AColDSS: Robust Unsupervised Automatic Color Segmentation System for Noisy Heterogeneous Document Images

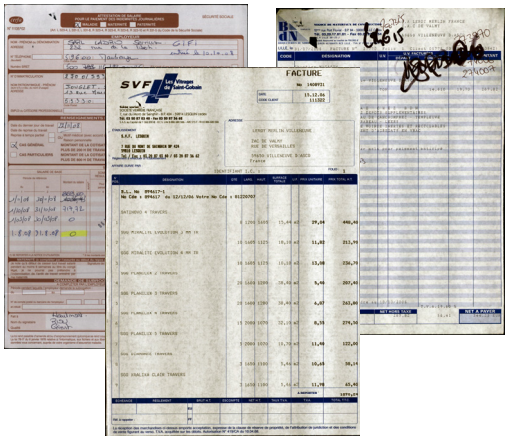
Louisa Kessi[1,2], Frank Lebourgeois[1,2] and Christophe Garcia[1,2]
[1]Université de Lyon, CNRS, France
[2]INSA-Lyon, LIRIS, UMR5205, F-69621, France
{louisa.kessi, franck.lebourgeois, christophe.garcia}@liis.cnrs.fr

Abstract. We present the first fully automatic color analysis system suited for noisy heterogeneous documents. We developed a robust color segmentation system adapted for business documents and old handwritten document with significant color complexity and dithered background. We have developed the first fully data-driven pixel-based approach that does not need a priori information, training or manual assistance. The system achieves several operations to segment automatically color images, separate text from noise and graphics and provides color information about text color. The contribution of our work is four-fold: Firstly, it does not require any connected component analysis and simplifies the extraction of the layout and the recognition step undertaken by the OCR. Secondly, it is the usage of color morphology to simultaneously segment both text and inverted text using conditional color dilation and erosion even in cases where there are overlaps between the two. Thirdly, our system removes efficiently noise and speckles from dithered background and automatically suppresses graphical elements using geodesic measurements. Fourthly, we develop a method to splits overlapped characters and separates characters from graphics if they have different colors. The proposed Automatic Color Document Processing System has archived 99 % of correctly segmented document and has the potential to be adapted into different document images. The system outperformed the classical approach that uses binarization of the grayscale image.

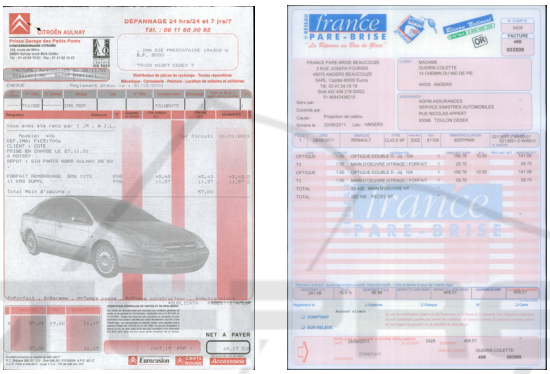
1 Introduction

Color document processing is an active research area with significant applications. In recent years, there has been an increasing need for systems which are able to convert pre-printed color documents into digital format automatically. Most of the time, the color image is converted into a grayscale image. However, the performance decreases when the segmentation fails. Nowadays, digitization systems can have to cope with dithering documents, complex color background and linear color variations, which amounts to not knowing if text is darker or lighter compared to the background, highlighting regions, corrective red overload on black text and not uniform color text/graphics overlapping. Indeed, some dithered documents may not lead to a correct automatic analysis. Smoothing most often permits to reduce dithering significantly but can also seriously damage the text. Therefore, the color information is significant. Then, a color-based segmentation could improve the process.

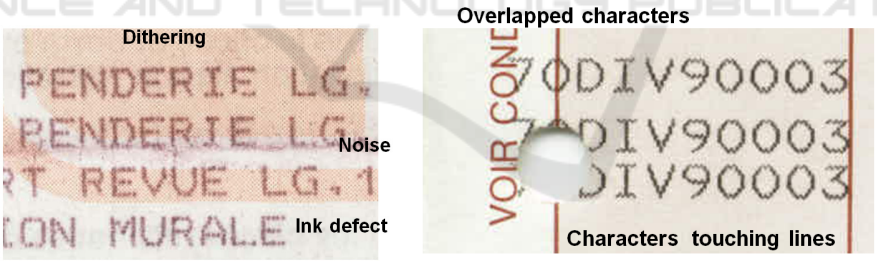
Lebourgeois F., Garcia C. and Kessi L.
AColDSS: Robust Unsupervised Automatic Color Segmentation System for Noisy Heterogeneous Document Images.
DOI: 10.5220/0006164200500063
In *European Project Space on Computer Vision, Graphics, Optics and Photonics (EPS Berlin 2015)*, pages 50-63
ISBN: 978-989-758-156-4



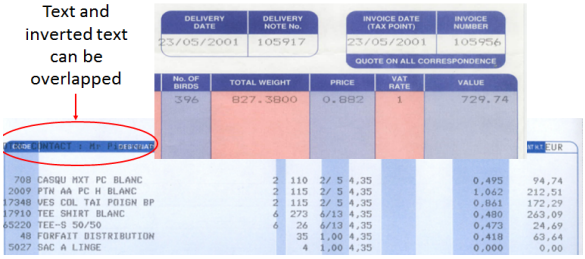
Text touching graphics



With complex background and color progressive change



Text and inverted text are commonly used



Text and inverted text are commonly used

Clean to noisy image color image

As far as the authors know, there are only few works about the use of the color for document analysis. The only referenced work for color documents come from [1] for the DjVu compression. Most of the research on color documents focuses mostly on-pixel classification approaches to reduce the number of colors found. The work of [2] introduces a new pseudo-saturation measure to separate color layers and monochrome layer. However, this global analysis of the image cannot make the difference between text colors and background color and works only for cleaned images. The authors propose in [3] a hierarchical clustering based approach to extract dominant color masks of administrative documents. Moreover, this approach requires user interaction for setting threshold parameters order to decide what a dominant color is or not. Our previous work [4] described the first robust Unsupervised Automatic Color Document Processing pixel-based system suited for business documents. This system achieved 99.25 % of correctly segmented document and is entirely based on the morphology framework. We present the first fully unsupervised color segmentation system adapted for business documents and old handwritten document with significant color complexity and dithered background. Our objective is achieved if all color characters are correctly segmented from the background, using an automatic procedure without any information provided by the user.

2 Our Proposition

Figure 1 illustrates the main steps of the proposed system. Each step will be detailed in the next sections following the organization of the paper.

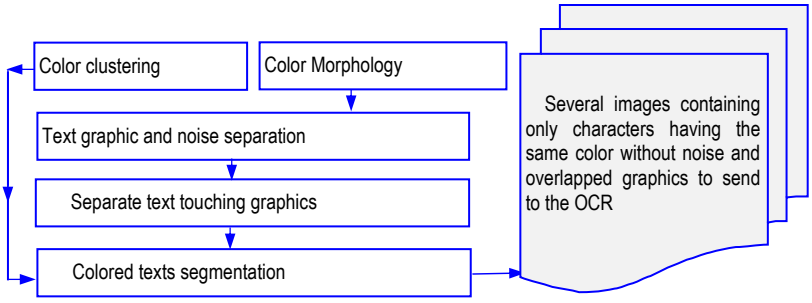


Fig. 1. Illustrates the overall scheme of the proposed.

3 Segmentation of thin Colored Objects by using Color Morphology

Our system is based on the use the color morphology which outperforms the classical grayscale morphology because characters with different colors may have the same luminance and cannot be separated in grayscale. We have decided to take benefit of the scalar color morphology with interleaved bits order proposed by Chanussot in [7] because it performs better than the color morphology using RGB vectors. He shows that it removes false and aberrant colors in complex cases. Therefore, we reduce the dissymmetry between color components by rotating the sequence of RGB (figure 2).

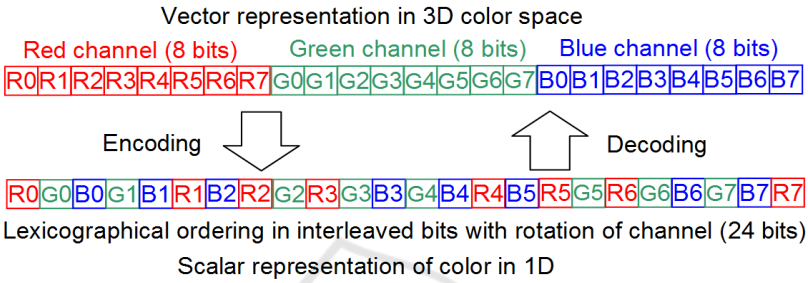


Fig. 2. Color coding in scalar by using interleaved bit and rotation of the sequence of RGB.

3.1 Thin Objects Segmentation

This step consists to generate a binary image of thin colored objects. We use the mathematical morphology because it can extract objects according to geometrical measures. In the great majority of documents, characters are darker than the background. A Black-Top-Hat (2) is efficient to extract all thin colored objects from any colored background which are darker than the background in luminance.

$$Closing(I) = Erode_B(Dilate_B(I)) \quad (1)$$

$$Black\text{-}Top\text{-}Hat(I) = Closing(I) - I \quad (2)$$

3.2 Extract All Colored Characters

To extract all colored characters even with fading ink, we apply Sauvola thresholding with variable adaptive windows which compute the best window size of the Sauvola thresholding for each pixel by using integral images [6]. This is significant in our case since we handle with dithered business documents or old deteriorated manuscripts.

3.3 Extract and Segment both Inverted and Non-Inverted Text

Unlike old manuscripts, business documents and forms show inverted printed text having text colors brighter than the background. The inverted text is important for the logical structure recognition, because it generally represents the labeling of a column

or a row in a table. To extract inverted text, we must apply a White-Top-Hat (4) which is the dual operation of the Black-Top-Hat transform.

$$Opening(I) = Dilate_B(Erode_B(I)) \quad (3)$$

$$White\text{-}Top\text{-}Hat(I) = I\text{-}Opening\ (I) \quad (4)$$

However, we cannot combine the Black-Top-Hat and the White-Top-Hat results at pixel-level. The local dominant colors bring a solution to segment simultaneously inverted and non-inverted text by morphology. This operation allows to measure the color of the background whether the text is inverted or not. To compute the dominant color, we have chosen the median filtering (5) with a large radius since its complexity has been seriously reduced and can be calculated in constant time [9] whatever the size of the window. The window size of the median must be greater than the height of the larger text. We fix a widow size of 30 to be sure to detect correctly the color background. After that, we detect and segment separately both inverted and non-inverted text by applying the conditional dilation $CD_B^M(Median(I))$ (6) which is the minimum of the dilation of the median image and the original image *I* and the conditional erosion $CE_B^M(median(I))$ (7) which is the maximum between the original image *I* and the erosion of the median image. *B* is the structural elements which have a size of the median filter.

$$DominantLocalColor(I) \approx Median(I) \quad (5)$$

$$CD_B^I(Median(I))=min(Dilate(Median(I)) , I) \quad (6)$$

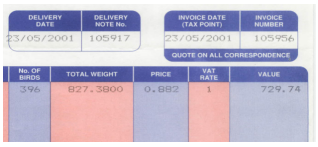
$$CE_B^I(Median(I))=max(Erode(Median(I)) , I) \quad (7)$$

$CE_B^I(Median(I))$ erases all not inverted texts which are darker in luminance than the background (Figure 3c). Accordingly, CD_B^I (Median (I)) deletes all inverted texts which are brighter than the background (figure 3d). To extract the non- inverted text, we define *Positive* (*I*) by taking the difference between CE_B^I (*Median*(*I*)) and the original image *I* (8). Accordingly, to segment inverted texts, we define *Negative* (*I*) by taking the difference between the original image *I* and CD_B^I (*Median* (*I*)) (*9*).

$$Positive(I)= CE_B^I\ (Median(I)) - I \quad (8)$$

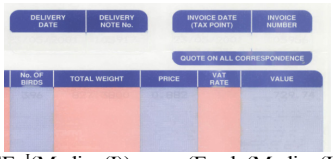
$$Negative(I)=I - CD_B^I\ (Median(I)) \quad (9)$$

The final segmentation of both normal and inverted texts and graphics is obtained by taking the maximum of the positive and negative image. Figure 3e) and 3f) show the positive and negative images represented in inverted luminance for a better appearance. The binary image is obtained by applying an adaptative thresholding with variable window size describe in [6]. The extra time of computation to process correctly the inverted texts for business documents can be avoided in the options when we process only ancient manuscripts.

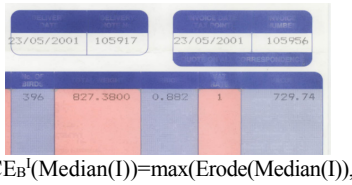


a) Original image I

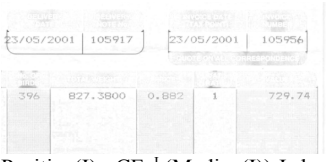
b) Median(I) filtering in constant time [9]



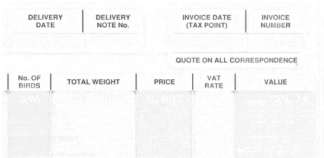
c) CE_B^I(Median(I))= max(Erode(Median(I)), I)



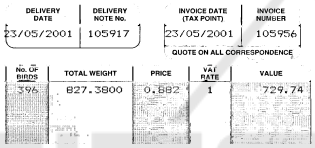
d) CE_B^I(Median(I))=max(Erode(Median(I)), I)



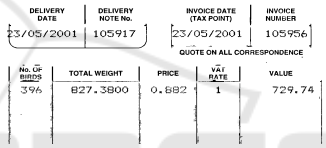
e) Positive(I)= CE_B^I (Median(I))-I shows not inverted texts



f) Negative(I)= I - CD_B^I (Median(I)) shows inverted texts



g) Max(Positive(I),Negative (I))



h) Final result after despeckle

Fig. 3. The succession of operations to binarize color images containing inverted and not inverted texts. e)..g) have been inverted.

4 Separation of Text from Graphics and Noise Removal

4.1 Despeckle

We apply a despeckle to quickly remove noise and dithered background. We apply a two pass distance transform in the 8-connectivity neighborhood of the binary image [10]. To propagate the maximal thickness value inside each object, we repeat until the morphological convolution of dilation with a null mask. The despeckle consists of erasing all objects having a maximal thickness of *ThicknessMin* threshold in one pass. Figure 3h and figure 4d shows the despeckle effects with *ThicknessMin*=2.

4.2 Separation between Graphics/Text/Dithered Parts

The extraction of several hundreds of thousands of connected components from K color images for each text and graphic color including large dithering zones is computationally too extensive. We propose a straightforward process without heuristics to remove both large graphic objects and noise without connected components extrac-

tion by using morphological geodesic operations. Our separation is based on the geodesic width and height of binary objects calculated by morphological convolution [8]. We repeat until there is no change, the morphological convolution of dilation with the Feret mask FERET90 and FERET0 for 90° and 0° direction on the binary image, respectively. All objects from the binary image having a geodesic height and width which lie in the intervals [HeightMin, HeightMax] and [WidthMin, WidthMax] respectively are shifted in the image TextsImage because they have the size to be potential characters. All the other objects are classified into the image GraphicsImage if the geodesic width or height exceeds WidthMax or HeightMax, respectively. They represent large objects that we consider as graphics (Algorithm 1). We design the system to be sure that all the characters are correctly classified into the image TextsImage. For that, we choose to set HeightMin and WidthMin to the value **3** in order to keep and thin characters like 'I' or '1'.For printing documents, we set WidthMax and HeightMax to value **64** for 300dpi images, because characters cannot exceed this size. For ligatured manuscripts we set WidthMax and HeightMax to value **512** and **128** respectively, to shift correctly handwritten words in the image TextsImage. Characters connected to the graphical elements are shifted in the GraphicsImage. The next step aims to separate characters connected to graphics.

Algorithm 1. Separation between graphics/text/dithering.

```
if ((GeodesicHeight(x, y)>0) &&
   (GeodesicWidth(x,y)>0))
  {
  if ((GeodesicHeight(x, y)<HeightMin)&&
      (GeodesicWidth(x, y) <WidthMin))
     Shift pixel(x, y) in SpecklesImage
       else
 if ((GeodesicHeight(x, y)<=HeightMax)&&
      (GeodesicWidth(x, y) <=WidthMax))
    Shift pixel(x, y) in TextsImage
       else
    Shift pixel(x,y) in GraphicsImage
 }
```

a) Original Image b) SpecklesImage c) GraphicsImage d) TextsImage

Fig. 4. Separation between noise/graphics/text, b) shows the large amount of speckles (noise and dots from dithered areas) deleted from the binary image, c) displays the graphics elements and large objects, d) provides the final binary image containing potentially all characters.

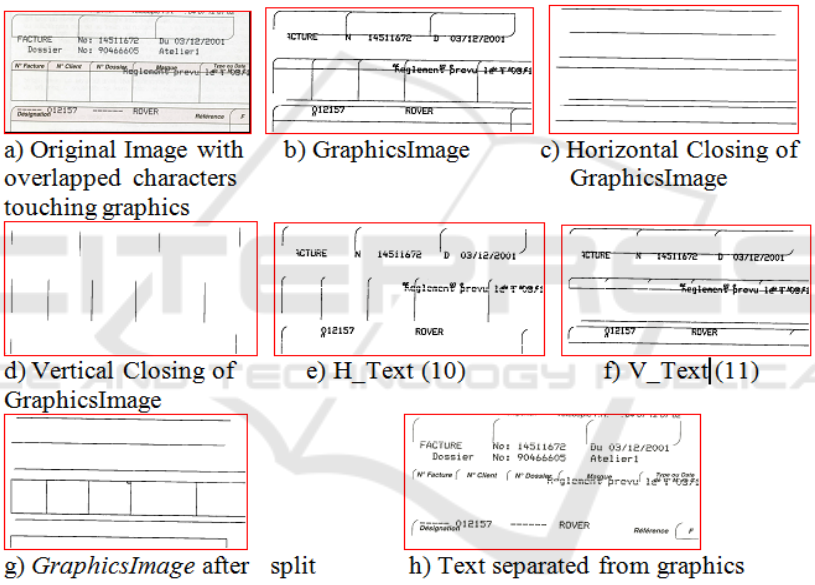
5 Separation of Characters Connected to Graphics

When colors cannot separate characters and graphics, the image GraphicsImage contains textual element which touch graphics (Figure 5b). To exceed this, we achieve a coarse separation between characters and graphics by using elementary morphological operation with the existing information provided by the system. The binary morphological closing of GraphicsImage, with an horizontal/vertical element Bh /Bv respectively, removes all characters that touch graphics both vertically / horizontally (figure 6c, 6d). We determine H_Text (10) and V_Text (11) by the difference between the horizontal/vertical closing of GraphicsImage with the image GraphicsImage itself, respectively.

X=GraphicsImage

$$H\_Text = HorizontalClosing_{Bh}(X)^c \cap X \quad (10)$$

$$V\_Text = VerticalClosing_{Bv}(X)^c \cap X \quad (11)$$



a) Original Image with overlapped characters touching graphics
b) GraphicsImage
c) Horizontal Closing of GraphicsImage
d) Vertical Closing of GraphicsImage
e) H_Text (10)
f) V_Text (11)
g) *GraphicsImage* after split
h) Text separated from graphics

Fig. 5. Separation of characters connected to graphics.

The reconstruction of characters after text/graphic separation is detailed in an additional submitted paper. Our proposed morphological operation works perfectly well for text touching straight lines only, rounded shapes remains unchanged (figure 5h). The geodesic measurement is tolerant to image skew.

6 Text Color Selection

This step consists to select automatically the right class of texts colors. We want all

characters having the same colors are segmented in the same layer. We combine the color information from the MeanShift clustering and the color segmentation by morphology to merge outlier color classes to the main color classes. We use the Fast Integral MeanShift [5] that reduces the complexity of the original MeanShift from $O(N^2)$ to $O(N)$. We have developed a straightforward and robust merging and selection process which takes into account the spatial co-occurrence of colors classes in the segmented image using the MeanShift. We use TextsImage In order to compute statistics about the connectivity of colors classes found by the MeanShift. We only focus on text color because the color of the background is useless for our application. In the image TextsImage, we compute the 2D spatial co-occurrence H2D (i,j) equal to the number of class color *i* connected spatially to class color *j* in all the inside characters of the image TextsImage. We use a 8-connectivity to count correctly in one pass H2D(i,j). We compute H1D (i) the number of occurrence of the class color from H2D(i,j). C(i,j)=H2D(i,j)/H1D(j) measures the degree of connectivity between the class color *i* with the class color *j*. Color outliers share a high connectivity with the main colors of characters (Algorithm 2).

Algorithm 2. Color Fusion.

```
for all pixel (x,y)
if (TextsImage(x,y)==0) // if character
  {
  i=ColorClass(x,y)   // from MeanShift
  if (TextsImage(x-1,y)==0) { j=ColorClass(x-1,y) H2D(i,j)++ }
  if (TextsImage(x,y-1)==0) { j=ColorClass(x,y-1) H2D(i,j)++ }
  if (TextsImage(x-1,y-1)==0) {j=ColorClass(x-1,y-1)H2D(i,j)++}
  }
for color class i
   for color class j > i
   if C(i,j)>cmin
     merge i to j if
        {
        * H1D(i)<H1D(j)
        * C(i,j) is maximal
        * H1D(j) is maximal
        * ColorDistance(i,j) minimal
        }
```

It is important to mention that in order to keep the color coherency we merge small classes to large classes but not the opposite. Algorithm 2 is repeated until there are no more merging operation. This is necessary to merge successively layers of colors around character contours. To select the right number of different text color, we set the number of text colors in the middle of the larger gap between successive color classes to limit the gap between two consecutive ranked color classes.

The time of the processing for a 300 dpi image 1640 x 2332 takes less than 3 sec. for color clustering, 6 sec. for color image segmentation by morphology and colors fusion and selection on one core without parallelization. The separation of text from graphics, image noise and dithered background can take less than a second for simple and clean image to several seconds for complex images with dithered background. All the algorithms are sequential and can be easily parallelized.

7 Results

7.1 Global Performances

As far as our knowledge, we do not find any previous research work in unsupervised document color segmentation (and not binarization) to compare with. We demonstrate the performances of our color automatic segmentation on a test image provided by the private company Janich&Klass Computertechnik GmbH which has developed the software DpuScan, which is well-known to be the best tool to separate text colors in business documents. But it is achieved manually by selecting each color background and text color. Figure 6 illustrates that our system finds text color automatically and achieve a good segmentation of this image. The thresholding of the luminance image (figure 7b) make difficult the separation between handwritten text and the preprinted forms. After a color analysis, the added handwritten text can be easily segmented.

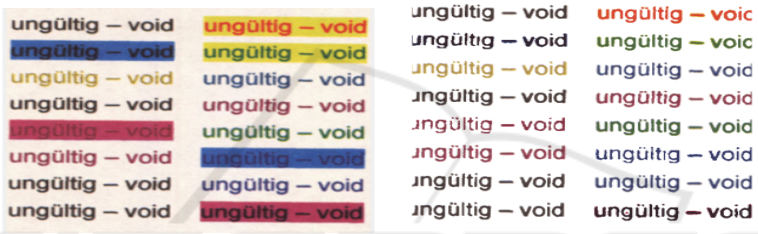


Fig. 6. Result of our system on DPUscan test image.

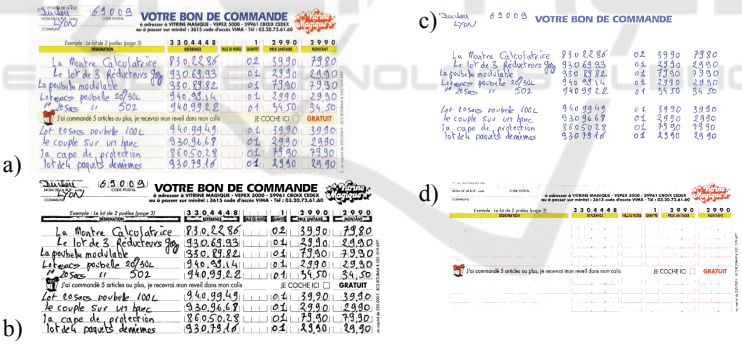


Fig. 7. Color segmentation outperforms the adaptative tresholding (Sauvola) of the lumiance for character segmentation.

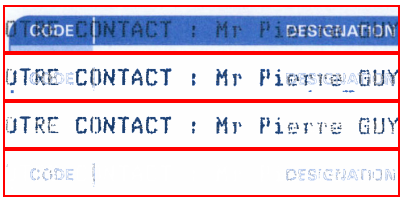


Fig. 8. Correct color separation with overlapped inverted and not inverted texts.

Figure 8 illustrates the color separation between overlapped texts in the worst case when inverted text crosses non inverted text. Our system separates correctly color handwritten text and the background even with highlighting regions (figure 9).

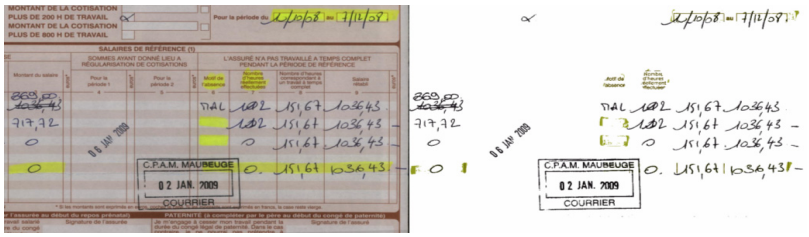


Fig. 9. Our system separate correctly color handwritten text and the background even with highlighting regions.

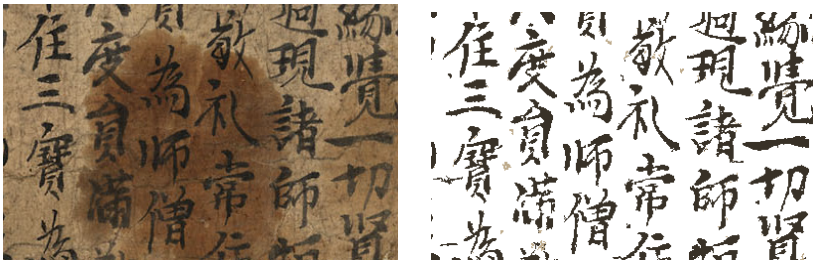
7.2 Evaluation on the Database

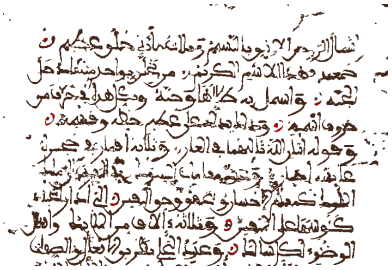
Evaluation on Business Document Database

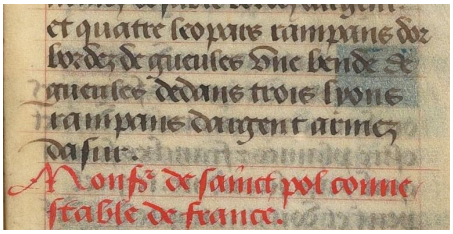
We have tested the proposed system on 929 color images of various invoices and form in real situation. We achieve a qualitative evaluation, among 929 images we found 5 images that present a change of text color, detected by the system, because of the ink bleed trough of the color background to the characters of the foreground (Figure10). Most of errors can be explained by the quality of the document itself. We have achieved 99.46 % of correctly segmented document.

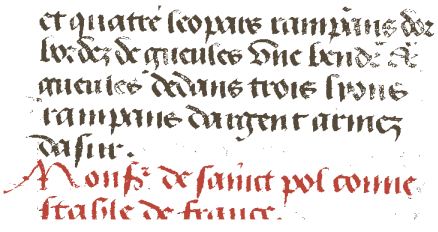
Evaluation on Ancient Manuscripts Database

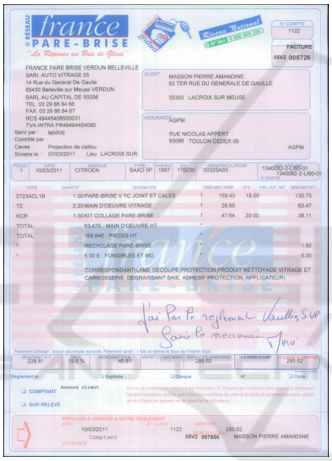
We have tested our system without the option of the inverted text detection on 250 color images of manuscripts from various digitization projects. The database contains 42 Latin manuscripts from projects with the IRHT, 20 Chinese manuscripts from Guwenshibie ANR project, 208 Arabic ancient manuscripts from Timbuktu with the digitization project Vecmas. Our system fails on 17 severely degraded manuscripts by the ink bleed-through and 8 manuscripts showing problem of ink fading. The system segment correctly 90% of the ancient manuscripts even degraded by stains in the worst cases (figure 10).

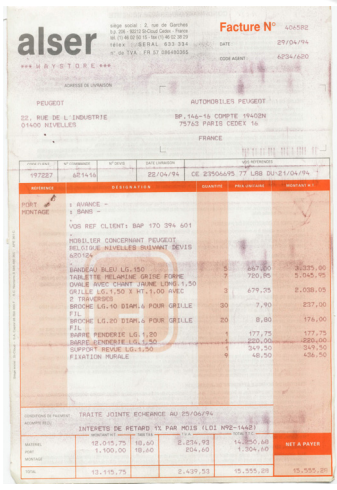


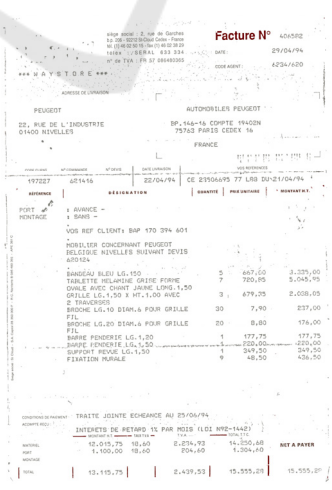












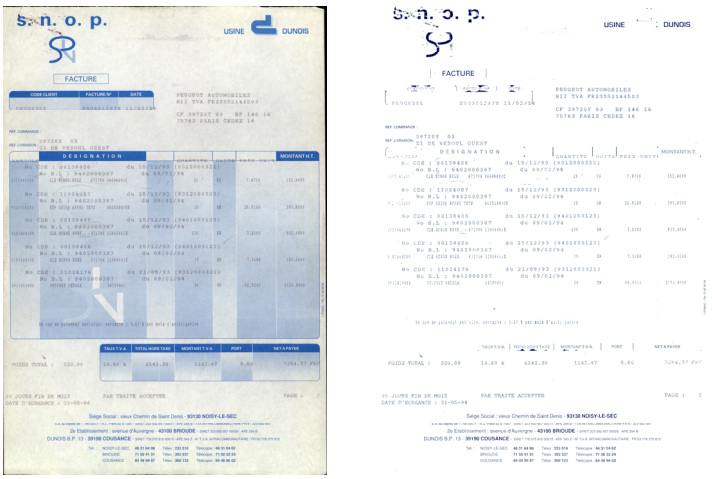


Fig. 10. Successes of our system on business and manuscripts. We display all color layers of text in the same image with an artificial white background to save place.

8 Conclusion

In this paper we have presented the first unsupervised fully automatic system adapted for color business document segmentation and old handwritten document with significant color complexity and dithered background. We have developed the first fully data-driven pixel-based approach that does not need a priori information, training or manual assistance. We have proved that color analysis outperforms binarization in the case of color document images. This is due to the fact that color analysis process provides K binary images, one for each color layer but binarization process provides a single image. Color information is lost to achieve separation between overlapped objects from different colors. Our developed system does not require heuristics and has only a very reduced number of parameters which are easy to tune by default values (average size of characters, size of windows for color median, despeckle steps). Each step of the system is independent from previous steps as the parameters are optimal for a very large range of documents. The proposed method has the following advantages: 1) It does not require any connected component analysis and simplifies the extraction of the layout and the recognition step undertaken by the OCR; 2) it processes inverted and non-inverted text automatically, using color morphology, even in cases where there are overlaps between the two.; 3) Our system removes efficiently noise and speckles from dithered background and automatically suppresses graphical elements using geodesic measurements; 4) it splits overlapped characters and separates characters from graphics if they have different colors. The proposed Automatic Color Document Processing System outperformed the classical approach that uses binarization of the grayscale image and has the potential to be adapted to various document images.

References

1. L. Bottou,P. Haffner, PG. Howard, Y. LeCun, Djvu: analyzing and compressing scanned documents for internet distribution. ICDAR.
2. A. Ouji, et al., Chromatic / achromatic separation in noisy document images, ICDAR 2011.
3. E. Carel et al., Dominant Color Segmentation of Administrative Document Images by Hierarchical Clustering , DocEng 2013.
4. L.Kessi ,et al.. "AColDPS :Robust and Unsupervised Automatic Color Document Processing System", VISAPP'15 (to appear).
5. F.LeBourgeois, et al. Fast Integral MeanShift : Application to Color Segmentation of Document Images. ICDAR 2013.
6. D. Gaceb et al. Adaptative Smart-Binarization Method for Images of Business Documents, in 12th ICDAR 2013, pp. 118-122.
7. J. Chanussot et al., "Total ordering based on space filling curves for multivalued morpholgy",ISMM'98 Amsterdam, pp 51-58.
8. S. Bres, J.M. Jolion, F. Lebourgeois, in book Traitement et analyse des images numériques, Hermes 2003, 412p.
9. S.Perreault et al, IEEE IP07, Median Filtering in Constant Time.
10. Chassery et al, Géométrie discrète en analyse d'images,Hermes 91, Paris,358p.