Accepted for publication in
Int. Journal of Pattern Recognition and Artificial Intelligence, 2005

Monte Carlo Video Text Segmentation

Datong Chen (1), Jean-Marc Odobez (1) and Jean-Philippe Thiran (2)

(1) IDIAP Research Institute,
Rue du Simplon 4, 1920 Martigny, Switzerland
(2) Swiss Federal Institute of Technology Lausanne (EPFL),
Signal Processing Institute (ITS), 1015 Lausanne, Switzerland

Abstract

This paper presents a probabilistic algorithm for segmenting and recognizing text embedded in video sequences based on adaptive thresholding using a Bayes filtering method. The algorithm approximates the posterior distribution of segmentation thresholds of video text by a set of weighted samples. The set of samples is initialized by applying a classical segmentation algorithm on the first video frame and further refined by random sampling under a temporal Bayesian framework. This framework allows us to evaluate an text image segmentor on the basis of recognition result instead of visual segmentation result, which is directly relevant to our character recognition task. Results on a database of 6944 images demonstrate the validity of the algorithm.

Keywords: particle filter, Bayesian filter, image segmentation, video OCR

1 Introduction

Text recognition in video sequences, which aims at integrating advanced optical character recognition (OCR) and text-based searching technologies, is now recognized as one of the key components in the development of content-based multimedia annotation and retrieval systems. Content-based multimedia database indexing and retrieval tasks require automatic extraction of descriptive features

that are relevant to the subject materials (images, video, etc.). The typical low level features that are extracted in images and videos include measures of color [21], texture [13], or shape [14]. Although these features can easily be extracted, the interpretation in terms of image content is hard to obtain. Extracting more descriptive features and higher level entities, for example text [2] or human faces [20], has attracted more and more research interest recently. Text embedded in video, especially captions, provide brief and important content information, such as the name of players or speakers, the title, location and date of an event, etc. These text can be considered as a powerful feature (keyword) resource. Besides, text-based search has been successfully applied in many applications while the robustness and computation cost of the feature matching algorithms based on other high level features are not adequate to be applied to large databases.

The recognition of characters has become one of the most successful applications of technology in the field of pattern recognition and artificial intelligence. However, current optical character recognition (OCR) systems are developed for recognizing characters printed on clean papers. Applying the current OCR systems directly on video text leads to poor recognition rates typically from 0% to 45% [10, 18]. The reason is that text characters contained in video can be of any grayscale values and embedded in multiple consecutive frames with complex backgrounds. For recognizing these video text characters, it is necessary to segment text from 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. Generally, a segmentation of text image can be regarded as a process that searches for a pair of thresholds (lower and upper) covering the grayscale values of text pixels. Lienhart [11] and Sobottka [19] clustered text pixels from images using a standard image segmentation or color clustering algorithm. Although these methods can somehow avoid the text detection work, they are very sensitive to noise and character size. Most video text segmentation methods are performed after pre-locating the locations of the text strings in the images. These methods generally assume that the grayscale distribution is bimodal and devote efforts to perform better binarization such as combining global and local thresholding [8], M-estimation [6] and simple smoothing [23]. Furthermore, multiple hypotheses segmentation method, which assumes that the grayscale distribution can be k-modal (k=2,3,4), has been proposed by [3] and shown to improve the recognition performance to 94% word recognition rate. In order to use the temporal information of a text string in consecutive frames, Sato [18] and Lienhart [12] computed the maximum or minimum value at each pixel position over frames. The values of the background pixels that are assumed to have more

variance through video sequence will be pushed to black or white while the values of the text pixels are kept. However, this method can only be applied on black or white characters. Li [10] proposed a multi-frame enhancement for unknown grayscale text which computes the average of pre-located text regions in multiple frames for further segmentation and recognition. The average image has a smaller noise variance but may propagate blurred characters in frames. A common drawback of these temporal methods is that they require accurate text image alignment at the pixel level.

In order to use the temporal information at a higher level than the pixel level, we can combine the different recognized text strings resulting from the application of an OCR system and segmented text images of the same text string extracted from different video frames. The threshold pairs computed in different frames may be different and therefore provide additional information in the recognition process. However, applying traditional segmentation on every frame causes two problems. One problem is that it is not efficient in terms of computation cost. For a video text string, the segmentation characteristics in different frames are varying but not completely unpredictable. Thus, the optimal threshold pair of the previous frame could be reused instead of performing individual segmentation again. The other problem is that a traditional segmentation algorithm usually relies on a predefined criterion which may not always correspond to the optimal threshold pairs in a video and, therefore, can not yield segmentation results that would lead to good recognition [22]. In other words, the segmentation quality in our case should be validated using recognition results instead of any predefined criterion on grayscale values of the image. Figure 1 shows an example of two segmentation results and their recognition results. The OCR software we used is RTK from EXPERVISION, which has about 99% recognition rate on clean page characters. Although the segmentation (a) of the word "lower" seems to be visually similar as the segmentation (b), it leads to worse recognition results.

To address these two problems, in this paper, we present a particle filtering based Monte Carlo method for the segmentation of text characters of any grayscale values, exploiting temporal information. The idea of particle filters was first developed in the statistical literature, and recently the same algorithm named as sequential Monte Carlo filtering [5, 1] or condensation algorithm [7] has shown to be a successful approach in several applications of computer vision [7, 15, 17]. The key point of this method is to represent the posterior distribution of text threshold pairs given the image data by a set of weighted random samples, referred to as particles. In other words, the method performs a traditional segmentation of the text image in the first frame and propagate the resulting

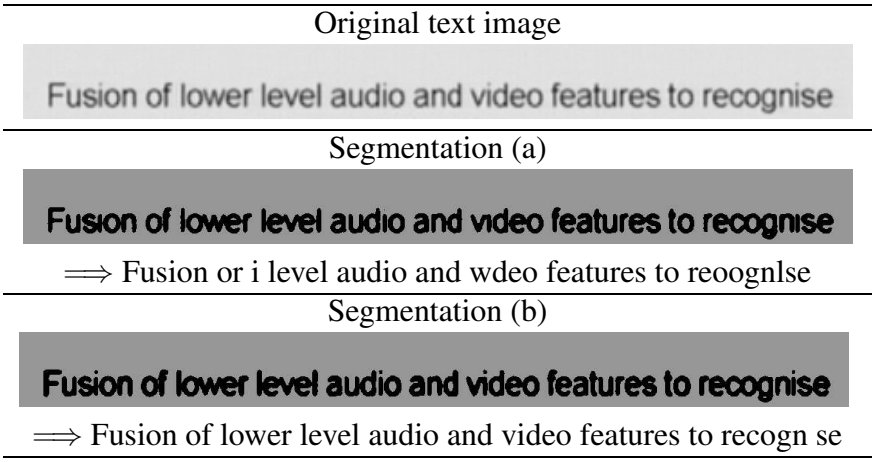

Figure 1: Different recognition results may be obtained from segmentation results, which are visually quite similar.

threshold pairs to other frames using particle filters. By introducing randomness in the exploration of the space of possible segmentation parameters in a Baysian framework, the particle representation allows to adapt to changes of grayscale values both in the text and background by simultaneously maintaining multiple-hypotheses. The advantage of the particle filtering in the presence of ambiguities and instabilities compensate OCR errors encountered when applying current OCR systems on video text due to the low resolution of characters (before resizing and interpolation), the short length of the string and their unknown font. In contrast to other filtering techniques that approximating posterior probabilities in parametric form, such as Kalman filters, this methodology allows to evaluate the likelihood of the segmentation parameters directly from the corresponding recognized text string based on language modeling and OCR statistics.

The details of this Monte Carlo segmentation algorithm are described in the next section and then the algorithm is evaluated and discussed with experiments in Section 3.

2 Monte Carlo video text segmentation algorithm

Monte Carlo video text segmentation (MCVTS) is a sequential Bayes filter that estimates the posterior distribution of segmentation thresholds conditioned on grayscale values of pixels. In this section, we will first introduce the Bayes fil-

tering framework, then investigated the two key components of Bayes filtering: dynamic model and data likelihood. Finally, we will give the particle approximation of the Bayes filters.

2.1 Bayes filtering

Bayes filters address the problem of estimating the state x of a dynamic system from observations. The posterior is typically called the belief and is denoted:

$$B(x_t) = p\left(x_t|O_1, O_2, \ldots, O_t\right). \tag{1}$$

Here x_t denotes the state at time t, and $O_1, O_2, \ldots, O_t$ denotes the observations starting at time 0 up to time t. For video text segmentation, the observations are the grayscale text images extracted and tracked in consecutive video frames. The state is the segmentation threshold pair of a text string, and the goal of video text segmentation is to find the states that lead to an accurate segmentation or, better, to a correctly recognized string.

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

$$B(x_t) = \alpha p\left(O_t|x_t, O_1, O_2, \ldots, O_{t-1}\right) p\left(x_t|O_1, O_2, \ldots, O_{t-1}\right) \tag{2}$$

where α is the normalization constant

$$\alpha = p\left(O_t|O_1, O_2, \ldots, O_{t-1}\right)^{-1}. \tag{3}$$

The prediction term $p\left(x_t|O_1, O_2, \ldots, O_{t-1}\right)$ can be expanded by integrating over the state at time $t-1$:

$$p\left(x_t|O_1, O_2, \ldots, O_{t-1}\right) = \int p\left(x_t|x_{t-1}, O_1, O_2, \ldots, O_{t-1}\right) p\left(x_{t-1}|O_1, O_2, \ldots, O_{t-1}\right) dx_{t-1}. \tag{4}$$

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

$$p\left(x_t|O_1, O_2, \ldots, O_{t-1}\right) = \int p\left(x_t|x_{t-1}, O_1, O_2, \ldots, O_{t-1}\right) B(x_{t-1}) dx_{t-1}.$$

According to the obvious independence between observations and an usual solution of avoiding high order statistical modeling, we assume independence of

observation conditioned on the states and a Markov model for the sequence of states. We therefore have:

$$p\left(O_t|x_t, O_1, O_2, \ldots, O_{t-1}\right) = p\left(O_t|x_t\right) \tag{5}$$

and

$$p\left(x_t|x_{t-1}, O_1, O_2, \ldots, O_{t-1}\right) = p\left(x_t|x_{t-1}\right). \tag{6}$$

Thus, we can simplify the recursive filtering equation 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. We will now present and evaluate them in sense of video text segmentation and recognition.

2.2 Probabilistic models for video text segmentation

2.2.1 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 upper (u) and lower (l) thresholds of text grayscales $x = (l, u)$. In this paper, we investigate four methods to model the transition probability.

Gaussian model - In this model, the change of the text thresholds is assumed to be due to additive noise, which is modeled as a Gaussian process with a constant variance σ. The transition probability is thus defined as:

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

Uniform model - The second method considers the transition model as a result of illumination or lighting change in the video sequence. The grayscale values of all or part of text characters increase or decrease by a constant value due to the

background motion behind transparent text or special visual effects. The transition probability is therefore defined as a uniform process:

$$p\left(x^{'}|x\right) = \begin{cases} \frac{1}{(l_{max}-l_{min})(u_{max}-u_{min}} & \text{if } l^{'} \in [l_{min}, l_{max}] \,\&\, u^{'} \in [u_{min}, u_{max}] \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

where the shifting range is modeled by a constant parameter α:

$$l_{min} = l - \alpha \quad \text{and} \quad l_{max} = l + \alpha,$$

and

$$u_{min} = u - \alpha \quad \text{and} \quad u_{max} = u + \alpha.$$

Adaptive uniform model - This is a relative of the uniform model in which the amount of shifting values of the thresholds depend on current state. Let two values $min = 0$ and $max = 255$ denote the minimum and the maximum values of the grayscale in image respectively. Given $x = (l, u)$, the shifting range l_{min} in equation (9) is adjusted by the distance between l and the min:

$$l_{min} = l - \alpha(l - min), \quad (10)$$

where $\alpha = 0.1$ is a constant experimentally decided. Similarly, we can defined:

$$l_{max} = l + \alpha(u - l), \quad (11)$$

and the shifting ranges of $u^{'}$ are defined as:

$$u_{min} = u - \alpha(u - l) \quad \text{and} \quad u_{max} = u + \alpha(max - u). \quad (12)$$

The typical distribution of $p\left(x^{'}|x = (150, 200)\right)$ in the adaptive uniform model is illustrated in figure 2.

Adaptive mixture model - To model the transition probability using both noise and light shifting, we can modify the above adaptive uniform model by applying a Gaussian noise model on the state space out of shifting range. Following the same definitions in equation (10), (11) and (12), the transition probability $p\left(x^{'}|x\right)$ is therefore defined as:

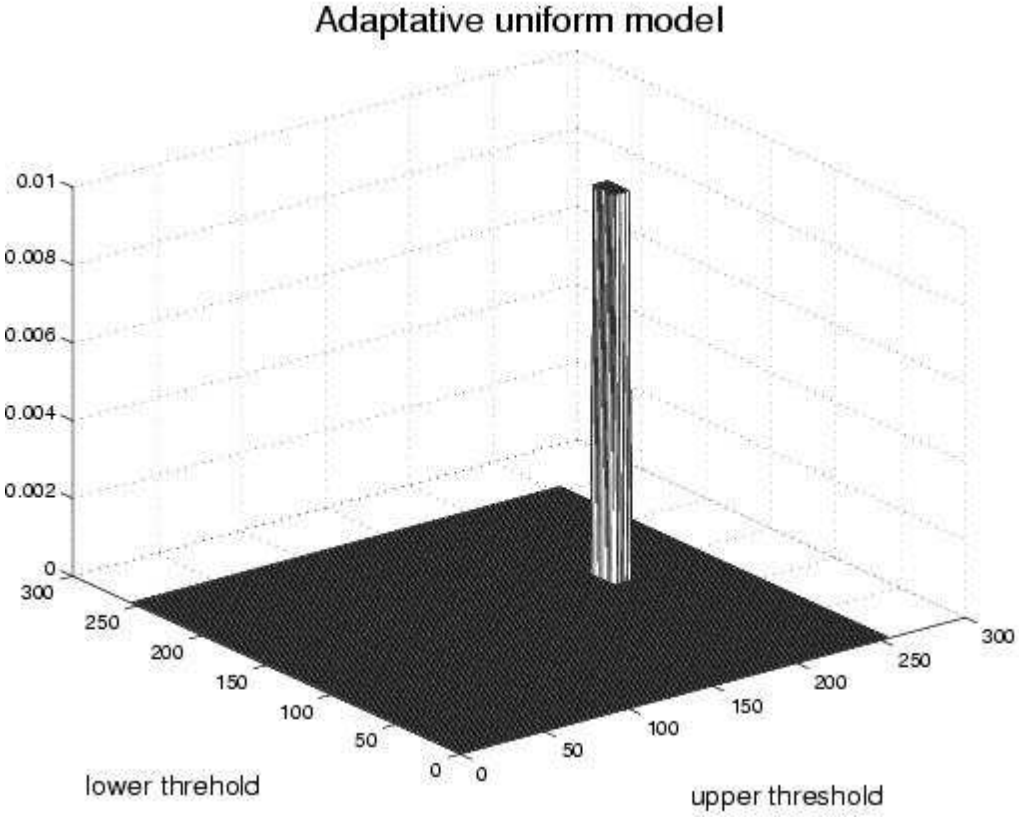

Figure 2: Adaptive uniform model of transition probability $p\left(x^{'}|x=(150,200)\right)$.

$$p\left(x^{'}|x\right)=\begin{cases}\frac{1}{\gamma} & \text{if } l^{'}\in[l_{min},l_{max}]\ \&\ u^{'}\in[u_{min},u_{max}]\\ \frac{1}{\gamma}e^{-\frac{(l^{'}-l_{min}^{max})^2+(u^{'}-u_{min}^{max})^2}{2\sigma^2}} & \text{otherwise,}\end{cases}\tag{13}$$

where

$$l_{min}^{max}=\begin{array}{ll}l_{min} & \text{if } l^{'}<l_{min}\\ l_{max} & \text{if } l^{'}>l_{max};\end{array}\tag{14}$$

and

$$u_{min}^{max}=\begin{array}{ll}u_{min} & \text{if } u^{'}<u_{min}\\ u_{max} & \text{if } u^{'}>u_{max}.\end{array}\tag{15}$$

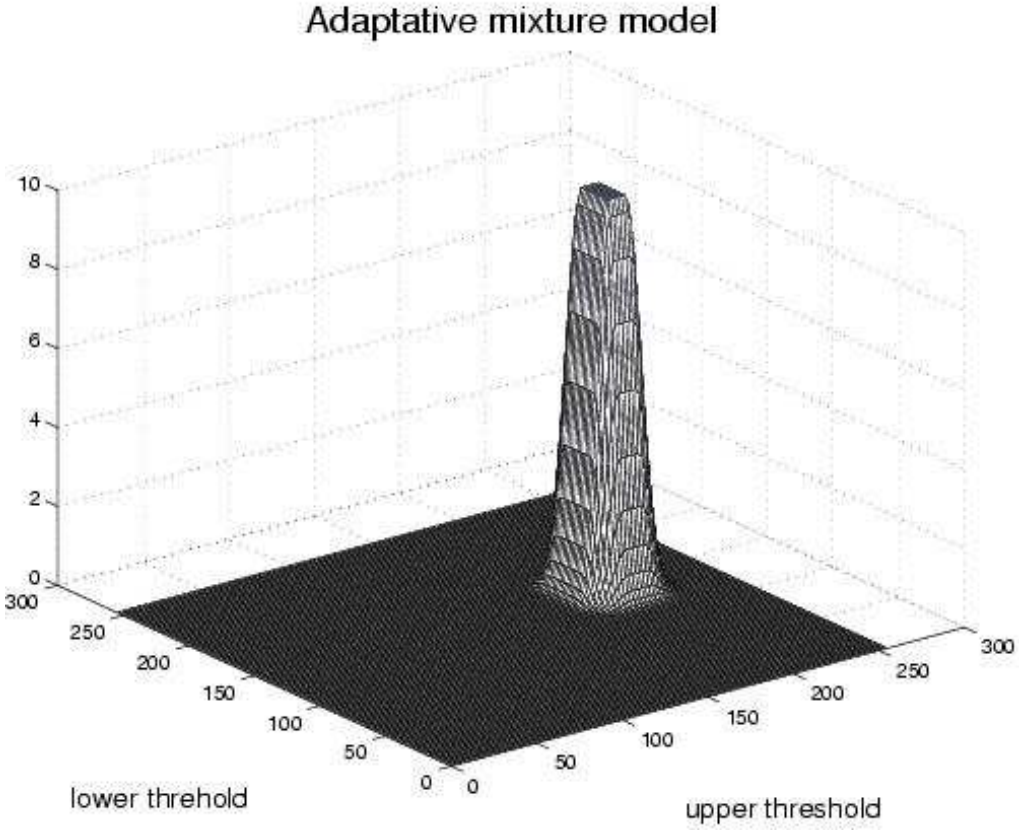

Figure 3: Adaptive mixture model of transition probability $p\left(x'|x=(150,200)\right)$.

γ is a normalization constant which does not affect the MCVTS algorithm. The typical distribution of $p\left(x'|x=(150,200)\right)$ in the adaptive mixture model is illustrated in figure 3.

2.2.2 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 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 noise regions using a connected component analysis step [3]. 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 exploit 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 us define a text string T as $T = (T_i)_{i=1..l_T}$ 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 likelihood is defined as the probability of accurate segmentation H_a given the string T:

$$p\left(O|x\right) \propto p(H_a|T) = \frac{p(T|H_a)p(H_a)}{p(T)}$$

Here $p(T)$ is given by:

$$p(T) = p(T|H_a)p(H_a) + p(T|H_n)p(H_n),$$

and the data likelihood is then proportional to:

$$p\left(O|x\right) \propto \frac{1}{1 + \frac{p(T|H_n)p(H_n)}{p(T|H_a)p(H_a)}}.$$

We estimated the noise free language model p($.|H_a$) by applying the wellknown CMU-Cambridge Statistical Language Modeling (SLM) toolkit on Gutenberg collections[1], which contains huge mount of text of books. A bigram model was

[1] www.gutenberg.net

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 $\mathrm{p}(.|H_n)$ model was obtained by applying the same toolkit on a database of strings collected from the OCR (RTK from EXPERVISION) 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:

$$p\left(O|x\right) \propto \frac{1}{1 + \frac{\prod_{i=1}^{l_T} \mathrm{p}(T_i|H_n)}{\mathrm{p}(T_1|H_a) \prod_{i=2}^{l_T} \mathrm{p}(T_i|T_{i-1},H_a)} * b}. \tag{16}$$

Figure 4 shows the groundtruth data likelihood, which is defined as $p(o|x) = 0$ if not all the words in the groundtruth are recognized, otherwise $p(o|x) = 1$. The figure also shows the proposed data likelihood of the image at all the possible states, illustrating that our probabilistic model is accurate. Even if the initial state (here provided by an Otsu algorithm [16] and shown with an arrow in the images) leads to an incorrectly recognized text string, the Bayesian filtering methodology , thanks to the introduction of random perturbation and our data likelihood model, will still be able to find a state that provides the correct string. The Bayesian filtering is implemented by a recursive particle filter that is described 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

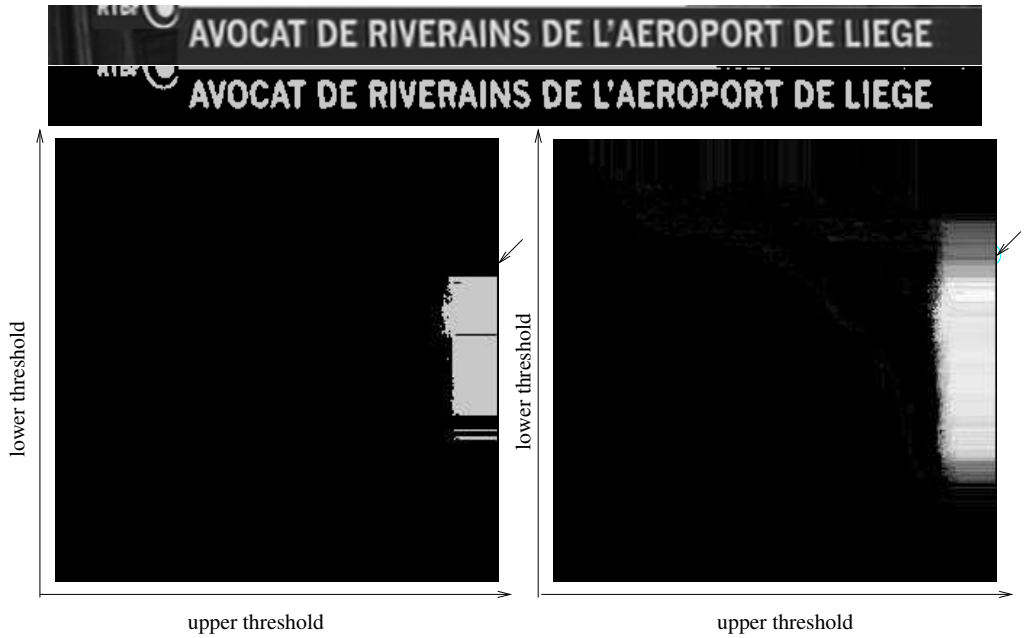

Figure 4: Data likelihood approximation: the observed text image is displayed at the top. The second image displays the results of applying Otsu binarization, which corresponds to OCR output "V AVOCAT DE RIVERAINS DE L AERO-PORT DE iIEGE". In the last row, the left image shows the states that lead to the recognition of all the words in the ground truth, the right image displays the proposed data likelihood at all the states.

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. Figure 5 shows the MCVTS algorithm presented in pseudo code.

Figure 6 illustrates the procedure of the MCVTS algorithm. The initial threshold pair $x = (120, 255)$ and $x = (0, 120)$ are obtained by using Otsu thresholding algorithm, which is not a correct solution in this case. After particle sampling in

1. initialize X using an Otsu algorithm;
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.

Figure 5: video text segmentation algorithm.

several frames, the states (threshold pairs) covered a wide range of thresholds in the state space. At the end, the threshold pair $x = (5, 82)$ gives the highest likelihood. The segmentation result using this optimal threshold pair leads to a correct OCR output as shown in the figure, though the pictogram at the right of "sabena" is interpreted as a "0".

3 Experiments and discussion

The MCVTS 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 [2]. 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 7 shows some image examples.

Performances are evaluated using character recognition rates (Recall) and precision rates (Precision) based on a ground truth. Recall and Precision are defined as:

$$Recall = \frac{N_r}{N} \ \ and \ \ Precision = \frac{N_r}{N_e}.$$

N is the true total number of characters in the ground truth, N_r is the number of correctly recognized characters and N_e is the total number of extracted characters.

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

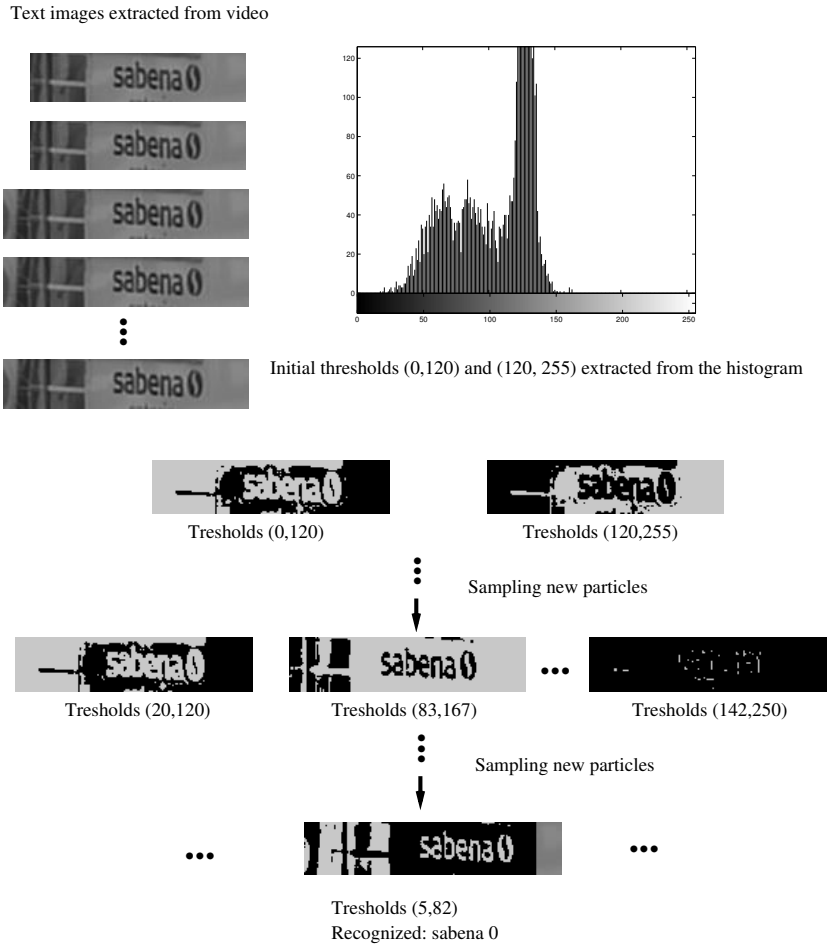

Figure 6: Video text segmentation using particle filtering.

In order to compare the performance of the MCVTS algorithm and former work, we implemented the average image method [10], which is the only method to our knowledge that works for unknown grayscale text and applied it on our database as a baseline system. Table 1 lists the results of the average image method and the MCVTS algorithm with $m = 3$. The results of baseline system show that around 89% of characters are able to be recognized in the database without introducing any randomness. All the four MCVTS algorithms gained some improvements in comparison with the baseline system. By checking the tested samples in the database, we found that the MCVTS algorithms performed better segmentation when the automatically detected text images were noisy, con-

Figure 7: Examples of located embedded text in video.

Methods	Ext.	Recall	Precision
Baseline system	3664	88.9%	80.1%
Gaussian MCVTS	3620	89.7%	81.8%
Uniform MCVTS	3584	90.5%	83.3%
Adaptive uniform MCVTS	3627	92.3%	84.0%
Adaptive mixture MCVTS	3637	93.9%	85.3%

Table 1: Performance comparison between the MCVTS (m=3) and the baseline system based on the average image method: extracted characters (Ext.), character recognition rate (Recall) and precision (Precision) The baseline system is the average image method re-implemented according to [10].

tained perturbation, or when the grayscale values of characters spanned a wide range, as shown in Figure 7. The results in table 1 also illustrates that the MCVTS algorithms not only significantly improves the character recognition but also the precision.

In all the four dynamic models proposed in the paper, the adaptive mixture model yields the best results in terms of character recognition rate and precision. Figure 8 illustrates the character recognition rates of MCVTS algorithms with varying m. All the four dynamic models give similar results when m is above 10, which shows that all these dynamic models lead convergence of the estimation of posterior belief. The dynamic model is an important factor only when the computation resource is limited (m is small).

The CPU cost of the MCVTS 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 with the adaptive mixture model does not change the performance of the algorithm. The average number of samples per text

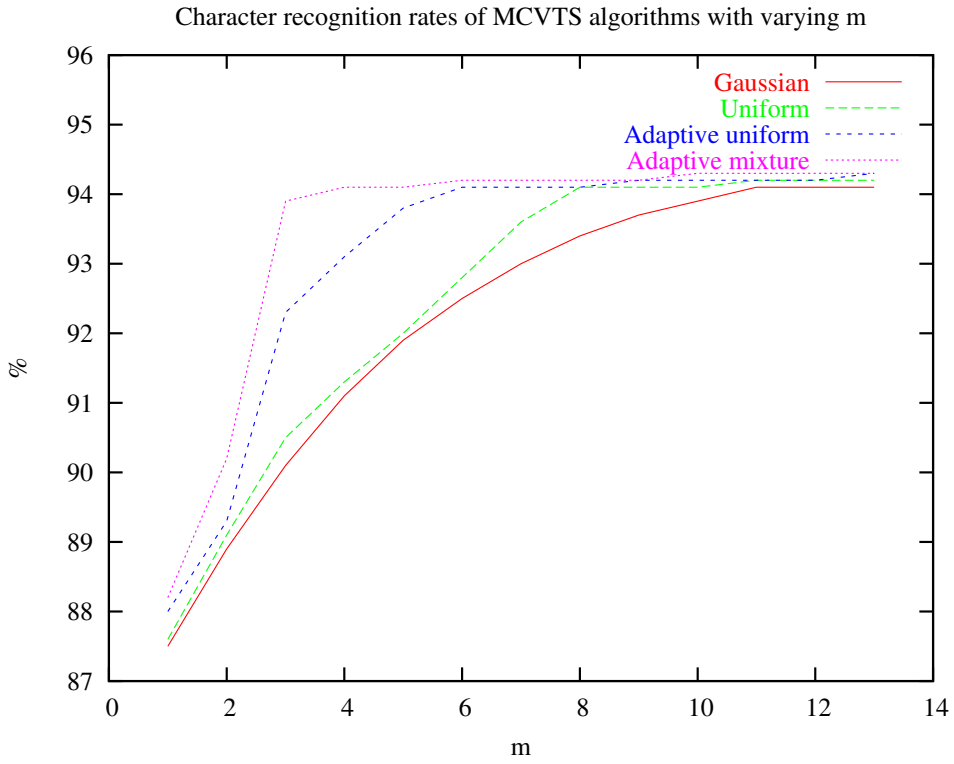

Figure 8: Character recognition rates of MCVTS algorithms with varying m.

string is thus around 80.

4 Conclusion

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 MCVTS algorithm has four main advantages for segmenting video text. Firstly, the algorithm proposes a methodological way to search for segmentation parameters that lead to accurate results. Secondly, the algorithm adapts itself to the data 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. Thirdly, the algorithm does not require precise tracking of text images among video frames at pixel level.

Finally, the MCVTS algorithm is very easy to implement and also easy to be extended to other state spaces, such as parameters of local thresholding techniques (e.g. Niblack binarization). An additional improvement of the MCVTS algorithm can be made by combining multiple recognition results of the same text string in character level instead of outputting the one that gives the highest data likelihood. Although this issue is not addressed in this paper, some details can be found in our recent work [4].

References

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian. *IEEE Trans. Signal Processing*, pages 100–107, 2001.

[2] D. Chen, H. Bourlard, and J-Ph. Thiran. Text identification in complex background using SVM. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 621–626, Dec. 2001.

[3] D. Chen, J.M. Odobez, and H. Bourlard. Text segmentation and recognition in complex background based on markov random field. In *Proc. IAPR Int. Conf. Pattern Recognition*, volume 2, Quebec City, Canada, 2002.

[4] Datong Chen and Jean-Marc Odobez. An algorithm for video character recognition error reduction using temporal information. IDIAP-RR-03 12, IDIAP, Feb. 2003.

[5] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[6] O. Hori. A video text extraction method for character recogntion. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 25–28, Sept. 1999.

[7] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *4th European Conf. Computer Vision*, volume 1, pages 343–356, 1996.

[8] H. Kamada and K. Fujimoto. High-speed, high-accuracy binarization method for recognizing text in images of low spatial resolutions. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 139–142, Sept. 1999.

[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*, 35:400–401, 1987.

[10] H. Li and D. Doermann. Text enhancement in digital video using multiple frame integration. In *Proc. ACM Multimedia*, volume 1, pages 385–395, Orlando, Florida, USA, 1999.

[11] R. Lienhart. Automatic text recognition in digital videos. In *Proc. SPIE, Image and Video Processing IV*, pages 2666–2675, Jan. 1996.

[12] R. Lienhart and A. Wernicke. Localizing and segmenting text in images and videos. *IEEE Trans. on Circuits and Systems for Video Technology*, 12(4):256–268, 2002.

[13] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(8):837–842, Aug. 1996.

[14] F. Mokhtarian, S. Abbasi, and J. Kittler. Robust and efficient shape indexing through curvature scale space. In *Proc. British Machine Vision Conference*, pages 9–12, 1996.

[15] K. Nummiaro, E. Koller-Meier, and L. Van Gool. Object tracking with an adaptive color-based particle filter. In *Proc. Symposium for Pattern Recognition of the DAGM*, Sep. 2000.

[16] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. on Systems, Man and Cybernetics*, 1(9):62–66, 1979.

[17] P. Perez, A. Blake, and M. Gangnet. Jetstream: Probabilistic contour extraction with particles. In *Proc. Int. Conf. on Computer Vision*, pages 424–531, Vancouver, July 2001.

[18] 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 *Proc. ACM Multimedia System Special Issue on Video Libraries*, pages 52–60, Feb. 1998.

[19] K. Sobottka, H. Bunke, and H. Kronenberg. Identification of text on colored book and journal covers. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 57–63, 1999.

[20] Rohini K. Srihari, Zhongfei Zhang, and Aibing Rao. Intelligent indexing and semantic retrieval of multimodal documents. *Information Retrieval*, 2(2/3):245–275, 2000.

[21] M. Swain and H. Ballard. Color indexing. *Int. Journal of Computer Vision*, 7:11–32, 1991.

[22] Christian Wolf, Jean-Michel Jolion, and Francoise Chassaing. Text localization, enhancement and binarization in multimedia documents. In *Proc. Int. Conf. on Pattern recognition*, pages 1037–1040, Quebec City, Canada., August 2002.

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