A color-based layout analysis to process censorship cards of film archives

Margherita Berardi Oronzo Altamura Michelangelo Ceci Donato Malerba
Dipartimento di Informatica – Università degli Studi di Bari
via Orabona 4 - 70126 Bari
{berardi, altamura, ceci, malerba }@di.uniba.it

Abstract

Processing censorship cards of the 20th century in order to support annotation and retrieval processes, leads to a number of challenges for many DIA systems. Problems due to the low layout quality and standard of such a material can be reduced by exploiting information conveyed by color. In this paper, taking into account lessons learned in the context of the IST project Collate, we propose a new method for image segmentation and layout analysis that takes full advantage of color information. The method has been implemented in the DIA system WISDOM++ and tested on a corpus of multi-format documents concerning historic film censorships.

1. Introduction

Many institutions which collect and preserve cultural heritage, as historical documents, have shown a great interest in the digitalization of their resources and in the exploitation of mechanisms to provide online access to digitalized products. Indeed, several research projects have been recently promoted for the purposes of preservation, storage, indexing, and on-line fruition. Interesting examples are: the MASTER project, that has developed a standard for computer-readable descriptions of medieval manuscripts in European libraries with retrieval objectives [10]; the MEMORIAL 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 [3]; the Bovary project, that concerns the digitalization of 5,000 original manuscripts handwritten by Gustave Flaubert [12]; the D-SCRIBE project, that aims to develop an integrated system for digitization and processing of Old Greek manuscripts [6].

This paper presents layout analysis issues and problems addressed in the EU funded project COLLATE, whose main goal is to provide film archivists adequate access to historic film-related documents and their associated metadata [5]. 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 it is. In this framework, we applied the DIA system WISDOM++ [1] to digitized documents available in three national film archives, namely Deutsches Filminstitut, Filmarchiv Austria and Národní Filmový Archiv (Czech Republic). WISDOM++ was originally developed to fully support the transformation of multi-page printed documents into XML format. Since most of information on the documents is typewritten, the system appeared to be useful for the conversion of scanned documents into a format (i.e. XML) suitable for storage and retrieval. Nevertheless, the low layout quality and standard of such a 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 to those layout components involved in the understanding processes.

To effectively process these documents, it is necessary to exploit information conveyed by color since signatures, stamps and manual annotations are often characterized by colors different from those present in the background and typewritten texts. In this way, it is possible to identify noise (e.g. strains, tears and irregular accumulation of dirt due to repeated handling [2]) on the basis of color homogeneity, as well as to isolate overlapping blocks of interest (e.g. as in legal documents, where blue stamps or revenue stamps often overlap signatures or typewritten text), but also to isolate interesting blocks from uninteresting ones (e.g. manual annotations that overlap layout components involved in annotation processes).

WISDOM++, originally developed to process black-and-white (binary) images, has been extended to take full advantage of color information in image segmentation and layout analysis steps. In particular, in the following section the description of the new algorithm is provided. In Section 3, results on censorship cards are reported and discussed. Finally, conclusions are drawn in Section 4.

2. The approach

A naïve approach to color document image processing would be to separate different colors and to process

Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05)
1520-5263/05 $20.00 © 2005 IEEE

images corresponding to each color separately, as independent binary images. However, this approach is based on the simplified assumption that a logical component can be associated with a single color. In practice, this assumption is rarely true since paper color darkens with age, while printed parts either handwritten or typed tend to fade [11]. Moreover, when the document is written or typed on both sides, and the backside is visible from the front side, further noise is introduced. For these reasons, a more sophisticated approach is necessary.

The proposed color image segmentation algorithm operates in three steps (see Fig. 1): color reduction and background removal, colorimetric merging and spatial merging. In the first phase, a color reduction that performs colors quantization in order to identify the set of relevant colors and to allow the user to manually select background colors is executed. Later on, the algorithm works on a set (`List`) of binary images, where 0 corresponds to pixels of background colors while 1 corresponds to one of the foreground colors. There are as many binary images as foreground colors.

The segmentation of each binary image is based on an efficient variant of the Run Length Smoothing Algorithm (RLSA) [15] and produces a list of rectangular blocks (`BBSet`). Once the set of basic blocks has been extracted, the colorimetric merging is performed. It aims to cluster binary images on the basis of the associated colors. Images belonging to the same cluster are removed from `List` and replaced by the merging result. The second merging step is performed on the updated `List` taking images in pairs. Images are segmented again and the spatial merging is applied on intersecting blocks. The result is an updated list of both binary and multicolor images. Both the merging steps take into account only pixels contained in the set of basic blocks. Pixels that do not contribute to the identification of a basic block are ignored as noise pixels. We observe that most of color image segmentation methods only operate in color space and do not take any spatial information into account. Thus, relations between color values and pixel positions in the image plane are not used [13] and the color homogeneity of spatially contiguous pixels is the only used criterion. This severe limitation is overcome by some methods [8, 9] that associate colors to layout components on the basis of both color and spatial information. Nevertheless, these approaches are based on the assumption that a layout component is associated to a single color. Differently, in our domain it is necessary to provide the system the capability to also identify multicolor blocks as pictures or revenue stamps.

```
procedure segment(OriginalImg, th_Incl,
                  th_Int, th_MinOcc, th_MaxOcc)
Output:  list of images
Begin
ReducedImage ← quantization(16,OriginalImg);
List    ← generateBinaryImg(ReducedImg,List);
List    ← removeBackground(List);
Forall ForegroundImg ∈ List
       BBSets ← BBSets ∪
RLSASegmentation(ForegroundImg);
List ← ColorimetricMerging(List, BBSets) ;
Forall Foreground1 ∈ List,
 Forall Foreground2 ∈ List-{Foreground1} {
   BBSet1 ← ApplyRLSASegm(Foreground1);
   BBSet2 ← ApplyRLSASegm(Foreground2);
   IntBlocks ←
        ComputeIntersections(BBSet1,BBSet2);
   List ←
        SpatialMerging(List,IntBlocks,th_Incl,
        th_Int,th_MinOcc,th_MaxOcc);}
return List
End;
```

Fig. 1 Top-level pseudocode of the segmentation algorithm.

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

The Quantization process. The quantization process follows the method proposed by [7] whose basic idea is to build a octree containing a maximum of K different leaves (a leaf corresponds to a color). Image is read twice. The first time, colors are iteratively added to the tree keeping at most K leaves. For our application domain, we set K=16 since archivists normally do not "see" more than nine colors in a censorship card. Quantization is performed at the second reading of the image and the K-colors image is transformed in K different binary images, each of which is related to a color. Among the K colors, the user manually selects a subset of m background colors. In our domain, m ranges between two and three due to document degradation. The exploration of the automatic selection of background colors is part of future work.

Colorimetric Merging. The list of K-m images is the merging phase input. The first merging process aims to merge those binary images whose colors can be considered as light variations. This is a necessary step, since the value K fixed a priori in the previous step may turn out to be too large.

The process is based on a hierarchical clustering algorithm. At each step, the dissimilarity between two clusters of colors (inter-cluster dissimilarity) is evaluated on the basis of two measures: a) the Euclidean distance between two colors taken from distinct clusters (nearest neighbor based dissimilarity); b) the Euclidean distance between the centroids of the two clusters (centroid-based dissimilarity). Two clusters of colors are merged when both computed measures are lower than a threshold. Clusters whose nearest neighbor based dissimilarity is lowest are considered firstly. The threshold is defined as the standard deviation value computed by considering all the distances between each color of one cluster and each color of another cluster. At the end, for each remaining cluster a new image representing the average color of original images is generated. All distances are computed in the CIELab space. CIELab space is obtained by a

nonlinear transformation of the original RGB space. We used CIELab since it is considered "visually uniform" because adjacent color samples represent equal intervals of visual perception [4].

Spatial Merging. By considering spatial information on the degree of overlapping of the layout extracted from different color images, it is possible to group together multicolor blocks and remove some useless low-density blocks (with few pixels) that capture color shades of the same layout component. Spatial merging operates on RLSA results when it is applied to the possibly reduced set of binary images determined by colorimetric merging. For each couple of binary images intersecting blocks are merged following three perceptual criteria.

The first criterion is summarized as follows:

```
Given BBx ∈ BasicBloks(Foreground1),
      BBy ∈ BasicBloks(Foreground2)
1) if perc_of_intersection(BBx,BBy) > th_Int
&& th_MinOcc<perc_of_occupation(BBx)<th_MaxOcc
&& th_MinOcc<perc_of_occupation(BBy)<th_MaxOcc
then Foreground1←removeArea(Foreground1,BBx);
  Foreground2←removeArea(Foreground2,BBy);
  NewForeground←GenerateMulticolor(BBx, BBy);
  List ← addElement(List, NewForeground);
```

This rule identifies multicolor layout components. For each couple of intersecting blocks, when the percentage of intersection exceeds a threshold (*th_Int*) and the percentage of occupation (i.e. 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 multicolor image is generated. The new image is built as the union of pixels of the original images enclosed in the blocks. Original binary images are also "cleaned" by removing pixels added to the multicolor image.

The second criterion is summarized as follows:

```
2) if perc_of_intersection(BBx,BBy) > th_Int
     && perc_Inclusion(BBx,BBy)+
     perc_Inclusion(BBy,BBx) ≥ th_Incl
     && Multicolor(Foreground1)
then Foreground1 ←
   addArea(Foreground1,Foreground2, BBy∩BBx);
   removeArea(Foreground2, BBy∩BBx);
```

This criterion is based on the rationale that if a block strongly overlaps a block of a multicolor image, the intersecting part has to be considered as composing the multicolor block. The pixels enclosed in the intersection are removed from the binary image (`Foreground2`) and added to the multicolor image (`Foreground1`).

A third criterion aiming at the extension of binary images is summarized by the following rules:

```
3) if perc_of_intersection(BBx,BBy) > th_Int
     && perc_Inclusion(BBx,BBy)+
     perc_Inclusion(BBy,BBx) ≥ th_Incl
     && perc_of_occupation(BBx) < th_MinOcc
then Foreground1 ←
   addArea(Foreground1,Foreground2,BBy);
   removeArea(Foreground2, BBy);
4) if perc_of_intersection(BBx,BBy) > th_Int
   && perc_Inclusion(BBx,BBy)+
     perc_Inclusion(BBy,BBx) ≥ th_Incl
   && density(BBx) < density(BBy)
then Foreground2 ←
   addArea(Foreground2,Foreground1, BBy∩BBx);
   removeArea(Foreground1, BBy∩BBx);
```

Rule 3 states that if a small block has a high degree of overlapping with a block of another image, it has to be considered a spurious block to include in the image associated to the "predominant" block (`Foreground1`). Rule 4 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.

Our algorithm allows the user to set four different thresholds: *th_Int* and *th_Incl* that define the minimal percentage of intersection and inclusion of merging blocks, respectively; *th_MinOcc* and *th_MaxOcc* that define the range of occupation for merging blocks. All the values are dependent on the specific type of 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 joined in a single set of blocks.

Layout Analysis. The segmentation algorithm returns (possibly) overlapping blocks that may contain either textual or graphical information and are either single color or multicolor. A first step towards the reconstruction of 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). This classification is performed by means of the decision tree learner ITI that builds a decision tree 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 color information obtained during the segmentation process.

Strategies for the extraction of layout structure have been traditionally classified as top-down or bottom-up [14]. WISDOM++ decomposes the document page in a hybrid way, since it combines the image segmentation and a bottom-up layout analysis method to assemble basic blocks into larger frames. 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 analyzed in

order to detect columns and sections/paragraphs, respectively. The levels of columns and sections are alternated, which means that a column contains sections, while a section contains columns.

2. A local analysis 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 (overlapping components), similarity (e.g. components of the same type, with an almost equal height) and color (i.e. components of the same color). Pairs of layout components that satisfy some of these criteria are grouped together. It is noteworthy that grouping affects either pairs of layout components extracted from the same binary image (i.e. with exactly the same color) or pairs of layout components labelled as multicolor. Therefore, the color associated with a new layout component is univocally determined from its constituents. Differently, it is possible to group together layout components with different content type. In this case, the associated type is set to mixed, otherwise it is set to the inherited type. The layout structure extracted for each document page is a hierarchy with five levels: basic blocks, lines, set of lines, frame1 and frame2.

3. Application

In this section we empirically evaluate the proposed approach in terms of the capability to isolate interesting blocks of different color for subsequent logical labeling. To evaluate this aspect, we compared the output of the new color-based layout analysis with the output of the black and white (b/w) layout analysis implemented in the original version of WISDOM++.

The corpus used in this study is composed by document images provided by the three film archives involved in the project COLLATE. Generally, documents are multi-page, where each page is a 256-colors image in TIFF format representing rare historic film censorships from the 20's and 30's. We applied WISDOM++ with both the layout analysis methods to 108 document images in all belonging to 3 distinct classes, one for each archive (see Table 1). In the case of the color-based setting, the following threshold values have been used: *th_Int* = 70%, *th_Incl*= 75%, *th_MinOcc* = 1.5% and *th_MaxOcc*= 4.5%. Once the layout structures have been extracted, to the same domain-expert user (archivist) is asked to manually label interesting components. The number of relevant labels is 12 for FAA, 7 for DIF and 13 for NFA.

In Fig. 2, 3 and 4, examples of layout analysis outputs are shown. It is noticeable that the color-based layout analysis is able to isolate interesting blocks better than the previous version. For example, in Fig. 2 the b/w 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, color-based layout analysis is able to isolate them. By closely looking at the image, we can draw another consideration: the *dep_signature* (in violet in the bottom) has not been represented at all in the b/w image, which is due to the approximation performed by the embedded binarization algorithm. Of course, this loss of layout components does not occur in color-based layout analysis. By looking at Fig. 3, we note that the color-based layout analysis is able to identify overlapping blocks, that is, *cens_signature* and *stamp*. On the contrary, the b/w layout analysis identifies two blocks, and the *stamp* has been split. In Fig. 4, a document image of the NFA class, that represents the most complex to analyze because of the overall low quality, 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*, in the middle of the page). The color-based layout analysis is able to isolate them, while the b/w 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

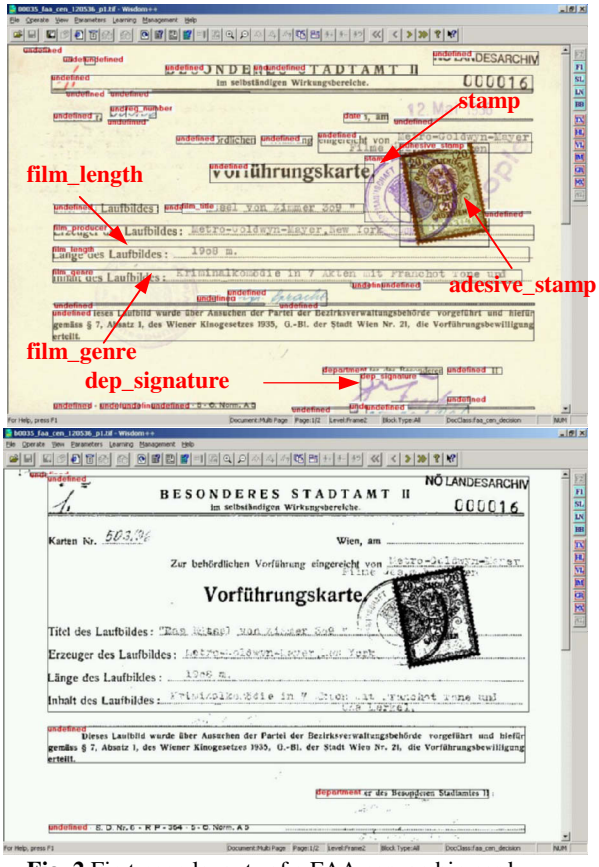

Fig. 2 First page layouts of a FAA censorship card.

Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05)
1520-5263/05 $20.00 © 2005 IEEE

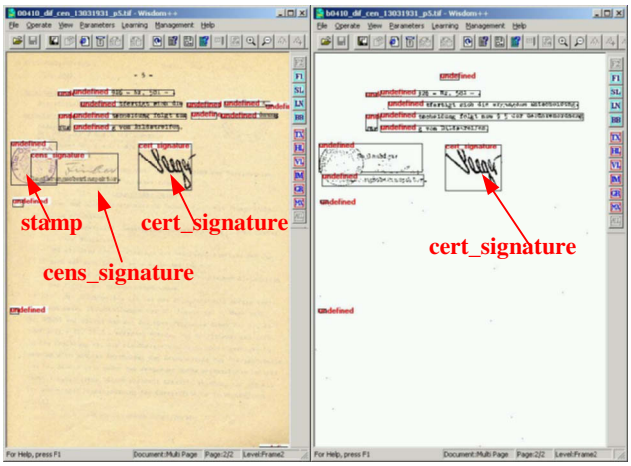

Fig 3 Second page layouts of a DIF censorship card.

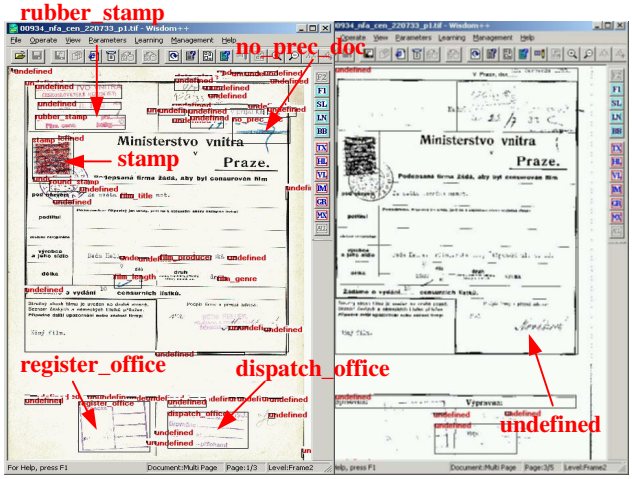

Fig. 4 First page layouts of a NFA censorship card.

colors are not differentiated.

The number of frame2 layout components that the user is able to label has been recorded. In the case of DIF cards, the color setting (i.e. 133 labels) is comparable with the b/w (i.e. 149 labels). This can be explained by the minor relevance of color in the case of DIF images. On the contrary, in the case of both FAA and NFA, several logical components are characterized by color information. Indeed, for the FAA class, 205 components have been labeled in the color setting against 140 in the b/w, while 64 against 12 for the NFA class.

4. Conclusions

In this paper, a new color-based layout analysis method has been proposed in order to meet challenges coming from processing censorship cards of European film archives of the 20ties and 30ties of the last century. A comparison of the method with the original b/w version has been provided. Results show that the color-based approach allows to isolate interesting blocks better than the previous version and to provide a more accurate base for understanding. For future works, we plan to evaluate the proposed approach in automatic/manual labeling.

Acknowledgements

The work presented in this paper is partial fulfillment of the research objective set by the ATENEO-2005 project on “Gestione dell'informazione non strutturata: modelli, metodi e architetture”.

References

1. Altamura O., Esposito F., & Malerba D.: Transforming paper documents into XML format with WISDOM++, *IJDAR*, 4(1), 2-17, 2001.
2. A. Antonacopoulos, D. Karatzas: The Lifecycle of a Digital Historical Document: Structure and Content. *ACM Symp. on Document Engineering*. 147-154. 2004
3. A. Antonacopoulos, D. Karatzas: Document Image Analysis for World War II Personal Records. *Workshop DIAL'04* , 23-24, 336-341. 2004.
4. H.-D. Cheng, X.H. Jiang, Y. Sun, J.L. Wang: Color image segmentation: advances and prospects. *Pattern Recognition*; 34: 2259-81, 2001 .
5. I. Frommholz, H. Brocks, U. Thiel, E. Neuhold, L. Iannone, G. Semeraro, M. Berardi, M. Ceci: Document-centered Collaboration for Scholars in the Humanities - The COLLATE System. *In Proc.ECDL'03*, 2003.
6. B. Gatos, K. Ntzios, I. Pratikakis, S. Petridis and T. Konidaris, S. J. Perantonis: A Segmentation-Free Recognition Technique to Assist Old Greek Handwritten Manuscript OCR. *DAS2004*, LNCS 3163, 63-74, 2004.
7. M. Gervautz, W. Purgathofer: A simple method for color quantization: octree quantization. New Trends in Computer Graphics, *Proc. of Computer Graphic Int.* 219-231, 1988.
8. H. Hase, M. Yoneda, S. Tokai, J. Kato, C.Y. Suen: Color segmentation for text extraction, *Int. Journal of Document Analysis and Recognition (IJDAR)* 6(4), 271-284, 2004.
9. J. He, A.C. Downton: Configurable Text Stamp Identification Tool with Application of Fuzzy Logic. *In proc. of DAS2004*, LNCS 3163, 201-212, 2004.
10. F. Le Bourgeois, H. Kaileh: Automatic Metadata Retrieval from Ancient Manuscripts, *In proc. of Document Analysis Systems DAS2004*, LNCS 3163, Italy, 75-89, 2004.
11. C.A.B. Mello and R.D.Lins: Image Segmentation of Historical Documents. *Visual 2000: 3rd Int. Conference on Visual Computing*, Mexico City, 2000.
12. S. Nicolas, T. Paquet, L. Heutte: Enriching Historical Manuscripts: The Bovary Project. *In proc. of Document Analysis Systems DAS2004*, LNCS 3163, 135-146, 2004.
13. T. Perroud, K. Sobottka and H. Bunke: Text Extraction from Color Documents – Clustering Approaches in Three and Four Dimensions, *ICDAR'01,* 2001.
14. S.N. Srihari, & G.W. Zack: Document Image Analysis. *Proc. of Int. Conf. on Pattern Recognition*, 434-436, 1986
15. K.Y. Wong, R.G. Casey, F.M. Wahl: Document analysis system. *IBM Journal of Research Development* 26(6),1982.