Pattern Analysis & Applications (2000)3:88–103

Technical Map Interpretation: A Distributed Approach

J.-M. Ogier, R. Mullot, J. Labiche and Y. Lecourtier

Laboratoire PSI, Université de Rouen, Mont Saint Aignan, France

Abstract: This paper deals with the general problem of document understanding. We propose the description of a formal architecture of a device capable of interpreting technical and cartographic documents. This device relies on two main points, i.e. a model of the document and the implementation of a set of 'builders', the aim of which is to progressively construct information of as high a semantic level as that provided by the document drawer. Two main stages are integrated in the reasoning process: the first one consists in constructing the information, through a bottom-up approach. Then, a cycling stage is triggered to solve ambiguities detected by the system and corresponding to inconsistent objects with regard to the document model. In this paper, the whole approach is presented in the context of the French cadaster interpretation. The first implementation has enabled us to quantify the interpretation results and to verify the relevance of the cycling stage.

Keywords: Consistency analysis; Document modelling; Document understanding; Engineering drawings; Interpretation cycle; Performance evaluation

1. INTRODUCTION

Nowadays, a very large number of paper documents are not yet integrated into a digital storage and exchange structure. This is particularly true for engineering plans, charts and cartographic data, for which the conversion from paper to digital format is very expensive and time-consuming. Filipski [1] indicated in 1992 that 3.5 billions of technical documents are drawn on paper support (US and Canada), and 26 million paper documents are created each year.

As far as we know, no automatic retro-conversion software offers a generic tool capable of transforming these paper documents into a digital format. The problem is mainly due to the large diversity of supports and graphic representations, which range from the blueprint for a car part in the automotive industry, or a diagram of a gas duct, to the drawing of a parcel on a cadastral map.

The work presented in this paper can be placed in the context of automatic retro-conversion software, and proposes an interpretation device for cadastral maps. This work tries to provide some answers to the difficulties which are encountered by classical systems. These problems are mainly due to the lack of genericity, because of the document domain knowledge which is generally mixed with the source code. These difficulties are also due to the fixed sequential ordering of the processes which are used in the interpretation device. This 'fixed' strategy induces some local mistakes which propagate themselves from one process to another. At the end of the global interpretation, this mistakes propagation is translated into inconsistent data which is not interpretable by the system. Generally, this kind of problem induces heavy human operator corrections, rendering the system uninteresting from the implementation point of view. In this paper, we propose an interpretation device based on a modelling of the document knowledge, in order to try to extract the domain document knowledge from the source code. Secondly, we present the 'perceptive cycle', which is a concept allowing us to solve the mistakes propagation.

First, in this paper we present the concepts proposed in our interpretation strategy, which are based on the integration of knowledge into a processing cycle. Then, we present the first experimentation of the device; a quantitative evaluation on a set of French cadastral maps illustrates the methodological contribution. Finally we provide an overview of the interpretation approaches found in the literature for technical and cartographic documents, and try to position our approach relative to these classical systems.

Received: 12 May 1999
Received in revised form: 8 February 2000
Accepted: 25 February 2000

2. SYSTEM OVERVIEW

The cadastral map, which is the principal object of this study, is a good example of the kind of document we are dealing with [2–4], as are the more general cartographic maps used in civil engineering [5] for telephone, water and gas networks, etc. The digitisation is performed in black and white, with a 400 dpi resolution.

A processing cycle, called 'the perceptive cycle' [6] Fig. 1, builds up the elements of the interpretation, and verifies the consistency of the extracted data, in accordance with the knowledge integrated in the device. This knowledge corresponds to the modeling of the document to be processed: it contains all the objects that one can find on the document, and all the spatial and semantic relationships between each of them.

The processing cycle consists of different stages, starting with a 'bottom-up' approach, also called a 'construction stage', and continuing with a re-cycling on those objects detected as inconsistent at the upper level process of the system.

As mentioned before, in this process there are two successive stages, respectively a 'construction stage' and a 'consistency management stage', plus a user interface at the end of the interpretation, allowing a human operator to manage ambiguities unresolved by the system.

- *Construction stage*
  This stage provides the first proposition for the interpretation of the document. The elements of the interpretation are built up on the basis of a set of modules, called 'builders', which extract the information to be interpreted. The knowledge integrated in the device has been distributed into different representation levels [2]. These levels have been developed from the observation of the documents, and they try to reproduce the way in which a cadastral agent draws the plans. Thus, a parcel will be at a higher level than strokes, hatched areas, and arrows, since a cadastral agent uses all of these lower level elements to represent the parcel.
  From this cadastral map model, we have implemented 'builders'; these are, in fact, processing modules in our interpretation device, and they build up the information from one N level to the next N+1 level. For instance, a parcel, which is a level 1 object, is built up from the elements in level 0 by associating the textures, characters, arrows and linear objects (strokes) extracted by the image processing operators.
- *Consistency management*
  This first interpretation generally carries a certain number of interpretation mistakes (parcels without numbers, badly segmented textures, etc.). An evaluation of the results is thus implemented to analyse the consistency of the data [5,7]. The analysis involves two steps: consistency definition and consistency management:
  — *Consistency definition* means that the information built up by the device is evaluated during the interpretation process [8].
  — *Consistency management* proposes processing alternatives to eliminate the inconsistencies. A particular attribute is assigned to each inconsistency, and as a function of this attribute, a set of remedial solutions is proposed. The remedial solution may either modify the parameters of the construction process used, or propose a completely different operator.

The human operator can either manage the inconsistency manually, or enrich the device by proposing a new remedial solution to be integrated into the interpretation process.

3. FIRST ANALYSIS: BOTTOM-UP APPROACH

3.1. Document Models and Knowledge

Specific knowledge is required to envisage the different construction and evaluation stages in the interpretation of documents. This knowledge can correspond to the drawing rules used by the cadastral agent to draw the map, the different graphic elements included in the map, the links between each element, and it can be linked to the construction and evaluation process applied to the information extracted from the document.

- The first part of our work consisted in establishing a list of the map elements, and in providing a digital representation of these elements as a set of objects. Figure 2 shows some objects identified after a fine analysis of the captions of different maps (discussion with the agents). Thus, the 20 most frequently-met objects considered as representative of the document are selected. The less common objects have also been listed, but have not yet been integrated in this version of our device. Each object has been integrated in a knowledge model. Observation of the French cadaster has highlighted four levels, which are represented on Fig. 2. For the interpretation, we have taken these four levels and associated each element of the cadaster (parcel, quarter, road, etc.) to an object (a

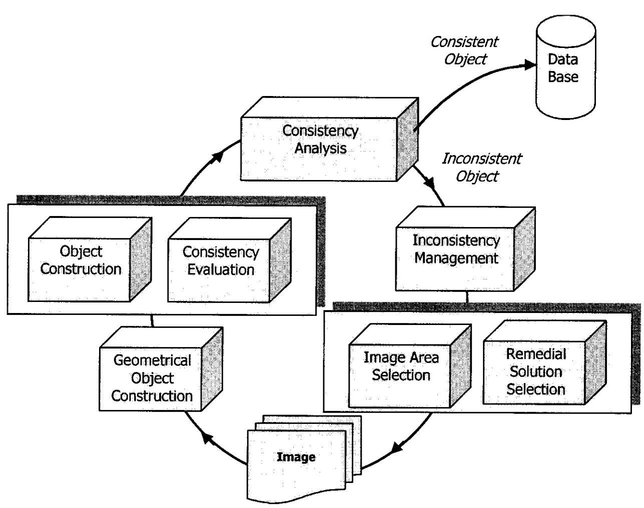


Fig. 1. Interpretation cycle.

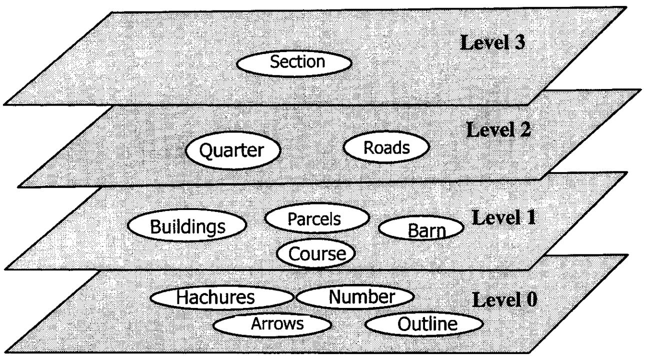


Fig. 2. Document model.

class instantiation) with the same name. Thus, a quarter corresponds to a level 2 object, and will be the digital object representing a set of contiguous parcels, thus forming an isolated block, separated from the others by roads. Parcel, quarter and road objects can be generalised to other countries: Japan [9], Italy [10], Holland [3], etc.

- The second type of knowledge integrated in our interpretation device deals with the conventions and drawing rules used by the cadastral agent when creating his document. This knowledge must be able to take into account the intrinsic variability of the documents drawn by the agent, and some explicit rules, similar to those used by the agent when he creates his plan, are assigned to them. We have chosen to clarify a certain number of properties used to construct the objects. First, the implemented rules deal with the linking and grouping of primitives or with the linking of different objects [2,11] on the same level in order to build an object with a higher semantic representation level.

  For instance, the properties of a parcel will depend upon the entities that make it up: strokes, toponyms (strings of characters), outlines and, optionally, arrows, textures, and so on. This construction is characterised by topological links, as is that proposed by den Hartog et al [5]. If one of these entities does not respect the rule governing the object considered, the interpretation device must emit 'a doubt', since the description does not correspond to the model. The doubt will be transcribed in the form of an inconsistency indicator.

From these different criteria, the construction of each of the objects of the model can be shown in the form of tables (Tables 1 and 2) integrating the assembly rules of the object. For instance, Tables 1 and 2 show some rules for the objects 'parcel' and 'quarter'.

The cadaster parcel is the most complex of the basic objects to be described. In comparison, the description of the quarter is much simpler.

As can be seen in these tables, this representation of the expertise is quite separate from the processes for extracting the information, which are image processing techniques and object construction algorithms. We have, therefore, based our approach on the agent's professional expertise and construction work, instead of on the extraction processes. Taking into account this expertise and integrating it into the interpretation device requires a great deal of work.

The integration of this knowledge in the device is interesting from two points of view. At first, it permits to extract the document domain knowledge from the source code, which is interesting for the genericity and the adaptability point of view. Secondly, it permits us to pre-define the consistency criteria that can be used in the processing cycle.

3.2. Geometrical Objects

The first stage of the bottom-up process concerns primitive extraction. These operators are primitive extractors, or image processing algorithms, which obtain geometrical primitives (geometrical objects) from a black and white image. The primitives are generally full or dashed lines, characters and strings, particular symbols, arrows, regular textures, as well as all the specific symbology of the document under analysis [6].

The low level process for constructing geometrical objects can either:

1. Apply different primitive extractors independently on the original image, or on an image with a different resolution, when the geometrical objects are not dependent.
2. Run the processing operators sequentially so that they label the image pixels as the chain progresses, by assigning to them semantic information pertinent to the current processing. For instance, on the French cadastral map, the parcel outlines are easier to extract if the hatched areas have been extracted beforehand [2,7].

All these treatments are integrated in a 'work plan', which describes the strategy for the extraction of geometrical objects at any given moment.

As Fig. 3 shows, our initial scenario for the primitive extraction is based on an approach that uses cues from global images (low resolution) and local images (high resolution). Thus, the extractors that process the low resolution image propose a hypothesis for the extraction of primitives from the high resolution image. An overall view of our approach to primitives extraction is presented elsewhere [7]. It relies on classical image processing techniques such as regular texture extraction [2,12], symbol and character recognition [13,14], and linear object analysis [3,15,16].

From a detailed point of view, our hatched area segmentation process relies on a regular texture characterisation: our texture is characterised by an elementary primitive and by a displacement vector. The elementary primitive is composed by a set of black and white occurrences corresponding, respectively, to the average thickness of the lines, and to the average space between two lines. This characterisation allows us, through an elastic template matching technique [17], to extract and localise the hatched areas.

The hatched area extractor is characterised by a set of parameters that correspond to the rigidity of the algorithm. The more these parameters are constrained, the more the hatched extractor will be severe in the detection process. This set of parameters will be used in the set of remedial solutions, when we will try to solve ambiguities at the upper level of the process. Some details concerning the

Table 1. Part of 'parcel' model (* = required number)

Object	$N(*)$	Rules	Fct	Entity	N	Rules	Specification
Outline	=1	Included		vectors	≥ 3	Shared	essential
			Or	Toponym	≥ 1	Included	essential
				And incoming arrow	=1	Shared	
			Or	And identifier	=1	neighbour	essential
Identifier	=1	Included		And identifying arrow	=1	Shared	
			Or	And identifier	=1	neighbour	essential
Hatching	=1	Included		2D Texture			optional
Dashed line	≥1			1D Linear texture			optional
Boundary mark	≥1			Symbol	≥1	Shared	optional
Mail box number	≥1	Neighbour		Toponym	=1	neighbour	optional
Out going arrow	≥1	Shared		Symbol	≥1		optional

Table 2. Part of 'quarter' model

Objects	N	Rules	Fct	Entity	N	Rules	Specification
Outline	=1	included		vector	≥3	shared	essential
parcel	≥1	included		parcel	≥1	included	essential

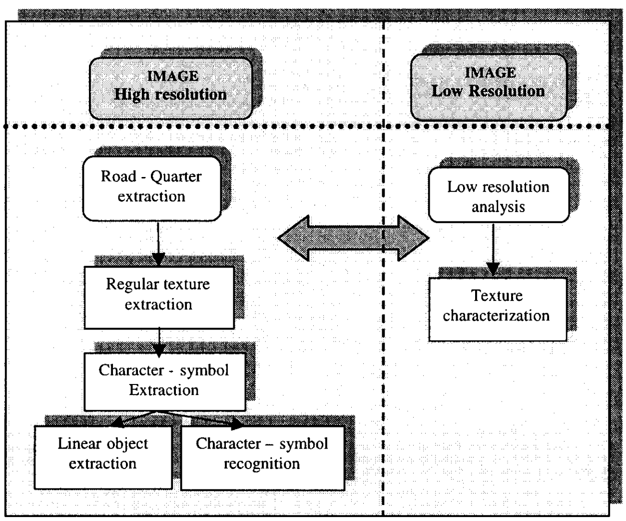


Fig. 3. Initial work plan during the bottom-up approach.

implementation of this algorithm can be found elsewhere [17].

Concerning the vectorisation process, our strategy is based on classical tools like that developed elsewhere [3,15,16]. The implementation of the measurement of the quality of our vectorisation process relies on criteria based on the Haussdorf distance [18,19].

Concerning the character and symbol recognition process, our strategy relies on the computation of Fourier Mellin invariants, allowing us to characterise multi-oriented and multi-scaled shapes. These Fourier Mellin invariants are introduced at the entry of a neural network for the classification process [20].

An illustration of the application of these techniques on a portion of an image is shown in Fig. 4.

In the context of this bottom-up approach, at each level of construction, some indicators are extracted form the data being constructed, in order to assign them a confidence rate with regard to their 'theoretical form'. If we consider low level primitives, this assignment will be performed by low level processing allowing us to check the pertinence of the

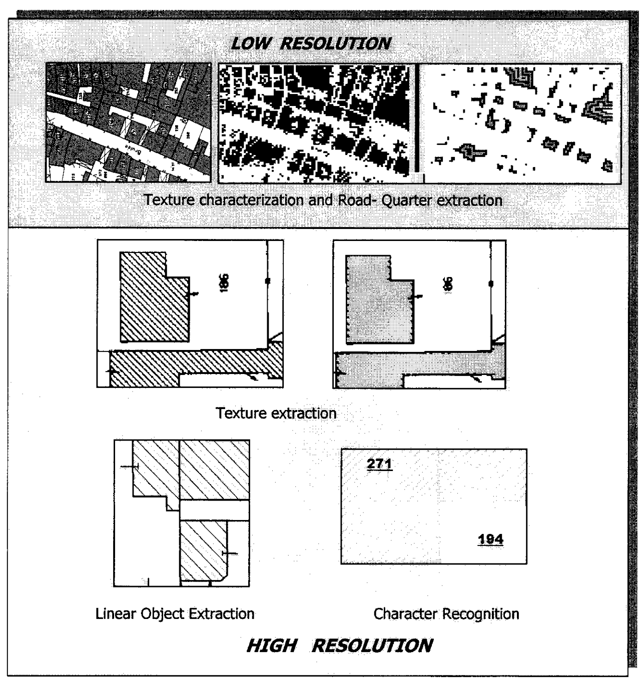


Fig. 4. Different stages of processing in the bottom-up approach.

extracted information with regard to the original data. If we consider high level data, the reference with which the constructed data will be compared will correspond to the model of the document.

In fact, each primitive is connected to a processing evaluation that assigns a confidence rate to each extracted primitive. In most cases, this pertinence of the primitive is estimated on the basis of a comparison between the geometrical object and the original image: the primitive is superimposed on the image in order to measure the matching rate, through the using of a distance similar to the Haussdorf distance [18]. The value of this estimation is only an indication for consistency evaluation, and must be considered with a certain amount of caution.

Once this basic scenario has been run, the extracted primitives make up the 0 level of the document model, and participate in the construction of the objects in the document. Necessarily, this initial approach leads to imperfections, mainly due to poorly extracted primitives.

3.3. Object Builder: Bottom-up Approach

A construction process is implemented for the different objects of the document model present on the map. A set of builders constructs the objects on each of the model levels as and when necessary. In our case, we have implemented a parcel builder (level 0 to level 1), a quarter builder (level 1 to level 2), and a road builder (level 2 to level 3). Each of these builders is located between two levels (upper and lower level), and must construct the upper level objects according to the primitives present on the lower level (level 1 for parcel construction).

The implementation strategy is that each module capable of building up objects consists of two parts: a set of rules for the construction of the upper level object; and an evaluation of the object thus created, enabling a consistency feature to be defined.

Object construction

Each builder develops the information from the primitives of the lower level taking into account the association rules and the different possible constitutions available in the model. These rules can be quite simple, as is the case for the quarter construction, or more complex, as is the case for the parcel construction.

In Table 3, the first row corresponds to the basic element which can be involved in the construction process. Among these elements, some can be indispensable, while some other can be optional in the construction process. The second row corresponds to the entities that can be associated with the elements of the first row. The third row corresponds to the rules that can be used in the association of the elements of the two first rows.

Figure 6 shows results from the construction process in the particular case of Fig. 5.

Consistency evaluation

Once these rules have been processed, it is possible to evaluate the construction by analysing those elements that must compulsorily be found in the object being built up. Each object is analysed in order to verify that all the indispensable elements for construction of the object have been found during the construction stage. Thus, a parcel with no identification will be declared inconsistent. If the object is declared consistent after analysis, the consistency is validated only if each of the lower level objects are also themselves consistent. Thus, a quarter will be declared inconsistent if one or several parcels that make it up are inconsistent (Fig. 7: Parcel 4 is inconsistent – Quarter 1 is then inconsistent). Thus, the consistency notion induces conformity of the objects, and of the elements that make them up. This is quite a strict criterion, on which it is possible to base our interpretation.

Table 4 gives some elementary rules to consistency evaluation adapted to object 'parcel'.

After the construction and consistency evaluation stages of all the objects present on the map, the control device

Table 3. Association rules examples: parcel construction

Object construction ('parcel' case): *see* Table 1.

Required graphic elements	**Associated entity**	
Outline	vectors – chain code	
Associated graphic elements	**Associated entity**	
Identifier	Characters, arrow	
Hatching	2D Texture	
Dashed Line	Linear Texture	
Boundary mark	Symbol	
Mail box number	Characters	
Out Going arrow	Symbol	
Parcel composition:		**Associated rules**
Outline		
Associated with	Identifier	included/shared
	Hatching	included
	Dashed line	shared
	Boundary mark	shared/included
	Mail box	neighbour
	Out going arrow	shared

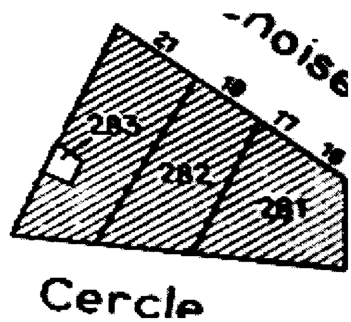


Fig. 5. Elementary example of a quarter.

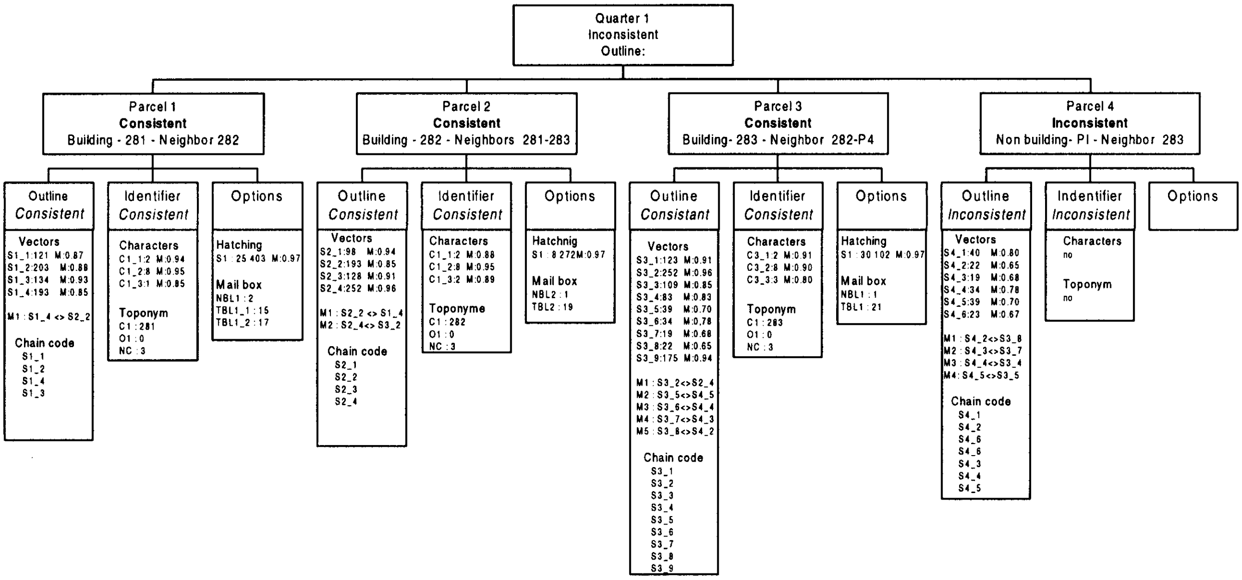


Fig. 6. Result on the elementary quarter.

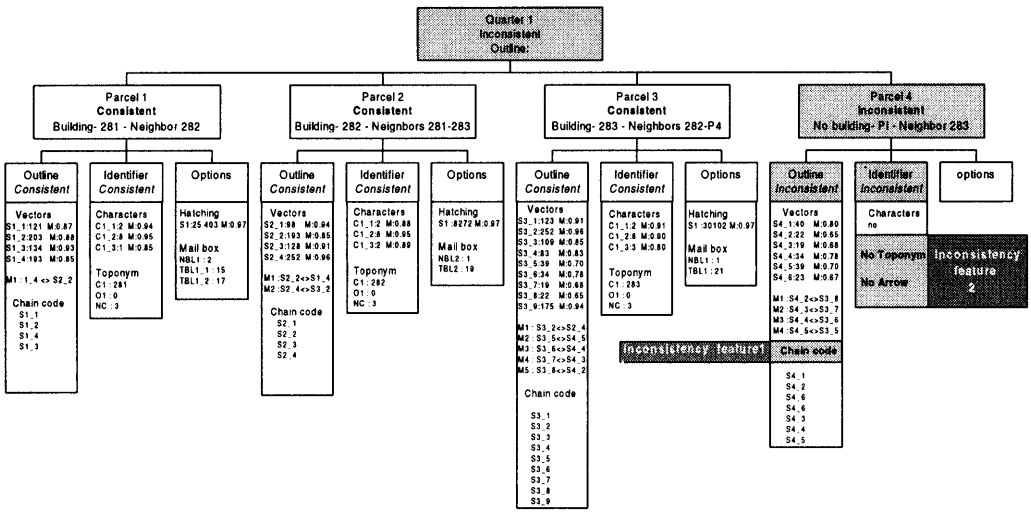


Fig. 7. Inconsistency criterion detection.

Table 4. Object consistency evaluation ('Parcel' case): *issued from Parcel model*

Graphic elements needed		needed	Complementary verification
Outline		1	incoming or out going vector
Identifier:			
	Characters (string)	⩾1	same orientation – no out going arrow
OR	Out going arrow	=1	no character

has some very precise information about the list of objects created, about their evaluation according to the model, and about their environment. By 'environment' we mean topology, which will be important when dealing with the influence of a given object on the consistency of neighbouring objects.

4. INCONSISTENCY MANAGEMENT PROCESS

As seen in Section 2, the control structure of the interpretation device is based on a continuous cycle (Fig. 1), composed of a bottom-up object construction stage and a top-down stage that tries to solve all of the interpretation problems encountered during the bottom-up approach. The basic principle of our document interpretation device is that a builder can be competent in a given context, but can also be completely unsuitable in another. The interpretation process is thus dictated by this approach which, in the bottom-up stage, proposes construction tools adapted to the greatest number of objects present on the document. Then, this approach allows for 're-training' of the builders associated with the objects considered as inconsistent, i.e. the objects for which the initial builder was not suitable. After a certain number of iterations, this cycle should produce an interpretation of all of the objects on the map, even if a human operator's intervention is sometimes required when the number of iterations is too high. This section describes the second stage of the interpretation based on inconsistency management.

The inconsistency management cycle processes the set of inconsistent objects. In a sequential process, each inconsistent object is localised and, according to the characteristics of the inconsistency detected, the control system proposes an alternative to the initial processing chain. This solution can either modify the parameters of the builder, or can use other builders present in the list associated with the inconsistency. Then, two situations are possible. Either the object is solved and the corresponding object is integrated in the consistent object list, or it is not solved and the control system can envisage another solution. After several attempts, if the control system has not succeeded in solving the inconsistency, the object is stored in a list containing all of the objects rejected during the interpretation stage. When such a situation occurs, the intervention of a human operator is triggered at the end of the interpretation cycle, so as to interpret this kind of object manually.

Figure 8 shows the global scheme of the inconsistency management process. This process is based on four stages for each inconsistent object; in fact, a classification of the inconsistencies is performed, depending on their features. This classification is performed through a classical decision tree. At each of the resulting inconsistency class there is a set of remedial solutions, allowing us to solve the problems encountered by adapting the process to the context of each object.

For each inconsistent object, the inconsistency management consists of:

- Stage 1: inconsistency characterisation. This stage allows us to classify each inconsistent object as a function of its features (Table 5). This classification is based upon the analysis of the graphic elements that are associated with the object.
- Stage 2: the class related to this inconsistency is selected. This class is associated with an ordered list, including 'remedial solutions'. These remedial solutions consist of alternative processing chains capable of solving the inconsistency.
- Stage 3: the first remedial solution is applied to the portion of an image that contains the inconsistent object and the neighbouring objects, even if these are considered as being consistent. Indeed, an object can be consistent even if it still contains a primitive which belongs to a neighbouring object. For instance, parcel 4 in Fig. 5 and Fig. 6 is inconsistent. The portion of the image which is processed is the one containing parcels 3 and 4.
- Stage 4: if the inconsistency is solved, the corresponding object is integrated into the consistent object list. Some rules have been applied to validate this solution as a function of the consistency of the processed object, and the evolution of neighbouring object consistency. If the

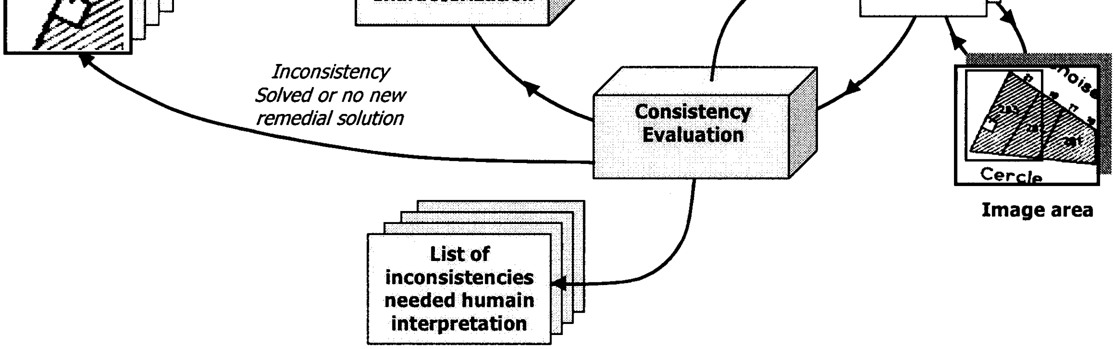


Fig. 8. Inconsistency management.

Table 5. Inconsistency classification

Class Number		
1	**Inconsistency features**	No Identifier, No incoming vector
	ClassicOrigin of Inconsistency	Stroke of hatched area confused with parcel outline **OR** Character occlusions when character is connected to outline
	Remedial solutions	1- Parameters relaxation on Hatched extractor 2- Detection of characters connected to the outline
	Frequency	High
2	**Inconsistency features**	No Identifier, many incoming vectors
	ClassicOrigin of Inconsistency	Part of hatched area not extracted (variability) **OR** character is connected to outline
	Remedial solutions	1- Parameters relaxation on Hatched extractor **2-** Complementary Hatched area extractor algorithm 3- Detection of characters connected to the outline
	Frequency	Low
3	**Inconsistency features**	Identifier, one incoming vector
	ClassicOrigin of Inconsistency	Character connected to parcel outline **OR** Arrow damaged during hatched area extraction
	Remedial solutions	1- Detection of characters connected to the outline **2-** Arrow pattern matching without hatched area extraction **3-** Parameters relaxation on Hatched extractor
	Frequency	Medium
4	**Inconsistency features**	Identifier, included vectors
	ClassicOrigin of Inconsistency	Character connected together **OR** Symbols included in parcel **OR** Part of hatched area not extracted (variability)
	Remedial solutions	1- Detection of characters connected together **2-** Detection of symbols included in parcel **3-** Parameters relaxation on Hatched extractor
	Frequency	Medium
...	...	...

inconsistency is not solved, the next remedial solution in the ordered list is applied on the part of the image.

The following sections provide some details on each part of this inconsistency management process.

4.1. Characterisation of Inconsistencies

The different inconsistencies have been classified on the basis of a set of inconsistency features and a set of tests developed in our laboratory. The classification characterises the interpretation mistakes detected so that the optimal processing for solving each of the inconsistencies can be proposed. At first, this classification operation was entirely supervised, and consisted of detecting the most frequent inconsistencies and characterising them as a function of the erroneous primitives (inconsistency indications), and as a function of the document model. From the indications, several inconsistency classes have been proposed, on the basis of one class for each indication set. Thus, two objects which contain two different inconsistencies, but are characterised by the same inconsistency indications, will be stored in the same processing class: the same operators will solve the inconsistencies for both objects. Thus, in this processing class are found the chains adapted to the first inconsistency, as well as those adapted to the second one. Figure 7 shows some inconsistency indications for parcel 4 of Fig. 5. In this example, the detected inconsistency on parcel 4 is linked with the ‘outline chain’ and ‘Toponym’ indications. Table 5 gives some information about the different inconsistency classes based on inconsistency features.

This first solution remains quite simple and is not optimal; it cannot process all types of inconsistency satisfactorily. Our current research deals with the implementation of an unsupervised and modifiable inconsistency classification, not yet available in this version of our device.

4.2. List of Builders Associated with Inconsistency Classes

For each kind of inconsistency in the previously presented classification, the control system contains a set of builders adapted to the inconsistency to be solved. The builder used during the bottom-up phase of the initial processing chain is put at the top of a list of substitution builders for each inconsistency class at process start-up. The first solution consists of modifying the parameters of the operators (if possible), within a predefined range, so that they are as well-adapted as possible to the local specificity. However, in a certain number of cases, the modification is insufficient and the inconsistency cannot be solved. In such cases, human expertise can be introduced into the device in order to enrich the builder’s substitution lists, and for a given builder the expert can propose a new set of substitution tools (for instance, replacing one skeletonisation algorithm with another), or a new substitution resolving method (modification of the sequential ordering of the tools, or a proposition for a completely different construction approach from the initial one).

At this stage, the builders' list is arranged so that the control system can give priority to the builder or builders most suitable for solving a particular inconsistency. To implement this strategy, we have defined an efficiency criterion for each builder, this criterion being used to characterise its suitability for the resolution of one inconsistency. For each builder, computation of the efficiency criterion is defined as a function of the number of inconsistency solving successes and the number of tests processed.

Each operator is evaluated according to this efficiency criterion each time it attempts to resolve an inconsistency. The device can then propose the operator list that frequently solves the inconsistencies in one considered class.

Table 5 shows some inconsistency classes (12 classes in our system). For each class, we present the inconsistency features which determine the class number, the usual situation in which each inconsistency occurs, the frequency of the inconsistency, and the main process which is involved in the remedial solution.

4.3. Inconsistency Validation

The last stage of inconsistency management consists in validating the consistency. At the end of processing on the portion of the image considered, several criteria permit the inconsistency to be solved.

First, just before the inconsistency management, the consistencies are characterised. All the objects which are consistent according to the model are defined as being 'internally' consistent with regard to the interpretation. This means that they contain all the intrinsic features for the interpretation. The 'internally' consistent objects whose neighbouring objects are also internally consistent in turn are particular objects for which the interpretation is validated regionally. As a consequence, these objects are considered to be reliably interpreted, so will be difficult to modify during an inconsistency management cycle. They are 'externally' consistent.

At the end of inconsistency management, when an inconsistency object is processed in a zone, the consistency of this object is analysed as a function of the evolution of the neighbouring consistency. This analysis is based on elementary rules (Table 6). The results of this analysis will determine whether the processing chain is an adequate mechanism for solving inconsistencies.

Table 6. Validation table for the inconsistency solving (* = after and before inconsistency management)

Neighbouring objects		External consistency	Validation
Before (*)	*After* (*)		
Consistent	Consistent	Not modified	Yes
Inconsistent	Consistent	Not modified	Yes
Consistent	Inconsistent	Not modified	Yes
Inconsistent	Inconsistent	Not modified	No
Consistent	Consistent	Modified	No
Inconsistent	Consistent	Modified	No
Consistent	Inconsistent	Modified	No
Inconsistent	Inconsistent	Modified	No

The results are definitively validated only if the consistency of the neighbouring objects is not affected, so the result of a local interpretation will be retained only if it improves the overall document interpretation.

Let us note, finally, that a sequential ordering of different remedial solutions is still possible for inconsistency solving. In each case, the global validation of the remedial solutions becomes effective only if a global improvement in the interpretation of the portion of the processed image can be established.

4.4. New Builder

This system permits us to easily introduce any new builder or algorithm in order to improve inconsistency management. When a new builder is introduced in the chain, the expert can either insert it into a precise inconsistency class, or insert it automatically into all of the classes containing the same kind of builder. The control device will directly associate a maximum efficiency criterion to this builder, which will be used as soon as an inconsistency is detected in the class considered.

When a new builder is consulted, its efficiency is rapidly assessed in comparison with the other solutions. The builder's efficiency enables the control device to optimise inconsistency solving.

4.5. Human Operator

As explained above, at each moment, the control device constantly updates the 'state' of the interpretation of the document. It also contains a list of objects that are inconsistent in the interpretation. According to the objectives given to the device, the human operator can take three actions, depending on whether he wants to obtain a rapid interpretation with no inconsistency solving, or allows an autonomous management of the inconsistencies by the system:

- Action 1: the human operator can intervene just after the bottom-up stage, when there are three confidence degrees for the interpretation of the objects:
  - *Degree 2*: consistent object whose neighbours are also consistent (external).
  - *Degree 1*: consistent object with one or more inconsistent neighbors (internal).
  - *Degree 0*: inconsistent object.

  The objects in degree 2 are the most reliable, since the device has not detected any inconsistency on the object or on its neighbours. Thus, the human operator must operate a correction session on the 0 degree objects, while also verifying the objects of degree, 1 since they are neighbours.
- Action 2: the human operator decides on the degree of autonomy to be given to the control system in solving the inconsistencies. This entails defining how many cycles

will be permitted for each inconsistency. A limited number of cycles means a limited autonomy for the system in resolving the inconsistencies.

- Action 3: the third action available consists in enriching the builder base associated with the resolution of the inconsistencies. An expertise session provides the device with tools and complementary methods encapsulated in the integrated builder list. If the user introduced an inadequate builder, it would very quickly be removed from the head of the list, since its efficiency would be very poor. The device tolerates some diagnostic mistakes of this nature without jeopardizing the strategy of the control system. The only consequence would be a slight time penalty.

The user is, therefore, completely integrated in the interpretation cycle, either providing the device with his expertise, or correcting errors in the data by giving a certain degree of autonomy to the system.

5. IMPLEMENTATION

The approach that we have presented has been implemented in a software system for French cadastral map interpretation, called 'SYRADOC'. This software was written in the C language, and has been tested on a set of 10 urban cadastral maps (urban maps were chosen because they are high density maps). The scale of the maps is 1/500, and they contain about 1900 parcels and 112 quarters. The digitisation was performed using a B/W scanner with a 400 dpi resolution.

All the maps were hand-drawn by cadastral agents under precise drawing rule constraints (which correspond to the legend). The quality of the maps is very variable, since some of them could be taken for computer assisted drawings, while others are very noisy with a lot of intrinsic variability. Figure 9 shows the variability of the processed representations. All of these experiments were performed on a Sun Sparc 10, equipped with 64 Mb of RAM.

We propose to present the results of this first implementation, and justify them in relation to the previously presented concepts (Fig. 10(a) and 10(b)). Then, we provide a quantitative evaluation for our extraction tools, by using the inconsistent object processing cycles. This kind of evaluation is similar to following work [5].

5.1. Bottom-up Approach

The bottom-up approach described in this document uses a set of builders and low level operators from the literature [21,22], as well as our team's specific research tools [7,20,23].

The tools for hatched area extraction, character segmentation and recognition, and linear object processing, as well as specific symbols (arrows, boundary markers, and so on) have been developed in the context of this application. Object construction tools have also been developed, based on a cadastral document model, which is described explicitly in the context files of the device.

In comparing the different tests performed, it is interesting to note that 35% of the low level primitives recognition mistakes occur when the drawing rules have not been respected by the cadastral agent: different slopes for the hatching in the same quarter; overlapping between the arrows and the parcel numbers; connection of the parcel numbers to the parcel outlines, or to the hatched strokes, etc.

Many of the remaining 65% of mistakes are due to the primitives extraction tools which could not be validated because of particular features: strokes too short to be classified as hatching strokes, segments that do not penetrate far enough inside a parcel to be counted as arrows, and so on.

However, these quite restrictive constraints ensure that the extracted primitives are accurate and reliable, since such stringent parameters have been used for the classification. This primitive consolidation principle is applied to all of the primitives, and has provided the results presented in Table 7.

Globally, the results presented provide quite an interesting set of primitives for object construction. These results will be improved by the inconsistency management cycle.

5.2. Object and Inconsistency

At the end of the inconsistency management cycle, the object builders are activated in order to propose a first

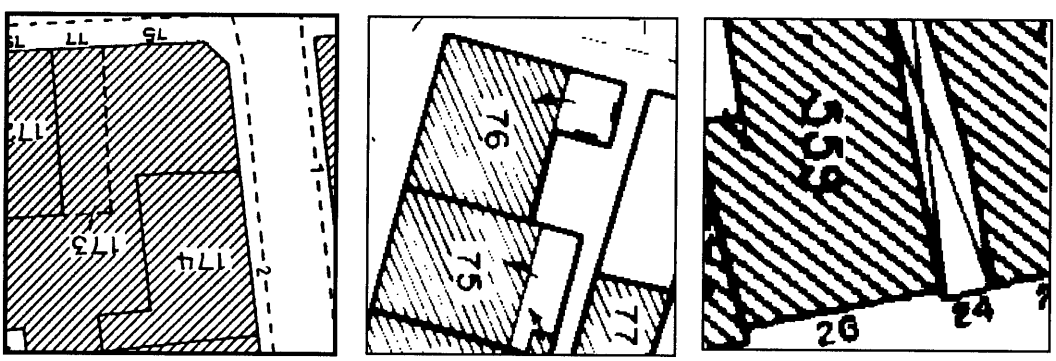
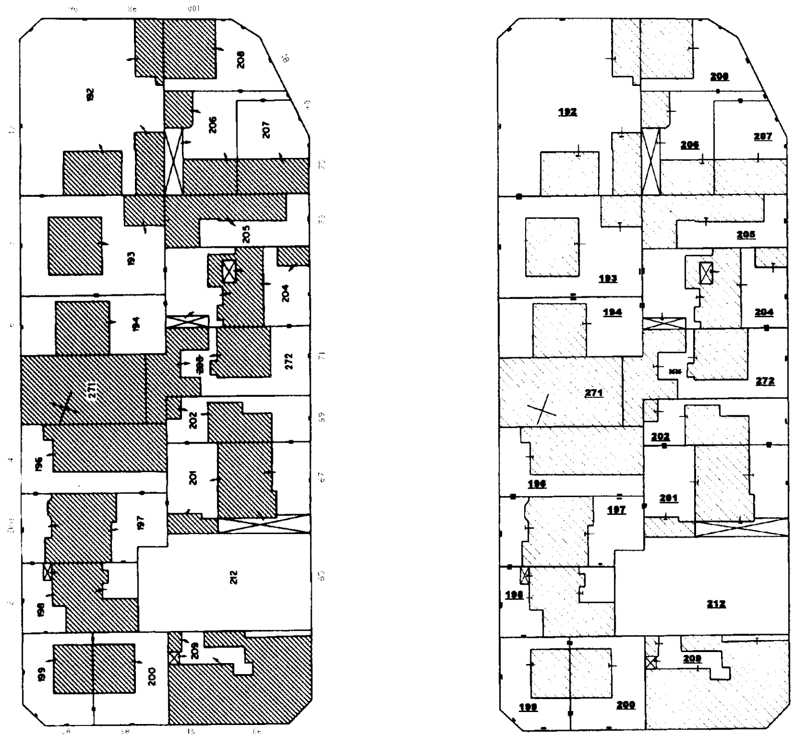


Fig. 9. Variability of document representation.



(a) (b)

Fig. 10. (a) Original image, (b) image built from interpreted data.

Table 7. Confusion matrix

Identified	Hatching strokes	Vectors (%)	Arrows (%)	Symbols (%)
Hatching	97.6	1.56	1.1	0.18
Vectors	2.3	98.12	4.2	0.26
Arrows	0.02	0.10	91.6	2.12
Symbols	0.04	0.22	3.1	97.44

Table 8. Parcel interpretation after the bottom-up stage

Number	Present on the document	Identified as consistent	Identified as inconsistent
Buildings	912	525	387
Gardens/Yards	952	699	253
Total	1864	1224	640
Percentage	100%	65.7%	34.3%

interpretation of the document. We have mainly considered the parcel and quarter objects, which represent the richest and the most complex information source.

Two kinds of builders are implemented, a parcel builder and a quarter builder, based on the proposition developed previously. The primitives considered are deliberately limited to such essential ones as hatched areas, strokes, characters and arrows, for example.

After the processing of the cadastral maps, 66% of the parcels are consistent with regard to the model. On the other hand, the consistency rate of the quarters is very weak, since in each quarter, one or several parcels are frequently inconsistent (8% of consistent quarters after the Bottom-up phase). Table 8 shows the result of this construction.

These rates must be examined in the light of the very stringent parameters used for the extractors. One of the tasks of the inconsistency management phase is to relax the constraints on the extractors, within a supervised framework.

Finally, a strong point of our approach is that, on the different tests, 65.87% of the parcels were interpreted without any mistakes, i.e. no parcel was declared consistent if it carried an erroneous primitive. The robustness principle shows its main advantage here. At the end of this construction, a list of inconsistent objects is created for the control device. The appropriate set of successive cycles is carried out on each inconsistency, according to its classification, with a maximum of 10 iteration cycles.

The supervised inconsistency classification has shown that 12 classes can be considered, taking into account the most frequently-met configurations. Class '12' characterises the 'waste basket inconsistency set', for which, so far, no automatic process can interpret the objects. This corresponds, therefore, to the 'human operator processing set', which is

Table 9. Parcel interpretation after consistency solving cycle

	Present on the document	Consistent after bottom-up sequence	Consistent after inconsistency management	Total consistent object
Buildings	912	525	239	764
Gardens/Yards	952	699	70	769
Total	1864	1224	309	1533
Percentage	100%	65.7%	16.6%	82.3%

triggered after recycling. For each of the other classes, the proposed treatments consist in varying the parameters of the initial processing chain, and for eight of the 12 classes, in proposing an alternative treatment, with different processing algorithms (skeletonisation, vectorisation process, hatched area extraction, etc.). The following results show the improvements obtained as a result of this technique.

Table 9 highlights the advantages of the recycling approach, which improves the interpretation rate of the cadastral objects by 17%, while the very important zero mistake rate is retained. Indeed, at the end of the recycling stage, no inconsistent object has been considered as consistent by the device. An analysis of the low level primitives from this recycling stage shows how useful the approach is, since the recognition rate increases by 1% for the vectors, 3.4% for the arrows and 0.5% for the characters (cf. Table 7 and Table 10).

The application of these treatments to a cadastral map requires from 1H30 to 2H00 CPU; the extent to which the human operator intervenes depends upon the complexity of the map, and on the number of inconsistencies to be processed. This human correction work is, however, guided by the device, since the operator intervenes only on zones which are highlighted by the device, the rest of the map being completely reliable (0% mistake rate). Thus, for this correction phase, the human operator can simply let himself be guided by the device when processing his intervention.

For the tests performed, no correction has been required outside of the zones proposed by the system, which confirms the reliability of the interpreted data.

6. RELATED WORKS

In this section, we do not intend to give an exhaustive state-of-the-art review concerning the different tools available for research purposes. However, we propose to highlight here the techniques and tools that appear to be 'mature', as well as the 'hard points' on which the research must still progress, in order to offer more autonomous and efficient interpretation systems.

6.1. Technical Documents Analysis

The research teams propose some wider-ranging solutions, using techniques ranging from image processing to high level interpretation processes based on distributed devices using artificial intelligent concepts. These studies are much more advanced than commercial systems, and generally remain targeted on one type or class of document. From this point of view, the literature is abundant and deals with primitives extraction and recognition, and also with more global devices.

The first of these two points represents the richest bibliographic source, since many authors propose algorithmic solutions for the extraction and recognition of graphical entities (primitives). Even if some authors propose some solutions based on grey level image acquisition [8,24,25,26], the extraction and recognition process of these graphical entities can be applied to black and white images. The processing of grey level images allows particular supports such as noisy or deteriorated documents or calques to be interpreted. Indeed, for this kind of document, the thresholding techniques that are directly integrated into the digitisation device (scanner) do not produce images of a quality sufficient to allow reliable interpretation. On this point, Sahoo [27] proposes a good state-of-the-art review of the thresholding technique in a general context. Concerning the processing of graphical entities, many bibliographic references have been presented since 1980. These papers deal with the separation of different graphical entities [28], the vectorisation of linear objects [12,15,29], the segmentation of

Table 10. Confusion Matrix after consistency solving cycle

Identified	Hatching strokes %	Vectors %	Arrows %	Symbols %
Hatching	99.44	0.54	0.4	0.12
Vectors	0.51	99.19	2.1	0.1
Arrows	0.02	0.08	95	1.8
Symbols	0.03	0.19	2.5	97.98

complex objects such as arcs and circles [30–32], dashed lines [33] and textures [2,7,12]. Many of these segmentation techniques can be considered as mature, as Tombre indicates [21]. Nevertheless, the most recent literature regularly proposes some relevant improvements dealing with residual problems [34]: dashed lines [35,36], vectorisation [3,16,37], dimension sets [13,38], arcs and circles [14,39], and global systems to convert scanned engineering drawings into vectorised files [4,40].

On symbol and character recognition, many works can be found in the literature [1,41], some of them processing classical problems (structured documents), while others deal with specific constraints: maps, industrial documents, and so on. However, some aspects of this pattern recognition problem are still the object of intensive research activities because of their specificity: multi-oriented and multi-scaled characters [20,23,42,43], connected characters [20,42,44], connected symbols [6], and geometric features identification [45]. This kind of problem is crucial, since characters and symbols carry very important information about the contents of a document: poor detection/recognition of such patterns leads to some very important problems in the interpretation process. The objective of this kind of approach is to progressively associate graphical primitives and characters, in order to construct higher level objects, the semantic representation of which is similar to that supplied by the document drawer. Next, these primitives and objects are integrated into a reconstruction process in order to build information of a higher level, depending on explicit or implicit rules.

Most of the proposed systems use strictly bottom-up approaches, based on a pre-fixed construction scenario using a low level indication extracted from the image [2,10,12,46–48]. These approaches are based on the robustness of the primitives extractors, and on an exhaustive explicit representation of the knowledge, transcribed into a set of construction methods for the interpretation.

This knowledge representation can be implemented in the form of a set of rules, either in the form of a description language [49–51], or directly integrated in the 'source code' of the device [9,10,52,53]. The interpretation approach is mainly based on a successive stacking of the extracted information. As a consequence, poor extraction/recognition of information creates unavoidable mistakes in the interpretation process. This remark highlights the main limit of these systems. In our opinion, the most advanced work in the field of technical interpretation devices is that proposed by Joseph and Pridmore [8] and den Hartog et al [5].

The ANON [8] system is based on the 'cycle of perception' proposed by Neisser [54]. This system is structured in three layers in order to separate spatial from symbolic processing. The first is composed of a large image analysis library associated to both search-tracking functions and management processes. The information extraction is adapted to the context by the second level 'schema' (prototypical drawing construct), which receives the entities from the lower layer, and interprets the result as a function of the current schema. A cycle of hypothesis-verification is thus proposed by the schema to the control system (highest layer). This control system analyses the proposition as a function of the current state of the proposed schema, and modifies it if necessary. The knowledge directed image analysis and the construction cycle according to the context are two interesting concepts, applied on 15 different schema classes.

More recently, Den Hartog et al [5] proposed a mixed approach based on a top-down control mechanism associated with bottom-up object recognition. The system decomposes the binary image into primitives (and not vectors) to obtain a good morphological representation of the information, and uses template matching to recognise each object. Then, contextual reasoning is performed, based on a loop including inconsistency detection and search action generation, in the Region of Interest (ROI). The control system defines an ordered search action list to search for specific object types in the ROI. Priorities are specified by the user to define the most important search actions, to assign a priority to the relationship between objects. A test of consistency is applied on each recognised object in order to verify the hypothesis defined at the system's top level, as a function of the knowledge of the object to be recognised. On the other hand, the knowledge framework of the device is essentially based on spatial relationships between primitives, without integrating and describing hierarchical relationships. In the case of particularly complex documents, this kind of system is penalised because of the drastically increasing number of relationships and the necessity to generate new search actions for the 'constructed objects'.

These enhanced approaches link the primitives extraction with an interpretation strategy by adapting the construction process to the local context and to the emitted hypotheses. These research axes appear to be very promising for the automatic interpretation of documents. There are two good state-of-the-art reviews of the different available techniques and tools for the interpretation of technical and cartographic documents [34,55].

6.2. Comparative Discussion

An objective comparison of our interpretation device with other systems is quite difficult to establish, because of the variability of situations which are considered.

From the consistency analysis point of view, from the hypothesis emission/validation point of view, and from the methodological point of view, one can say that our contribution could be categorised with several other contributions [5,8]. In any case, a comparative discussion should be based on an objective performance evaluation of our system. Actually, performance evaluation of algorithms or systems is still a crucial problem for the document analysis working community. In fact, in the literature, only the low level operators have been the object of a comparison attempt. Indeed, Phillips et al [56] propose a protocol allowing us to evaluate low primitives extraction. Complete interpretation systems are more difficult to compare, each author basing his own evaluation on his application. For example, let's consider the MARIS device [9], which allows us to interpret a Japanese cadaster. This system is based on a strictly bottom-up approach, relying on a set of sequential processes.

An ergonomic and interactive human interface allows one to correct recognition errors and to process unrecognised layers. The authors claim that the MARIS system allows them to improve the global conversion process by 25% (in comparison with a complete manual process). The claimed recognition error rate is about 15% for the buildings, 9% for the roads, and 3% for the lines, without considering localisation indicators. In the same manner, Boatto [10] proposes an interpretation system adapted to an Italian cadaster. The process is almost the same as MARIS, and does not integrate inconsistency detection. The stated recognition rates are excellent (buildings: 91.4%, characters: 92.5%), and may be due to the exceptional quality of the processed documents. As far as we know, den Hartog et al [5] is probably the most sophisticated system integrating inconsistency detection. The author proposes to evaluate low level primitives extraction and object recognition. The automatic conversion process rejects 20% of objects on two tested images comprising approximately 3.4 m of drawn pipeline, which is equivalent to 1700 m of pipeline. Nevertheless, the author does not propose inconsistency object management in order to decrease the misclassification rate, and highlights the fact that poor segmentation of the grey level images introduces a lot of rejects. Finally, Langrana et al [45] and Yu et al [6] propose an object construction process based on a combination of primitives proposed by the vectorisation stage. These systems consist in converting mechanical documents, and do not consider the recognition rate associated with the global process. This bibliographic synthesis, the absence of a test image database, and the lack of an objective evaluation criterion make a comparison of our system with other devices very difficult, or even impossible at this time.

7. CONCLUSION

The device that we have proposed in this paper relies on a set of innovative concepts, the aim of which is to render the system as autonomous and reliable as possible.

The first experiment that we carried out on a set of French cadastral maps highlights the interests of our approach: indeed, this first version of the device has shown the limits of a strictly bottom-up approach, since only 65.7% of the objects were considered as consistent after a first construction stage. The advantages of the interpretation cycle were also highlighted in the improvement of the classification rate corresponding to the consistent objects (from 65.7% to 82.3%). These elements constitute important factors if we consider the autonomy of the device and its capacity to evaluate its own results. However, from the implementation point of view, the most important asset of the device lies in the zero mistake rate of the conversion process, since no object has been badly interpreted. This characteristic is fundamental, since it proves the robustness of the device. Nevertheless, the interpretation system needs to be validated on a larger set of samples, and on different categories of documents.

The implementation of the device has permitted us to define different directions for more detailed investigations. Further research must deal with specific points of the device, such as unsupervised inconsistency classification, the introduction of new operators after feedback from the human operator inconsistency management, and also multi-inconsistency management. From the low level processing point of view, some current developments in our laboratory are trying to integrate some consistency indicators as soon as possible in the processing chain. Among them, one can cite vectorisation performance evaluation or a fuzzy character recognition process. Our research perspectives also deal with essential points such as adapting the approach to other kinds of maps and charts. On this last point, research studies are currently in progress to measure the limits of our approach and, although the validity of the system itself is not in question, the implementation of a generic and adaptable expertise representation is causing problems which have yet to be resolved. We consider that this key point will strongly influence the evolution of research into document analysis.

Finally, our research activities also consider the knowledge modelling problem. This point aims at categorising the different knowledge which is involved in an interpretation process, in order to find a generic modelling structure, adapted to each kind of knowledge. In our opinion, this kind of reflection (which is similar in general vision problems) should permit the construction of dynamically adaptable systems.

References

1. Filipski AJ, Flandrena R. Automated conversion of engineering drawings to CAD form. Proc IEEE 1992; 80(7):1195–1209
2. Antoine D. CIPLAN: A model-based system with original features for understanding French Plats. Proc ICDAR '91, St Malo, France, 1991; 647–655
3. Janssen R, Vossepoel A. Adaptive vectorization of line drawing images. Computer Vision and Image Understanding 1997; 65(1): 38–56
4. Wenyin L, Dori D. A generic integrated line detection algorithm and its object-process specification. Computer Vision and Image Understanding 1998; 70(3):420–437
5. den Hartog JE, ten Kate TK, Gerbrands JJ. Knowledge-based interpretation of utility maps. Computer Vision and Image Understanding 1996; 63(1):105–117
6. Yu Y, Samal A, Seth S. Isolating symbols from connected lines in a class of engineering drawings. Pattern Recognition 1994; 27(3):391–404
7. Ogier JM, Mullot R, Labiche J, Lecourtier Y. Attribute extraction for French map interpretation. Proc ICDAR '93, Tsukuba Science City, Japan, 1993; 672–675
8. Joseph SH, Pridmore P. Knowledge-directed interpretation of line drawing images. IEEE Trans PAMI 1992; 14(9):928–940
9. Suzuki S, Yamada T. MARIS: Map Recognition Inpute System. Pattern Recognition 1990; 23(8):919–933
10. Boatto L et al. An interpretation system for land register maps. IEEE Computer 1992; 25(7):25–33
11. Chhabra AK. Graphic Symbol Recognition: An Overview. Lecture Notes in Computer Science 1389, 1998; 68–79
12. Kasturi R et al. A system for interpretation of line drawing. IEEE Trans PAMI 1990; 12(10):978–992

13. Lai CP, Kasturi R. Detection of dimension sets in engineering drawings. IEEE Trans PAMI 1994; 16(8):848–855
14. Dori D. Vector-based arc segmentation in the machine drawing understanding system environment. IEEE Trans PAMI 1995; 17(11):1057–1068
15. Lam L, Lee SW, Suen CY. Thinning methodologies – A comprehensive survey. IEEE Trans PAMI 1992; 14(9):869–885
16. Di Zenzo S, Cinque L, Levialdi S. Run-based algorithms for binary image analysis and processing. IEEE Trans PAMI 1996; 18(1):83–89
17. Ogier JM, Mullot R, Labiche J, Lecourtier Y. French map interpretation: a sound approach. Proc IEEE-SMC'93, Le Touquet, France, 1993; 3:452–458
18. Huttenlocher DP, Klanderman G, Rucklidge J. Comparing images using the Haussdorf distance. IEEE Trans PAMI 1993; 15(9):850–863
19. Ogier JM, Mullot R, Labiche J, Lecourtier Y. Multilevel approach and distributed consistency for technical map interpretation: Application to cadastral maps. Computer Vision and Image Understanding 1998; 70(3):438–451
20. Cariou C, Ogier JM, Mullot R, Gardes J, Lecourtier Y. An original algorithm for the classification of multi-oriented shapes – application to character recognition on technical maps. IJPRAI 1999; 13(7):1201–1218
21. Tombre K. Analysis of drawings: state of the art and challenges. Lecture Notes in Computer Science 1389, 1998; 257–264
22. Kasturi R, Raman R, Chennubhotla C, O'Gorman L. An overview of techniques for graphics recognition. In: Structured Document Analysis (HS Baird, H Bunke, K Yamamoto, eds.), Springer-Verlag, Berlin/New York, 1992; 285–324
23. Ogier JM, Cariou C, Mullot R, Gardes J, Lecourtier Y. Interpretation of technical document: Application to French telephonic network. Proc SCI'98, Orlando, FL, July 1998; 12–14
24. Trier OD, Taxt T. Evaluation of binarization methods of document images. IEEE Trans PAMI 1995; 17(3):312–315
25. den Hartog J, ten Kate T, Gerbrands J. Knowledge-Based Segmentation for Automatic Map Interpretation. Lecture Notes in Computer Science 1072, Springer-Verlag, 1996; 159–178
26. Kamel M, Zhao A. Extraction of binary character/graphics images from grayscale document images. CVGIP 1993; 55(3): 202–217
27. Sahoo PK et al. A survey of thresholding techniques. CVGIP 1998; 41(2):233–260
28. Fletcher LA, Kasturi R. A robust algorithm for text string separation from mixed text/graphics images. IEEE Trans PAMI 1988; 10(6):910–918
29. O'Gorman L, Kasturi R. Executive Briefing – Document Image Analysis. IEEE Press, 1997
30. Liao CW, Huang JS. Stroke segmentation by Bernstein–Bezier curve fitting. Pattern Recognition 1990; 23(5):475–484
31. Davies ER. A modified Hough scheme for general circle location. Pattern Recognition Letters 1988; 7:37–43
32. Thomas SM, Chan YT. A simple approach for the estimation of circular arc center and its radius. CVGIP 1989; 45:362–370
33. Lai C, Kasturi R. Detection of dashed lines in engineering drawings and maps. Proc ICDAR'91, St Malo, France, 1991; 507–514
34. Tombre K, Chhabra AK (eds). Graphics Recognition – Algorithms and Systems. Lecture Notes in Computer Science 1389, Springer-Verlag, 1998
35. Dori D, Wenyin L, Peleg M. How to Win a Dashed Line Detection contest. In: Lecture Notes in Computer Sciences 1072, 1996; 286–300
36. Kong B, Phillips IT, Haralick RM, Prasad A, Kasturi R. A Benchmark: Performance Evaluation of Dashed-Line Detection Algorithms. Lecture Notes in Computer Sciences 1072, 1996; 270–285
37. Dori D, Liu W. Sparse pixel vectorization: an algorithm and its performance evaluation. IEEE Trans PAMI 1999; 21(3): 202–215
38. Das AK, Langrana NA. Recognition and integration sets in vectorized engineering drawings. Computer Vision and Image Understanding 1997; 68(1):90–108
39. Wenyin L, Dori D. Incremental arc segmentation algorithm and its evaluation. IEEE Trans PAMI 1998; 20(4):424–431
40. Chen Y, Langrana NA, Das AK. Perfecting vectorized mechanical drawings. Computer Vision and Image Understanding 1996; 63(2):273–286
41. Mori S, Suen CY, Yamamoto K. Historical review of OCR research and development. Proc IEEE 1992; 80(7):1029–1058
42. Deseilligny MP, Le Men H, Stamon G. Character string recognition on maps, a rotation-invariant recognition method. Pattern Recognition Letters 1995; 16:1297–1310
43. Trier OD, Taxt T, Jain AK. Features extraction methods for character recognition – a survey. Pattern Recognition 1996; 29(4):641–662
44. Trier OD, Taxt T, Jain AK. Data capture from maps based on gray scale topologic analysis. Proc ICDAR'95, Montreal, Canada, 1995; 923–926
45. Langrana NA, Chen Y, Das AK. Feature identification from vectorized mechanical drawings. Computer Vision and Image Understanding 1997; 68(2):127–145
46. Shimotsuji S, Hori O, Asano M, Suzuki K, Hoshino F, Ishii T. A robust recognition system for a drawing superimposed on a map. IEEE Computer 1992; 25(7):56–64
47. Dori D, Wenyin L. Automated CAD conversion with the machine drawing understanding system: concepts, algorithms, and performances. IEEE Trans SMC – Part A 1999; 29(4): 411–416
48. Ah-Soon C, Tombre K. Variations on the analysis and recognition. Proc ICDAR'97, Ulm, Germany, 1997; 347–351
49. Pasternak B. The role of taxonomy in drawing interpretation. Proc ICDAR'95, Montreal, Canada, 1995; 799–802
50. Dori D. A syntactic/geometric approach to recognition of dimensions in engineering machine drawings. CVGIP 1989; 47: 271–291
51. Goodson KJ, Lewis PH. A knowledge based line recognition system. Pattern Recognition 1990; 11:295–304
52. Antoine D, Collin S, Tombre K. Analysis of technical documents: The REDRAW system. In: Structured Document Analysis (HS Baird, H Bunke, K Yamamoto, eds), Springer-Verlag, Berlin, 1992; 385–402
53. Deseilligny MP, Mariani R, Labiche J, Mullot R. Topographic maps automatic interpretation: Some proposed strategies. Lecture Notes in Computer Sciences 1389, Springer-Verlag, 1998; 175–193
54. Neisser U. Cognition and Reality. Freeman, New York, 1976
55. Kasturi R, Tombre K (editors). Graphics Recognition – Methods and Applications. Lecture Notes in Computer Science 1072, Springer-Verlag, 1996
56. Phillips IT, Liang J, Chhabra AK, Haralick R. A performance evaluation protocol for graphics recognition systems. In K Tombre, AK Chhabra (eds), Lecture Notes in Computer Science 1389, Springer-Verlag, 1998

Jean-Marc Ogier was born in 1967 in Neufchatel en bray, France. He received his PhD degree in computer science from the University of Rouen, France, in 1994. During this period (1991–1994), he worked on cadastral map interpretation. From 1994 to 1998, he was an associate professor at the Institute of Technology of Lannion, France. Since 1998, he has been an associate professor in computer science at the University of Rouen. Dr Ogier's present research interests are relative to image interpretation systems, including low level operators (segmentation, filtering, etc.), pattern recognition, and high level processes issued

from artificial intelligence (knowledge modelling, multi-agents systems, etc.). Dr Ogier is a member of the GRCE (the French Research Group dealing with document analysis and recognition), and is also member of the IAPR Technical Committee on Graphics Recognition (TC10).

Rémy Mullot was born in 1963 in Rouen, France. He received his Master of Sciences and Technology degree from the University of Savoie in 1986. He also received his PhD degree from the university of Rouen in 1991. Since 1991, he has been associate professor of electrical engineering and computer science at the University of Rouen, in the physics department. He is carrying out research at the PSI laboratory, in the document analysis team. His current research interests include engineering drawing interpretation, knowledge-based systems in document understanding, distributed artificial intelligence for document analysis and perception systems. He has been working with French institutes and industries in this research area: France Télécom, EDF, Matra, etc. Dr Mullot is a member of the GRCE (French Research Group dealing with document analysis and recognition), and is also member of the IAPR.

Jacques Labiche was born in 1945 in Louviers, France, and received his PhD degree from the University of Rouen, France in 1975. Since 1975 he has been teaching biophysics at the University of Casablanca, Morocco, and has worked in the prosthetics team. Since 1987, after having received a second PhD degree from the University of Bordeaux, France, he has been teaching data processing, computer science and computer vision at the Universities of Caen and Rouen, France, and has worked in the 'document interpretation engineering' team at the LaP and PSI laboratories. Since 1994 he has been teaching computer science and computer vision at the University of Rouen, and has worked in the 'document interpretation engineering' team of the PSI Laboratory. His research interests deal with pattern recognition, knowledge modelling and cognitive sciences. Applications concern technical documentation forms, workflow and information systems. He has also served on conference program committees CNED, CIDE and CIFED.

Yves Lecourtier received his 'Doctorat de $3^{ème}$ cycle' degree in signal processing in 1978 and his 'Doctorat d'Etat' degree in physics (automatic control) in 1985 from the University of Paris-Sud, France. His major interest was in the structural properties of dynamical models. He was an Associate Professor at the Institute of Technology of Saint-Denis from 1974 to 1987. He then joined the University of Rouen as a Professor. His research area is now in pattern recognition and neural networks methodologies, especially for optical text recognition. Dr Lecourtier is author and co-author of more than 100 papers on these various topics, and of one book, *Gracet and Sequential Logic*.

Correspondence and offprint requests to: Rémy Mullot, Laboratoire PSI-La3i, Université de Rouen, 76821 Mont Saint Aignan Cedex, France. Email: Remy.Mullot@univ-rouen.fr