Character Segmentation-by-Recognition Using Log-Gabor Filters

Céline Mancas-Thillou, Bernard Gosselin
Faculté Polytechnique de Mons, Avenue Copernic 1, 7000 Mons, Belgium
celine.thillou@tcts.fpms.ac.be

Abstract

Natural scene images coming usually from low-resolution sensors in embedded context suffer from low text recognition results. Due to several types of degradations, existing algorithms are not robust enough. In order to improve recognition, we present in this paper a character segmentation using Log-Gabor filters to take advantage simultaneously of gray-level variation and spatial location. The recognition step is used to determine dynamically some of the parameters needed for the filter. Finally, several quantified results are presented to highlight the efficiency of this method against several issues.

1. Introduction

Character segmentation has been a well-investigated field over the last decade and its main aim was to provide individual characters to optical character recognition (OCR) algorithms. More recently, robust but time consuming algorithms appear to avoid character segmentation and to realize character recognition without segmentation and even without binarization. With the huge progress done in small sensors, camera-based applications appear everywhere, with mobile phones for example. Natural scene images coming from these pictures face new challenges to image processing adding multiple degradations such as low resolution, uneven lighting, perspective, non-formatted text, complex backgrounds, artistic fonts due to advertisement and so on. Until now, much effort have been given to the steps of text detection, localization and text extraction to pre-process text as much as possible to provide good-looking characters to OCR algorithms. Due to all camera-based degradations and embedded context applications, recognition algorithms are not robust enough to handle these problems. In order to increase natural scene character recognition, effort must be brought on character segmentation into individual characters to speed up results and to ensure satisfying recognition rates.

2. State-of-the-art

As described in the introduction, natural scene character segmentation into individual components has not been much investigated so far.

Classical character segmentation algorithms, grouped into a survey in [1], cannot properly handle natural scene character segmentation into individual characters. Algorithms dealing with vertical projection or contour analysis assume well-aligned and not distorted texts and the ones dealing with sliding windows are quite time consuming. Bae et al. [2] use supervised learning to find cut places for character segmentation. All kinds of touching characters are fed into a multilayer perceptron to train the character segmenter and give right segmentation. In natural scene images, learning character segmentation is quite impossible due to the infinite number of fonts and touching characters which can be found.

Some methods have been proposed to increase natural scene recognition results without segmentation. To circumvent non extraction of isolated and touching characters of existing algorithms, Negishi et al. [3] use feature points to extract characters individually in any images. The extraction of characters one by one could be quite consuming for images with much text and this algorithm presents results with a restricted number of fonts.

Papers dealing with natural scene character segmentation can be found sparingly. Sun et al. [4] propose recognition-based segmentation using a dual eigenspace decomposition. The frequency domain is used to extract features for recognition and the initial image domain to extract features for segmentation. These two sets of features are compared to the principal components analysis features and results for Japanese character recognition are increased. Chen et al. [5] propose a gray-level consistency constraint (GCC) to segment character and increase recognition results. After a connected component analysis, the GCC step removes extra pixels from components with non-consistent gray-level values assuming that a character is composed of homogeneous gray-level values. In one of our previous works [6], we segment characters individually by using Log-Gabor filters in a static way, presenting problems with strongly degraded characters or non-consistent characters thicknesses.

In this paper, we propose a dynamic character segmentation into individual components using Log-Gabor filters. The definition and the use of these filters in the character segmentation application are explained in Section 3. We then detail our algorithm by explaining how to choose properly Log-Gabor filter parameters. Finally in Section 5, some results are given to appreciate the effectiveness of the algorithm with respect to character recognition and broken characters minimization. In the last section, we conclude this paper by presenting our future works.

3. Log-Gabor filters

3.1. Definition

In order to segment characters properly, we need to have simultaneously spatial information to locate the character separation in the image and frequency information to use gray level variation to detect these separations. Gabor filters could be a choice to address this problem.

Gabor filters have been extensively used to characterize texture and more specifically in our context to detect and localize text into an image. In this aim, Gabor filters are quite time consuming because several directions and frequencies must be used to handle the variability in character sizes and orientations. Gabor filters present a limitation in the bandwidth where only bandwidth of 1 octave maximum could be designed. Log-Gabor filters, proposed by Field [7], circumvent this limitation. They always have a null DC component and can be constructed with arbitrary bandwidth and the bandwidth can be optimized to produce a filter with minimal spatial extent.

Log-Gabor filters in frequency domain can be defined in polar coordinates by $H(f,\theta) = H_f \times H_\theta$ where H_f is the radial component and H_θ, the angular one:

$$H(f,\theta) = \exp\left\{\frac{-[ln(f/f_0)]^2}{2[ln(\sigma_f/f_0)]^2}\right\} \times \exp\left\{\frac{-(\theta-\theta_0)^2}{2\sigma_\theta^2}\right\}$$

with f_0, the central frequency, θ_0, the filter direction, σ_f, which defines the radial bandwidth B in octaves with $B = 2\sqrt{2/ln2} * |ln(\sigma_f/f_0)|$ and σ_θ, which defines the angular bandwidth $\Delta\Omega = 2\sigma_\theta\sqrt{2ln2}$.

As we are looking for vertical separation between characters, we use only two directions for the filter, the horizontal and the vertical one. Hence, for each directional filter, we got a fixed angular bandwidth of $\Delta\Omega = \Pi/2$. Log-Gabor filters are not really strict with directions and defining only two directions enables to handle italic and/or misaligned characters.

Only two parameters remain to define, f_0 and σ_f, used to compute the radial bandwidth.

3.2. Parameter influence on character segmentation context

The central frequency f_0 is used to handle gray level variations to detect separation between characters. It is quite logical to get a central frequency close to the inverse of the thickness of characters to get those variations. Nevertheless, the thickness of characters can not be very accurate depending on the presence of degradations. In order to handle all kinds of degradations, we compensate inaccurate thickness estimation with the second parameter σ_f. If the thickness of characters is not consistent inside a character such as in Figure 1, some character parts can be removed permanently. In this case, by increasing the bandwidth, we can support the variability in the thickness of characters. Moreover, sometimes with very degraded characters or very close characters, the thickness is very difficult to estimate and the filter must be very sharp to get each small variation in the gray level values such as in Figure 2; in this case, the bandwidth is chosen to be quite narrow.

Figure 1. From left to right: original image, segmentation with misestimated thickness, segmentation with the same thickness corrected by a larger bandwidth

Figure 2. Original image (top left), binary version (top right), segmentation with large bandwidth (bottom left), segmentation with narrow bandwidth (bottom right)

As degradations and conditions of frequency estimation are quite unexpected, we chose the bandwidth filter in a dynamic way using recognition results. In the following paragraph, we detail our method and how each parameter is estimated.

4. Proposed algorithm

Text detection step is already done and an approximate binarization of the detected area is also available using a color clustering algorithm [8] with different clustering distances to handle uneven lighting and complex backgrounds. The character segmentation is processed on gray-level images.

4.1. Frequency estimation

Text embedded in natural scene images presents a quite consistent wavelength, which is very different from the background. We decided to use for our filter a wavelength inversely related to the average of the character thicknesses. This is computed by using the ratio between the number of pixels of the first mask obtained by color clustering and its skeleton as shown in Figure 3. Hence, the central frequency f_0 can be estimated approximately by:

$$f_0 = \sum_{i,j} Skeleton(i,j) / \sum_{i,j} Mask(i,j)$$

Figure 3. Top: original image, bottom: Mask and Skeleton images

4.2. Bandwidth estimation-by-recognition

Due to the large variation in character fonts and sizes as explained in Section 3.2, the bandwidth has to be chosen dynamically. As objects to be segmented are text, we can use segmentation-by-recognition to choose the convenient bandwidth. We fixed the initial and final values for the bandwidth estimation: from around 2 octaves to around 0.3 octaves, which makes σ_f / f_0 vary with a step of 0.1, we process six filters and provide the result to an OCR engine. The result is composed of the phase of the vertical filter only because the phase result of the filter shows a local map which makes a good separation between the background and the textual information; this intermediate result is then multiplied by the first mask from the color clustering to remove possible noise around characters as explained in [6]. To choose the bandwidth for filters, we use a home-made OCR algorithm composed of a multi-layer perceptron with geometrical features to recognize characters, which is already trained and is used to assess how well characters are segmented. Figure 4 shows two examples with varying bandwidths and results from recognition, which enables to take the right decision for the bandwidth estimation. Recognition rates for each character or assumed character are averaged and the maximum score enables to estimate the bandwidth. The first example is an image with little contrast between characters and background and the second one presents a misaligned and slanted text. This estimation needs six straightforward filters with only one frequency which enables to use Log-Gabor filters for character segmentation in an embedded context. A priori rules on context could be used to decrease even more the computation time.

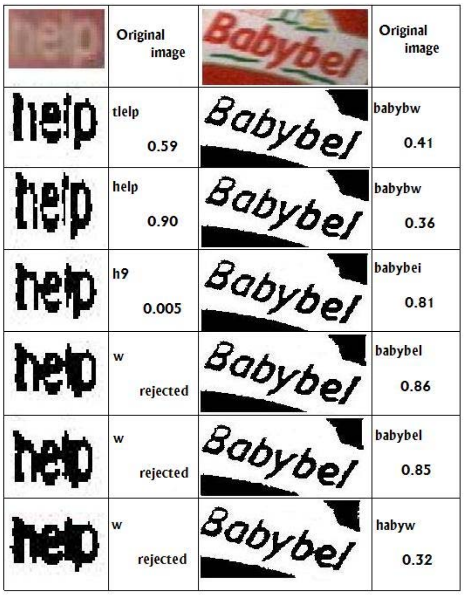

Figure 4. 1st and 3rd columns: character segmentation with bandwidth varying from 0.3 to 2 octaves, 2nd and 4th columns: OCR results with average recognition rate

5. Some results

In this section, we present some results under various forms. Table 1 shows the increase of the recognition rate for our tested database. We use the ICDAR2003 public sample database, which includes 171 already detected text areas, with some examples even not recognizable by humans. Comparisons are done between the behaviour of an efficient commercial OCR against initial images, after a color clustering to get binary images, where efficiency results have been written in [8] and after the segmentation step to show the efficiency and necessity of this method to improve recognition results. Recognition rates are computed using the Levenshtein distance [9] between the ground truth and the resulting text.

In Table 2, we compare the number of broken characters and touching characters after character segmentation. Actually, if character segmentation is not really appropriate, the text will be either under-segmented or over-segmented. The aim is to reduce the number of touching characters without increasing the number of broken characters. Compared with results with a fixed bandwidth of 1.7, which is the best value for our database, the number of touching characters decreases and the number of broken characters falls down also with this dynamic character segmentation.

Table 1. Usefulness of character segmentation in natural scene images

	color	binary	binary with seg.
FineReader8	29.1 %	60.4 %	80.5 %

Table 2. Impact of segmentation-by-recognition

	Touching characters	Broken characters
fixed bandwidth	7.8 %	11.8 %
seg.-by- reco.	4.9 %	6.4 %
improvement	62.8 %	54.2 %

Finally, in this proposed character segmentation, the bandwidth is estimated with the recognition step and we compute the efficiency rate of this decision. Some erroneous choices could be taken due to our majority vote on the whole text and the decision is rightly taken in 98.8% of images. Errors are mostly avoided with this character segmentation-by-recognition as each decision is checked with other steps dynamically. Main errors are due to the OCR engine with very degraded characters.

6. Conclusion and future work

We proposed a new character segmentation for natural scene camera-based text. This dynamic segmentation uses the recognition step to give more satisfactory results and to estimate parameters of Log-Gabor filters. These filters are very appropriate for character segmentation by using as central frequency the inverse of an estimation of character thickness and as bandwidth a dynamic choice to handle all unexpected variations in natural scene characters. Slanted or misaligned characters are largely supported. By using only one direction and one frequency for the filters, the process is computationally efficient. Recognition results are greatly increased and the number of touching characters decreases without increasing the one of broken characters.

Natural scene images have numerous and various degradations and to handle these issues, it is very important to link consecutive steps together to take dynamic decisions. As future works, we are currently working on an extraction-by-segmentation-by-recognition algorithm to get better results.

References

[1] R.G.Casey, E.Lecolinet. *A survey of methods and strategies in character segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(7):690-706, 1996.

[2] J.H.Bae, K.C.Jung, J.W.Kim, H.J.Kim. *Segmentation of touching characters using an MLP*, Pattern Recognition Letters, 19(8):701-709, 1998.

[3] K.Negishi, M.Iwamura, S.Omachi, H.Aso. *Isolated Character Recognition by Searching Features in Scene Images*, First International Workshop on Camera-Based Document Analysis and Recognition, pp. 140-147, 2005.

[4] J.Sun, Y.Hotta, K.Fujimoto, Y.Katsuyama, S.Naoi. *Grayscale Feature Combination in Recognition based Segmentation for Degraded Text String Recognition*, First International Workshop on Camera-Based Document Analysis and Recognition, pp. 39-44, 2005.

[5] D.Chen, J-M.Odobez, H.Bourlard. *Text detection and recognition in images and video frames*, Pattern Recognition, 37:594-608, 2004.

[6] C.Mancas-Thillou, M.Mancas, B.Gosselin. *Camera-based Degraded Character Segmentation into Individual Components*, Proc. of International Conference on Document Analysis and Recognition, pp. 755-760, 2005.

[7] D.J.Field. *Relations between the statistics of natural images and the response properties of cortical cells*, Jour. of the Optical Society of America, pp. 2379-2394, 1987.

[8] C.Mancas-Thillou, B.Gosselin. *Color Text Extraction from Camera-based Images-the Impact of the Choice of the Clustering Distance-*, Proc. of International Conference on Document Analysis and Recognition, pp. 312-316, 2005.

[9] V.I.Levenshtein. *Binary codes capable of correcting deletions, insertions and reversals*, Sov. Phys. Dokl., 6:707-710, 1966.