J Real-Time Image Proc (2014) 9:329–345
DOI 10.1007/s11554-011-0227-4

ORIGINAL RESEARCH PAPER

Classification of business documents for real-time application

Djamel Gaceb · Véronique Eglin · Frank Lebourgeois

Received: 29 January 2010 / Accepted: 3 October 2011 / Published online: 30 November 2011

Abstract In this paper, we present a new document classification based on physical layout features and graph b-coloring modeling. In order to reduce the computing time and to increase the performance of our automatic reading system, we propose to pre-classify the business documents by introducing an Automatic Recognition of Documents stage as a pre-analysis phase. This phase guides others involved in the recognition process of the documents contents. Once the document type is identified, the reading system will use its corresponding information source to improve the recognition of its logical layout, the selection and parameterization of the OCR, and the final decision of sorting. The graph coloring model is introduced for both layout analysis and document classification. The proposed method is reliable, robust to various constraints and guarantees a real-time answer to the sorting of business documents.

Keywords Layout extraction · Classification of business documents · Document sorting by industrial vision · Pattern recognition · Real-time processing · Graph coloring

D. Gaceb (✉) · V. Eglin · F. Lebourgeois
LIRIS INSA de Lyon, 20, Av. Albert Einstein,
69621 Villeurbanne Cedex, France
e-mail: djamel.gaceb1@insa-lyon.fr

V. Eglin
e-mail: veronique.eglin@insa-lyon.fr

F. Lebourgeois
e-mail: flebourg@insa-lyon.fr

1 Introduction

The automatic processing of documents is a significant added value to the companies. It makes more accessible the rich documentary heritage and allows new services which can improve the organization of companies. In particular, the automatic sorting of documents saves time and reduces the costs of manual handling. This field of experimentation of new technologies requires all the analytical steps from the lowest level (preprocessing and segmentation of images) to the highest level (recognition and decision). Current trends are moving towards increasing the accuracy and robustness of the embedded approaches of recognition to process images of documents that have heterogeneous content (printed or handwritten). In order to break the actual limits of the OCR, the solution consists of improving the overall organization of the computer vision system by introducing feedback loops and other processes which bring new information about document contents at each stage of the processing. By taking into account the type of documents, the layout, the text, the typography of character (fonts and style for printing and writing type for handwritten documents), we can achieve an intelligent recognition. Any recognition system of documents requires the introduction of prior knowledge related to the type of document to be recognized [1]. Most of these recognition systems embed this knowledge into the program directly which becomes difficult to adapt for new documents.

The Automatic Recognition of Documents (ARD) system is used for document classification, which provides information to various stages like the OCR, the layout analysis stage, the decisions stage and the selection of the adapted dictionaries (Fig. 1).

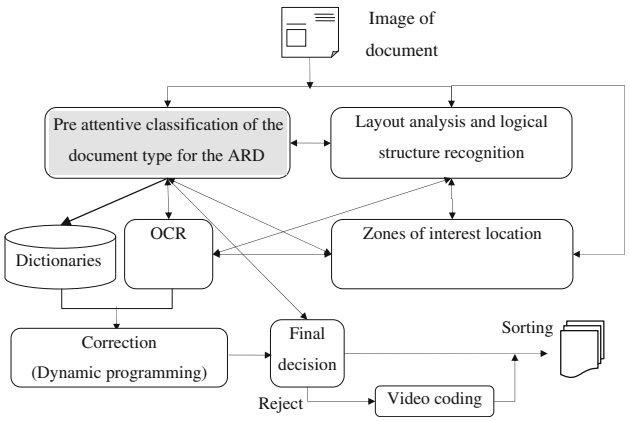

Fig. 1 Location of the ARD stage into the general scheme of the documents sorting systems

The introduction of an ARD stage in the overall scheme of a documents sorting system remains an unsolved problem, which must respect several constraints:

- A large variety of documents from different origins having various structures which contain both handwritten and printed text using textured background (Fig. 2);
- A real-time processing (only a fraction of second for the recognition);
- A high recognition result to avoid the expensive manual correction by video coding;
- A high image resolution (300 dpi) which slows down the analysis;
- The superposition of different information layers (marks, logo, handwritten notes …).

To satisfy all these constraints, we propose a flexible ARD architecture based on a new approach, which uses the hierarchical coloring of graphs. Until now, this powerful approach has never been used in document image analysis.

The paper is organized as follows: the next section describes the different existing approaches of document classification and their limits. In Sect. 3, we present the graph b-coloring method and its application in an ARD stage. At the end of the paper, we will present our results which are obtained with a large database.

2 Classification methods

2.1 Different levels of representation

A document can be considered as a complex organization of various objects (text, graphic, notes and other symbols of all types) located randomly and having an irregular arrangement. The recognition of document consists in clustering documents having similar structure and text contents into the same class. Several unsupervised and supervised classification methods can be used to classify documents images like the K-means, Markov chains, decision trees, isomorphism of graphs, SVM, neural networks, and various statistical approaches. These methods can use different types of features at different levels of representation:

- Features extracted from image only without a segmentation stage and/or,
- Features extracted from the physical layout and/or,
- Features extracted from logical layout and/or,
- Features extracted from text content.

2.2 Representation based on low-level image features

This description is based on primitives that are relevant to the effective characterization of the content without segmenting the document image. These methods try to adapt a characterization of images to a suitable classification method. Some approaches such as developed in [2] use features that are directly extracted from the image of document without having to segment it into different blocks. These features may be related to: the image information density (through the calculation of moments for example), the statistics that are calculated over all connected components, the layout structural information, the gap between the rows and columns, the measures related to the font size and to other associated typographical effects. Shin et al. [2] worked on this type of approach by calculating the image features from four types of windows: rectangular, horizontal or vertical band and page windows. A similarity measure based on the matching between the different windows is then used to compare the document images. Two types of classifiers were used for the classification of documents: a decision tree and a self-organizing map.

2.3 Representation based on the physical layout

Most methods for document classification based on the physical layout use a hierarchical representation of blocks (word, text lines, blocks, graphics, checked box, tables…). This representation simplifies the comparison between each element of the layout. Heroux [3] described a document with a tree, where each node describes an element of the layout. A comparison of trees allows the classification of documents. Esposito [4] used a simple language to describe the elements of the layouts and their relation. Cesarini [5] compared X–Y trees to classify the documents. Baldi [6] and Diligenti [7] proposed to modify the X–Y tree into XYM tree. Baldi compared with a K-NN rule the distance edition between XYM trees and Diligenti used the tree to build a Hidden Tree Markov Model. Bagdanov [8]

Fig. 2 A large variety of business documents

proposed a document classification based on graph theory and used a First Order Gaussian Graph (FOGGs) where both nodes and edges are described by probabilities that are learned from a training set.

2.4 Representation based on the logical layout

This description is based mainly on an analysis of logical labels used to describe the semantic of each physical block of the document (title, logo, date, name, ACII code, address, amount, signature, etc.). Dengel and Dubiel presented in [9] a classification of business letters that is founded on this type of description. This system is based on the construction of an object hierarchy from a logical manual labeling of blocks and a ranking of the letters in specific categories. For this, the system initially sets the spatial relationships between different blocks using the initial set of labels (subject, sender, receiver, etc.). Then it constructs a decision tree from the document learning dataset. The classification of a new document is then performed by analyzing the decision tree based on elements extracted from image to classify. This approach is limited by the segmentation problems that may occur during the extraction of blocks.

Using the results of the functional labeling, Eglin and Bres [10] presented a complete methodology for the characterization and categorization of documents. This method used statistical measures based on primitive textures and inspired by the mechanisms of human visual perception. The separation process of functional blocks into subclasses is based on a K-means unsupervised classification method.

We can cite, as examples, other classification methods that use the description of both physical and logical layout

based on n-grams [11], pattern matching [12], the Winnow algorithm [13] or logical isomorphism of graph [14], etc.

2.5 Representation based on the textual features (OCR output)

The description of textual content typically uses character frequency, n-grams or keywords. The document classification methods that depend on this description are based on syntactic or semantic analysis. They can also add machine learning methods such as: regression models, k-NN approaches [15], SVM classifier [16], naive Bayesian approaches, decision trees [17] and methods based on knowledge or artificial neural network [18]. Other methods presented in the literature combine the textual with physical layout description [19] or logical layout description [20]. These types of methods are very expensive in terms of time computation and are not suitable for our real-time application.

2.6 The need for a new approach

ARD systems based on logical layout or text contents are difficult to use for a real-time application. The features extracted only from images (without segmentation stage) are not sufficient to provide a discriminating representation of documents. Moreover, the amount of information provided by a simple description of the document image without segmenting and analyzing its physical structure cannot distinguish documents with a high layout variability [2, 3]. These constraints require a simple and distinctive description of content to allow a rapid classification of all documents that may appear in a sorting machine. For a better adaptation to the needs of speed and efficiency required by our sorting application, we are interested in approaches based on the description of the physical layout of pages.

The existing methods use a complex data structure for both the classification and the description of the layout. They require knowledge extraction from a large training set, which must contain representative documents with all possible layouts. Because of the great variability of the layouts, systems described previously are difficult to control. To answer to the industrial needs, we offer an efficient tool, which guarantees stable and coherent results and respects real-time constraints. We propose a new architecture based on graph coloring.

3 Formal aspects of the graph coloring

Graph coloring is a very important branch of graph theory. Its applications are numerous in various scientific fields (optimization of transportation or communication networks, chemical formulas, etc.). The definitions of graph coloring are simple and real research problems can be posed in a well-structured form where the formulation can cover major practical difficulties.

Various classification problems can be modeled by the graph coloring. The general form of these applications requires the formation of a graph by the nodes (vertex) which represent the objects of interest (documents) and the edges (arcs) which define the relations between these objects.

One wants, for example, to break up a set of items into several homogeneous classes without knowing their a priori number. To do that, it is sufficient to represent each item i by a node v_i and to add an edge $E(v_i, v_j)$ between each pair of different individuals. The finite graph $G = (V, E)$ is defined by the finite set $V = \{v_1, v_2 \ldots, v_n\}$, $(|V| = n)$ whose elements are called nodes, and by the finite set $E = \{e_1, e_2 \ldots, e_m\}$ $(|E| = m)$ whose elements are called edges.

3.1 Graph coloring

The coloring of the nodes of the graph $G(V, E)$ consists in assigning to all nodes a color so that two adjacent nodes do not carry the same color. These colors will correspond to the various classes of items. A coloring with k colors is thus a partition of the set of nodes V in k homogeneous subsets. The number of colors used to color the graph G of n nodes is called chromatic number χ which represents the smallest integer k for which there is a partition of V into k homogeneous subsets.

On the graph G of order $|V| = 11$ in Fig. 3, whose set of nodes is $V = \{1 \ldots, 11\}$, four colors were needed to color the nodes so that two adjacent nodes cannot have the same color. $\chi(G) = 4$ is the minimal chromatic number.

3.2 Graph b-coloring

The coloring is called b-coloring, if for each color c_i, there exists at least a colored v_i node included in c_i whose neighborhood is colored by all the other colors. The node v_i is known as a dominating node for the color c_i. The example of Fig. 4 presents the possibility of b-coloring of the nodes of a color class using the other colors (show nodes 1 and 8 of the color c_5).

The b-chromatics number of a graph G, defined by $b(G)$, is the maximum integer number of colors k_b so that G can have a b-coloring by the k_b colors. It can be easily noticed that:

$$\chi(G) \leq b(G) \leq \Delta(G) + 1 \quad (1)$$

where $\Delta(G)$ is the maximum degree of G, called the degree of the node v_i, and its number of incidental edges is noted $d(v_i)$.

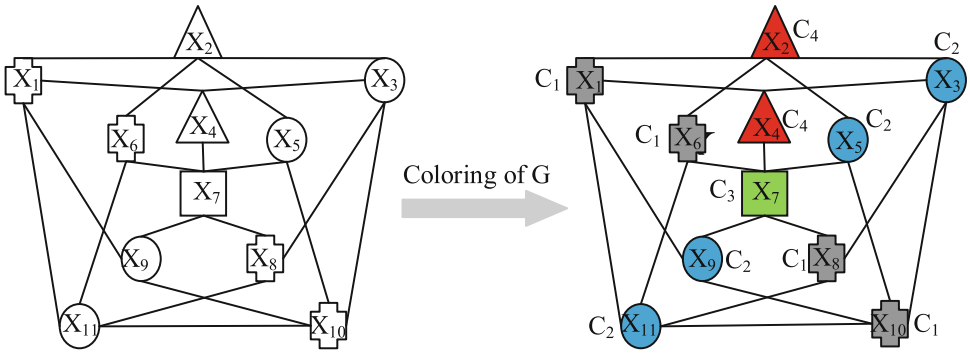

Fig. 3 Coloring of graph G with four colors

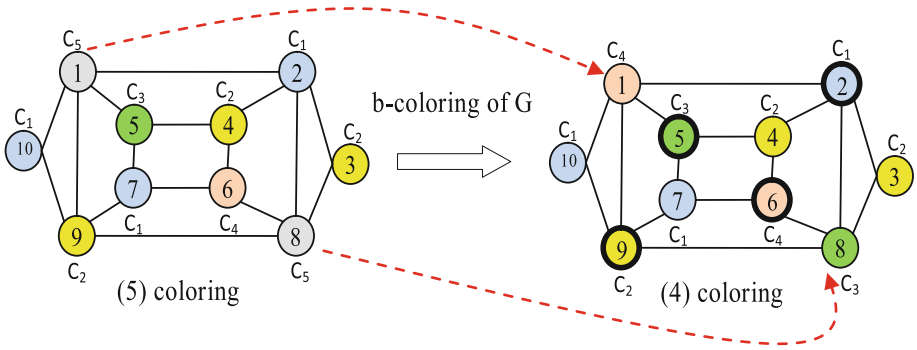

Fig. 4 B-coloring example, the nodes 2, 5, 6 and 9 are the dominating nodes

3.3 Implementation

The majority of the evaluations of $\chi(G)$ and $b(G)$ come from coloring algorithms. There exist many of them. So, we have chosen to limit our choice of the fastest and most recent ones.

New graph coloring and b-coloring algorithms have been proposed by Effantin and Kheddouci [21]. More details on the approximation of the b-chromatic and a good literature review is presented in [22] and [23]. All of these algorithms were efficiently introduced into Elghazel's works [24] who proposed a new unsupervised classification method of medical data based on graph b-coloring where the number of classes is not a priori known. On the same database, the comparison between this method and different approaches like the agglomerative hierarchical classification, the approach of Hansen and the classification of DRG, show that the b-coloring provides a correct representation of classes by the dominant individuals and guarantees a better disparity between classes.

For our automatic sorting application of corporate documents, we thought that the properties of the b-coloring approach could be very effectively used for solving the problems of segmentation and classification of documents. We have therefore paid a particular attention to adapting this approach to our study. More specifically, we found that the facilities offered by the exploitation of distributed algorithms of coloring and b-coloring, such as those proposed by Effantin and Kheddouci in [21], meet the time constraints that are imposed by the industrial real-time applications, as is the case for us.

3.4 Formalization of the document classification problem

The classification process is applied on a training corpus V of n document images $V = \{d_i, \ldots, d_n\}$. We associate to each of the n documents d_i a node v_i of a simple graph G, and to each pair (d_i, d_j) of document that cannot be regrouped together, we associate an edge (v_i, v_j) of this graph. Remember that this edge expresses the dissimilarity between two nodes (thus practically between two documents), a notion which will be defined in detail in the Sect. 4. The objective is to group the documents in homogenous classes. This classification leads to define between each pair of document (d_i, d_j) a similarity measure that reflects the membership or not of the documents to the same class. Two questions related to the classification will then follow from:

- What is the minimum number of classes necessary to regroup the documents in a secured way (by ignoring the constraints of the size of the diverse classes)?
- What are the class representatives that will be defined during the learning phase and that will be used by the recognition phase?

We will reformulate these two central questions in terms of b-coloring of graph G (Fig. 5) and will expose in Sect. 4

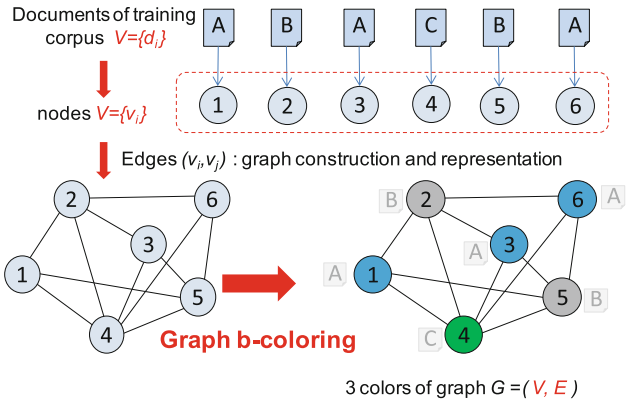

Fig. 5 Example of document classification based on graph b-coloring

the theoretical details and the implementation of these approaches (Fig. 6).

4 Graph coloring in ARD system

We present in this section, the different steps of our ARD system (Fig. 6).

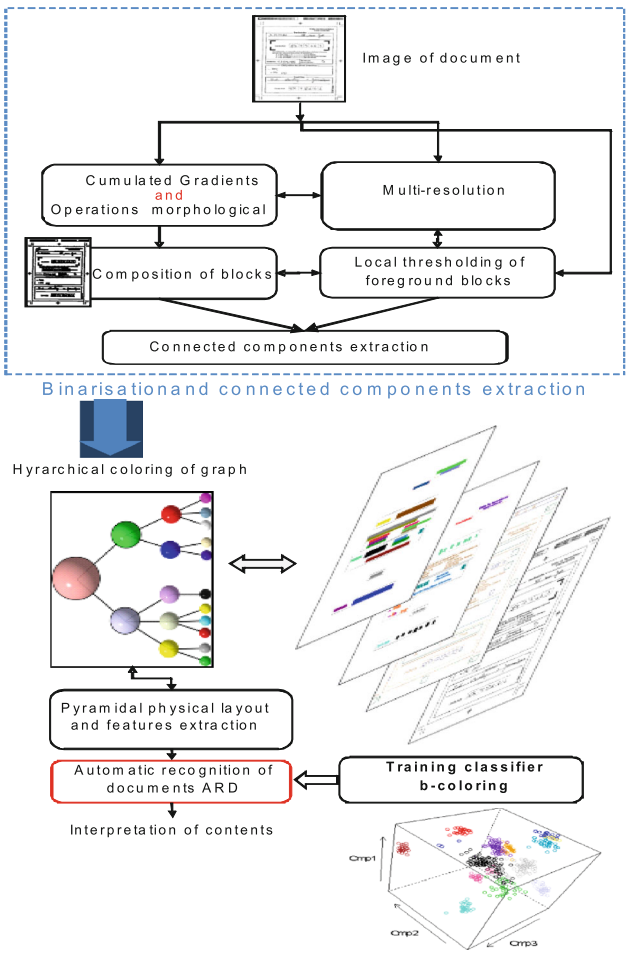

Fig. 6 Functional diagram of the proposed approach

4.1 Layout analysis

4.1.1 Binarisation and detection of connected components (CCs)

The binarisation (or thresholding) is applied in the first stage and has a very strong impact on the performances of the sorting system. The thresholding methods are in general divided into two categories: global (e.g.: Otsu's method [25]), and local (e.g.: Sauvola's method [26]). The simplest methods using a global thresholding has the advantage of being extremely fast but with the change of lighting; the presence of various graphics printed on document with different color inks are rapidly decreasing the quality of binarisation. The local methods exceed these limits and are more adapted to local changes of contrast. However, they require more calculations; thus, they are slower and unsuitable for real-time applications. Although they provide a good efficiency on the documents that are concerned in our application, these local binarisation approaches have mainly the following disadvantages:

- prohibitive time computation depending on the size of the analysis window;
- over-segmentation of the defects and textures of the background of the image;
- difficult treatment of documents whose characters vary in size (the analysis window is fixed throughout the processing).

The separation between the binarisation and text zones location phases considerably increases the computation time and lead to an over-segmentation of the noise and of the paper texture on empty zones of the image. Indeed, none of the traditional methods (whether global or local) efficiently combines all the required conditions, especially a low time consuming. We have managed to optimize this stage by applying a local threshold only near the text zones located by the cumulated gradients method with the multi-resolution and mathematical morphology.

The regularity of text is calculated from a sequence of pixels with high gradients. To better adapt to the large size of our images and the real-time constraints. We divide the image into rectangular blocks of size $dx \times dy$, then we calculate in each block the sum of vertical and horizontal gradients as shown in the following formulas:

$$\mathrm{Gr}(x_0, y_0) = \frac{1}{dx\, dy} \sum_{i=1}^{dy} \sum_{j=1}^{dx} \left| \frac{\partial I(x_0 + j, y_0 + i)}{\partial x} \right| + \left| \frac{\partial I(x_0 + j, y_0 + i)}{\partial y} \right|.$$

$$\text{with } \frac{\partial I}{\partial x}(u, v) = I(u - 2, v) - I(u + 2, v)$$

$$\text{and } \frac{\partial I}{\partial y}(u, v) = I(u, v - 2) - I(u, v + 2). \quad (1)$$

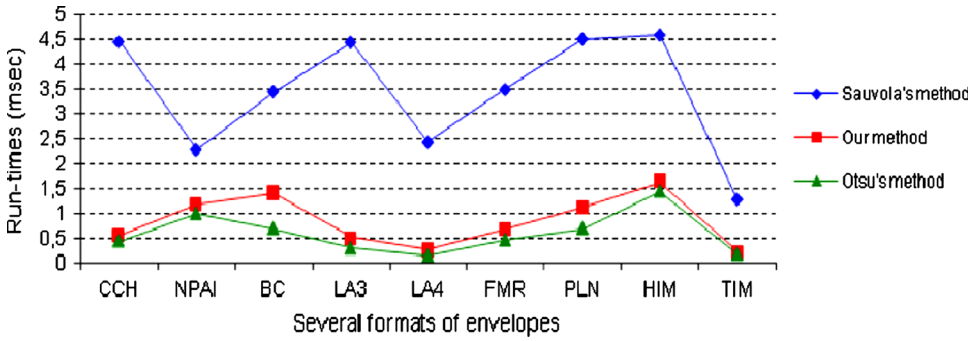

Fig. 7 Run-times comparison of various thresholding methods on 9 classes of documents

The accumulation of the gradients by blocks gives a quick low-resolution picture Gr where the text zones are clearly the brightest areas. We apply successively on the grayscale image *Gr*, *4 dilations, 4 erosions, 2 dilations and 2 erosion*s application of this morphological processing allows, on one hand, to redensify the text and therefore to agglomerate it into blocks and, and on the other hand, to take an adequate margin around the line to be able to include pertinent information that are carried by background (texture and color) for better thresholding.

The morphological processing is continued by a Fisher global thresholding that gives a binary mask which contains the different blocks of the textual zones. This method quickly calculates a global threshold from the histogram of the greyscale image. This binary mask is used to direct the local thresholding in full resolution to the zones of text and may be considered as a first segmentation into blocks of physical structure of the image of document.

This rapid emphasizing on blocks plays two important roles in terms of computation time: on one hand, it can effectively reduce the local thresholding time to make it almost similar to that of global thresholding; on the other hand, it speeds up the extraction phase of the physical layout that we will see later on in details. To obtain a final binary map of the foreground in full resolution, we decided to use the method of Sauvola for its rapidity with respect to the other local methods and for its performance (the Wolf method is specific to videos and is not suitable for our application). This local thresholding is applied only on textual zones. We have produced comparative curves that show that the run-times of our hybrid method of binarisation are approximately similar to those of global methods and very inferior than those of local methods. These run-times are calculated on a set of 29,225 document images divided into nine classes (Fig. 7).

In addition to these advantages, our hybrid method of thresholding has also reduced the computation time of the connected components by the reduction of black pixels in all large black areas (that most often correspond to pictures or publicity indications) that are located as black edges with white centers.

After the binarisation stage, an analysis of CCs is carried out to extract various vital information for the incoming phases. Formally, a connected component is a set of foreground pixels immediately adjacent to each other. Typically, in a machine-printed text, under ideal digitizing conditions, each alphanumerical character is a separate CC. In order to reduce the processing time necessary to the CCs detection, several methods were developed. A good literature review is presented in [27]. In our study we have been interested in Pavlidis' [28] work who has modeled the problem of CCs detection by a line adjacency graph (LAG). The physical layout extraction is then based on a hierarchical analysis on each pyramid level of the bounding boxes. Each level contains different features. These CCs constitute a significant information source, very often used during the description process (Figs. 8 and 9).

4.1.2 Layout extraction by hierarchical coloring of connected components

The physical layout segmentation of the document image is mostly based on its decomposition into constitutive elements containing homogeneous features. These elements are often spaced and form elementary geometrical blocks, based on rectangle bounding boxes. The CCs merging segmentation methods (progressive regrouping of CCs, RLSA, segmentation by scaling method of cumulated gradients) are more used by the bottom-up strategies [29, 30], whereas the methods of segmentation by splitting (profile projection, segmentation by spaces analysis, Hough's transform) are adapted to the top-down strategies [31, 32]. Other methods, known as hybrid take advantage of the two strategies at the same time [1]. Hybrid segmentation approaches gather both strategies in the same time and can benefit from the advantages of one strategy to fill the disadvantages of the other. Our concept of physical layout extraction is based on the same principle of a hybrid strategy. High stages of our approach are based on the Hierarchical Graph Coloring (HGC) that largely makes use of all the levels of the pyramidal structure and the coloring effectiveness, so as to extract, to characterize and precisely

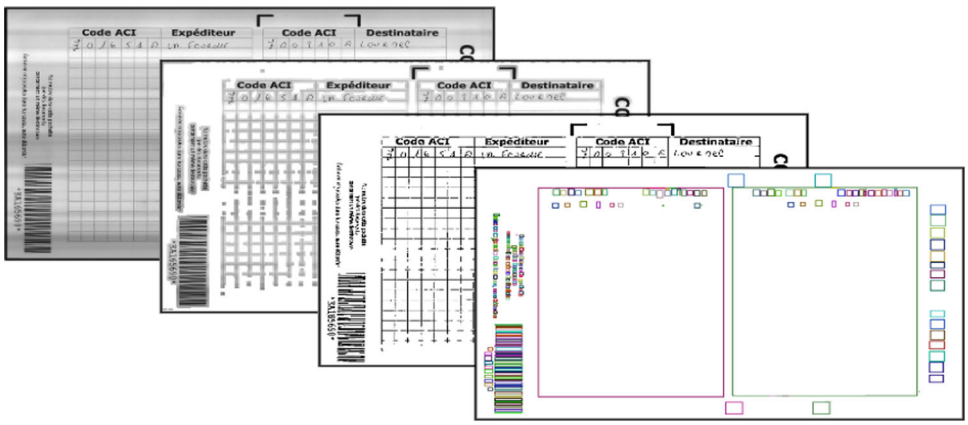

Fig. 8 Example of our hybrid approach of binarisation (text localization/thresholding) and bounding boxes of connected components

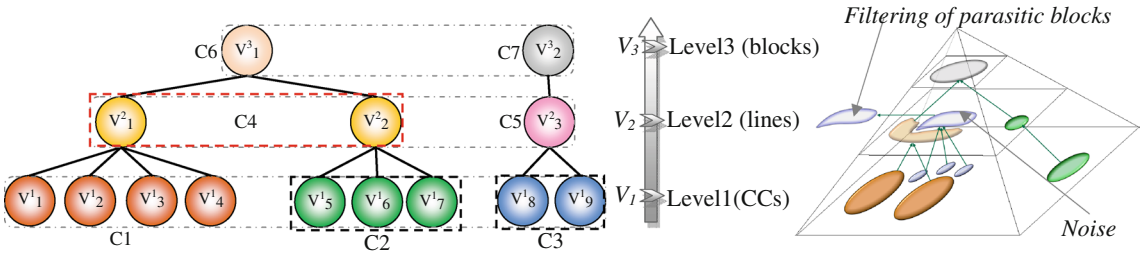

Fig. 9 Hierarchical graph coloring (c_i are the colors of nodes v_j^k at the level k)

to group objects of same type. The HGC is introduced, to correct the over (and/or under) segmentation of the documents into blocks and the b-coloring is used, to train the classifier to identify the type of document.

The key idea is to extract the layout by using a pyramidal strategy based on the graph coloring method. It allows to separate relevant elements and to group them into homogeneous classes and simultaneously reject irrelevant elements. By coloring of CCs, we separate textual regions from non-textual zones, and then we group the CCs of textual zones into text lines. The method is detailed in [27]. The coloring process uses a hybrid strategy of progression into the hierarchy of the graph: the colors of a level take part in the formation and the description of the nodes of the next level (see Fig. 9).

Let G be a non-oriented graph at three independent levels defined by the following relationship: $G(V,E) = \cup_{k=1}^{3} G_k(vd_{Lk}, E_{Lk>Sk})$ with $vd_{Lk} = \{vd_{Lk}(\mathrm{i})\}_{\mathrm{i}=1\ldots n_k}$ is the finite set of represented nodes starting from the descriptors (CC level: position, height, width, inter-character space, density; line level: position, line height, line width, inter-line space, density, alignment, eccentricity, overlapping degree, standard deviation; block level: position, block width, number of lines, eccentricity, spatial relations, density, uniformity, standard deviation). The set $V(L_k)|_{k=1,2 \text{ and } 3}$ of n_k constitutive elements of the data pyramid at level k (Fig. 9), and $E_{Lk>Sk}$ is the finite set of edges represented by the pairs of adjacent nodes. Taking into account the fact that each node is represented by a features vector, two nodes are then considered as adjacent if and only if their dissemblance $d_{i,j}$ (distance between their two features vectors) is strictly greater than the threshold S_k. The choice (or the training) of the optimal value of threshold is automated by using a ground truth (see the technique of supervised evaluation based on a segmentation of reference in [27]).

The effectiveness of our physical segmentation method has been tested on a set of 10,000 document images that were rejected by the old sorting system because of their layout complexity. More than 95% of documents are correctly segmented by our method, in opposition to 60% by RLSA method and 30% by profile projection method. The analysis of these results shows that several errors of over- or under-segmentation introduced by RLSA and profile projection methods can be considerably reduced by using our graph coloring-based method. These performances can be justified by the effectiveness of our method in extracting and separating the textual components (characters, lines and blocks of text) and by its ability to reject most of parasitic components. Thanks to this robustness, the HGC method is definitely more efficient for noisy images segmentation by comparison to classical approaches.

The increase of coherence between the different segmentation phases of our proposition led to a considerable reduction of processing time. To justify this assertion, we show in Fig. 10 the run-time comparison of our method and two standards: the RLSA and the profile projection.

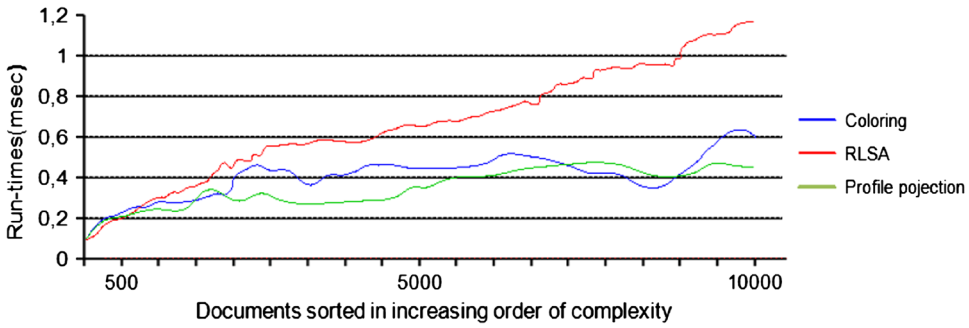

Fig. 10 Run-time comparison of three methods of physical layout extraction (ours is named *Coloring*)

4.2 Document feature extraction

The goal of feature extraction is to reduce as minimum as possible the size of information necessary for the document representation and to improve the document clustering into different homogenous classes. To reduce the processing time, this stage is applied progressively at each hierarchical level of the physical layout extraction. We describe each document with a reduced number of features computed from their layouts: 15 global features, which describe the entire document body; and 20 local features, measured on every text line item. Our features normalization technique uses the mean (μ) and standard deviation (σ) for each feature across a training set of documents to normalize each input feature vector. The normalization of each feature x_i is given by:

$$x_i^{'} = \frac{(x_i - \mu_i)}{\sigma_i} \tag{2}$$

- *The 20 local features* are extracted from each text line Li of the document. They combine the geometric features (number, average height and average width of connected components and width, area, eccentricity, position and skew angle of this line) and the spatial relationship between this line and the other lines in this document. The skew angle of every text line is measured during its formation from the second coloring of connected components. This angle is used to increase the robustness of this description to the skewed lines.
- *The 15 global features* are extracted from the physical layout of the entire document. These features are: global density, number of connected components, number of printed text lines, standard deviation of vertical or horizontal text lines alignment, standard deviation of printed text lines heights or widths, vertical or horizontal regularity of the profile projection, etc.

4.3 Representation of documents

The representation of each document pattern is based only on the description of its physical layout. We use two types of representations:

1. *Structural local representation* Each document j is described in the Rs^n space by a ranked sequence of n text lines: $Rs(j) = \left(L_1^j, L_2^j, \ldots, L_n^j\right)$ where the line L_t is represented by a feature vector of $p = 20$ dimensions $L_t = \left(x_1^t, x_2^t, \ldots, x_p^t\right)$.
2. *Global representation* Each document j is represented in the Rv^m space by a vector of m global features $Rv(j) = \left(y_1^j, y_2^j, \ldots, y_m^j\right)$.

4.4 Distances measures

To compare two documents, we combine two distances (D_{Rv} over Rv^{m} space and D_{Rs} over Rs^n space) given by the following equation:

$$DT = \gamma D_{Rv} + (1-\gamma) D_{Rs} \text{ with}$$
$$\gamma = \{k \in [0,1] \text{ which gives a maximum value of}$$
$$\Psi_k | k \in \{0, 0.1, 0.2, 0.3, \ldots, 0.9, 1\} \tag{3}$$

The value γ must be determined to maximize the quality of the classification Ψ (formulas 10 and 11). If two documents are separated by a small distance DT then they are similar.

The distance D_{Rv} between two documents, represented by the features $Rv(i)$ and $Rv(j)$ is given by the equation:

$$D_{Rv}[Rv(i), Rv(j)] = \left[\sum_{k=1}^{m} \left|y_k^i - y_k^j\right|^a\right]^{\frac{1}{a}} \text{ where } \alpha = 2 \tag{4}$$

The edit distance D_{RS} creates a spatial mapping between $Rs(i)$ of lines n_i and $Rs(j)$ *of* lines n_j by using a Dynamic Time Warping (DTW). DTW has been widely used to match 1D signals in the speech processing, bio-informatics, and also the online handwriting communities. DTW offers a robust comparison to small deformations often found in documents of the same family. These distortions are due to the confusion of a line of a handwritten text in capital letters with a line of a printed text. The nonlinear matching between $Rs(i)$ and $Rs(j)$ is described by the runs: $C = c_{1,}, c_2, \ldots, c$ with $c_k = (i_k, j_k)$ (Fig. 11).

The weighted sum of errors along of the optimal path C of the matching is given by:

$$D(c) = \frac{\sum_{k=1}^{K} d(c_k) \cdot w_k}{\sum_{k=1}^{K} w_k} \text{ with } d(c_k) = d(L_t^i, L_t^j) = \sqrt{\sum_{l=1}^{p} \left[x_l^t(i) - x_l^t(j)\right]^2} \quad (5)$$

where w_k is a positive weighting coefficient used as denominator to reduce the effect of K (number of the warping function points). ti_k and tj_k must be increasing functions and must satisfy some continuity conditions such as:

Monotony: $ti_k \geq ti_{k-1}$ *et* $tj_k \geq tj_{k-1}$
Continuity: $ti_k - ti_{k-1} \leq 1$ *et* $tj_k - tj_{k-1} \leq 1$
Limitations: $ti_1 = 1$, $tj_1 = 1$, $ti_K = n_i$ *et* $tj_K = n_j$.

The weighting coefficients are:

$$w_k = ti_k - ti_{k-1} + tj_k - tj_{k-1} \text{ and } \sum_{k=1}^{K} w_k = n_i + n_j. \quad (6)$$

In this case, the problem to solve becomes:

$$D_{Rs}[Rs(i), Rs(j)] = \frac{1}{n_i + n_j} \min_{C} \sum_{k=1}^{K} d(c_k) \cdot w_k. \quad (7)$$

The number of possible paths grows exponentially with the number of text lines within the documents we have to compare. This problem can be solved efficiently by the Dynamic Programming Algorithm (DPA) which finds an optimal matching between text lines. To save computation time, we do not compare all possible matching but only text lines which are spatially comparable (Fig. 12). We compute a cost limited along the diagonal in the table of the DPA (Fig. 11).

For each point in the space $Rs^{ni} \times Rs^{nj}$, simply find the best path that follows the continuity conditions and minimizes the contribution to the accumulation of global distance. It is therefore sufficient to study the allowed transitions and applying the local recursive relation:

$$f(1,1) = 2 \times d(L_t^1, L_t^1)$$

$$f(ti_k, tj_k) = \begin{Bmatrix} f(ti_k - 1, tj_k) + d(L_t^{ik}, L_t^{jk}) \\ f(ti_k - 1, tj_k - 1) + 2 \times d(L_t^{ik}, L_t^{jk}) \\ f(ti_k, tj_k - 1) + d(L_t^{ik}, L_t^{jk}) \end{Bmatrix} \quad (8)$$

$$\text{with} \begin{vmatrix} ti_k = 1 \ldots n_i \\ tj_k = 1 \ldots n_j \end{vmatrix}$$

$$D_{Rs}[Rs(i), RS(j)] = \frac{1}{n_i + n_j} f(n_i, n_j).$$

where $f(n_i, n_j)$ is the cumulative distance along the optimal path from the point of departure $(1, 1)$ to the point of arrival (n_i, n_j). f can be calculated from the path column by column or row by row.

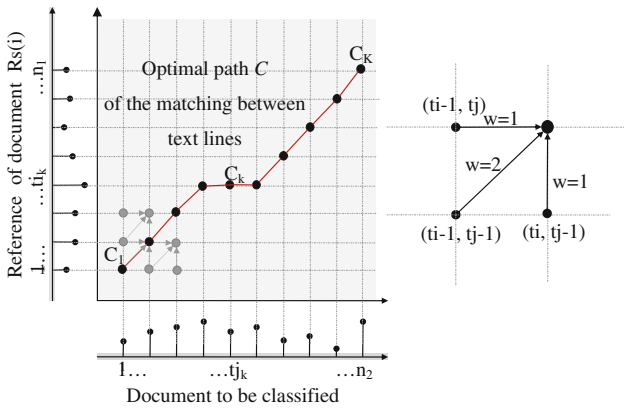

Fig. 11 Dynamic matching between text line sets of the unknown document and the reference document

4.5 Classification of documents

Graph coloring is also used for document classification. We represent a set R of N documents in a graph $G_{\geq \text{SDT}} = \left(V = \{v_1, \ldots, v_j\}, E_{\geq \text{SDT}}\right)$ where each node in the graph is the representation of document in R. Two different nodes v_i and v_j are adjacent if and only if the distance DT between the documents i and j is strictly superior to a threshold S_{DT}. The determination of this threshold is defined by the formula (9). The adjacency between the nodes is given by:

$$E_{\geq \text{SDT}}[v_i, v_j] = \begin{cases} 1 & \text{if } DT(v_i, v_j) > S_{\text{DT}} \\ 0 & \text{otherwise} \end{cases}. \quad (9)$$

To decompose the set R into homogeneous subsets, we colorize the graph G then we apply the algorithm of b-coloring described in [21, 27].

This b-coloring assigned to each node of $G_{\geq \text{SDT}}$ a color so that two adjacent nodes (the dissimilarity between the represented documents is higher than the S_{DT} threshold) do not carry the same color, and each color class must have at least one dominating node (node adjacent to all other colors). A classification associated with each S_{DT} threshold value is then returned with a supervised measure of this classification quality. The best classification uses the threshold that ensures maximum of classification quality ψ, returned by the following formula:

$$S_{\text{DT}}^{\text{Optimal}} = \underset{Si \in [S\min, S\max]}{\arg\max} \{\psi(S_i)\}. \quad (10)$$

The criterion ψ compares locally and globally the result of a coloring (or classification) C with the reference coloring C_{ref} called ground truth. This truth is defined by manually associating the class label to each node

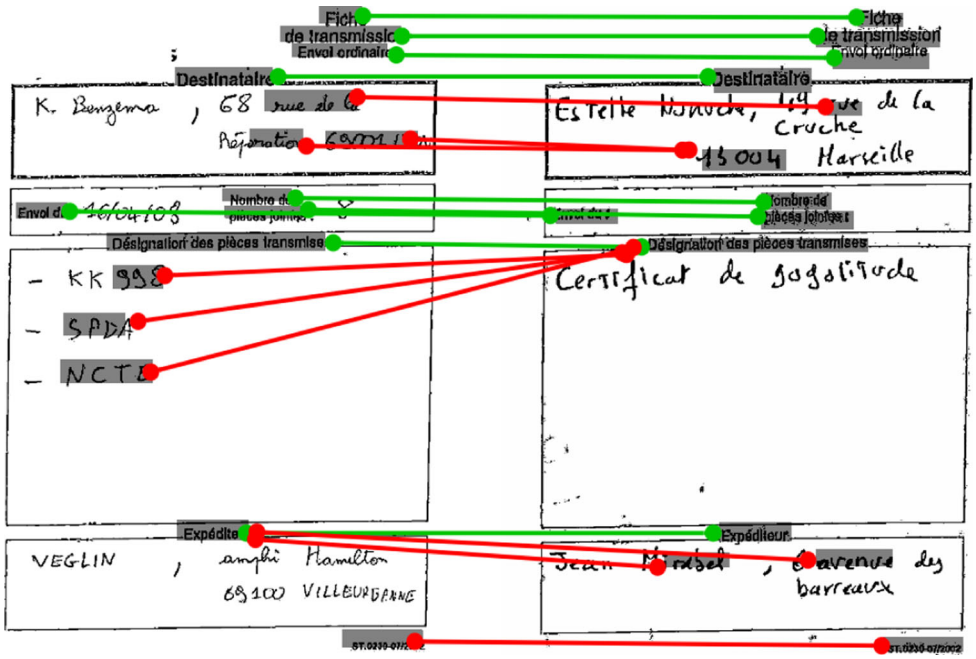

Fig. 12 Dynamic comparison of documents

(document). To measure ψ we adapt the criterion of Martin and al. [33] to our method as follows:

$$\psi(s_{\mathrm{DT}}) = Mg(C(G_{\geq \mathrm{SDT}}), C_{\mathrm{ref}}(G_{\geq \mathrm{SDT}})) = \frac{1}{n}\sum_{i=1}^{n} \min\{E_{\mathrm{RL}}[c(i), c_{\mathrm{ref}}(i)], E_{\mathrm{RL}}[c_{\mathrm{ref}}(i), c(i)]\} \tag{11}$$

where E_{RL} is the error of local refinement and is defined as follows :

$$E_{\mathrm{RL}}[c(i), c_{\mathrm{ref}}(i)] = \frac{\mathrm{card}[L\{c(v_i)\}] - \mathrm{card}[L\{c(v_i)\} \cap L\{c_{\mathrm{ref}}(v_i)\}]}{\mathrm{card}[L\{c(v_i)\}]} \tag{12}$$

where $L\{c(v_i)\}$ is the nodes set of the graph G that have the same color of node v_i, and $L\{c_{\mathrm{ref}}(vi)\}$ is the nodes set of G that have the same reference color of node v_i, and $C_{\mathrm{ref}}(v_i)$ is the reference color of v_i.

The quality criterion Mg, in its final form, take into account the global estimation of incorrectly colored nodes or confused and calculates, class by class, the misclassification using the E_{RL} local indicator.

4.6 Embedded learning mechanisms

During this step, we provide a training dataset R of $N = 512$ documents already classified into *14* classes. Our training approach uses graph b-coloring algorithm (detailed previously) to arrange the documents of the training set into homogeneous classes.

The prior grouping (labeling) of the training set of documents into 14 classes is given in the following table (Table 1; Fig. 13).

The following curve shows the value of classification criteria ψ (supervised evaluation) for each value of the adjacency threshold which varies in the interval]0, 1[by a step 0.02 (Fig. 14).

The best classification is given by the b-coloring that corresponds to the optimal threshold $S_{\mathrm{DT}} = 0.34$ (shown in Fig. 11). This optimal b-coloring provides automatically, to the training system output, a set of N^* dominating nodes

$$R^* = \{R_1^*, \ldots, R_{N^*}^*\} \tag{13}$$

representing the classes which are used for a real-time recognition of an unknown document (Fig. 15).

We compared the performance of our classification method based on graph b-coloring with two other classification methods (K-means and nonlinear SVM using Gaussian kernel) applied on the same training set of documents and features.

4.6.1 Why the K-means?

The K-means approaches are simple to implement and easily understood. They have relatively a low complexity compared to other classification methods [complexity in O(k.n), where n and k are, respectively, the number of documents to be classified and the number of classes]. In this type of methods the number of classes must be fixed in the beginning: they have very poor ability to classify noisy

Table 1 Prior labeling of the training set of documents

Classes	Documents	Classes	Documents
C1	1–64	C8	289–320
C2	65–128	C9	321–352
C3	129–160	C10	353–384
C4	161–192	C11	385–416
C5	193–224	C12	417–448
C6	225–256	C13	449–480
C7	257–288	C14	481–512

data or close to several classes simultaneously. The results depend strongly on the initial draw of the points representing the center of classes (Fig. 16).

4.6.2 Why the SVM?

SVMs are more advanced compared to the K-means, when the classes are not linearly separable; they consist in projecting the data into high-dimensional space by a transformation that is based on kernel Gaussian function. In this transformed space, classes are separated by linear classifiers that maximize the margin. The complexity of an SVM classifier will therefore not depend on the size of the data space, but the number of support vectors needed to achieve the separation, so the size of the training set. Moreover, these methods require a tedious step of labeling of all the documents of the training dataset (Fig. 17).

We have used the measure ψ to evaluate the confusion percentage, the relevance and accuracy of the classification of 512 documents for the training set obtained by each of the three methods (K-means, SVM and b-coloring). The more the classification is correct, the more this measure is close to 100%. The histogram below shows that the b-coloring gives a better classification compared to the other two methods (Fig. 18).

4.7 Recognition of the document class

The real-time recognition phase of the type of a document passing through a sorting chain exploits the learning result by b-coloring under the form of class representatives (dominating nodes). To perform this recognition, we compare the recognition results obtained by the three scenarios, with:

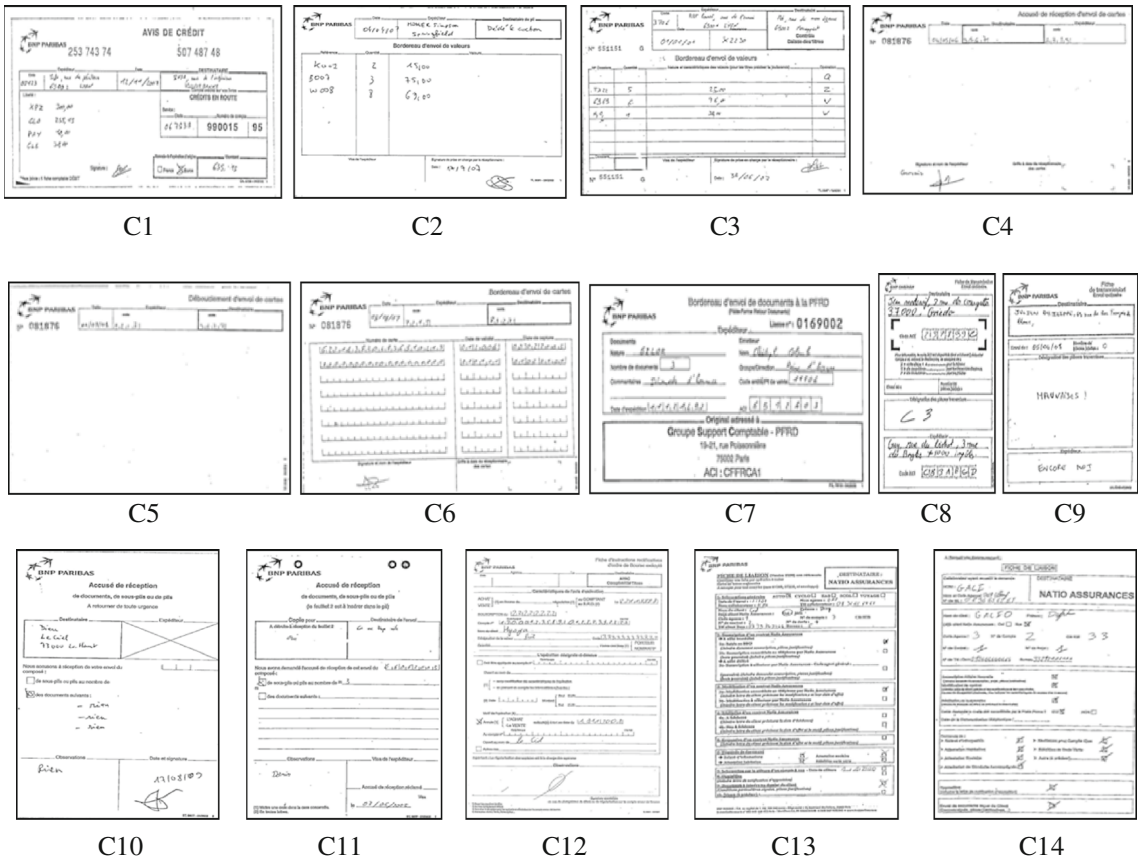

Fig. 13 Example of different classes of documents

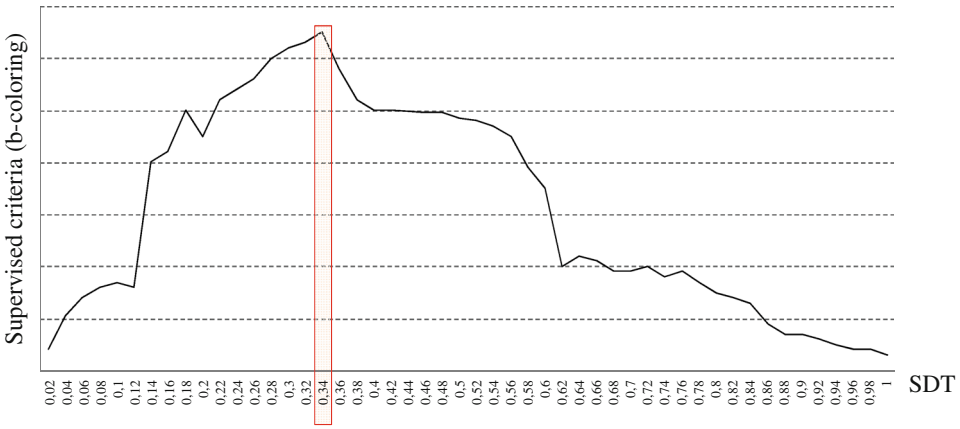

Fig. 14 The classification quality associated with each threshold, the peak in the curve represents the threshold that provides optimal quality of classification ($S_{DT} = 0.34$)

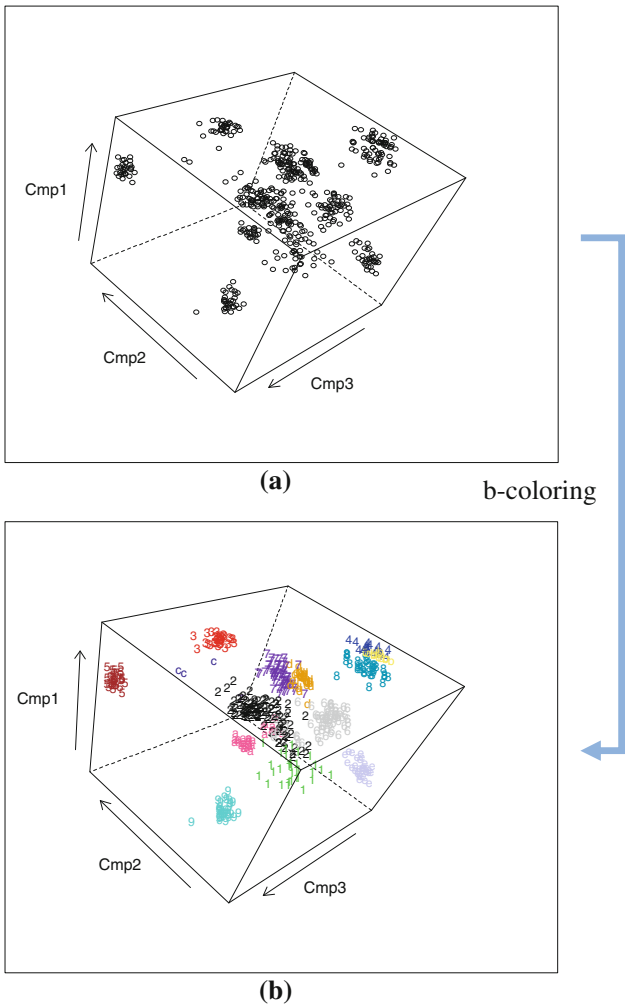

Fig. 15 **a** 512 documents projected in the feature space; **b** 14 clusters found by the b-coloring

Scenario 1 Minimum distance between classes (uses the dominating node as representative of the classes).
Scenario 2 Barycentric approach (each class is represented by the barycenter of its dominating nodes).

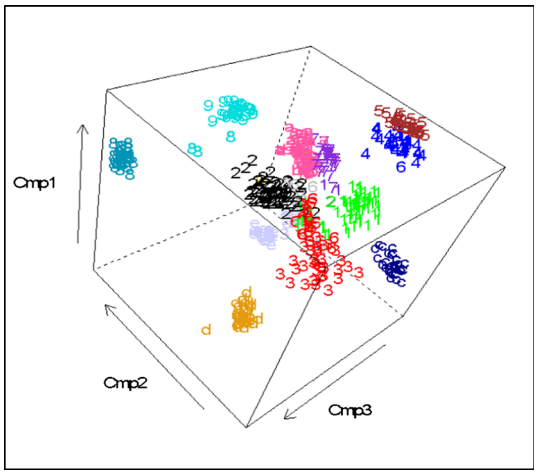

Fig. 16 3D Representation of the 14 classes that are formed by the K-means method on the training dataset R

For the first two scenarios the decision function is almost the same. Given an input document $T(i)$, the objective of the recognition system is to compare its description with those of all class representatives (nodes with the highest dominance or the dominating nodes barycenters of each class) of R^* from the learning phase (formula 12). The matching algorithm recognizes in real-time the type of document $T(i)$ from the nearest type in R^* as follows:

$$\text{Type}[T(i)] = \begin{cases} \text{Reject if } \arg\min_{k=1\ldots N^*} \left(\text{DT}\left[T(i), R_k^*\right]\right) > S_{\text{DT}} \\ \text{Type}\left(R_k^* | \arg\min_{k=1\ldots N^*} \left(\text{DT}\left[T(i), R_k^*\right]\right)\right) \text{ otherwise} \end{cases}. \quad (14)$$

The adjacency threshold S_{DT} also determines the knowledge of the classifier to reject the documents that it did not learn to recognize.

By way of illustration, the example in the figure below (Fig. 19) shows two documents (*T1* and *T2*) to recognize by using the dominating nodes numbered from 1 to 14 (representing the 14 classes) obtained during the learning

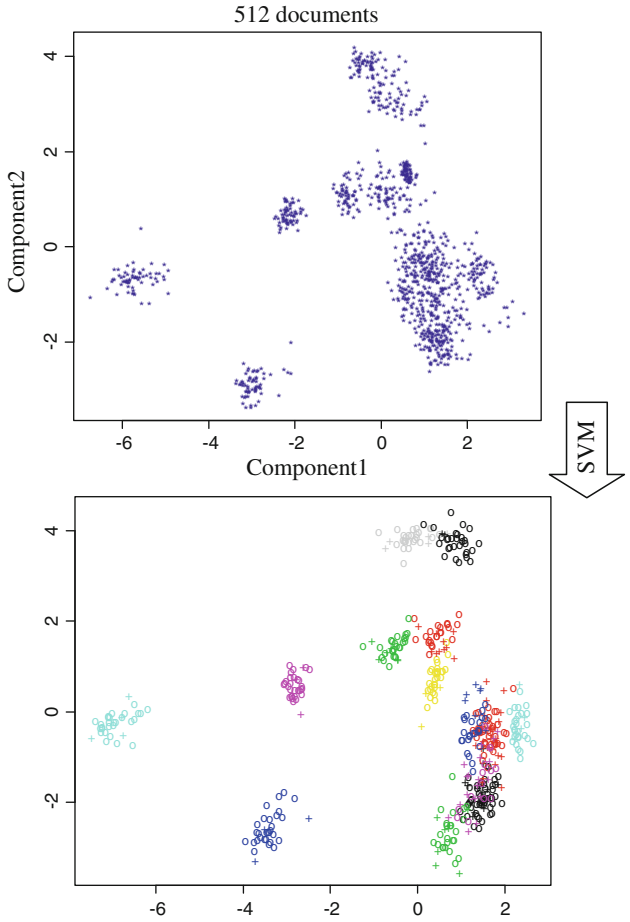

Fig. 17 Documents projection on the principal axes 1,2. 14 classes have been formed by the SVM from the training dataset (the 112 support vectors are presented by the "+" sign)

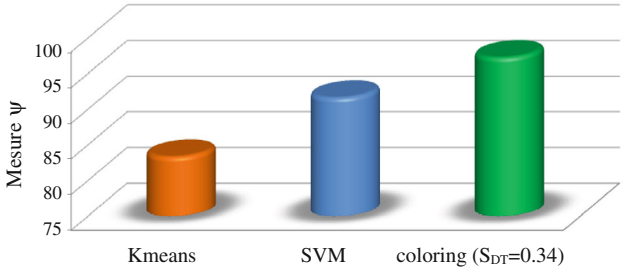

Fig. 18 Comparison of classification methods by using the quality measure ψ for each method [Ψ(K-means) = 83.32%, ψ(SVM) = 91.60%, ψ(b-coloration, $S_{DT} = 0.34$) = 97.32%]

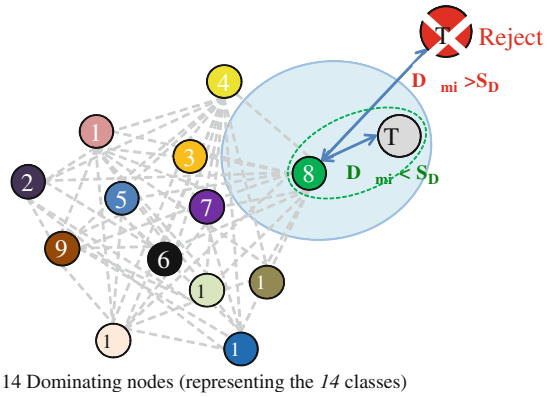

Fig. 19 Recognition or rejected documents examples by using the representatives of classes that are issued from learning by b-coloring

by b-coloring. The document *T1* is closer to the dominating node 8 with a distance less than S_{DT}. The distance of document *T2* with respect to the closest dominating nodes is greater than S_{DT}: document *T2* then must be rejected by the system.

Scenario 3 Choice of a neighborhood density function. Instead of using the barycenter of dominating nodes or the most dominating node as a unique prototype of a class, the method of *K* Nearest Neighbor involves the *kd* most dominating nodes of each class (experimentally $kd = 5$).

Table 2 Documents distribution of the test dataset on 16 classes of 576 documents. 14 learned classes (1–14) and 2 classes of rejection not learned (15 and 16)

Class	Document number	Class	Document number
C1	1–64	C8	289–320
C2	65–128	C9	321–352
C3	129–160	C10	353–384
C4	161–192	C11	385–416
C5	193–224	C12	417–448
C6	225–256	C13	449–480
C7	257–288	C14	481–512
C15	513–544	C16	545–576

5 Recognition and rejection evaluation

We have tested the three scenarios with a test dataset of 576 documents divided into 14 classes whose type has been learned and 2 classes whose type has not been learned (Table 2).

The curves in the figure below (Fig. 20) shows the recognition rate of the 14 known classes and their rejection rate on the two unknown classes according to the three scenarios. The curves show that the third scenario improves the recognition rate compared to the first two by effectively reducing the allocation errors and providing a better rejection decision on the nodes that the type is not learned during the learning stage.

We have finally compared the recognition performance of our method (that uses scenario 3) with respect to the methods that are based on the K-means and the SVMs. The curves in the figure below (Fig. 21) show the recognition

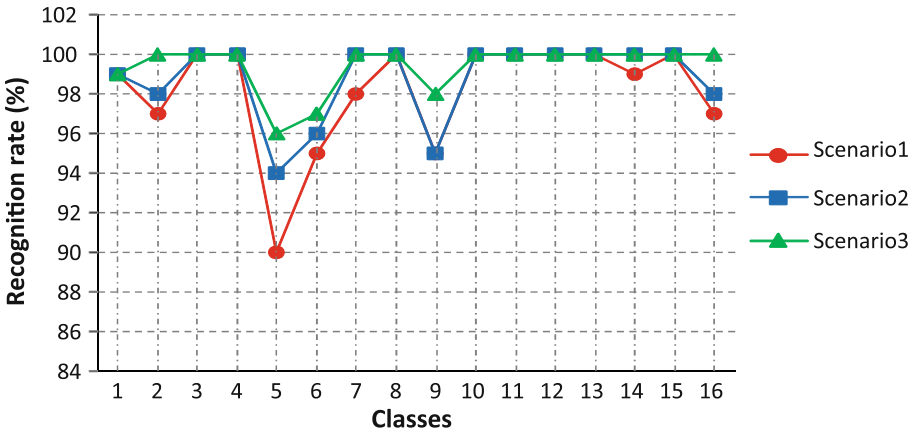

Fig. 20 Comparison of the recognition performance of the three scenarios

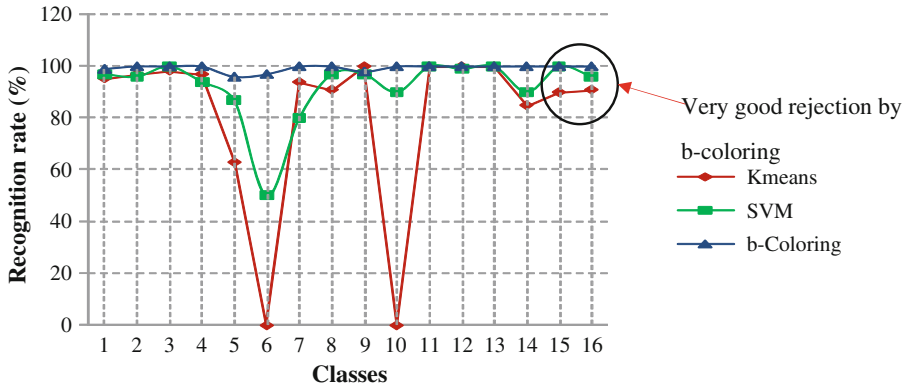

Fig. 21 Comparison of the three classifiers

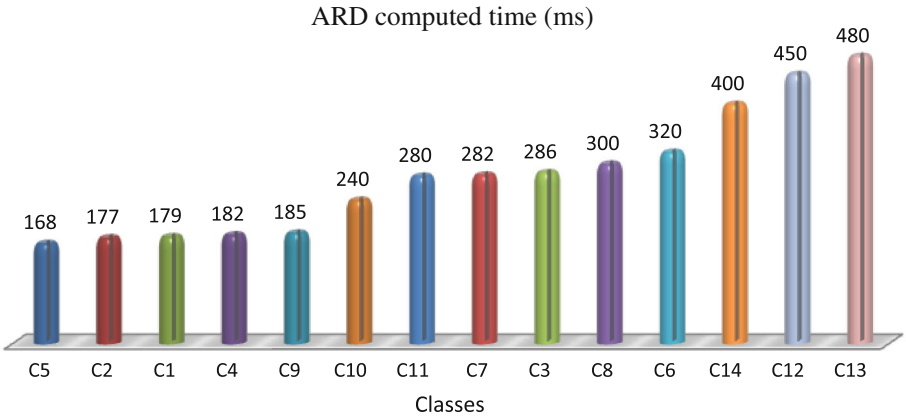

Fig. 22 Time needed at all the stages of the automatic recognition of the type of documents

rate on the 14 known classes and the rejection rate on the 2 unknown classes. The b-coloring gives better performance both in terms of recognition and the level of rejections. We note that the recognition system based on K-means fails to recognize the classes C_6 and C_{10}. This returns to the confusion of the class C_6 with the class C_3 and of the class C_{10} with the class C_{11} because of their similar physical structures. The recognition that is based on the SVMs presents a couple of confusions at the level of class C_6 while the b-coloring shows a higher reliability.

The following curve shows the average time required for binarisation, the physical layout extraction and recognition of the document nature of each class. For documents of high complexity (class C_{13}) time does not exceed 480 ms on a machine with 1 GB RAM and 1.6 GHz speed. On newer machines this time can be divided by four (Fig. 22).

6 Conclusion

We have presented a new method for the classification of business documents based on the hierarchical coloring of graphs. Documents are represented by their layouts. The hierarchical coloring of graph has been introduced during the layout analysis step to improve the robustness of the segmentation. The b-coloring has also been used during the training step to find the representative documents for each class. Because of the small constraints required by the b-coloring, this new method can be an answer to a large variety of classification problems. It can process documents having variable layouts and provide a real representation of document classes by using dominant documents. Moreover, the b-coloring allows the increase of the coherence between different phases of the ARD system and reduces the overall computation cost of the system. In future works, we propose to extend this method for the incremental training of the rejected documents. This new step will allow reclassifying documents which have been rejected by the system.

References

1. Mullot, R.: Les documents écrits de la numérisation à l'indexation par le contenu, pp. 365. Hermes Science Publication, Paris (2006)
2. Shin, C., Doermann, D., Rosenfeld, A.: Classification of document pages using structure based features. Int. J. Doc. Anal. Recognit. **3**(4), 232–247 (2001)
3. Heroux, P., Diana, S., Ribert, A., Trupin, E.: Classification method study for automatic form class identification. In: The 14th ICPR, Brisbane, Australia, pp. 926–929 (1998)
4. Esposito, F., Malerba D, Lisi, F.A.: Machine learning for intelligent processing of printed documents. J. Intell. Inf. Syst. **14**(2-3), 175–198 (2000)
5. Cesarini, F., Lastri, M., Marinai, S., Soda, G.: Encoding of modified X–Y trees for document classification. In: 6th ICDAR'01, pp. 1131–1136 (2001)
6. Baldi S., Marinai S., Soda G.: Using tree-grammars for training set expansion in page classification. In: 7th ICDAR'03, pp. 829–833 (2003)
7. Diligenti, M., Frasconi, P., Gori, M.: Hidden tree Markov models for document image classification. IEEE Trans. Pattern Anal. Mach. Intell **25**(4), 519–523 (2003)
8. Bagdanov, A.D., Worring, M.: First order Gaussian graphs for efficient structure classification. Pattern Recognit **36**(6), 1311–1324 (2003)
9. Dengel A., Dubiel, F.: Computer understanding of document structure. Int. J. Imaging Syst. Technol. **7**, 271–278 (1996)
10. Eglin, V., Bres, S.: Document page similarity based on layout visual saliency: application to query by example and document classification. In: The 7th ICDAR, Scotland, pp. 1208–1212 (2003)
11. Brugger, R., Zramdini, A., Ingold, R.: Modeling documents for structure recognition using generalized n-grams. In: 4th International Conference on Document Analysis and Recognition, ICDAR'97, vol. 1, pp 56–60 (1997)
12. Kochi T., Saitoh, T.: User-defined template for identifying document type and extracting information from documents. In: Proceedings of the 5th International Conference on Document Analysis and Recognition, Bangalore, India, 20–22 September 1999, pp. 127–130
13. Nattee, C., Numao, M.: Geometric method for document understanding and classification using on-line machine learning. In: Proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, USA, 10–13 September 2001, pp. 602–606
14. Liang, J., Doermann, D., Ma, M., Guo, J.K.: Page classification through logical labelling. In: Proceedings of the 16th International Conference on Pattern Recognition, Quebec, Canada, 11–15 August 2002, pp. 477–480
15. Yang Y., Liu X.: A re-examination of text categorization methods. In: Proceedings of the 22nd ACM SIGIR Conference, pp. 42–49 (1999)
16. Yang, J., Wang, S.: Extended VSM for XML document classification using frequent subtrees. In: Focused retrieval and evaluation. Lecture Notes in Computer Science, vol. 6203, pp. 441–448 (2010)
17. Lewis, D.D., Ringuetee, M.: A comparison of two learning algorithms for text categorization. In: Proceedings of 3rd Annual Symposium on Document Analysis and Information Retrieval, pp. 81–93 (1994)
18. Mohamed, H.K.: Automatic documents classification. In: IEEE ICCES'07, pp. 33–37
19. Sako, H., Seki, M., Furukawa, N., Ikeda, H., Imaizumi, A.: Form reading based on form type identification and form-data recognition. In: Proceedings of the 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland, 3–6 August 2003, pp. 926–930
20. Liang, J., Doermann, D.S.: Logical labeling of document images using layout graph matching with adaptive learning source lecture notes. In: Computer Science; Archive Proceedings of the 5th International Workshop on Document Analysis Systems V (DAS), vol. 2423, pp. 224–235 (2002) (ISBN:3-540-44068-2)
21. Effantin, B., Kheddouci, H.: A distributed algorithm for a b-coloring of a graph. In: IEEE ISPA'2006, Serrento, Italy (2006)
22. Paschos, V.: Optimisation combinatoire5: problèmes paradigmatiques et nouvelles problématiques, p. 270. Lavoisier, France (2007)
23. Gaceb, D.J., Eglin, V.: Improvement of postal mail sorting system. Int. J. Doc. Anal. Recognit. **11**(2),67–80 (2008)
24. Elghazel H., Hacid, M., Khedouci, H., Dussauchoy, A.: A new clustering approach for symbolic data: algorithms and application to healthcare data. BDA 2006, Lille, France
25. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. SMC **9**(1), 62–66 (1979)
26. Sauvola, J., et al.: Adaptive document binarisation. In: Document Analysis and Recognition, ICDAR, Proceedings of the Fourth International Conference, 18–20 August 1997, vol. 1, pp. 147–152
27. Gaceb, D.J., Eglin, V.: Address block localization based on graph theory. In: DRR XIV, SPIE, USA, pp. 12 (2008)
28. Pavlidis, T.: Structural Pattern Recognition, vol. 1, p. 302. Springer, Berlin (1977)
29. Drivas, D., Amin, A.: Page segmentation and classification utilising a bottom-up approach. In: Document Analysis and Recognition, ICDAR, Proceedings of the Third International Conference, vol. 2, pp. 610–614 (1995)
30. Shi, Z., Govindaraju, V.: Line separation for complex document images using fuzzy runlength. In: Document Image Analysis for Libraries, DIAL 2004, Proceedings, First International Workshop, pp. 306–312 (2004)
31. Déforges, O., Barba, D.: A fast multiresolution text-line and non text line structures extraction and discrimination scheme for document image analysis, ICPR 94, pp. 134–138 (1994)

32. Pavlidis, Z., Zhou, J.: A page segmentation and classification. CVGIP **54**(6), 484–496 (1992)
33. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: 8th International Conference on Computer Vision, July 2001, pp. 416–423

Author Biographies

Djamel Gaceb received engineer degree in Electronics, Signal and Image Processing Laboratory, from BLIDA University, Algeria, in 2002, and the master degree in computer science from Lyon I University, France, in 2005. From 2006 to 2009 he worked on automatic mail sorting system at CESA Company. He got a PhD in 2009 from INSA of Lyon in computer sciences. He is working since September 2009 as Temporary Teaching and Research Assistant (ATER) at the Lyon I University and is attached since 2005 to the LIRIS UMR 5205 laboratory. He is currently working on the topic of mail sorting and document recognition, retrieval and analysis, iPhone vision and real-time applications at the LIRIS laboratory.

Véronique Eglin graduated from the INSA of Lyon in 1995 and holder in 1998 of the PhD in computing science on "the document structure analysis", she is working since September 2000 as associate professor in the INSA of Lyon and is attached since 2003 in the LIRIS UMR 5205 laboratory. She has been contributing since 2003 to different digitization and valorisation projects of cultural inheritance. Her research domains are essentially centered on the characterization and the classification of handwritten and printed documents, the writer identification, the texture analysis for the typographies, and for documents layouts characterization.

Frank Lebourgeois graduated in 1987 from University of Lyon I with a master of science in mathematics then he got a PhD in 1992 at INSA de Lyon in computer sciences. He is currently an assistant professor in the LIRIS laboratory and works on document images restoration and analysis.