

# A NEW APPROACH FOR TEXTURE FEATURES EXTRACTION: APPLICATION FOR TEXT LOCALIZATION IN VIDEO IMAGES

*Bassem Bouaziz<sup>1,2</sup>, Walid Mahdi<sup>1</sup>, Mohsen Ardabilain<sup>2</sup>, Abdelmajid ben hamadou<sup>1</sup>*

<sup>1</sup>MIRACL, Higher Institute of Computer Science and Multimedia, Sfax, Tunisia

<sup>2</sup>LIRIS UMR 5205 CNRS/INSA /Ecole Centrale de Lyon, France

bassem.bouaziz@fsegs.rnu.tn; {walid.mahdi, benhamadou.abdelmajid} @isims.rnu.tn  
mohsen.ardabilian@ec-lyon.fr

## ABSTRACT

In this paper we present a new texture feature extraction approach. Existing methods are generally time consuming and sensible to image complexity in terms of texture's regularity, directionality and coarseness. So that we propose a method which provide both rapidity and accuracy to extract and characterize texture features. It's based on Hough Transform technique combined with an extremity segment's neighbourhood analysis and a new computation algorithm to extract segments and detect regularity. Experimental results show that this approach is robust and can be applied not only to texture analysis but also to detect text within video images.

## 1. INTRODUCTION

Texture has long been an important research topic in image processing. Successful applications of texture analysis methods have been widely found in industrial, biomedical, and remote sensing areas. Texture analysis [2] allows locating pattern (ie. Object) within an image and achieve one of demanded applications in computer vision : image or video indexing and retrieval based on textural information [3] [4][10]. Without losing generality, two stages are required. The first one consists on assigning a texture feature vector to each pixel in an image. The feature extractor, which is usually based on filtering the textured image [5] [7], must be then able to discriminate among several textures. The goal of the second stage is indexing based on the extracted features vectors. In order to allow dimensionality reduction, the feature vectors are also combined to provide a new index of texture features as a quantitative description which is useful for image searching and retrieval by means of a similarity measure. Texture feature extraction lies essentially on image processing techniques. It is usually performed by linearly transforming or filtering the textured image [5][7] followed by some energy measure or non-linear operator.

In this work, we focus extraction of perceptual textural attributes which are regularity, directionality and coarseness. We remind here that these features have been specified in MPEG-7 [6] and exactly in (TBD) Texture Browsing Descriptor. Such features can be used as interesting clues in video images text detection and localization process.

The rest of the paper is organized as follows. The proposed computation method is presented in Section 2. In Section 3, experimental results are described. Finally, we conclude in Section 4.

## 2. THE PROPOSED METHOD

Our method is based on local application of Hough transform combined with use of transformation matrix to obtain rapidity and accuracy. We indicate that local application of Hough transform minimize influence of noise provoked by isolated pixels, this influence is insignificant in segment detection context. As for rapidity, it's judicious to minimize computation time when using complex primitives and costly operations. In the following we will detail our approach which is adapted to features texture extraction (ie. regularity and directionality) and particularly in video text localization. This approach present four steps: Sweeping the image, detecting segments, storing information about these segments and detecting regularity.

### 2.1. Image sweeping

Initially we studied access manners to line segment pixels appearing in an image.

Let S a set of collinear pixels forming a line segment within an image. A simple and rapid way to get extremities of a given segment is to sweep sequentially the image from top to bottom and from left to right. In this case, the angles  $\theta$  formed by segment with the horizontal axis verify the inequalities  $0 \leq \theta \leq \pi$ . When a line segment is detected, stored and removed from the image, the sequential search continues until sweeping the whole image.

## 2.2. Segment's detection

When a line segment extremity is reached, we sweep all directions  $\theta$  between 0 and  $\pi$  to find direction where most connected pixels exists. In all cases, when the number of connected pixels is important or exceed a given threshold information concerning the segment  $(r, \theta)$  are stored and the segment is removed from the image (figure 1). In other cases it's considered as noise (isolated pixels, short segments) and it's simply removed.

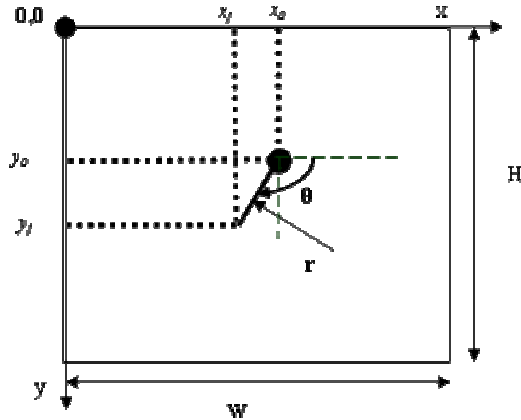


Figure 1. Features segment representation

The detection of segment direction is the most costly step in line detection algorithm, so requires a particular attention. In the discrete image space, line segment pixels having a direction  $\theta$ , length  $r$  and beginning coordinates  $(x_0, y_0)$  can be represented by:

$$P(X_{0..r,0..\theta}, Y_{0..r,0..\theta}) = P(x_0 + 1 + E(m \cdot \cos(\theta)), (y_0 + 1 + E(m \cdot \sin(\theta))))$$

Where  $1 \leq m \leq r$ ,  $0 < \theta < \pi$  and  $E$  is entire part of a number.

In order to improve performances and avoid call of trigonometric functions inside our algorithm, we compute two transformation matrices in the initialization step.

$TX_{0..r,0..\theta}$  (table 1) and  $TY_{0..r,0..\theta}$  (table 2).

Table 1. Transformation matrix  $TX_{0..r,0..\theta}$

0	←	...	→	180
$1+E(r.\cos(0))$		...		$1+E(r.\cos(180))$
...		...		...
$1+E(2.\cos(0))$		...		$1+E(2.\cos(180))$
$1+E(1.\cos(0))$		...		$1+E(1.\cos(180))$

Table 2. Transformation matrix  $TY_{0..r,0..\theta}$

0	←	...	→	180
$1+E(r.\sin(0))$		...		$1+E(r.\sin(180))$
...		...		...
$1+E(2.\sin(0))$		...		$1+E(2.\sin(180))$
$1+E(1.\sin(0))$		...		$1+E(1.\sin(180))$

We note  $r$  the maximal length of line segment that we should detect in a direction between 0 and 180°.

$x_0, y_0$  are the coordinates of line segment extremity identified when sweeping the image. So each element  $X = x_0 + TX_{0..r,0..\theta}$  and  $Y = y_0 + TY_{0..r,0..\theta}$  represents pixel coordinates in the image space.

## 2.3. Segment's information storing

Let  $V_{0..r,0..\theta}$  be an accumulator, which values are initialized to zero in a new line segment extremity detection process, It's content is updated as follows :

$$V_{0..r,0..\theta} = \begin{cases} 1 & \text{if } (I(X(m, \theta), Y(m, \theta))) = 1 \\ 0 & \text{if } (I(X(m, \theta), Y(m, \theta))) = 0 \end{cases}$$

Where  $1 \leq m \leq r$ ,  $0 < \theta < \pi$

The obtained matrix  $V_{0..r,0..\theta}$  represents neighbourhood's information of a detected extremity concerning connected pixels.

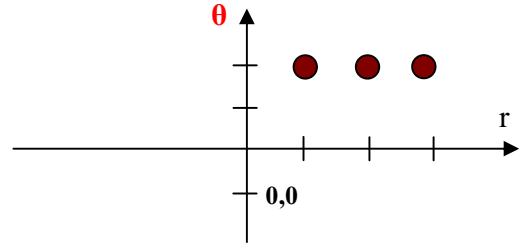


Figure 2. Space parameter  $(r, \theta)$  of a detected line segment

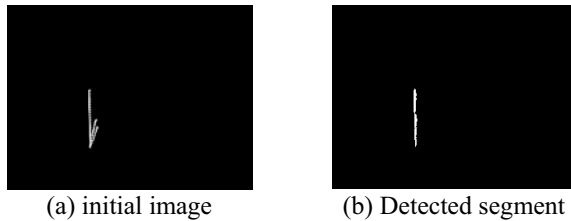
Starting from the assertion that some images are imperfect we propose amelioration to our approach. So, in one hand we introduce a security pace  $p_s$  that when exceeded we stop the computation in a given direction. In the other hand we introduce a neighbourhood  $\theta_v$  to a given direction. So an element of accumulator  $V_{0..r,0..\theta}$  is set to 1 only if :

$$(V_{r-p_s, \theta} = 1) \text{ or } (V_{r, \theta - \theta_v} = 1) \text{ or } (V_{r, \theta + \theta_v} = 1)$$

This consideration of neighbourhood makes our method able to detect imperfect segment as the case of an edge image.

## 2.4. Segment's removing

The last step of our algorithm consists on removing segment's pixels having length exceeding a threshold  $S_r$ . This threshold represents the minimal length of segment that we should detect. This step is fast because information about these segment were already stored in the accumulator  $V_{0...r,0...θ}$ . Indeed an accumulator's cell set to 1 represents the transform of an edge pixel in parameter space.



**Figure 3.** Segment detection having direction  $=86^\circ$  and length  $r > S_r$

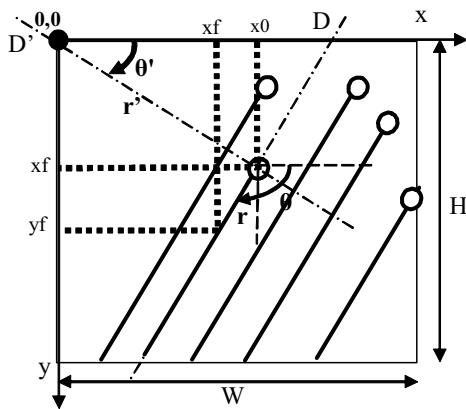
## 2.4. Detecting segment's regularity

Let  $D'$  the line passing through the origin  $(x_0, y_0)$ . This line is perpendicular to  $D$  that's the support line of segment to be detected.

$r'$  is the perpendicular distance from the origin to the line which support segment to be detected.

$\theta'$  is the angle between the norm of the line and the x axis.

So, for a given angle  $\theta'$  we can determinate, for a given direction  $\theta$ , all segments perpendicular to  $D'$  but having different values of  $r'$  (figure 4).



**Figure 4.** Segments having same direction but different distances from origin.

Regularity can be detected if distance ( $Dps$ ) between parallel segment is similar for a specified value of  $\theta'$ . This representation is very important and makes possible responding for a query that must return regular texture with a given direction or with any direction.

To quantify regularity, first we should express  $r'$  and  $\theta'$  in step with  $r$  and  $\theta$  that we had stored in the accumulator.

For  $\theta \in [0, \pi]$  two cases are presented:

$$\text{if } \theta \in [0, \pi/2] \text{ then } \theta' = \theta + \pi/2$$

$$\text{if } \theta \in [\pi/2, \pi] \text{ then } \theta' = \theta - \pi/2$$

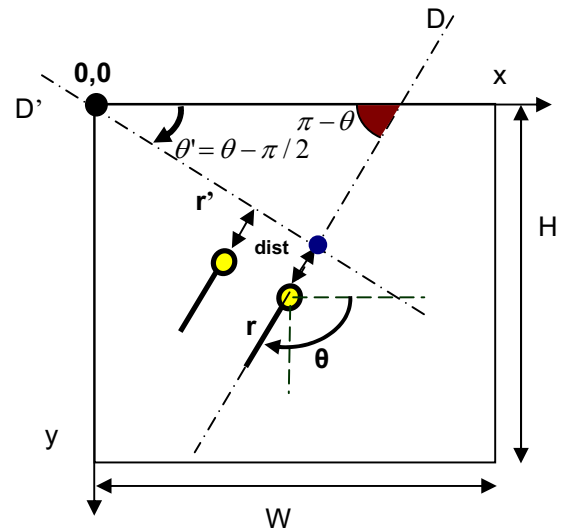
But,  $r'$  can be written in all  $\theta \in [0, \pi]$  as :

$$r' = |x_0 \sin(\theta) - y_0 \cos(\theta)|$$

## 2.5. Video text localization

Starting from hypothesis that a text is characterized by a regular texture and that characters forming a text line are oriented and aligned [1][8][9], we present a new technique to detect text within video image.

The regularity can be described by a similar distance between edge segment having the same orientation, but alignment needs another feature to be determinate (figure 5).



**Figure 5.** Added feature to detect segment alignment

Let  $dist$  be the distance between the extremity point of a segment and the intersection point between the support line of the same segment and the perpendicular passing from the origin. Aligned segment are those which have similar value of  $dist$ . So  $dist$  can be presented as follow :

$$dist = |y_0 \sin(\theta) + x_0 \cos(\theta)|$$

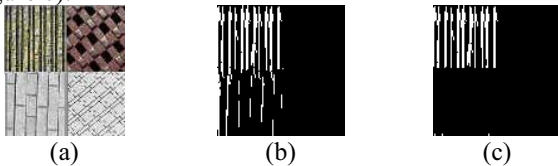
## 3. EXPERIMENTAL RESULTS

The experimental data are issued from two databases. The first one contains 120 images characterized by different textures with different orientations. The second concerns various videos images captured from some movies and

various genres. All experimental data have been chosen for the variety of text style and image complexity. Following sub-sections, illustrate good experimentation results.

### 3.1. Experiments on Texture

According to our method, texture can be represented by three features: direction ( $\theta$ ), segment length ( $r$ ) and regularity (distance between parallel segments ( $Dps$ )). In the following we present our texture features extraction results (figure 6):

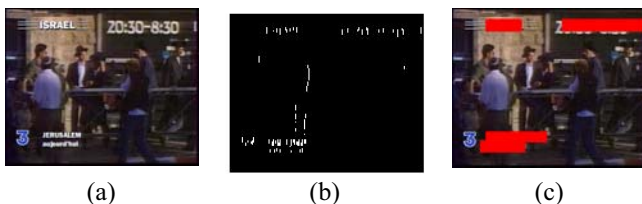


**Figure 6.** Texture features extraction : (a) original image (b)  $80 \leq \theta \leq 100, r=10 px, 2 \leq Dps \leq 20$   
(c)  $80 \leq \theta \leq 100, r=10 px, 2 \leq Dps \leq 6$

As we see the first block is maintained because we decreased the limit of  $Dps$  from 20 to 6.

### 3.2. Experiments on video text localization

Experimentations realized to detect text in video, prove that our method is flexible and can easily be used for video text localization (figure 7). So that, we added to features already described above the feature  $dist$  which represent the alignment of text's segment.



**Figure 7.** Video text localization process  
(a) original image  
(b)  $80 \leq \theta \leq 100, r=4 px, 2 \leq Dps \leq 6$   
(c)  $b+ dist$  : remained regions after ' $dist$ ' feature integration

For the statistical experimentation, we have carried out exhaustive evaluations and comparison with other interesting methods presented respectively by Y.hao et al [8] and wolf et al [9] using a video database containing 205 video frames (Total\_frames) containing 964 text regions (Total\_text\_Regions). Note that we mean by false alarm the hallucination of text region.

**Table 3.** Statistical Detection Results for our method.

Method	Y.	W	Our
Total Missed Text Regions	17	59	13
Total False Alarms	97	63	24
Correct Localization Rate	88.17 %	87.34 %	96%
False Alarm Rate	10,06 %	6.54 %	2,48%

## 4. CONCLUSION

In this paper, we presented an efficient Hough transform based approach for texture feature's extraction. we proposed a method that provides both rapidity and accuracy to extract these features. We have carried a new computation geometric technique for detecting line segment's based on neighbourhood of segment's extremity.

The flexibility and performance of our method is proved by its applicability on video text detection. The experimentation that we have done on large video images selected from various kinds of movies shows that our technique is very efficient, and able to locate text regions with different character sizes and different styles, even in case of texts occurring within complex image background. The successful rate of text localisation is 96% and the false alarm rate is only 2,48%.

## 11. REFERENCES

- [1] B. Bouaziz, W. Mahdi, A. BEN Hamadou, "Automatic Text Regions Location in Video Frames", *The International conference on signal-image technology and internet based system*, IEEE SITIS'05, Yaoundé, pp. 2-9, November 27th - December 1st 2005
- [2] Text of ISO/IEC 15938-3 Multimedia Content Description Interface-Part 3: Visual, Final Committee Draft, ISO/IEC/JTC1/SC29/WG11, Doc. N4062, March 2001.
- [3] H. Tamura, S. Mori, T. Yamawaki, "Texture features corresponding to visual perception", *IEEE Trans. Syst. Man Cybern.* SMC8, pp.460-473, 1978.
- [4] F. Liu, R.W. Picard, "Periodicity, directionality and randomness: Wold features for image modeling and retrieval", *IEEE Trans. Pattern Anal. Mach. Intell.* pp. 722-733. 1996.
- [5] P. Wu, et al., "A texture descriptor for browsing and similarity retrieval", *Signal Process. Image Commun.* pp.33-43, 2000.
- [6] D.H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes", *Pattern Recogn.* pp. 111-122. 1981.
- [7] B.S. Manjunath, W.Y. Ma, "Texture features for browsing and retrieval of image data", *IEEE Trans. Pattern Anal. Mach. Intell.* pp.837-842, 1996.
- [8] Y. Hao, Y. Zhang, H. Zeng-guang and T. min, "Automatic Text Detection In Video Frames Based on Bootstrap Artificial Neural Network And CED", *Journal of WSCG*, WSG'2003, Plzen, Czech, February 3-7, 2003.
- [9] C. Wolf, J.M Jolion and F. Chassaing. "Text Localization, Enhancement and Binarization in Multimedia Documents". *International Conference on Pattern Recognition (ICPR)*, IEEE Computer Society, Canada, 2002.
- [10] K.L. Lee, L.H. Chen, "A new method for rough texture classification and class weight estimation for texture retrieval", *Pattern Recogn. Image Anal.* pp. 400-410, 2002.