Modelisation and Classification for Retrospective Conversion of Documents

Pierre Héroux, Eric Trupin, Yves Lecourtier

Modelisation and Classification for Retrospective Conversion of Documents

Pierre Héroux, Éric Trupin, and Yves Lecoutier

Laboratoire Perception, Systèmes et Information
UFR des Sciences et Techniques
Université de Rouen
76821 Mont-Saint-Aignan Cedex - France
tel: (+33) 2 35 14 67 86
fax: (+33) 2 35 14 66 18
Pierre.Heroux@univ-rouen.fr
http://www.univ-rouen.fr/psi

Abstract This article presents a document representation oriented towards retrospective conversion. Retrospective conversion of documents allows to convert document images into an electronic format. Classification methods using the proposed representation are presented. Some of these classification methods are used to classify the constituant elements of documents. Others are used to classify whole documents. These classification methods are combined into a global strategy for retrospective conversion. This strategy is based on a cycle where different understanding levels interact. This so called "understanding cycle" is designed around the notion of inconsistency detection and solving. Some results concerning object classification as well as document classification are presented. These results emphasize the influence of the interpretation cycle in the retrospective converision process.

1 Introduction

This article deals with retrospective conversion of documents. The retrospective conversion of a document is a process which consists in constructing a high level representation of the document from the document image.

The retrospective conversion interest lies in the fact that this process allows paper documents to benefit of the advantages given to high level electronic documents. For example, documents converted this way could be diffused on networks. They could also be edited, modified or updated with softwares. They could also be archived and indexed on their content or their structure.

The fomalism which is used to model document in the retrospective conversion process must be as complete as possible so that the document representation can easyly be converted into a convenient format which may be understood by softwares.

On the other hand, the formalism used to build the document representation must be adapted to the retrospective conversion process. In particular, this

means that the document representation should allow to describe each of the understanding level of the document from the input low level (document image) to the high level (interpreted document).

A document can be considered from different points of view [6]. First, a document can be considered as an image. An other analysis level could consider a document according to its layout structure. The logical structure is an other way to describe the document [3].

The *layout structure* of a document represents its visual organization. It is composed of *layout objects* (simple objects or compound objects) organized in a hierarchical way. Each layout object represents a graphical or material entity of the document (volume, page, column, bloc...). Figure 1 represents the image of a document and its layout structure.

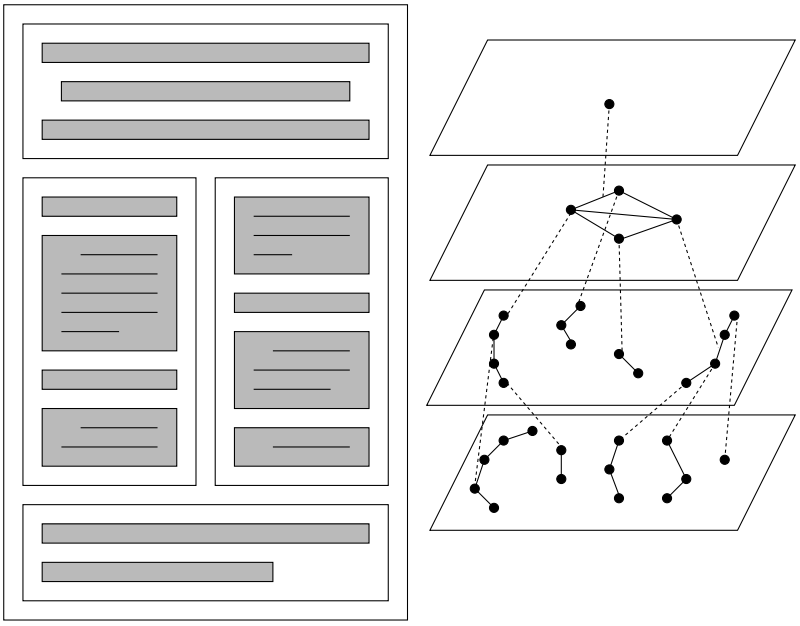


Figure1. A document image and the layout structure

The *logical structure* represents the organization of a document considering meaningful entities. This structure is consistuted of *logical objects*. It is a hierarchical representation which describes the way a document can be parted into title, chapters, sections, subsections, paragraphs... Figure 2 represents what could be the logical structure of the document shown in Figure 1.

The process which aims at constructing the layout structure of a document from the document image is called *document analysis* [1]. *Document understanding* is the process which aims at constructing the logical structure.

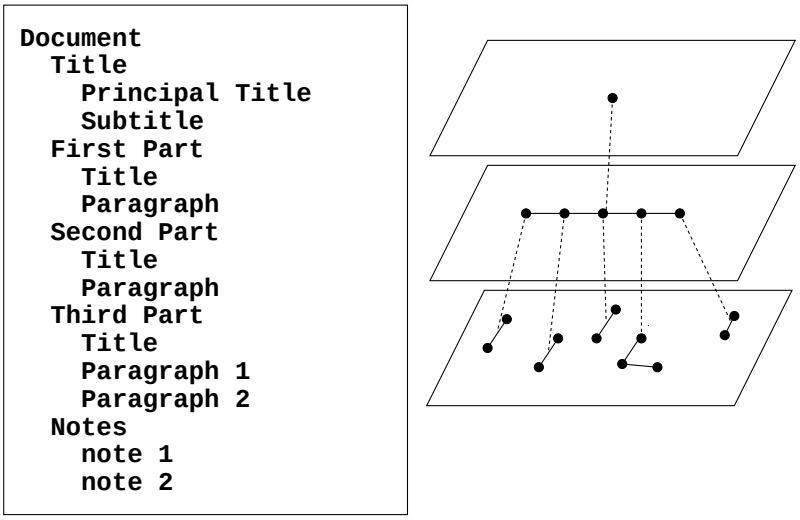


Figure2. The logical structure

We define a *document class* as a set of documents which shares a part of their layout structure and a part of their logical structure. The common part of the structures is called the *generic structure*. Then, a document class is decribed by a generic layout structure and a generic logical structure.

Objects can also be grouped into classes according to common features. An *object class* is a set of objects which shares common features. An object class is described by a *generic object* which express these common features.

This paper presents a strategy for the retrospective conversion of documents. This strategy can be applied to a wide range of documents. On one hand, the documents should be grouped into document classes. On the other hand, the document components (layout and logical ojects) have also to be grouped into object classes. This strategy is based on the *understanding cycle*. It has been inspired by the understanding strategy by perceptive cycles described by Ogier in [5]. This strategy makes document analysis and document understandind processes interact. Each iteration of the cycle consists in three steps. First the pertinence of the current representation is evaluated. Then inconsistencies of the representation are localized. Finally, an updated representation which tries to correct the representation is provided. The cycle ends when no more inconsistency occurs.

Key elements of this strategy are the classification methods used for object and structure classification and the consistency management.

Section 2 details the representation of the document which is used in the understanding process. This section also describes the classification methods used in object and structure recognition. Section 3 presents some results concerning

object and document classification. These results conduct to emit some prospects which are reported in section 4.

2 Document Representation and Classification

2.1 Objects

In this section, the methods used to classify objects are presented. The object classification is performed by comparing the specific object to be identified with the generic objects representing objects classes.

Each object contains two kind of information (see Figure 3). On one hand, a vector contains numerical features describing object intrinsic information independently of its context. The feature vector of layout objects is constituted of information relative to the visual aspect of the object (location, dimension, grey level...). The feature vector of logical objects contains information relative to formatting information (indentation, alignment, size, boldness, underlinement, italic...). On the other hand, a structural information describes the object environment in the structure. This information is presented as a graph which details the way the object is inserted in the document structure. This graph is composed of a node representing the class of the hierarchically superior object and nodes representing the object components. The edges of this graph describe hierarcal and neighbouring relations between the objects.

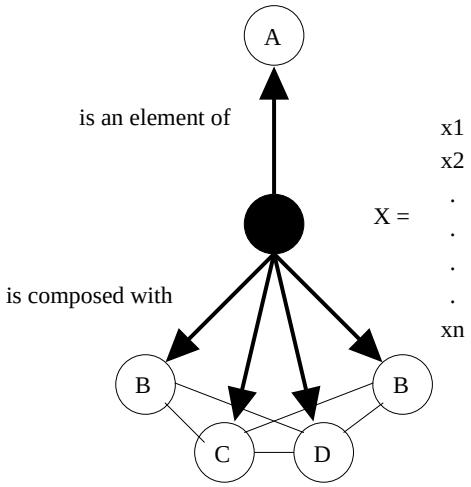


Figure3. Structure of an object

2.2 Training Object Classes

A generic object describes an object class. It contains information shared by all the object constituting the class. Generic objects are created by performing a supervised training. The resulting generic object is contituted of a feature vector, which is the mean feature vector of the objects from the training database. It is also constituted of the graph representing the structural information of the object which correspond to the subgraph isomorphism of the graphs of the training set.

2.3 Object Classification

Object classification consists in identifying the class of an object and then, it leads to the recognition of the element described by the object. Object classification is performed by searching which of the known generic objects representing the object classes is the most similar to the specific object to be identify. The intrinsic and structural information contained in the object description is used by two classifiers to perform this classification. Each of the classifiers provides as an output a weighted list of hypothesis for each object class. The weight associated to each hypothesis qualifies the confidence degree in the hypothesis which is linked to the distance between the specific and the generic object.

A first classification method uses the object intrinsic properties (nurical feature vector) by computing the Euclidean distance between the feature vector of the object to be identified and the feature vectors of the generic objects representing object classes. This distance expresses the proximity between the object to be identified and the object classes in the representation space. Finally, this classification method returns a weighted list of plausible classes. The weight associated to each hypothesis is inversely proportional to the distance.

On the other hand, a structural classification method provides an other weighted list of hypothesis. This list is weighted by the inverse of the distance between the graph representing the structure of the object to be identified and the graphs representing the structures of the generic objects. The graph distance is computed by the search of a subgraph isomorphism [4]. Two overlapping rates t_1 and t_2 are computed by dividing the number of nodes of the subgraph isomorphism by the number of nodes of each of the compared graphs. If one of this rates equals 1, one of the graph is included in the other. In this case, if the other rates is very small, the included graph is very small in regard to the other one. If the compared graphs are equal, t_1 and t_2 equal 1. Then, the similarity measurement [2] is established as

$$d = \frac{1}{t_1.t_2} - 1.$$

The final decision concerning the class attributed to the object to be identified is established by merging the weighted list of hypotesis. Several methods have been tested to perform the fusion of the lists.

Object classification is performed at each iteration of the understanding cycle. During the first iteration of the understanding cycle, only the classification

method performing the numerical feature vector is used because this first iteration initialises the object structure. Once, each object has been classified, the hypothesis choosed as the class of the object is communicated to the connected objects (parent, component and neighbour objects) in the structure.

In the following steps, the object classification method takes into account not only intrinsic features of the object, but it also refers to the hierarchical context of the object.

2.4 Document Structure Classification

During each iteration of the understanding cycle, once every objects have been classified, layout and logical structures are submitted to a structural classifier to determine the document class. The classification of the layout and logical structures is performed by comparing the structure to be identified to each of the generic structures. Finally, an hypothesis concerning the document class is determined as the class given by the layout and logical structure classification if the hypothesis provided each classifier give the same result. In other cases, the hypothesis which is choosed corresponds to the best confidence degree.

2.5 Classification Cycle

A classification cycle includes object classification and structure classification. This cycle is shown on Figure 4. It aims at providing a stable representation of the document.

At each iteration, each of the layout and logical objects are classified.

Then, layout and logical structures are also classified. This step aims at providing an hypothesis concerning the class of the document.

The results of these classifications (object and structure classification) is exploited during the update of the document representation. These results are propageted in the document representation so that this representation is consistent for the following iteration of the classification cycle. Each hypothesis concerning the class of an object is communicated to the object which are connected with it.

3 Results

To confirm our approach, an experiment has been lead on a synthetic document database. A graphical user interface allows an operator to seize the representation of a document. From the document representation, a synthetic document database is generated by applying different modifications. These modifications are the addition of a Gaussian noise on the feature vector of of the object, the objects deletion in the representation and the modification of label associated to some objects. The dimension of modifications brought to the original document depends on three parameters :

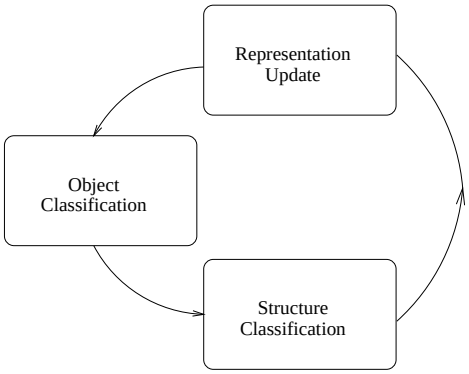


Figure4. Classification cycle

- the dimension of the noise gaussien ;
- the rate of deleted objects ;
- the rate of label modification.

Figure 5 shows the graphical user interface used for the seizure, the display and the modification of the document representation. The left part represents the logical structure of the document while right part represents its visual aspect. Figures 6(a) and 6(b) illustrate the effect of modifications brought by the algorithm of synthetic document generation. Figure 6 represents the visual aspect of the document seized by the operator and the Figure 7 represents the automatically generated document. This algorithm of automatic generation of synthetic documents database is used to build a training database of which will be extracted all the objects and the structures to build generic objects and generic structures. This algorithm is also used to build another synthetic documents database which will be used as a test database.

A first experiment consisted in evaluating the various classification methods of objects extracted from the document test database. This classification was made without using the understanding cycle. The obtained results show that only the classifieur exploiting the intrinsic features of the object gives acceptable results, the recognition rate varying between 70 and 75% according to the size of the training database which varied from 5 to 50 documents by class with 10 document classes, the test database being, for its part, constituted by 100 documents also distributed in 10 classes. The object extraction of each of the documents constituted a test object database of about 1900 objects distributed in 18 classes.

The results given by the classifiers only using the structural information gave results lower than 30%. However, we noticed that the noise added to create the synthetic documents led to considerable disturbances on the structural information. Indeed, due to position and dimension variations of layout objects, the structural connections between objects which were connected in the original document may have been broken. It would be interesting to consider the evolution of

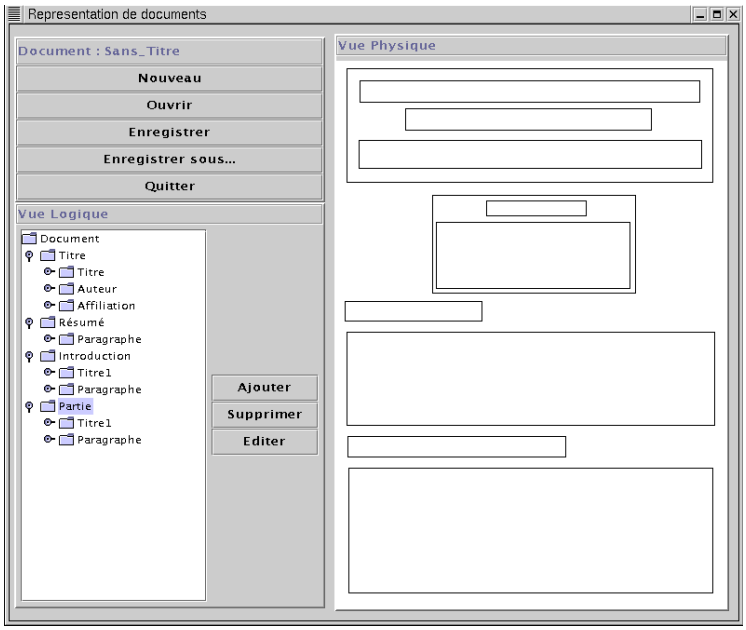
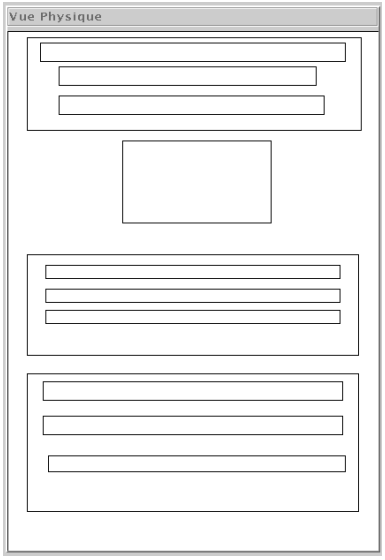
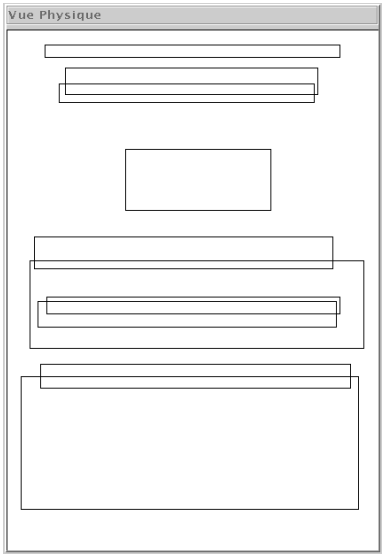


Figure5. Graphical User Interface



(a) Synthetic document seized by the oparator



(b) Automatically generated document

Figure6. Visual aspect of synthetic documents

the recognition rate of methods exploiting the structural information when the noise added to the intrinsic feature vector has a lower influence on the structure of the document.

In a second time, several combinations of this intrinsic and structural information were tested. Only the method consisting in retaining, as class of the object to be identified, the class present in better rank in lists supplied with classifieurs gave interesting results. However these results stay of the same order as the classification from the simple feature vector.

On the other hand, the structural information seems to be interesting when they are used in combination with intrinsic information during the understanding cycle. Indeed, the recognition rates improved during iterations when the results of object classification of the previous iteration is propagated in the representation (see Table 1).

Size of the training set (documents per class)	Iteration 1	Iteration 2	Iteration 3	End of cycle
5	52%	65%	62%	60%
10	49%	58%	55%	56%
20	46%	48%	56%	does not converge

Table1. Evolution of the object recognition rate

The fact that the recognition rates degrade on two successive iterations is caused by the fact that some interpretations being stable, they are not questioned any more. Therefore only documents the most difficult to recognize are afterward classified. It is interesting to notice that the recognition rate of recognition at the end of cycle, even though it remains low, is superior to rate obtained during the first iteration. Finally, even though the results of the classification of objects representing the constituent elements of documents seems low, it seems sufficient for the document structure recognition. Indeed, we noticed that if only 50% of the objects were recognized, the 100% of the document structures were recognized. Although our database is not large enough, we explain this phenomenon by the fact that only the most stable objects which are the easiest to recognize, are taken into account in the generic structure.

4 Conclusion

In this article, we presented a document representation adapted to the retrospective conversion process as well as classification methods of objects representing the elements of documents. We presented document classification methods as

well. The results obtained from a synthetic documents database can be considered according to different point of view. First, the object recognition rate of recognition seems weak. However, this proportion of recognized objects is sufficient to identify the document class in a reliable way. On the other hand, even if the structural information only brings a little contribution to the classification, its interest can be considerable. Indeed, this information could be used in the resolution of localized inconsistencies.

Finally, we could envisage to improve the recognition rate by making search and resolution of inconsistencies in the document representation.

References

1. Sébastien Diana, Éric Trupin, Frédéric Jouzel, Jacques Labiche, and Yves Lecoutier. From acquisition to modelisation of a form base to retrieve information. In *Fourth International Conference on Document Analysis and Recognition*. IAPR, 1997.
2. Pierre Héroux, Sébastien Diana, Éric Trupin, and Yves Lecourtier. A structural classifier to automatically identify form classes. *Advances in Pattern Recognition, Lecture Notes in Computer Science*, 1451:429–439, 1998.
3. International Standard Organization. *ISO 8613 : Information Processing. Text and Office System, Office Document Architecture (ODA) and Interchange Format*, 1989.
4. Laurent Miclet. *Méthodes structurelles pour la reconnaissances de formes*. Eyrolles, 1984.
5. Jean-Marc Ogier, Rémy Mullot, Jacques Labiche, and Yves Lecourtier. Interprétation de document par cycles perceptifs de construction d'objets cohérents. application aux données cadastrales. *Traitement du Signal*, 12(6):627–637, 1995.
6. Yves Stern. Les quatre dimensions du document. *Document numérique*, 1(1):55–60, 1997.