

Camera-based Degraded Character Segmentation into Individual Components

Céline Mancas-Thillou
Faculté Polytechnique
de Mons, Belgium
thillou@tcts.fpms.ac.be

Matei Mancas
Faculté Polytechnique
de Mons, Belgium
mancas@tcts.fpms.ac.be

Bernard Gosselin
Faculté Polytechnique
de Mons, Belgium
gosselin@fpms.ac.be

Abstract

In this article we present a novel fully automatic character segmentation for camera-based images. This is a top-down approach inspired by the human visual system: the high level step is based on color clustering while the low level one on the phase of Log-Gabor filters. First, this latter step deals with remaining touching characters after the color clustering approach and then takes into account the neighborhood information to address the problem of broken characters. Results are very encouraging and the method should be fast enough to be used on embedded platforms like personal digital assistants.

1. Introduction

With the emergence of personal digital assistants and mobile phones with embedded cameras, new challenges appear in image processing and Optical Character Recognition (OCR) to handle these natural scene camera-based images. Numerous applications such as text-to-speech for visually impaired or text translation require high recognition rates. Camera-based images induce new degradations compared to scanner-based ones such as perspective distortion, blur, shadow, uneven lighting and complex backgrounds which lead to very low recognition rates. Hence, segmentation-free applications are considered to get high recognition rates. Nevertheless, a compromise could be done between quality and computation time to enable this analysis on embedded platforms. In this paper, we shall describe a new fast character segmentation based on color clustering and a Log-Gabor filter, which aims versatility to handle very different images. For our experiments, we used a public database from the robust reading competition of ICDAR 2003 [1].



Figure 1: Images from the ICDAR 03 database.

This database includes a total of 2266 words as shown in Figure 1. The purpose of this paper is character segmentation and not text extraction; hence we removed samples like those in Figure 2 where previous processing is required for character segmentation. Our database concerns 2073 words of the whole ICDAR 2003 database.



Figure 2: Samples of removed images from the database for results.

After detailing the previous work for character segmentation, we shall describe our method step by step in Section 3 and provide intermediate and final results in Section 4. Finally, a conclusion will precede a short discussion on linguistic information in Section 5.

2. Previous work

Character segmentation literature deals more with scanner-based images than camera-based ones. A complete survey in [2] details all techniques from the analysis of the profile of a line or a word to recent methods for oriental languages. Nevertheless all these

methods are either not versatile enough for natural scene databases or computationally too expensive. A recognition-based segmentation is put forward by Kim [3], for example, by applying sliding windows on a word. In our context of images, this technique is not efficient because we have no a priori information on fonts or sizes of characters.

Algorithms used for video sequences for text retrieval and recognition could be considered as well. An image enhancement based on multiple frame integration is used to increase the resolution of characters, in order to make easier character segmentation. In our context, no video information is available and text cannot be enhanced so easily.

Interestingly, the camera-based images analysis focuses on text detection and provides text areas into a recognizer without segmentation. Hence, we shall describe a segmentation method to handle complex camera-based images with no a priori information and no additional clue with multiple frames.

3. Our proposed method

We use a top-down approach inspired from the human visual system (HVS). A first step is based on a high level approach (HL) which lets us obtain an approximation of the text area. We analyze then each text group by using image low level (LL) details. The proposed method is fully automatic and it needs no parameters in order to remain as general as possible.

The first step is the text extraction one using color information and hard clustering to highlight text information from background.

The second step aims to analyse every object (connected component) obtained after color clustering. These objects can be either isolated for the least degraded images, or touching each other. Some broken characters might appear from the HL color-based approach. To drastically reduce the number of broken and touching characters, we get the grey level variation information by using Log-Gabor filtering. The frequency of this filter is dynamically computed for each text group. A closer look on our method is given in the next subsections.

3.1. High level: color clustering

The HVS is very sensitive to colors due to the additional information absent in grey levels. Moreover, grouping areas with similar colors is one of the first steps of image comprehension as similar light properties should come from similar objects. In order not to lose information during the text extraction step,

we use a recent algorithm described in [4]. Several steps are used: a wavelet-based denoising followed by a color clustering using a K-means algorithm. Discrimination between images to separate clean and complex backgrounds enables to apply particular algorithms to handle every situation. A dynamic combination of resulted clusters is then applied or not depending on difference between colors of cluster centroids. This text extraction method has proven efficiency and gives good results with a small amount of noise, which enables to perform character segmentation into individual components in a satisfying way.

3.2. Low level: Log-Gabor filters

In order to refine results from HL approach, we need to have simultaneously spatial information to locate the characters separation in the image and frequency information to use grey level variation to detect these separations. Gabor filters are a traditional choice to address this problem. They offer the best simultaneous localisation in space and frequency and it was showed [5] that cells in the V1 visual cortex area and Gabor filters have a similar behavior. Gabor filters are cosine-like filters having a given direction and modulated by a Gaussian window. We can see in Figure 3 the shape of a Gabor filter in image space.

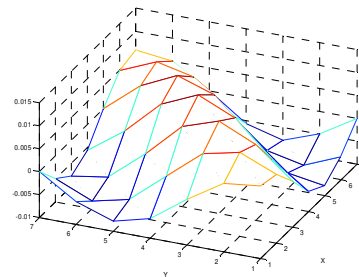


Figure 3: Gabor filter shape.

However the Gabor filter has a major drawback: large bandwidth filters induce a significant DC component. Field [6] proposed an alternative function called Log-Gabor which lets us choose a larger bandwidth without producing a DC component. He suggested that natural images are better coded by filters that have a Gaussian transfer function on a logarithmic frequency scale. Moreover, he showed that natural images spectrum statistically fall off at approximately $1/F$ which corresponds well to where the Log-Gabor filter spectrum fall off on a linear scale (Figure 4 on bottom).

From a HVS point of view, our visual cells have symmetric response on a logarithmic frequency scale (Figure 4 on top), so the Log-Gabor filters are well adapted to natural scene images. We use here the Log-Gabor filter convolution implementation done by Kovesi [7] and more precisely the convolution phase of the image by the chosen Log-Gabor filter.

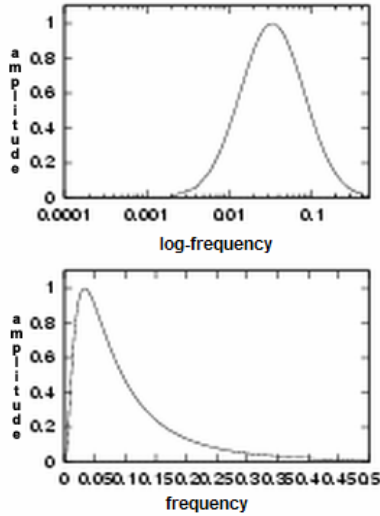


Figure 4: The Log-Gabor function spectrum in logarithmic scale frequency (top) and in linear scale frequency (bottom).

3.2.1 Touching characters segmentation

First of all, we set the filter properties to find the vertical cuts in right locations to separate touching characters. We compute vertical and horizontal filters but only use the vertical one to separate characters. Our test showed that the computation of these two directions only is the best choice.

The second very important parameter is the filter frequency. A classical way to deal with this problem is to use a “wavelet-like method”. This means to try several frequencies and to get a good result for one of those frequencies. This method is time consuming due to several convolutions with multiple frequency filters.

This “blind” approach is here replaced by considering the consistent thickness of characters in a same word. The text has a “constant” wavelength which is very different from the background wavelength. We decided to use for our filter a wavelength directly related to the mean of the characters thickness. This is computed by using the ratio between the first mask obtained by color clustering (M) and its skeleton (Figure 5). We perform then the convolution of the image with the horizontal (HF) and vertical filters (VF). Only the phase

magnitude of the convolution of the image with the vertical filter will be used as the character separation is mainly vertical.

Let us call $PhaseResult = Abs(Phase(VF))$ and $MagResult = Abs(VF)$ for the rest of this article, where Abs is the absolute value.



Figure 5: In top-down order: initial image, the mask M, M and its skeleton.

As the text and background information have different wavelengths, $PhaseResult$ contains much more information than $MagResult$ (Figure 6, two last images). The latter one mostly contains the grey-level information which is not relevant as color information has already been used by the first HL approach. In the last figure of Figure 6, $PhaseResult$ shows a local phase map which makes a good separation between the background and the textual information. Finally, we use the mask M previously computed by multiplying it to the $PhaseResult$ image.

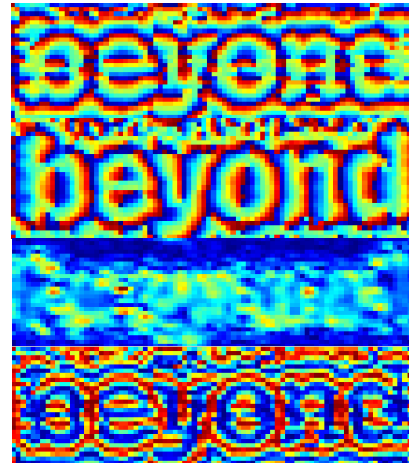


Figure 6: In top-down order: HF, VF, $MagResult$, $PhaseResult$.

As shown in Figure 7 on top, characters have mainly low intensities and higher background intensities. In

order to remain parameter-free we use now an Otsu [8] binarisation method, which automatically chooses the threshold to minimize the intra-class variance of the thresholded black and white pixels. After this binarisation, we get the final result in Figure 7 on bottom.



Figure 7: In top-down order: *PhaseResult* multiplied by the mask *M*, final result after Otsu-based thresholding.

3.2.2 Broken characters processing

Most of the broken characters are due to the initial mask *M* after the color clustering. The first LL method is applied on each object, so if this one is already broken the result will not be improved.

Another category of broken characters are due to the false alarms from the first LL approach. The crucial parameter to avoid false alarms is a good choice of the frequency (*F*) of the Log-Gabor filter. In our case, the simple approach using the mask and its skeleton work well at 97.7%. All the false alarms are due to an incorrect estimation of the frequency caused by a too coarse mask *M*. In order to solve this problem we decided to use the objects by pairs instead of using them alone. All the process is described in Figure 8. We obtain a first mask *M* after the HL approach based on color clustering. This one is composed of 7 objects or connected components: 'M', 'A', 'X', 'I', 'M' and the first part of the 'U', 'the other part of the 'U' and the final 'M'.

We apply the first LL method based on Log-Gabor filtering in the step 1. The 'M and the first part of the 'U' object is correctly separated. 'M', 'A', 'I' and the final 'M' were correctly detected as single characters so they were not separated. The 'X' object was broken: this is a false alarm. The two parts of the U were not put together because they were initially two objects: this is a HL approach error. To consider objects by pairs, we apply an iterative dilatation between them until they become only one component in step 2.

The step 3 is a second iteration of the Log-Gabor filtering. For this iteration if the two fused objects were really two characters they will be separated a second time. If they were in reality two parts of the same character as for the 'U' or 'X', they will remain fused.

Moreover, for our computation we use a new frequency (*F2*) for the filter. This one is computed on the new mask obtained after step 2 and its skeleton. As this new mask comes after a first Log-Gabor segmentation, it is more accurate than the initial mask *M*. This fact helps us to correct some of the false alarms as the 'X' which was incorrectly split during the first Log-Gabor iteration because of a wrong estimation of the frequency *F*.

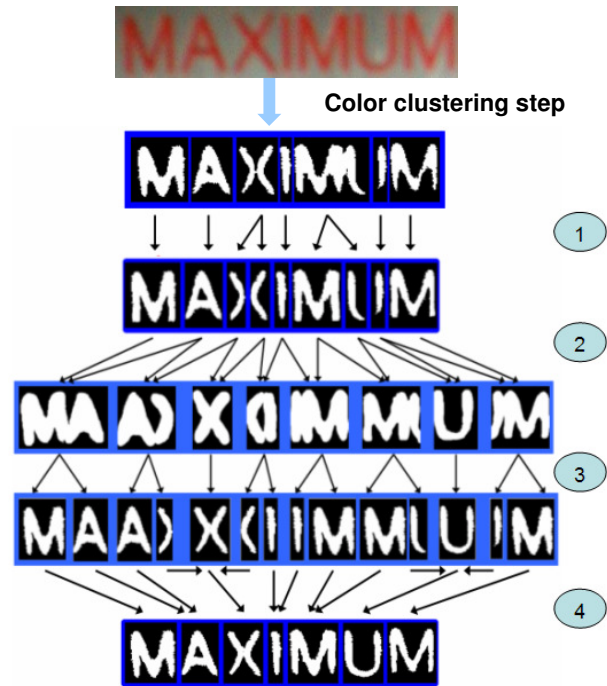


Figure 8: In top-down order: initial image, mask *M*, first Log-Gabor iteration, objects pair grouping, second Log-Gabor iteration, and final character selection.

Finally the last step 4 consists in taking the decision on the final character segmentation. If an object is segmented twice in the same way that means it contains a unique character. If this is not the case, the character was broken and we have to fuse the two objects to get the entire character. Excepting the first and last objects, 'A', 'I', and 'M' are segmented twice in both pairs of objects: there were correctly segmented and they are single characters. For 'X' and 'U', the objects are not twice the same. In this case we obtain three objects: two lateral ones from the broken characters, and a central one which contains the fused objects into a single character. We shall choose the fused object in order to eliminate the broken parts. At this step we can add a validation to have more robust results: by fusing the lateral broken objects, we should obtain the same

central object containing the fused character. If it is not the case, an error occurred at the third step, so the final result will be a word with one (or more) lacking character(s). Hence our iterative Log-Gabor method has not only the advantage to improve the segmentation results but also to validate them or not.

4. Character segmentation: some results

After the first HL approach based on color clustering [4], only 29% of the objects contain touching characters and 9% with broken characters. Results during our proposed method with intermediate steps (after the first and second pass of our LL approach) are detailed in Table 1. Our first LL approach adds 2.3% of false alarms, but then with the second one the number is drastically reduced and no false alarms for touching characters are noticed. When applying our Log-Gabor based LL method, the amount of objects containing touching characters decreased by 76% and the ones of broken characters by 52.4%.

Table 1: Percentage of objects containing touching (TC) and broken (BC) characters.

| | TC | BC |
|-----------------------------|-------|-------|
| HL approach | 29.0% | 9.0% |
| 1 st LL approach | 7.0% | 11.3% |
| 2 nd LL approach | 7.0% | 5.4% |
| Improvement | 76.0% | 52.4% |

Figure 9 shows more results to appreciate the refinement obtained by our additional LL approach.

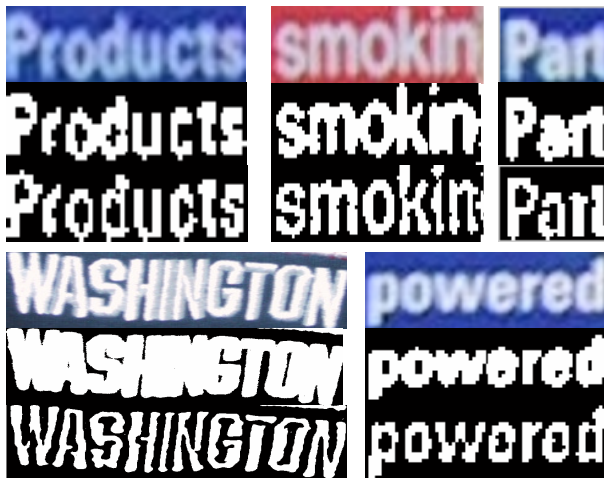


Figure 9: Initial image (top), after our HL approach (middle) and our HL+LL segmentation method (bottom).

5. Discussion and conclusion

We presented in this article a novel method for degraded camera-based character segmentation, based on a HVS-like top-down approach.

In a first step we use a high level color clustering technique to get a good approximation of the textual information. From this first approximation we refine the results using a two-pass low level method based on Log-Gabor filters phase which drastically reduces the number of touching and broken characters.

Our technique is fully automatic and no parameter is needed in order to be as versatile as possible.

Moreover, the LL approach has low computational cost and could be even implemented on embedded platforms such as personal digital assistants because Log-Gabor filtering is roughly as computationally expensive as any other filtering as we compute only two directions and one frequency.

To improve recognition rates after this system, we could use linguistic information as a few errors left and this following step is currently under progress. This constitutes our future work to build a whole efficient and dynamic character recognition-based segmentation by using all available information.

6. Acknowledgements

This work is part of the project Sypole and is funded by Ministère de la Région wallonne in Belgium.

7. References

- [1] Robust Reading Competition, retrieved May 31st, 2005 from <http://algoval.essex.ac.uk/icdar/RobustWord.html>.
- [2] R. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation", *IEEE Trans. On Pattern Analysis and Mach. Intel.*, vol.18, 1996, pp.690-706.
- [3] H. Kim, "Segmentation-Free Printed Character Recognition by Relaxed Nearest Neighbor Learning of Windowed Operator", *Proc. Sibgrapi - Brazilian Symp. on Comp. Graph. and Image Proc.*, 1999, pp. 195-204.
- [4] C. Thillou and B. Gosselin, "Color Binarization for complex camera-based images", *Proc. of the Electronic Imaging Conference of SPIE/IS&T*, 2005, pp. 301-308.
- [5] D.H. Hubel, "Eye, Brain, and Vision", *Scientific American Library*, n°22, 1988.
- [6] D.J. Field, "Relations between the statistics of natural images and the response properties of cortical cells", *Jour. of the Optical Society of America*, 1987, pp. 2379-2394.
- [7] P. Kovess, "Image Features From Phase Congruency", *Videre: A Journal of Computer Vision Research*, vol.1, 1999.
- [8] N. Otsu, "A thresholding selection method from gray-level histogram", *IEEE Trans. on Systems, Man, and Cybernetics*, 1979, pp. 62-66.