Structural 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 paper describes the structural classification method used in a strategy for retrospective conversion of documents. This strategy consists in an cycle in which document analysis and document understanding interact. This cycle is initialized by the extraction of the outline of the layout and logical structures of the document. Then, each iteration of the cycle consists in the detection and the processing of inconsistencies in the document modeling. The cycle ends when no more inconsistency occurs.
A structural representation is used to describe documents. This representation is detailed.
Retrospective conversion consists in identifying each entity of the document and its structures as well. The structural classification method based on graph comparison is used at several levels of this process. Graph comparison is also used in the learning of generic entities.

Keywords: retrospective conversion, document structure.

1 Introduction

This paper describes a strategy used for retrospective conversion of document. Retrospective conversion of documents consists in constructing a document representation from the document image. The obtained representation can easily be modified to an electronic format. Retrospective conversion is useful because it allows paper documents to benefit of advantages of electronic documents which can be edited, diffused, indexed and archived.

Retrospective conversion of document is often constituted of two major steps (cf. Fig.1): document analysis and document understanding. Document analysis consists in extracting the layout structure of a document from its image. Document understanding aims at building the logical structure of the document.

In this paper we propose a strategy for retrospective conversion of documents based on structural classification. Section 2 details the document representation.

F.J. Ferri et al. (Eds.): SSPR&SPR 2000, LNCS 1876, pp. 154–162, 2000.

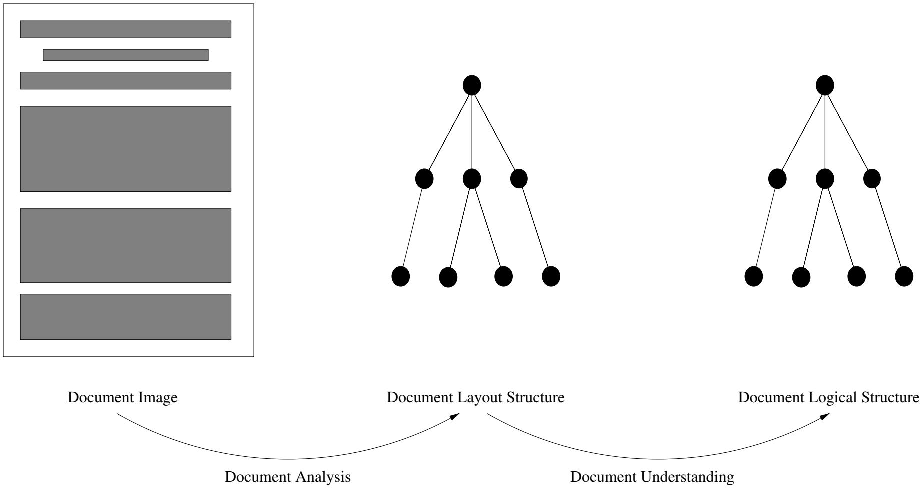


Fig. 1. Retrospective Conversion of a Document

The algorithm used for structural classification is presented in section 3. Section 4 details the different steps of document understanding

2 Document Representation

A document can be described by two structures: the layout structure and the logical structure [3]. The layout structure hierarchically models the visual aspect of documents. It is obtained by extracting and classifying graphical elements of the document image. These graphical elements are represented by so called layout objects. The logical structure represents the document organization on the basis of the meaning of the content. The logical structure describes the way a document can be parted into title, sections, subsections, paragraphs... Each logical element is described by a logical object.

Documents can grouped into classes. A document class is a set of documents which share a part of their layout structure and logical structure. The part of the structure which is shared by all the documents from a class is the generic structure. It defines a structure class. Then each document class is represented by a generic layout structure and a generic logical structure.

Objects can also be grouped into classes. A generic object (generic layout object or generic logical object) describes an object class and is constituted of the features common to each object of the class.

In our document representation, layout objects represent graphical elements of the document image (a text line, a text block, a text column, an image...) and logical objects represent meaningful entities (title, section, subsection...).

An object can be a basic object or a compound object. Each object has four attributes (see Fig. 2):

- its label;
- a numerical feature vector;
- the label of its parent object;
- its structure;

The label of the object is the name of the class it belongs to. The object classification process consists in determining this attributes.

The numerical feature vector contains intrinsic informations. The feature vector of a layout object contains visual indices concerning the graphical entity represented (location, dimension, black pixel density). The feature vector of a logical object contains formating information (alignment, style, size).

A graph $G_{obj}(V_{obj}, E_{obj}, \alpha_{obj}, \beta_{obj})$ represents the structure of each object. If the object is a basic object, the graph is empty, but if it is a compound object, each node of the graph is labeled by the class of its components. An edge is established between two nodes if the components present a neighboring relation.

The feature vector, the label of the parent object and the structure of the object are used in the object classification. This process is detailed in section 4.2.

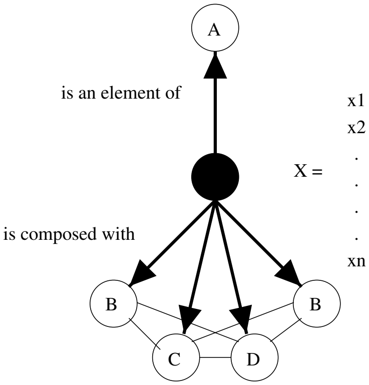


Fig. 2. Structure of an object

The structures describe the way the objects are organized in the document. Two graphs $G_{lay}(V_{lay}, E_{lay}, \alpha_{lay}, \beta_{lay})$ and $G_{log}(V_{log}, E_{log}, \alpha_{log}, \beta_{log})$ represent the layout (Fig. 3) and logical (Fig. 4) structures of the document. The nodes of these graphs are labeled by the label of the objects and edges describe hierarchical and neighboring relations.

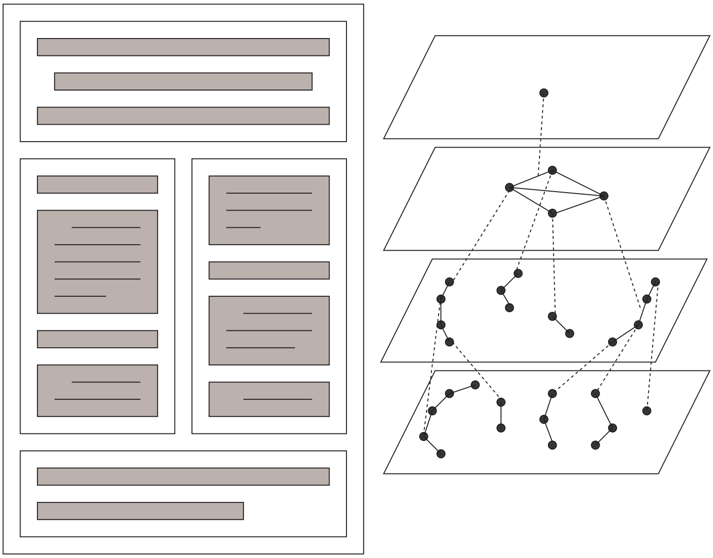
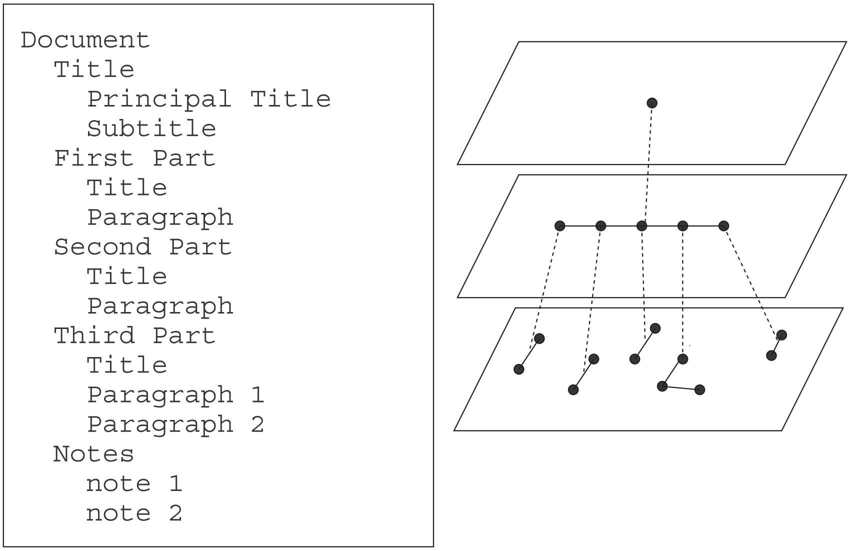


Fig. 3. Layout structure of a document



```
Document
  Title
    Principal Title
    Subtitle
  First Part
    Title
    Paragraph
  Second Part
    Title
    Paragraph
  Third Part
    Title
    Paragraph 1
    Paragraph 2
  Notes
    note 1
    note 2
```

Fig. 4. Logical structure of a document

3 Structural Classification

Section 2 describes the different elements used for document representation. This structural representation uses graphs to represent document layout and logical structures and object structure.

The retrospective conversion consists in extracting the different elements of the document representation from its image and in determining the class of each object and finally the class of the document. Structural classification helps in that task. This section details the method used for structural classification.

Our structural classification is based on the search of a subgraph isomorphism [4] between the graph to be identified and graphs representing generic entities. For example, if the graph to be identified represents the structure of layout object, its is compared with all graphs representing the structure of generic layout objects. If the graph represents the logical structure of the document, it is compared with the graphs representing the generic logical structures.

Each comparison of two graphs $G_1(E_1, V_1, \alpha_1, \beta_1)$ and $G_2(E_2, V_2, \alpha_2, \beta_2)$ produces a graph $G_3(E_3, V_3, \alpha_3, \beta_3)$ which is constructed as follow. First, the greatest matching between equivalent edges from V_1 and V_2 is searched. Two edges are consisdered equivalent if their label are equals and if the label associated to their extremities are equals. This produces an initial version of G_3. E_3 is completed by finding the greatest matching between nodes from E_1 and E_2 which have not been associated during the first step.

We define a similarity measurement [2] $\delta(G_1, G_2)$ between G_1 and G_2. Two overlapping rates t_1 and t_2 are determined. t_1 is defined by the number of nodes of G_3 divided by the number of nodes of G_1 and $t2$ is equal to the number of nodes of G_3 divided by G_2. If one of these rates equals 1, this means that one of the graph is included in the other one. In this case, if the other rate is very small, then the included graph is very small in regard to the other one. If the compared graphs are equal, t_1 and t_2 are equal to 1. A similarity measurement can be established as

$$\delta(G_1, G_2) = \frac{1}{t_1 . t_2} - 1.$$

4 Retrospective Conversion

4.1 Interpretation Cycle

A complete retrospective conversion of documents has to construct a document modeling which represents, at least, the layout structure and the logical structure of the document. Our strategy is based on a cycle inspired by Ogier in [5]. This cycle makes document analysis and document understanding interact. The cycle (see Fig. 5) is initialised by a phase which provides primitive versions of layout and logical structures.

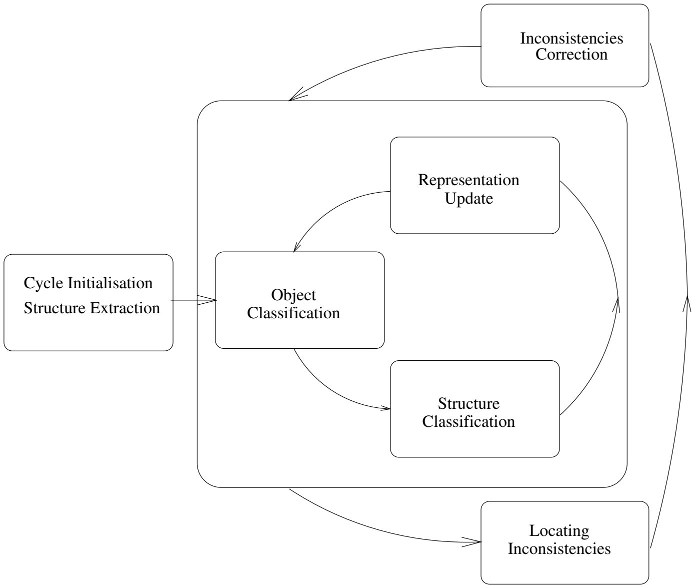


Fig. 5. Interpretation cycle

The outline of the layout structure is obtained by extracting graphical objects from the document image [1]. This is performed by a segmentation algorithm applied on the document image after low level processing (deskewing and binarisation). Extracted objects are then associated in composite layout objects according to size and proximity criteria. Then, they are labelled (text, graphic, image...) according to graphic criteria (size, black pixel density...). New composite objects are then constructed with adjacent objects which are identically labelled. Finally, a first version of the layout structure is obtained.

The structural classification method which compares a specific structure to be identified with structures representing document classes gives a first hypothesis concerning the document class. Assuming that a document class contains not only a generic layout structure but also a generic logical structure, the outline of the logical structure is built by instanciating the generic logical structure corresponding to this hypothesis. This instanciation is performed by associating a logical equivalent to basic layout objects.

This initialises the interpretation cycle. Each iteration of the cycle consists in the locating and the processing of inconsistencies in the document representation. Each time the class attributed to an object and the structure is called into question. So objects and structures should be classified every time.

Different level of consistency are examined. First, we define what we call intrinsic inconsistency. It refers to the fact that no generic object contains the features observed for the specific object. The object can not be associated to any of the known object classes. On the contrary, an object is said intrinsically consistent if its features are able to occur in regard to the known object classes.

The next consistency level is called contextual neighboring consistency. An object is said to be consistent at the neighboring contextual level if there is at least one generic object which includes this object and its neighbors as its constituents in the observed configuration.

The hierarchical consistency deals with the fact that an object associated to a specific class can, or not, be a constituent of an object of an other class. An object is said to be hierarchically consistent if its class is compatible with the class of the hierarchically superior object.

Finally, we define the abstraction level consistency. It deals with the compatibilty between the class of a logical object and the class of the corresponding layout object. This mapping between layout and logical object is not always possible. A logical object not always correspond to a single layout object. For instance, a paragraph can be split into two text blocks on two columns. However the abstraction level consistency can always be evaluated for structures. The results of layout structure and logical structure classification must correspond to the same document class.

4.2 Object Classification

The object classification aims at attributing each object to a known class which represented by a generic object. The object to be identified is compared to all generic objects. It is performed by making three classifiers cooperate. Each of these classifiers gives a list of hypothesis weighted by a similarity measurement.

A statistical classifier (Nearest Neighbor) uses the distance between the feature vector of the object to be identified and the feature vector of the generic object it is compared to. The list of hypothesis is weighted by the inverse of the distance.

The structural classifier presented in section 3 is used. It compares the graph G representing the structure of the object to the graphs G_{gen} representing the structure of generic objects. This classifier provides a list of hypothesis weighted by the similarity measuerment $\delta(G, G_{gen})$.

The third classifier uses as information the label of the parent object.

The results of the three classifiers are exploited by computing an weighted sum of the weight associated to each hypothesis.

4.3 Structure Classification

After that each object has been classified, the layout and logical structures are updated by labeling the nodes of the graphs by the label of the object. Then, the layout and logical structures are independently classified. The graph representing the structure is compared to the graphs representing generic structures. The label attributed to the structure to be identified is the class described by the generic structure whose similarity measurement is the greatest, but a weighted list of hypothesis is established.

4.4 Document Classification

Finally, once the structures have been classified, an hypothesis concerning the class of the document is given. This hypothesis depends on the structure classification. Both layout structure classification and logical structure classification have given a weighted list of hypothesis concerning the class of the document. The choosed hypothesis corresponds to the unweighted sum of the list given by the structure classification.

5 Structural Training

The graph comparison presented in section 3 is used in structural training. Structural training aims at building generic objects or generic structures. A training database is constituted from specific graphs from the same class. The graph representing the generic object or the generic structure is built by searching the greatest subgraph.

6 Conclusion

This paper proposes a strategy for retrospective conversion of documents. This is based on the interpretation cycle which consists in classifying each object analysing the consistency of the description and solve the inconsistencies. This cycle makes document analysis and document understanding dynamically interact. On one hand, the logical structure is initialised from the knowledge of the layout structure. On the other hand, the layout structure is not fixed and inconsistencies in the logical structure can lead to call into question the layout structure.

The document representation describes three different contextual relations between objects (neighboring relations, hierarchical relations, layout-logical relations). These differents levels of relation are exploited by the classification methods.

This strategy is being implanted in a document processing system which should be able to process a wide range of documents and provide a convenient representation. Fig. 6 represents the graphical user interface of our system which allows a user to verify, edit and correct the representation of a document. It is also used to build a database of synthetic documents. Even if the first results are not significant, they are encouraging.

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 *Fouth 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.

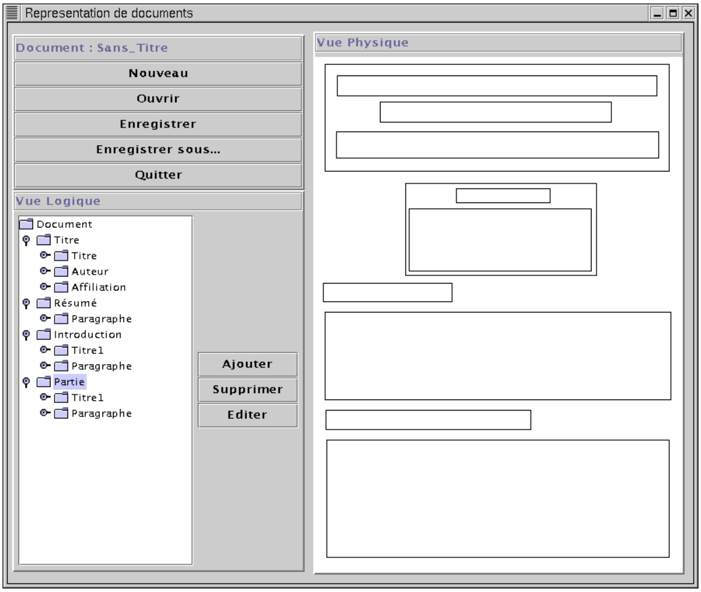


Fig. 6. Graphical User Interface

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.