IDIAP RESEARCH REPORT

Text Segmentation and Recognition in Complex Background Based on Markov Random Field [1]

Datong Chen, Jean-Marc Odobez and Hervé Bourlard
IDIAP, Switzerland
{chen, odobez, bourlard}@idiap.ch

IDIAP–RR 02-17

Apr. 2002

Institut Dalle Molle d'Intelligence Artificielle Perceptive • CP 592 • Martigny • Valais • Suisse

téléphone +41−27−721 77 11
télécopieur +41−27−721 77 12
adr.él. secretariat@idiap.ch
internet http://www.idiap.ch

Rapport de recherche de l'IDIAP 02-17

Text Segmentation and Recognition in Complex Background Based on Markov Random Field[1]

Datong Chen, Jean-Marc Odobez and Hervé Bourlard
IDIAP, Switzerland
{chen, odobez, bourlard}@idiap.ch

Apr. 2002

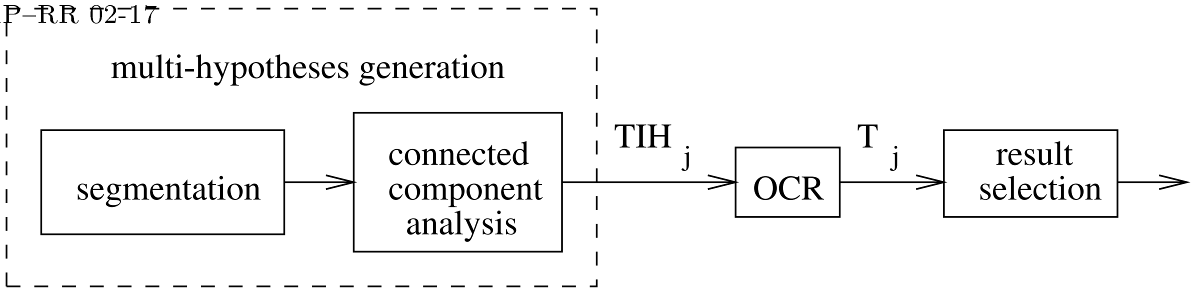


FIG. 1 –. Text recognition scheme

Abstract

In this paper we propose a method to segment and recognize text embedded in video and images. We modelize the gray level distribution in the text images as mixture of gaussians, and then assign each pixel to one of the gaussian layer. The assignment is based on prior of the contextual information, which is modeled by a Markov random field (MRF) with online estimated coefficients. Each layer is then processed through a connected component analysis module and forwarded to the OCR system as one segmentation hypothesis. By varying the number of gaussians, multiple hypotheses are provided to an OCR system and the final result is selected from the set of outputs, leading to an improvement of the system's performances.

1 Introduction

Text recognition in video and images aims at integrating text-based search and advanced OCR technologies. It is now recognized as a key component in the development of advanced video and image annotation and retrieval systems. Text characters contained in video are of low resolution, of any grayscale value (not always white) and embedded in complex background even when the whole text string is well located. Thus, applying conventional OCR technology directly leads to poor recognition rate. Efficient segmentation of text characters from background is therefore necessary to fill the gap between video documents and the input of a standard OCR system.

Previous work on text segmentation from complex background have been published in recent years. Lienhart [?] and Bunke [?] clustered text pixels from images using a common image segmentation or color clustering algorithm. Although these methods can somehow avoid the text location work, they are very sensitive to noise and character size. Most top down text segmentation methods are performed after text string is located in images. These methods assume that the grayscale distribution is bimodal and that characters correspond a priori to either the white part or the black one, but without providing a way of choosing the right one on-line. Great efforts are thus devoted to perform better binarization, combining global and local thresholding [?], or M-estimation [?], or simple smoothing [?]. However, these methods are unable to filter out background regions with similar grayscale value to characters. Text enhancement methods, if the character grayscale value is known, can help the binarization process [?]. However, without estimation of scales, the designed filters can not enhance character stroke with varying width [?]. In video, multi-frame enhancement [?] also can reduce the number of background regions, but only when text and background have different movements.

In this paper, we present a method based on MRF and multi-hypotheses generation to improve both the segmentation and recognition of embedded text with unknown grayscale value. In the next section we describe our method, then present commented results and we finish with some concluding remarks.

2 Description of the method

We located text using former work presented in [?]. In this method, text-like textures are first detected by integrating horizontal and vertical edges, and further segmented into single line text candidates using baseline location. A support vector machine is then used to identify text regions from the candidates.

This text location step provides us with text images as those presented in figure 2. As we mentioned before, OCR software can not be applied directly. Indeed, experience shows that OCR performances are quite unstable, as already mentioned by others [?], and significantly rely on the segmentation quality, in the sense that errors made in the segmentation are directly forwarded to the OCR. In our case, we propose a softer scheme (see figure 1) in

FIG. 2 –. Examples of located text in video

which multiple text layer candidates are provided to the OCR, delaying the hard decision, if any, after the OCR step. Our algorithm works as follows: first, text image hypotheses TIH_j are generated, relying on a segmentation step followed by a connected component analysis; then hypotheses are processed by the OCR and the result is selected from the output strings $(T_j)_j$.

2.1 Segmentation methods

Let S denote the set of sites s (pixels), and o the observation field $o = \{o_s, s \in S\}$, where o_s corresponds to the grayscale value at site s. We model the image intensities in terms of the combination of K simple random processes, also referred to as layers. Each simple process is expected to represent regions of the image having similar gray levels, one of them being text. Thus, the segmentation consists in the mapping of pixels to processes. It is stated as a statistical labeling problem, where the goal is to find the label field $e = \{e_s, 1 \le e_s \le K, s \in S\}$ that best accounts for the observations, according to a given criterion.

To perform the segmentation, we tested 2 algorithms EM and GEM. In the 2 cases, the probability that a gray value arise at a given site within a particular layer i is modeled by a gaussian, i.e. $\mathrm{p}_i(o_s) = \mathcal{N}(m_i, \sigma_i)$.

2.1.1 The basic EM algorithm

Here, individual processes are combined into a probabilistic mixture model according to :

$$\mathrm{p}(o_s) = \sum_{k=1}^{K} \mathrm{p}(o_s | e_s = k)\mathrm{p}(e_s = k) = \sum_{k=1}^{K} \pi_k \mathrm{p}_k(o_s) \tag{1}$$

Given an image, we use an EM algorithm to find the set of parameters $\varphi = (\mu_i, \sigma_i, \pi_i)$ which provides a maximum likelihood fit to the data set o, i.e. which maximizes $\mathrm{L}^{\varphi}(o) = \sum_{s \in S} \ln \mathrm{p}(o_s)$ based on the complete data (o, e) with respect to the unknown data (e) [?]. After maximization, labels are assigned to the individual layer according to :

$$\forall s \in S \qquad e_s = \underset{i}{\operatorname{argmax}}\ \mathrm{p}_i(o_s) \tag{2}$$

2.1.2 The Gibbsian EM algorithm (GEM)

While the EM algorithm is able to capture most of the gray level distribution properties, it does not model the spatial correlation between assignment of pixels to layers, resulting in noisy label images. To overcome this, we introduce some prior by modeling the label field as a MRF. Then, instead of using the simple rule (2), we perform a Maximum a-posteriori (MAP) optimization. Due to the equivalence between MRF and Gibbs distribution ($\mathrm{p}(e) = \frac{1}{Z(V)} e^{-U_1^V(e)}$) [?], this is equivalent to the minimization[1] of an energy function $U(e,o) = U_1^V(e) + U_2^{\varphi}(e,o)$ with $U_2^{\varphi}(e,o) = \sum_{s \in S}(-\ln \mathrm{p}_{e_s}(o_s))$ expressing the adequacy between observations and labels, as in the EM algorithm, and U_1 is equal to :

$$\sum_{s \in S} V_{11}(e_s) + \sum_{<s,t> \in \mathcal{C}_{hv}} V_{12}^{hv}(e_s, e_t) + \sum_{<s,t> \in \mathcal{C}_{diag}} V_{12}^{d}(e_s, e_t) \tag{3}$$

where $\mathcal{C}_{hv}$(resp. $\mathcal{C}_{diag}$) denotes the set of two neighbor pixels in the horizontal, vertical (resp. diagonal) direction. The V are the (local) interaction potentials which expresses the prior on the labels.

One may wish to learn these potentials off-line from examples. However, this would require to know the correspondence between learned labels/layers and current ones.[2] Moreover, the optimum values are very dependent on

1. with respect to e
2. Remind that text may be black or white or gray.

the character scale. Thus, we propose the following algorithm to estimate all the parameters $\Theta{=}(\mu_i{,}\sigma_i{,}V)$ using an EM procedure. Recall that the E step involves the computation of :

$$\mathbb{E}\left[\ln \mathrm{p}^{\Theta}_{eo}|o{,}\Theta^n\right] = \sum_e \ln\left(\mathrm{p}^{\Theta}_{o|e}(e{,}o)\mathrm{p}_e(e)\right)\ \mathrm{p}^{\Theta^n}_{e|o}(e{,}o) \tag{4}$$

which is then maximized over Θ. Two problems arise here. Firstly, this expectation on $\mathrm{p}^{\Theta^n}_{e|o}$ can not be computed explicitly nor directly. Instead, this law will be sampled using Monte Carlo methods with a Gibbs sampler, and the expectation will be approximated along the obtained Markov chain. Secondly, the joint log-likelihood probability p^{Θ}_{eo} is not completely known, because of the presence of the uncomputable normalizing constant $Z(V)$ in the expression of $\mathrm{p}(e)$. To avoid this difficulty, we replace $\mathrm{p}(e)$ by its pseudo-likelihood $\mathrm{p_S}(e)$ [?] defined from the conditional probabilities :

$$\mathrm{p_S}^V(e) \doteq \prod_{s\in S} \mathrm{p}(e_s|e_{\mathcal{G}_s}) \tag{5}$$

where $e_{\mathcal{G}s}$ represents the label in neighborhood of s. Using this new criterion, the maximization of the expectation (4) can be executed, providing new estimates of $(\mu_i{,}\sigma_i)$ and of V. The reader is referred to [?] for more details on the procedure that we have adapted to our need. Complexity of the GEM algorithm is around 3 times of EM complexity, with non optimized code.

2.2 Connected component analysis

After segmentation, for each label, a binary text image hypothesis is generated by assuming that this label corresponds to text and all other labels corresponds to background. In each hypothesis, the text regions are refined by using a simple connected component analysis. We keep only connected components that satisfy the following constraints: size is bigger than 5 pixels; width/height ratio is between 4.5 and 0.1; the width of the connected component is less than 2.1 of the height of the whole text region.

2.3 Text recognition

The basic recognition of a text string from a single binary text image is performed using an OCR software. Indeed, in our method, we provide multiple text image hypotheses to the OCR and select the final result from the set of output strings based on a string confidence evaluation.

2.3.1 The initial hypotheses generation

In the EM and GEM segmentation methods, if we have K different labels, we get K hypotheses. The right choice of K is another important and difficult issue. One general way to address the problem consists in checking whether the increase in model complexity really provides a better fit of the data. This can be done for instance by using the minimizing description length criterion. However, this information theoretic approach may not be appropriate for qualifying a good text segmentation. Therefore, we use a more conservative approach, by varying K from 2 to 4, generating in this way nine text image hypotheses TIH_j.

2.3.2 Result selection and confidence value

Each text image candidate TIH_j is processed by the OCR software [3] thus providing a string T_j. The final string result is the string T_j that provides the largest confidence value CV which is defined as :

$$CV(T) = \sum_{i=0}^{l_T-1} f(T[i]) + \sum_{i=1}^{l_T-1} g(T[i-1]{,}T[i])\,.$$

where, l_T is the length of string T, $g(x{,}y)$ is a simple bi-gram language model defined as :

$$g(x{,}y) = \begin{cases} -4 & if\, x = lower\, case, y = upper\, case \\ 0 & \text{otherwise} \end{cases}$$

3. we use an open OCR toolkit (OpenRTK) from Expervision

methods	Ext.	CRR	Prec.	WRR
Otsu	**9009**	**88.4%**	**93.8%**	**89.3%**
EM.2	7589	70.6%	88.9%	59.4%
GEM.2	**9081**	**88.7%**	**93.4%**	**90.8%**
EM.3	9071	88.7%	93.5%	84.8%
GEM.3	9249	**89.9%**	92.9%	84.9%
EM.4	9055	87.5%	92.4%	83.0%
GEM.4	9128	**88.8%**	93.0%	85.2%
EM.4.3.2	9481	90.4%	91.2%	88.1%
GEM.4.3.2	**9432**	**92.5%**	**93.8%**	**91.7%**

TAB. 1 –. Recognition results in extracted characters (Ext.), character recognition rate (CRR), precision (Prec.) and word recognition rate (WRR).

$$\text{and } f(x) = \begin{cases} 4 & if\, x = upper\, case \\ 2 & if\, x = lower\, case \\ 0 & if\, x = i, I, l \\ -1 & \text{otherwise} \end{cases}$$

Some characters (i, I, l) and lower case characters are given lower weights because background noise is more often recognized as these characters.

3 Experiments

The whole scheme was tested on text regions located and extracted from one hour of sports video provided by the BBC, using the algorithm presented in [**?**]. It correctly located 98.7% text regions while providing 0.38% false alarms. We randomly selected 1208 images from the extracted text regions, mainly by time sub-sampling, providing a database of 9562 characters or 867 words. Figure 2 shows some examples. Text characters are embedded in complex background with JPEG compression noise, and the grayscale value of characters is not always the highest.

We first report results where K, the number of mixture is kept as a constant, and only the segmentation method vary (for instance, EM.2 means EM algorithm with 2 gaussians). The character recognition rates (CRR) and precision rates (Prec) are computed on a ground truth basis as:

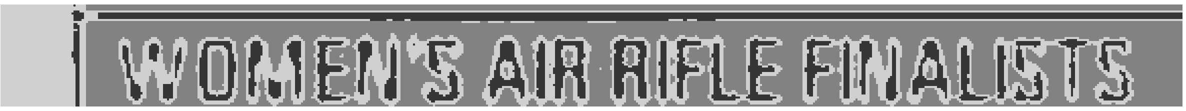
$$CRR = \frac{N_r}{N} \quad and \quad CRR = \frac{N_e}{N}$$

N is the true total number of characters, N_r is the number of correctly recognized characters and N_e is the total number of extracted characters. Additionally, we compute the word recognition rate to get an idea of the coherency of character recognition within one solution. For each text image, we count the words from the ground truth of that image that appear in the string result. The results are listed in table 1. It can be seen that, whatever the value of K, the GEM algorithm provides the best recognition results and similar precision comparing with standard Otsu's method and EM segmentation method. It is mainly due to the GEM adaptability, which, by learning the local spatial properties of the grayscale distribution, is noise adaptive and is able to better avoid over segmentation, as can be seen from the example of figure 3. Also, we can notice that the usual bimodality (K=2) hypothesis is not the best one in terms of CRR. The explanation might be the fact that, in some instances, text images are composed of the grayscale values of characters, some contrast region around characters, and background (see figure 2). The last rows of table 1 lists the results obtained by generating 9 hypotheses (using K=2 to 4). Even with our simple confidence criteria, the results are improved and attain a 92.5% CRR and 91.7% word recognition rate, which constitutes a reduction of about 35.3% for character and 22.4% for word error rate with respect to the standard Otsu's method.

The word recognition can be improved by keeping results from two or three hypotheses with the highest confidences. We obtain a 95.3% (resp. 97.8%) word recognition rate using the highest two (resp. three) hypotheses with GEM, which significantly improve the result of using only one hypothesis. This can yield better text searching results by offering more precise keywords in image/video indexing and retrieval system.

EM.2 : “WOMEN3 RIFLE FINALISTS”

GEM.2: “WOMEN3 A RIFLE FINALISTS”



EM.3: “',˜Oi'.lErJ 5 AIFI RIFLE FIrJALIS r ”

GEM.3: “ WOMEN'S AIR RIFLE FINALISTS”

FIG. 3 –. Noise adaptive: Segmentation output of the present 2 algorithms and recognition results using 2 and 3 Gaussians.

4 Conclusion

In this paper, we proposed a multi-hypotheses scheme for segmenting and recognizing embedded text of any grayscale value in image and video. Two segmentation algorithms (EM and GEM) are presented and compared. The experiments show that combining the hypotheses generated by different number of Gaussians using GEM algorithm improves the segmentation and recognition performances of using Otsu or basic EM with single number of Gaussians.