Optical Recognition of Braille Writing Using Standard Equipment

Jan Mennens, Luc van Tichelen, Guido François, and Jan J. Engelen

***Abstract*— The reproduction of Braille writing has been, up until now, a purely manual job. To overcome this problem, many researchers have tried to develop a Braille reading machine in some way or another. Their efforts have not given a satisfying solution. The goal of the research described in this text is to develop a system that converts, within acceptable constraints, Braille (image) to a computer readable form (text). Having the text on a computer, a Braille printing house can reproduce it using an electronic Braille embosser.**

***Index Terms*— Optical Character Recognition, Relief writing, Braille, Digital Image Processing.**

I. Introduction

BRAILLE writing is a widely spread means of communication for blind or partially sighted people. It consists of a system of six or eight possible dot combinations that are arranged in a fixed matrix, called a cell (see Fig. 1).

Every dot can be set or cleared, giving 64 possible combinations in six-dot and 256 combinations in eight-dot Braille.

All dots of a Braille page should fall on the intersections of an orthogonal grid. When texts are printed double-sided (recto-verso), the grid of the verso text is shifted so that its dots fall in between the recto dots (see Fig. 1).

Braille has a low information density. An average page of 25 × 29 cm, can have 32 characters on a line and 27 lines in a page. A typical dot has a diameter of 1.8 mm.

Braille writing can not be processed with standard optical character recognition (OCR) software. This is due to the fragmented nature of the characters, and the fact that characters on both sides can be seen simultaneously. A different approach has to be considered. Using the property that Braille characters are always positioned on a fixed matrix, we first try to build a grid consisting of horizontal and vertical lines that run through all the dots, and then we check if there is a dot present on each of the intersection points. The grid construction must be flexible because there are cases where it can be deformed or irregular. This depends greatly on the quality of the original.

The main reason for developing a system that can read Braille is to preserve and multiply large volumes of manually crafted books. Many books on mathematics or music are very difficult, even for a skilled copyist, to retype due to the special rules that apply in Braille.

Manuscript received June 28, 1994; revised September 7, 1994. This research was funded as part of the Braille project of the Catholic University of Leuven.

The authors are with the Research Group of Applied Electronics and Optics, Faculty of Applied Sciences, Catholic University of Leuven, Belgium.

IEEE Log Number 9406708.

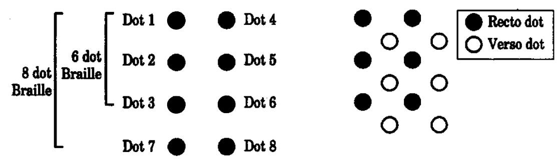


Fig. 1. The six dot and eight dot Braille cell (left). Positioning of recto and verso Braille (right).

This text will describe the essential parts of the hardware and the conversion software.

II. Research

There has already been a lot of research done in the field of reading Braille with machine vision. We will summarize the most important here.

- In the "Spermalie Institute" in Bruges, Belgium, a simple scanning device was made, using a plotter. The pen was replaced with four light-sensitive elements. This method could only be used for single-sided Braille. An average fault of 2% was claimed. No publications were made on this research.
- At the Technical University of Delft, the Netherlands, a research in cooperation with Farrington Data Processing [4], [15] produced a Braille Reading Tablet. This device uses a ruler with a reading head that contains three light-sensitive cells. The head needs to be moved along the lines of Braille, using the rulers. The system is very sensitive to positioning errors.
- In Le Centre Marie Morel [13], Paris, France, an ordinary scanner was used to digitize the Braille pages. The sensitivity of the scanner was turned to a very low value (10%) so the Braille dots would show up as dark spots in the resulting image. The results seemed to be promising, but after contacting the authors, it seems that there was no further research done on this subject.
- At the University of Sciences and Technology [14], [17] in Lille, France, a CCD camera with 512 × 512 pixels was used to digitize the image. Because the resolution of this camera is too low, the original is digitized in two steps. The system has an error of 1% and an average conversion time of 7 s per line. Newer hardware could improve these results. The system was never used outside the laboratory.
- At the University of Agriculture and Technology [21]–[23] in Tokyo, Japan, research was done on the optical recognition of both opaque and transparent Braille.

A CCD camera was used for the detection of bright regions caused by the LED illumination. The articles describe the recognition probability of the different set-ups. No indications were given on practical use.

- At the University of Sherbrooke [19] in Québec, Canada, an optical probe was developed to read and spell Braille. The reading speed is five characters per second and the recognition rate seems to be very high. The device can read both single- and double-sided Braille.
- At the University of Manchester [7], England, the original is illuminated from two different angles and digitized with a CCD camera. Both images are subtracted from each other. The result is an image that is very insensitive for stains or stripes on the paper's surface. At the moment, the average conversion error is 8%. This work is still in the experimental phase. Recognition of double-sided Braille will be attempted.

The study of "reading" Braille texts in our research group is spread over a few years now. As a scanning device, a laser was used [12], a vidicon [1], [2], [16], [18], and a CCD camera [3], [10]. None of the approaches gave satisfying results.

The most important conclusions that could be drawn were

- Illumination is very important. It should be oblique and as uniform as possible over the whole page.
- Braille reading needs a relatively high resolution to give reliable results.

III. Constraints

Before starting the present project, we have put some constraints on the development of a system that should recognize Braille characters.

- The final application should run on commercially available computers that are of moderate cost and easy to maintain (*Standard Equipment*).
- Conversion of a single page should be finished within 30 s [25] and the error rate should be low.
- Both single- and double-sided texts should be recognized [25].

IV. Problems

Although Braille dots are placed on a fixed matrix, certain Braille production techniques cause this matrix to be irregular. Two major types of deformation can be distinguished.

1) Type 1: Deformations of the Braille Cell:

- Spacing between the dots and the cells may vary in both directions.
- Degradation of the dots (wear).
- In case of double-sided Braille, both sides are seen simultaneously. In some cases, it is difficult to separate them because the light/shade patterns of one side tend to fade in those of the other side.

2) Type 2: Deformation of the Grid Where Braille Characters are Positioned:

- The page went askew while the text was being typed.
- The text was created using manual cut-and-paste techniques.

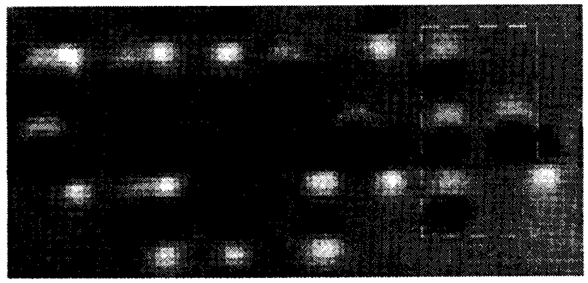


Fig. 2. A fragment of an image with some recto and verso Braille dots (resolution: 120 dpi[1]). A single recto cell is marked with a dashed rectangle.

- The text's layout sometimes causes large gaps in the text and, as a consequence, in the calculated grid.

Braille characters cannot be recognized on a character basis because the position of the dots in the Braille cell can only be found by relating them to neighboring characters. If we find a single dot, we need more information to decide to which of the six (or eight) possible cell positions it belongs. If we find two dots, they can belong to the same cell or they can come from different cells. The distance between the dots can not be used as a decisive criterion.

V. Optics

Because we wanted to make a check on the experimental results of previous research in our group, a 3-D model of a single Braille dot was developed [20]. An idealized dot, a spherical cap, was illuminated with a light source that could be either flat or curved. The reflection map was calculated using ray tracing techniques [5]. The model predicted that Braille dots could easily be distinguished from the background and recto dots could be distinguished from verso dots, as long as the illumination was asymmetrical. Quality improves when the angle of incidence becomes low and when the opening angle of the light beam becomes small.

Standard scanners try to illuminate the paper's surface as symmetrically as possible. Making the light source *asymmetrical* makes the Braille relief visible in the scanned image. The dots appear in the image as regions of high and low intensity (see Fig. 2). In our setup, recto dots have the light area above the dark area and verso dots appear vice versa.

The study of the illumination, together with experiences in further stages of the conversion, indicate that the lower limit of the resolution is 80 dpi[1]. If the resolution gets too high, too much detail is seen in the image and this confuses the algorithms that search for grid lines. A practical upper limit is 200 dpi.

VI. Strategy

It is conceivable to create a mask for a standard recto and verso Braille dot. This mask could then be convolved with the image, resulting in peaks on the position of the Braille dots. These peaks would then have to be grouped in cells to convert them to characters. This means that we also have to find the

[1] Dots per inch. This expression refers to resolvable image pixels, not Braille dots.

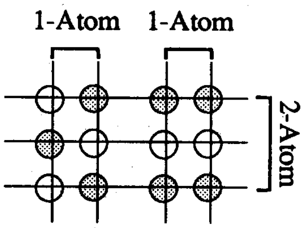


Fig. 3. Example of atoms in a six-dot Braille cell. One pair of lines is called a 1-Atom, two pairs of lines are called a 2-Atom.

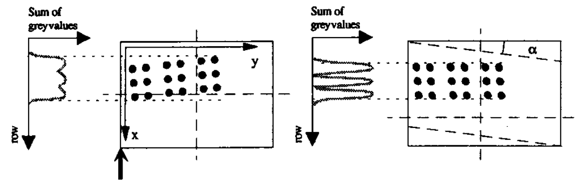


Fig. 4. Slanting the image in the vertical direction to find the rotation angle α.

grid on which the dots are positioned. The convolution of the image with two 8×10 pixel masks (this is a typical size at 100 dpi) takes a lot of computation time. The major problem of this technique is that Braille dots vary a lot in size and shape, even on one and the same page. This means that there will be a large variation in the size and position of the found peaks. Also, a lot of false peaks are generated on locations where recto and verso dots are clustered together.

In view of these problems, we decided to adopt a different strategy. Using a very simple mask, we first try to find the position of the grid lines. In order to cope with *type 2 deformations*, the image is divided in subimages and grids will be calculated for each subimage separately (local linearization of the deformations). Grids of horizontally and vertically adjacent subimages are then taken together in horizontal and vertical *strips*: horizontal strips contain vertical grid lines, vertical strips contain horizontal grid lines.

The lines in a strip will then be grouped in *atoms*. An atom is a group of lines that belongs to a single Braille cell. An n-atom has n line pairs (see Fig. 3). In six-dot Braille, we have 1-atoms and 2-atoms. In eight-dot Braille, 1-atoms and 3-atoms are found. The grouping in atoms has to be flexible because lines are not necessarily found at fixed distances (*type 1 deformations*).

In the last step, the relationship between the strips is restored, and at the cross-sections of horizontal and vertical atoms, Braille characters can be found.

The conversion from original image to text will be described in more detail in Sections VIII to X.

VII. Equipment

Due to the size of the Braille texts, an A3 scanner was needed to read all possible originals. In our tests, we used the AGFA HORIZON scanner.

Because of the graphical nature of the problem, a graphical development environment was selected. Large amounts of memory need to be handled and standard fast interfaces need to be available. We chose the Apple Macintosh.

The first draft of the software was developed in MATLAB on VAX and Apple Macintosh computers [8]. The final test program is written in C++ in the MacApp 3.01 environment of the Apple Computer [9].

VIII. Preparation of the Image

Braille pages are never perfectly flat. This is due to the tension in the paper caused by the production of the Braille dots. It introduces low frequency distortions (shadows) that can be eliminated by subtracting a local averaged image from the original.

The image will be divided in subimages to minimize the influence of *type 2 distortions* (Local linearization). If the grid is very distorted, the subimages will become small. Typical subimages are 12 characters wide and 6 lines of text high.

The calculated grid must be aligned with the main axes of the text. It is possible to work with a rotated grid or to align the complete image to orthogonal axes first. The latter has certain advantages in the organization of the data so this solution was chosen.

The Discrete Fourier Transform (DFT) can be used to calculate the rotation angle [6], but the image's structure causes important and recognizable data peaks in the DFT image to lie too close to the origin. This gives an error on the calculated rotation angle, and a further refinement is necessary. We followed a different approach. Using the deviation over the sum of the rows, the image was slanted over an angle. Each time the image is slanted one pixel in the vertical direction, the deviation over the sum of the rows is calculated. A maximum is obtained when the dots are aligned horizontally (see Fig. 4).

To speed up this routine, the resolution was lowered. A stepwise approach was followed. In the horizontal direction, a fixed decrease in resolution was used. In the vertical direction, the resolution was gradually increased from an initial low value back and up to the original resolution. In each step, the rotation angle was searched within the error interval of the previous step. The image was rotated with an approximated rotation that used a horizontal and a vertical slant [11], [24].

To increase speed even more, the calculations are not executed on the entire image but on a subimage that *probably* contains data. The standard deviation (σ) of the gray values of every subimage is used as a selection criterion. This method gives us *no* certainty that this subimage actually contains Braille, but if it does, this parameter tells us where the dot-density is most significant. To obtain an acceptable error, the subimage should not be too small. An additional benefit of working in a subimage is that it takes certain deformations of the text grid into account. This technique needs far less computation time, compared to the FFT method. Eventually, it will take only 2% of the total conversion time.

In the digitized Braille page, dots are represented with light and dark areas (see Fig. 2). We classify these areas by making a three-value image in which all pixels with a value between $+n \cdot \sigma$ are made zero. Pixel values above this range are made $+1$ and below this range -1 (n is typically between 1 and 2, and allows adaptation to different illumination conditions).

$$\begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ +1 \end{bmatrix}$$

Fig. 5. Correlation mask for an image with a distance of five pixels between the center of the dark and light zones.

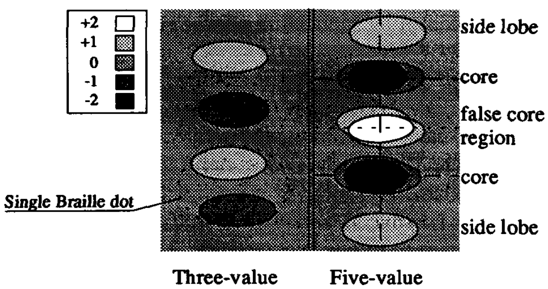


Fig. 6. Neighboring dots introduce false core regions or ghost-points.

The actual center of the dots can be found by correlating the image with a mask as shown in Fig. 5. This mask is an absolute simplification of a Braille dot.

The vertical size of the mask depends on the size of the Braille dots and equals the distance between the centers of the dark and light area of an average dot. After correlation, we obtain an image with five values. Regions with value -2 (recto) and $+2$ (verso) are called core regions, regions with value $+1$ (recto) and -1 (verso) are called side-lobes, and 0 is background. Two vertically neighboring dots of the same side introduce a false core region (ghost points) for the other side (see Fig. 6).

IX. Grid Calculation

Grid lines are searched by making histograms of rows and columns in the five-value image. These histograms have five bins (-2, -1, 0, 1, 2) (see Fig. 7).

Firstly, the histograms for the rows are calculated. When a plot is made of the number of pixels in bin ($+2$) and bin (-2) (*bin-curves*), they show maxima where rows of recto and verso dots reside.

Next, the histograms of the columns are calculated. Because the previously found horizontal grid lines are eligible to contain Braille dots, it is sufficient to consider these lines only in creating the histograms. This reduces computer time and provides even extra protection against other possible disturbances. Another improvement was made by making combinations of values found in different bins. In the case of recto dots, the presence of pixels with a value -2 was rewarded and the presence of pixels with a value -1 was penalized. These pixels are always found around the center of a Braille dot. This procedure will result in narrower peaks.

After we found the vertical grid lines, the horizontal lines were searched again, but in a reduced image that consists of

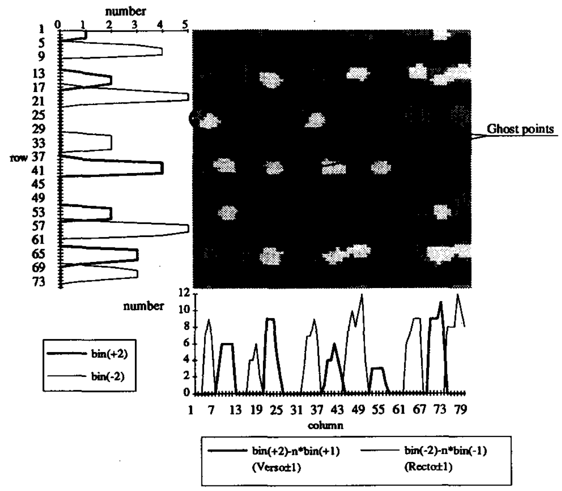


Fig. 7. Histograms of rows and combined histograms of columns mark the positions of the grid lines. Three examples of ghost-points are highlighted.

the vertical grid lines only. This is not only more efficient but effectively enhances the results by eliminating the influence of false core regions typically found on vertical grid lines of the opposite side.

Because grid lines were calculated separately for each sub-image, the data needs to be regrouped in strips, to get an overview of the complete image.

Before we can do any interpretation of the Braille dots, we must find all atoms in the vertical and horizontal grid lines. An algorithm was developed to restore the vertical and horizontal grids to atoms. Distances and tolerances on line positions are always calculated on a local basis, i.e. the routine *learns* from local conditions to decide where candidate lines can be found.

X. Dots to ASCII Code

When all atoms are found, a small test (a minimal local correlation) will check for the presence of a dot on the intersection of two lines. These dots are grouped in a binary Braille character set (Binary Braille: each dot is represented by a bit position), which can be translated using an appropriate translation table to any other code.

XI. Test Results

We tested the algorithms on various types of Braille: both single-sided and double-sided, printed on carriers with different textures and colors. Every set contained an equal amount of samples. The results can be divided into three groups. The first group (four sets) was converted *without any errors*. In the second group (seven sets), an average error rate of 0.25% on a character basis, was obtained (this is two errors on an average page with 864 characters). In this group, errors were caused by defects or stains on the paper's surface. The last group (one set) could not be read with an acceptable error rate because of a particular deformation of the character matrix with which the program could not cope.

Conversion times are typically 60 s for single-sided, and 80 s for double-sided Braille on a Macintosh IIfx. Higher conversion speeds could be obtained on faster models of this computer.

XII. Conclusion

Using a commercially available scanner and a set of fairly basic calculation methods to obtain a maximum conversion speed, a text containing Braille characters can be converted to any other character code. The most important characteristic of all our routines is that they try to work on a local basis, without losing contact with the global context, and try to postpone complex calculations until the data set has been significantly reduced.

Acknowledgment

The authors would like to acknowledge Dr. F. Librecht and his crew of AGFA GEVAERT for help and advice on the AGFA HORIZON scanner, and the Apple R&D fund for their contribution in the project.

References

[1] J. Branders and G. Bussinck, "Patroonherkenning van Braille Bladzijden M.B.V. Een TV Camera," Eindverhandeling Toegepaste Wetenschappen, K. U. Leuven, 1983.

[2] P. Calders and A. Keymis, "Patroonherkenning van Braille Bladzijden M.B.V. Een TV Camera," Eindverhandeling Toegepaste Wetenschappen, K. U. Leuven, 1984.

[3] I. De Baere and P. De Boeck, "Ontwerp en Ontwikkeling van Een Stuur—en Interface Eenheid Voor CCD-Camera en VME-bus," Eindverhandeling Toegepaste Wetenschappen, K. U. Leuven, 1988.

[4] *Farrington Data Processing*. Haarlem: Gebruikershandleiding van de Braille Lees Tablet, 1984.

[5] A. S. Glassner, *et al.*, *An Introduction to Ray Tracing*. London: Academic Press Ltd., 1989.

[6] R. C. Gonzalez and P. Wintz, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1987.

[7] T. W. Hentzschel, "An optical Braille reading system," M.Sc. thesis, Univ. of Manchester-UMIST, 1993.

[8] J. Mennens, L. Van Tichelen, and P. Vandeputte, "Matlab source code for Cobra/M," Int. Rep., Katholieke Universiteit Leuven, België, 1993.

[9] ——, "C++ source code for Cobra," Int. Rep., Katholieke Universiteit Leuven, België, 1993.

[10] M. Poels and Y. Poullet, "Ontwikkeling van Een CCD-Camera Interface en Dubbelzijdige Patroonherkenning," Eindverhandeling Toegepaste Wetenschappen, K. U. Leuven, 1989.

[11] W. K. Pratt, *Digital Image Processing*. New York: Wiley & Sons, 1991.

[12] L. Vander Voorde and G. Vandensande, "Patroonherkenning van Braillebladzijden," Eindverhandeling Toegepaste Wetenschappen, K. U. Leuven, 1982.

[13] J. Abouassly, A. Dahmani, E. Hachem, C. Fluhr, P. Radasoa, and A. Vasseur, "Les travaux de recherche au Centre Marie Morel," in *Proc. 6th Int. Workshop on Applications for the Visually Handicapped*, Leuven, Sept. 1990.

[14] M. Benjelloun, V. Devlaminck, F. Wauquier, P. Altmayer, and J. P. Dubus, "Application du traitement d'images á la conception d'un systÈme de transmission de relief Braille en texte noir," *Traitement du Signal*, vol. 6, no. 4, pp. 291–300, Février 1989.

[15] F. Bruggeman, *et al.*, "Braille optical reader," Int. Rep., TH Delft, The Netherlands, 1984.

[16] P. Calders, J. Mennens, and G. François, "Optical pattern recognition of Braille originals," in *Proc. 3rd Int. Sympos. Optical and Optoelectronic Applied Sciences and Engineering*, Innsbruck, pp. 229–232, Apr. 1986.

[17] J. P. Dubus, M. Benjelloun, V. Devlaminck, F. Wauquier, and P. Altmayer, "Image processing techniques to perform an autonomous system to translate relief Braille back into ink called LectoBraille," in *IEEE 10th Int. Conf. in Medicine and Biology Society*, New Orleans, Nov. 1988, pp. 1585–84.

[18] G. François and P. Calders, "The reproduction of Braille originals by means of Optical Pattern Recognition," in *Proc. 5th Int. Workshop on Computerized Braille Production*, Winterthur, Oct. 1985, pp. 119–122.

[19] G. Lachiver, J.-P. Vachon, and W. D. Seufert, "An optoelectronic device to read and spell Braille—Braillect," *IEEE Trans. Biomed. Eng.*, vol. BME-31, no. 8, pp. 560–563, Aug. 1984.

[20] J. Mennens and L. Van Tichelen, "Using asymmetrical illumination to recognize double sided Braille," Int. Rep., Catholic University of Leuven, Belgium, 1992.

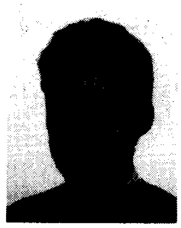
[21] S. Nitta and M. Ohkawa, "The support system of Braille reading for view handicapped people," in *Proc. '87*, 1987, pp. 981–984.

[22] ——, "Recognition probability of Braille-reading prototype system," in *Proc. SICE '88*, 1988, pp. 799–802.

[23] S. Nitta, J. Oshima, and M. Ohkawa, "Recognition algorithm on transparent Braille," in *Proc. SICE '89*, 1989, pp. 1017–1020.

[24] W. Postl, "Detection of linear oblique structures and skew scan in digitized documents," in *Proc. 8th Int. Conf. on Pattern Recognition*, Paris, Oct. 1986, pp. 687–689.

[25] World Blind Union Research Committee, "Priorities for technical research and development for visually disabled persons: A summary-report," Apr. 1992.



Jan Mennens was born in 1960. He received the M.Sc. degree in electrical engineering from the Katholieke Universiteit Leuven (Belgium), in 1984. He obtained the Ph.D. in applied sciences in 1993 on the optical recognition of Braille originals by means of standard equipment.

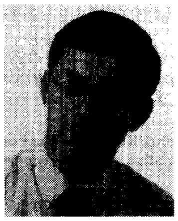
In 1985, he became a Research Assistant in the Applied Electronics and Optics Section of the Katholieke Universiteit Leuven where he did research in the field of automatical conversion of text to Braille, Braille text processing and production, and the optical recognition of Braille. Currently, he is involved in the commercialization of the COBRA software and in the development of an advanced Braille production software package.

Luc van Tichelen was born in 1967. He received the M.Sc. degree in electrical engineering from the Katholieke Universiteit Leuven (Belgium), in 1990.

From 1990 to 1993, he was a Research Member of the Applied Electronics and Optics Section of the Katholieke Universiteit Leuven. In 1993, he joined Lernout and Hauspie Speech Products, Ieper, Belgium, where he has been developing several algorithms to facilitate the development of multilingual Text-to-Speech systems. His current interests include image and signal processing, artificial intelligence, and spoken language processing.

Guido François was born in 1936. He graduated from the State University of Ghent in 1960. A graduate fellowship from the Belgian American Educational Foundation allowed him to continue his studies at Stanford University, where he obtained the Ph.D. on the study of nonlinear objects in 1965.

Currently, he is a Professor at the Katholieke Universiteit Leuven (Belgium), and is working on various topics in the field of optics, including some successful industrial applications. Since the end of the 1970's, he has focused his research efforts on the "Braille Project." This project aims at giving the blind and visually handicapped better access to written information by the application of electronic systems. This work has resulted in both software and hardware for Braille production that have found a wide application throughout Europe.



Jan J. Engelen was born in 1945.

He is a Professor at the Katholieke Universiteit Leuven (Belgium), and has been involved in laser optics and, more recently, in the field of rehabilitation engineering for the benefit of visually impaired persons. He has been involved in software production for the first ELEKUL embossers (currently marketed by Interpoint, NY). In 1985, he founded the InfoVisie technical advisory group. Former research activities include conversion of word processing files, access to TV/Teletext and Home Banking systems. Currently, his research focus is on the large field of accessibility of structured electronic documents for reading impaired persons. This research was mainly sponsored by the European Community through its DG XIII/TIDE program (Technology Initiative for Disabled and Elderly).