

# Mobile Reading Assistant for Blind People

Vincent Gaudissart, Silvio Ferreira, Céline Thillou, Bernard Gosselin

Faculté Polytechnique de Mons, Laboratoire de Théorie des Circuits et Traitement du Signal  
Bâtiment Multitel – Parc Initialis, 1, avenue Copernic, 7000, Mons, Belgium

Phone: +32 (0)65 374761 Fax: +32 (0)65 374729

*vincent.gaudissart , silvio.ferreira , celine.thillou @tcts.fpms.ac.be*

## Abstract

This paper describes an embedded device dedicated for blind or visually impaired people. The main aim of this system is to build an automatic text reading assistant using existing hardware associated with innovative algorithms. A personal digital assistant (PDA) was chosen because it combines small-size, computational resources and low cost price. Three key technologies are necessary: text detection, optical character recognition and speech synthesis. Moreover, to be as efficient as possible, a specific interface was created to answer blind people requests.

## 1. Introduction

Blind or partially sighted people represent 17.5 million people in Europe. For them, many of the routine tasks associated with daily life can be frustrating. Most information, which often exists in a written or imaged form, is not easily accessible to their handicap. Thankfully, electronic helps to lower many of these barriers. By using computing technology for tasks such as reading documents, communicating and looking for information on the Internet, people with disabilities are able to handle a wide range of activities independently.

Several efforts have been made in order to give access to textual information to the blind or the visually impaired. The first approach tries to adapt directly the information medium to the degree of blindness, by using either optical zooming device to expand characters or Braille language. The other method consists in transforming textual information into speech. Some solutions currently exist, such as combining a scanner, a pair of loudspeakers and a computer. OCR (Optical Character Recognition) software aims at converting images from the scanner into text information while TTS (Text to Speech) technologies convert text into a speech signal. These solutions are efficient, but are not perfect. Indeed, those systems are often heavy and cumbersome.

Moreover, textual information is everywhere, not only in the user's living room, and can exist under different forms such as newspapers, books, or text in natural scenes (signs, screen, schedules, etc.). Mobility is one of the important keys for the independence of visually impaired people.

The main challenge of Sypole project is to remedy all those needs by the realization of a prototype device,

which will be portable, autonomous, small-sized and easy to use, especially designed for these people. The idea of this system is to increase their autonomy by using wireless devices anytime and anywhere. To realize this dedicated platform, we have adopted a user-centred design in close relationship with low-vision people.

Figure 1 gives an overview of the system. Users interact with a dedicated human-machine interface, specifically created for their disabilities. The image taken by the embedded camera is sent to the text detection module. When the text zone is established, the OCR module tries to extract the useful information and sends it to the text-to-speech module.

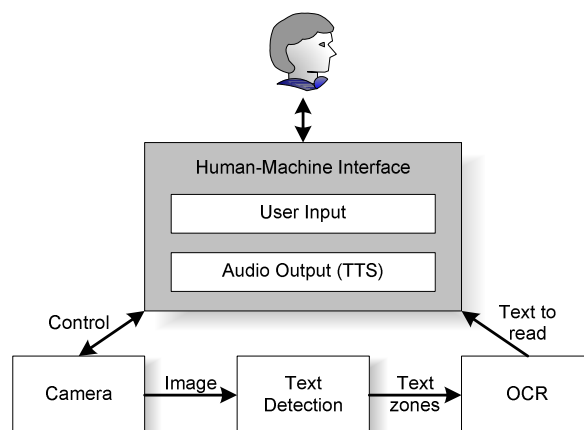


Figure 1. Overview of the System

This paper is organized as followed. Section 2 describes text detection challenges and the approach we have followed. Section 3 proposes our binarisation and recognition algorithms. Section 4 details the choices we made for the human-machine interface. In the final section, we address perspectives in research activities and conclude the paper.

## 2. Text detection

In this section, we address the problem of automatically finding text areas in images taken by a digital camera. The difference between text, figures or background is obvious to a human reader. It is, however, complex to formalize for an automatic system. Actually, camera-captured images present a bunch of degradations, missing in scanner-based ones [1] such as blur, perspective distortion, uneven lighting, moving

objects or sensor noise. In our case, the user's movement can in addition cause unstable input images and the system must be implemented in real time using limited resources. The main design choice is the kind of text occurrences, between scene images and document images [2]. A text is considered as a scene text when the text is recorded outdoor (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 [1].

At the current state of research advancement, we will describe below a text detection algorithm used for indoor document images with nearly uniform backgrounds: paper documents contain figures and text with unknown characters sizes and fonts. Paragraphs can also be disorientated. Illumination variations affect the images with a low incidence due to indoor conditions in comparison with scene situations. Figure 2 illustrates several images of our test database.

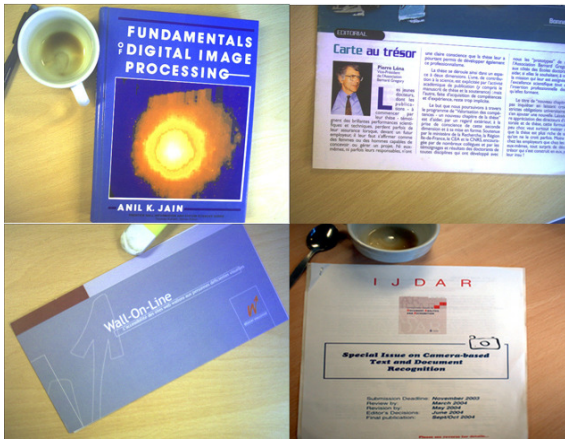


Figure 2. Samples of images from our database

Text detection techniques can broadly be classified as edge [3] [4] [5], colour [6] [7], or texture-based [8] [9] [10].

Edge-based techniques use edges information in order to characterize text areas. They are based on the basic idea that the edges of text symbols are typically stronger than those of noise or background areas. Gradient energy must be firstly highlighted. Filters such as Sobel or Canny detectors are commonly used to emphasize gradient information. These methods operate essentially in greyscale format and do not require much processing time. Nevertheless, they do not cope with more complex text images like pictures of magazines or scene images where edge information alone is not sufficient to separate text from a noisy background.

The use of colour information allows segmenting the image into connected components of uniform colour. A reduction of the colour palette is necessary in order to keep a reasonable number of candidate layers. The remaining candidates must be reduced by using various properties of the image [7]. The main drawbacks of this approach consist of the high colour processing time and

the high sensibility to uneven lighting and sensor noise. But they can be applied to more complex documents in controlled lighting conditions.

Texture-based techniques try to capture certain texture aspects of text. Some features operate at different text scales and are designed to identify text lines, while others measure certain attributes of text paragraphs.

In our approach, the document image consists of several different types of textured regions, one of which results from the text-content in the image. Thus, we pose the problem of locating text in images as a texture discrimination problem. After image characterization, each pixel is represented by a feature vector. The image is 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. The following parts describe each step.

## 2.1. Texture segmentation

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 the scene. Instinctively, text has the following distinguishing properties:

- Characters contrast with their background since artificial text is designed to be read easily
- Text shows spatial cohesion: characters appear in clusters at a regular distance aligned to a virtual line.
- Text possesses specific frequencies and orientation information. Most of the time the orientation of these virtual lines is horizontal since that is the natural writing direction.

Our method for texture characterization is based on Gabor filters which have been used earlier for a variety of texture classification and segmentation tasks [11] [12]. We use a subset of Gabor filters proposed by Jain and Farokhnia [13] associated with an edge density measure. These features are designed to identify text paragraphs. Each individual filter will still confuse text with non-text regions but an association of filters will complement each other and allow identifying text unambiguously.

A two-dimensional Gabor function consists of a sinusoidal plane of particular frequency and orientation modulated by a two-dimensional Gaussian envelope. Physically interpreted, the Gabor transform acts like the Fourier transform but only for a small Gaussian window over the image. In polar coordinates, we can express the orientation selectivity of the two-dimensional Gabor filter:

$$h(x, y, \sigma_x, \sigma_y, w, \theta) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)} + jw(x\cos\theta + y\sin\theta)$$

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 texture region is tuned to the frequency and orientation of the Gabor filter.

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

This configuration of filters parameters is a multi-scale compromise in order to detect non-horizontal text at different font scales. A magnitude operation is required after each Gabor filtering. Indeed, to simulate human texture perception, some form of nonlinearity is desirable [12]. Nonlinearity is introduced in each filtered image by applying the following transformation [10]:

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

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. The feature value is computed from the output of the nonlinear stage as the mean value in a small overlapping window centred at each pixel. **Erreur ! Source du renvoi introuvable.** illustrates an application of a bank of Gabor filters to a document image.

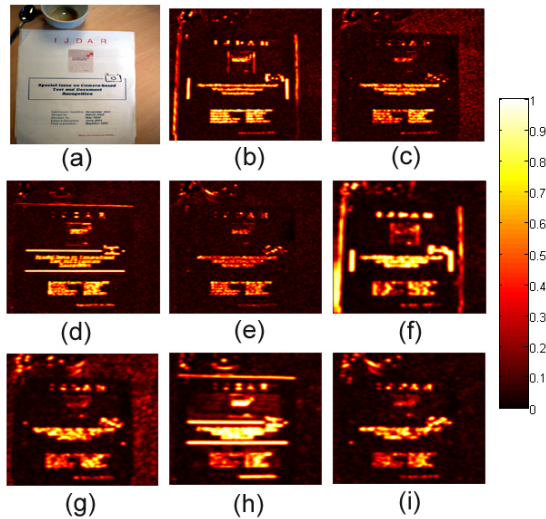


Figure 3. Features images. (a) Original image (b-i) Gabor filters: (b)  $w = 0,35$   $\theta = 0^\circ$ ; (c)  $w = 0,35$   $\theta = 45^\circ$ ; (d)  $w = 0,35$   $\theta = 90^\circ$ ; (e)  $w = 0,35$   $\theta = 135^\circ$ ; (f)  $w = 0,17$   $\theta = 0^\circ$ ; (g)  $w = 0,17$   $\theta = 45^\circ$ ; (h)  $w = 0,17$   $\theta = 90^\circ$ ; (i)  $w = 0,17$   $\theta = 135^\circ$

We associate to our features scheme a partially redundant feature, a local edge density measure based on

Sobel filters [13]. 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.

## 2.2. Text region clustering

We use a reduced K-means clustering algorithm to cluster feature vectors [8]. 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 centre is closest to the origin of feature vector space is labelled as background while the furthest one is labelled as text. Several text detection results are shown on Figure 4.

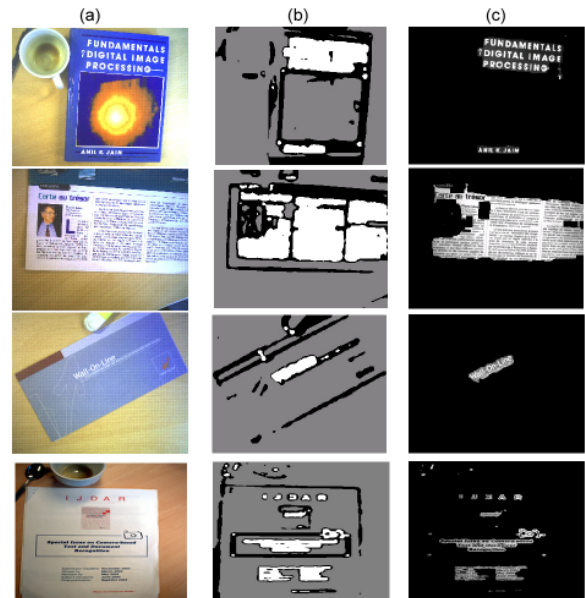


Figure 4. Several text detection results. (a) Original images (b) Text region clustering (c) Final results

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. We have applied text detection module on a set of 100 test images where there are one or two text areas per image. Table 1 summarizes detection results.

	Text zones detected	False detections
Results	77 / 112	24

Table 1. Text 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.

### 3. From binarisation to character 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 the 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, and then it is free-context like scene images and free-layout with a wide diversity of characters.

Our database is built from several documents with complex backgrounds or not, all taken by a low-resolution camera. Some of them are from the conference ICDAR 2003, which organized a robust reading competition to recognize camera-based characters. The resolution on our database is about 80 dpi for a 30-cm distance. For comparison, commercial OCRs need 300 dpi to recognize characters.

Several steps are then explained to perform character recognition such as binarisation, character segmentation, recognition, and error correction.

#### 3.1. Binarisation

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.

Most existing binarisation techniques are thresholding related. Basically, these techniques can be categorized into two categories: global [15] and local or adaptive [16]. Global methods binarise the image with a single threshold. Among the most powerful global techniques, Otsu's algorithm [17] can achieve high performance with simple backgrounds and without parameters to tune. By contrast, local methods change the threshold dynamically over the image according to local information. However in our context, image processing systems need to process a large number of documents with different styles and without pre-specified parameters, which can be a failure for local methods such as the well-known Niblack [18] one. Moreover, all these techniques perform poorly under complex backgrounds.

An important problem for thresholding methods comes from an uneven illumination which introduces noise, especially for “real-world” pictures. This kind of illumination appears as wide noisy areas, so the noise is assumed to have a lower frequency spectrum than the characters one. To correct this “real-world” constraint, we use a wavelet decomposition as described in [19].

The wavelet transform splits the frequency range of the greyscale image into equal blocks and represents the spatial image for each frequency block which gives a multiscale decomposition. We use a level 8 wavelet transform using the Daubechies 16 wavelet [20] and remove low frequency subimages except the lowest one for reconstruction.

The well-known Otsu method is then applied directly on the reconstructed image. This thresholding method is simple, free-parameter and therefore general.

At this point, touching characters, a frequent problem in character segmentation, can already be handled. Then, a skeletonization of “assumed-characters” is applied to pick only characters grey level value. An average of intensities of skeleton on the initial image is computed. Also, the global threshold is chosen as 85% of this mean to take into account a colour gradation. This threshold is applied only on thick characters.

Thick characters, which often become touching characters, are detected thanks to the computation of the ratio  $R$  between the number of pixels of each component and the one of the skeleton. The improvement of this binarisation method can be seen in Figure 5.



Figure 5. Top: the first image is got after Otsu thresholding and the second one after our binarisation. Bottom: the first image is the thresholded reconstructed image, the second one is the result of skeletonization on the image and the third one is the result with the new threshold for thick characters.

Our method can threshold various degraded document images automatically without need of any prior knowledge of the document images nor manual fine-tuning of parameters. It keeps useful information more accurately without broken strokes of the characters and by minimizing the number of touching ones. For more decorative backgrounds like in complex natural scenes, the use of colour information is currently under investigation to complete this thresholding technique.

#### 3.2. Character segmentation

In order to segment text into lines, words and characters, the document needs to be only composed of

text and a top-down method from lines to words is used. As the ratio  $R$  using properties of each connected component is already computed, their mean height is known. An approximate number of lines  $N_L$  is found with the ratio between the mean height and the height of the text area.

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. A tolerance is given for  $N_L$  by considering  $N_L+1$  and  $N_L-1$  and different clusters are computed. All clusters which have parts belonging to others clusters are removed. This line segmentation is robust enough for our database, it induces no errors.

Because of strong degradations, many characters are broken in several parts or touch each other as shown in Figure 6. To have a good segmentation, it is really important to fix some of these troubles before the recognition step [21]. Thanks to the mean character width and the line segmentation, all overlapping parts are grouped to be only one character. With a satisfying threshold of overlapping distance, italic characters are not merged.

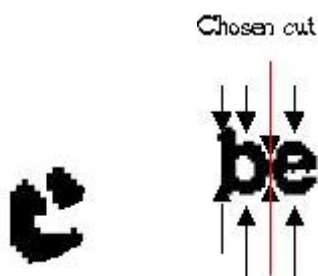


Figure 6. Left: superposition of two parts of the character 'e'. Right: the chosen cut between 'b' and 'e' with the caliper distance.

On the other hand, some touching characters can still be present. The detection of this problem is done thanks to the same neural network used in character recognition with 2-class output (touching or not). They are separated into two or more characters with the caliper distance. A caliper histogram is formed plotting the distance between the uppermost and bottommost pixels in each column. Moreover a weak weight is applied on 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 statistics can also be updated after this merge and split operation. With these recent computations, mistaken characters are removed because of their shape.

Thanks to this better character segmentation, the inter-character distance can be then 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 the grammatical class for a right pronunciation.

The inter-word distance  $IWD$  is defined as :

$$IWD > \text{std}(ICD) + \text{mean}(ICD)$$

with  $ICD$ , the inter-character distance and  $\text{std}(ICD)$  and  $\text{mean}(ICD)$ , respectively its standard deviation and its mean.

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

### 3.3.Character recognition

Most algorithms try to skeletonize characters to free from different fonts and noise [22]. 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.

A multi-layer perceptron neural network [23] is used with about 180 nodes in the unique hidden layer.

According to [24], the training database has to be at least ten times larger than the feature vector size for each class. Therefore a corpus of around 29000 characters taken in different conditions with a low-resolution camera was constituted. The testing database is the database described in previous sections with around 1000 degraded characters.

The feature 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 centre. There are 47 inputs after Principal Component Analysis [22].

For commercial OCRs, several ones were tested and the recognition rate of around 44% is an average of all results. With the overall system, our recognition rate is around 83%. This result still needs some improvements and work is currently under investigation to find invariant-moments for feature extraction.

As character recognition rate is quite low considering camera-based pictures, a strong focus on automatic error correction step is essential to provide a high-quality speech synthesis.

### 3.4.Recognition error correction

A comparison of many techniques was written by [25] and with some of these methods and new ones, we are trying to correct first word segmentation and then character recognition in several levels. The main one is to take into account confidence levels of OCR output without dictionary yet in order to choose the right character in the N-best list instead of always considering the better recognized one. The best answer is given by

the Viterbi algorithm which uses probabilities of trigrams (a sequence of 3 letters), bigrams and unigrams of a large corpus (newspapers during 10 years). These probabilities are computed by maximum likelihood and the Katz [26] model, as smoothing technique, is used to handle absent N-grams. The current work aims at sharpening correction results based on a dictionary and linguistic levels.

#### 4. Human-machine interface

Personal digital assistants (PDA) have become increasingly popular in the last few years, but partially sighted people are unable to use these devices. PDA interface has been designed using visual input and feedback, which is inaccessible to the visually impaired user.

Thus, providing an easy use of those applications on a standard PDA was a part of the hard work. With the aim of answering the users' requests at best, a dedicated interface has been developed. First, we had to find an intuitive way to evolve in this brand new environment.

##### 4.1. Input

Data introduction by visually impaired people is the main problem. The small number of buttons and PDA's touch screen make their use almost impossible. Speech recognition would be a solution. This system begins to give some results, but a lot of errors are introduced by this too early technology, the bad quality of the micros on PDAs and the noise sensitivity. Its limitations have been showed by tests in real situation.

Adding new buttons seemed to be a better solution. Blind people can easily use the touch screen with an overlay on this one: this way turns its drawback into advantage (Figure 7). Upon pressing the keys of the overlay, the user applies pressure precisely on a specific part of the display. This allows impaired operators to use the touch screen by providing sensitive indicators on the overlay buttons. The system can be easily removed and adapted on most personal assistants.

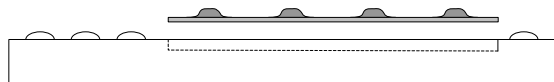


Figure 7. Overlay for PDA touch screen

##### 4.2. Output

Several methods can be used by visually impaired people to have access to digital information, but only a few are suitable within the framework of mobile applications.

Based on eight mobile dots, Braille cells are often used on specialised keyboards and blindness personal assistants [27]. But as opposed to what one believes, this kind of written language is only known by a minority of partially sighted people. A lot of tardily handicapped

persons, following an accident or due to elderly are dissuaded by a very complex learning. Moreover, the excessive price of each one of these mobile cells makes the total cost of this kind of assistant rising.

Another usual method to provide information to visually impaired people would be the pre-recorded messages, especially when the quantity of information to be transmitted is very important (like books or newspapers). A great sound quality does not make the user tired. However, this system does not fit in with a user interface because our application requires dynamically produced messages.

The use of speech synthesis seems to be the most relevant way. It offers the freedom of use and the expected dynamics. A lightweight text-to-speech engine on the user's mobile device generates the spoken output needed by the user [28]. Text files are much smaller than the audio files containing the same information.

##### 4.3. Menu Navigation

Using speech synthesis combined with a virtual keyboard provides a global system, including all the necessary applications to partially sighted people, without any investment of costly and cumbersome material. Some modules have already been integrated such as memento, contacts, calculator, etc.

The communication with users is based on natural language. Each menu item consists on either an introductory text followed by an enumeration of the subtopics or an information text. The vocal interface can also say the systems status and information about current function on demand. So, the blind user can know what the PDA is doing anytime.

This system has been developed after a series of interviews with potential users from the Creth [29], a help centre for people with disabilities.



Figure 8. Prototype using the overlay

## 5. Conclusions

This paper presented a system for camera-based off-line text recognition on an embedded system designed for blind and visually impaired people. This particular design is challenging and relevant for both the document recognition and the embedded systems fields.

Currently, the system performs well in detection and recognition of text with nearly uniform backgrounds. But it is not accurate enough to take into account more complex pictures like outdoor situations. New approaches using colour information are under investigation.

A first prototype device has already been implemented in order to have a quick feedback from the blind users. This methodology of work is the key aspect of our user-centred approach. Even though the system is in its infancy and needs further works, first opinions are satisfying and promising. Practical applications such as those already implemented can easily be added in order to build a complete talking assistant.

## 6. References

- [1] D. Doermann, "Progress in camera-based document image analysis", 7th ICDAR Conference, vol.1, pp. 606 – 616, August 2003
- [2] 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
- [3] A.K. Jain and B. Yu, "Automatic Text location in images and video frames", Pattern Recognition, vol. 31, n.12, pp. 2055-2076, 1995
- [4] J. Ohya, A. Shio, S. Akamatsu, "Recognizing characters in scene images", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, pp. 214-220, 1994
- [5] M. Pietikainen and O. Okun, "Text extraction from grey scale page images by simple edge detectors", Proceedings of the 12th Scandinavian Conference on Image Analysis, pp. 628-635, 2001
- [6] 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
- [7] Y. Zhong, K. Karuand A.K. Jain, "Locating text in complex color images", Pattern Recognition, vol. 28, n.10, pp. 1523-1535, 1995
- [8] V. Wu, R. Manmatha, "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
- [9] 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
- [10] A.K. Jain, S. Bhattacharjee, "Text segmentation using Gabor filters for automatic document processing", Machine Vision and Applications, vol. 5, pp. 169-184, 1992
- [11] A.C. Bovik, M. Clark, W. Geisler, "Multichannel texture analysis using localized spatial filters", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, pp. 55-73, 1990
- [12] D. Dunn, W. E. Higgins, "Texture segmentation using 2-D Gabor elementary functions", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, pp. 130-149, 1994
- [13] A.K. Jain, F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters", pattern Recognition, vol. 24, pp. 1167-1186, 1991
- [14] P. Clark, M. Mirmehdi, "Recognizing text in real scenes", International Journal on Document Analysis and Recognition, vol.4, pp. 855-877, 1998
- [15] P.K. Sahoo, S. Soltani, "A survey of thresholding technique", CVGIP vol. 41 (1988) 233-260
- [16] J. Kittler, J. Illingworth, "Threshold selection based on a simple image statistic", CVGIP vol. 30 (1985) 125-147
- [17] N. Otsu, "A thresholding selection method from gray-level histogram", IEEE Transactions on Systems, Man, and Cybernetics, vol. 9 (1979) 62-66
- [18] W.Niblack, "An introduction to Digital Image Processing", Englewood cliffs, N.J.: Prentice Hall(1986) 115—116
- [19] C.Thillou and B.Gosselin "Robust thresholding based on wavelets and thinning algorithms for degraded camera images", Proceedings of ACIVS 2004, (2004)
- [20] I.Daubechies, "Ten lectures on wavelets", SIAM (1992)
- [21] 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
- [22] O.D.Trier, A.K.Jain and T.Taxt, "Feature extraction methods for character recognition", a survey, Pattern Recognition, vol. 29, n°4, pp 641-662, 1996
- [23] B.Gosselin, "Applications de réseaux de neurones artificiels à la reconnaissance automatique de caractères manuscrits", PhD thesis, Faculté Polytechnique de Mons, 1996
- [24] I.Guyon, "A scaling law for the validation-set training-set size ratio", AT&T Bell Laboratories, 1997
- [25] K.Kukich, "Techniques for automatically correcting words in text", ACM Computing Surveys, vol. 24, n°4, 1992
- [26] S.F.Chen and J.Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling", Proceedings of the 34th Annual Meeting of the ACL, 1996
- [27] Brailnote, <http://www.pulsedata.com>, visited in July 2004
- [28] Acapela group, <http://www.acapela-group.com/>, visited in July 2004
- [29] Creth, <http://www.creth.be/>, visited in July 2004