IDIAP RESEARCH REPORT

A New Method of Contrast Normalization for Verification of Extracted Video Text Having Complex Backgrounds

Datong Chen and Jean-Marc Odobez
IDIAP, Switzerland,
chen, odobez@idiap.ch

IDIAP–RR 02-16

Oct. 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 02-16

A New Method of Contrast Normalization for Verification of Extracted Video Text Having Complex Backgrounds

Datong Chen and Jean-Marc Odobez
IDIAP, Switzerland,
chen, odobez@idiap.ch

Oct. 2002

One of the difficulties of extracting text contained in images or videos comes from the variation of the grayscale values of the text and backgrounds. In this paper we propose a new method to normalize the contrast between text characters and backgrounds so that a trained machine learning tool can verify characters of grayscale values that have never been seen before. Experiments show that the proposed method used in training either a multilayer perception or a support vector machine yields better text verification comparing with other typical contrast measures.

1 Introduction

Content-based multimedia database indexing and retrieval tasks require automatically extracting descriptive features which are relevant to the subject materials (images, video, etc.). Text embedded in images and video, especially captions provide brief and important content information, such as the name of a player or speaker, the title, location and date of an event etc., and can be a powerful feature (keyword) resource above speech content. Technically, text-based searching have been successfully applied in many applications while the robustness and computation cost of the feature matching algorithms based on many high level features are not efficient enough to be applied on large databases. Therefore, text recognition in video and images, which aims at integrating advanced optical character recognition (OCR) and text-based searching technologies, is now recognized as a key component in the development of advanced video and image annotation and retrieval systems. However, text characters contained in images and videos can be any grayscale values (not always white), low resolution, variable size and embedded in complex backgrounds. Experiments show that applying conventional OCR technology directly leads to poor recognition rates. Therefore, an efficient algorithm for extracting text characters from background is necessary to fill the gap between image or video documents and the input of a standard OCR system.

Previous methods show that characters can be detected by exploiting the characteristics on vertical edge, texture and edge orientations. One system for localizing text in covers of Journals or CDs [9] regarded that text were contained in regions with high horizontal variance, and satisfied certain spatial properties. Smith et al. [6] localized text by first detecting vertical edges with a predefined template, then grouping vertical edges into text regions using a smoothing process. Wu et al. [8] described a text localization method based on texture segmentation. Texture feature was computed at each pixel from the derivatives of the image at different scales. In a more recent work, Garcia et al. [3] proposed a feature, called variance of edge orientation, for text localization which exploited the fact that text string contained edges of different orientations. These methods are usually fast but produce many false alarms because many background regions may also have strong contrast patterns.

Instead of manually designing features, some text detection systems trained the detectors using neural networks [4] [5] based on features extracted from fix-size blocks of pixels. Because the neural network based classification was applied on the whole image, the detection system is not very efficient in terms of computation cost and is not robust to the characters of any sizes or any grayscale values.

In one of our previous work [2], we proposed a localization/verification scheme to overcome these two problems. In this scheme, text blocks are quickly extracted in images with a low rejection rate and then verified using a SVM based on typical contrast measures.

However, if both grayscale values of characters and backgrounds are varying, the derivatives give out different values. In fact, the contrast of a text character is background dependent, which implies that the contrast may not be a stable feature for text verification. In this paper, we proposed a new method, called constant gradient variance (CGV), to normalize local contrast using both local and global variance of the gradient image.

2 A contrast normalization method

One of the main characteristics of text texture is that characters usually have strong contrast with backgrounds. To develop an text verifier with low false alarm rate, we will train machine learning tools on the basis of this contrast characteristic of the text.

2.1 Contrast measures

Local contrast in an image can be measured by computing its spatial derivatives. The first order spatial derivatives gives a high value at the position that has high contrast with respect to its neighbors. The second order spatial derivatives does not indicate contrast directly. It shows a position with the local maximum contrast as a zero-crossing and can therefore be used to detect edges.

Some common image transformations can also be good measures of local contrast for example, the discrete cosine transform (DCT), which is widely used in JPEG and MPEG compression scheme, is a representative feature in the frequency domain. The transform coefficients (without the mean) are representative feature of contrast in the frequency domain.

A character with a fixed grayscale value produces different contrast in different backgrounds. On the other hand, embedding different grayscale characters at the same position of a background also produces different contrasts. Thus, the contrast normalization aims at scaling the contrast so that the measure is independent to varied combinations of characters and backgrounds grayscale values.

Thus, we considered thresholding the contrast so that it has less variance in certain range. This leads to edges or more robustly a distance map, which only relies on positions of edges in images. The distance map [7] DM of a window X is defined as :

$$\forall p = (x,y) \in X, DM(p,B) = \min_{q=(x_i,y_i)\in B} d(q,p) \tag{1}$$

where, B is a set of edge points included in X, and d is a distance function. Although the distance map is independent of the grayscale value of characters, the base set B still relies on the contrast between text and background and the threshold employed in edge detection.

2.2 Constant gradient variance

To avoid the need for setting any threshold, we propose a new feature, called constant gradient variance (CGV), to normalize the contrast at a given point using the local contrast variance computed in a neighborhood of this point. Let us denote by $g(x,y)$ as the gradient magnitude at point (x,y). We compute the local mean $LM(x,y)$ and the local variance $LV(x,y)$ in a neighborhood S of the point (x,y):

$$LM(x,y) = \frac{1}{|S|} \sum_{(i,j)\in S} g(i,j) \tag{2}$$

$$LV(x,y) = \sum_{(i,j)\in S} \left(g(i,j) - LM(x,y)\right)^2 \tag{3}$$

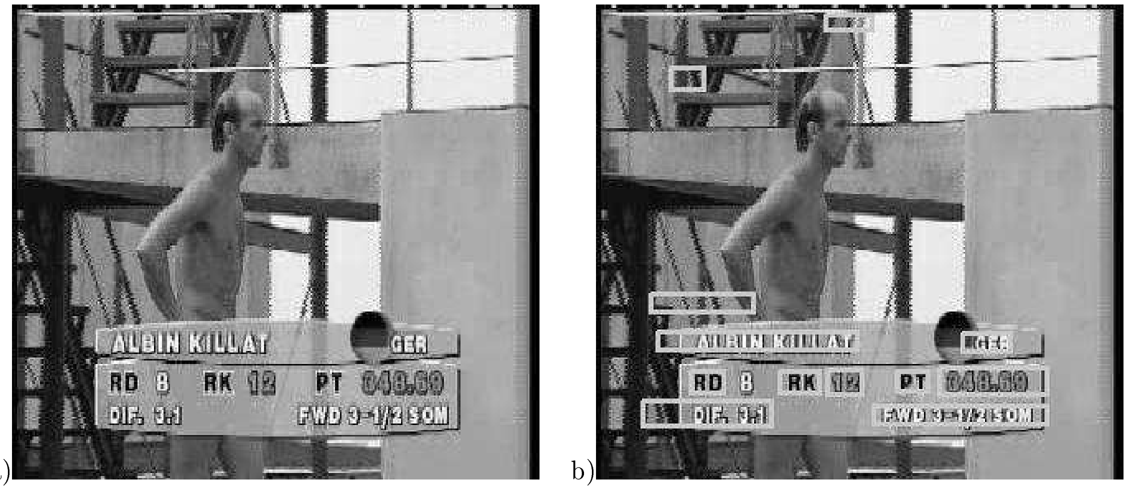
Then, the CGV value of (x,y) is define as:

$$CGV(x,y) = (g(x,y) - LM(x,y)) \sqrt{\frac{GV}{LV(x,y)}} \tag{4}$$

where GV denotes the global variance of the whole gradient image. Assuming that $g(x,y) \sim \mathcal{N}(LM(x,y), LV(x,y))$, i.e. follows a normal law with $LM(x,y)$ mean and $LV(x,y)$ variance, it easy to show that:

$$\begin{aligned} E[CGV(x,y)] &= 0 \\ E\left[(CGV(x,y))^2\right] &= GV \end{aligned} \tag{5}$$

where E denotes the expectation operator. Statistically, each local region in the CGV image thus has the same contrast variance. Note, however, that a site with a high CGV value still corresponds to an edge with a high local brightness contrast. In general, this method will also enhance the noise in regions with a uniform grayscale value. However such regions will be very rare in our case since the localization step only provides candidate text images that contain many vertical and horizontal edges.



a) b)

FIG. 1 – *Preprocessing of the text verification:(a) original image, (b) the rectangle boundaries of candidate text lines.*

3 Text verification

3.1 Preprocessing

The candidate text that need to be verified are provided by using a localization procedure with low CPU cost, proper false alarm rate and, importantly, low rejection rate [2].

This localization procedure can be addressed by estimating at each pixel position (x,y) the probability $P(x,y)$ of belonging to a text block and then grouping the pixels with high probabilities into regions. In order to obtain a fast algorithm, we exploit the fact that text regions contain short edges in vertical and horizontal orientations, and that these edges are connected each other due to the connections of character strokes. First, vertical and horizontal edges are detected using Canny algorithm [1]. Then, according to the type of edge (vertical or horizontal), different dilation operators are used so that the vertical edges are connected in horizontal direction while horizontal edges are connected in vertical direction. We consider the regions that only covered by both the vertical and horizontal edge dilation results as candidate text regions.

In order to deal with text lines rather than paragraphs, we detect the top and bottom baselines of horizontally aligned text strings. An additional step is then employed to discard the resulting regions that does not satisfy some typical text strings characteristics, such as fill factor, horizontal-vertical aspect ratio. Figure 1 illustrates a video frame and the located text lines using the this localization procedure.

3.2 Feature extraction

To test the performance of the proposed CGV model, we compare the performance of our text verification with input features extracted from spatial derivative images, distance map images and DCT coefficients. In each case, the training/testing feature vector will be computed from a 16×16 sliding window. The size of the neighborhood in the CGV method is 9x9 pixels. The spatial derivatives are computed using 2x2 operator. The edge set of distance map is detected by using the Canny algorithm.

Figure 2 illustrates some examples of the derivative features, the distance map feature and the CGV feature. DCT feature images are not shown in this figure because, visually, they are not very meaningful. It can be seen that the CGV features provided similar values around the characters for text of different grayscale values (see for instance the "UWE PESCHEL" and "RK" images.

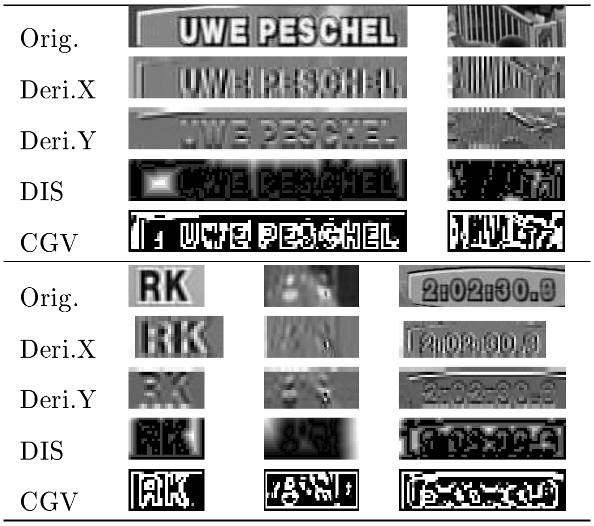


FIG. 2 – *Examples of three training features (Derivatives, distance map and CGV). The grayscale values shown in the feature images are scaled into the range of 0-255 for display.*

3.3 Machine learning tools

We train a verifier using either a multilayer perceptron (MLP) or a support vector machine (SVM). A MLP is based on empirical risk minimization, which minimizes the error over the data set, while a SVM is based on structural risk minimization, aims at minimizing a bound on the generalization error of a model in a high dimensional space.

MLP is a widely used neural network that consists of multiple layers (an input layer, hidden layers and an output layer) of neurons. The neurons in the hidden layer are fully connected to the input layer and are activated by using an Sigmoid function. The training of the MLP is performed by using backpropagation algorithm. SVM is a technique motivated by statistical learning theory and has been successful applied to numerous classification tasks. The key idea of SVM is to implicitly project input vectors into a space of higher dimension (possibly infinite), called feature space, where the two classes are hopefully more linearly separable. This projection is implicit because the learning and decision process only involve inner dot product in the feature space that can be computed using a kernel defined on the input space. We use typical Radial basis function (RBF) as the kernel. The kernel bandwidth σ as well as the number of neurons in the hidden layer of the MLP are chosen by using a K-fold cross-validation process. The MLP or SVM are trained on both positive (text) and negative (false alarms) examples resulting from the preprocessing step.

3.4 Verification

The feature vectors for text verification are extracted from using sliding windows with a slide step of 4 pixels. Thus, for each candidate text line r, we obtained a set of feature vectors $Z_r = (z_1^r, \ldots, z_l^r)$, where l is one-fourth of the length of text line r. Let $G(z_i^r)$ denotes the output of the MLP or the magnitude of the SVM, which indicates the confidence that the vector z_i^r belongs to a text line. The confidence of the whole candidate text line r is then defined as:

$$Conf(r) = \sum_{z_i^r \in Z_r} G(z_i^r) . \frac{1}{\sqrt{2\pi}\sigma_0} e^{\frac{d_i^2}{2\sigma_0^2}} \tag{6}$$

where, d_i is the distance in pixels from the center of the ith sliding window to the center of the text

TAB. 1 – *Error rate of SVM and MLP for text verification. DIS: distance mapping feature; DERI: derivative of image; CGV: constant gradient variation feature; DCT: DCT coefficients*

Training Tools	DIS	DERI	CGV	DCT
MLP	5.28%	4.88%	4.40%	4.95%
SVM	2.56%	3.99%	1.07%	2.92%

region r. We experimentally set $\sigma_0 = f(lenth)$. A candidate text line r is classified as a text region if $Conf(r) \geq 0$.

4 Experiments

Experiments were carried out on a database consisting of of 30 minutes video including advertisements, sports, interviews, news, movies, and compressed images including the covers of journals, maps, flyers. Each video frame or image has 352x288 or 720x576 resolution in JPEG or MPEG format and has been decompressed and converted into grayscale before applying text location and verification algorithms. Some video frames contain the same closed captions but with different backgrounds. After preprocessing, we extracted 9369 text lines and 7537 false alarms with a zero rejection rate.

To train MLP and SVM, we randomly selected 2,400 candidate text regions, resulting from the text localization step, and extracted 76,470 feature vectors (including 15.6% false alarms). The vectors are equally partitioned into two sets, a training set and a test set. This was done for each of the four test features and we insured that the training set for each of them contained the vectors extracted from the same windows (i.e. same image and location).

Table 1 lists the error rate of the test set of each of the four kinds of features using either MLP or SVM. Comparing among the four features, the CGV method gives the best result of in the both cases, which shows its superiority in modeling various contrast for text verification problem. Fusing all these four feature yielded a little better result, (0.72% error rate) than CGV result. However, it costs more CPU due to higher dimension of feature vectors.

Using the confidence value computed by Eq. 6, we can remove 7255 the 7537 false alarm regions (97% precision rate) while only reject 23 true text lines (0.24% rejection rate). This is better than the typical MLP detection error rates are 13-30% in literatures [5] although they are not really comparable. The SVM gave better results than the MLP using any of the four features because the SVM minimized the bound on the generalization error instead of the error over the data set. This may yield a better generalization for unseen backgrounds in the test set.

The final recognition results are given by using an OCR software [1] based on a segmentation scheme, and obtained a 96.8% character recognition rate and a 93.9% word recognition rate.

5 Conclusion

In this paper, a new feature extraction method was proposed for verifying text of any grayscale values in images or videos using machine learning tools. This method normalize the gradient image so that each local region has the same local variance. The variation of the contrast produced by varying grayscale values of characters and backgrounds is therefore reduced or ideally becomes a constant in the proposed CGV feature space. Comparison with other typical contrast measures showed that this CGV method could greatly improve the performance of text verification using MLP and SVM learning tools.

1. Expervision

6 Acknowledgment

The authors would like to thank Dr. Samy Bengio, and Ronan Collobert for their comments on this work.

Références

[1] J. F. Canny, "A computational approach to edge detection," **IEEE Trans. on Pattern Analysis and Machine Intelligence**, vol. 8, no. 1, pp. 679–698, 1986.

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

[3] C. Garcia and X. Apostolidis, "Text detection and segmentation in complex color images," in **Int. Conf. on Acoustics, Speech and Signal Processing**, 2000, pp. 2326–2329.

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

[5] 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.

[6] M. A. Smith and T. Kanade, "Video skimming for quick browsing based on audio and image characterization," Techn. Report CMU-CS-95-186, **Carnegie Mellon University**, July 1995.

[7] J. Toriwaki and S. Yokoi, "Distance transformations and skeletons of digitized pictures with applications," **Pattern Recognition**, pp. 187–264, 1981.

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

[9] Y. Zhong, K. Karu and A. K. Jain, "Locating text in complex color images," **Pattern Recognition**, vol. 10, no. 28, pp. 1523–1536, 1995.