network. All knowledge can be adapted at run time, and, as a consequence, the time needed to develop an interpretation system is reduced significantly as reprogramming can be circumvented.

### 5.5.2 Basic objects and relationships

The concept of contextual reasoning is based on the assumption that all objects in a map are interrelated. Basically, the procedural knowledge used by the contextual reasoning mechanism should specify when to search and where to search for an object, while the declarative knowledge should describe how to recognize objects. The procedural knowledge is represented by the spatial relationships between objects. The declarative knowledge is represented as a feature-based geometry description for each object type.

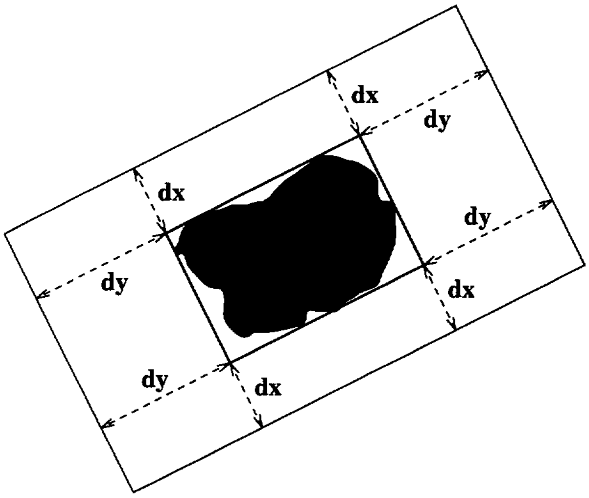



**Figure 5.6**: *The search area is based on the extension of the MAER of an object.*

#### 5.5.2.1 Object descriptions

Object recognition is based on matching each applicable primitive, i.e. each primitive overlapping with the search area, with an allowed range of attribute values in the object description. The description offers a large set of standard features such as area, length, width along the skeleton and a set of features based on the minimum area enclosing rectangle (MAER). For a detailed overview of all attributes the reader is referred to [6].

The syntax of the object description is simple as is demonstrated in the following example describing the object type *double headed arrow*:

```
DEFINE OBJECT
    objname       DoubleArrow
    id            10
    max_width     [4.0, 7.5]
    avg_width     [1.4, 7.0]
    length        [60.0, 1000.0]
    maer_ratio    [0.0, 0.4]
    maer_width    [0.0, 20.0]
    my_func       [-17.1, 22.5]
ENDDEF
```

Besides the standard set of attributes, it is easy to expand the object description by the addition of new functions. This is illustrated in the example above where a new function is added named *my_func.*

#### 5.5.2.2 Relationship descriptions

The description of the relationships between objects (Sec. 5.4.1.1) is similar to the object description. The standard attributes of the relationship description include a unique id, the object names of the two object types involved, the type of the relationship, a priority number and a specification of the search area. Analogous to the object description, a set of standard relational features is provided, such as inclusion and angle. Again, the user can extend the set of features by adding new functions.

```
DEFINE RELATION
    id            100
    from          DoubleArrow
    to            Dimensioning
    type          Essential
    priority      3
    search_area   <70,0>
    my__rel_func  [2.1, 2.3]
ENDDEF
```

The example indicates that if a double-headed arrow is found, it is likely to find a dimension in its vicinity. Based on this relationship, a search action for the dimension is generated and added to a search action list which is sorted on the priorities of each search action. In this case, the search action is assigned priority 3. The search area is based on the MAER of the previously detected object. The search area has two arguments, *dx* and *dy*, specifying the extension of the width and the length of the search area respectively. Fig. 5.6 shows an example.

### 5.5.3 Knowledge for resegmentation

The knowledge needed for resegmentation of the grey value image is represented in a way similar to the previous examples. The segmentation knowledge has to be tailored to optimally extract a single object which cannot be found by using the initial segmentation. Since the resegmentation will be carried out locally, the knowledge description has to provide an argument to specify the size and the location of the part of the image to be segmented. It should also be possible to specify the image processing functions, their arguments and the execution order of the functions.

```
DEFINE GRAY_OBJECT
    id             20
    from           DoubleArrow
    to             House
    reseg_area     <10,100>

    median_filter  5 5
    local_thresh   31 31 0.4

    max_width      [3.0, 5.0]
    avg_width      [2.6, 3.4]
    my_house_func  [1.0, 4.0]
ENDDEF
```

In the example above, the knowledge specification for the optimal segmentation **and** recognition of a grey value object is shown. The segmentation description is applicable to the situation when a double-headed arrow is found and a house segment is searched directly in line with the arrow. The part of the image to be resegmented is calculated

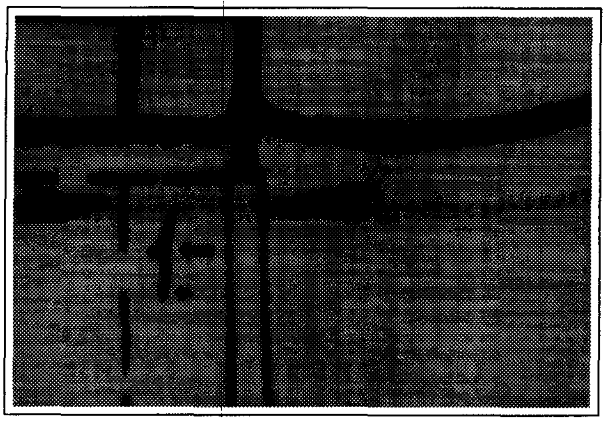


**Figure 5.7**: *An arrow with its head drawn on the front and the tail drawn on the back.*

from the MAER of the arrow and the reseg-area specification. Each image processing function that is defined in the knowledge specification, is succeeded by an argument list. In the example, two image processing techniques are applied to the grey value image. First, the image is processed with a median filter to remove the paper texture. Next, the filtered image is segmented by using a local and specialized thresholding algorithm. The binary image that results contains new primitives which are added to the list of existing primitives. If the resegmentation was successful, a search action for the house using the description in the remainder of the object specification should result in its recognition.

## 5.6 Two examples

The application for our research is introduced first and is followed by two examples to illustrate the top-down resegmentation process.

### 5.6.1 The application

The main concern of the public utility is to convert the position of the pipelines and conduits on their maps to a digital description in world coordinates acceptable by a GIS. Currently, we consider the conversion of the low-voltage electricity maps. On these maps, the relative position of the conduit is depicted by the perpendicular distance between the conduit and clearly distinctive landmarks such as the corners of the houses. On the utility maps, the distance between two points is represented by an arrow and a dimension. A dimension consists of multiple objects which can be categorized as either a dot or a digit. Since the exact location of the houses can be obtained in a digital format from the Dutch cadaster, the recognition of the conduit, houses, arrows and their dimensions should be sufficient to reconstruct the exact position of the conduit.

Unfortunately, this rather simple model has to be extended. Often, objects such as houses and roads are drawn on the back of the map, while some arrows are drawn partially on the front and partially on the back. Fig. 5.7 shows an example where the head of the arrow is drawn on the front while its tail consists of the outline of a road drawn on the back. Fig. 5.3 shows the even more difficult case where the head of the arrow is drawn on the front while its tail is also part of a house drawn on the back. The main reason for drawing objects on both sides of the paper is efficiency when updating the maps. The electricity infrastructure is not static and both the positions and contents of the conduits are often subject to change. It is, therefore, necessary to update the paper maps regularly. For the draftsman, redrawing the situation on a map is much more convenient if the most static part of the objects on the maps, i.e. roads and real estate, are drawn on the other side.

### 5.6.2 Example #1, segmentation of the dimension

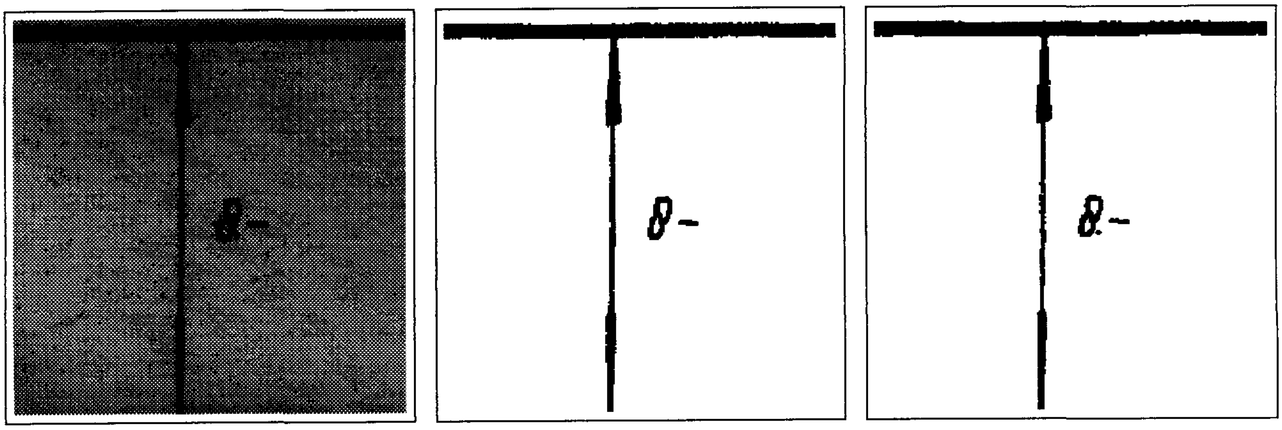
The dimension is a very important object type to recognize properly as it represents the numeric value of the distance indicated by an arrow. The position of the conduit is indicated on the map by an arrow depicting its distance to the houses. The proper segmentation of the arrow and its dimension is thus vital for the success of the interpretation. However, the global segmentation is optimized for the graphics in the image and as a consequence the segmentation of the dimensions is often less adequate.

A dimension is made up of either three or four digits and a dot, or, a digit, a dash and a dot. For the reliable recognition of the dimension all these symbols have to be segmented correctly, but due to the global segmentation, small objects may be lost or separate objects may become connected. Fig. 5.8a shows the grey value image of a conduit segment and, perpendicular to it, an arrow with its dimension. Fig. 5.8b shows that the initial segmentation results in an acceptable segmentation of the arrow while the segmentation of the dimension results in the loss of small objects.

After recognition of the conduit, a search action is generated for the double-headed arrow. When the arrow is detected, the proper recognition of the dimension will fail due to its poor segmentation. In this situation where a double-headed arrow is drawn perpendicular to the conduit, a specific drawing rule requires the dimension to be drawn approximately midway between the arrow heads on either the left-hand or the right-hand side of the arrow. Therefore, a very specific resegmentation action for the dimension can be generated for a small neighborhood. In this example, the blurred original image is sharpened only once and then thresholded at an appropriate threshold level. The result of local resegmentation is shown in Fig. 5.8c.

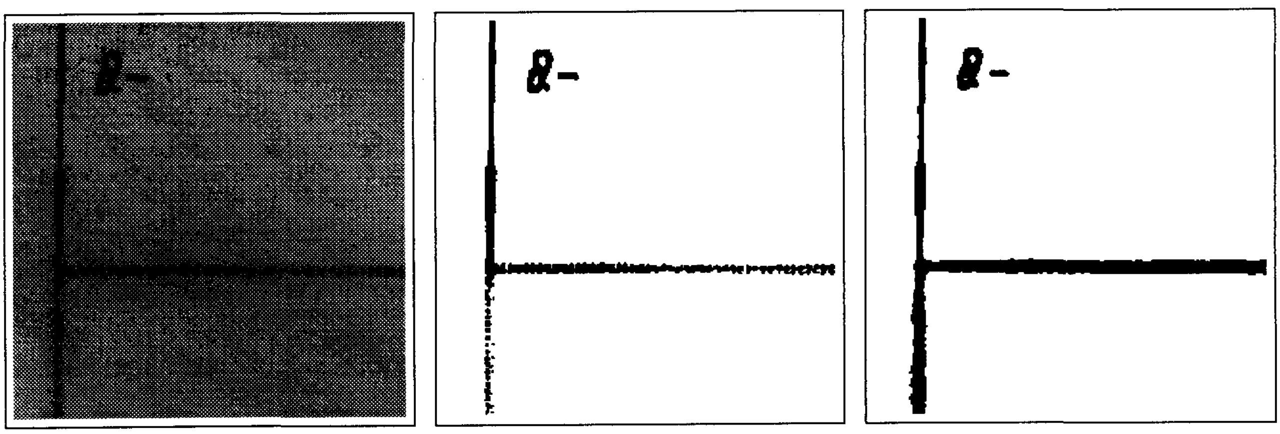
### 5.6.3 Example #2, segmentation of the houses

The second example is a more complex segmentation problem. In Fig. 5.9a, an arrow and its dimension are



(a) (b) (c)

**Figure 5.8**: *The grey value image of an arrow and its dimension is shown in (a). The global segmentation in (b) shows a poor segmentation of the dimension. A local specialized segmentation results in (c).*



(a) (b) (c)

**Figure 5.9**: *The grey value image in (a) shows an arrow, its dimension and, drawn on the back, the outline of a house. Global segmentation results in (b) while in (c) the result after a local specialized segmentation is shown.*

shown together with the fragment of the front of a house. The house fragment is drawn on the back of the map. There are two causes for the more difficult segmentation of the graphics on the back. The maps are often made of linen and there is interference between the texture of the linen and the objects drawn on the back of the map. Compared to the objects on the front, the range of possible grey values is much wider for the objects on the back. A simple and cheap global threshold operation would result in either a very poor segmentation of the houses or a very noisy image. Thresholding with hysteresis, which is used for the initial segmentation, would discard the houses entirely. The result of thresholding the image with a global threshold value is shown in Fig. 5.9b.

Similar to the first example, knowledge about the context of the houses can be used to generate top-down segmentation actions. For example, if the conduit, the double-headed arrow and its dimension are found, the region of where to expect the house is limited to a small strip directly in line with the arrow and a strip perpendicular to the arrow. Therefore, two object specific resegmentation actions can be generated for limited parts of the image. From the detected house segments new segmentation actions can be generated to detect other house segments. In this case, the resegmentation procedure consists of two relatively expensive operations. To prevent distortion by the paper texture, a standard median filter is applied. After filtering the texture, a local threshold operation [3] is applied which leads to the result shown in Fig. 5.9c.

## 5.7 Experiments and results

### 5.7.1 Data acquisition

Two original 30-year-old linen maps were available for our experiments. These A0-maps are hand-drawn to a scale of 1 to 500. Each map was scanned in 256 grey values at a density of 400 dpi. The first map was used to optimize the

| | ground truth | correct # | correct % | misclass. # | misclass. % | rejects # | rejects % |
|---|---|---|---|---|---|---|---|
| Conduit $m$ | 8.10m | 7.90m | 97.6 | 0 | 0.0 | 0.20m | 2.4 |
| Conduit # | 712 | 620 | 87.1 | 0 | 0.0 | 92 | 12.9 |
| Front arrows # | 463 | 339 | 73.2 | 0 | 0.0 | 124 | 26.8 |
| Back arrows # | 96 | 0 | 0.0 | 0 | 0.0 | 96 | 100.0 |
| Houses $m$ | 7.32m | 4.50m | 61.5 | 0.05m | 0.7 | 2.77m | 37.8 |
| Houses # | 535 | 315 | 58.9 | 0 | 0.0 | 220 | 41.1 |
| Dim. digit # | 1310 | 649 | 49.5 | 6 | 0.5 | 655 | 50.0 |
| Dim. dot # | 440 | 216 | 49.1 | 2 | 0.5 | 222 | 50.5 |

**Table 5.1**: *Recognition performance with resegmentation.*

interpretation system while the second map was used for testing and evaluation. There was no overlap between the maps.

### 5.7.2 Evaluation of the experiments

To be able to compare the results of top-down segmentation with straightforward bottom-up segmentation, it is important to develop an evaluation strategy for the segmentation. In literature, several approaches to segmentation evaluation have been proposed, such as a uniformity criterion for regions [17], visual criteria for map images [27] and a criterion based on the accuracy of measurements compared to measurements from a reference image [28]. In this chapter, however, we consider a specific application and the aim is to improve the recognition performance. To evaluate the top-down segmentation of the utility maps we therefore propose an evaluation criterion based on the recognition performance on the segmented objects. The main assumption in this approach is that an improvement of the segmentation stage will lead to an increase in the number of correctly classified objects. An important advantage of a recognition-based evaluation is the possibility to quantify the segmentation performance. Moreover, if the classification of each object in the test set is made available once as part of the ground truth, the experimental results can be evaluated automatically.

For each object class, the evaluation is limited to the following cases:

1. **correct**, the percentage of correctly classified objects with respect to the total number of objects in that specific class.

2. **reject**, the percentage of the objects which remained unclassified with respect to the total number of objects in that specific class.

3. **misclassification**, the proportion of all the incorrectly classified objects with respect to the total number of objects in that specific class.

4. **false accepts**, the proportion of objects from other classes which were accepted incorrectly with respect to the total number of accepts for this specific class.

Note that in general a false accept may be counted as a misclassification of another object class.

To calculate the above classification statistics automatically, the list of primitives is compared with the ground truth. For each vector and for each symbol in the ground truth the overlapping primitive is searched first. There are four options:

1. No overlapping primitive could be found and the number of rejects is increased by one.

2. The overlapping primitive remained unclassified and again the number of rejects is increased by one.

3. The overlapping primitive was wrongly classified and the number of misclassifications is increased by one.

4. The overlapping primitive was classified correctly and the number of correct classifications is increased by one.

| | correct | false accepts # | false accepts % |
|---|---|---|---|
| Conduit $m$ | 7.90m | 0.00m | 0.0 |
| Conduit # | 620 | 1 | 0.2 |
| Front arrows # | 339 | 0 | 0.0 |
| Back arrows # | 0 | 0 | 0.0 |
| Houses $m$ | 4.50m | 0.30m | 6.3 |
| Houses # | 315 | 27 | 7.9 |
| Dim. digit # | 649 | 13 | 2.0 |
| Dim. dot # | 216 | 12 | 5.3 |

**Table 5.2**: *The false accept rate with resegmentation.*

| | correct | false accepts # | false accepts % |
|---|---|---|---|
| Houses $m$ | 4.50m | 0.02m | 0.5 |
| Houses # | 315 | 2 | 0.6 |

**Table 5.3**: *The false accepts for houses when corrected for the border effect.*

| | ground truth | correct # | correct % | misclass. # | misclass. % | rejects # | rejects % |
|---|---|---|---|---|---|---|---|
| Conduit *m* | 8.10m | 7.90m | 97.6 | 0 | 0.0 | 0.20m | 2.4 |
| Conduit # | 712 | 620 | 87.1 | 0 | 0.0 | 92 | 12.9 |
| Front arrows # | 463 | 339 | 73.2 | 0 | 0.0 | 124 | 26.8 |
| Back arrows # | 96 | 0 | 0.0 | 0 | 0.0 | 96 | 100.0 |
| Houses *m* | 7.32m | 0 | 0.0 | 0 | 0.0 | 7.32m | 100.0 |
| Houses # | 535 | 0 | 0.0 | 0 | 0.0 | 535 | 100.0 |
| Dim. digit # | 1310 | 392 | 29.9 | 6 | 0.5 | 912 | 69.6 |
| Dim. dot # | 440 | 125 | 28.4 | 1 | 0.2 | 314 | 71.4 |

**Table 5.4**: *Recognition performance* ***without*** *resegmentation.*

Following this step, the primitives which were labeled as objects of a specific class, but could not be matched with the ground truth, can then be classified as false accepts.

### 5.7.3 Results

The interpretation is based on the classification of the primitives which are extracted either from the binary image (Sec. 5.2.4) or during top-down resegmentation. In general, arrows, as well as the dots and digits are represented by a single primitive. However, houses and conduits are usually represented by several primitives, and as a consequence, it may happen that these objects are only partially recognized. A partial recognition can be evaluated in two ways: by the *number* of recognized primitives with respect to the *total number* of primitives which make up the object, or by the *length* of the recognized primitives with respect to the *total length* of the primitives composing the object. The former method of evaluation is related to the number of interactions to correct the result, whereas the latter method is related to the conversion throughput. Both evaluation methods are relevant measures of the performance of the system, depending on whether the interest is focused on the number of corrective actions or on the length to be reclassified. From a practical point of view, it seems sensible to provide both results for the object types house and conduit. In the result tables the performance in meters is denoted with an **m** while the performance in the number of primitives is marked with a #.

| | correct | false accepts # | false accepts % |
|---|---|---|---|
| Conduit *m* | 7.90m | 0.00m | 0.0 |
| Conduit # | 620 | 1 | 0.2 |
| Front arrows # | 339 | 0 | 0.0 |
| Back arrows # | 0 | 0 | 0.0 |
| Houses *m* | 0.00m | 0.00m | 0.0 |
| Houses # | 315 | 0 | 0.0 |
| Dim. digit # | 392 | 5 | 1.3 |
| Dim. dot # | 125 | 8 | 6.0 |

**Table 5.5**: *The false accept rates* ***without*** *resegmentation.*

In Table 5.1 and Table 5.2 the recognition performance with contextual reasoning and top-down segmentation is presented. Besides evaluation of the overall recognition performance, it is equally important to evaluate the effect of the top-down resegmentation. Therefore, the recognition performance has to be compared with an interpretation strategy without top-down resegmentation. Table 5.4 and Table 5.5 show the recognition performance on the same data that was used in the first experiment, however, without using knowledge about top-down resegmentation. In this case, it was not possible to detect any of the objects drawn on the back of the map or to solve inconsistencies automatically.

Since the conduit is the thickest entity in the image and is always drawn on the front of the map, it is the easiest object to segment and to recognize and therefore no top-down segmentation strategy was needed. As can be expected, the recognition performance did not change without resegmentation and remained at 97% and 87% respectively without any misclassifications and almost no false accepts. The difference in performance on the conduit measured in length and number of primitives can be explained by the high rejection rate of very short conduit segments.

Most of the arrows are drawn on the front but about 20% are drawn partially on the front and partially on the back. Because of their different appearance, we decided to distinguish between "front arrows" and "back arrows" in the result tables. None of the back arrows were detected, and their detection is still one of the major problems which remains to be solved. Their proper recognition is obstructed by the difficulty in recognizing the small-sized arrow heads reliably. These arrows often lack a clear context from which they can be detected. If the arrow heads cannot be recognized, it is very difficult to generate an accurate and successful segmentation action for the tail.

Even when a distinction is made between the front arrows and the back arrows, still 27% of the front arrows remain unclassified. Several causes can be identified for these rejects. The main cause is the contextual reasoning mechanism; an arrow is only searched if the related object was detected earlier. Thus, if the related object was not found, the arrow cannot be recognized either. Another cause of rejects is the decomposition process. Arrows are often intersected by lines and, as a consequence, such an arrow is decomposed into multiple primitives, which may obstruct its recognition.

In the maps used for the experiment, all houses are drawn on the back and can therefore only be extracted with resegmentation. Approximately 60% of the houses are detected correctly and only a very small number are misclassified, but the rate of false accepts is considerable (6.3% and 7.9% resp.). Approximately 90% of these false accepts can be explained by the existence of a border effect. The outline of the map-area is drawn on the back of each map. Unfortunately, this border is drawn with the same pen that is used to draw the houses. In the rare cases where an arrow is just within the map area and very close to the border, the top-down resegmentation for the house results in segmentation of the border. The rest of the border is found too from these initial border segments. Table 5.3 shows that if the false accept rate for the houses is corrected for this border effect, the number of false accepts drops to 0.5%. The reject rate of approximately 40% can be explained entirely by the contextual reasoning mechanism. Since the interpretation fails to classify a significant number of arrows, no segmentation actions were generated to detect the related houses. Furthermore, about 15% of the houses on the map are not related to any other objects and cannot be found with the contextual reasoning approach.

About half of the dimension digits and dots are correctly classified while very few are misclassified. Again, the rather high rejection rate can be explained almost entirely by the contextual reasoning mechanism. The dimensions are only searched if the corresponding arrow is found, but all of the arrows drawn on the back and 27% of the front arrows remain unclassified. The false accept rate of the dimension dots is caused mainly by transposition of the dot and dash in a dimension. If a dimension represents a whole number, it consists of a main digit, a dot and a dash. Sometimes these dashes are very small and easily mistaken for a dot. The misclassified digits in Table 5.1 are all classified as a dot, thus causing 50% of the falsely accepted dots. The importance of top-down resegmentation for the dimensions is clearly illustrated by Table 5.4. If only the initial segmentation is available, the recognition performance drops to 29%.

The computational costs are acceptable. The processing of an A0-sized map, scanned in grey value at 400 dpi (16400×14000), on a Sun Sparc 20, including preprocessing, interpretation and resegmentation requires approximately 45 minutes.

## 5.8 Discussion

The proposed generic framework allows for the representation of knowledge for the detection and handling of segmentation problems. Multiple specialized segmentation algorithms can be used during interpretation where each algorithm is applied only when necessary. Since limitations of the initial segmentation can be corrected during the interpretation, it is possible to obtain a significant increase in the final recognition performance. The resegmentation framework, which is driven by generic events such as inconsistencies and object detections, should be applicable to other application areas. Its use for the recognition of roads in aerial images has been demonstrated in a case study described in [5] (see also Chapter 6).

The explicit representation format allows the design of a resegmentation strategy at run time. This concept increases the system's flexibility and yields a significant reduction in the time and effort required to adapt the knowledge to a specific application.

The current concept has some limitations. The parameter specification of the image processing functions in the knowledge file can only handle static predefined parameters. The flexibility could be improved if during the interpretation the parameter values can be adapted to local variations in the map.

In the current concept of top-down segmentation, the generation of resegmentation actions strongly depends on contextual evidence. If the context for these actions becomes more complex, it is no longer possible to describe the context in the current knowledge base. This limitation is clearly demonstrated by the inability to detect any of the "back arrows".

## 5.9 Conclusions

A new framework has been presented which guides the interpretation and segmentation by using a priori knowledge. Because the framework allows the combination of multiple segmentation algorithms, each specialized in the segmentation of a specific object, the local segmentation of objects can be improved when needed. In the experiments, a significant increase in the recognition performance was obtained by using top-down segmentation.

The results indicate that a fully automatic system is not yet feasible; however, the developed techniques can be very useful to assist the operator in a semi-automatic environment. During semi-automatic conversion, the operator selects parts of the image which appear to be suitable for automatic conversion. The results of the automatic interpretation are displayed and can then be accepted, rejected or manually adjusted by the operator. In the latter case, the system can assist again by guiding the operator to inconsistent or rejected parts of the image. If the operator solves an inconsistent situation or reject, the automatic interpretation may continue again.

From the experiments, we conclude that the current model for knowledge representation is too limited to handle very complex situations. In the model, all procedural knowledge is represented by spatial relationships. As the model becomes more and more complex, and the number of spatial relationships and potential inconsistencies increases, the need to further structure the knowledge base will arise. Our future research, therefore, has to concentrate on the design of a more sophisticated knowledge representation model.

# Bibliography


[1] D. Antoine. CIPLAN, a model-based system with original features for understanding French plats. In *First Int. Conf. on Document Analysis and Recognition (Saint Malo)*, volume 2, pages 647–655, October 1991.

[2] D. Benjamin, P. Forgues, E. Gulko, J.B. Massicotte, and C. Meubus. The use of high-level knowledge for enhanced entry of engineering drawings. In *9th IAPR Int. Conf. on Pattern Recognition (Rome)*, volume 1, pages 119–124, November 1988.

[3] J. Bernsen. Dynamic thresholding of grey-level images. In *8th IAPR Int. Conf. on Pattern Recognition (Paris)*, pages 1251–1255, 1986.

[4] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[5] M.E. de Gunst and J.E. den Hartog. Knowledge-based updating of maps by interpretation of aerial images. In *12th IAPR Int. Conf. on Pattern Recognition (Jerusalem)*, volume 1, pages 811–814, October 1994.

[6] J.E. den Hartog, T.K. ten Kate, and J.J. Gerbrands. Knowledge-based interpretation of public utility maps. accepted for publication in *Computer Vision and Image Understanding*, 1995.

[7] J.E. den Hartog, T.K. ten Kate, J.J. Gerbrands, and G. van Antwerpen. An alternative to vectorization: decomposition of graphics into primitives. In *Third Annual Symposium on Document Analysis and Information Retrieval (Las Vegas)*, pages 263–274, April 1994.

[8] L. Eikvil, T. Taxt, and K. Moen. A fast adaptive method for binarization of document images. In *First Int. Conf. on Document Analysis and Recognition (Saint Malo)*, pages 435–443, 1991.

[9] C.S. Fahn, J-F. Wang, and J-Y. Lee. A topology-based component extractor for understanding electronic circuit diagrams. *Computer Vision, Graphics and Image Processing*, 44:119–138, 1988.

[10] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Addision Wesley, Reading, Massachusetts, 1992. ISBN 0–201–50803–6.

[11] K.J. Goodson and P.H. Lewis. A knowledge-based line recognition system. *Pattern Recognition Letters*, 11(4):295–304, 1990.

[12] M. Ilg. Knowledge-based understanding of road maps and other line images. In *10th IAPR Int. Conf. on Pattern Recognition (Atlantic City)*, volume 1, June 1990.

[13] S.H. Joseph and T.P. Pridmore. Knowledge-directed interpretation of mechanical engineering drawings. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(9):928–940, September 1992.

[14] T. Kaneko. Line structure extraction from line-drawing images. *Pattern Recognition*, 25(9):963–973, 1992.

[15] R. Kasturi, S.T. Bow, W. El-Masri, J.R. Gattiker, and U.B. Mokate. A system for interpretation of line drawings. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(10):978–992, 1990.

[16] H.P. Kramer and J.B. Bruckner. Iterations of a non-linear transformation for enhancement of digital images. *Pattern Recognition*, 7:53–58, 1975.

[17] M.D. Levine and A. Nazif. Dynamic measurement of computer generated image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 7(2), 1985.

[18] T. Nagao, T. Agui, and M. Nakajima. An automatic road vector extraction method from maps. In *9th IAPR Int. Conf. on Pattern Recognition (Rome)*, pages 585–587, Nov 1988.

[19] V. Nagasamy and N.A. Langrana. Engineering drawing processing and vectorization system. *Computer Vision, Graphics and Image Processing*, 49:379–397, 1990.

[20] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Systems, Man and Cybernetics*, SMC-9(1):62–66, January 1979.

[21] B. Pasternak and B. Neumann. ADIK: an adaptable drawing interpretation kernel. In *Avignon 1993, 13th Int. Conf. on AI, Expert Systems and Natural Language (Avignon)*, volume 1, pages 531–540, May 1993.

[22] P. Puliti and G. Tascini. Interpreting technical drawings. *The Computer Journal*, 33(5), 1990.

[23] T.W. Ridler and S. Calvard. Picture thresholding using an iterative selection method. *IEEE Trans. Systems, Man and Cybernetics*, 8(8):630–632, August 1978.

[24] J. Serra. *Image analysis and mathematical morphology.* Academic Press, London, 1982. ISBN 0-12-637240-3.

[25] M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis and machine vision.* Chapman Hall, Cambridge, Great Britain, 1st edition, 1993. ISBN 0-412-45570-6.

[26] S. Suzuki and T. Yamada. MARIS: map recognition input system. *Pattern Recognition*, 23(8):919–933, 1990.

[27] O.D. Trier and T. Taxt. Evaluation of binarization methods for utility map images. In *First IEEE Int. Conf. on Image Processing (Austin)*, November 1994.

[28] Y.J. Zhang and J.J. Gerbrands. Objective and quantative segmentation evaluation and comparison. *Signal Processing*, 39:43–54, September 1994.

1234567

# *Knowledge-based updating of maps by interpretation of aerial images*

This chapter is based on the following publication:

M.E. de Gunst and J.E. den Hartog, Knowledge-based updating of maps by interpretation of aerial images. In *Proc. of the 12th IAPR Int. Conf on Pattern Recognition (Jerusalem)*, volume 1, pages 811-814, October 1994.

# 6

# Knowledge-based updating of maps by interpretation of aerial images

## 6.1 Introduction

In the preceeding chapters, the research concentrated on the development of techniques and methods for automatic map interpretation. Because the main concept of contextual reasoning is based on the assumption that the objects in the image share clear spatial relationships, the proposed framework should apply to a broader range of problems. The spatial relationships are used to generate hypotheses for the interpretation process, to verify its results, and to control the low-level segmentation. It therefore seems logical to explore the potential of contextual reasoning for other areas. In this chapter, we concentrate on the interpretation of aerial images. Aerial images of urban areas are usually highly structured as they contain carefully planned and constructed man-made objects. A road network, for example, is built according to strict construction rules describing aspects such as the maximum curvature of a road, the optimal angle between a speedway and an exit, etc. However, the interpretation of aerial images is far from trivial for several reasons. An aerial image may contain many objects, densely packed in a small area, while only a small number of these objects is relevant. Furthermore, a complex object, such as a road network, is composed of many different and specialized elements, e.g. roads, lanes, bridges and fly-overs. Each of these objects may require a specialized segmentation and recognition strategy. Finally, a single object type, e.g. a house, may appear in the aerial image in a wide range of manifestations, in a different context and on a different scale. Thus, a knowledge-based approach to the interpretation of these complex images using some kind of contextual reasoning strategy seems appropriate, if not necessary.

The development of techniques for interpretation of aerial images is driven by a clear demand from the suppliers of geographical information. In the management of geographical information, a shift has taken place from the production of analog maps to storage of digital information in a Geographic Information System (GIS). A GIS provides a flexible environment to edit and manipulate the information, while new information can be added easily. Because a GIS allows for the easy combination of multiple information sources, it is possible to discover the relationships between geographic entities and to exploit the data better and more efficiently than ever possible with analog maps. To be effective for most applications, however, a GIS requires accurate and up-to-date data. An example of such an application is monitoring of the enviroment. Mankind is causing rapid ecological and environmental changes, and the major part of the data about these changes is acquired from aerial photographs and satellite images. The photogrammetric processing of the data, however, is the major bottleneck in the supply of geographical information. In the current process, each relevant object is outlined manually in the photographs, which is a labor intensive and time consuming task. To be able to support fast decision-making on geographical issues, there is an urgent need for automatic methods and techniques to collect information from these images efficiently and reliably.

## 6.2 Related work

Automatic interpretation has been a research topic for more than two decades now. One of the first influential papers on this subject was published by Bajscy and Tavakoli in 1976 [1]. In their strict bottom-up approach, potential road pixels are found by applying a simple threshold based on the expected intensity of road pixels. The result is then scanned for pixels with the proper intensity profile using templates. In the next phase, probable road points are linked based on constraints about the maximum distance between such two points. Another line of early research concentrated on road tracking. In road tracking, the interpretation starts from parts of the road, also called seeds, which are already known. From these seeds, the related parts of the road are found using some general road model about the shape of the road [5] or the intensity [11]. A more recent approach to road tracking, using two independent methods based on road edges and the intensity profile perpendicular to the road, has been reported by McKeown and Denlinger [10]. The main disadvantage of road tracking is that the interpretation requires the road seeds to start from. Furthermore, if the road tracking algorithm fails and "wanders off", it is not possible without additional knowledge, to detect and recover from this situation.

To be able to detect and correct the potential flaws of these specific algorithms, the research in the eighties concentrated on knowledge-based systems. Several knowledge-based approaches have been proposed for aerial image interpretation. In MISSEE, which was described by Glicksman [6], the interpretation is based on a semantic network consisting of schemata and the binary links between them. For a detailed description of the schema concept, the reader is referred to the work reported by Hanson and Riseman [7]. Instantiation of a schema, i.e. recognition of an object represented by the schema, leads to the evaluation of other related schemata. A limitation of this approach is that MISSEE strictly separates image processing and high-level reasoning as the interpretation expects a segmented image which is used throughout the entire process. A similar approach is SIGMA, which has been reported by Matsuyama and Hwang [9]. SIGMA is a frame-based system where each frame represents knowledge about an object class. Spatial relationships with other objects are represented by means of procedural attachment. When a frame is instantiated, the attached procedure is used to generate a hypothesis about a related object. As opposed to MISSEE, SIGMA does not depend on an initial segmentation as hypotheses are handled by the top level of the interpretation which combines and delegates them through an intermediate level to a low-level vision module. A disadvantage all these systems and methods have in common, is their lack of reliability and accuracy. This is partly caused by the fact that they all, at some stage during processing, depend on an initial segmentation. In the next section, we will concentrate on another interpretation concept which does not have this disadvantage.

## 6.3 The concept for updating

As was discussed above, all methods and systems reported so-far lack sufficient reliability and accuracy to make them operational in a production system. In this section, we propose an approach to increase the reliability of aerial image interpretation using existing information from a GIS. Only a few authors (e.g. [8, 12, 13]) use existing maps as a knowledge source for aerial image interpretation. This is the more surprising if we realize that outdated maps are often present and the locations of the objects in these maps create a logical framework to search for new objects. Thus, when using existing maps, new objects may be found more reliably and more accurately. In a recent paper, Yu and Berthod [13] described an approach which uses information from a GIS to improve the segmentation of urban areas. Their technique is based on pixel labeling using Markov random fields and the a priori information from the GIS is used to construct a binary mask image to initialize the pixel labeling process. A different approach is described by van Cleynenbreugel et al. [12]. In their paper, information from a GIS about the type of terrain under study, is used to classify potential road segments. For example, in a case-study concerning a mountainous area, only those road candidates are accepted which more or less follow the contour lines of a digital terrain model provided by the GIS.

In this chapter, we explore techniques for automatic updating of map information by the interpretation of aerial images, and we therefore assume that the changes in the image have a very specific relationship to the outdated maps. In contrast with van Cleynenbreugel et al. [12], we use the information from the outdated maps to be able to skip the potentially unreliable bottom-up segmentation. Furthermore, compared to the approach by Yu and Berthod, we define higher-level spatial relationships than the local relationships between single pixels. The a priori map information is employed by a knowledge-based interpretation strategy to generate specific top-down segmentation actions. The control of this interpretation strategy is presented hereunder.

## 6.4 Control strategy

The classic hypothesize-and-test paradigm has been adopted as the control strategy for this application. In the first step, the top-down approach is based on potentially outdated map information, stored in a GIS, to generate hypotheses to detect what has changed in the image. A hypothesis is verified by a top-down search action. Each action consists of a restricted region of interest in the grey value domain, an image processing technique and the corresponding parameter values which are optimized for segmentation of the specific requested type. The results of the top-down segmentation are matched with an a priori object model which consists of a collection of geometrical properties for each object type. If the match succeeds, it is assumed that the object did not change. If the match fails, however, the object is assumed to have changed. An example of such a change is a lane of a highway which has been extended by the addition of an exit. Following its detection, the possible change has to be verified by checking related objects. In this case, the change of the lane has to be verified by detection of the exit. Detection

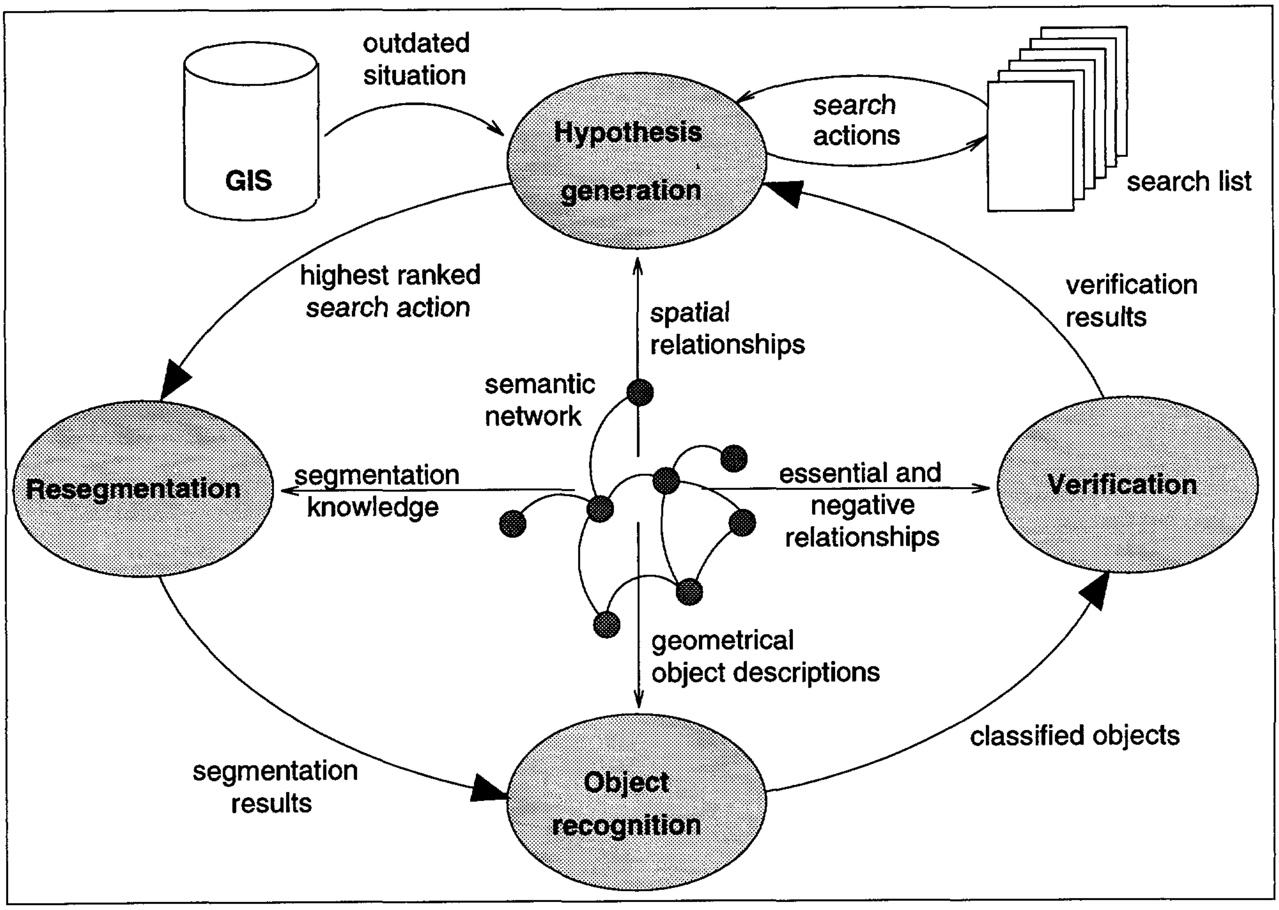



**Figure 6.1**: *The concept of contextual reasoning is depicted as a cycle of four sub-processes.*

of the exit, in its turn, will lead to new hypotheses about the presence of other related objects.

The generation of hypotheses is based on the concept of contextual reasoning which has been described in [3, 4]. Contextual reasoning consists of an interpretation cycle of four subsequent processes which all use a priori knowledge. Figure 6.1 shows the information flow between the processes. The core of the interpretation process is the knowledge base as it provides all subprocesses with vital information. We have chosen the semantic network as knowledge representation formalism because it reflects the graph-like nature of the knowledge which mostly consists of objects and their spatial relationships. For more information on this representation format, the reader is referred to [3, 4]. In the model shown in Fig. 6.1, the interpretation starts with hypothesis generation to detect differences between the image and the outdated situation. The segmentation process then receives specific search actions to segment the objects present in the outdated situation. Following segmentation, the results are matched with the object models and verified with the information retrieved from the GIS to detect possible changes. After change detection, new objects are searched in the image starting from the detected changes. Each time a new object or an object which has changed is found, new hypotheses about the occurrence of related objects are generated.

The generation of search actions, based on a priori known spatial relationships, is a powerful mechanism to control the interpretation process. By generating the proper goals, it is possible to combine a variety of image processing techniques in an effective way. Further control is provided by the **essential** spatial relationship. An essential relationship implies that if an object is found, the related object **must** be present too. For example, an exit of the highway should always end in some type of node. The essential relationship provides a simple and generically applicable mechanism to detect inconsistencies in the interpretation. Each time an object is found which shares this relationship with another object type, the related object type is searched first. In the case of an inconsistent outcome, i.e. the related object could not be detected, there are several options. For example, the classification which led to the inconsistency could be rejected. However, if there is a priori knowledge about the cause of the inconsistency, a top-down search action could be generated to obtain new results and solve the inconsistency. Finally, if this approach fails, the help of the operator could be invoked.

## 6.5 Case-study

A prototype system has been implemented to update a road map of the Dutch highways from a large-scale scanned aerial photograph with a ground resolution of 1.60 meters


**Figure 6.2**: *The aerial photograph which represents the current situation is presented in (a). In (b), the digital map of the old situation is projected on the photograph in white.*

per pixel. The road network can be regarded as an abstract graph-like structure consisting of arcs and nodes. Examples of nodes are crossings and fly-overs. Arcs are road segments such as lanes and exits, starting and ending in either different nodes or the image border. In this chapter we will refer to arcs and nodes as general types representing arbitrary types of road segments and their junctions.

In this case-study, a rather simple scene and a simplified road model are used. The two lane highway, which is shown in shown in Fig. 6.2a, is assumed to have been extended by the addition of several on-ramps and exits. The available digital road map, which represents the old situation, is shown in Fig. 6.2b. In this figure, the a priori known roads are depicted in white. To update the map, two successive tasks are distinguished. First, all changes have to be detected, followed by a top-down search starting from the changed parts for all related roads of the network.

### 6.5.1 Change detection

Basically, the roads are regarded as arcs in the graph composed by the road network. In the first part of the interpretation, we are primarily interested in whether a part of any arc changed into a node, which might be caused by the construction of a new exit or an on-ramp. When considering their representation in the grey value domain, a road, i.e. an arc, can be distinguished by its characteristic grey value profile. A node, however, may be detected by the lack of such a profile. Figure 6.3 shows two example profiles. Analogous to the work reported by McKeown [10], we assume in this case-study that an arc can be defined completely by such a set of profiles. To derive the profile set, the image is first resampled perpendicularly to the axis of the roads present in the GIS. The profile set can then be extracted easily as each column of pixels in the resampled image represents a single profile.

For successful detection of changes, it is sufficient to match the profile set taken from the resampled image with a simple profile model derived from the GIS. An example of such a profile model is shown in Fig. 6.4. The criterion for the presence of an arc is based on the cross-correlation between the model and the profile set from the resampled image which should exceed a predefined threshold value. If the cross-correlation does not exceed this threshold for a significant part of the arc, the presence of a node is assumed. In the example given in Fig. 6.5, on each lane two parts are detected which have possibly changed into nodes.

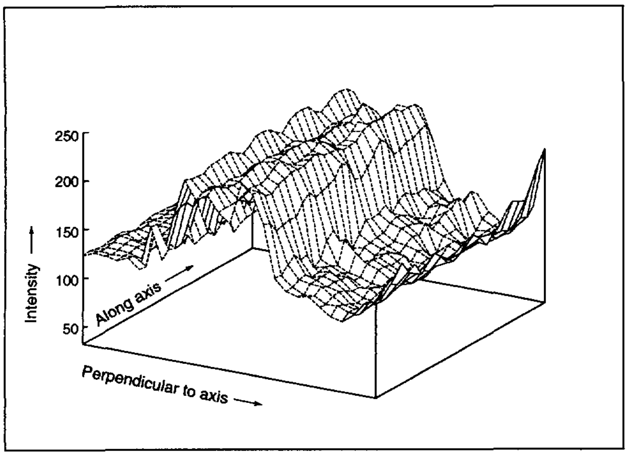
### 6.5.2 Further interpretation

The contextual reasoning mechanism is based on the spatial relationships between the parts which compose the road network. In this case-study we concentrate on two low-level basic spatial relationships between arcs and nodes:

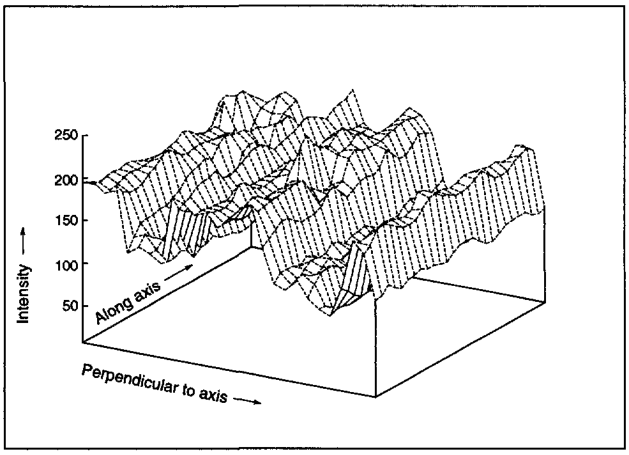
1. Nodes are **always** connected to three or more arcs.

2. Arcs start and end in either two different nodes or a node and the image border.

The first essential spatial relationship defines that if a new node is found which is not yet connected to three or more arcs, at least one new arc has to be found. The arc is




(a)




(b)

**Figure 6.3**: *An example of a profile set is shown in (a). The profiles are taken from a part of the aerial image which has been resampled perpendicular to the axis of the road. The grey value of the road is significantly higher than its background, while the profile of the node, which is presented in (b), clearly lacks such a prominent feature.*

searched in a rectangular area starting from the center of the node and oriented in a direction with high probability of finding a new arc. For example, it is known a priori that the angle between an exit and a highway never exceeds a certain value. A new node which is detected on a highway arc is most likely due to the construction of an exit. This immediately generates expectations as to where and in which direction to search for the exit. However, if no arc is detected within the initial search area, the direction is gradually adapted. For each direction, the search area is resampled perpendicular to its axis followed by the extraction of a profile set. The profile set is then matched with the profile model of the expected arc provided by the semantic network. In the example, three of the nodes are confirmed by the presence of arcs. This result is shown in Fig. 6.6a. The fourth node, however, is located at the image border and its presence cannot be confirmed by the detection of an arc, and as a consequence, its detection is rejected.

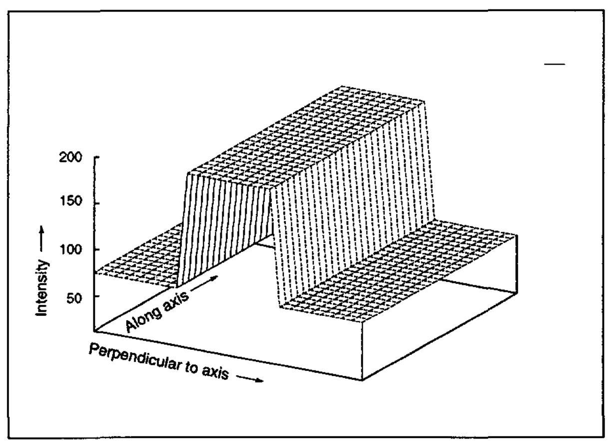



**Figure 6.4**: *An example of a profile model derived from the GIS.*

The second spatial relationship defines that if the node at the end of a new part of an arc is not yet detected, the arc is extended until either the node is found or the image border is reached. The rest of the arc is found by defining a rectangular search area that elongates the arc. The search area is again resampled perpendicular to the axis and the result is matched with the profile model of the arc. This process is repeated until a node is found. The adaptations to the direction of the search area and its length are based upon the result of the segmentation. For more details, the reader is referred to [2]. In the case-study, contextual reasoning results in prolongation of the new arcs until the image border is reached. Figure 6.6b shows the final result.

**Figure 6.5**: *Possible new nodes, which are shown in black, are identified on the arcs during change detection.*

(a) (b)

**Figure 6.6**: *The new arcs, shown in white in (a), which are related to the new nodes, are identified during contextual reasoning. The final result of the interpretation is shown in (b).*

## 6.6 Discussion and conclusions

In this chapter, we have presented a concept for the interpretation of aerial images as a potential solution to the automatic updating of maps. In contrast with previous work, a priori information from an outdated digital map is used to obtain initial results for the interpretation process. Thus, it is possible to circumvent potentially unreliable bottom-up processing and to generate accurate top-down search actions to detect changes in the infrastructure. After the initial change detection, the contextual reasoning mechanism continues the interpretation. Based on the results and a priori knowledge about applicable image processing techniques and the spatial relationships between objects, the new parts of the road are searched only in restricted parts of the image with appropriate segmentation algorithms. As a consequence, the use of computationally expensive algorithms can be limited to small parts of the image.

In the case-study, the results have been shown for a rather simple scene in the aerial photograph. For such a straightforward situation, the use of a road model composed of arcs and nodes is sufficient. However, knowledge about the types of the various parts of the road network and their mutual spatial relationships is necessary to handle more complex situations and to tailor the image processing techniques accurately to the specific object types. The current knowledge specification therefore has to be extended with new objects and spatial relationships as well as object-specific image processing knowledge. Our future research therefore has to concentrate on the development of a more elaborate model to describe complex situations and to recognize other important object types such as fly-overs, speed-ways, etc. To improve the accuracy of the top-down segmentation, this model should include knowledge about the standards for road construction. Furthermore, roads which are not connected to the road network, due to the limited size of the aerial image, will not be found. To solve this problem, the interpretation framework should be extended with methods for combining neighboring photographs. Notwithstanding the current limitations, we have shown that the applicability of contextual reasoning is not restricted to maps which creates important prospects for new lines of research.

### Acknowledgments

This research project was funded by the TNO Institute of Applied Physics, and the Survey Department of Rijkswaterstaat, both in Delft, the Netherlands.

# Bibliography


[1] R. Bajcsy and M. Tavakoli. Computer recognition of roads from satellite pictures. *IEEE Trans. on Systems, Man and Cybernetics*, 6(9):623–637, 1976.

[2] M.E. de Gunst, C.S.L.A. Han, M.J.P.M. Lemmens, and R.J. van Munster. Automatic extraction of roads from SPOT images. In *Proc. of ISPRS/OEEPE Joint Workshop on Updating Digital Data by Photogrammetric Methods*, volume 27, pages 131–140, 1991.

[3] J.E. den Hartog, T.K. ten Kate, and J.J. Gerbrands. Knowledge-based interpretation of public utility maps. accepted for publication in *Computer Vision and Image Understanding*, 1995.

[4] J.E. den Hartog, T.K. ten Kate, and J.J. Gerbrands. Knowledge-based segmentation for automatic map interpretation. Submitted to *Image Vision Computing*, 1995.

[5] M.A. Fischler, J.M. Tenenbaum, and H.C. Wolf. Detection of roads and linear structures in aerial imagery by computer. In *Image Understanding Workshop*, pages 87–100, November 1979.

[6] J. Glicksman. Using multiple information sources in a computational vision system. In *Proc. of the 8th Int. Joint Conference on Artificial Intelligence*, pages 1078–1080, 1983.

[7] A.R. Hanson and E.M. Riseman. VISIONS: a computer system for interpreting scenes. In Hanson and Riseman, editors, *Computer vision systems*, pages 303–333. Academic Press, New York, 1978.

[8] M.J.P.M. Lemmens and K.F. Verheij. Automatically GIS updating from classified satellite images using GIS knowledge. In *Int. Archives of Photogrammetry and Remote Sensing*, volume 27, B4, Comm. IV, pages 198–206, 1988.

[9] T. Matsuyama and V. Hwang. SIGMA: a framework for image understanding - integration of bottom-up and top-down analyses. In *Proc. of the 9th Int. Joint Conf. on Artificial Intelligence*, pages 908–915, 1985.

[10] D.M. McKeown and J.L. Denlinger. Cooperative methods for road tracking in aerial imagery. In *Proc. of Computer Vision and Pattern Recognition (Ann Arbor)*, pages 662–672, 1988.

[11] L.H. Quam. Road tracking and anomaly detection in aerial imagery. In *Image Understanding Workshop*, pages 51–55, May 1978.

[12] J. van Cleynenbreugel, F. Fierens, P. Suetens, and A. Oosterlinck. Delineating road structures on satellite imagery by a GIS-guided technique. *Photogrammetric Engineering and Remote Sensing*, 56(6):893–898, 1990.

[13] S. Yu and M. Berthod. Urban area detection in satellite images using map knowledge by a feedback control technique. In *12th IAPR Int. Conf. on Pattern Recognition (Jerusalem)*, volume 1, pages 98–102, October 1994.

# 7

# Discussion and conclusions

# 7

# Discussion and conclusions

In this thesis, a framework for knowledge-based map interpretation has been proposed. The framework is based on three concepts, namely the low-level description of the graphics, the explicit representation of the knowledge and the contextual reasoning mechanism. In this chapter, both the merits and the limitations of these concepts are discussed. Following this discussion, our final conclusions are given.

## 7.1 Decomposition of the graphics

In Chapter 2 a new low-level method based on decomposing graphics was proposed as an alternative to the conventional method of vectorization to represent the graphics in the map. The main advantage of this approach is that compared to standard vectorization the morphological information is retained. Instead of approximating the graphics with a vector description, the graphics are decomposed into graphical primitives where each primitive is stored in a separate bitmap. As a consequence, it is possible to apply image processing algorithms to individual primitives and to calculate discriminating features. The primitive-based description therefore facilitates object recognition whereas with a plain vector description this is a difficult task, if possible at all.

The decomposition method also has an important drawback. Similar to vectorization, the decomposition method depends on the skeleton, and anomalies in the skeleton may therefore have drastic consequences for the resulting primitives. However, the same limitations are encountered with vectorization. This is illustrated in a practical situation with the example in Fig. 7.1. After the decomposition of the graphics, the vertical arrow will be decomposed into two separate primitives; a more or less triangularly shaped object and a thin straight line representing the head and the tail respectively. The current interpretation process is based on the classification of primitives, but in this case, the arrow-head is small and often distorted thereby obstructing its proper recognition. Even when using the context of these arrows, it is very difficult to design a reliable recognition strategy for them. A solution to this frequently encountered situation has to be found in the development of detectors which do not depend on the skeleton. This subject is not within the scope of this thesis and it therefore remains an important topic for future research.

In Chapter 2, we also found that the distance transforms which are used by the decomposition algorithm are expensive in computational costs and memory requirements. The straightforward implementation based on the constrained distance transform can be improved significantly. For the standard distance transformation and skeletonization, the method described by Verwer [5] can be used. In this method, the distance transformation and the skeletonization are combined in a single step while instead of a large grey value image, a memory-efficient bucket data structure is used. Further, for the binary image a special data structure is used which stores the pixels in bitwords where each bitword contains 32 pixels. Instead of processing single pixels, all processing is carried out on the bitwords, thus allowing the processing of 32 pixels in parallel [3, 4]. As a consequence, the decomposition can be sped up considerably, while memory requirements are drastically decreased. Experiments with the new implementation indicate that a binary A0-sized map (400 dpi, $16400 \times 14000$ sq. pixels) can be decomposed on a Sun Sparc 20 in approximately 200 seconds thereby requiring less than 110 mega bytes of memory. Since the decomposition step is the most expensive in terms of memory costs, it is possible to process A0-sized drawings on a professional PC equipped with sufficient memory.

Based on the above discussion, we conclude that the de-

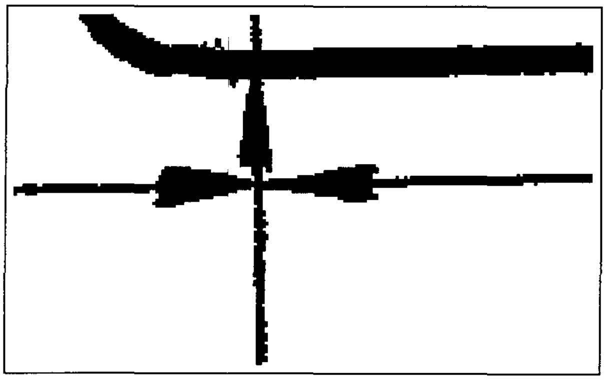


**Figure 7.1**: *The vertical arrow is touching two horizontal arrows. The decomposition step will split this arrow in two primitives which are very difficult to recognize individually.*

composition of graphics is an important first step in the interpretation of maps and its application is no longer limited to expensive workstations. Nevertheless, there are situations in the maps when neither decomposition nor vectorization is sufficient for reliable object recognition and future research should therefore aim at the development of complementary recognition strategies.

## 7.2 Explicit knowledge

In Chapter 1, we discussed the potential importance of a knowledge-based approach. The large variety of potential applications within a single utility requires a flexible solution. In this thesis, we proposed the separation of knowledge and implementation by using an explicit knowledge representation language (KRL).

When compared to a rigid implementation where all knowledge is hidden in program code, this approach appears to be advantageous. Because all knowledge can be adapted at run time, and a mechanism is provided to add new functionality, the time needed to develop an interpretation system is reduced significantly as laborious reprogramming can be circumvented. Moreover, the KRL is simple and dedicated to a single task, and its use is therefore more restricted than a general-purpose programming language such as C, but it also offers more insight in the behavior of the system than a general-purpose language. Even though the KRL decreases the time needed to develop an interpretation system, the fine-tuning of the knowledge may still be a conscientious task requiring many experiments. To further reduce development time and to facilitate the design of the knowledge base, it is necessary to explore methods and techniques for efficient knowledge acquisition. For example, an operator could select a collection of objects or pairs of objects. From these examples, it should be possible to generate parts of the knowledge file automatically.

In the current implementation, the object description consists of a list of features, and for each feature a range of allowed values is given. To classify a primitive as an instantiation of an object type, its features should be within the allowed ranges given by the object description. In other words, if an object is described by $n$ features, the list of features constructs an $n$-dimensional feature space where the ranges of allowed values represent a rectangular $n$-dimensional sub-space containing all possible objects. This recognition model assumes that all features are completely independent, but in general this will not be true. Therefore, a more sophisticated recognition scheme is required to improve the recognition performance. There are two possibilities to tackle this problem; to concentrate on developing specialized detectors such as the arrow detector described in Chapter 3, or to develop a recognition strategy which allows for the specification of relationships between features.

Another limitation of the current KRL concerns the setting of the parameters for resegmentation. The current KRL can only handle static predefined parameters for the image processing functions specified in the knowledge description file. The flexibility of the resegmentation could be improved significantly if the parameter settings of the image processing could be adapted during the interpretation. To support run time parameter adaptation, the KRL should therefore provide means to model a priori knowledge about segmentation results and, in the case the results are inadequate, how to adapt the parameters to further improve the segmentation.

For all the experiments carried out in this study, the a priori knowledge was modeled with a single semantic network. However, even in a single map, parts may differ from each other considerably. For example, in the case that a map represents a suburban area with a transformer station, most of the map will contain standard objects such as conduits, houses and dimensions. However, at the transformer station, the map may become very complex as the main conduit branches out into sub-conduits, and for each conduit the contents are depicted. For these different situations, different interpretation strategies are required. If these strategies have to be merged in a single representation, this will yield an obscure and unmanageable model. For an operational system, the use of multiple knowledge description files may be a better approach. In the automatic conversion method, which was discussed in Chapter 1, the operator selects an area suitable for automatic conversion. In the case that the map contains different situations, a practical solution may be to provide the operator with a number of interpretation strategies, and each time the operator selects an area, he or she also selects the most appropriate interpretation model.

During the research described in this thesis, the KRL proved to be a promising concept which allows for the easy and flexible development of an interpretation system. Although the current object recognition scheme provides a mechanism to add new functionality easily, the current editor-based user interface is too limited and the development of efficient tools for knowledge acquisition is a topic

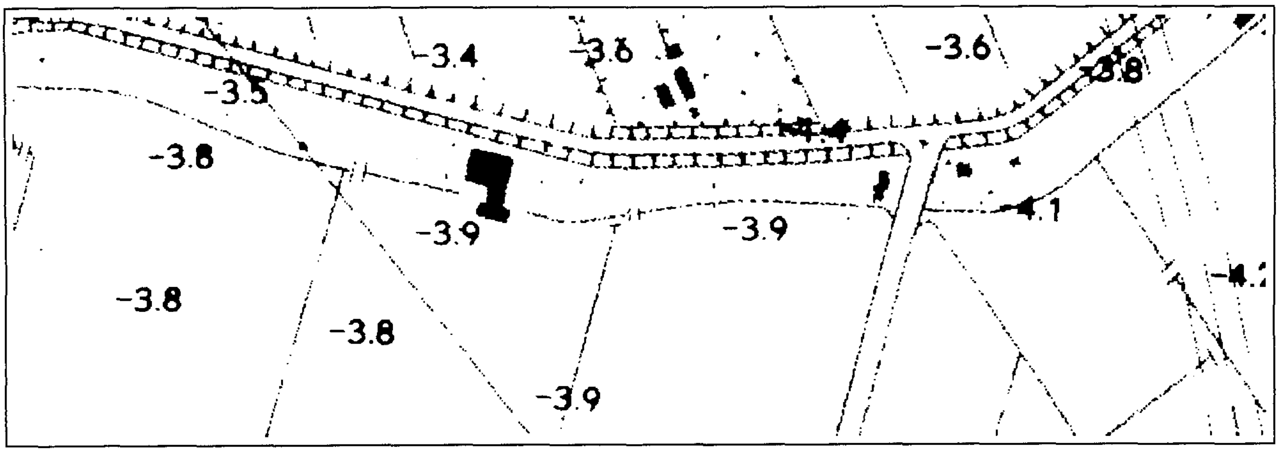



**Figure 7.2**: *A section of an altitude map. The altimetries are distributed randomly over the map and do not share a clear relationship with each other or other objects.*

for future developments. Furthermore, the object recognition is not perfect and its improvement remains an important subject of future research.

## 7.3 Contextual reasoning

The reasoning mechanism, described in this thesis, is a generic concept based on the spatial relationships between the objects, and it should apply to most map applications as well as other structured data. Its potential use for the interpretation of aerial images was demonstrated with a case-study in the previous chapter. In Chapter 4, it was shown that contextual reasoning is a simple but useful mechanism to guide the interpretation into promising directions and to verify its outcomes. Experiments indicated that without contextual reasoning, the amount of false accepts increases drastically. In Chapter 5, this concept was extended with a method to control the resegmentation process when necessary, which proved to be successful in the reported experiments. We therefore conclude that contextual reasoning is an important concept which can also be used to combine the joined effort of multiple specialized segmentation algorithms in an efficient way.

Although contextual reasoning proved to be successful, there are still some limitations. Contextual reasoning is based on a single concept shared by almost all map applications, namely that each object has one or more spatial relationships with other objects. Using the spatial relationships, the contextual reasoning mechanism decides which objects need to be searched and which actions have to be undertaken when a relationship is or is not found. In some applications, however, this concept may be only partly valid. This is illustrated by the example in Fig. 7.2, which shows a small part of an altitude map. This map indicates the altitude of an area in meters. The altimetries are distributed more or less randomly over the map and do not share a clear spatial relationship with other object types. Thus, for this type of map, the relationship model is not applicable and contextual reasoning will therefore not be useful for its reliable interpretation.

A more serious drawback is that the interpretation system suffers from myopia. To explain this, first a distinction has to be made between simple and composite objects. A simple object is a primitive with an object label attached to it, for example a digit, an arrow or a piece of the conduit. A composite object, however, is composed of multiple simple objects. A dimension, for example, may be composed of three digits and a dot. An arrow and a dimension compose a more complex composite object. The knowledge file, however, only describes binary spatial relationships between simple objects, and it is therefore not possible to obtain an understanding of the structure of the map at a high level. Furthermore, in the current concept all relationships are binary, but within a composite object poly-relationships may be distinguished. For example, a dimension is presently described as a set of two binary relationships between a dot and a digit, and, a digit and a digit. However, this is not sufficient for a complete description of the dimension. To improve both the interpretation process and the resulting map description, it should be possible to describe composite objects as a hierarchy of simple objects in terms of poly-relationships.

Another direction for improvement may be found in the integration of contextual reasoning with inexact reasoning. Presently, if an object is detected, no certainty measure is attached. The object is simply assumed to be correct until proof of the contrary is indicated by an inconsistency. However, the inconsistency detection is not infallible, and misclassifications may slip through. Inexact reasoning may offer a method to solve ambiguous situations. For example, in the current concept it is not possible to model the behavior of the system if, for a classification, both conflicting and supporting evidence exists. In this case, inexact reasoning tries to make a decision based on a statistical model. An example is the assignment of a certainty factor (CF) to each hypothesis, similar to the approach taken in the MYCIN project [2]. In this approach, the certainty factor represents the belief in the hypothesis, and the CF is affected by the available evidence. The decision of whether to accept or reject the classification is based on the value of CF. If CF is positive, the classification is believed to be true. If CF is negative there is stronger evidence against the classification and it is rejected consequently.

An inexact reasoning approach immediately raises a number of important questions:

- How should the required statistical model be obtained?
- How should probabilities propagate along with the interpretation?
- How can it be determined whether two events are independent and may be handled as independent evidence?
- For which situations is inexact reasoning appropriate?

An initial discussion on these problems can be found in *Artificial intelligence* by Rich [1]. The last question, however, is of special importance and therefore it is here discussed further. According to Rich, statistical reasoning is only appropriate when the relevant world is really random, or when there is not enough information to support exact decisions in the deterministic world. However, if the lacking information is due to an inadequate model of the deterministic world, then this should never be compensated by inexact reasoning. Since the maps are certainly not drawn at random, we can therefore restrict ourselves to the question of whether there is indeed a lack of information, or that the current model is inadequate.

Of course, there is some randomness in the maps. During image acquisition, the analog signal from the sensor, i.e. a scanner, is converted to a digital representation. In this step, the original image may be distorted as noise is added, and shading and blur are introduced. Moreover, a digital image is only a discrete approximation of a continuous world, and as a consequence, measurements in the image are always inaccurate to some extent. Further, most objects are hand-drawn and the variance in the shape of many objects can only be modeled statistically. It therefore seems a logical step to use statistical techniques for object recognition. However, all objects are recognized within their context, and the uncertainty affects the structure of the map to a much lesser extent than the appearance of the objects. It is therefore not yet clear whether it is meaningful to model contextual knowledge also statistically. As was discussed above, the current model is clearly restricted in its ability to model the context and to recognize complex composite objects. It therefore seems sensible that future research, concerning contextual reasoning, should focus first on improving the deterministic model, instead of directly assuming a lack of information and approximating the world with a statistical model.

## 7.4 Conclusions

The goal of the research described in this thesis was the development of a flexible and generally applicable framework for knowledge-based map interpretation. For a reliable and accurate interpretation, the framework has to provide methods to model and to manipulate various kinds of knowledge as well as a reasoning mechanism to control the image processing steps. From the results described in this thesis we conclude that this goal is reached. The framework proposed in this thesis is a promising and flexible approach to map interpretation. The experimental results reported in Chapters 4 and 5 indicate that the framework provides an effective mechanism to guide the interpretation and the segmentation. The objects are recognized reliably but at the cost of a significant reject rate. The fully automatic interpretation of maps is therefore not yet feasible, however, integration within a semi-automatic environment should be possible.

## 7.5 Towards an operational system

Because the development of a system for fully automatic drawing interpretation is not realistic within the coming few years, future developments should aim at incorporation of automatic techniques within an operator-assisted conversion system. The improved reconstruction, which was described in Chapter 1, seems most suitable for integration with automatic interpretation. In such an integrated approach, the operator maintains control of the conversion process by selecting the parts of the map which appear to be suitable for automatic conversion. The interpretation results are displayed on the monitor and the operator then either accepts, rejects, or manually corrects them. In the latter case, the system can provide efficient support and guide the operator to the parts of the map where problems were encountered. Currently, the TNO Institute of Applied Physics is involved in a research project, together with the Delft University of Technology and the University of Amsterdam, to develop an operational semi-automatic conversion system for the PNEM. For more details on the PNEM application, the reader is referred to Chapter 1. In this project, the research concentrates on a number of topics, i.e. improved object recognition, reasoning methodologies, the man-machine interface, the interface between the automatic interpretation and the GIS, and improvement of the warping process mentioned in Chapter 1. The system should become operational early 1997.

## 7.6 Other research directions

From the case-study presented in Chapter 6, we conclude that the applicability of the interpretation framework is not limited to maps and it should be extendable to other areas. Since the interpretation relies heavily on a priori knowledge of the spatial relationships between objects, the interpretation framework can only be expected to be effective for structured images such as the aerial images of road networks. However, the current framework expects initial results to guide the interpretation. In the map application, for example, the results are provided by the initial segmentation, while in the road application, a database provides the old situation as a start for the interpretation process. Therefore, expanding the use of the framework to other less structured domains where initial results are not available, e.g. medical imagery, is not trivial, and this remains an important topic for future research.

# Bibliography


[1] E. Rich. *Artificial Intelligence.* McGraw-Hill series in artificial intelligence. McGraw-Hill, 8th printing edition, 1988. ISBN 0–07–052261–8.

[2] E.H. Shortliffe. *Computer-based medical consultations: MYCIN.* Number 2 in Artificial intelligence series. Elsevier, New York, 1976. ISBN 0-444-00179-4.

[3] University of Amsterdam, Faculty of mathematics and computer science, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands. *SCIL-image user's manual,* March 1994. Version 1.2.

[4] R. van den Boomgaard and R. van Balen. Methods for fast morphological image transforms using bitmapped binary images. *CVGIP: Graphical Models and Image Processing*, 54(3):252–258, 1992.

[5] B.J.H. Verwer. Improved metrics in image processing applied to the Hilditch skelton. In *Proceedings 9th international conference on Pattern Recognition, Rome,* pages 137–142. IAPR, Nov. 14–17 1988.

# Summary


In the industrialized world, there is an urgent need for solutions to drawing conversion. Many institutions, such as public utilities, have large numbers of paper maps to record information about their properties and infrastructure. To exploit the full benefit of computers in the management of the infrastructure, it is necessary that the information in the maps is made available in a digital format. For example, if the management of a public utility requires an estimate of the costs and the time needed to replace a specific type of conduit in a particular area, finding the answer may require a manual query involving thousands of maps. However, if the information about the location and the contents of the conduit is available in a Geographic Information System (GIS), an automated query should provide all required information within minutes. However, current methods to convert the paper maps are labor intensive and, as a consequence, they are also very expensive. Therefore, there is a great interest in the development of automatic techniques to speed up the conversion of maps.

The goal of the research described in this thesis is the design and development of techniques which are capable of automatic drawing interpretation. Such automatic techniques should satisfy two important demands. First, the automatic conversion should be very reliable as errors in the interpretation may seriously affect later processing. Secondly, the conversion techniques need to be flexible and easy adaptable because of the enormous diversity of applications which need to be converted. Even within a single public utility, there are usually a number of applications, such as water, gas, low-voltage and high voltage networks, etc. To meet both requirements, we have chosen for a knowledge-based approach to automatic conversion in this study. To obtain a reliable interpretation, it is necessary to employ all available knowledge about the application. Since maps are drawn according to specific drawing rules using a limited set of symbols, for each type of map there is much knowledge available prior to the interpretation. The use of knowledge about applicable image processing functions and the optimal interpretation strategy further increases the reliability of the results. A knowledge-based approach also facilitates the change to another type of map. Most of the reprogramming of the application may be circumvented because it is sufficient to adjust the knowledge base. A knowledge-based approach may therefore considerably speed up development for the multitude of applications.

Within this study, a framework has been developed for knowledge-based map interpretation which integrates a dedicated knowledge representation language, image processing, and a reasoning mechanism. In Chapter 1, some basic aspects concerning the design of such a framework are discussed and a detailed description of the application is given. Chapters 2 and 3 concentrate on the preprocessing needed to be able to recognize objects in the maps. In Chapter 2, a new low-level representation is proposed as an alternative to the standard approach of vectorization. The main advantage of this approach is that, compared to standard vectorization, the morphological information is retained. Instead of approximating the graphics with a vector description, the graphics are decomposed into their graphical primitives where each primitive is stored in a separate bitmap. As a consequence, it is possible to apply image processing algorithms to individual primitives and to calculate discriminating features. The primitive-based description therefore facilitates object recognition whereas with a plain vector description this is at the least a difficult task, if possible at all. During the decomposition process, a compact and complete set of features is calculated. In Chapter 3, this feature set is used as the input for a neural network for the recognition of arrows. The experiments with the neural net indicate that with these features, it is possible to classify arrows very reliably, although at the expense of 8% reject.

Chapter 4 focuses on the use of knowledge to guide the interpretation process by means of contextual reasoning. The a priori map knowledge consists of the objects which occur in the maps, their shape, and their interrelationships. The concept of contextual reasoning is based on the observation that each object type is related to other object types. For example, an arrow usually depicts a distance between two objects, e.g. a house and a conduit. Further, related to an arrow is a dimensioning, a numerical representation of the depicted distance. Thus, detection of an object immediately generates expectations about other objects in its neighborhood, which are very suitable to generate new goals for the interpretation process. If in the map image the conduit is found, it is no longer necessary to search in the entire image for houses and arrows as these objects only appear in the direct neighborhood of the conduit. After detection of the conduit, the system can search directly for arrows. Each time an arrow is found, there should be a corresponding dimensioning. In the case that both the arrow and its dimensioning are found, a house is expected at the other end of the arrow.

However, often the situation occurs where the system expects an object which cannot be detected. For example, an arrow is always accompanied by a dimensioning, but sometimes the arrow is recognized while its dimensioning

cannot be found. Usually, there are three possible causes for such an inconsistency:

- The recognition of the arrow is incorrect.
- The segmentation of the dimensioning is incorrect causing, for example, the loss of small objects.
- The draftsman violated the drawing rules.

To be able to detect these types of inconsistencies, the model is extended with essential and negative relationships. The arrow which is always accompanied by a dimensioning is an example of an essential relationship. The negative relationship indicates that two object types should never share a specific relationship. For example, a conduit cannot have a distance to a house of less than 0.5 meters.

Most inconsistencies are due to a poor global segmentation. To be able to solve these inconsistencies automatically, the model is further refined with knowledge about image processing in Chapter 5. In the new model, both the cause of the inconsistency (the global segmentation) and its solution (a new, object-specific segmentation) can be represented. For example, in the case of the arrow and the missing dimensioning, the system first verifies with an alternative classification technique whether the detection of the arrow is correct. If the arrow's recognition is confirmed, the inconsistency is assumed to be caused by a poor global segmentation. The system then opts for resegmentation. Because the dimensioning is always drawn on either the left or the right side of the arrow, the resegmentation is limited to this area to minimize computational costs. Following resegmentation, the interpretation process retries to detect the dimensioning. If this succeeds, the interpretation continues, while the arrow is rejected otherwise.

The developed concepts were evaluated with utility maps from a Dutch public utility. In the experiments described in Chapter 4, we found that contextual reasoning is successful in guiding the interpretation process and in detecting potential errors. With contextual reasoning, the number of erroneous classifications decreased drastically when compared to a straightforward approach without use of contextual evidence. However, contextual reasoning reduces the number of errors at the cost of a significant increase of rejects. The experiments in Chapter 5 show that by using knowledge-based resegmentation, many inconsistencies, which otherwise led to rejects, can be prevented. For example, with knowledge-based segmentation, it is possible to increase the percentage of recognized dimensionings more than 65%.

In Chapter 6, the applicability of the developed concepts for aerial image interpretation is explored. Aerial images are usually highly structured as they contain carefully planned and constructed man-made objects. A complex object, such as a road network, is composed of many different and specialized elements, e.g. roads, lanes, bridges and fly-overs, and all these objects share specific spatial relationships with each other. Each of these objects may require a specialized segmentation and recognition strategy. Thus, a knowledge-based approach to the interpretation of these complex images using some kind of contextual reasoning strategy seems appropriate. The potential use of contextual reasoning for this specific domain is illustrated with a case-study.

The developed framework and the experimental results are discussed in Chapter 7. The main conclusion is that the framework proposed in this thesis is a promising and flexible approach to map interpretation. The experimental results indicate that the framework provides an effective mechanism to guide the interpretation and the segmentation. The objects are recognized reliably but at the cost of a significant reject rate. The fully automatic interpretation of maps is therefore not yet feasible, however, a considerable speed-up can be achieved by providing efficient support to the operator with the developed automatic techniques.

Because the development of a system for fully automatic drawing interpretation is not realistic within the coming few years, future developments should aim at incorporation of automatic techniques within an operator-assisted conversion system. Currently, the TNO Institute of Applied Physics is involved in a research project, together with the Delft University of Technology and the University of Amsterdam, to develop an operational semi-automatic conversion system for a Dutch utility which should become operational early 1997.

Summary of the thesis *A framework for knowledge-based map interpretation.*

Author: Jurgen den Hartog, June 1995.

# Samenvatting

Veel instellingen in Nederland, zoals bijvoorbeeld nutsbedrijven, beheren enorme archieven met kaartmateriaal. De mogelijkheden van de steeds verder oprukkende automatisering kunnen echter niet worden benut zolang de informatie in deze kaarten niet voor de computer beschikbaar wordt gemaakt. Dit probleem zal worden toegelicht aan de hand van een voorbeeld: stel, het management van een electriciteitsbedrijf wil een precieze schatting maken van de hoeveelheid tijd en geld die nodig is voor de vervanging van een bepaald type leiding. Dit komt in de praktijk neer op het handmatig nalopen van het volledige kaartbestand, wat wel 10.000 kaarten kan bedragen, zodat het antwoord enige tijd op zich kan laten wachten. Als daarentegen alle relevante informatie over het netwerk in een database beschikbaar is, dan hoeft een gerichte automatische zoekactie slechts enkele minuten te duren. De conclusie is dat voor het efficiënt beheren van allerlei geografische informatie de conversie van kaartmateriaal naar een digitaal formaat noodzakelijk is.

De huidige conversietechnieken zijn zeer arbeidsintensief en daardoor zeer duur. Bij één van deze technieken wordt de informatie op de kaart door de tekenaar overgetekend op een beeldscherm waarbij de tekenaar zijn aandacht moet verdelen over één of meer kaarten én het beeldscherm. De handmatige conversie van alle kaarten en tekeningen van de nutsbedrijven, PTT Telecom, de gemeenten, enz. kost naar schatting 10.000 manjaar werk, wat neerkomt op ongeveer 1 miljard gulden. De ontwikkeling van automatische technieken die dit tijdrovende en kostbare proces kunnen versnellen heeft dan ook een grote maatschappelijke relevantie.

De doelstelling van het in dit proefschrift beschreven onderzoek is de ontwikkeling van dergelijke automatische technieken. In de praktijk komt automatische kaartconversie allereerst neer op het *scannen* van de kaarten. Tijdens het scan-proces wordt de papieren kaart omgezet naar een numerieke representatie: het digitale beeld. Vervolgens moeten in het digitale beeld de afzonderlijke objecten en hun onderlinge structuur worden herkend met behulp van digitale beeldverwerking.

Voor succesvolle toepassing van automatische conversietechnieken moeten deze betrouwbare resultaten opleveren, aangezien fouten in de geografische database door kunnen werken in de hierop gebaseerde besluitvorming. De betrouwbaarheid van de interpretatie wordt vergroot door alle beschikbare relevante voorkennis over de kaarten te benutten. Deze voorkennis beslaat kennis over de vaste tekenregels, de te volgen interpretatiestrategie en bruikbare beeldverwerkingstechnieken. Het gebruik van kennis maakt een interpretatiesysteem bovendien flexibeler, mits deze is beschreven in een makkelijk manipuleerbare vorm. Vanwege de grote diversiteit aan te converteren kaarten is dit eveneens een belangrijk aspect. Om op een efficiënte manier een betrouwbaar interpretatiesysteem te bouwen is in dit onderzoek een raamwerk ontwikkeld waarin een eenvoudige kennis-representatietaal is geïntegreerd met een redeneermechanisme en een beeldverwerkingssysteem.

Het principe van kennisgestuurde interpretatie zal worden uitgelegd nadat eerst is ingegaan op de voorbewerkingen die nodig zijn om herkenning van objecten mogelijk te maken. De grafische objecten in een kaart zijn meestal met elkaar verbonden en vormen een"klont" pixels waarin door de computer weinig structuur ontdekt kan worden. Een veel gebruikte methode om herkenning mogelijk te maken is vectorisatie. De klont pixels wordt benaderd met een graafachtige structuur, opgebouwd uit rechte lijnen en hun verbindingspunten, zoals bijvoorbeeld te zien is in figuur 2.1 van hoofdstuk 2. Een vectorisatie is meestal een redelijke benadering van de grafische topologie, maar de benadering van verschillend gevormde objecten met een rechte lijn gaat ten koste van de morfologische informatie wat de herkenning bemoeilijkt. Als alternatief voor vectorisatie onderzochten wij het opdelen van graphics in primitieven, zoals bijvoorbeeld te zien is in figuur 2.3 en figuur 2.6. Deze methode geeft een abstracte representatie van de topologie, echter zonder verlies van de zo belangrijke morfologische informatie.

Tijdens het opdelen van de grafische structuur in primitieven wordt een compacte en volledige set kenmerken berekend. Deze set van ongeveer 80 getalswaarden blijkt een grafisch symbool te kunnen representeren zonder verlies van morfologische informatie. Kenmerkend voor bijvoorbeeld een pijl zijn de driehoekige verdikkingen aan één of beide uiteinden die zijn verbonden door een dunne, rechte lijn. In hoofdstuk 3 wordt uitgelegd hoe deze kenmerkverzameling wordt berekend en hoe een neuraal netwerk hiermee getraind kan worden. Uit de experimenten met het neurale netwerk blijkt dat pijlen met een grote betrouwbaarheid herkend kunnen worden.

In hoofdstuk 4 wordt een eerste concept beschreven om met behulp van kennis de interpretatie te sturen. Kennisgestuurde kaartinterpretatie kan met het volgende simpele praktijkvoorbeeld wat inzichtelijker worden gemaakt:

Een electriciteitsbedrijf tekent op zijn kaarten de positie van de electriciteitsleiding ten opzichte van de huizen die ermee van electriciteit worden voorzien. Bij onderhoud van een stuk leiding wordt eerst de straat en het dichtst

bij gelegen huis opgezocht. Uitgaande van de gevel kan dan eenvoudig de plaats van de ondergrondse leiding worden gevonden. Op de kaarten komen de volgende objecten voor:

- Leidingen, die worden weergegeven door een dikke en regelmatige rechte lijn.
- De gevels van huizen, gerepresenteerd door een rechte hoekige lijn. Binnenin het huis is vaak een huisnummer getekend.
- Bematingen, die de afstand tussen leiding en huis aangeven. Een bemating wordt getekend als een pijl met een getal dat de door de pijl weergegeven afstand tot op een decimeter nauwkeurig weergeeft.

Bij deze applicatie bestaat de voorkennis dus onder anderen uit informatie over welke objecten er zijn, hoe ze eruit zien en in welke samenhang ze voorkomen. Als bijvoorbeeld eerst in het kaartbeeld de leiding is gezocht én gevonden, dan is het niet meer nodig om in het gehele beeld naar de andere objecten te zoeken. Immers, de huizen en de bematingen komen alleen in de buurt van de leiding voor en zoekacties naar deze objecten kunnen daarom hiertoe beperkt worden. Nadat de leiding gevonden is, kan het interpretatiesysteem gericht gaan zoeken naar pijlen. Telkens als er een pijl is gevonden dan moet daar ook een getal bij staan. Als zowel pijl als getal zijn gevonden dan moet aan de andere kant zich een huis bevinden. Na detectie van het huis kan in het kleine gebied binnen het huis naar een huisnummer worden gezocht. Op deze manier treedt er een soort sneeuwbaleffect op waarbij elk gevonden object weer tot nieuwe zoekacties leidt die op hun beurt weer nieuwe zoekacties kunnen veroorzaken. De basis van de interpretatie is dus een van te voren opgesteld model waarin wordt beschreven welke objecten voorkomen en welke spatiële relaties ze onderling delen.

Maar dan rijst vervolgens de vraag hoe te handelen als het beeld niet aan dit model blijkt te voldoen. Het is bijvoorbeeld bekend dat een bemating altijd uit een pijl en een getal bestaat. Maar soms kan wel de pijl worden gevonden maar niet het getal. Voor een dergelijke tegenstelling tussen model en beeld zijn drie mogelijke oorzaken:

- De herkende pijl is geen echte pijl.
- Beeldruis verstoorde de voorbewerkingen op het beeld, waardoor bijvoorbeeld de cijfers in het getal aan elkaar gegroeid zijn of juist zoek zijn geraakt.
- Het getal is niet volgens de regels getekend.

Om dit soort fouten te kunnen detecteren is het model uitgebreid met twee soorten spatiële relaties: de essentiële en de inverse relatie. De pijl die altijd samen met een getal voor moet komen is een voorbeeld van een essentiële relatie. Een inverse relatie geeft juist het tegenovergestelde aan: twee objecten mogen niet op een bepaalde manier samen voorkomen. Een leiding mag bijvoorbeeld wel langs de huizen lopen, maar niet naar binnen gaan. Met behulp van dit model kan de kennis over potentiële tegenstellingen op een eenvoudige manier gemodelleerd worden. De enige voorwaarde voor toepasbaarheid van dit model is dat de objecten in de kaart duidelijke spatiële relaties met elkaar delen.

In hoofdstuk 5 wordt verder ingegaan op het afhandelen van gevonden tegenstellingen. Heel vaak ligt de oorzaak bij de segmentatiestap van de voorbewerking. Segmentatie van kaarten komt grofweg neer op het onderscheiden van voorgrond en achtergrond. Het originele kaartbeeld bestaat uit een enorme verzameling pixels. Elk pixel representeert de grijswaarde van een zeer klein vierkant gebiedje in de kaart. Idealiter is deze grijswaarde een betrouwbare maat voor de hoeveelheid inkt binnen dit gebied. Er blijken echter allerlei factoren te zijn die het scanproces verstoren. Zo is het bijvoorbeeld niet mogelijk om de meting van de intensiteit tot precies het gewenste pixel te beperken omdat omliggende pixels de meting beïnvloeden. Verder is het kaartmateriaal van wisselende kwaliteit en tot slot worden de kaarten vaak aan beide kanten betekend. Kortom, het is niet triviaal om inkt van papier te onderscheiden en één enkele eenvoudige segmentatiestap is niet afdoende. Maar ook hier kan het gebruik van kennis soelaas bieden. Door in het model zowel de oorzaak van de tegenstellingen (de segmentatiestap) als de oplossing hiervoor (een nieuwe, gespecialiseerde segmentatiestap) te modelleren, kan het systeem een belangrijk deel van de problemen volautomatisch oplossen. Op leidingkaarten wordt bijvoorbeeld vaak het bematingsgetal niet herkend door een slechte segmentatie. De hieruit volgende tegenstelling wordt als volgt opgelost: Omdat het getal altijd links of rechts van de pijl staat, kan nadat de pijl is gevonden een nieuwe segmentatiestap worden beperkt tot een klein gebied aan weerszijden van de pijl. Hierdoor blijven de rekenkosten beperkt. Na de hersegmentatie, gespecialiseerd in optimale extractie van het getal, wordt opnieuw geprobeerd het getal te vinden. Als dit lukt dan gaat de interpretatie weer gewoon door en anders wordt aangenomen dat de pijl foutief herkend is en wordt deze herkenning vervolgens verworpen.

Uiteraard zijn er ook experimenten uitgevoerd om een indicatie te krijgen hoe het systeem in een praktijkomgeving zou werken. Uit deze experimenten blijkt dat het redeneren met relaties een goede manier is om de interpretatie te sturen en ongerechtigheden in de interpretatie te ontdekken. Het aantal foutieve classificaties blijkt af te nemen, vergeleken met een rechttoe-rechtaan interpretatie die geen gebruik maakt van kennis over de samenhang van objecten. Wel neemt het het aantal ongeclassificeerde objecten sterk toe, maar hier blijkt het mechanisme voor hersegmentatie succesvol te zijn. Met kennisgestuurde hersegmentatie is het bijvoorbeeld mogelijk om het percentage van herkende bematingsgetallen met meer dan 65% te verbeteren.

Uit dit alles kan de conclusie worden getrokken dat het ontwikkelde raamwerk het mogelijk maakt om op een redelijk eenvoudige en flexibele manier de kennis over een bepaald type kaart te modelleren. Het redeneren met relaties levert een betrouwbare interpretatie op waarbij typische foutpercentages tussen de 0% en de 1% liggen. Het blijkt echter niet mogelijk om de gehele kaart automatisch te interpreteren. Het aantal niet geclassificeerde objecten kan per type sterk verschillen. De leiding zelf kan bijvoorbeeld vrijwel volledig worden gevonden terwijl van de pijlen ongeveer driekwart wordt gedetecteerd. De belangrijkste conclusie van dit onderzoek is dan ook dat met het huidige systeem volautomatische kaartinterpretatie nog niet haalbaar is, maar dat het zeker mogelijk is om de handmatige conversie aanzienlijk te versnellen met behulp van semi-automatische technieken.

Op dit moment is er een project gaande waarbij de Technisch Physische Dienst van TNO, in samenwerking met de faculteiten Technische Natuurkunde en Electrotechniek van de Technische Universiteit Delft, en de faculteit Wiskunde en Informatica van de Universiteit van Amsterdam, een semi-automatisch conversiesysteem gaat ontwikkelen voor de PNEM, de Provinciale Noordbrabantse Energiemaatschappij. In dit project zal ondermeer aandacht worden besteed aan de verdere ontwikkeling en verbetering van het redeneermechanisme en de symboolherkenning. De huidige symboolherkenning is te afhankelijk van de opdeling in primitieven zoals beschreven in hoofdstuk 2. Hoewel dit een zeer bruikbare abstracte representatie oplevert zijn er situaties waar er behoefte is aan andere methoden van objectherkenning. Een tekortkoming van het huidige redeneermechanisme is dat het niet mogelijk is om samengestelde objecten te representeren en als zodanig te herkennen. Toekomstig onderzoek zou zich dan ook deels moeten richten op zowel een verbetering van het kennismodel als aanpassing van de redeneerstrategie. Verder is er behoefte aan technieken om tegenstrijdige informatie te hanteren, waarbij het gebruik van inexacte redeneertechnieken van belang kan zijn. Een ander belangrijk punt van aandacht binnen het project is de koppeling tussen het conversiesysteem enerzijds en het geografisch informatie systeem anderzijds, zodat het mogelijk wordt om de informatie uit de geïnterpreteerde kaarten efficiënt op te slaan. Het is de bedoeling dat het te ontwikkelen systeem begin 1997 wordt ingezet binnen het lopende conversie-project van de PNEM.

Het ontwikkelde raamwerk is gebaseerd op algemene principes en het zou daarom bruikbaar kunnen zijn voor andere toepassingen. Omdat het huidige interpretatie-systeem volledig afhankelijk is van kennis over de spatiële relaties tussen objecten, kan van dit concept alleen verwacht worden dat het bruikbaar is voor gestructureerde beelden. De potentiële mogelijkheden voor het vinden van wegen in luchtfoto's zijn in hoofdstuk 6 met een case-study aangetoond. Het huidige interpretatie-systeem verwacht echter wel initiële resultaten om de interpretatie te sturen. Bij de kaartinterpretatie worden de eerste resultaten verkregen door een initiële segmentatie, terwijl een database met daarin de oude wegsituatie voor een vliegende start zorgt bij de interpretatie van luchtfoto's. De toepassing van de ontwikkelde methoden op minder gestructureerde beelden zonder initiële resultaten is daarom een ander belangrijk onderwerp voor vervolgonderzoek.

Samenvatting behorende bij het proefschrift *A framework for knowledge bases map interpretation.*

Auteur: Jurgen den Hartog, juni 1995.

# Curriculum Vitae

I was born in Rotterdam on September 3, 1968. In 1986, I obtained the VWO-diploma from the Rijnlands Lyceum in Sassenheim, and I continued my education with a study Computer Science at Leiden University. In August 1990, I started my graduation research on interactive segmentation and visualisation of volume data, resulting in my university degree in Applied Computer Science in February 1991.

Following my graduation, I started with my Ph.D. research at the Information Theory group of the department of Electrical Engineering of the Delft University of Technology, and the TNO Institute of Applied Physics (TPD). As a researcher at the TPD, I worked the first two years on ROCKI (Raster to Object Conversion aided by Knowledge based Image processing), an ESPRIT project sponsored by the European Community. In this project, the TPD, Océ-Nederland and three European partners developed a toolbox for automatic interpretation of maps and office documents. After the successful conclusion of ROCKI in April 1993, I concentrated on the further development of techniques for knowledge-based map interpretation. Currently, I am working on the sequel to ROCKI, a project in which the TPD, the Delft University of Technology and the University of Amsterdam participate. The aim of this project is to develop a semi-automatic conversion system for a Dutch public utility which should become operational early 1997.

During my research I presented my work on the 3rd Symposium on Document Analysis and Information Retrieval (Las Vegas, April 1994), the 12th International Conference on Pattern Recognition (Jerusalem, October 1994), and the 1st Graphics Recognition Workshop (Penn State, August 1995). In April 1994, I visited several research sites in the United States and Canada.

Besides knowledge-based image processing, my interests include squash, hiking, scuba-diving, and ways to get enough time for these activities.

# Tot slot

Hoewel de promovendus in een promotie-onderzoek moet laten zien dat hij in staat is om zelfstandig onderzoek te doen, is een promotie nooit het werk van één enkel persoon. In zijn directe omgeving zijn daar allereerst de begeleiders. Ik zelf heb het geluk gehad om van diverse kanten sturing, kritiek, ideeën, kortom begeleiding te krijgen. Hiervoor wil ik als eerste Jan Gerbrands en Eric Backer bedanken, niet in de laatste plaats voor het bewaken van de grote lijn en de planning van dit onderzoek. Als TPD-er heb ik het grootste deel van het onderzoek "aan de overkant" verricht waar ik de dagelijkse begeleiding kreeg van Ton ten Kate. Ton, jouw stimulerende en altijd kritische begeleiding zijn voor mij enorm waardevol geweest. De fijne samenwerking maakte al die "5-minuten-projectjes" meer dan goed. Albert Vossepoel en Bob Duin van de vakgroep Patroonherkennen wil ik graag bedanken voor de vele vruchtbare discussies tijdens de KGB-zittingen.

Bij de TPD kan ik twee soorten collega's onderscheiden: de collega's waarmee je samenwerkt en de collega's die je van je werk houden. Dit laatste onder meer met squash, onderhandelingen over de toekomst van Europa, jongleren en onzinnige weddenschappen met een vlaai als inzet. Beide soorten collega's zijn even hard nodig en gelukkig bleek er tussen beide ook een behoorlijke overlap te zijn. Speciaal wil ik Marlies de Gunst en Bernardus Holtrop bedanken. De samenwerking met jullie heeft in belangrijke mate bijgedragen aan mijn onderzoek, mijn proefschrift, mijn software en zelfs mijn conditie, maar ik wil jullie vooral bedanken voor alle vriendschap en gezelligheid, zowel binnen als buiten werktijd, die voor mij minstens zo belangrijk zijn geweest. Met Rik Janssen deelde ik bijna 4 jaar lang de kamer en een voorkeur voor de enige echte tekstverwerker. Rik, bedankt voor de hulp als ik weer eens problemen had met LaTeX.

Ook buiten het werk zijn er mensen die indirect een belangrijke bijdrage hebben geleverd. Marcel bedank ik voor zijn vriendschap in de afgelopen jaren. Henk en Tinie hebben me van jongs af aan gestimuleerd om te studeren en dit werk is ook daarvan het resultaat. Met mijn "grote" broer deel ik dezelfde interesses voor de belangrijke dingen in het leven. Roland, in de afgelopen periode heb je meer voor mij betekend dan slechts de voorkant van dit proefschrift. Tot slot wil ik Marina bedanken. Zonder jou had ik dit boekje waarschijnlijk drie maanden eerder af gehad maar dan waren de afgelopen drie jaar nooit zo bijzonder geweest.

# STELLINGEN

behorende bij het proefschrift

# A framework for knowledge-based map interpretation

door

Jurgen den Hartog

1. Een gedegen evaluatie bij al het onderzoek over kaartinterpretatie zou de toepasbaarheid hiervan aanzienlijk vergroten.
2. Het modelleren van kennis over kaarten is belangrijker dan het ontwikkelen van robuuste algoritmen.
3. Eén leidingkaart zegt meer dan duizend vectoren.
   *Hoofdstuk 2 van dit proefschrift.*
4. Het verkrijgen van een goede segmentatie van kaartmateriaal is vrijwel nooit triviaal en deze essentiële stap blijft ten onrechte onderbelicht in de literatuur.
   *Hoofdstuk 5 van dit proefschrift.*
5. De Amerikaanse spelling is logischer dan de Britse en zou daarom voorkeurspelling moeten zijn in de wetenschap.
6. De grootste obstakels van professioneel PC-gebruik zijn de slechte standaardisatie van de hardware en Microsoft Corporation.
7. Uit de besluitvorming rond de uitbreiding van Schiphol en de aanleg van de Betuwelijn blijkt dat in Nederland niet alleen kennis maar ook desinformatie macht is.
8. Een drastische beperking van de vleesconsumptie is de gezondste en ecologisch meest verantwoorde oplossing voor het wereldwijde tekort aan voedsel.
9. Met de huidige groei van het verkeer op Internet zal men ook op de digitale snelweg van 9 tot 5 in de file staan.
10. Een proefschrift op A4-formaat steekt ook in milieu-vriendelijkheid uit boven een proefschrift op standaardformaat.