EURASIP Journal on Applied Signal Processing 2005:13, 2127–2135
© 2005 Hindawi Publishing Corporation

An Embedded Application for Degraded Text Recognition

Céline Thillou
Laboratoire de Théorie des Circuits et Traitement du Signal, Faculté Polytechnique de Mons, Bâtiment Multitel-Initialis, 1 avenue Copernic, 7000 Mons, Belgium
Email: celine.thillou@tcts.fpms.ac.be

Silvio Ferreira
Laboratoire de Théorie des Circuits et Traitement du Signal, Faculté Polytechnique de Mons, Bâtiment Multitel-Initialis, 1 avenue Copernic, 7000 Mons, Belgium
Email: silvio.ferreira@tcts.fpms.ac.be

Bernard Gosselin
Laboratoire de Théorie des Circuits et Traitement du Signal, Faculté Polytechnique de Mons, Bâtiment Multitel-Initialis, 1 avenue Copernic, 7000 Mons, Belgium
Email: bernard.gosselin@tcts.fpms.ac.be

Received 24 December 2003; Revised 30 November 2004

This paper describes a mobile device which tries to give the blind or visually impaired access to text information. Three key technologies are required for this system: text detection, optical character recognition, and speech synthesis. Blind users and the mobile environment imply two strong constraints. First, pictures will be taken without control on camera settings and a priori information 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. We will present the overall description of the system from text detection to OCR error correction.

Keywords and phrases: text detection, thresholding, character recognition, error correction.

1. INTRODUCTION

A broad range of new applications and opportunities are emerging as wireless communication, mobile devices and camera technologies are becoming widely available and acceptable. These mature technologies introduce new research areas. One of the most fascinating frontier projects in the field of artificial intelligence is machine understanding of text.

Extensive efforts have been made in order to give the blind or visually impaired access to text information; two complementary approaches are generally used. The first approach directly adapts 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 these shortfalls. We are trying to design a camera-based system capable to capture characters from the photo image, making recognition, then giving speech output. This innovative system is embedded in a powerful mobile device like a personal digital assistant.

Figure 1 gives an overview of the system and the interactions between each subsystem.

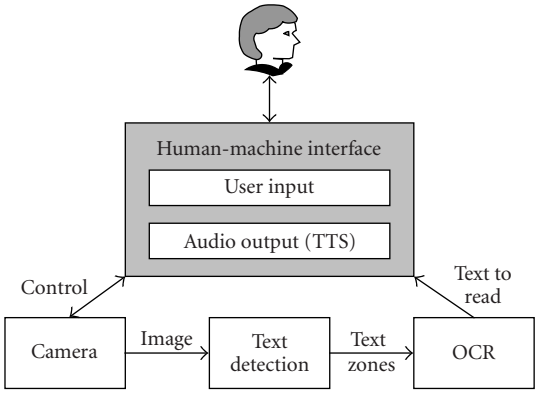

Figure 1: System overview.

Figure 2: Samples of text images from our database.

This paper is organized as follows. Section 2 describes text detection challenges and the approach we follow. Section 3 proposes our character segmentation and recognition algorithms. Section 4 shows the importance of a correction step with degraded images. Finally, Section 5 addresses perspectives in research activities and concludes the paper.

2. TEXT DETECTION

In this section, we address the problem of automatically finding text in images taken by a digital camera. Camera-captured images present a bunch of degradations, missing in scanner-based ones [1], 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 [2].

A text is considered as a scene text when the text is recorded from a part of a scene (e.g., road signs, posters on the street, street names). Unlike document text, characters in scene images originally exist in 3D space, and can therefore be distorted by a slant or a tilt, and by the shape of objects on which they are printed [1]. 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, and so forth. 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.

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 [5, 8, 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.

2.1. Texture segmentation

Our text detection technique is based on a texture segmentation approach. Text in a document is considered as a textured region to isolate; nontext 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.

(i) Characters contrast with their background.

(ii) Text possesses some frequencies and orientation information.

(iii) Text shows spatial cohesion: characters appear in clusters at a regular distance aligned to a virtual line.

We characterize text with Gabor filters which have been used earlier for a variety of texture classification and segmentation tasks [15, 16]. A subset of Gabor filters proposed by Jain and Bhattacharjee [14] is used. Feature images are then classified into several regions using an unsupervised clustering algorithm. The final step of this approach is to find the representative cluster of text region.

2.2. Text characterization

By processing 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 nontext regions but a collection 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. In spatial domain, the two-dimensional Gabor filter $h(x, y)$ is given by

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

where σ_x and σ_y 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 2D Gabor filter is rewritten in polar coordinates as

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

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.

(i) It has been shown to be optimal in the sense of minimizing the joint two-dimensional uncertainty in space and frequency [15].

(ii) Gabor filters closely resemble the mechanism of multichannel representation of the retinal images in biological visual system [17].

(iii) Gabor filters can extract features in the presence of additive noise.

(iv) Gabor filters have bandpass nature, which is essential in analyzing a textured image.

A more detailed description of Gabor filters is given in [18]. A magnitude operation is required after each Gabor filtering. Indeed, to simulate human texture perception, some form of nonlinearity is desirable [16]. Nonlinearity is introduced in each filtered image by applying the following transformation to each pixel:

$$\Psi(t) = \tanh(\alpha t) = \frac{1 - \exp^{-2\alpha t}}{1 + \exp^{-2\alpha t}}. \quad (3)$$

For $\alpha = 0.25$, 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 window centered at each pixel. Before clustering, features are normalized to prevent a feature from dominating the other ones. In Figure 3, an example of the whole text characterization is shown.

2.3. Text region clustering

We use an adapted K-means clustering algorithm to cluster feature 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 nonclustered 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 feature vector space is labelled as background while the furthest one is labelled as text. Text boxes rotation is applied after the estimation of document skew. The angle is estimated from 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 identifies false text boxes by using heuristic rules about aspect ratio, global intensity indicators, and so forth.

We have applied text detection module on a set of 100 test images where there are one or two text areas per image. A text zone is correct when all the lines of the text area are included. A partly detected text zone is considered as an error. On the contrary, a false detection occurs when the detected zone does not contain any text information. Table 1 summarizes detection results.

Figure 4 illustrates the results of sample images of Figure 2. 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 multiresolution approach can drastically reduce this problem.

3. CHARACTER SEGMENTATION AND RECOGNITION

Character segmentation and recognition have 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.

Typically, commercial OCRs need 300 dpi to recognize characters. Our system was tested on images taken from a 40 cm distance, with 60 dpi resolution. For information, Figure 5 shows an example of the quality of our database.

Our database is built with more than one hundred documents, some with complex backgrounds. Several steps need to be performed such as binarization, character segmentation and recognition.

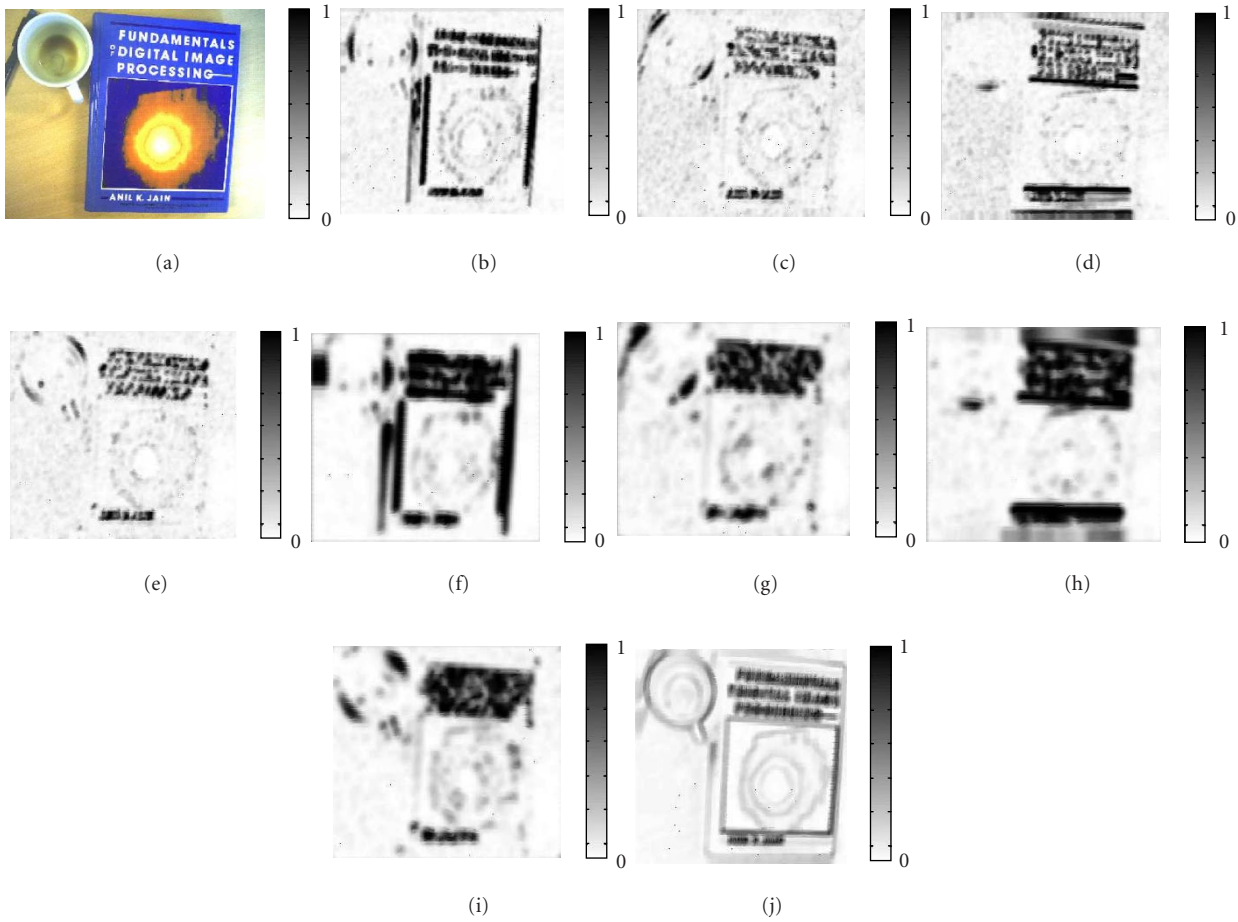

Figure 3: Feature images. (a) Original images. (b)–(i) Gabor filters: (b) $u_0 = \sqrt{2}/4\Theta = 0°$; (c) $u_0 = \sqrt{2}/4\Theta = 45°$; (d) $u_0 = \sqrt{2}/4\Theta = 90°$; (e) $u_0 = \sqrt{2}/4\Theta = 135°$; (f) $u_0 = \sqrt{2}/8\Theta = 0°$; (g) $u_0 = \sqrt{2}/8\Theta = 45°$; (h) $u_0 = \sqrt{2}/8\Theta = 90°$; (i) $u_0 = \sqrt{2}/8\Theta = 135°$. (j) Edge density measure.

Table 1: Text detection results.

Candidates	Text zone detected	False detections
Results	77/112	24

3.1. 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 [19] 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. Nevertheless when text is already located, it becomes global information of the picture and it is possible to apply independent global thresholds for each text region of a same document. Other papers [20, 21] appeared and are still appearing on this subject for degraded documents. Adaptive thresholding is mostly used to reduce degradation effects such as uneven lighting or salt and pepper noise. In our general context, some work well for some pictures but really bad for other ones. It is quite difficult to find a thresholding algorithm which is tuned to all pictures.

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 aims at choosing the mean character 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 character gray level values, end points of lines and small objects are removed. An average of intensities of skeleton is computed. Also, the global threshold is chosen as 85% of this mean to take into account a color gradation. This global thresholding is strict and not all characters pixels are well binarized but with some further postprocessings as filling holes, the result is better than Otsu thresholding for complex background pictures or strongly degraded documents, as illustrated in Figure 6.

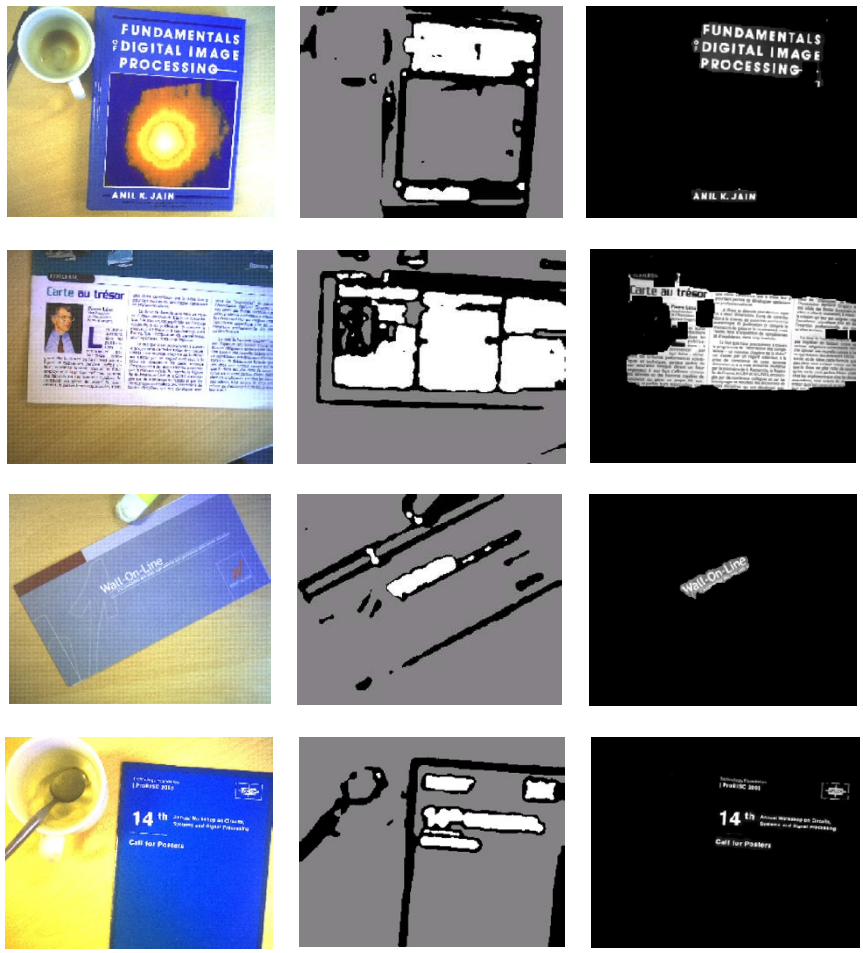

Figure 4: Text detection results. (Left column) Original images. (Middle column) Text region clustering. (Right column) Final results.

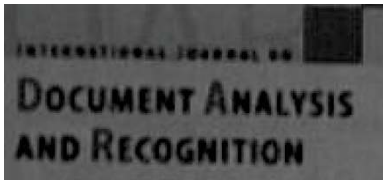

Figure 5: Sample after text detection.

In order to apply this improvement with the skeleton only on complex backgrounds, an automatic first discrimination between various types of pictures is done. These pictures, with strong degradations, are typically the scene pictures, such as advertisements, magazine pages. They 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 $D = N_{cc} * Y/X$. The improvement combination is applied only on pictures with a low density text and the Otsu method on the other ones.

This technique is general and works well in all pictures in our database but the most important thing is that the processing is a morphological one and does not need many resources in our embedded platform. We could use some other filters or complex techniques such as [22] but it will be a disadvantage for our embedded constraints.

3.2. Character segmentation

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

We use different steps such as lines removal with a RLE method, segmentation into lines with a K-means clustering, segmentation into words with rules and segmentation into characters with a connected components algorithm, see [18] for more details.

Two important problems in character segmentation still need to be addressed. Because of strong degradations, many characters are broken in several parts or touching each other. To have a good segmentation, it is really important to address some of these problems before the recognition step.

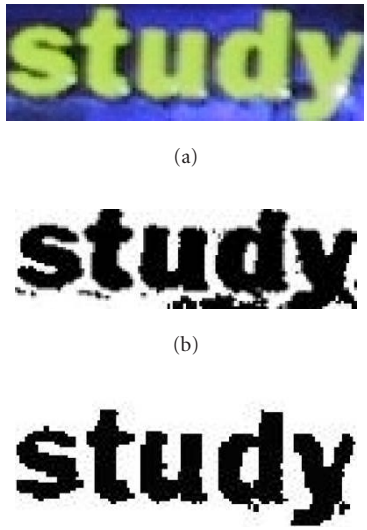

Figure 6: (a) Original RGB image, (b) Otsu thresholding, and (c) our thresholding algorithm.

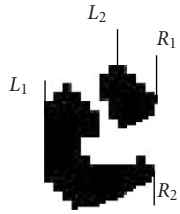

Figure 7: Superposition of two parts of character.

Thanks to the mean characters width and the line segmentation, all overlapping parts are grouped to be only one character, like for the letter "e" in Figure 7. With a satisfactory threshold for considering overlapping distance, italic characters remain unconnected despite of their slant which can make overlap some parts between components.

On the other hand, a few touching characters are cut with the caliper distance. A caliper histogram is formed plotting the distance between the uppermost and bottommost 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, as shown in Figure 8.

The threshold works pretty well but some touching "thin" characters such as "ri" are not cut and some "m" or "w" could be cut. Nevertheless this step is primordial and even if it adds a few errors, the global recognition rate increases as shown in Table 2.

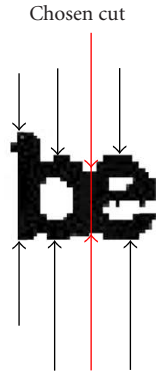

Figure 8: Caliper algorithm: the chosen cut between "b" and "e."

Table 2: Character recognition rate.

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

'

Another approach is still in progress to avoid to take decisions at this segmentation step. It is important to consider all different character segmentation to make less errors at this step and to have a right validation with the following recognition step.

3.3. 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 preprocessings are applied to make characters thicker in order to smooth their edges. This is quite important because our character recognizer is especially based on edges. Preprocessing steps are

(i) to fill isolated white pixels surrounded by eight black neighbors,
(ii) to connect components grouped during the merge step in an 8-connected neighborhood,
(iii) to smooth edge by thickening components,
(iv) to normalize characters in a $16 * 16$ pixels bounding box.

A multilayer 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 28 140 characters taken in different conditions with a low-resolution camera was constituted. The feature vector is based on the edges of characters and a probe is sent in each direction to get locations of edges. Moreover to get the information of holes like in "*B*," some interior probes are sent from the center as illustrated in Figure 9.

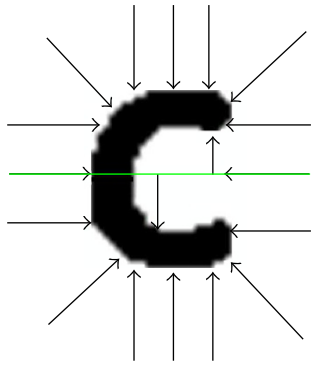

Figure 9: Probes sent for the letter "C" to get location of edges.

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 unit costs for all operations. For commercial OCRs, several well-known ones (ABBYY FineReader 7.0, ScanSoft OmniPage Limited Edition, and Readiris Pro 9) were tested and the rate mentioned in Table 2 is an average of all results.

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

4. RECOGNITION ERROR CORRECTION

A comparison of many techniques was described in [26] about high-level correction, such as using a dictionary or syntactic information. With our error rate, this high-level information can degrade results even more.

Therefore, we consider here a low-level error correction. We take into account confidence levels of OCR output in order to choose the right character in the N-best list instead of always considering the best recognized one.

The main question is how to find "house" in Table 3.

This error correction is done before applying high-level error correction. Therefore, this step has to be fast and not to require too many resources because of our constraint of embedded platform. No dictionary is used but character n-grams of a particular language.

Several steps are required.

(i) Our database for n-gram probabilities is built with more than one hundred million characters of a French newspaper. With maximum likelihood estimation, probabilities are computed by

$$P\left(c_i \middle| c_{i-n+1}^{i-1}\right) = \frac{\left| c_{i-n+1}^{i-1} c_i \right|}{\sum_c \left| c_{i-n+1}^{i-1} c \right|} \tag{4}$$

with $c_{i=1,\ldots,37}$ for 37 classes: 26 lower-case non-accented letters, 10 digits, and space between words.

(ii) For unknown probabilities (n-gram absent of the corpus), we use the Katz [27] model smoothing technique which

Table 3: Example of a 3-best list for the word "house."

1st	H	A	U	R	E
2nd	8	O	V	T	A
3rd	R	E	1	S	O

Table 4: Character recognition rate after the first correction.

Tests	3-best/trigram	2-best/trigram	2-best/bigram
Rate	64.7%	67.5%	72.1%

is as follows for trigrams:

$$P(c_i|c_{i-2}c_{i-1}) = \begin{cases} P(c_i|c_{i-2}c_{i-1}) & \text{if } \left|c_{i-2}c_{i-1}c_i\right| > 0, \\ \alpha P(c_i|c_{i-1}) & \text{otherwise.} \end{cases} \tag{5}$$

Probability mass is distributed to models with an inferior order.

(iii) In order to find the best path to get the right word, we use the Viterbi algorithm which computes maximal probabilities by iteration.

An experimental rate is applied to favor the best characters of OCR output. We choose a rate 30% larger than those applied to the other characters in the N-best list.

We could directly use OCR probabilities but they were very correlated to the character and its degradation. To sum up, we take from OCR output the order of characters and the information of the best ones only.

Several tests were made dependant of n in n-grams to consider and results are given in Table 4.

With a 3-best list and trigram information, the result is worse than before the error correction step. It is mainly due to the fact that some errors come from the segmentation step. Therefore, if some characters are badly segmented, the number of characters in a word is false and linguistic information is not useful.

With a 2-best list and trigram information, the recognition rate is better than with a 3-best list but worse than before using this step. Between the 3-best list and the 2-best list for our database, we lose 1.5% of right characters for error correction.

We can conclude that less confusion was introduced and even if we lose useful information (the N-best list is shorter), correction performs better.

It is the same principle for 2-best list and bigram information. The confusion is a significant parameter in the case of no dictionary use. This step increases our rate of about 2.5%. However, the result is higher in words with a few errors (from 0 to 20% inside) than in words with more errors.

Therefore, this step will be more useful after other global improvements in the general algorithm.

The following steps aim at sharpening correction results based on a dictionary and high-level information. This kind of error correction is included into the natural language processing part of the synthesizer eLite [28] we use.

This final step, which returns an audio answer to blind or visually impaired people, is essential. Low-vision people

have difficulties to interact with small devices. For input, a dedicated keyboard can be used but for output, two methods can only be taken into account : the prerecorded messages and the speech synthesis. The latter one seems to be more relevant because it offers the freedom of use and text files are much smaller than audio files containing the same information.

Thanks to the Creth [29], a center of new technologies for handicapped people in Belgium, we could get some feedbacks from users concerning the whole interface and more particularly the audio output. Opinions are really satisfying because speech synthesis is of high quality and makes users neither tired nor bored.

The eLite TTS synthesizer is 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 or nonuniform units-based speech synthesis.

5. CONCLUSIONS AND FUTURE WORK

We have built a complete text recognition system to be used on a mobile environment by blind or visually impaired users. This innovative application is 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 kinds 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 among with touching and broken ones.

A large study on low-level OCR error correction was presented. Results are already promising but this step will have a larger impact once character recognition rate will be higher thanks to some improvements, such as in character segmentation.

Future work in text detection consists in modifying our approach to a real multiresolution system by applying the same algorithm to different instances of the image at different resolutions. An expansion to text detection embedded into natural scenes is currently under investigation. A model could be created depending on some kinds of documents or types of degradations to improve the recognition rate drastically. For the time being, it is unrealistic to create a generic recognition system that reaches significant results for all kinds of text images.

ACKNOWLEDGMENTS

We want to thank the TTS Department of Multitel, which 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] D. Doermann, J. Liang, and H. Li, "Progress in camera-based document image analysis," in *Proc. 7th IEEE International Conference on Document Analysis and Recognition (ICDAR '03)*, vol. 1, pp. 606–617, August 2003.

[2] R. Lienhart and A. Wernicke, "Localizing and segmenting text in images, videos and web pages," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 4, pp. 256–268, 2002.

[3] X. Fernandez Hermida, F. Martin Rodriguez, J. L. Fernandez Lijo, F. Pita Sande, and M. Perez Iglesias, "An OCR for vehicle license plates," in *Proc. International Conference on Signal Processing Applications & Technology (ICSPAT '97)*, San Diego, Calif, USA, September 1997.

[4] J. Gao and J. Yang, "An adaptive algorithm for text detection from natural scenes," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 2, pp. 84–89, Kauai, Hawaii, USA, 2001.

[5] J. Ohya, A. Shio, and S. Akamatsu, "Recognizing characters in scene images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 2, pp. 214–220, 1994.

[6] K. Wang and J. A. Kangas, "Character location in scene images from digital camera," *Pattern Recognition*, vol. 36, no. 10, pp. 2287–2299, 2003.

[7] P. Clark and M. Mirmehdi, "Recognizing text in real scenes," *International Journal on Document Analysis and Recognition*, vol. 4, no. 4, pp. 243–257, 2002.

[8] A. K. Jain and B. Yu, "Automatic text location in images and video frames," *Pattern Recognition*, vol. 31, no. 12, pp. 2055–2076, 1995.

[9] M. Pietikäinen and O. Okun, "Text extraction from grey scale page images by simple edge detectors," in *Proc. 12th Scandinavian Conference on Image Analysis*, pp. 628–635, Bergen, Norway, June 2001.

[10] W.-Y. Chen and S.-Y. Chen, "Adaptive page segmentation for color technical journals' cover images," *Image and Vision Computing*, vol. 16, no. 12–13, pp. 855–877, 1998.

[11] Y. Zhong, K. Karuand, and A. K. Jain, "Locating text in complex color images," *Pattern Recognition*, vol. 28 , no. 10, pp. 1523–1535, 1995.

[12] V. Wu, R. Manmatha, and E. M. Riseman, "Textfinder: an automatic system to detect and recognize text in images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 11, pp. 1224–1229, 1999.

[13] H. Li and D. Doermann, "Automatic text detection and tracking in digital video," *IEEE Trans. Image Processing*, vol. 9, no. 1, pp. 147–156, 2000.

[14] A. K. Jain and S. Bhattacharjee, "Text segmentation using Gabor filters for automatic document processing," *Machine Vision and Applications*, vol. 5, no. 5, pp. 169–184, 1992.

[15] A. C. Bovic, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 1, pp. 55–73, 1990.

[16] D. Dunn, W. E. Higgins, and J. Wakeley, "Texture segmentation using 2-D Gabor elementary functions," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 2, pp. 130–149, 1994.

[17] T. N. Tan and A. G. Constantinides, "Texture analysis based on a human visual model," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP '90)*, pp. 2091–2110, Albuquerque, NM, USA, April 1990.

[18] S. Ferreira, C. Thillou, and B. Gosselin, "From picture to speech: an innovative application for embedded environment," in *Proc. 14th Annual Workshop on Circuits, Systems and Signal Processing (ProRISC '03)*, Veldhoven, the Netherlands, November 2003.

[19] Ø. D. Trier and T. Taxt, "Evaluation of binarization methods for document images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 3, pp. 312–315, 1995.

[20] H. Li and D. S. Doermann, "Text enhancement in digital video using multiple frame integration," *ACM Multimedia*, vol. 1, pp. 19–22, 1999.

[21] G. Leedham, C. Yan, K. Takru, J. H. N. Tan, and L. Mian, "Comparison of some thresholding algorithms for text/background segmentation in difficult document images," in *Proc. 7th IEEE International Conference on Document Analysis and Recognition (ICDAR '03)*, pp. 859–864, Edinburgh, Scotland, August 2003.

[22] Z. Ping and C. Lihui, "Document filters using morphological and geometrical features of characters," *Image and Vision Computing*, vol. 19, no. 12, pp. 847–855, 2001.

[23] Ø. D. Trier, A. K. Jain, and T. Taxt, "Feature extraction methods for character recognition—A survey," *Pattern Recognition*, vol. 29, no. 4, pp. 641–662, 1996.

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

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

[26] K. Kukich, "Techniques for automatically correcting words in text," *ACM Computing Surveys*, vol. 24, no. 4, pp. 377–439, 1992.

[27] S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 35 , no. 3, pp. 400–401, 1987.

[28] Multitel website, "http://www.multitel.be/tts," visited in July 2004.

[29] Creth website, "http://www.creth.be/," visited in July 2004.

Céline Thillou received her M.S. degree in audiovisual systems and networks engineering from École Supérieure en Informatique et Génie des Télécommunications, France, in 2002, and another M.S. degree in applied sciences from the Faculté Polytechnique de Mons, Belgium, in 2004, both with the highest honors. She is currently a Ph.D. Candidate in applied sciences at the Faculté Polytechnique de Mons, Belgium. Previously, she worked in US in data warehousing with real-time computation in Thomson. In 2002, she worked on image processing, and more specially on augmented reality in Dassault Systèmes. Since 2003, she has been a researcher at the Faculté Polytechnique de Mons and her research deals with color text segmentation and character recognition in degraded images, especially camera-based images.

Silvio Ferreira received his M.S. degree in multimedia applications from the Faculté Polytechnique de Mons, Belgium, in 2002. Concurrently, the Top Industrial Manager for Europe Program (TIME) allowed him also to obtain an M.S. degree in electrical engineering from the École Supérieure d'Electricité de Paris, France. He is currently pursuing the Ph.D. degree at the Faculté Polytechnique de Mons. In 2002, he joined Multitel Research Center, Belgium, where he was an R&D engineer. Since 2003, he has been a researcher at the Faculté Polytechnique de Mons. His research interests are in image processing and multimedia applications. His recent work has included image segmentation and data fusion.

Bernard Gosselin received his M.S. degree in electrical engineering and his Ph.D. degree in applied sciences both from the Faculty of Engineering, Mons, Belgium, in 1990 and 1996, respectively. In 1990, he joined the Signal Processing Department, the Faculty of Engineering, Mons, where he was a Research Associate. In 2004, he became a Professor at the Faculty of Engineering, Mons. In 1999, he became a Visiting Professor at the University of Orleans, France. Since 2001, he has headed the Image Processing Research Group, the Faculty of Engineering, Mons. Professor Gosselin is also a Research Manager in Multitel Research Center, where his responsibilities cover image processing activities. He is a Member of Program Committee of IEEE ProRISC Workshop. His research interests include character recognition, medical imaging, and pattern recognition.