Pattern Anal Applic (2013) 16:679–693
DOI 10.1007/s10044-012-0282-x

INDUSTRIAL AND COMMERCIAL APPLICATION

A hierarchical and scalable model for contemporary document image segmentation

Asma Ouji · Yann Leydier · Frank LeBourgeois

Received: 24 June 2011 / Accepted: 4 July 2012 / Published online: 20 July 2012

Abstract In this paper, we introduce a novel color segmentation approach robust against digitization noise and adapted to contemporary document images. This system is scalable, hierarchical, versatile and completely automated, i.e. user independent. It proposes an adaptive binarization/quantization without any penalizing information loss. This model may be used for many purposes. For instance, we rely on it to carry out the first steps leading to advertisement recognition in document images. Furthermore, the color segmentation output is used to localize text areas and enhance optical character recognition (OCR) performances. We held tests on a variety of magazine images to point up our contribution to the well-known OCR product Abby FineReader. We also get promising results with our ad detection system on a large set of complex layout testing images.

Keywords Color segmentation · Document image · Noisy image · Text detection · Advertisement classification

1 Introduction

Nowadays, we encounter more and more digitized documents with overlaying color layers owing to DTP (Desktop publishing). However, few researches processing such images exist in the literature. Even the existing ones target specific applications such as mixed raster content (MRC) [2]. Without prior processing of the colors in some document pages, several applications, such as optical character recognition (OCR) and layout segmentation, cannot be efficient. Color information is imperative for further issues such as advertisement detection.

Digitized documents are commonly spoiled by a conventional series of operations (printing, digitization, image compression, etc.) that affect the original colors and introduce undesirable ones.

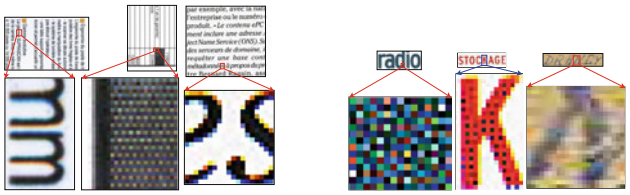
We propose to go back in time, that is to get closer to the document as it was before being spoiled, like it was designed by its author. This will enhance the image sharpness, remove noise and render more efficient many applications such as text detection, image classification, OCR, etc.

The first distortion that affects a document is its printing. Printers usually use halftones of four colors (cyan, magenta, yellow and black) to simulate the original colors of the document. Though humans should not perceive the halftoning, scanners can. Furthermore, the perceived colors are not exactly equal to the original.

Most of the scanners, especially when they are improperly set, introduce several kinds of distortions in the digitized images. Figure 1 shows two kinds of color noise commonly encountered in digitized documents. We call "saturation noise" the chromatic pixels around black strokes (Fig. 1a). This noise is often encountered on images digitized by linear cameras. We call "Hue noise" the altered pixels within chromatic areas (Fig. 1b). This is generally caused by a digitization resolution that does not match the halftoning. Low resolution also affects the gray-level elements: it spreads out some black strokes so that they appear gray in the resulting image; thus binarization would mangle the patterns. Finally, lossy image compression methods intensify the distortions by introducing additional noise.

In Sect. 2, we introduce a generic and versatile color processing system for document images. Such a processing is useful for diverse applications like OCR, layout segmentation, block classification, etc. We will demonstrate

A. Ouji (✉) · Y. Leydier · F. LeBourgeois
Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205,
20 av. Albert Einstein, Villeurbanne 69621, France
e-mail: asma.ouji@liris.cnrs.fr



(a) Saturation noise **(b)** Hue noise

Fig. 1 Noise samples in document images

the possibilities offered by our method on two applications: OCR segmentation improvement and advertisement detection. Respective details and discussion are reported in Sect. 3 and 4. These two applications are important steps in implementing the first advertisement recognition system for press images.

2 A generic multi-layer color segmentation system

2.1 Overview

We aim to get as close as possible to the original colors of the document. To do so, we propose the color segmentation scheme in Fig. 2.

- We first separate the chromatic layer from the achromatic one. A given pixel is called chromatic if it has a defined hue (red, green, blue, yellow, etc). Otherwise, it is achromatic (shades of gray, including black and white). This step is fundamental as chromatic and achromatic pixels cannot always be treated in the same way. Indeed, it would be meaningless to apply some processes that examine the hue values to achromatic pixels, since the hue for these pixels is either undefined, or unreliable 19. . Additionally, the saturation noise is removed at this stage.
- The chromatic layer is split into monochromatic and multi-chromatic layers:
  - A monochromatic layer consists of elements printed in flat tints; i.e. representable by only one color such as text. A monochromatic layer's quantization may not imply any information loss.
  - The multi-chromatic layer corresponds to the photo-like areas. Let us remind that this paper is designed to document images; applying our global approach on natural images would be uninteresting.
- The achromatic layer is split into gray-scale, black and white layers:
  - The B and W layer consists of the zones that were originally black and white. Its binarization does not cause any Information loss. It would rather enhance the contrast and remove some noise.
  - The gray layer usually consists in graphical elements that would be spoiled if binarized.

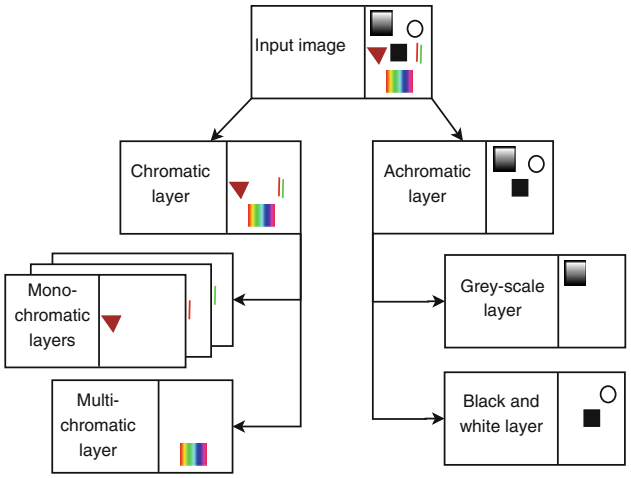


Fig. 2 Color separation outline

The work reported in this paper is part of an industrial framework. Therefore, throughout this paper each decision will be driven by both accuracy and speed concerns. All the proposed approaches do not require any document model or any a priori information on the document class.

As a part of industrial needs and constraints, our color segmentation system encompasses several self-contained and complementary steps. Given the necessary input, each step is reusable independently of the other ones. For instance, if we know in advance that the input image is gray-scale, only the achromatic split can be executed.

Section 2.3 estimates a preliminary stroke thickness measure that will be used in the following sections.

2.2 Previous work

Very few works dealing with the chromatic/achromatic separation exist in the literature. In [21], the separation is simply carried out by thresholding the saturation channel. Such a method is not suited to noisy images since some achromatic noisy zones would be detected as chromatic ones. Karatzas et al. [19] perform the chromatic/achromatic separation in web images based on the human perceptual model. This method consists in a thresholding of the HSV colors based on the human perception. However, web images do not include much color noise. Thus, the chromatic /achromatic separation would be simpler here than on digitized images.

Once the chromatic and achromatic layers are created, each one shall be segmented in turn. A large variety of

color quantization methods exist in the literature [3, 37]. Most of them are not suited to document images. In [23, 24], a color segmentation adapted to noisy ancient documents is presented. An elaborated k-means classification is applied to perform the segmentation. As we do not have any prior knowledge about the input images, we cannot take advantage of such an algorithm since the number of classes and the initial samples have to be set in advance. Many algorithms handle color classification and automatically determine the number of classes [1, 13, 33, 45]. Such algorithms group together in visually similar colors. However, the colors that are not sufficiently represented in the image are assigned to wrong classes and this may cause a significant loss of information that we have to avoid. In [34, 35], Pujol et al. present a quantization method that optimises the number of color classes. Their method has an effect in merging similar color classes while extracting poorly represented colors depicting significant details (e.g. a small bird in a sky image). Such a method fits natural images, not noisy document images since the color noise would create wasted classes. Another color classification algorithm using a new perceptual color-space is presented in [38]. The classification is achieved by recursive analysis of histograms peaks. The histograms analysis seems to be applicable on our document images. Yet, we do not use any human perceptual model. A color reduction algorithm suited to document images is presented in [30]. It is a mean-shift-based procedure using 3D color histograms and an edge preserving smoothing filter. Such an approach produces good results on text images, but it implies too much information loss on document images including non-text (photo) regions.

We segment the achromatic layer regardless of the chromatic one since they have different features. As we aim to identify the black and white (B.W.) regions to binarize them, we separate the achromatic pixels into two sub-layers: the gray layer and the B.W. one. A lot of locally adaptive binarization methods suited to degraded document images exist in the literature [5, 10, 17, 27, 31, 39]. Several authors pre-process [28] or restore [8] the images before binarizing it to insure better results. However, even if the binarization method is very efficient, the binarization of gray (graphical) zones causes a penalizing loss of information. An alternative to this problem is to binarize after a structural decomposition. The logical segmentation [4, 6, 16] identifies text elements, graphical zones, table, etc. Only text zones are usually binarized afterwards. Such an approach would provide good results on text elements. However, we cannot apply structure-based methods since they are time consuming and vastly dependent on the document class. Furthermore, some bitonal graphics or large title would not be binarized since they would not be identified as text elements. Therefore, we are proposing an achromatic segmentation approach that comes before any binarization process.

2.3 Foreword: stroke thickness estimation

Most of the document image processing methods depends on a resolution-related parameter (e.g. width of a structuring element, size of a convolution matrix, etc.). To determine such values, it would be rather futile to lean on the digitization resolution. Indeed even if all documents were digitized in the same resolution (some providers proclaim that 300 dpi is universally suitable), each document has its own typographic characteristics. Modern letters and invoices are usually composed with 10 or 12 points fonts, but advertisements, flyers and journals have unstable typographies. Given a specific size, the strokes of a font may have variable thickness depending on the typeface anatomy (see Fig. 3).

In order to ensure that the algorithms presented in this paper will be free of a "resolution" parameter, we will base all our metrics on an estimation of the font's stroke thickness.

A quick and easy way to estimate the strokes' width and height without binarizing the image is to compute the autocorrelation of the image along the horizontal and vertical axes. The estimation would obviously be more accurate if we determined the main orientation of the strokes, but this would cost much time for a little gain.

Let $T_{\mathrm{h}}(I, \delta)$ be the translation of the gray-scale image I (defined on the plane Ω) of δ pixels along the horizontal axis. We define the sequence $\left(\mathcal{D}^{\mathrm{h}}(I)_n\right)_n$ with:

$$\begin{aligned} \mathcal{D}^{\mathrm{h}}(I)_0 &= 0 \\ \mathcal{D}^{\mathrm{h}}(I)_n &= \sum_{p\in\Omega} \|I(p) - T_{\mathrm{h}}(I,n)(p)\| \end{aligned} \tag{1}$$

To estimate the mean stroke width $\mathcal{S}_{\mathrm{w}}$ of an image, we compute the sequence $\left(\mathcal{D}^{\mathrm{h}}(I)_n\right)_n$ until the slope becomes lower than 0.1. Then, we set $\mathcal{S}_{\mathrm{w}} = n$.

The computation of the mean strokes' height $\mathcal{S}_{\mathrm{h}}$ goes the same with the vertical translation $T_{\mathrm{v}}(I, \delta)$ and the sequence $\left(\mathcal{D}^{\mathrm{v}}(I)_n\right)_n$.

We tested this algorithm on various documents (see Fig. 4) and we measured the strokes' width and height manually. The strokes' thickness range was from 2 to 10 pixels inclusive. The mean error between $\mathcal{S}_{\mathrm{w}}$ and the measures was 1.25 pixels, which is quite good since the images are never binarized in the process. Due to

Fig. 3 Two fonts with the same size but with different stroke thickness

Fig. 4 Samples of the images used to test the strokes' thickness estimation

prominence of vertical strokes in the writing, the mean error between $\mathcal{S}_h$ and the measures was 1.75 pixels.

Since a stroke's edge in a gray-scale image is nearly always smoothed, it is impossible to precisely measure its thickness. Therefore, having a mean error lower than 2 pixels tends to prove that our estimator is accurate.

In the following, as we do not know the orientation of the document images, we will use the estimator $\mathcal{S}_t = \max(\mathcal{S}_w, \mathcal{S}_h)$. Such a measure will be used to render parameter-free several stages of our color analysis system.

2.4 Color segmentation

2.4.1 *Chromatic/achromatic split*

The chromatic/achromatic separation consists in creating a binary mask where each entry indicates whether the corresponding pixel is chromatic or not.

The final mask $\mathcal{M}_F$ is computed using two main data: a saturation measure S^* and the coarse mask $\mathcal{M}_C$:

$$\text{Saturation}(S^*) \longrightarrow \text{Coarse detection}(\mathcal{M}_C) - \rightarrow \text{Localisation}(\mathcal{M}_F)$$

Computation of the saturation The chromatic/achromatic split is usually achieved by thresholding the saturation channel [21]: the chromatic content is generally highly saturated whereas the achromatic pixels have low values of saturation. However, computing the saturation on dark pixels is insignificant, since its computation involves division by low lightness values. Figure 5b shows black (achromatic) zones having greater values of saturation than the green (chromatic) background.

For this reason, we introduce a new measure of pseudo-saturation, which is defined below, instead of the traditional saturation.

$$\begin{aligned} &I : \Omega \rightarrow \mathbb{N}^3 \\ &p \mapsto (R_p, G_p, B_p) \end{aligned} \tag{2}$$

The pseudo-saturation S^* of I is defined by:

$$\begin{aligned} &S^*(I) : \Omega \rightarrow \mathbb{N} \\ &p \mapsto \max(|R_p - G_p|, |R_p - B_p|, |G_p - B_p|) \end{aligned} \tag{3}$$

The asset of this pseudo-saturation is that it is valid on dark pixels (as well as light ones) since its computation does not imply any division as opposed to standard saturation formula (see Fig. 5c).

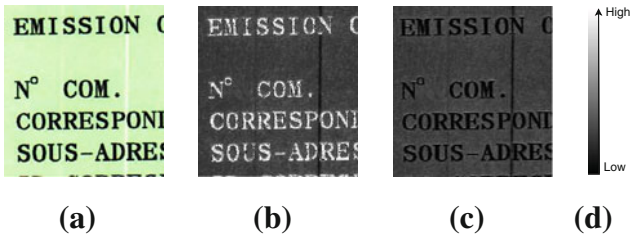


Fig. 5 Saturation versus pseudo-saturation: **a** Color image, **b** saturation image (HSV), **c** pseudo-saturation image, **d** Saturation caption

Whatever the saturation's formula, we cannot get rid of the saturation noise directly. Figure 6 represents the pseudo-saturation output of a sample color image (blue handwritten text on the top, black printed text below). It shows that it is impossible to find an appropriate threshold that removes the noise and detects chromatic regions at the same time.

To get rid of the color noise, we operate a coarse detection of the chromatic areas. Then, we refine the results by localizing those zones accurately. Such a decomposition has a practical purpose: it is possible to stop at the end of the first step if we just want to know whether an image includes chromatic (or achromatic) regions, which is time saving.

Coarse detection of chromatic zones The purpose of this section is to identify all the chromatic zones and to remove the saturation noise, even if the localization is not precise. The output at this level will be the coarse mask $\mathcal{M}_C$.

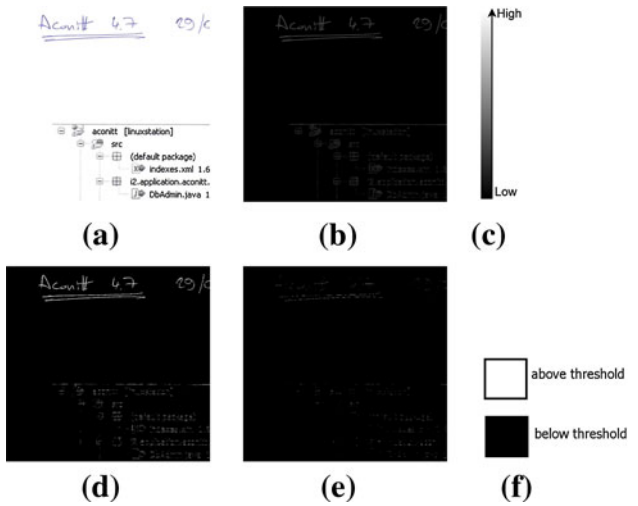


Fig. 6 **a** Color image, **b** pseudo saturation image, **c** pseudo-saturation caption, **d** pseudo-saturation thresholded at 25 %, **e** pseudo-saturation thresholded at 40 %, **f** Thresholded images caption

We begin by reducing the image size using a Gaussian re-sampling. The scale reduction smoothes the image; thus, it eliminates some noise. Furthermore, processing a reduced image is faster than handling the full size one. The appropriate scale reduction factor should be computed automatically. It is fixed to $\mathcal{S}_t$. Such a value reduces the color noise without destroying the small chromatic zones (such as text) since the scale depends on the strokes' thickness. Let us remind that the use of $\mathcal{S}_t$ renders all our processings independent to the acquisition resolution. Furthermore, this scaling standardizes the font metrics in the images. Thus, it is possible to process all the images using the same parameters as they all have the same stroke width/height.

As the saturation noise is located next to black text pixels, we apply a morphological color dilatation of the dark elements. This replaces the remaining chromatic noise with regular text pixels without destroying the truly chromatic zones. Since the image size has already been normalized, the size of the dilatation structural element can be common to all images and is fixed to a low value, 3×3, to avoid further loss of information. As the noise is always narrower than the text's strokes, we can remove noise without destroying the text. We call the resulting image I^r.

The pseudo-saturation measure is computed over I^r (see Fig. 7b). The image $S^*(I^r)$ is then thresholded to create the coarse mask $\mathcal{M}_C$. The pseudo-saturation values under the threshold correspond to achromatic colors; the other ones represent chromatic regions. The pseudo-saturation threshold estimation relies on $S^*(I^r)$'s histogram's peaks.

The mask displayed in Fig. 7c shows that the color noise is successfully removed. The chromatic area localization is refined in the next section.

Accurate chromatic/achromatic split Provided $\mathcal{M}_C$, we can now precisely extract the shapes and create $\mathcal{M}_F$.

This can only be done using the full scale image. Therefore, we will combine the previous mask $\mathcal{M}_C$ (Fig. 7c) with a new one called $\mathcal{M}_A$ given by thresholding the full-sized pseudo-saturation image (Fig. 6d).

The most intuitive combination method may be the logical intersection of $\mathcal{M}_C$ and $\mathcal{M}_A$. However, such an operation is not efficient on images with a chromatic background (see Fig. 8).

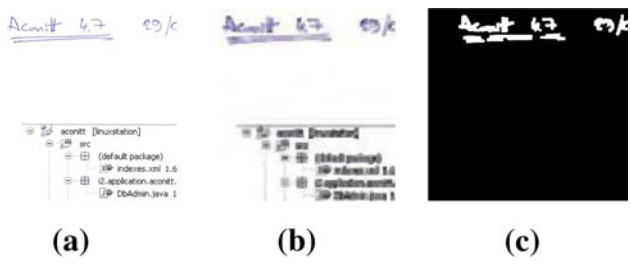


Fig. 7 Chromatic/achromatic mask computed on the reduced image: **a** Original image I, **b** reduced and dilated image I^r, **c** $\mathcal{M}_C$

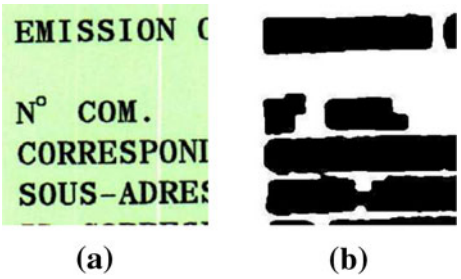


Fig. 8 **a** Green background image, **b** mask obtained by logical intersection of $\mathcal{M}_C$ and $\mathcal{M}_A$: approximately the same as $\mathcal{M}_C$

To overcome this problem, we propose to compute the final mask as the logical intersection of $\mathcal{M}_C$'s bounding boxes (in the original scale) and $\mathcal{M}_A$. The red boxes in Fig. 9a represent the bounding boxes extracted from $\mathcal{M}_C$; Fig. 9b displays the mask $\mathcal{M}_A$ and Fig. 9c $\mathcal{M}_F$.

Results The chromatic/achromatic split results are very satisfying. Indeed, the color detection precision reaches 99.88 % [32]. The engine has managed to remove all the color noise in Figs. 1 and 10.

2.4.2 *Chromatic segmentation*

A set of chromatic pixels grouped together constitute either a monochromatic or a multi-chromatic zone. Monochromatic colors are separated from one another. Multi-chromatic zones (photos) are kept unchanged. Text elements above a multi-chromatic background are considered as a part of a multi-chromatic zone; thus the area is not extracted here.

As mentioned earlier, we are interested in how the document is composed, not how it is perceived. Thus, we chose the HSV color-space to represent colors. The most important factor in terms of color discrimination in this color-space is Hue [29, 42]; so our chromatic split is based on the Hue channel.

When an image contains both monochromatic and multi-chromatic zones, the global Hue histogram is impossible to analyze. Thus, each chromatic zone is processed at a local level.

A Hue histogram with wide peaks corresponds to a multi-chromatic zone, whereas narrow peaks indicate the presence monochromatic areas. The multi-chromatic areas

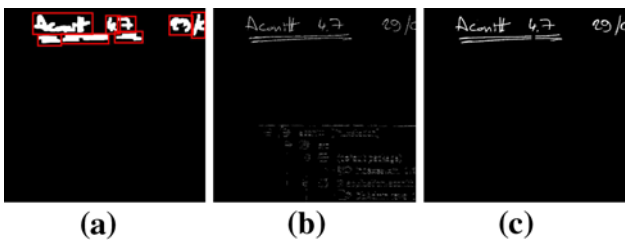


Fig. 9 **a** $\mathcal{M}_C$, **b** $\mathcal{M}_A$, **c** $\mathcal{M}_F$

Fig. 10 Chromatic/achromatic split results

are thus easily identified. A given peak is considered wide if it is larger than 10 % of the Hue spectrum (the ratio 10 % has been determined experimentally).

The monochromatic zones can optionally be quantified, if the algorithm is used in a MRC [2] process for instance. The achromatic regions are obviously ignored at this stage.

Definition of the local processing zones We assume that it is more reliable to process small zones (such as characters) together than processing each one independently of its context. For this reason, we use the bounding boxes of the mask $\mathcal{M}_C$ (Sect. 2.4.1) instead of $\mathcal{M}_F$ to define the processing zones.

Multi-chromatic color separation We will call $\mathcal{H}_i^l$ the local histogram of the zone i. Histograms containing large peaks always correspond to multi-chromatic zones and are deftly classified as such.

Monochromatic color separation To improve the consistency, we will base our segmentation on a color model corresponding to the set of colors present in the image, excluding the multi-chromatic areas.

From now, we ignore the $\{\mathcal{H}_i^l\}_i$ histograms associated with the multi-chromatic zones. A global hue histogram $\mathcal{H}^g$ is created from the addition of all the remaining $\{\mathcal{H}_i^l\}_i$ histograms. Thus, the global histogram consists of exclusively discrete chromatic colors. The color model is deduced from the global histogram's modes. We will create a color layer for each mode.

The classification is handled at the pixel level, not at the connected component or the zone level, since two distinct colors can be present side by side in the same zone.

Most of the chromatic zones include hue noise pixels (altered color pixels inside the homogeneous chromatic zones). Figure 11 displays two samples of layers resulting from a simple classification where each pixel's hue is compared to the color model. The resulting layers show that such a method cannot handle hue noise (*e.g.*: purple pixels around the red cross).

The sole global histogram is not enough to determine whether a pixel is part of the noise or a relevant color pixel.

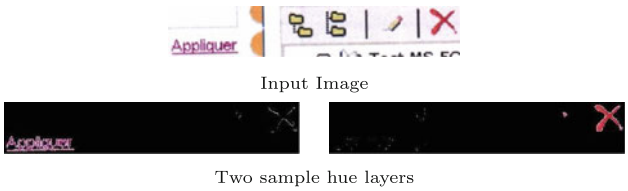


Fig. 11 Hue noise not removed by simple classification

The local histogram is not sufficient because of inevitable local variations of hue in the document due to electronic noise and to the mix of colors used by printers.

Therefore, we propose a classification method based on a double validation. Both the local and the global hue histograms are used to validate the pixel assignment. A noise pixel's hue is drifting away from the modes and to be locally assigned to a random color class. As the global histogram typically contains more modes, the same pixel is likely to be globally assigned to another random color class. Therefore, a pixel whose local and global classifications disagree is considered to be noise.

Let $\mathcal{M}^g(P)$ be the nearest mode to a given pixel P in $\mathcal{H}^g$, $\mathcal{M}^l(P)$ the nearest mode to P in its corresponding local histogram $\mathcal{H}_i^l$. Because of the color variations in the image, a local mode can have a value slightly different from its global counterpart. Therefore, we cannot compare these $\mathcal{M}^g(P)$ and $\mathcal{M}^l(P)$ directly. The correct way is to compute $\mathcal{M}^g(\mathcal{M}^l(P))$, the nearest mode to $\mathcal{M}^l(P)$ in $\mathcal{H}^g$. If $\mathcal{M}^g(P) = \mathcal{M}^g(\mathcal{M}^l(P))$ then P is assigned to the layer associated with $\mathcal{M}^g(P)$; else P is a color noise.

The process is illustrated in Fig. 12:

- case of the blue pixel ($Hue(P) = 150$): $\mathcal{M}^g(P) = 159$, $\mathcal{M}^l(P) = 160$, $\mathcal{M}^g(\mathcal{M}^l(P)) = 159$; here $\mathcal{M}^g(P) = \mathcal{M}^g(\mathcal{M}^l(P))$, so the pixel is assigned to the blue layer;
- case of the pink pixel ($Hue(P) = 5$): $\mathcal{M}^g(P) = 8$, $\mathcal{M}^l(P) = 21, \mathcal{M}^g(\mathcal{M}^l(P)) = 21$; this time $\mathcal{M}^g(P) \neq \mathcal{M}^g(\mathcal{M}^l(P))$, so the pixel is rejected as hue noise.

Results Figures 13 and 14 show the respective results of three input images. The chromatic layers are displayed on a black background. As expected, the multi-chromatic zone in Fig. 13 is not decomposed. In this figure, the pale pink dithering is a separate layer. Figure 14 shows that even the smallest color components are correctly separated.

2.4.3 Achromatic segmentation

Zone classification The achromatic layer is segmented into a B.W. and a gray sub-layers in a similar way as the previous section.

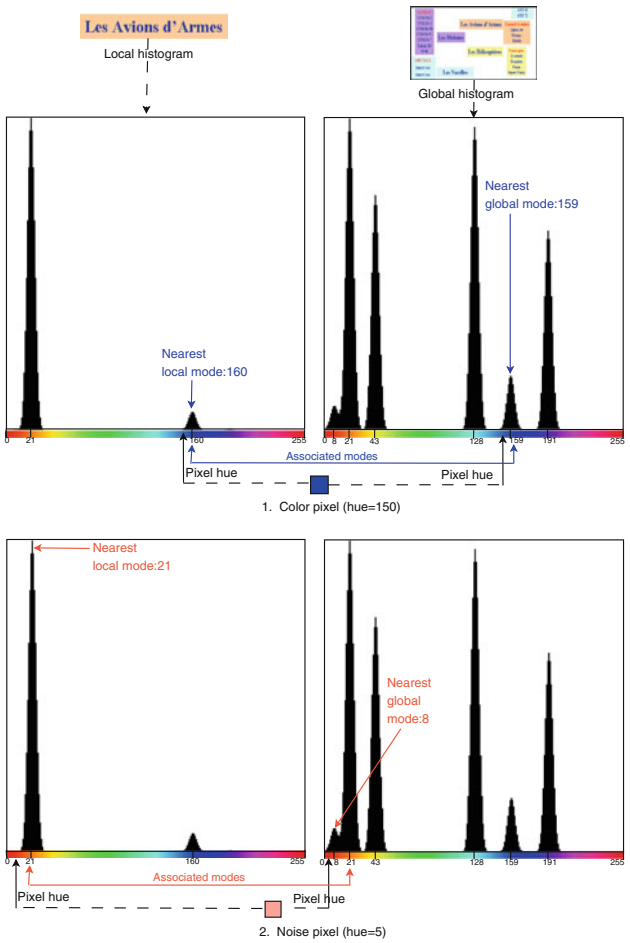


Fig. 12 The classification method applied to a chromatic pixel and a noise pixel

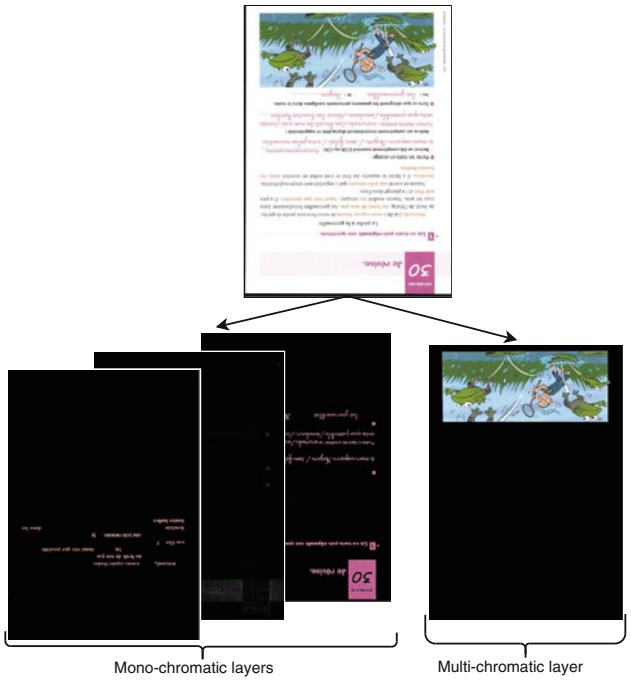


Fig. 13 Chromatic split

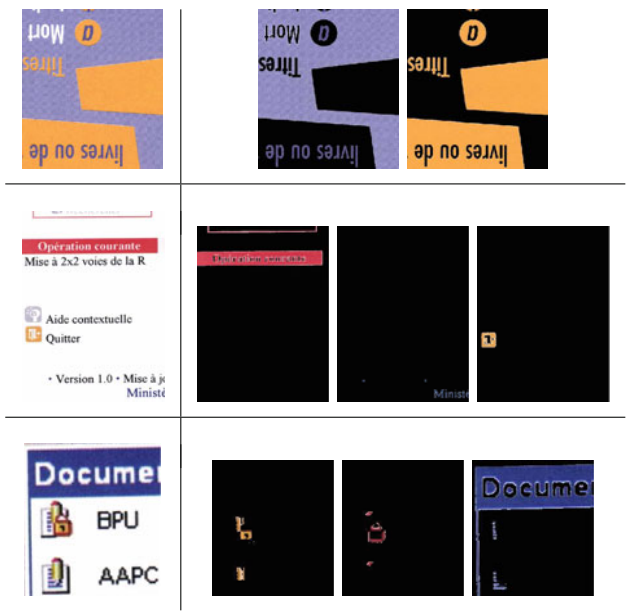


Fig. 14 Hue layers

Using the global luminance histogram's first and last peaks, we compute black and white thresholds. The lightest pixels (whose luminance is greater than $\mathcal{T}_B$) are immediately assigned to the B.W. layer. The white area in Fig. 15c corresponds to those pixels. The remaining pixels are merged into distinct processing zones (connected components), so that each zone is locally classified. Figure 15d shows the resulting processing zones (in white).

Digitization, especially bad quality digitization, spreads out some black elements so that they appear gray in the resulting image. Thus, classifying the pixels independently of their neighborhood would be completely unreliable. For this reason, our classification candidates are the processing zones.

Large-sized zones are probably graphical elements and they are more likely to be gray-scale. Hence, a specific method is used to classify them. Here again, the $\mathcal{S}_t$ measure is used to define the size separation (large and small zones). We estimated that a character's size is generally about $11\mathcal{S}_t \times 11\mathcal{S}_t$ pixels. Here, we consider that a zone is 'large' if it encompasses more than 30 characters, i.e. its area is greater than $3500\mathcal{S}_t^2$.

Large zones may correspond to huge black titles. Such zones feature a small ratio of gray noise. Therefore, a large zone is assigned to the B.W. layer if and only if more than 90 % of the pixel luminance values are under the $\mathcal{T}_B$ threshold. The ratio 90 % has been experimentally determined: we achieved the best results with such a value on a

Fig. 15 **a** Input image, **b** $\mathcal{M}_F$, **c** obvious white pixels, **d** processing zones

dataset composed of various images. Please notice that a light modification of this parameter (85–95 %) does not affect the results significantly.

A feature vector is associated with each small zone. It consists of the first peak's abscissa in the local histogram and the width of that peak. Zones with large peaks are more likely to belong to the gray class.

Since the $\mathcal{T}_B$ threshold is computed on the global histogram, all the zones have contributed to the computation of this value. However, small zones contain lighter pixels than large zones (due to a generalized disrespect of the Shannon-Niquist sampling theorem) and $\mathcal{T}_B$ is not adapted to correctly classify these zones. Thus, a new threshold $\mathcal{T}'_B$ is introduced. It is extracted from an histogram composed by adding all the local histograms associated with small zones.

A small zone is assigned to the B.W. layer if its features are respectively lower than the $\mathcal{T}'_B$ threshold and the average value of the chosen peaks' width.

Contextual classification Now that all the zones are classified, small zones assignment is reviewed to rectify confusing situations that were impossible to classify correctly without any knowledge on their neighborhood. Let us consider a text image where small zones correspond to characters. It is unlikely to encounter a gray letter among a black text. Hence, we assign all isolated gray zone to the B.W. class so that the classification is consistent.

Rules detection Rules are straight lines, table borders, etc. Such elements are usually composed by thin and shaded strokes. Hence the previous classification assigned them to the gray layer. This section aims to detect rules to reassign them to the B.W. layer. The following algorithm is applied on large gray zones.

1. A morphological gray-scale erosion is applied on the image. The structural element size is fixed to $(2 \cdot \mathcal{S}_t) \times (2 \cdot \mathcal{S}_t)$. This erosion should clear all the thin lines. Indeed, if the stroke width is $\mathcal{S}_t$, such an erosion should delete all the dark strokes.
2. If there is no more dark pixels (i.e. pixels with a luminance below $\mathcal{T}_B$), the zone is considered to contain rules. Otherwise, the region remains in the gray layer.

Results A sample of the resulting films is shown below. The gray films are displayed in gray-scale on a blue background (for visibility reasons). The B.W. film is binarized and displayed on a white background. Results are very satisfying even on poor quality documents (see Fig. 16).

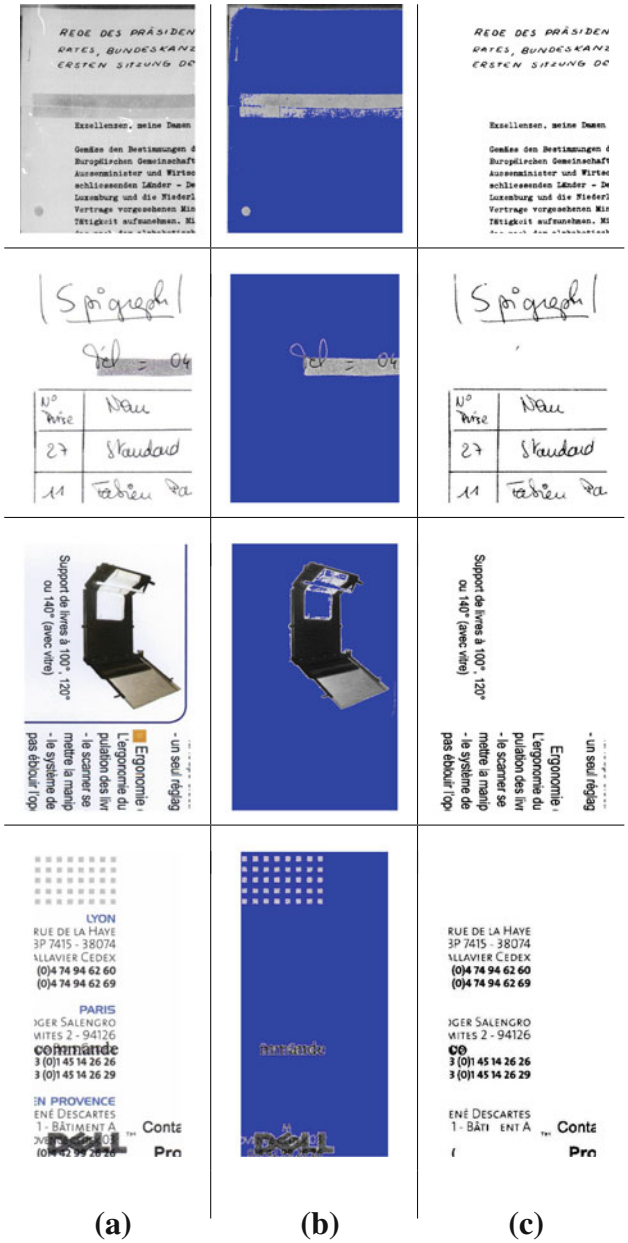


Fig. 16 **a** Input images, **b** gray layers, **c** B.W. layers

The last example in Fig. 16 shows that the overlaying text lines creating ambiguous configurations are assigned to the gray layer. This avoids penalising information loss by binarization.

Fig. 17 *1*. Input images, *2*. Color processed images

2.4.4 Color segmentation result's summary

A final output image given by stacking up all the layers is associated with each input image. A sample of the resulting images is displayed in Fig. 17. We can see that the recomposed images are cleaner than the original ones and color information is preserved.

Figure 18 shows that, unlike our approach, even the most competitive existing color reduction methods (DjVu [12] and the one in [30]) do not succeed to assign one color to one word which would penalize the text extraction stage. Let us remind that our method does not perform a blind binarization but separates color, gray, black and white shapes.

Using a 2.3 Ghz CPU machine, the mean execution time for an A4 image of resolution 300 dpi is about 4 s, i.e. the execution time is about 0.04 ms per 1,000 pixels. Table 1 shows the time ratio consumed by the principal stages of the color analysis algorithm.

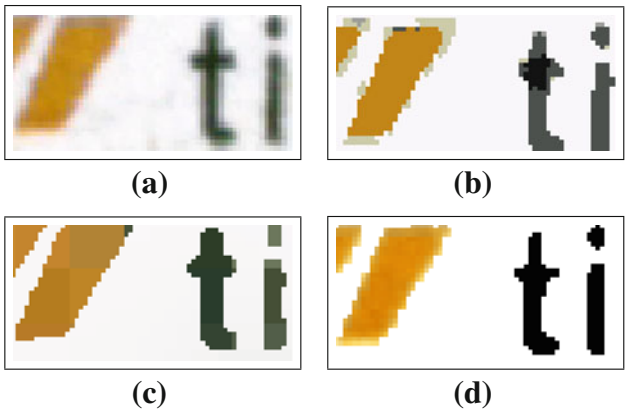


Fig. 18 Result comparisons: **a** input image, **b** color reduction [30], **c** DjVu [12], **d** our method

The execution time varies according to the image content: the algorithm is much faster on a simple layout than a sophisticated one. Furthermore, if we have some a priori information on the input image several steps can be omitted. For instance, if we desire to binarize all the achromatic regions, the third stage can be omitted, which makes the algorithm approximately three times as fast.

3 Application to Text localization to improve OCR results

A variety of approaches to text information extraction, that goes from detection to recognition, from images and video have been proposed for specific applications, including page segmentation, address block localization, license plate localization, etc. In spite of such extensive studies, there is still no general-purpose system. A survey of text information extraction methods is given in [18].

Several researches have been done in text extraction in compressed domain (MPEG, JPEG...) [26, 48]. However, they concern videos and use motion information to extract text. Our method processes compressed images as well as regular ones since the color segmentation stage efficiently filters the compression damage.

We chose a connected component-based method for text location to get profit from our color segmentation system. CC-based method using color reduction is also used in [15, 20]. As our color segmentation method adapts to each region's content, we can provide a text detection method that is suited to any context: black, white or colored text on white, black, colored or photo background. Such genericness distinguishes our method from the literature.

Text extraction and enhancement usually generate the input for an OCR algorithm. Indeed, the most sophisticated OCR

Table 1 Time ratio per step

Step	Chromatic/achromatic split	Chromatic split	Achromatic split
Time ratio	13 %	11 %	76 %

softwares cannot retrieve text on multi-chromatic background and are not efficient on noisy documents. Extraction and enhancement are independent issues. Color segmentation is related to text extraction, but cannot significantly improve the character shapes. Therefore, in this section, we will evaluate the OCR performance on text line segmentation.

3.1 Text extraction in monochromatic films

Inside each monochromatic and B.W. layer, text lines are composed by connected component grouping. The grouping criteria are horizontal alignment, similar heights and a small horizontal distance between the components.

This method detects horizontal and slightly slanted text lines. This restriction is, however, acceptable since vertical and highly slanted text are rare in press documents. Moreover, this kind of algorithm answers an important industrial constraint as it is fast.

3.2 Text extraction in multi-chromatic and gray films

The multi-chromatic and gray layers generally correspond to photos and may embed text. We use the cumulated gradients [22] method to estimate the approximate text position. We define the horizontal partial derivative d_x of a color pixel $p(x, y)$ as:

$$d_x(p) = \mathcal{M}/|\mathcal{M}| = \max\left\{\left|\frac{\partial R_p}{\partial x}\right|, \left|\frac{\partial G_p}{\partial x}\right|, \left|\frac{\partial B_p}{\partial x}\right|\right\} \quad (4)$$

The size of the accumulation window is set to $45 \times \mathcal{S}_t$ (this threshold corresponds to an average word width value that we computed on various documents).

Areas having high values of cumulated d_x indicate the presence of text lines (Fig. 19b). Thus, we track text inside such zones: each one is quantized (see Fig. 19c). For each color, connected components are extracted and we apply successively to them the same algorithm as in Sect. 3.1. Let us consider Fig. 19b to illustrate the process. The quantization gives rise to three color films. The connected components grouping method locates text in the 'yellow' components.

3.3 Results: Line segmentation enhancement

Obviously, it would be little interesting to apply the proposed segmentation system on documents with regular layout such as novels. Actually, our method is fit for colored documents with complex structures. Women's magazines are thus perfect candidate to demonstrate its advantages.

We held tests on 448 pages coming from the issue "3321" of the French magazine "Elle" and the issue "73 02 10" of "Glamour". The resulting ground truth consists of 14,303 text lines. The corpus includes text embedded in photo areas as well as regular text lines. The results will be given in terms of recall (R) and precision (P) values computed on the bounding boxes of the text lines. We used the Deteval experimental protocol to evaluate the results. Details on the method are reported in [43]. Let us remind that:

$$R = \frac{\#\,\text{correctly detected rectangles}}{\#\,\text{rectangles in the database}}$$

$$P = \frac{\#\,\text{correctly detected rectangles}}{\#\,\text{detected rectangles}}$$

Let us point out that our text detection approach is not designed to perform line segmentation, but to indicate the text positions. Thus, it is affected by over-segmentation and sub-segmentation problems resulting in low precision values.

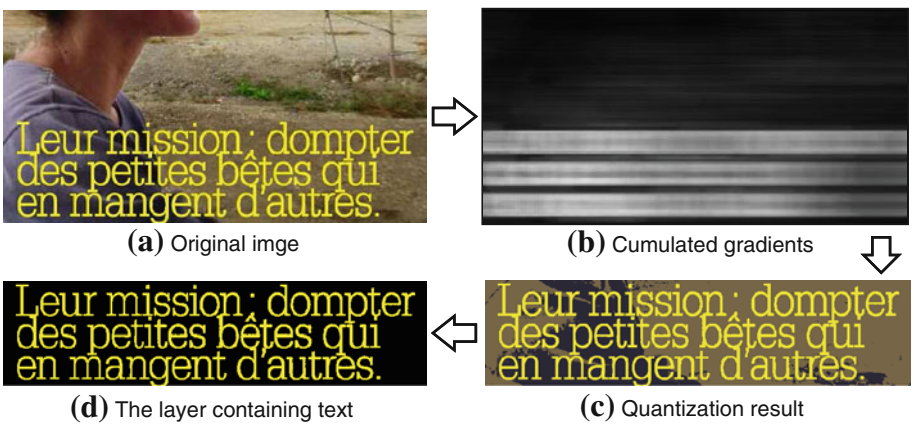


Fig. 19 Text extraction in multi-chromatic areas

Table 2 Line segmentation results on a database of 14,303 elements

Input	*R*	*P*
Lines from our text localization method	88.58	54.22
Finereader applied to original images	81.63	93.70
Finereader applied to images processed with our method	91.03	90.89

Consequently, we have decided to rely on the commercial product Abbyy Finereader 8.1 to compute the segmentation. Indeed, FineReader is known to achieve good segmentation (and recognition) rates. Furthermore, it will be easier to compare our system with the existing ones using a commercial product than using an ad-hoc method. To do so, we automatically generated images with the content of the text zones localized by our method. Such images are free of graphic zones but do contain the text that was embedded in photos.

Table 2 shows the results given on, respectively, the original images and our generated text images.

Taking advantage of the fact that Finereader tries to recognize the characters, we set a filter to remove lines composed of more than 50 % non-alphanumeric characters and lines containing less than 3 characters. Finereader's results recorded in Table 2 are achieved after this simple filtering that improved the precision by almost 10 % and keeps the recall unchanged.

Finereader's recall on the original image is lower that our method's recall. However, Finereader's recall being higher than ours on images generated from our text localization is explained by the method of evaluation (Deteval) that penalizes over-segmentation.

The recall amelioration is mainly due to our text detection inside multi-chromatic areas. Indeed, Finereader efficiently recognizes regular text lines but usually does not look for text in graphical areas (see result samples in Fig 20).

We notice a light precision drop when applying Finereader on our generated images. This is due to some regular textures in graphical areas that may be wrongly detected and recognized as text lines.

We studied the resulting lines and found that Finereader's recall on our generated images should be greater. It appears that a number of the text lines that we have correctly detected are dropped. This happens because of by fancy fonts or page layouts that Finereader cannot handle.

Fig. 20 **a** Finereader result samples, **b** caption

Comparing our results with the previous ones is difficult as we do not handle the same input. Indeed a number of researches related to text detection and recognition in videos exist in the literature. For instance, [44] achieves text detection at $R = 93.5$ in video frames. However, in document images, we obviously cannot profit from the frame frequency features to track text.

4 Application to advertisement detection and localization

Provided the previous text lines detection output and the color Information, we can build robust features to classify document images' blocks. In this section, we focus on a specific application: advertisement detection in magazine and newspaper images. This application answers two industrial problems:

1. removing ads from a corpus of articles to build a press review,
2. allowing an advertiser to check if a newspaper or journal did really publish all the advertisements that were ordered.

Figure 21 shows representative samples of advertisement and non-ad blocks. We can notice that the distinction between them can be a very complicated task, especially without semantic knowledge.

Several researches have been carried out to detect ads within web images and videos. In [25] and [11], image features and text features extracted from the HTML code are used to categorize web images. Semantic features are also used in [36] to remove advertisements from web pages. Ads detection in video frames usually relies on frame frequency features and audio features [41, 47] that we obviously cannot use. Color, texture and edge features are also used to detect ads in video streams in [40]. Texture features can be time consuming and not practical to handle by final users. Furthermore, they might not be discriminatory enough on contemporary printed documents. Multi-scale Gabor Filters are used to detect a special kind of advertisements in [46]. However, such an approach cannot be generalized to detect all the magazine ads.

Classifiers are usually SVM [7], AdaBoost [9] and neural networks [14].

4.1 Blocks extraction and pre-classification

At first, we extract text blocks and graphic blocks. Text blocks contain only text lines whereas graphic blocks contain graphic elements (photo, figure, etc.) and possibly text lines.

Neighboring text lines with similar heights are grouped together to create a text block. Such blocks generally correspond to articles.

Graphic blocks are directly inferred from multi-chromatic and gray zones and from connected components from any other layer that were not included in a line during the text localization step. Text lines embedded in such blocks are naturally a part of them.

Only graphic blocks including text lines are candidates for ads detection. Indeed, an advertisement always contains graphical elements, at least a frame. Similarly, ads necessary include text.

(a) an *Ads* sample **(b)** a *Graphic* sample

Fig. 21 Candidate blocks extracted from the same issue. **a** An *Ads* sample, **b** a *Graphic* sample

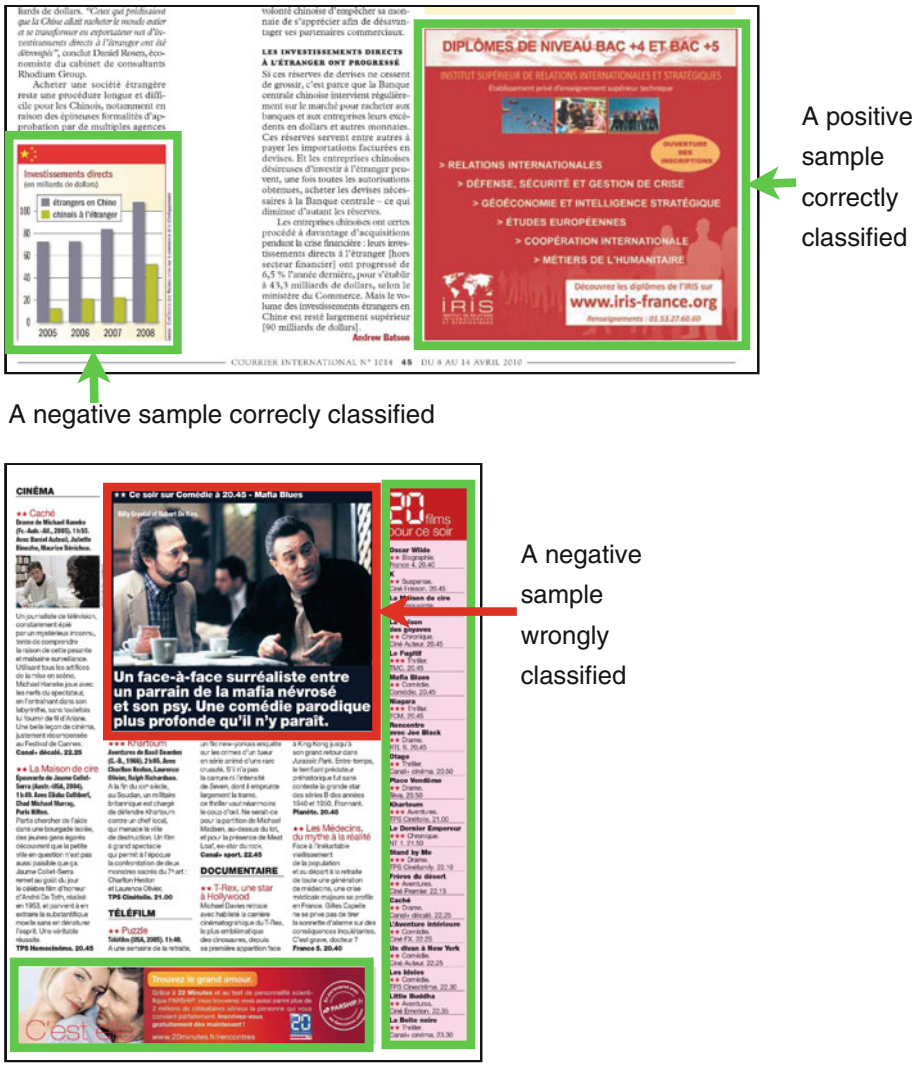


Fig. 22 Classification results of five sample blocks

4.2 Classification features

In order to specifically evaluate our method, we choose to use only visual feature and no semantic ones. Moreover semantic features are best used to find a given ad in the corpus (e.g. a product or merchant's name, words that are specific to the kind of product that is advertised...). As we want to build a system able to retrieve any advertisement, it is impossible to select universal enough semantic data.

The main visual properties of advertisement blocks are: colored text and/or colored background, multiple colors and/or photo areas, irregular text lines (varying widths and heights), few text lines, the block is close to an edge of the page, the block is not included in a text article. Unfortunately, these properties are never all relevant at the same time.

We expressed them in terms of a set of numeral features associated with each candidate block. This set consists of:

- the (respectively) black, white, gray, monochromatic and multi-chromatic pixels ratio;
- the number of different monochromatic colors;
- the percentage and density of (respectively) black, white, gray, monochromatic and multi-chromatic text lines (multi-chromatic text lines are text lines included in a multi-chromatic area);
- the variance of the text lines width and height;
- the intersection area ratio with the other blocks;
- the block position within the page.

Thus, the feature vector's dimension is 22.

4.3 Classification results

We used AdaBoost [9] for its ability to select the most effective features for each document.

We held tests on about 400 simple and double pages coming from various word-wide magazines and newspapers. The data set contains issues of press in several languages digitized with different qualities and resolutions provided by our industrial partner. The total number

Table 3 Advertisement detection results on 547 blocks

R	*P*
91.30	82.75

blocks is 3,458. 547 blocks are candidate for classification (i.e.: graphical blocks containing text lines); 63 % are advertisements.

Figure 22 illustrates the classification results of five sample blocks.

Once again, results are given in terms of precision and recall. The results reported in Table 3 are given with a repeated random sub-sampling validation (100 times). The learning database contains 50 % of the blocks. Considering that most of the image classification methods need a 80-20 or 90-10 split, we can already consider that our features are quite generic.

The AdaBoost weights show that overall the 22 features are useful, different feature subsets being selected for each document. We have carried out a PCA that confirms this assessment.

In [25], web images classification as advertisement reaches $R = 87.66$ % and $P = 72.33$ % with AdaBoost. The classification accuracy achieves 92.55 % in soccer videos [41]. The recall value is close to ours, yet video features are much richer than document features.

The proposed color processing system makes it possible to reach high performances on par with the closest research fields.

5 Conclusion

In this paper, we presented an efficient color segmentation system for noisy document images. The proposed system is generic, since it is applicable on any document structure. All the parameters are automatically computed using a novel stroke thickness estimation.

We introduced a new measure of pseudo-saturation to detect chromatic pixels and got rid of the saturation noise.

Within the chromatic layer, we distinguished the homogeneous areas representable by only one color from the multi-chromatic ones (photos). This segmentation is achieved using a double classification using local and global Hue histograms.

We similarly used luminance histograms to separate the B.W. and the gray layers. Such a decomposition allows high quality and local (targeted) binarization avoiding any penalizing loss of information.

Two valued applications have been proposed to validate the color processing system.

The resulting layers made the text detection easy and efficient. We improved Finereader's line segmentation's recall by ten points on women journals with complex layouts. Indeed, text tracking enables the OCR to retrieve additional lines, especially the ones embedded in multi-chromatic areas.

The acquired color and text information was also used to detect ads in press images. Such an issue is innovative as it is the first one to handle ads in complex document images. We inferred simple visual features from our segmentation and classified them with AdaBoost. In spite of the complexity of the processed data, we reached very good results compared to the ones obtained on web images and video.

In future work, we aim to combine OCR results with our advertisement detection method to generate a complete ad recognition system.

References

1. Atsalakis A, Papamarkos N, Kroupis N, Soudris D, Thanailakis A (2004) Colour quantisation technique based on image decomposition and its embedded system implementation. VISP 151(6):511–524
2. Bottou L, Haffner P, Howard PG, Simard P, Bengio Y, Lecun Y (1998) High quality document image compression with djvu. J Electron Imaging 7:410–425
3. Braquelaire J, Brun L (1997) Comparison and optimization of methods of color image quantization. IEEE Trans Image Process 6:1048–1051
4. Cattoni R, Coianiz T, Messelodi S, Modena CM, irst Via Sommarive I (1998) Geometric layout analysis techniques for document image understanding: a review. Technical report
5. Chen Q, Sun QS, Ann Heng P, Xia DS (2008) A double-threshold image binarization method based on edge detector. Pattern Recogn 41(4):1254–1267
6. Chowdhury S, Mandal S, Das A, Chanda B (2007) Segmentation of text and graphics from document images. In: ICDAR07, pp 619–623
7. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20:273–297. doi:10.1007/BF00994018
8. Drira F, Lebourgeois F, Emptoz H (2007) A coupled mean shift-anisotropic diffusion approach for document image segmentation and restoration. In: IEEE (ed.) ICDAR, pp 814–818
9. Freund Y, Schapire RE (1995) A decision-theoretic generalization of on-line learning and an application to boosting. In: Proceedings of the second European conference on computational learning theory. Springer, London, pp 23–37
10. Gatos B, Pratikakis I, Perantonis S (2006) Adaptive degraded document image binarization. Pattern Recogn 39(3):317–327
11. Gong C, Zhu F (2010) On detection of contextual advertisements. In: CAR'10: Proceedings of the 2nd international Asia conference on Informatics in control, automation and robotics. IEEE Press, Piscataway, pp 29–32
12. Haffner P, Bottou L, Lecun Y, Vincent L (2002) A general segmentation scheme for djvu document compression. In: ISMM'02, international symposium on mathematical morphology. CSIRO Publications, Sydney
13. Heckbert P (1982) Color image quantization for frame buffer display. SIGGRAPH Comput Graph 16(3):297–307
14. Hopfield JJ (1988) Neural networks and physical systems with emergent collective computational abilities. MIT Press, Cambridge, pp 457–464

15. Jain A, Yu B (1998) Automatic text location in images and video frames. In: Proceedings of Fourteenth international conference on pattern recognition, 1998, vol 2, pp 1497–1499
16. Jain AK, Yu B (1998) Document representation and its application to page decomposition. IEEE Trans Pattern Anal Mach Intell 20(3):294–308
17. Jang J, Hong K (1999) Binarization of noisy gray-scale character images by thin line modeling. Pattern Recogn 32(5):743–752
18. Jung K, Kim KI, Jain AK (2004) Text information extraction in images and video: a survey. Pattern Recogn 37(5):977–997
19. Karatzas D, Antonacopoulos A (2007) Colour text segmentation in web images based on human perception. Image Vision Comput 25(5):564–577
20. Kim HK (1996) Efficient automatic text location method and content-based indexing and structuring of video database. J Vis Commun Image Represent 7(4):336–344
21. Kim JH, Shin DK, Moon YS (2009) Color transfer in images based on separation of chromatic and achromatic colors. In: MIRAGE '09: proceedings of the 4th international conference on computer vision/computer graphics collaboration techniques. Springer, Berlin, pp 285–296
22. LeBourgeois F, Emptoz H (1999) Document analysis in gray level and typography extraction using character pattern redundancies. In: ICDAR '99. IEEE Computer Society, Washington, DC, pp 177–180
23. Leydier Y, Lebourgeois F, Emptoz H (2004) Serialized k-means for adaptative color image segmentation: application to document images and others. In: DAS2004. Lecture Notes in Computer Science. Springer, pp 252–263
24. Leydier Y, Lebourgeois F, Emptoz H (2004) Serialized unsupervised classifer for adaptative color image segmentation: application to digitized ancient manuscripts. In: ICPR 2004, pp 494–497
25. Li D, Wang B, Li Z, Yu N, Li M (2007) On detection of advertising images. In: ICME, pp 1758–1761. doi:10.1109/ICME.2007.4285011
26. Lim YK, Choi SH, Lee SW (2000) Text extraction in mpeg compressed video for content-based indexing. In: Proceedings of 15th international conference on pattern recognition, 2000, vol 4, pp 409–412. doi:10.1109/ICPR.2000.902945
27. Liu Y, Srihari S (1997) Document image binarization based on texture features. IEEE Trans Pattern Anal Mach Intell 19(5):540–544
28. Mo S, Mathews V (1998) Adaptive, quadratic preprocessing of document images for binarization. IEEE Trans Image Process 7(7):992–999
29. Moghaddamzadeh A, Bourbakis N (1997) A fuzzy region growing approach for segmentation of color images. PR 30(6):867–881
30. Nikolaou N, Papamarkos N (2009) Color reduction for complex document images. Int J Imaging Syst Technol 19(1):14–26
31. Oh H, Lim K, Chien S (2005) An improved binarization algorithm based on a water flow model for document image with inhomogeneous backgrounds. Pattern Recogn 38(12):2612–2625
32. Ouji A, Leydier Y, LeBourgeois F (2011) Chromatic / achromatic separation in noisy document images. In: IEEE International Conference on Document Analysis and Recognition, pp 167–171
33. Papamarkos N, Atsalakis AE, Strouthopoulos CP (2002) Adaptive color reduction. IEEE Systems Man Cybern Part B 32:44–56
34. Pujol A, Chen L (2007) Color quantization for image processing using self information. In: International conference on information communications and signal processing (ICICS)
35. Pujol A, Chen L (2008) Coarse adaptive color image segmentation for visual object classification. In: 15th international conference on systems, signals and image processing
36. Rowe NC, Coffman J, Degirmenci Y, Hall S, Lee S, Williams C (2002) Automatic removal of advertising from web-page display. In: JCDL'02: proceedings of the 2nd ACM/IEEE-CS joint conference on digital libraries. ACM, New York, pp 406–406. doi:http://doi.acm.org/10.1145/544220.544354
37. Scheunders P (1997) A comparison of clustering algorithms applied to color image quantization. Pattern Recogn Lett 18(11–13):1379–1384
38. Tominaga S (1988) A color classification algorithm for color images. In: Pattern recognition in practice, vol 301
39. Trier OD, Taxt T (1995) Evaluation of binarization methods for document images. IEEE Trans Pattern Anal Mach Intell 17:312–315
40. Wang J, Duan L, Liu Q, Lu H, Jin JS (2007) Robust commercial retrieval in video streams. In: ICME
41. Watve A, Sural S (2008) Soccer video processing for the detection of advertisement billboards. Pattern Recogn Lett 29(7):994–1006. doi:10.1016/j.patrec.2008.01.022
42. Weeks A, Hague G (1997) Color segmentation in the hsi color space using the k-means algorithm. SPIE 3026:143–154
43. Wolf C, Jolion JM (2006) Object count/area graphs for the evaluation of object detection and segmentation algorithms. Int J Document Anal Recogn 8(4):280–296
44. Wolf C, Jolion JM, Chassaing F (2002) Text localization, enhancement and binarization in multimedia documents. In: Proceedings of the international conference on pattern recognition, vol 2, pp 1037–1040
45. Wu X (1992) Color quantization by dynamic programming and principal analysis. ACM Trans Graph 11(4):348–372
46. Yang J, Zhu SJ (2009) A multi-scale algorithm for graffito advertisement detection from images of real estate. In: AICI'09: proceedings of the international conference on artificial intelligence and computational intelligence. Springer, Berlin, pp 444–452
47. Zhang L, Zhu Z, Zhao Y (2007) Robust commercial detection system. In: IEEE international conference on multimedia and expo, 2007, pp 587–590. doi:10.1109/ICME.2007.4284718
48. Zhong Y, Zhang H, Jain A (2000) Automatic caption localization in compressed video. IEEE Trans Pattern Anal Mach Intell 22(4):385–392. doi:10.1109/34.845381