IJDAR (2007) 9:281–297
DOI 10.1007/s10032-006-0021-1

ORIGINAL PAPER

Using colour information to understand censorship cards of film archives

Oronzo Altamura · Margherita Berardi · Michelangelo Ceci · Donato Malerba · Antonio Varlaro

Received: 9 March 2005 / Revised: 11 October 2005 / Accepted: 28 May 2006 / Published online: 8 August 2006

Abstract Many European film archives are involved in the digitization of 20th century historical paper documents. In the context of the IST project COLLATE three of them were interested in the semi-automatic annotation of censorship cards and their subsequent retrieval on the basis of both annotations and content. Processing censorship cards, which is the main subject of this paper, leads to a number of challenges for many document image analysis (DIA) systems. Problems arise due to the low layout quality and standard of such material, which introduces a considerable amount of noise in its description. The layout quality is often negatively affected by the presence of stamps, signatures, ink specks, manual annotations and so on that overlap those layout components involved in the understanding or annotation processes. In order to effectively reduce the presence and the effect of noise, we propose an improved version of the knowledge-based DIA system WISDOM++ allowing it to take full advantage of the use of colour information in all processing steps: namely, image segmentation, layout analysis, document image classification and understanding. Experiments have been conducted on a corpus of multi-format documents concerning rare historic film censorships provided by the three film archives involved in the COLLATE project.

Keywords Historical paper documents · Color image segmentation · Inductive learning from examples

O. Altamura · M. Berardi · M. Ceci (✉) ·
D. Malerba · A. Varlaro
Dipartimento di Informatica, Università degli Studi,
via Orabona, 4, 70126 Bari, Italy
e-mail: altamura@di.uniba.it

M. Berardi
e-mail: berardi@di.uniba.it

M. Ceci
e-mail: ceci@di.uniba.it

D. Malerba
e-mail: malerba@di.uniba.it

A. Varlaro
e-mail: varlaro@di.uniba.it

1 Introduction

Nowadays the governments of many countries are interested in the valorisation of cultural heritage, since it is widely recognized that cultural heritage resources have significant implications for development (both as a knowledge basis and in terms of commercial exploitation). For this aim, many institutions which collect and preserve cultural heritage have shown a great interest in the digitalization of their resources and in the exploitation of mechanisms to provide online access to digitalized products.

According to the definition reported in the 1972 UNESCO "World Heritage Convention" - Article 1- cultural heritage refers to "monuments", "groups of buildings" and "sites" which are of outstanding universal value historically, artistically or scientifically. However, the concept of cultural heritage has recently assumed a broader connotation and includes, among other things, tangible, moveable objects such as works of art, artefacts, scientific specimens, photographs, books, manuscripts and recorded moving image and sound.

Historical documents are a kind of cultural heritage for which several research projects have recently been promoted for the purposes of preservation, storage, indexing and on-line fruition.

One of the first EU-funded projects is MASTER [17] that has developed a standard for computer-readable descriptions of medieval manuscripts. The specification of metadata (e.g. coloured drop capitals, original text, text from the second copyist, illustrations) has led to the development of a document image analysis (DIA) system that localizes all connected components and then classifies them according to the features selected by users. MEMORIAL [3] is another EU-funded project whose goal is the establishment of a digital document workbench enabling the creation of distributed virtual archives of typewritten documents related to prisoners in World-War II concentration camps. Templates of physical and logical entities on a document guide the application of the OCR to specific locations in the image, thus producing a "filled" document structure (XML file). Two national projects strictly related to DIA are Bovary [28] and D-SCRIBE [10,11]. The former concerns the digitalization of 5,000 original manuscripts handwritten by the French writer Gustave Flaubert. The non-linear arrangement of the text and numerous editorial marks define a complex structure layout whose extraction is performed by a set of hand-coded layout rules. D-SCRIBE aims to develop an integrated system for digitization and processing of Old Greek manuscripts. Binarization of original grey-scale images [11] and character ligature detection are the two main challenges.

This paper presents some issues encountered and problems addressed in the EU-funded project COLLATE (Collaboratory for annotation, indexing and retrieval of digitized historical archive material), whose main goal is that of providing professional users operating in film archives with adequate access to historic film-related documents and their associated metadata [9]. The cultural material consists of digitized versions of documents on European films in the 1920s and 1930s. Such documents can be censorship documents, newspaper articles, posters, advertisement material, registration cards and photos that cannot be used for access, indexing and retrieval as they are. Therefore, methods have to be applied to extract as much information from both textual and pictorial material and make them as machine-accessible as possible. This implies going beyond mere OCR techniques for textual documents, and applying "intelligent" DIA techniques in order to extract both logical and semantic content.

In the framework of the COLLATE project we investigated the applicability of the DIA system WISDOM++[1] [2] to a portion of the original collection of digitized document pages available in three national film archives, namely Deutsches Filminstitut (Germany), Filmarchiv Austria and Národní Filmový Archiv (Czech Republic). Specifically, censorship cards that share a common layout and logical structure have been selected for our pilot study, since the document structures can be automatically built by means of machine learning methods, instead of being manually specified as in MEMORIAL. Indeed a distinctive feature of WISDOM++ is the extensive usage of machine learning tools in order to automatically extract useful knowledge from training examples.

WISDOM++ was originally developed to fully support the transformation of multi-page printed documents into XML format. Since most of information in the documents is typewritten, the system appeared to be a potentially useful tool for the conversion of scanned documents into XML format. Nevertheless, some problems which arose in processing the available historical documents have shown that a more sophisticated approach was required. The low layout quality and standard of such material introduces a considerable amount of noise in its description. The layout quality is often negatively affected by both the degradation of the documents and the presence of frames, stamps, signatures, ink specks and manual annotations that overlap those layout components involved in the understanding or annotation processes.

To effectively process these documents, it is necessary to exploit information conveyed by colour since signatures, stamps and manual annotations are often characterized by colours which are different from those present in the background and typewritten texts. Actually, this consideration does not apply specifically to our application domain, but to historical documents in general, since colour information permits DIA systems to catch differences that, otherwise, cannot be adequately represented. First, it is possible to identify noise on the basis of colour homogeneity. This is particularly useful when the original document presents stains, tears and has an irregular accumulation of dirt due to repeated handling [4]. Second, it is possible to isolate and consider separately overlapped interesting blocks. This aspect is particularly interesting in legal documents, where blue stamps or revenue stamps often overlap signatures or typewritten text. Third, it is possible to isolate interesting blocks from uninteresting ones. This is particularly useful when there are manual annotations (typically of a different colour) that overlap those layout components involved in the understanding or annotation processes.

A naïve approach to colour document image processing would be to separate different colours by means of a well-established colour quantization method and to process images corresponding to each colour separately, as

[1] http://www.di.uniba.it/∼malerba/wisdom++/

if they were many independent black and white images. However, this approach is based on the simplified assumption that a logical component can be associated with a single colour. In practice, this assumption is rarely true. First, historical documents tend to deteriorate: paper colour darkens with age, while handwritten or typed printed parts tend to fade [25]. These two factors acting simultaneously narrow the discrimination gap between background and textual components. Second, when the document is written or typed on both sides, and the back side is visible from the front side, the system should be able to filter out the introduced noise. Third, the locality of components should be taken into account because some portions of a document page require different treatment with respect to others. For this reason, a more sophisticated approach is necessary.

In the literature, several colour segmentation algorithms have been proposed. They make use of several different clustering techniques such as "histogram-based" [38], "Euclidean Minimum Spanning Trees" [39], "Fuzzy c-means" [5]. However, according to Cheng [7], "the image segmentation is basically one of the psychophysical perception, therefore not susceptible to a purely analytical solution". Therefore, there are many methods for image segmentation and each method is tailored for a particular kind of image and a particular kind of application [21]. We also observe that most of proposed methods only operate in colour space (typically on the original RGB (red blue green) space representation [35]) and do not take any spatial information into account [31]. This turns out to be a severe limitation in document layout analysis, where colours should be associated to layout components. Notable exceptions are the works [34] and [13] on the identification of textual components, and [14] on the identification of stamps in historical documents. Nevertheless, these works are based on the assumption that a layout component is associated to a single colour. On the other hand, in our domain, where not only textual components are of interest, it is necessary to provide the system with the capability to also identify multicolour logical components such as pictures or revenue stamps.

Two works that combine colour and spatial features to identify multicolour blocks are reported in [27] and [15]. In the former segmentation is performed by a fuzzy region growing method that aims at capturing variations of the same colour. However, in our case, multicolour logical components may be characterized by the presence of completely different colours. The work by Karatzas and Antonacopoulos faces the problem of text segmentation of web images, such as banners, headers and illustrations. The limited applicability to textual components and the absence of problems typical of historical documents, justify the investigation of different solutions.

WISDOM++, originally developed to process black-and-white (binary) TIFF images, has been substantially extended to take full advantage of colour information in all processing steps, namely, image segmentation, layout analysis, document image classification and understanding. In particular, colour quantization, obtained by standard libraries, is followed by a preliminary step that separates foreground from background colours. Afterwards, the black/white projection of each foreground colour is generated and the classic RLSA segmentation algorithm is applied. Subsequently, a merging phase, based both on spatial and colourimetric measures is performed in order to group together blocks of different colours that should contribute to the identification of single layout components. Blocks are then classified according to the type of content and the layout analysis is performed to detect structures among blocks. Taking into account the information on the colour, the result of the layout analysis allows the system to identify overlapping components. A proper logic description of layout structures is generated and used to train a machine learning system to classify and understand document images. In this step, colour information helps both to isolate noise and to identify better colour-dependent components of interest. Actually, the use of a richer representation also raises efficiency problems in the learning phase, which are resolved by the integration of some caching techniques in the learning strategy.

All these aspects are explained in detail in the following sections, where the new document processing steps implemented in WISDOM++ are reported. The new layout analysis method is illustrated in Sect. 2, while the document understanding process is described in Sect. 3. The new system has been tested on a collection of multi-format censorship cards of historic films from the 20s and 30s. The goal is to evaluate the effectiveness of colour information in the complex process of layout analysis and document understanding. Results are reported and discussed in Sect. 4. Finally, conclusions are drawn in Sect. 5.

2 Colour-based layout analysis

In WISDOM++ the paper document process is organized in several steps:

WISDOM++ makes an extensive use of knowledge and XML technologies for semantic indexing of paper documents. This is a complex process involving several steps:

1. The image is segmented into basic layout components (basic blocks), which are classified according to the type of content (e.g. text, pictures and graphics).
2. A perceptual organization phase (layout analysis) is performed to detect a tree-like layout structure, which associates the content of a document with a hierarchy of layout components.
3. The first page is classified to identify the membership class (or type) of the multi-page document (e.g. type I or II of a censorship card).
4. The layout structure of each page is mapped into the logical structure, which associates the content with a hierarchy of logical components (e.g. film title and censurer name).
5. OCR is applied only to those logical components of interest for the application domain (e.g. film title).

The result of the process is an XML file that represents the layout structure, the logical structure, and the textual content returned by the OCR for some specific logical components.

Four out of five processing steps make use of explicit knowledge expressed in the form of decision trees and rules which are automatically learned by means of two distinct machine learning systems: ITI [36], which returns decision trees useful for block classification (first step), and ATRE [22], which returns rules for layout analysis correction (second step) [23], document image classification (third step) and document image understanding (fourth step) [24].

In this section we focus our attention on the first two steps, while details on the subsequent two steps are reported in the next section. For the last processing step, an off-the-shelf OCR software has been integrated in the current version of WISDOM++ with the aim of reading texts from segments of the original colour image. Indeed, the usage of German and Czech dictionaries and the variety of fonts do not permit the application of experimental OCR programs.

The colour image segmentation algorithm implemented in WISDOM++ operates in three steps. First, a colour reduction is performed by means of the simple octree-based clustering algorithm that performs a quantization of the colours [12]. Second, a colour merging strategy is applied to further reduce the number of colours. Third, overlapping regions are merged together on the basis of spatial criteria. This multi-step approach is similar to those proposed by Tremeau et al. [35] and Lucchese and Mitra [20]: the main difference is that colour homogeneity of spatially contiguous pixels is not the only criterion, since in our application it is also important to isolate multi-colour blocks.

Algorithm 1 Top-level pseudocode of the segmentation algorithm

```
 1: procedure segment(OriginalImage, th_InclSM,
        th_IntSM, th_MinOcc, th_MaxOcc) Output: list of images
 2: ReducedImage←quantization(16,OriginalImage);
 3: List←generateBiColourImages(ReducedImage,List);
 4: List←removeBackground(List);
 5: for all ForegroundImage ∈ List do
 6:    BBSets ← BBSets ∪
          RLSASegmentation(ForegroundImage);
 7: end for
 8: List←ColourimetricMerging(List, BBSets);
 9: for all Foreground1 ∈ List do
10:    for all Foreground2∈List–Foreground1 do
11:       BBSet1←ApplyRLSASegmentation(Foreground1);
12:       BBSet2←ApplyRLSASegmentation(Foreground2);
13:       IntersectingBlocks←ComputeIntersections(BBSet1,
              BBSet2);
14:       List← SpatialMerging(List,IntersectingBlocks,
              th_InclSM, th_IntSM,th_MinOcc,th_MaxOcc);
15:    end for
16: end for
17: return List
```

In Algorithm 1, a high-level description of the algorithm is reported. In the first phase, the quantization algorithm reduces the number of colours and the user can manually select background colours. Later on, the algorithm works on a *List* of binary images. Each binary image corresponds to a foreground colour. For each binary image, 1 corresponds to the foreground colour and 0 corresponds to either the background colours or to the other foreground colours.

Each binary image is segmented by applying a variant of the run length smoothing algorithm (RLSA) [37], which scans the image only twice, with no additional cost [33]. Furthermore, the smoothing parameters are adaptively defined [2]. The output of RLSA, when applied on each binary image, is a list of rectangular blocks (*BBSet*).

Once the set of basic blocks, for each binary image, has been extracted, the colourimetric merging is performed. This merging step aims to cluster colours associated to binary images on the basis of a colourimetric distance measure. Binary images whose colours belong to the same cluster are removed from the *List* and replaced by the merging result.

The second merging step is performed on the updated list of binary images, taking images in pairs. Images are segmented again and the spatial merging of intersecting blocks is performed. The result is an updated list of both binary and multi-colour images.

Both merging steps take into account only the pixels contained in the set of basic blocks. Noise pixels that do not contribute to the identification of a basic block are ignored.

In the following subsections, the three steps of colour image segmentation as well as the layout analysis procedure are described in detail.

2.1 The quantization process

The quantization process follows the method proposed by Gervauz and Purgathofer [12], whose basic idea is to build a tree structure always containing a maximum of K different leaves, where each leaf corresponds to a colour.

The tree represents the RGB colour space. In level i Bit i of each node the red, green and blue binary value is used as the selector for the successors. So, an inner node stores a triple of bits, one for each basic colour. Since each red, green and blue value is between 0 and 255, the maximum depth of the tree is eight and an internal node has at most 8 successors. The leaves keep information of the colour value, the colour index and a counter for the pixels that are already mapped into that leaf.

The image is read twice. The first time, colours are iteratively added to the tree. When the tree already contains K colours and a further colour has to be added to the tree, its colour value has to be merged with the most likely one that is already in the tree. Both values are substituted by their frequency-weighted mean and the counter is updated.

For our application domain, we set $K = 16$ since humans normally do not "see" more than nine colours in a censorship card. Obviously, human perception is biased from background knowledge on the different sources of colours (typewritten text, manual annotations, stamps, signatures). Nevertheless, this should be considered as a clear indication that a much finer discrimination among colours would be useless.

It is noteworthy that the tree structure depends on the sequence of the colours to be added. However, this aspect is marginal in our application, where the number of selected colours is relatively small and frequent colours suddenly tend to prevail over others.

Quantization is performed at the second reading of the image. The colour values of each pixel are used for traversing the octree data structure and the search along a path through the tree is finished when a leaf node is reached. Thus, the K-colours image is transformed into K different binary images, each of which is related to a colour expressed in RGB colour space. Among the K colours, the user manually selects a subset of m background colours. In practice, we observed that m ranges between two and three in our application domain because of the various forms of degradation of the documents. The exploration of the automatic selection of background colours is postponed for future research.

2.2 Colourimetric merging

The list of K-m images is used in the merging phase. The first merging process is a colourimetric merging which aims to merge those binary images whose colours can be considered to be light variations. This is a necessary step, since the value K fixed a priori in the quantization process may turn out to be too large for a specific document.

The process is based on a hierarchical clustering algorithm. At each step, the dissimilarity between two clusters of colours (inter-cluster dissimilarity) is evaluated on the basis of two measures: (a) the lowest Euclidean distance between two colours taken from distinct clusters (nearest neighbour-based dissimilarity); (b) the Euclidean distance between the centroids of the two clusters (centroid-based dissimilarity). Two clusters of colours are merged when both computed measures are lower than a threshold. Clusters whose nearest neighbour-based dissimilarity is lowest are considered first. The threshold is automatically defined as the standard deviation value computed by considering all the distances between each colour of one cluster and each colour of another cluster. In extreme cases, all K-m non-background colours can be grouped together, or else none of them can be grouped. Finally, for each remaining cluster a new image representing the average colour of original images is generated.

All distances are computed in the CIELab space. CIELab space is obtained by a non-linear transformation of the original RGB space. In CIELab space, the principal dimension is luminosity and colours are disposed in two dimensions, such that green and red as well as blue and yellow are opposite. We used CIELab for a twofold reason: it takes hue, value and chroma into account and is considered "visually uniform" because adjacent colour samples represent equal intervals of visual perception [7]. In Fig. 1 the result of the colourimetric merging is shown.

2.3 Spatial merging

The degree of overlapping of the layout extracted from different colour images can also be an indication of the fact that two images have to be merged together. By taking into account this spatial information that is ignored in colourimetric merging, it is possible to group together multi-colour blocks and remove some useless low-density blocks (containing few pixels), that generally capture colour shades of the same layout component.

Spatial merging operates on the results of RLSA when it is applied to the possibly reduced set of binary images determined by colourimetric merging. For each

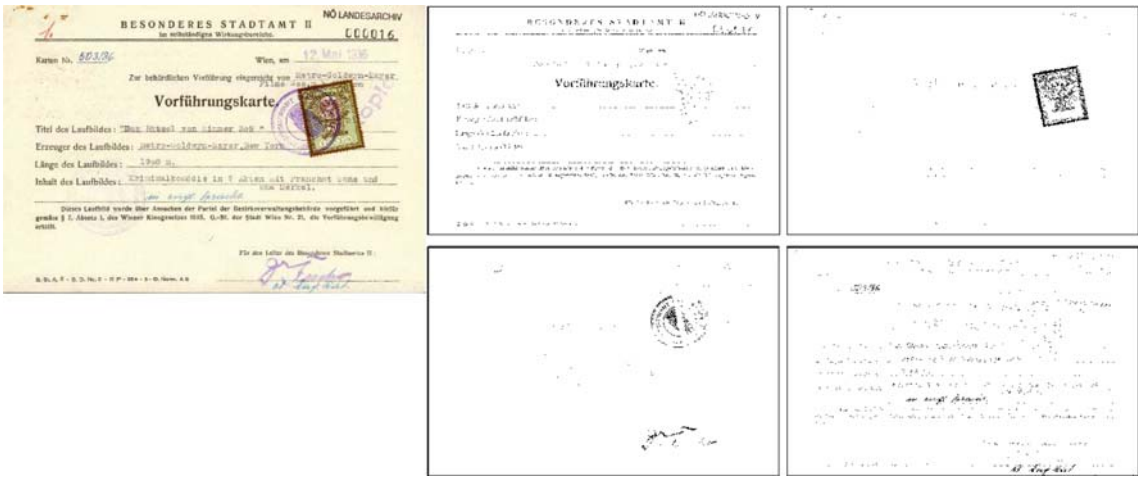

Fig. 1 Result of the colourimetric merging: (*left*) the original multi-colour image of a censorship card and (*right*) the four binary images

couple of binary images, intersecting blocks are identified and eventually merged on the basis of three perceptual criteria hereinafter specified. The first criterion aims to identify multicolour blocks. The second criterion aims to extend multicolour blocks by enclosing overlapping spurious blocks from binary images. The third criterion aims to extend binary images by enclosing overlapping spurious blocks from other binary images.

The first criterion can be summarized by the production rule in algorithm 2. It identifies multicolour layout components, such as revenue stamps. For each couple of intersecting blocks, when the percentage of intersection exceeds a user-defined threshold (*th_IntSM*) and the percentage of occupation (defined as the ratio between the area of the block and the entire image area) for both candidate blocks is in the interval [*th_MinOcc*, *th_MaxOcc*], then a new multicolour image is generated. The new image is built as the union of partitions of the original images enclosed in the blocks. Original binary images are also "cleaned" by removing pixels added to the multicolour image.

The second criterion (algorithm 3) is based on the rationale that if a block strongly overlaps a block of a multicolour image, the intersecting part has to be considered as composing the multicolour block. The pixels enclosed in the intersection are removed from the binary image (*ImgForeground2*) and added to the multicolour image (*ImgForeground1*).

The third criterion (algorithm 4) aims to extend binary images by enclosing overlapping spurious blocks from other binary images. The first rule is based on the rationale that if a small block has a high degree of overlap with a block of another image, it has to be considered a spurious block and included in the image associated to the "predominant" block (*ImgForeground1*).

Algorithm 2 First Criterion

1: **Given: BBx** ∈ **BasicBloks(ImgForeground1),**
BBy ∈ **BasicBloks(ImgForeground2)**
2: **if** (perc_of_intersection(BBx,BBy)> *th_IntSM* &&
th_MinOcc <perc_of_occupation(BBx)< *th_MaxOcc*&&
th_MinOcc <perc_of_occupation(BBy)< *th_MaxOcc*) **then**
3: ImgForeground1←removeArea(ImgForeground1,BBx);
4: ImgForeground2←removeArea(ImgForeground2,BBy);
5: NewForeground ← GenerateMulticolour(BBx, BBy);
6: List ← addElement(List, NewForeground);
7: **end if**

Algorithm 3 Second Criterion

1: **if** perc_of_intersection(BBx,BBy) > *th_IntSM* &&
perc_Inclusion(BBx,BBy) + perc_Inclusion(BBy,BBx)
≥ *th_InclSM* && Multicolour(ImgForeground1) **then**
2: ImgForeground1←addArea(ImgForeground1,
ImgForeground2, BBy∩BBx);
3: removeArea(ImgForeground2, BBy∩BBx);
4: **end if**

Algorithm 4 Third Criterion

1: **if** perc_of_intersection(BBx,BBy)> *th_IntSM* &&
perc_Inclusion(BBx,BBy) + perc_Inclusion(BBy,BBx)
≥ *th_InclSM* && perc_of_occupation(BBx) < *th_MinOcc*
then
2: ImgForeground1←
addArea(ImgForeground1, ImgForeground2, BBy);
3: removeArea(ImgForeground2, BBy);
4: **end if**
5: **if** perc_of_intersection(BBx,BBy) > *th_IntSM* &&
perc_Inclusion(BBx,BBy) + perc_Inclusion(BBy,BBx)
≥ *th_InclSM* && density(BBx) < density(BBy) **then**
6: ImgForeground2← addArea(ImgForeground2,
ImgForeground1, BBy∩BBx);
7: removeArea(ImgForeground1, BBy∩BBx);
8: **end if**

The second rule states that if two blocks have a high degree of overlapping, then the intersecting part of the block with lower density (non-predominant block) has to be added to the image of the predominant one. The density of a block is defined as the ratio between the number of pixels contained in a block and the area of the block.

Like most of the algorithms for colour segmentation proposed in the literature, our algorithm allows the user to set thresholds [35]. In particular, we use four different thresholds listed in the following:

- *th_IntSM* defines the minimal intersection percentage for merging blocks.
- *th_InclSM* defines the minimal inclusion percentage for merging blocks.
- *th_MinOcc* and *th_MaxOcc* define the range of occupation for merging blocks.

All the values are user-defined and depend on the specific type of processed documents. Although it is possible to find the optimal value of these parameters on the basis of a training set of documents, this aspect has not been explored in this work.

At the end of the spatial merging process, *List* contains the final list of binary images. The RLSA segmentation is applied to each image separately (if not yet computed) and each RLSA execution returns a set of rectangular blocks that are grouped together in a single set of basic blocks. Each basic block can be either associated to a single colour or labelled as multicolour.

2.4 Layout analysis

The segmentation algorithm returns (possibly) overlapping blocks that may contain either textual or graphical information and are either single colour or multicolour. A first step towards the reconstruction of the layout structure consists of classifying the blocks according to their content type: text, horizontal line, vertical line, picture (i.e. halftone images) and graphics (e.g. line drawings). The classification of blocks is performed by means of a decision tree automatically built from a set of training examples (blocks) of the five classes.

The layout structure is built by exploiting not only the result of the classification of basic blocks and their geometrical features, but also the colour information obtained during the segmentation process. WISDOM++ combines the top-down segmentation of the image into basic layout components with a bottom-up layout analysis method to assemble basic blocks into larger components. More precisely, the layout structure is extracted in two steps:

1. A *global* analysis of the document image to determine possible areas containing paragraphs, sections, columns, figures and tables. This step is based on an iterative process, in which the vertical and horizontal histograms of text blocks are alternatively analysed in order to detect columns and sections/paragraphs, respectively.
2. A *local* analysis of the document to group together blocks which possibly fall within the same area. Four perceptual criteria are considered in this step: *proximity* (e.g. adjacent components belonging to the same column/area are equally spaced), *continuity* (e.g. overlapping components), *similarity* (e.g. components of the same type, with an almost equal height) and *colour* (i.e. components of the same colour). Pairs of layout components that satisfy some of these criteria are grouped together. The new layout component corresponds to the minimal bounding box of all its constituents. It is noteworthy that grouping affects either pairs of layout components extracted from the same binary image (i.e. with exactly the same colour) or pairs of layout components labelled as multicolour. Therefore, the colour associated with a new layout component is univocally determined from its constituents. On the other hand, it is possible to group together layout components with different content type (e.g., text and graphics). When the content type of constituent blocks is homogeneous, the same type is inherited by the new logical component; otherwise, the associated type is set to *mixed*.

The layout structure extracted for each document page is a hierarchy with five levels: *basic blocks*, *lines*, *set of lines*, *frame 1* and *frame 2*. In Fig. 2 blocks of the *frame 2* level, for each final colour are shown.

3 Document image understanding

Document image understanding, that is, mapping the layout structure of a document page into a corresponding logical structure, is based on the assumption that documents can be understood by means of their layout structures alone. For instance, the date of a censorship decision is usually located at the top of the first page of a censorship card, while the signature of the censurer is at the bottom of the last page. The mapping of the layout structure into the logical structure can be performed by means of a set of rules.

Generally, logical structures depend on the particular class of documents. For instance, the censorship card provided by Deutsches Filminstitut in the COLLATE

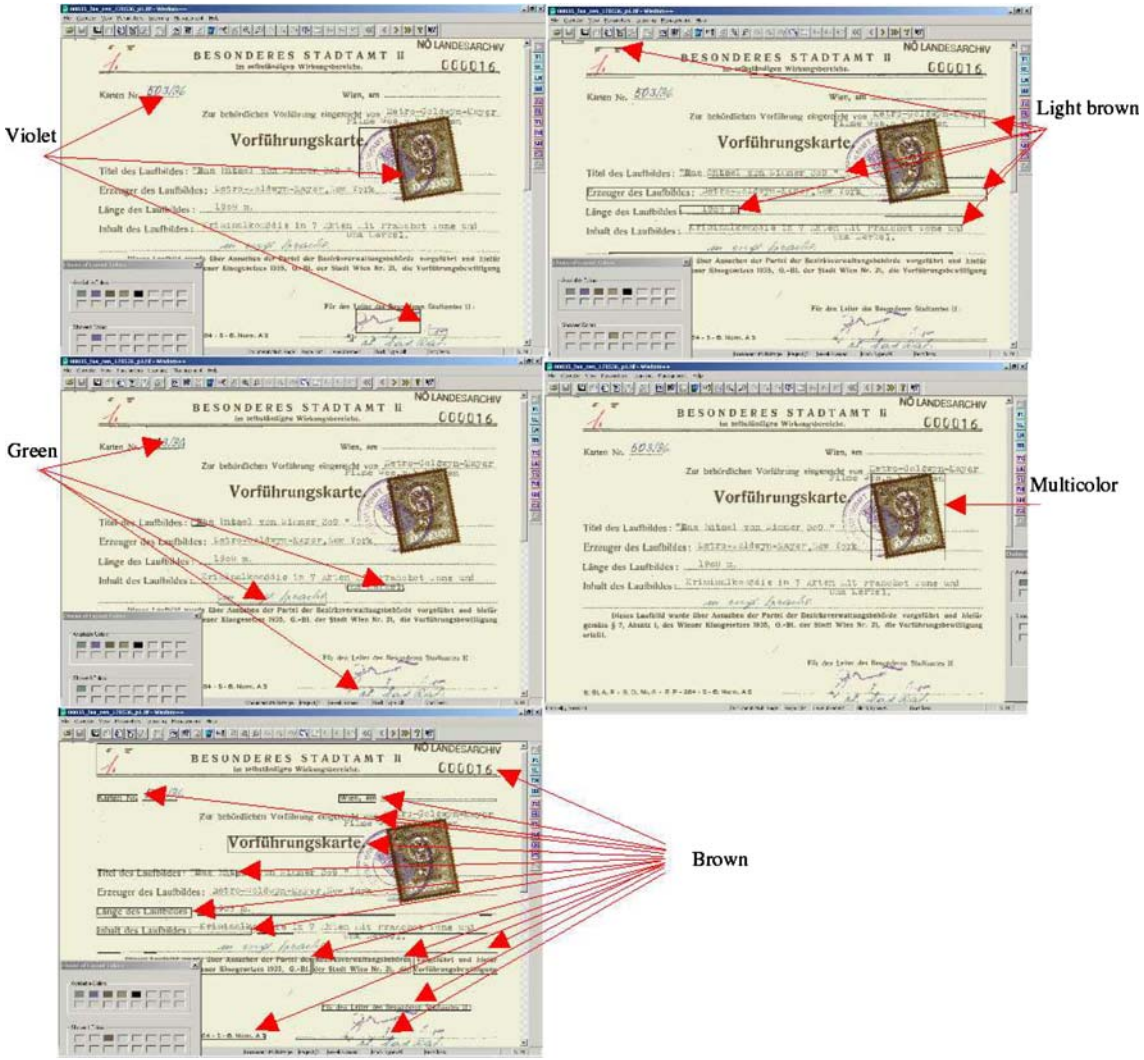

Fig. 2 Result of the layout analysis. Images show blocks at frame 2 level

project includes the names of assessors on the top right of the first page, while there is no such information in a censorship card made available by FilmArchiv Austria. Hence, the mapping of the layout structure into the logical structure is subordinated to document classification. In WISDOM++, the document classification is also performed by matching a set of rules.

Several works report a similar rule-based approach to document image understanding. Lee et al. [18] use handcoded rules to identify sensible components in binary images of technical journals, while Niyogi and Srihari [29] propose a rule-based approach to understand newspaper pages. Klink and Kieninger [16] organize document image understanding in two phases: the first phase concerns the recognition of common document structures like headings, footnotes and lists, while the second phase concerns the recognition of domain-dependent logical elements. As in the previous two works, the rule base is handcoded and no colour information is exploited. A machine learning approach is proposed by Aiello et al. [1] and Palmero et al. [30]. The former apply the decision tree learning system C4.5 [32] to learn classification rules for textual layout components. The latter develop a neuro-fuzzy learning algorithm that ranks, for each new (unseen) block, candidate labels and selects the best.

As already pointed out, WISDOM++ makes extensive use of machine learning methods, also for document image understanding. Differently from some related works, where machine learning tools are likewise involved, in WISDOM++ only layout information is used to identify logical components. Text, font size and additional information returned by the OCR cannot be used, since document image understanding precedes the application of OCR. Swapping the order of the two processing steps would not help in any case, because the

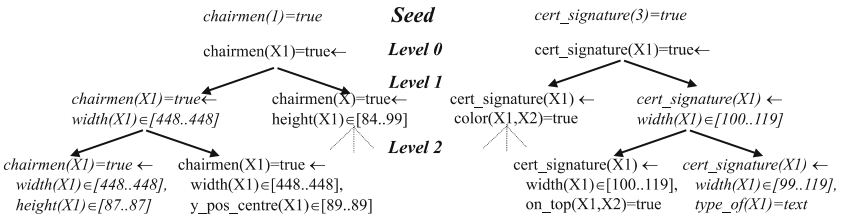

Fig. 3 A learning step example of the parallel search for the predicates *chairmen* and *cert_signature*. Starting from each seed, a specialization hierarchy is explored by adding a literal characterizing the seed example for each specialization level. The exploration proceeds until a user-defined number of consistent clauses (*in italics*) is generated

high level of noise present in our historical documents prevents a direct and effective application of the OCR. Another important difference with respect to related works is the more powerful representation formalism adopted in WISDOM++ to describe the layout of training documents. WISDOM++ resorts to first-order logic, which allows spatial relations between layout components to be effectively and naturally represented. On the contrary, competing systems resort to decision trees and neural network which are unsuitable for representing a variable number of spatial neighbours of a layout component together with their attributes. Details on the representation languages adopted by the learning system ATRE, embedded in WISDOM++, are explained in the following subsection.

3.1 Learning rules with ATRE

ATRE[2] is the system used to learn rules for document image understanding. The learning problem solved by ATRE can be formulated as follows:

Given

- a set of *concepts* C_1, C_2, ..., C_r to be learned
- a set of *observations* O described in a language L_O
- a *background theory* BK
- a *language* of hypotheses L_H
- a user's *preference criterion* PC

Find

A logical theory T expressed in the language L_H and defining the concepts $C_1, C_2, \ldots, C_r$, such that T is complete and consistent with respect to O and satisfies the preference criterion PC.

The *completeness* property holds when the theory T explains all observations in O of the r concepts C_1, C_2, ..., C_r, while the *consistency* property holds when the theory T explains no counter-example in O of any concept C_i. The satisfaction of these properties guarantees the correctness of the induced theory, with respect to the given observations O. Whether the theory T is correct with respect to additional observations not in O is an extra-logical matter, since no information on the generalization accuracy can be drawn from the training data themselves. In fact, the selection of the "best" theory is always made on the grounds of an inductive bias embedded in some heuristic function or expressed by the user of the learning system (preference criterion).

In the document image understanding domain, each concept to be learned corresponds to a logical label of interest.

As to the representation languages, the basic component is the *literal*, which can be of the two distinct forms:

$$f(t_1, \ldots, t_n) = \text{Value (simple literal)}$$
$$f(t_1, \ldots, t_n) \in \text{Range (set literal)},$$

where f and g are function symbols called *descriptors*, t_is are *terms* (constants or variables) and *Range* is a closed interval of possible values taken by f. Some examples of literals are: colour(X1) = red, height(X1)∈[1.1 ...1.2], and ontop(X, Y) = true. The original list of descriptors used to represent the layout of multi-page documents for document understanding tasks (see [23]) has been extended to exploit colour information. Each block is described not only in terms of its geometry and content type, but also in terms of its corresponding RGB values (descriptors red, green and blue). For multi-colour layout components, such as revenue stamps, the only predicate multi-colour is used, since no RGB value can be univocally associated.

In ATRE, training observations are represented by ground multiple-head clauses [19], called *objects*, which have a conjunction of simple literals in the head. The head of an object contains positive and negative examples describing the set of user-assigned logical labels, while the body contains the description of layout components on the basis of geometrical features (e.g. width, height, centroid position of a block) and topological relations (e.g. vertical and horizontal alignments) existing among blocks, as well as the type of the content

[2] http://www.di.uniba.it/∼malerba/software/atre

(e.g. text, horizontal line, image). Terms of literals in objects can only be constants, where different constants represent distinct layout components within a page. An example of a training object is the following:

```
object('faa_cen_120536_1-2',[class(1)=faa_cen_decision,
registration_au(2)=true, ..., dep_signature(2)=false, …
reg_number(14)=true, registration_au(14)=false, …,
dep_signature(14)=false, ..., adhesive_stamp(26)=true,
registration_au(26)=false, ..., dep_signature(26)=false, …
dep_signature(48)=false],
[page(1)=first,
part_of(1,2)=true, …, part_of(1,14)=true, …,
part_of(1,26)=true, colour(2,c0)=true, …,
colour(14,c3)=true, …, colour(48,c3)=true,
red(c0)=159, green(c0)=146, blue(c0)=109,
red(c3)=120, green(c3)=142, blue(c3)=142, …
multicolour(26)=true,
width(2)=28, width(3)=19, …, width(48)=113,
height(2)=5, height(3)=15, ..., height(48)=18,
type_of(2)=text, type_of(3)=graphics, …,
type_of(48)=text, x_pos_centre(2)=31,
x_pos_centre(3)=65, …, x_pos_centre(48)=452,
y_pos_centre(2)=5, y_pos_centre(3)=16, …,
y_pos_centre(48)=424,
on_top(2,26)=true, on_top(3,9)=true, …,
on_top(45,46)=true,
to_right(12,15)=true, to_right(12,13)=true, …,
to_right(25,27)=true,
alignment(5,36)=only_right_col,
alignment(3,10)=only_left_col, …,
alignment(43,45)=only_lower_row]).
```

where the constant 1 denotes the whole page and the remaining constants identify distinct layout components. In this example, the block number 2 corresponds to a block enclosing the textual information on the *registration authority* of the censorship card. The block is 28 pixels wide and 5 pixels high and it is characterized by the colour 'c0', whose RGB values are $\langle 159, 146, 109 \rangle$. In addition, it is placed over the block number 26 which is a multi-colour block.

The learned theory is composed of a set of logical clauses, each of which describes some conditions that characterize a subset of positive examples of a concept C_i. Two examples of learned clauses are:

```
chairmen(X1)=true ← width(X1)∈[269..304],
    height(X1)∈[20..37], colour(X1,X2)=true, red(X2)∈[7..75]
assessors(X1)=true ← width(X1)∈[348..364],
    on_top(X1,X2)=true, chairmen(X2)=true
```

The first clause states that if a block is quite short (height is between 20 and 37 pixels) and wide (width is between 269 and 304 pixels) and its red RGB component has a value in the range between 7 and 75, then it can be classified as the *chairmen* of the censorship card. The second clause states that if $X1$ is quite a large block (width is between 348 and 364 pixels) placed over the block related to the *chairmen*, then it can be labelled as the *assessors* of the censorship card.

The last clause exemplifies a distinguishing characteristic of ATRE, namely the possibility to discover dependencies between concepts to be learned (e.g. *assessors* and *chairmen*). Indeed, logical components may be related to each other, and such dependence can be reflected by some geometric relationships between the layout components associated to those logical components. Rules learned for document image understanding should reflect dependencies between logical components to enable a context-sensitive recognition. However, most of the studies on inductive learning presented in the machine learning literature make the implicit assumption that concepts are independent (*independence assumption*). The learning strategy implemented in ATRE is a notable exception.

The high-level learning algorithm in ATRE belongs to the family of *sequential covering* (or *separate-and-conquer*) algorithms [26], since it is based on the strategy of learning one clause at a time (conquer stage), removing the covered examples (separate stage) and iterating the process on the remaining examples.

The most relevant novelties of the learning strategy implemented in ATRE are embedded in the design of the conquer stage. Firstly, the conquer stage of our algorithm aims to generate a clause that covers a specific positive example, called *seed*. Secondly, the search space explored by ATRE is a forest of as many search-trees (called *specialization hierarchies*) as the number of chosen seeds, where at least one seed per incomplete concept definition is kept. Each search-tree is rooted with a unit clause and ordered by generalized implication. The forest can be processed in parallel by as many concurrent tasks as the number of search-trees (parallel-conquer search). Each task traverses the specialization hierarchies top-down (or general-to-specific), but synchronizes traversal with the other tasks at each level. Initially, some clauses at depth one in the forest are examined concurrently. Each task is actually free to adopt its own search strategy and to decide which clauses are worth being tested. If none of the tested clauses is consistent, clauses at depth two are considered. The search proceeds towards deeper and deeper levels of the specialization hierarchies until at least one consistent clause is found (see Fig. 3). Task synchronization is performed after all "relevant" clauses at the same depth have been examined. A supervisor task decides whether the search should carry on or not, on the basis of the results returned by the concurrent tasks. When the search is stopped, the supervisor selects the "best" consistent clause according to the user's preference criterion PC.

This strategy has the advantage that simpler consistent clauses are found first, independently of the concepts to be learned. Moreover, the synchronization allows tasks to save much computational effort when the distribution of consistent clauses in the levels of the different search-trees is uneven. This separate-and-parallel-conquer search strategy provides us with a solution to the problem of interleaving the induction process for distinct concept definitions [23].

This separate-and-parallel-conquer search presents some efficiency problems and leaves a large margin for optimization. One of the reasons is that every time a clause is added to the partially learned theory, the specialization hierarchies are reconstructed for a new set of seeds, which may intersect the set of seeds explored in the previous step. Therefore, it is possible that the system explores the same specialization hierarchies several times, since it has no memory of the work done in previous steps. This is particularly evident when concepts to be learnt are not mutually dependent. Intuitively, caching the specialization hierarchies explored at a certain step of the separate-and-conquer strategy and reusing part of them in the following step seems to be a good strategy to decrease the learning time while keeping memory usage under acceptable limits.

The second source of inefficiency is the repeated execution of "coverage" tests of generated clauses on the set of observations. Luckily, it can be proved that for independent clauses (i.e. clauses that do not express any concept dependency), the lists of negative examples remain unchanged between two subsequent learning steps. Therefore, by caching the list of negative examples for each independent clause, the learning system can save the computational cost of many redundant tests. A different observation concerns the list of positive examples covered by independent clauses, since it can decrease between two successive learning steps. We observe, however, that the set of positive examples of a clause C' generated at the $(i+1)$-th step can be calculated as the intersection of the cached set computed at the i-th step of the learning strategy and the set of positive examples covered by the parent clause of C' in the specialization hierarchy computed at the $(i+1)$-th step. Therefore, by caching also the list of positive examples it is possible to reduce the computational complexity [6].

The side-effect of caching is the increased space complexity and the additional time spent to retrieve cached statistics. Whether caching is actually useful depends on the task in hand. In the next section we will also evaluate the time performance of ATRE for the specific task of learning rules for document image understanding.

4 Application to censorship cards

In this section we empirically evaluate the proposed approach on a corpus of documents collected in the context of the project COLLATE. We want to empirically prove the effectiveness of the colour-based approach both in the identification of the layout structure and in the document image understanding task. Results are compared with the original version of WISDOM++ that processes black-and-white images.

Before describing the results, we illustrate the corpus used in this study.

4.1 Description of the corpus

Processed document images have been provided by three European film archives, namely Deutsches Filminstitut (DIF), Filmarchiv Austria (FAA) and Národní Filmový Archiv (NFA), involved in the project COLLATE[3]. Generally, documents are multi-page, where each page is a colour image representing rare historic film censorships from the 1920s and 1930s.

Each document page corresponds to an RGB 24 bit colour image in TIFF format. We considered 58 multi-page documents belonging to 3 distinct classes, one for each archive (see Table 1) and we applied WISDOM++ to 108 document images in all. Documents belonging to the same class present similar layout structures.

For each document class, film archivists defined a set of logical labels, useful for indexing and retrieval purposes, to be associated to layout components. For the FAA class, labels are *dep_signature, adhesive_stamp, stamp, registration_au, date_place, department, applicant, reg_number, film_length, film_producer, film_genre, film_title*. For the DIF class, labels are *cens_signature, cert_signature, object_title, cens_authority, chairmen, assessors, session_data, representative*. For the NFA class, labels are *round_stamp, date_place, film_content, delivery_date, cens_card, cens_process, recommendation, dispatch_office, no_reels, film_genre, applicant, film_length, film_producer, film_title, rubber_stamp, stamp, no_prec_doc, no_censor_cards, registration_au, register_office, official_notes, exec_date, cens_advisory, applic_notes*.

4.2 Layout analysis: improvements over the b/w layout analysis

In this section we evaluate the colour-based layout analysis effectiveness in terms of capability to isolate

[3] http://www.collate.de/

Table 1 Main features of processed censorship cards

Class	No. of docs	Total no. of pages of pages	Size (pixel)	Resolution (dpi)	Image size (mm)
FAA	25	50	4,836 × 3,408	600	204.72 × 144.27
DIF	25	50	2,460 × 3,474	300	208.28 × 294.13
NFA	8	8	2,528 × 3,988	300	214.05 × 337.66

interesting blocks of different colour for subsequent logical labelling. To evaluate this aspect, we compared the output of the proposed colour-based layout analysis with the output of the black and white (b/w) layout analysis implemented in the original version of WISDOM++ [2].

Both layout analysis algorithms were applied to the same set of 108 document pages. In the case of the colour-based layout analysis, the following threshold values were used: *th_IntSM* = 70%, *th_InclSM* = 75%, *th_MinOcc* = 1.5% and *th_MaxOcc* = 4.5%. For a fair comparison, in both cases no preprocessing was applied and the layout was not manually corrected [24]. Once the layout structures had been extracted, the same user manually labelled interesting components according to interesting labels.

In Figs. 4, 5 and 6, examples of outputs of the black-and-white and colour-based layout analysis processes are shown. It is noticeable that the colour-based layout analysis is able to isolate interesting blocks better than the previous version. For example, in Fig. 4 the black-and-white layout analysis returns very few blocks. In particular, labels such as *stamp, film_genre, film_length, adhesive_stamp* have not been separated and co-occur in the same *frame2* block. On the contrary, colour-based layout analysis is able to isolate them. By closely looking at the image, we can draw another consideration: the *dep_signature* (in violet at the bottom of the original image) has not been represented at all in the black-and-white image. This can be explained by the approximation performed by the embedded binarization algorithm that has been used to transform the original colour image into black-and-white. This loss of layout components does not occur in colour-based layout analysis, where binarization is not necessary.

By looking at Fig. 5, we note that the colour-based layout analysis is able to identify overlapping blocks, that is, *cens_signature* and *stamp*. On the contrary, the black-and-white layout analysis identifies two blocks, and the stamp has been split.

In Fig. 6, a document image belonging to the class with the most complex layout structure, that is, NFA, is shown. In this case, the document contains manual annotations (*no_prec_doc*, top right-hand corner), blue stamps (*register_office* and *dispatch_officer*, bottom page), red stamps (*rubber_stamp*, top left-hand corner) and revenue stamps (*stamp*, middle page). The

Fig. 4 Comparing colour-based layout structure and black-and-white layout structure on the first page of a FAA censorship card

colour-based layout analysis is able to isolate them, while the black-and-white layout analysis returns a single layout block for the whole central part of the document image and two spurious blocks extracted from the bottom of the image. This poor result is due to the presence of both vertical and horizontal lines, which affect the RLSA segmentation, especially when colours are not differentiated.

In Table 2, we present statistics on the number of *frame 2* layout components that the user is able to label. While in the case of DIF censorship cards the number of labelled components with the colour-based layout analysis is comparable to the case of b/w layout analysis, the situation is different for the other classes, where it is possible to identify more logical components. This can be explained by the minor relevance of colour in the case

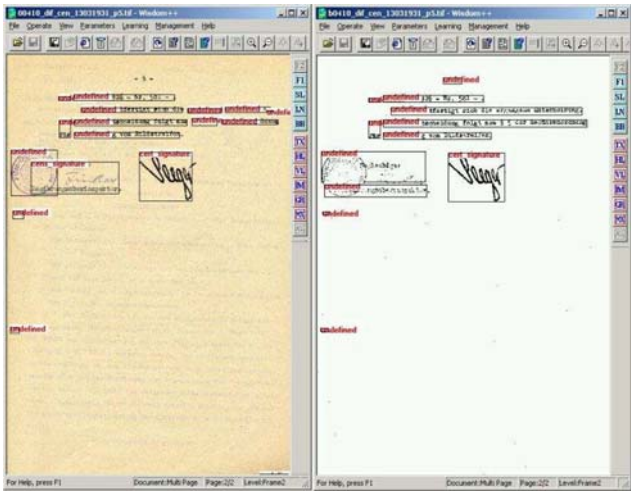

Fig. 5 Comparing colour-based layout analysis (*left*) and black-and-white layout analysis (*right*) on the second page of a DIF censorship card

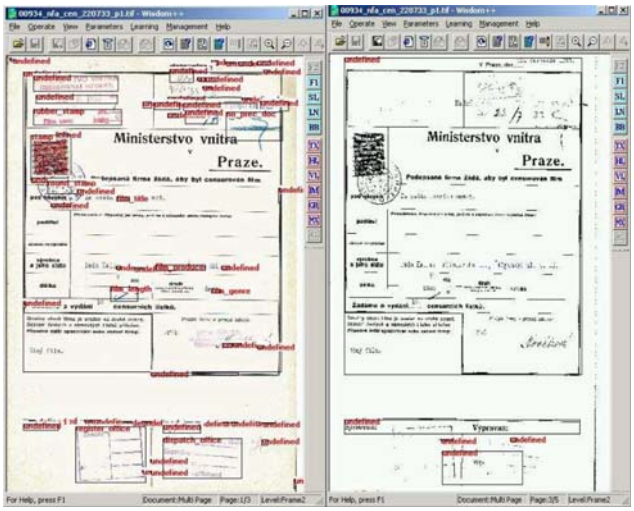

Fig. 6 Comparing colour-based layout analysis (*left*) and black and white layout analysis (*right*) on the first page of an NFA censorship card

of DIF images. On the contrary, in the case of both FAA and NFA, several logical components are characterized by colour information. This is particularly evident in the case of NFA documents that represent the most complex task because of the overall low quality of the paper and because they contain manual annotations, rubber stamps and revenue stamps of different colours.

4.3 Document image understanding

In this subsection we investigate both effectiveness and efficiency of the proposed solution in the document image understanding task.

Table 2 Labelled logical components

	Colour-based layout analysis	B/W layout analysis
DIF	133 (76%)	149 (85%)
FAA	205 (68.3%)	140 (46.7%)
NFA	64 (33%)	12 (6.2%)

Percentages of labelled logical components out of the total number of human-recognized components are reported in brackets

To investigate the effectiveness of the proposed solution in the document image understanding problem, we analysed the DIF and FAA documents described in Table 1 by means of a fivefold cross-validation, that is, each dataset was first divided into five folds of near-equal size (five documents per fold) and for every fold, ATRE was trained on the union of the remaining folds and tested on the hold-out fold. On the other hand, for the NFA dataset we used a leave-one-out cross-validation, which is simply an n-fold cross-validation, where n is the number of documents in the dataset. The leave-one-out validation is due to the very low number of available examples.

Two settings are compared: in the first setting documents are processed by means of the colour-based layout analysis and the ATRE descriptions include colour information, in the second setting, documents are processed by means of the black-and-white layout analysis and ATRE descriptions do not include colour information.

For each learning problem, the number of omission/commission errors is recorded. *Omission* errors occur when logical labelling of layout components is missed, while *commission* errors occur when wrong logical labelling is "recommended" by a rule.

The number of objects for ATRE corresponds to the total number of document pages, namely 50 for FAA, 50 for DIF and 8 for NFA documents. The total number of examples is 21,073 for FAA, 16,552 for DIF and 8,112 for NFA documents in the colour setting and 6,036 for FAA, 7,600 for DIF and 1,768 for NFA in the black-and-white setting. They correspond to the total number of literals in the heads of clauses. Given the set of concepts to be learned, positive examples are only 140 for FAA (2.32%), 149 for DIF (1.96%) and 12 for NFA (0.68%): they correspond to recognized layout components in the black-and-white setting. The situation is different for the colour-based setting, where only 205 out of 21,073 (0.97%) for FAA, 133 out of 16,552 (0.8%) for DIF and 64 out of 8,112 (0.79%) for NFA examples are positive. This means that there is a disparity between the number of positive and negative examples. Moreover, it is important to consider that this disparity is much more evident in the colour-based representations, where the

Table 3 Number of omission and commission errors w.r.t. positive and negative examples, respectively

Results are obtained on the FAA dataset, running the colour setting and the B/W setting

Fold	No. of learned rules		No. of positive examples		No. negative examples		Omissions		Commissions	
	Col	*BW*	*Col*	*BW*	*Col*	*BW*	*Col*	*BW*	*Col*	*BW*
F1	73	37	163	106	17,192	4,430	32/ 42	23/ 34	17/ 3,676	9/ 1,466
F2	80	38	164	111	15,527	4,545	27/ 41	17/ 29	17/ 5,341	14/ 1,351
F3	80	49	161	104	17,506	4,816	37/ 44	28/ 36	13/ 3,362	5/ 1,080
F4	72	51	169	114	16,757	4,566	24/ 36	12/ 26	11/ 4,111	4/ 1,330
F5	81	48	163	125	16,490	5,227	34/ 42	7/ 15	17/ 4,378	9/ 669
Avg.							*.747*	*.594*	*.004*	*.007*

Table 4 Number of omission and commission errors w.r.t. positive and negative examples, respectively

Results are obtained on the DIF dataset, running the colour setting and the B/W setting

Fold	No. of learned rules		No. of positive examples		No. of negative examples		Omissions		Commissions	
	Col	*BW*	*Col*	*BW*	*Col*	*BW*	*Col*	*BW*	*Col*	*BW*
F1	31	31	109	121	13,483	5,879	17/ 24	12/ 28	9/ 2,936	1/ 1,572
F2	38	29	102	119	13,674	5,913	17/ 31	12/ 30	6/ 2,745	12/ 1,538
F3	36	27	106	116	12,574	5,876	11/ 27	6/ 33	8/ 3,845	6/ 1,575
F4	38	26	113	124	13,631	6,260	10/ 20	9/ 25	7/ 2,788	4/ 1,191
F5	27	28	102	116	12,314	5,876	22/ 31	14/ 33	13/ 4,105	5/ 1,575
Avg.							*.574*	*.359*	*.003*	*.004*

Table 5 Number of omission and commission errors w.r.t. positive and negative examples, respectively

Results are obtained on the FAA dataset, running the colour setting and the B/W setting. Table only concerns the *adhesive_stamp* concept

Fold	No. of learned rules		No. of positive examples		No. of negative examples		Omissions		Commissions	
	Col	*BW*	*Col*	*BW*	*Col*	*BW*	*Col*	*BW*	*Col*	*BW*
F1	3	2	13	5	1,322	373	0/ 4	0/ 2	0/ 282	0/ 123
F2	3	2	14	6	1,193	382	0/ 3	1/ 1	0/ 411	2/ 114
F3	3	2	14	5	1,345	405	1/ 3	2/ 2	0/ 259	0/ 91
F4	3	2	14	5	1,288	385	2/ 3	2/ 2	0/ 316	0/ 111
F5	2	2	13	7	1,268	439	2/4	0/0	3/336	1/57
Avg.							.201	.75	.002	.007

percentage of positive examples is (in two cases out of three) less than half of the corresponding value for the b/w setting. This means that the learning task in the colour setting is more difficult than in the black-and-white setting.

In Tables 3 and 4, cross-validation results for the document understanding task are reported. Results concern the FAA and the DIF datasets. It is noteworthy that, although the omission errors increase in the case of the colour setting, the percentage of commission errors decreases. Rules learned in the colour setting are actually more specific, so they are more precise, but do not capture all the variability of the layout.

By analysing some results in more detail, we observe that in the case of b/w setting, some concepts that are strongly dependent on the colour information are very hard to understand and, sometimes the classifier is not able to identify them at all. In Table 5, the example of the *adhesive_stamp* of the FAA class is reported. In this case, the classifier learned in the colour setting outperforms by a wide margin the classifier in the b/w setting both in terms of omission and commission errors.

As regards the learned theories, in the following some examples of rules for the classes FAA and DIF are reported:

FAA:

adhesive_stamp(X1)=true ← height(X1)∈[100..119],
    multicolour(X1)=true

registration_au(X1)=true ← y_pos_centre(X1)∈[21..23],
    colour(X1,X2)=true, red(X2)∈[75..111]

film_title(X1)=true ← alignment(X1,X2)=only_middle_col,
    width(X1)∈[139..169], width(X2)∈[8..8]

DIF:

session_data(X1)=true ← alignment(X2,X1)=only_upper_row,
    cens_authority(X2)=true, width(X1)∈[50..200]

cens_authority(X1)=true ← y_pos_centre(X1)∈[19..19],
    colour(X1,X2)=true, red(X2)∈[72..72]

The first rule expresses the condition that a multicolour block, whose height is between 100 and 109 is an *adhesive stamp* affixed to the censorship card.

The second and the last rules take into account colour information. In particular, the second rule states that a coloured block with a red component of the RGB space varying between 75 and 111 (the admissible interval is 0. . .255) and with its barycentre at a point between 21 and 23 on the *y*-axis is the *registration authority* of the censorship document. The last rule states that a layout component with its barycentre at a point of 19 on the *y*-axis and that is characterized by a

Table 6 Number of omission and commission errors w.r.t. positive and negative examples, respectively

Results are obtained on the NFA dataset, running the colour setting

Fold	No. of learned rules	No. of positive examples	No. negative examples	Omissions	Commissions
Leaved out 1	33	57	7,444	7/7	0/604
Leaved out 2	37	58	7,209	5/6	5/839
Leaved out 3	35	57	6,924	5/7	1/1,124
Leaved out 4	31	56	6,678	6/8	3/1,370
Leaved out 5	34	55	7,368	8/9	5/680
Leaved out 6	34	57	6,612	6/7	2/1,443
Leaved out 7	34	53	7,084	7/11	3/964
Leaved out 8	32	55	7,017	8/9	5/1,031
Avg.				.820	.0032

Table 7 Learning times (results obtained on a PentiumIV 1.6GHz running MS WIN2k) and accuracy comparison of ATRE vs Optimized

ATRE. Results are obtained on the DIF dataset, running the colour setting and the B/W setting

Fold	Execution times (s)			Omissions		Commissions	
	ATRE	OPT. ATRE	Time gain rate	ATRE	OPT. ATRE	ATRE	OPT. ATRE
F1	1,936.26	858.62	55.66%	15/24	17/24	7/2,936	9/2,936
F2	2,764.90	1,251.58	54.73%	18/31	17/31	5/2,745	6/2,745
F3	2,441.56	1,125.40	53.91%	10/27	11/27	11/3,845	8/3,845
F4	2,454.09	1,024.68	58.25%	12/20	10/20	8/2,788	7/2,788
F5	1,601.77	619.01	61.35%	20/31	22/31	14/4,105	13/4,105
Avg.	2,239.71	975.86	56.43%	56.42%	57.48%	0.27%	0.26%

red component of 72 is the *censorship authority* of the censorship card.

The third and fourth rules are examples of rules expressing spatial relations among layout components. More specifically, the third rule expresses that quite a large layout component (width between 139 and 169) that is vertically aligned with a short layout component (width equal to 8) is the *title* of the film. The fourth rule shows a conceptual dependency. In fact, it expresses that a large layout component (width between 50 and 200) that is horizontally aligned with the layout component labelled as *cens_authority* is the *session_data* of the censorship document.

A different analysis should be made in the case of the NFA dataset. In fact, in this case it was not possible to run ATRE in the b/w setting since the user was able to label only 12 examples for three different concepts (*date_place, register_office, dispatch_office*). If we had run ATRE, the learned theory would have been based on very few training examples and almost no test examples. On the contrary, in the colour setting, we had enough positive examples to run ATRE (results are shown in Table 6).

As in the case of DIF and FAA, in the following we report some examples of learned rules for the NFA dataset.

```
rubber_stamp(X1)=true ← colour(X1,X2)=true,
      red(X2)∈[104..124], green(X2)∈[51..58]
stamp(X1)=true ← y_pos_centre(X1)∈[202..213],
      colour(X1,X2)=true, red(X2)∈[95..161]
```

The first rule shows that a *rubber_stamp* can be classified only considering colour information (in this case both red and green RGB dimensions are used as discriminating features). The second rule expresses the condition that a layout component with its barycentre at a point between 202 and 213 on the *y*-axis and with the red RGB component between 95 and 161 is a *stamp*.

The second aspect we investigate concerns the efficiency of the learning system ATRE. In particular, we evaluate the effective advantage of the use of the caching technique described in Sect. 3. For this aim, we present results concerning the comparison, in terms of time complexity, of the original version of ATRE with respect to the optimized version obtained by caching techniques. In Table 7, results on the DIF dataset in the colour setting are reported. Results show that the gain in terms of learning time of the optimized version is of more than 56% on average. The table also shows that this reduced complexity does not negatively affect the classification accuracy.

5 Conclusions and future work

In this paper, we presented some extensions of the DIA system WISDOM++ required to meet the challenges arising from the processing of censorship cards from European film archives of the 1920s and 1930s. WISDOM++ was originally developed to support the transformation of printed documents from black-and-white

images into an HTML/XML format. However, to effectively process low layout quality and standard documents, such as censorship cards, the exploitation of information provided by colour has been necessary. Indeed, signatures, stamps and manual annotations, which are the main cause of noise in processing these documents, are often characterized by colours which differ from those present in the background and in typewritten texts. To take full advantage of colour information, all processing steps in WISDOM++ have been substantially revised. In particular, a new colour image segmentation algorithm has been proposed. Colour information extracted by means of this step has been profitably exploited in the layout analysis, as well as in the steps of image classification and understanding. Moreover, the richer representation employed in document classification and understanding raises efficiency problems and has led to the implementation of some caching techniques which have been integrated in the learning system ATRE embedded in WISDOM++. The application of the improved version of WISDOM++ on a collection of multi-format censorship cards provided by three national film archives, namely Deutsches Filminstitut (Germany), Filmarchiv Austria and Národní Filmový Archiv (Czech Republic) has been presented. The goal of the experimental study is to empirically evaluate the effectiveness of the proposed approach both in the identification of the layout structure and in the document image understanding task. For this purpose, firstly, we prove the effectiveness of the colour-based layout analysis. Secondly, we evaluate the effectiveness of the use of colour information in document image understanding. Thirdly, we prove the efficiency of the learning task. Results permit us to draw four main conclusions:

1. For the colour-based layout analysis, a comparison with the original version of WISDOM++ which processes black-and-white images shows that the system is now able to: (a) isolate better noise components, such as manual annotations, ink specks etc.; (b) separate better colour-dependent components of interest, such as rubber stamps, signatures; (c) identify multicolour blocks, such as revenue stamps; (d) distinguish overlapping components; (e) identify low density and light-coloured components that, otherwise, in the b/w approach are lost in the binarization process. This more accurate layout structure allows archivists to label a greater number of components of interest.
2. The learning system suffers from an increase in the complexity of the task with respect to the b/w setting. Although the percentage of commission errors decreases, the omission errors increase. This is due to a lower percentage of positive examples which generally leads to a specificity of learned rules with a low percentage of coverage of training examples. Specificity of learned rules is due to the fact that ATRE is asked to generate a complete theory, that is, a set of rules that explains all positive examples. Another difficulty lies in the low layout standard that leads to a non-homogeneous distribution of the layout components to be understood, that is, to a non-uniform distribution of positive examples. However, in one of three datasets, we were not able to use the b/w approach because of the low quality of the layout structure.
3. Some logical components, strictly related to colour features such as revenue stamps, are very difficult to recognize in the b/w setting. This is not only due to the different layout structure in the b/w layout analysis, but primarily to the fact that the learned theory benefits from the colour description. As expected, in such cases, the classifier learned in the colour setting outperforms the classifier learned in the b/w setting.
4. The proposed caching techniques implemented in the learning system ATRE show better performances in terms of time complexity with respect to the previous version embedded in the original WISDOM++. This is an important result considering the complexity of the learning task arising from the extension of the representation language to handle colour information.

For future work, we intend to explore the possibility to support the layout analysis with the automatic prediction of the spatial merging threshold values. Moreover, difficulties met for document image understanding suggest the investigation both of representation issues and problems related to uncertainty management in learning and classification. In particular, we intend to further enrich the representation language adopted to describe layout structures and to explore the opportunity of relaxing the definition of a subsumption test between clauses [8], in order to weaken the conditions of applicability of rules and to significantly recover omission errors.

References

1. Aiello, M., Monz, C., Todoran, L., Worring, M.: Document understanding for a broad class of documents. Int. J. Doc. Anal. Recogn. **5**(1), 1–16 (2002)
2. Altamura, O., Esposito, F., Malerba, D.: Transforming paper documents into XML format with WISDOM++. Int. J. Doc. Anal. Recogn. **4**(1), 2–17 (2001)

3. Antonacopoulos, A., Karatzas, D.: Document image analysis for World War II personal records. In: 1st International Workshop on Document Image Analysis for Libraries (DIAL 2004), pp. 336–341 (2004)
4. Antonacopoulos, A., Karatzas, D., Krawczyk, H., Wiszniewski, B.: The lifecycle of a digital historical document: structure and content. In: Munson, E.V., Vion-Dury J.Y. (eds.) Proceedings of the 2004 ACM Symposium on Document Engineering, pp. 147–154. ACM (2004)
5. Bensaid, A., Hall, L.O., Bezdek, J.C., Clarke, L.P.: Partially supervised clustering for image segmentation. Pattern Recogn. **29**(5), 859–871 (1996)
6. Berardi, M., Varlaro, A., Malerba, D.: On the effect of caching in recursive theory learning. In: Camacho, R., King, R.D., Srinivasan A. (eds.) Inductive Logic Programming, Lecture Notes in Computer Science, vol. 3194, pp. 44–62. Springer, Berlin Heidelberg New York (2004)
7. Cheng, H.D., Jiang, X., Sun, Y., Wang, J.: Color image segmentation: advances and prospects. Pattern Recogn. **34**(12), 2259–2281 (2001)
8. Esposito, F., Malerba, D., Marengo, V.: Inductive learning from numerical and symbolic data: an integrated framework. Intell. Data Anal. **5**(6), 445–461 (2001)
9. Frommholz, I., Brocks, H., Thiel, U., Neuhold, E.J., Iannone, L., Semeraro, G., Berardi, M., Ceci, M.: Document-centered collaboration for scholars in the humanities – the collate system. In: European Conference on Research and Advanced Technology for Digital Libraries, pp. 434–445 (2003)
10. Gatos, B., Ntzios, K., Pratikakis, I., Petridis, S., Konidaris, T., Perantonis, S.J.: A segmentation-free recognition technique to assist old greek handwritten manuscript ocr. In: Marinai, S., Dengel A. (eds.) International Workshop on Document Analysis Systems, Lecture Notes in Computer Science, vol. 3163, pp. 63–74. Springer, Berlin Heidelberg New York (2004)
11. Gatos, B., Pratikakis, I., Perantonis, S.J.: An adaptive binarization technique for low quality historical documents. In: Marinai S., Dengel A. (eds.) International Workshop on Document Analysis Systems, Lecture Notes in Computer Science, vol. 3163, pp. 102–113. Springer, Berlin Heidelberg New York (2004)
12. Gervauz, M., Purgathofer, W.: A simple method for color quantization: octree quantization. Graphic Gems, pp. 287–293 (1990)
13. Hase, H., Yoneda, M., Tokai, S., Kato, J., Suen, C.Y.: Color segmentation for text extraction. Int. J. Doc. Anal. Recogn. **6**(4), 271–284 (2003)
14. He, J., Downton, A.C.: Configurable text stamp identification tool with application of fuzzy logic. In: Marinai S., Dengel A. (eds.) International Workshop on Document Analysis Systems, Lecture Notes in Computer Science, vol. 3163, pp. 201–212. Springer, Berlin Heidelberg New York (2004)
15. Karatzas, D., Antonacopoulos, A.: Two approaches for text segmentation in web images. In: International Conference on Document Analysis and Recognition, pp. 131–136 (2003)
16. Klink, S., Kieninger, T.: Rule-based document structure understanding with a fuzzy combination of layout and textual features. Int. J. Doc. Anal. Recogn. **4**(1), 18–26 (2001)
17. Le Bourgeois, F., Kaileh, H.: Automatic metadata retrieval from ancient manuscripts. In: Marinai, S., Dengel A. (eds.) International Workshop on Document Analysis Systems, Lecture Notes in Computer Science, vol. 3163, pp. 75–89. Springer, Berlin Heidelberg New York (2004)
18. Lee, K.H., Choy, Y.C., Cho, S.B.: Geometric structure analysis of document images: A knowledge-based approach. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11), 1224–1240 (2000)
19. Levi, G., Sirovich, F.: Generalized and/or graphs. Artif. Intell. **7**(3), 243–259 (1976)
20. Lucchese, L., Mitra, S.K.: An algorithm for fast segmentation of color images,. In: Proceedings of IEEE 10th Tyrrhenian Workshop on Digital Communication, pp. 110–119 (1998)
21. Lucchese, L., Mitra, S.K.: Advances in color image segmentation. In: Proceedings of Globecom'99, pp. 2038–2044 (1999)
22. Malerba, D.: Learning recursive theories in the normal ilp setting. Fundamenta Informaticae **57**(1), 39–77 (2003)
23. Malerba, D., Esposito, F., Lisi, F.A., Altamura, O.: Automated discovery of dependencies between logical components in document image understanding. In: International Conference on Document Analysis and Recognition, pp. 174–178 (2001)
24. Malerba, D., Esposito, F., Altamura, O., Ceci, M., Berardi, M.: Correcting the document layout: a machine learning approach. In: International Conference on Document Analysis and Recognition, p. 97 (2003)
25. Mello, C.A.B., Lins, R.D.: Image segmentation of historical documents. In: Visual2000: 3rd International Conference on Visual Computing (2000)
26. Mitchell, T.: Machine Learning. McGraw Hill, New York (1997)
27. Moghaddamzadeh, A., Bourbakis, N.G.: A fuzzy region growing approach for segmentation of color images. Pattern Recogn. **30**(6), 867–881 (1997)
28. Nicolas, S., Paquet, T., Heutte, L.: Enriching historical manuscripts: The bovary project. In: Marinai S., Dengel A. (eds.) International Workshop on Document Analysis Systems, Lecture Notes in Computer Science, vol. 3163, pp. 135–146. Springer, Berlin Heidelberg New York (2004)
29. Niyogi, D., Srihari, S.N.: Knowledge-based derivation of document logical structure. In: International Conference on Document Analysis and Recognition, pp. 472–475 (1995)
30. Palmero, G.I.S., Dimitriadis, Y.A.: Structured document labeling and rule extraction using a new recurrent fuzzy-neural system. In: International Conference on Document Analysis and Recognition, pp. 181–184 (1999)
31. Perroud, T., Sobottka, K., Bunke, H., Hall, L.: Text extraction from color documents – clustering approaches in three and four dimensions. In: International Conference on Document Analysis and Recognition, pp. 937–941 (2001)
32. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc. (1993)
33. Shih, Y., Chen, S.S.: Adaptive document block segmentation and classification. IEEE Trans. Syst. Man Cybern. Part B **26**(5), 797–802 (1996)
34. Sobottka, K., Kronenberg, H., Perroud, T., Bunke, H.: Text extraction from colored book and journal covers. Int. J. Doc. Anal. Recogn. **2**(4), 163–176 (2000)
35. Trémeau, A., Borel, N.: A region growing and merging algorithm to color segmentation. Pattern Recogn. **30**(7), 1191–1203 (1997)
36. Utgoff, P.: An improved algorithm for incremental induction of decision trees. In: Proceedings of the Eleventh Internatinal Conference on Machine Learning. Morgan Kaufmann (1994)
37. Wong, K., Casey, R., Wahl, F.: Document analysis system. IBM J. Res. Dev. **26**(6), 647–656 (1982)
38. Zhong, Y., Karu, K., Jain, A.K.: Locating text in complex color images. Pattern Recogn. **28**(10), 1523–1535 (1995)
39. Zhou, J., Lopresti, D.P.: Extracting text from www images. In: International Conference Document Analysis and Recognition, pp. 248–252. IEEE Computer Society (1997)