From Picture to Speech: an Innovative Application for Embedded Environment

{Silvio Ferreira, Céline Thillou}, Bernard Gosselin
Faculté Polytechnique de Mons, Laboratoire de Théorie des Circuits et Traitement du Signal
Bâtiment Multitel - Initialis, 1, avenue Copernic, 7000, Mons, Belgium
Phone: +32 (0)65 354717 Fax: +32 (0)65 354729
E-mail: {silvio.ferreira,celine.thillou}@tcts.fpms.ac.be

Abstract— **Our research goals are to extend offline OCR technologies to embedded platforms. It implies two strong constraints. First, pictures will be taken without control on camera settings and a priori on text (font or size) and background. The second issue is to link several techniques together with an optimal compromise between computational constraints and recognition efficiency. Preliminary experiments led us to consider two operating modes in order to improve global results. The first situation is pictures of natural scenes while the other one is pictures of documents. Our algorithm aims at handling numerous situations despite hardware constraints, typical of mobile environment. The paper will present the overall description of the system and its future improvements.**

Keywords—**Text Detection, Thresholding, Image Segmentation, Character Recognition**

I. Introduction

Blind and visually impaired people represent at least one percent of the European population (Belgium counts more than one hundred thousand low-vision people). Most of their vision troubles do not allow them to have access to textual information. Reading is nevertheless of prime importance for daily autonomy, text being present everywhere, either under the form of documents (newspapers, books, mails, magazines, commercial products' label) or text in natural scenes (signs, screen, schedules).

Several efforts have been made in order to give the blind or visually impaired access to such information; two complementary approaches are generally used. The first approach tries to directly adapt the information support to the degree of blindness, by using either an optical zooming device that expands the character or Braille language. These solutions are not perfect. On one hand, optical enhancement solutions are cumbersome and not applicable in all cases, on the other hand, Braille language requires a complex learning and by the fact most of blind people do not know it. The second method consists in transforming textual information into speech information. Some solutions combining a scanner, a pair of loudspeakers and a computer currently exist. In addition to this material, the computer must be equipped with OCR, *Optical Character Recognition*, and TTS, *Text-To-Speech*, technologies. OCR software aims at converting images from the scanner into text information while TTS software converts text information into a speech signal. This method has proved to be efficient with paper documents but presents the inconveniences of being limited to home-use and to be exclusively designed for documents that can be put into a scanner.

In this paper, we will describe the development of a mobile automatic text reading system, which tries to remedy all these shortfalls. We will focus on the description of the system design taking into account first experiments' results. Three key technologies are required for this system: text detection, optical character detection and speech synthesis. The integration of all these technologies in embedded environment remains in itself a challenging problem due to numerous constraints:

- *Text image deterioration*: text acquired on a camera, is an alternative to scanners but brings text identification and characters segmentation problems. Solutions to the poor sensors resolution, image stabilization or variable lighting conditions need to be found.
- *Low computational resources*: the use of a mobile device such as a PDA or a smartphone limits the processing time and the memory resources. This adds practical integration difficulties in achieving an acceptable execution time.
- *Human-machine interface*: special care has been brought to the user interface due to limited abilities of visually impaired users. Research and development efforts are still deployed to customize an interface dedicated to blind users.

The next figure gives an overview of the system and the interactions between each sub-system.

This paper is organized as followed: section 2 describes text detection challenges and the approach we follow. Sec-

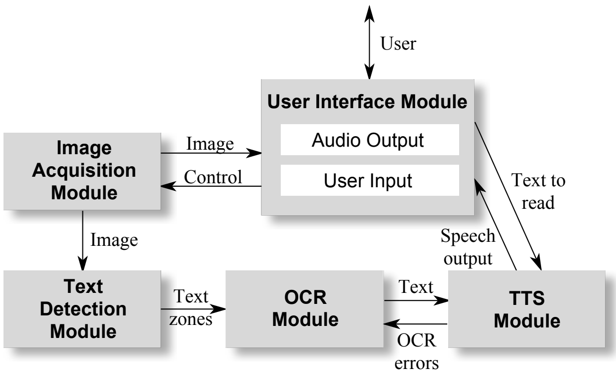

Fig. 1. System overview

tion 3 proposes our character segmentation and recognition algorithms. Section 4 provides a brief overview of our standard speech synthesis module adapted to this application in order to correct some OCR errors. In the final section, we address perspectives in research activities and conclude the paper.

II. Text detection

In this section, we address the problem of automatically finding text in a complex image. We mean images taken by a digital camera. Actually, camera-captured images present a bench of degradations, missing in scanner-based ones [2], such as blur, perspective distortion, complex layout, interaction of the content and background, compression, uneven lighting, wide-angle lens distortion, zooming and focusing, moving objects, sensor noise, intensity and color quantization.

Characters cannot be segmented by simple thresholding, and the color, size, font and orientation of the text are unknown. The main design choice is the kind of text occurrences, between scene text and document text [1].

A text is considered as a scene text when the text is recorded from a part of a scene (eg: road signs, posters on the street, street names). Unlike document characters, characters in scene images originally exist in 3-D space, and can therefore be distorted by a slant or a tilt, and by the shape of objects on which they are printed [2]. Text extraction from a natural scene has been studied, in projects such as vehicle license plates detection [3] or more general text detection algorithms [4] [5] [6]. A recent research study about text recognition operated by a robot deals with the same problems [7].

The other aspect of our investigations on text detection is to localize text areas from printed documents of any kind. We aim at developing a technique that will work for a wide range of printed documents like newspapers, books, restaurant menus, etc. Preliminary experiments led us to consider two global cases: images of text with nearly uniform background (mail, book) and more complex documents with degraded and/or textured background (commercial brochure, CD folder, etc.) in which text zones overlay a complex background. The development of the text detection algorithm will be separated into several steps according to the type of text support, from the simplest cases (pictures of documents) to the most complicated cases (scenes images of signs). Degradations due to image acquisition affect all these situations but not with the same degree of incidence.

At the current state of research advancement, we will describe below a text detection algorithm used for printed documents images with nearly uniform backgrounds: paper documents contain texts, full of characters with unknown size, font and orientation. Moreover, pictures are taken under variable lighting conditions. We process single frames independently (no video OCR) to reduce computational requirements and battery consumption in the mobile device. Most of the previous research works focus on extracting text from video. Techniques applied to images or video keyframes can broadly be classified as edge [8] [5] [9], color [10] [11], or texture based [12] [13] [14]. Each approach has its advantages and drawbacks concerning accuracy, efficiency and difficulty in improvement and implementation. Figure 2 illustrates several images of our database.

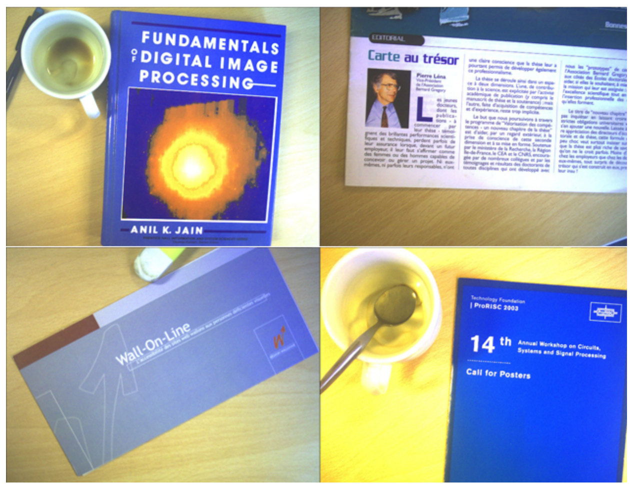

Fig. 2. Samples of text images from our database

A. Texture segmentation

The text detection technique is based on a texture segmentation approach. Text in a document is considered as a textured region to isolate; non-text contents in the image, such as blanks, pictures, graphics and other objects in the image must be considered as regions with different textures. The human vision can quickly identify text regions without having to recognize individual characters because text has textural properties that differentiate it from the rest of a scene. Instinctively, text has the following distinguishing characteristics:

- Characters contrast with their background
- Text possesses some frequencies and orientation information
- Text shows spatial cohesion: characters appear in clusters at a regular distance aligned to a virtual line

Features vectors represent each pixel and features images are then classified into several regions using an unsupervised clustering algorithm. The final step of this approach is to find the cluster representative of text region.

Our system captures colored images with a resolution of 1280 * 1000 pixels, theoretically enough to operate OCR. Manual focus is fixed at a distance of 40 cm in order to be able to enclose an A4-sized document. Images require specific pre-processing operations during the determination of regions of interest. Firstly images are converted into grayscale images and undersampled to a 256 * 341 pixels image due to computational limitations. Undersampling is operated by bicubic method just after a low pass filtering. Finally a contrast adjustment is operated in order to normalize lighting conditions.

B. Text Characterisation

By treating text as a distinctive texture, we propose a text characterization based on a bank of Gabor filters associated with an edge density measure. The features are designed to identify text paragraphs. None of them will uniquely identify text regions. Each individual feature will still confuse text with non-text regions but a society of features will complement each other and allow identifying text unambiguously. Physically interpreted, the Gabor transform acts like the Fourier transform but only for a small Gaussian window over the image, not the entire image. Mathematically, the 1-D Gabor transform can be expressed as

$$S_w(u,t) = \int_{-\infty}^{+\infty} s(x) * w(x-t) \exp^{-j2\pi u_o x} dx$$

S_w gives the Gabor transform of s(x) using the Gaussian window function w(x) centered at t. Thus, a Gabor function consists of a sine wave, with particular frequency, u_0, modulated by a Gaussian function. The concept can be extended to two dimensions as a sinusoidal plane of particular frequency and orientation modulated by a two-dimensional Gaussian envelope. In spatial domain, the two-dimensional Gabor filter h(x,y) is given by

$$h(x,y,\sigma_x,\sigma_y,w_x,w_y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp^{-\frac{1}{2}(\frac{x^2}{\sigma_x{}^2}+\frac{y^2}{\sigma_y{}^2})+j(xw_x+yw_y)}$$

where σ_x and σ_x are the standard deviations of the Gaussian envelope along the x and y directions, and w_x and w_y are the centered frequencies of the filter. One important characteristic of Gabor filter is its orientation selectivity, which can be understood when the expression of 2-D Gabor filter is rewritten in polar coordinates as

$$h(x,y,\sigma_x,\sigma_y,w,\Theta) = \frac{1}{2\pi\sigma_x\sigma_y} \exp^{-\frac{1}{2}(\frac{x^2}{\sigma_x{}^2}+\frac{y^2}{\sigma_y{}^2})+jw(xcos\Theta+ysin\Theta)}$$

where $\Theta = tan^{-1}(w_y/w_x)$ is the orientation and $w = \sqrt{w_x{}^2+w_y{}^2}$ is the radial frequency. The pixel intensity values in the output of the Gabor filter specify the extent to which the textured region is tuned to the frequency and orientation of the Gabor filter. The use of a bank of Gabor filters in extracting text features is motivated by various factors:

- It has been shown to be optimal in the sense of minimizing the joint two-dimensional uncertainty in space and frequency [15]
- Gabor filters closely resemble the mechanism of multichannel representation of the retinal images in biological visual system [16]
- Gabor filters can extract features in the presence of additive noise
- Gabor filters have band-pass nature, which is essential in analyzing a textured image

The bank of filters is composed of eight Gabor filters. Two frequencies have been adjusted to $\sqrt{2}/4$ and $\sqrt{2}/8$ and for each frequency, filters are designed in four orientations (0°, 45°, 90°, 135°).

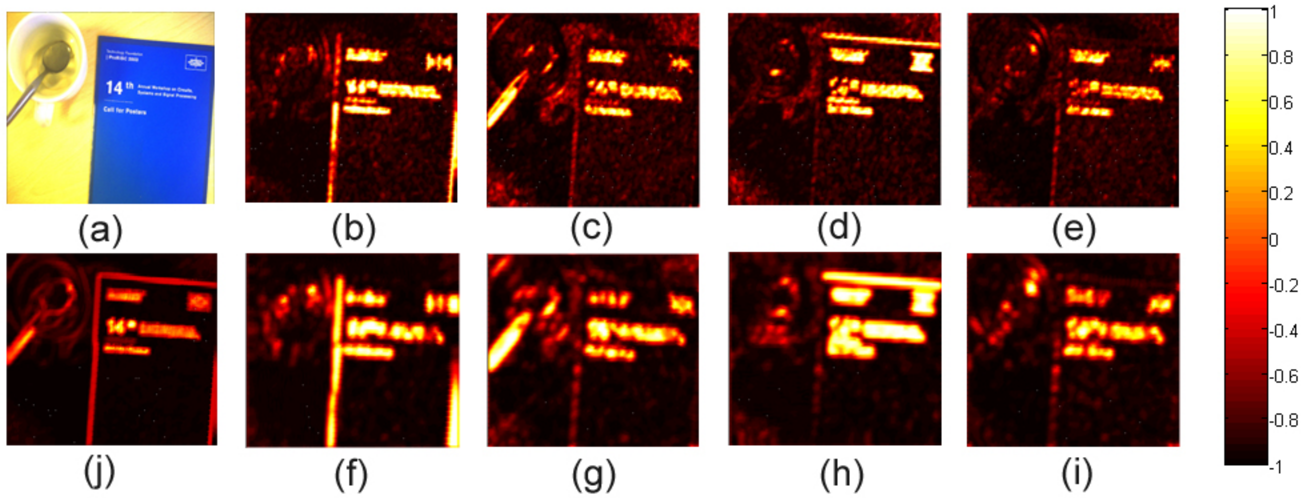

Fig. 3. Features images. (a) Original images (b-i) Gabor filters: (b) $u_0 = \sqrt(2)/4$ Θ = 0°; (c) $u_0 = \sqrt(2)/4$ Θ = 45°; (d) $u_0 = \sqrt(2)/4$ Θ = 90°; (e) $u_0 = \sqrt(2)/4$ Θ = 135°; (f) $u_0 = \sqrt(2)/8$ Θ = 0°; (g) $u_0 = \sqrt(2)/8$ Θ = 45°; (h) $u_0 = \sqrt(2)/8$ Θ = 90°; (i) $u_0 = \sqrt(2)/8$ Θ = 135°; (j) Edge density measure

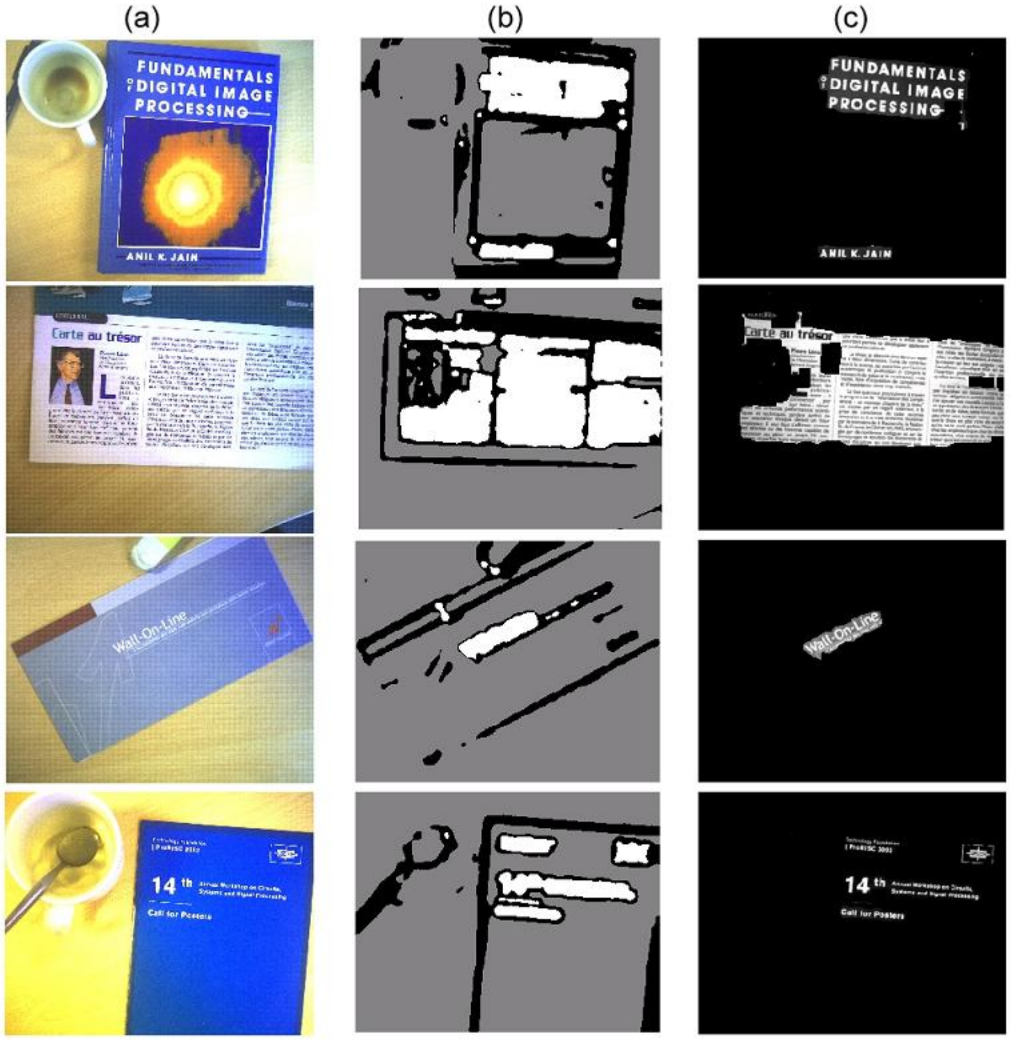

Fig. 4. Text detection results. (a) Original images (b) Text region clustering (c) Final results

This configuration of filters parameters allows our method to detect non-horizontal text at different fonts. A magnitude operation is required after each Gabor filtering. Indeed, to simulate human texture perception, some form of nonlinearity is desirable [17]. Nonlinearity is introduced in each filtered image by applying the following transformation [14]:

$$\Psi(t) = tanh(\alpha t) = \frac{1 - \exp^{-2\alpha t}}{1 + \exp^{-2\alpha t}}$$

For α = 0.25, this function is similar to a thresholding function like a sigmoid. The last operation before attaining feature vectors used on the clustering stage is a local averaging operation. Feature value is computed from the output of the nonlinear stage as the mean value in a small overlapping windows centered at each pixel. We associate to our features scheme a partially redundant feature, a local edge density measure based on Sobel filters [7]. This feature improves the accuracy and robustness of this method while reducing false detections. Before clustering, features are normalized to zero mean and unit standard deviation to prevent a feature from dominating the other ones.

C. Text region clustering

We use a reduced K-means clustering algorithm to cluster features vectors [12]. In order to reduce computational time, we apply the standard K-means clustering to a reduced number of pixels and a minimum distance classification is used to categorize all surrounding non-clustered pixels. Empirically, the number of clusters (value of K) was set to three, value that works well with all test images. The cluster whose center is closest to the origin of features vector space is labeled as background while the furthest one is labeled as text. Text boxes rotation is applied after the estimation of document skew. The angle is estimated due to the shape and the centroids of all text boxes. The final stage of text detection module is a validation module that confirms text boxes. It tries to identify false text boxes by using heuristic rules about aspect ratio, global intensity indicators, etc.

We have applied text detection module on a set of 50 test images where there are one or two text areas per image. Table below summarizes detection results.

Detection errors occur mostly when an image contains several text zones with important differences in character size or text orientation. This is due to the fact that our clustering scheme considers text areas as one homogeneous class per image. Only a truly multi-resolution approach can reduce drastically this problem.

	Text zone detected	False detections
Results	57/62	11

TABLE I
TEXT DETECTION RESULTS

III. CHARACTER SEGMENTATION AND RECOGNITION

Character segmentation and recognition has been performed for several decades, especially typewritten characters from scanner. Commercial OCR softwares perform well on ‘clean’ documents or need user to select a kind of documents, for example forms or letters. The challenge is at different levels of character processing: first, document is degraded by taking a picture with a low-resolution camera, then it is free-font, free-size and can contain forms, complex backgrounds for example.

We tested resolution on several images and it is about 60 dpi for a 40-cm distance. For information, commercial OCRs need 300 dpi to recognize characters.

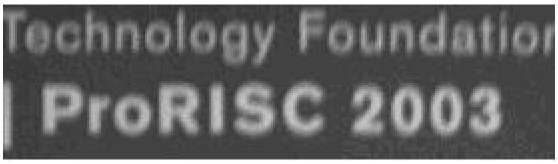

Fig. 5. Sample after text detection

Our database is built with 43 documents with complex backgrounds and on 19 other documents taken by a low-resolution camera. Several steps need to be performed such as binarization, characters segmentation and recognition.

A. Binarization

Until this step, text boxes are located and deskewed for a better segmentation and recognition. For pictures with low-contrast, a contrast enhancement with a top-hat and bottom-hat filtering is done first. This operation reduces the blur part around characters in order to enhance the contrast with the background.

About thresholding algorithms, many researches [18] have been done to evaluate all binarization methods for document images and the main conclusion is that local thresholding is better than global thresholding especially for partial degradations such as uneven illumination. But when text is already located, it becomes global information of the picture and it is also possible to apply independant global thresholds for each text region of a same document. Others papers [19], [20] appeared and are still appearing on this subject for degraded documents. Adaptive thresholding is mostly used to reduce the degradation effect such as uneven lighting or salt and pepper noise. But in our general context, some work well for some pictures but really bad for others ones. It is quite difficult to find a thresholding algorithm which works better for all pictures.

The Otsu [21] method chooses a global threshold to minimize the intraclass variance of the thresholded black and white pixels. Nevertheless, the main information is the gray level value of characters. For our algorithm, we assume that characters are in the same color, therefore almost in the same gray level. The method is to choose the mean characters gray level value as a global threshold.

First, an Otsu thresholding is performed, followed by a skeletonization of “assumed-characters”. In order to pick only characters gray level value, end points of lines and small objects are removed. An average of gray level values of skeleton is computed. Also, the global threshold is chosen as 85 % of this mean to take in account a color gradation. This global thresholding is strict and not all characters pixels are well binarized but with some further post-processings as filling holes, the result is better than Otsu thresholding for complex background pictures or strongly degraded documents.

Degradations are weaker and following steps are better in those cases. Actually, an automatic first discrimination between kinds of pictures is done to apply our algorithm only in complex background pictures, which could be advertisements, magazines pages and have a lower density text than letters have for example. With the first Otsu thresholding, the number of connected components N_{cc} is known and based on the picture size ($X * Y$), a density text value D is calculated as $D = N_{cc} * Y/X$. It is useful to apply only Otsu method on some pictures or a combination of both on some other pictures.

B. Character segmentation

In order to segment text into lines, words and characters, the document needs to have text only. Figures are already removed by previous steps but underlinings left.

From the RLE (Run-Length Encoding) method, following black pixels are detected. With the previous connected components step, some statistics can be computed such as

Fig. 6. From top to bottom: original RGB image, Otsu thresholding and our thresholding algorithm

mean height or mean width in order to choose a satisfactory threshold to decide to remove following black pixels or not.

With those statistics and the bounding box of the document, the number of lines is found. To segment documents into lines, a horizontal histogram could be sufficient but with degraded or slightly skewed ones, it becomes insufficient because a threshold has to be chosen to find start and end lines.

Therefore all y-coordinates of characters centroids are clustered with a vector quantization using K-means algorithm, K being the number of lines.

This technique works pretty well and on our database of 62 documents, there were 2 bad segmentations because of a wrong number of lines. As it could be difficult to know precisely the number of lines, some techniques to estimate it will be studied later.

Because of strong degradations, many characters are broken in several parts or touched each other. To have a good segmentation, it is really important to fix some of these troubles before the recognition step [22]. Thanks to the mean characters width and the line segmentation, all overlapping parts are grouped to be only one character. With a satisfactory threshold of overlapping distance, italic characters are not merged.

On the other hand, a few touching characters are cut

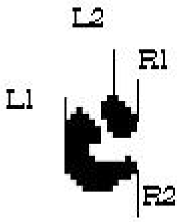

Fig. 7. Superposition of two parts of character

with the caliper distance. A caliper histogram is formed plotting the distance between the uppermost and bottom-most pixels in each column and a weak weight is applied for minima in strategic positions (which is the middle for two assumed characters or one third and two thirds for three assumed characters) and a strong weight for the borders of characters.

Characters with a ratio height/width inferior to 0.75 are considered to be more than one character and the caliper distance is computed to find right cut places.

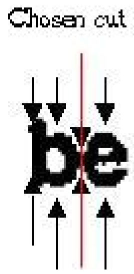

Fig. 8. Caliper algorithm: the chosen cut between ‘b’ and ‘e’

The threshold works pretty well but some touching “thin” characters such as ‘ri’ are not cut and some ‘m’ or ‘w’ could be cut. But this step is primordial and even if it adds a few errors, the global recognition rate increases as shown in the results table next section.

Characters statistics can also be updated after this merge and split operation. With these recent computations, mistaken characters are removed because of their shape.

The inter-characters distance is calculated to segment lines into words. This information is really important for the speech synthesis part. Actually a natural linguistic parser precedes the speech synthesis in order to identify words for a right pronunciation.

Ex: in French, the phonetic transcription can be different:

“ les tas ” $\Rightarrow$ [l e t a]
“ lestas ” $\Rightarrow$ [l ϵ s t a]

Therefore a good word segmenter is really useful.
The inter-word distance IWD is defined as

$$IWD > std(ICD) + mean(ICD)$$

with ICD, the inter-character distance and std (ICD) and mean(ICD), respectively its standard deviation and its mean.

With our database of 135 words:

	Complex backgrounds	Clean backgrounds
Rate	35/44	82/91

TABLE II
WORD SEGMENTATION RATE

Finally, character segmentation is performed inside words in order to get new indices of characters.

C. Character recognition

Most algorithms try to skeletonize characters to free from different fonts and noise [23]. On the contrary, in our algorithm, to homogenize degraded characters, different pre-processings are applied to make characters thicker in order to smooth their edges. Actually, our character recognizer is especially based on edges. Pre-processing steps are:

- to fill isolated white pixels inside black 8-connected pixels
- to connect components grouped during the merge step in a 8-connected neighborhood
- to smooth the edge by thicken components
- to normalize characters in a 16*16 pixels bounding box

A multi-layer perceptron neural network [24] is used with about 180 nodes in the unique hidden layer.

According to [25], the training database has to be at least ten times larger than the feature vector size for each class. Therefore a corpus of 28140 characters taken in different conditions with a low-resolution camera was constituted. The features vector is based on the edges of characters and a probe is sent in each direction. Moreover to get the information of holes like in ‘B’, some interior probes are sent from the center.

Errors are counted according to the Levenshtein distance, which computes an alignment that minimizes the number of insertions, deletions and substitutions when comparing two different words with units costs for all operations. For commercial OCRs, several ones were tested and the rate mentioned below is an average of all results.

Commercial OCRs	Otsu	Otsu + caliper	Our thresholding + caliper
36.2%	55.7%	66.0%	69.7%

TABLE III
CHARACTER RECOGNITION RATE

Our thresholding algorithm works better than Otsu thresholding for our database but a validation with a larger database is required.

IV. Text to Speech

Results in the table above show that a correction is useful in a future work to reduce the error rate. A comparison of many techniques was written by [26] and with some of these methods and new ones, we would try to correct first, word segmentation and then character recognition based on a dictionary and N-grams method especially.

Actually, on a given character error rate, the word error rate is higher and finally the speech synthesis quality is pretty bad.

A Text-To-Speech (TTS) synthesizer is a computer-based system that should be able to read any text aloud. In this definition, TTS means automatic production of speech, through a grapheme-to-phoneme transcription of the sentences to utter. For performing the grapheme-to-phoneme transcription, the TTS synthesizer involves a Natural Language Processing module that analyzes the text. The transcription is then processed by a Digital Signal Processing module, which generates the corresponding speech signal.

The used TTS synthesizer is eLite [1] (Multitel ASBL [27]), a multilingual research platform which easily deals with important linguistic issues: complex units detection (phone numbers, URLs), trustworthy syntactic disambiguation, contextual phonetization and acronyms spelling. Moreover, important researches, still in progress, will be integrated within eLite, like page layout detection, spelling correction or non-uniform units-based speech synthesis.

[1] eLite stands for Enhanced, LInguistically-based TExt-to-speech synthesizer.

V. Conclusions and future work

We have developed a system able to automatically identify and recognize text zones in images taken from a camera. It performs well for a wide range of document images and no prior knowledge concerning document layout, character size, type, color and orientation has been used.

A new thresholding algorithm has been proposed and a discrimination between kind of documents enables to apply this new method on corresponding documents, such as strongly degraded ones. Segmentation and recognition steps aim at considering degraded characters with touched and broken ones. Preliminary results are encouraging but a larger database is necessary to confirm all results.

Future improvements of the text detection consist in modifying our approach to a real multi-resolution system by applying the same algorithm to different instances of the image at different resolutions. An expansion to text detection embedded into natural scene is currently under investigation. The system will require new text localisation indicators such as the use of color information in order to deal with more complex images.

Then, added to the recognition errors correction, future works are to reduce the heuristic part of this system. A model could be created depending on some kinds of documents or types of degradations to improve the recognition rate drastically [28]. For the time being, it is unrealistic to create a generic recognition system that reaches significant results for all kind of text images.

VI. Acknowledgements

We want to thank the TTS department of Multitel, who provided the eLite speech synthesizer for our research. This project is called Sypole and is funded by Ministère de la Région wallonne in Belgium.

References

[1] R.Lienhart, A.Wernicke, *Localizing and segmenting text in images , videos and web pages*, IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, n°4, pp 256-268, 2002

[2] Doermann, *Progress in camera-based document image analysis*, 7^{th} ICDAR Conference, 2003

[3] X.Fernandez Hermida and al., *An OCR for vehicule license plates*, Proceedings of ICSPAT 97, 1997

[4] J. Gao and J. Yang, *An Adaptive Algorithm for Text Detection from Natural Scenes*, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2001

[5] J. Ohya and al., *Recognizing characters in scene images*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, n°2, 1994

[6] K. Wang and J.A. Kangas, *Character location in scene images from digital camera*, Pattern recognition, vol. 36, pp 2287-2299, 2003

[7] P. Clark and M. Mirmehdi, *Recognizing text in real scenes*, International Journal on Document Analysis and Recognition, vol.4, pp 243-257, 2002

[8] A.K. Jain and B. Yu, *Automatic Text location in images and video frames*, Pattern Recognition, vol. 31, n°12, pp 2055-2076, 1995

[9] M. Pietikäinen and O. Okun, *Text extraction from grey scale page images by simple edge detectors*, Proceedings of the 12^{th} Scandinavian Conference on Image Analysis, pp 628-635, 2001

[10] W.-Y. Chen and S.-Y. Chen, *Adaptative page segmentation for color technical journals' cover images*, Image and Vision Computing, vol. 16, pp 855-877, 1998

[11] Y. Zhong, K. Karuand A.K. Jain, *Locating text in complex color images*, Pattern Recognition, vol. 28, n° 10, pp 1523-1535, 1995

[12] V. Wu and al., *Textfinder: an automatic system to detect and recognize text in images*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, n° 11, pp 1224-1229, 1999

[13] H. Li, D. Doermann, *Automatic text detection and tracking in digital video*, IEEE Transactions on Image Processing, vol. 9, n°1, pp 147-156, 2000

[14] A.K. Jain, S. Bhattacharjee, *Text segmentation using Gabor filters for automatic document processing*, Machine Vision and Applications, vol. 5, pp 169-184, 1992

[15] A.C. Bovic, M. Clark and W.S. Geisler, *Multichannel texture analysis using localized spatial filters*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, n°1, pp 55-73, 1990

[16] T.N. Tan and A.G. Constantinides, *Texture Analysis based on a Human Visual Model*, Proc. IEEE Int. Conf. On Acoust., Speech, Signal Proc., pp 2091-2110, Albuquerque, New Mexico, 1990

[17] D. Dunn and al., *Texture Segmentation Using 2-D Gabor Elementary Functions*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, n°2, 1994

[18] Trier, Taxt, *Evaluation of binarization methods for document images*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994

[19] Huiping Li et al., *Text enhancement in digital video*, Proceedings of CIKM, 8^{th} ACM International Conference on Information and Knowledge Management, 1999

[20] Graham Leedham and al., *Comparison of some thresholding algorithms for text/background segmentation in difficult document images*, 7^{th} ICDAR Conference, 2003

[21] N.Otsu, *A thresholding selection method from gray-level histogram*, IEEE Transactions on Systems, Man, and Cybernetics 9, pp 62-66, 1979

[22] R.G.Casey, E. Lecolinet, *A survey of methods and strategies in character segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, n°7, 1996

[23] Trier, Jain and Taxt, *Feature extraction methods for character recognition, a survey*, Pattern Recognition, vol. 29, n°4, pp 641-662, 1996

[24] B.Gosselin, *Applications de réseaux de neurones artificiels à la reconnaissance automatique de caractères manuscrits*, PhD thesis, Faculté Polytechnique de Mons, 1996

[25] I.Guyon, *A scaling law for the validation-set training-set size ratio*, AT&T Bell Laboratories, 1997

[26] K.Kukich, *Techniques for automatically correcting words in text*, ACM Computing Surveys, vol. 24, n°4, 1992

[27] Multitel website, `http:\\www.multitel.be\tts\elite`

[28] A.Souza and al., *Automatic filter selection using image quality assessment*, 7^{th} ICDAR Conference, 2003