IDIAP RESEARCH REPORT

Monte Carlo Video Text Segmentation

Datong Chen and Jean-Marc Odobez
Dalle Molle Institute for Perceptual Artificial Intelligence
Rue du Simplon 4, CH-1920 Martigny, Switzerland
{chen, odobez}@idiap.ch

IDIAP–RR 03-07

Jan. 2003

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 03-07

Monte Carlo Video Text Segmentation

Datong Chen and Jean-Marc Odobez
Dalle Molle Institute for Perceptual Artificial Intelligence
Rue du Simplon 4, CH-1920 Martigny, Switzerland
{chen, odobez}@idiap.ch

Jan. 2003

Abstract - This paper presents a probabilistic algorithm for segmenting text embedded in video based on Monte Carlo sampling. The algorithm approximates the posterior of segmentation thresholds of video text by a set of weighted samples, referred to as particles. The set of samples is initialized by applying a traditional segmentation algorithm on the first video frame and further refined by random sampling under a temporal Bayesian framework. Results on a database of 6944 images demonstrated the validity of the algorithm.

1 Introduction

Text recognition in video is one of the key components in the development of advanced video annotation and retrieval systems. Text characters contained in video can be any grayscale value and embedded in multiple consecutive frames with complex backgrounds. For recognizing these video text, text characters are necessary to be segmented from complex backgrounds even when the whole text string is well located. Therefore, a large amount of work on text segmentation from complex background has been published in recent years.

Sobottka [1] clustered text pixels from images using a standard 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 devote efforts to perform better binarization such as combining global and local thresholding [2], M-estimation [3] and simple smoothing [4]. Furthermore, multiple hypotheses segmentation method, which assumes that the grayscale distribution can be k-modal (k=2,3,4), has been proposed by [5] and shown to improve the recognition performance. For using the temporal information of a text string in consecutive frames, Sato [6] and Lienhart [7] computed the maximum or minimum value at each pixel position over frames. However, this method can only be applied on black or white characters. Li [8] proposed a multi-frame enhancement which computes the average of pre-located text regions in multiple frames for further segmentation and recognition. The average image has smaller variance of noise but may propagate blur characters in frames. There is a common drawback of these temporal methods that they require accurate text image alignment in pixel level.

In this paper, we present Monte Carlo method for the segmentation of text characters of any grayscale values using temporal information. This method performs traditional segmentation on the text image in the first frame and then use particle filters to propagate this segmentation characteristics into other frames. The likelihood of the segmentation parameters is evaluated from the corresponding recognized text string based on language modeling and OCR statistics. By introducing randomness in the exploration of the space of possible segmentation parameters, the method allows to adapt to changes of grayscale values both in the text and background and to compensate OCR errors due to the low resolution of characters (before resizing and interpolation), the short length of the string and their unknown font.

2 Monte Carlo segmentation (MCS)

2.1 Bayes filtering

MCS is a sequential Bayes filter that estimates the posterior distribution of segmentation thresholds conditioned on grayscale values of pixels. Bayes filters address the problem of estimating the state x of a dynamic system from observations. For example, in video text segmentation the dynamic system is a video sequence, the state is the segmentation thresholds (up and low) of a text string, and the goal is to find the states that lead to an accurate segmentation or a correctly recognized string. The posterior is typically called the belief and is denoted:

$$B(x_t) = p\left(x_t|O_{0\ldots t}\right). \tag{1}$$

Here x_t denotes the state at time t, and $O_{0...t}$ denotes the observations starting at time 0 up to time t. For video texts, the observations are the grayscale text images extracted and tracked in consecutive video frames.

To derive a recursive update equation, we observe that expression (1) can be transformed by Bayes rule to

$$B(x_t) = \alpha p\left(O_t | x_t, O_{0...t-1}\right) p\left(x_t | O_{0...t-1}\right) \tag{2}$$

where α is the normalization constant

$$\alpha = p\left(O_t | O_{0...t-1}\right)^{-1}. \tag{3}$$

The prediction term $p\left(x_t | O_{0...t-1}\right)$ can be expanded by integrating over the state at time $t-1$:

$$p\left(x_t | O_{0...t-1}\right) = \int p\left(x_t | x_{t-1}, O_{0...t-1}\right) p\left(x_{t-1} | O_{0...t-1}\right) dx_{t-1}. \tag{4}$$

Substituting the basic definition of the belief (1) back into (4), we obtain a recursive equation

$$p\left(x_t | O_{0...t-1}\right) = \int p\left(x_t | x_{t-1}, O_{0...t-1}\right) B(x_{t-1}) dx_{t-1}.$$

Assuming independence of observation conditioned on the states and Markov model for the sequence of state, we have:

$$p\left(O_t | x_t, O_{0...t-1}\right) = p\left(O_t | x_t\right) \tag{5}$$

and

$$p\left(x_t | x_{t-1}, O_{0...t-1}\right) = p\left(x_t | x_{t-1}\right). \tag{6}$$

Thus, we can simplify the belief as:

$$B(x_t) = \alpha p\left(O_t | x_t\right) \int p\left(x_t | x_{t-1}\right) B(x_{t-1}) dx_{t-1}. \tag{7}$$

The implementation of equation (7) requires to know two conditional densities: the transition probability $p\left(x_t | x_{t-1}\right)$ and the data likelihood $p\left(O_t | x_t\right)$. Both models are typically time-invariant so that we can simplify the notation by denoting these models $p\left(x^{'} | x\right)$ and $p\left(O | x\right)$ respectively, and we present them in the next subsection.

2.2 Probabilistic models for segmentation

Transition probability

In the context of video text segmentation, the transition probability $p\left(x^{'} | x\right)$ is a probabilistic prior on text threshold variations. The state space is a 2-D space constructed by the up (u) and low (l) thresholds of text grayscales $x = (l, u)$. We assume, in this paper, that the change of the text thresholds is basically yielded by noise, and the transition model is therefore defined as:

$$p\left(x^{'} | x\right) = \frac{1}{2\pi\sigma} \exp - \frac{(l^{'} - l)^2 + (u^{'} - u)^2}{2\sigma^2} \tag{8}$$

where the noise is modeled as Gaussian process with a constant variance σ.

Data likelihood

The data likelihood $p\left(O | x\right)$ provides an evaluation of the segmentation quality of the observed image

O given a pair of thresholds $x = (l,u)$. This evaluation could rely on the segmented image. However, computing accurate measures of segmentation quality in term of character extraction is difficult without performing some character recognition analysis. Besides, visually well segmented image does not always lead to a correct recognition. The OCR may produce errors due to the short length and the unknown font of the text string. Therefore, since ultimately we are interested in the recognized text string, the data likelihood will be evaluated on the output T of the OCR.

To extract the text string T, we first binarize the image O using x, and then remove non-characters using a connected component analysis step. A connected component is a group of pixels that connected in a binary image. We keep the connected components that satisfy constraints on size, height and width ratio and fill-factor as character components and apply an OCR software on the resulting binary image to produce the text string T.

To evaluate the data likelihood using string T, we need some prior information on text strings and on the OCR performance based on language modeling and OCR recognition statistics. From a qualitative point of view, when given text-like background or inaccurate segmentation, the OCR system produces mainly garbage characters like ., ,!, & etc and simple characters like i,l, and r. Let $T = (T_i)_{i=1..l_T}$ where l_T denotes the length of the string and each character T_i is an element of the character set $\mathcal{T}$:

$$\mathcal{T} = (0, \ldots, 9, a, \ldots, z, A, \ldots, Z, G_b)$$

in which G_b corresponds to any other garbage character. Finally, let us denote by H_a (resp. H_n) the hypothesis that the string T or the characters T_i are generated from an accurate (resp. a noisy) segmentation. The data likelyhood is defined as the probability of accurate segmentation H_a given the string T:

$$\begin{aligned} p(O|x) &\propto p(H_a|T) \\ &= \frac{p(T|H_a)p(H_a)}{p(T)} \\ &= \frac{p(T|H_a)p(H_a)}{p(T|H_a)p(H_a)+p(T|H_n)p(H_n)} \\ &= \frac{1}{1+\frac{p(T|H_n)p(H_n)}{p(T|H_a)p(H_a)}}. \end{aligned}$$

We estimated the noise free language model p$(.|H_a)$ by applying the CMU-Cambridge Statistical Language Modeling (SLM) toolkit on Gutenberg collections[1]. A bigram model was selected. Cutoff and backoff techniques [9] were employed to address the problems associated with sparse training data for special characters (e.g. numbers and garbage characters). The noise language model p$(.|H_n)$ model was obtained by applying the same toolkit on a database of strings collected from the OCR system output when providing the OCR input with either badly segmented texts or text-like false alarms coming from the text detection process. Only a unigram model was used because the size of the background dataset was insufficient to obtain a good bigram model. The prior ratio on the two hypotheses $\frac{p(H_n)}{p(H_a)}$ is modeled as:

$$\frac{p(H_n)}{p(H_a)} = b,$$

where the b is a bias that can be estimated from general video data. The data likelihood is then given by:

$$p(O|x) \propto \frac{1}{1+\frac{\prod_{i=1}^{l_T} \mathrm{p}(T_i|H_n)}{\mathrm{p}(T_1|H_a)\prod_{i=2}^{l_T} \mathrm{p}(T_i|T_{i-1},H_a)} * b}.$$

1. www.gutenberg.net

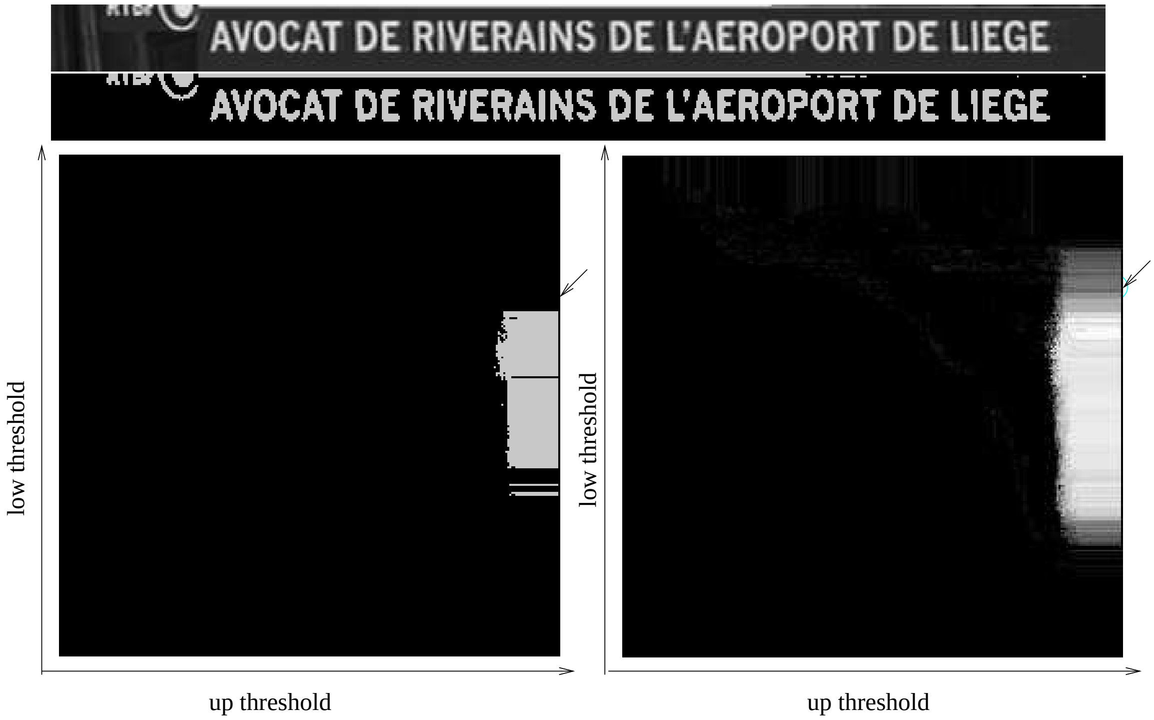

FIG. 1 – *Data likelihood approximation: the observed text image is displayed at the top. The second image displays the results of applying Otsu binarization, which corresponds to OCR output "V AVOCAT DE RIVERAINS DE L AEROPORT DE iIEGE". In the last row, the left image shows the states that correspond to the recognition of all the words in the groundtruth, the right image displays the proposed data likelihood at all the states.*

Figure 1 shows the groundtruth data likelihood and the proposed data likelihood of the image at all the possible states, illustrating that our probabilistic model is accurate. If the initial state (here provided by an Otsu algorithm and shown with an arrow in the images) leads to an incorrectly recognized text string, we will still be able to find a state that provide the correct string through the adaptation of a recursive particle approximation presented below.

2.3 Particle approximation

The idea of particle filter is to represent the belief $B(x)$ by a set of m weighted samples distributed according to $B(x)$:

$$B(x) \approx \sum_{i=1}^{m} w^i \delta\left(x^i - x\right),$$

where δ is the mass choice function ($\delta(0) = 1$, otherwise $\delta(x) = 0$). Each x^i is a sample of the random variable x, that is a hypothesized state (pair of thresholds). The initial set of samples represents the initial knowledge $B(x_0)$ (approximated by a set X of samples) and can be initialized using an Otsu algorithm applied on the first image. The recursive update is realized in three steps. First, sample x_{t-1}^i from the approximated posterior $B(x_{t-1})$. Then, sample x_t^i from the transition probability $p\left(x_t|x_{t-1}^i\right)$. Finally, assign $w^i = p\left(O_t|x_t^i\right)$ as the weight of the ith sample. In our case, since the number of samples per image will be low, we will add the new particles to the set X of samples instead of replacing the old values with the new ones.

1. initial X using an Otsu algorithm;

FIG. 2 – *Examples of located embedded text in video.*

methods	Ext.	CRR	Prec.	WRR
Average	3664	88.9%	80.1%	61.9%
MCS	3637	93.9%	85.3%	72.0%

TAB. 1 – *Performance comparison between the MCS (m=3) and the average image method: character recognition rate (CRR), precision (Prec.) and word recognition rate (WRR)*

2. for each frame $t = 1, \ldots, n$ do step 3 and 4;
3. for $i = 1$ to m do
   sample $x_{t-1}^i \sim X$;
   sample $x_t^i \sim p\left(x_t^i | x_{t-1}^i\right)$;
   set $w_t^i = p(O_t | x_t^i)$;
4. add the m new samples (x_t^i, w_t^i) to X.
5. output the text string that corresponds to the segmentation with the highest data likelihood.

3 Experiments and discussion

The MCS algorithm was tested on text regions located and extracted from one hour of video provided by the CIMWOS [2] project, using the algorithm presented in [10]. The whole database consists of 250 text strings (3301 characters or 536 words) in 6944 text images (about 28 images per text string in average). Figure 2 shows some image examples.

Performances are evaluated using character recognition rates (CRR) and precision rates (Prec) that are computed on a ground truth basis as:

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

N is the true total number of characters in the groundtruth, 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. Table 1 lists the results of the average image method [8] and the MCS algorithm with $m = 3$. The MCS algorithm performs better segmentation when the text image is noisy or the grayscale of characters span a wide range as shown in Figure 2. The results illustrate that the MCS algorithm significantly improves the character recognition, the precision, as well as the word recognition rate.

2. "Combined Image and Word Spotting" project granted by the European IST Programme

The CPU cost of the MCS algorithm depends on the size of state space, the number of samples, the thresholding operation and OCR computation. Using more than $m = 3$ particles per image does not change the performance of the algorithm. The average of samples per text string is thus around 60.

In this paper, we proposed a Monte Carlo method for segmenting and recognizing embedded text of any grayscale value in image and video based on particle filter. The MCS algorithm has two main advantages for segmenting video text. First, the MCS algorithm is adaptable to the computational resources by sampling in proportion to the posterior likelihood. This enable us to propose an accurate probability model based on OCR results instead of estimating the posterior of segmentation based on segmented images. Second, the MCS algorithm is very easy to implement and also easy to be extended to other state space, such as parameters of local thresholding techniques (e.g. Niblack binarization).

4 Acknowledgment

This work has been performed partially with in the frameworks of the “Combined Image and Word Spotting (CIMWOS)” project granted by the European IST Programme.

Références

[1] K. Sobottka, H. Bunke, and H. Kronenberg, “Identification of text on colored book and journal covers,” in *Int. Conf. on Document Analysis and Recognition*, 1999, pp. 57–63.

[2] H. Kamada amd K. Fujimoto, “High-speed, high-accuracy binrization method for recognizing text in images of low spatial resolutions,” in *Int. Conf. on Document Analysis and Recognition*, Sept. 1999, pp. 139–142.

[3] O. Hori, “A video text extraction method for character recogntion,” in *Int. Conf. on Document Analysis and Recognition*, Sept. 1999, pp. 25–28.

[4] V. Wu, R. Manmatha, and E. M. Riseman, “Finding text in images,” in *Proc. ACM Int. Conf. Digital Libraries*, 1997, pp. 23–26.

[5] D. Chen, J.M. Odobez, and H. Bourlard, “Text segmentation and recognition in complex background based on markov random field,” in *Int. Conf. Pattern Recognition*, 2002, vol. 2.

[6] T. Sato, T. Kanade, E. K. Hughes, M. A. Smith, and S. Satoh, “Video OCR: indexing digital news libraries by recognition of superimposed caption,” in *ACM Multimedia System Special Issue on Video Libraries*, Feb. 1998, pp. 52–60.

[7] R. Lienhart and A. Wernicke, “Localizing and segmenting text in images and videos,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, no. 4, pp. 256–268, 2002.

[8] H. Li and D. Doermann, “Text enhancement in digital video using multiple frame integration,” in *ACM Multimedia*, 1999, pp. 385–395.

[9] S.M. Katz, “Estimation of probabilities from sparse data for the language model component of a speech recognizer,” *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 35, pp. 400–401, 1987.

[10] D. Chen, H. Bourlard, and J-Ph. Thiran, “Text identification in complex background using svm,” in *Int. Conf. on Computer Vision and Pattern Recognition*, Dec. 2001, pp. 621–626.