01 Jan 1989

Personal Computing for the Visually Impaired

Bruce M. McMillin
Missouri University of Science and Technology, ff@mst.edu

P. Y. McMillin

Recommended Citation

B. M. McMillin and P. Y. McMillin, "Personal Computing for the Visually Impaired," *IEEE Potentials*, Institute of Electrical and Electronics Engineers (IEEE), Jan 1989.
The definitive version is available at https://doi.org/10.1109/45.31590

Personal computing for the visually impaired

Bruce M. McMillin
Patricia Y. McMillin

As with foreign languages, direct translations—say, into sound or touch—often don't work

Imagine it's the first day of a new semester. You are sitting in your laboratory section in front of a PC (personal computer) and a student walks in with a white cane. This person may be totally blind, able to see light, partially sighted, or just legally blind. Still he or she is in college, just like you. The only difference is that this person cannot interact with the world in exactly the same way you do.

Technology can facilitate this interaction. Often, technological aids for the physically challenged seem to evolve from disciplines other than engineering. Indeed, requirements for such systems may be generated from the social sciences, education, and rehabilitation fields. However, the engineer is responsible for interpreting and implementing these requirements. This article will provide just such a set of engineering requirements for technological solutions for visually impaired persons. But, perhaps more importantly, it will show how engineering techniques are used in the "human oriented" sciences.

The major difficulty a visually impaired person must overcome in a visually oriented world is the problem of interacting with that world. Traditional media, such as textbooks and handouts, are useless. The information must be made available in a special tactile language called Braille or it must be available in voice (either on tape or read out loud). If we have control over the original format of the information, then disseminating the information into these forms becomes easier. A particularly interesting problem and solution arises when the original format is stored in a personal computer. As it stands, the material isn't terribly useful to someone who can't see. The output appears silently on a video screen, and without someone there to read it, there is no way for the visually impaired to obtain the information present.

Data entry, in its abstract form, seems to present no problem. Most visually impaired people know how to type on a typewriter keyboard at a very early age. But anticipating all the possible contingencies—for example, try invoking a spreadsheet or logging onto a mainframe with no feedback—is very, very tough.

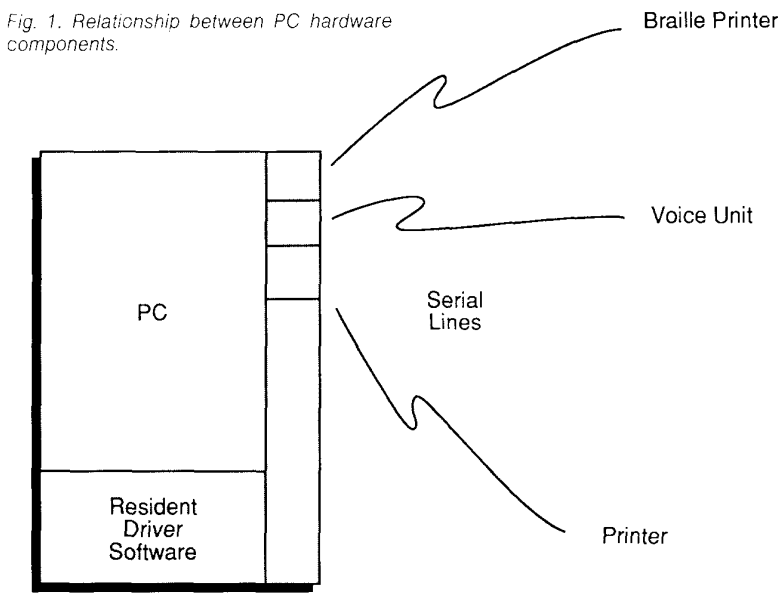
Feedback is the fundamental problem for operating a personal computer. Feedback is normally by sight. When sight is not available, feedback must take other forms. The conversion of printed text to an output form understandable by the visually impaired generally involves a lot of manual labor. However, converting computer screen text can be accomplished in an automated fashion. Many inexpensive hardware devices exist that can synthesize speech. Other hardware can produce tactile output. Linking these to digitally stored data eliminates any manual intervention. Since they have the necessary feedback available, a visually impaired user may use a machine without any external help. Figure 1 depicts the physical relationship between the PC and various specialized hardware devices.

Realize that hardware is not a magic box. Hardware and software must cooperate to provide a meaningful interface. Two important aspects to consider are meaningful input and meaningful output. Figure 2 depicts the possibilities for each.

Traditional output techniques

Tactile output has been the mainstay of written communication for the visually impaired for over 100 years. The Braille alphabet consists of Braille cells each containing a fixed array of six positions numbered one thru six. Each position or dot is either raised or not (binary). Thus, there are 2^6 or 64 possible combinations. Figure 3a depicts the configu-

Fig. 1. Relationship between PC hardware components.



0278-6648/89/0005-0017$01.00 © 1989 IEEE

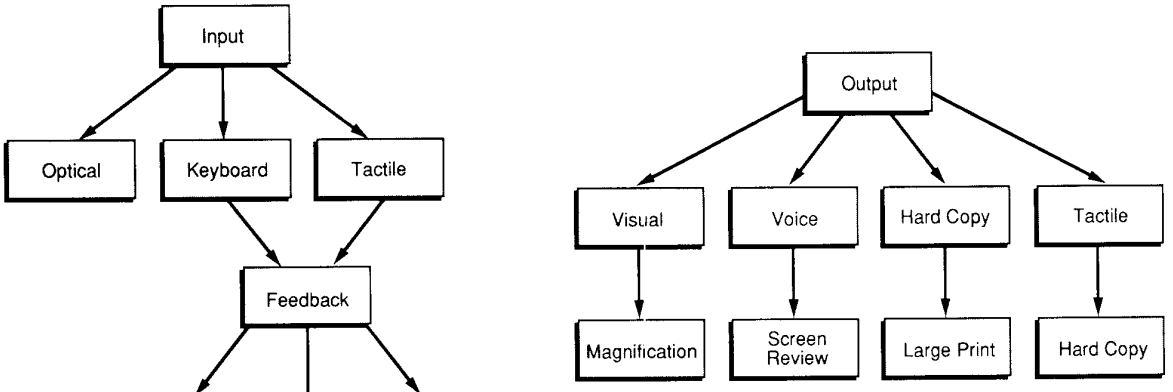


Fig. 2. Input and output requirements.

ration of the Braille cell. The dots are imprinted on heavy stock and read by touch. The Braille alphabet is shown in Figure 3b. You may have noticed Braille in most public elevators. It takes practice to train one's sense of touch but, once mastered, it is a valuable skill.

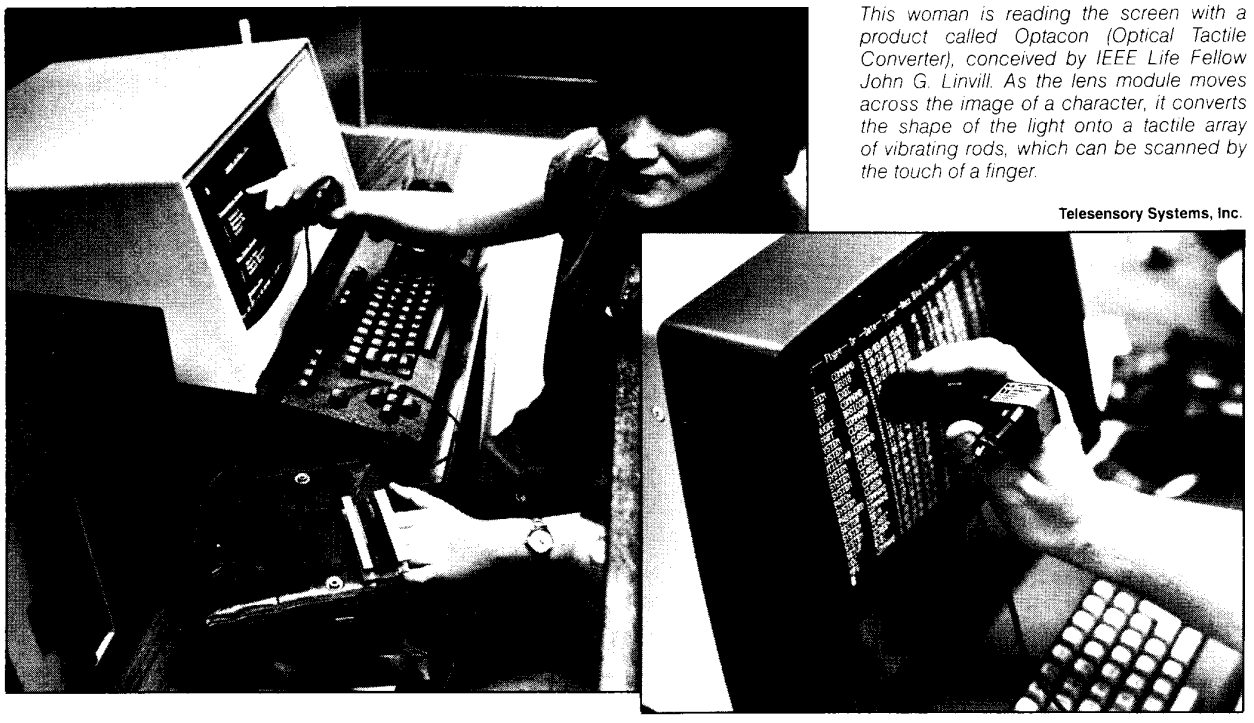
Since there are 63 possibilities (blank is not a valid symbol other than for delimiting purposes), each of the 26 letters of the alphabet is assigned to a particular configuration. For example, the letter "o" is formed by raising dots 1-3-5. The remainder of the configurations are assigned to punctuation, commonly used pairs of letters, and common words. Certain Braille cells are used as prefix codes to indicate different characters. For example, the number sign prefix (3-4-5-6) used with a letter indicates a string of numbers and a letter sign prefix switches back to letter interpretation.

Braille, however, is not a simple transformation from character to Braille cell. To save space in the resulting text, certain cell configurations take on a different meaning based on context. The letter "k," for instance, also indicates the word "knowledge" if it appears alone in a sentence. The combination of letters "w" and "d" denotes the word "would." Even with these abbreviations though, Braille texts can fill rooms.

Computer input

Input can take several forms. The most common form is natural language text entered from a standard typewriter keyboard. This type of input requires no special hardware. Since touch typists do not look at the keyboard, there is no learning curve for most people.

A second alternative is to use tactile input. A Braillewriter is a mechanical device which formulates



This woman is reading the screen with a product called Optacon (Optical Tactile Converter), conceived by IEEE Life Fellow John G. Linvill. As the lens module moves across the image of a character, it converts the shape of the light onto a tactile array of vibrating rods, which can be scanned by the touch of a finger.

Telesensory Systems, Inc.

Braille cells directly. The device consists of six keys, one for each dot of the cell and a space bar. To print the letter "g," keys 1-2-4-6 are depressed simultaneously. In a mechanical device, these keys are connected to hammers which make indentations in the paper. While this action may seem tedious, an accomplished brailler is amazingly fast. A simple data structure can be used to store Braille cells (since each Braille cell is essentially a bit mask, each cell can be stored as a 6 bit integer).

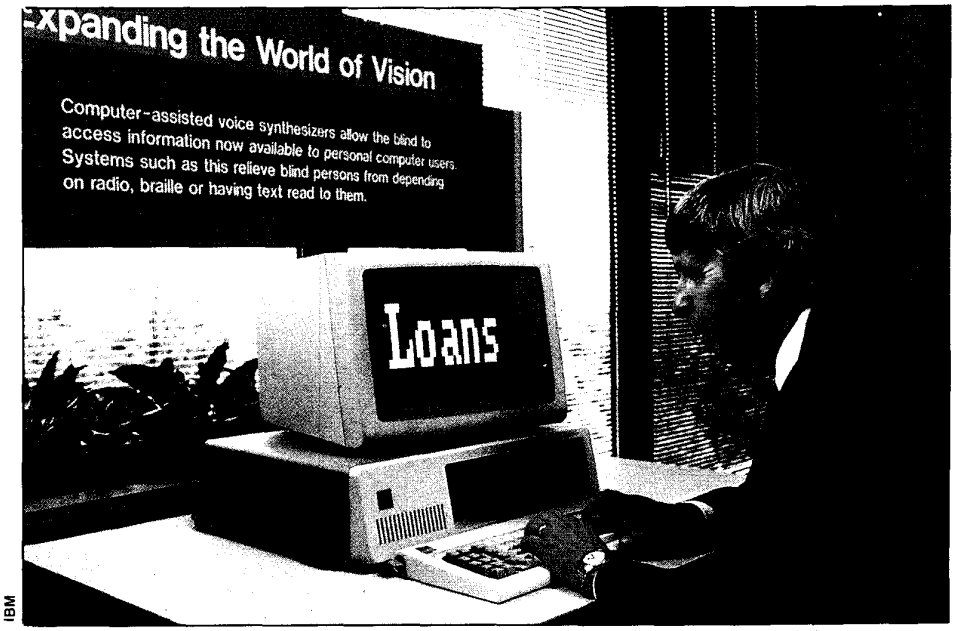
A final alternative, one used for input and display of already printed texts, is the optical scanner. Scanned characters from a printed page are recognized and stored character by character. They can then be displayed by any of the standard output methods described in this paper.

Computer output and feedback

The output and feedback functions have roughly the same requirements with respect to interactive usage. Feedback can be either voice or tactile. For purely visually impaired users, there is no requirement for voice input of text. A first approximation to voice output would be to echo each character as it is typed or displayed. This may seem like a good idea until you try to listen to it. A continuous stream of letters becomes just so much noise. (Try it, have a friend read this last sentence out loud to you a letter at a time.) Thus, the output software must have some intelligence.

Words may be parsed by surrounding spaces. The word itself, in a first attempt, may be pronounced by echoing the individual sounds together. This is fine for a word such as "hat" in which the morphemes (a meaningful speech unit with no smaller parts) used for pronunciation is easily obtainable from the word, itself. However, a word such as "technology" (pronounced tek-nol-e-ji) causes an obvious problem. The software must be able to understand the typewritten word and pronounce it correctly using a combination of phonetic rules and dictionaries.

Even with correct pronunciation based on words alone, the resulting output may be incorrect. Consider words whose pronunciation changes with the context of their usage such as "live" ("I live in the city" and "He's a live wire"). Another problem is homonyms, words that sound alike but mean different things (e.g., "boar" and "bore"). To a lesser extent, the same problem is seen in spelling checkers with words that are spelled correctly but used incorrectly. What is necessary is a pronunciation system that interprets the word in the context of the sentence.



The Personal System/2 Screen Reader™ allows the user to hear the text being displayed. A key pad enables the user to listen at his or her own pace and manner (for example, by character, word, sentence or paragraph).

This discussion on pronunciation has assumed that the text is in natural language (i.e., English). What about the verbalization of a program? In most programming languages, acronyms and special symbols dominate program code. In the C language, comments are denoted by /* and */. Do we want to burden the listener with these noise words? How about parsing words by spaces? This idea no longer works when given the C language statement, for (i = 1; i < MAX; i ++). The speech generator must understand the delimiters of the particular language. In English, the delimiters are spaces. In programming languages, the delimiters have a much wider range.

Word pronunciation is not always desired. Under some circumstances character by character pronunciation may be desired. In checking spelling or in the use of acronyms, individual characters must be made available. So an output device must be capable of switching between the two, perhaps dynamically.

Tactile feedback in the form of Braille cells can be produced on special devices that formulate the Braille character via raised pins on a tactile grid. This method roughly mimics the way one uses a Braillewriter. It may be used both with Brailler input and with keyboard input. Although with keyboard natural language input, the Braille generator should understand the language transformations noted in our discussion of the Braille language. Again, this consideration is for the same reason we don't want to echo each character with voice.

Complicated screen formats pose another problem. We want informational messages to be heard, but screen formatting presented to a sighted user may need to be silenced. These requirements dictate an integrated hardware, software, and application package solution. Simply hooking a voice module capable of word, or even special language, parsing is not sufficient. Any menu driven application would produce so much noise for every screen displayed that listening to it all the time would soon result in discarding that package. For a sighted user, this does not present a problem, as a sighted user simply ignores these extraneous messages.

Another task that is not covered by simple word feedback is the need to review what exists on the screen or in a file. This review cannot be brute force, i.e., every word on the screen is spoken when a new screen is displayed. Imaging using an editor to locate a word. When the word is found, only that word and its surrounding context is important. The

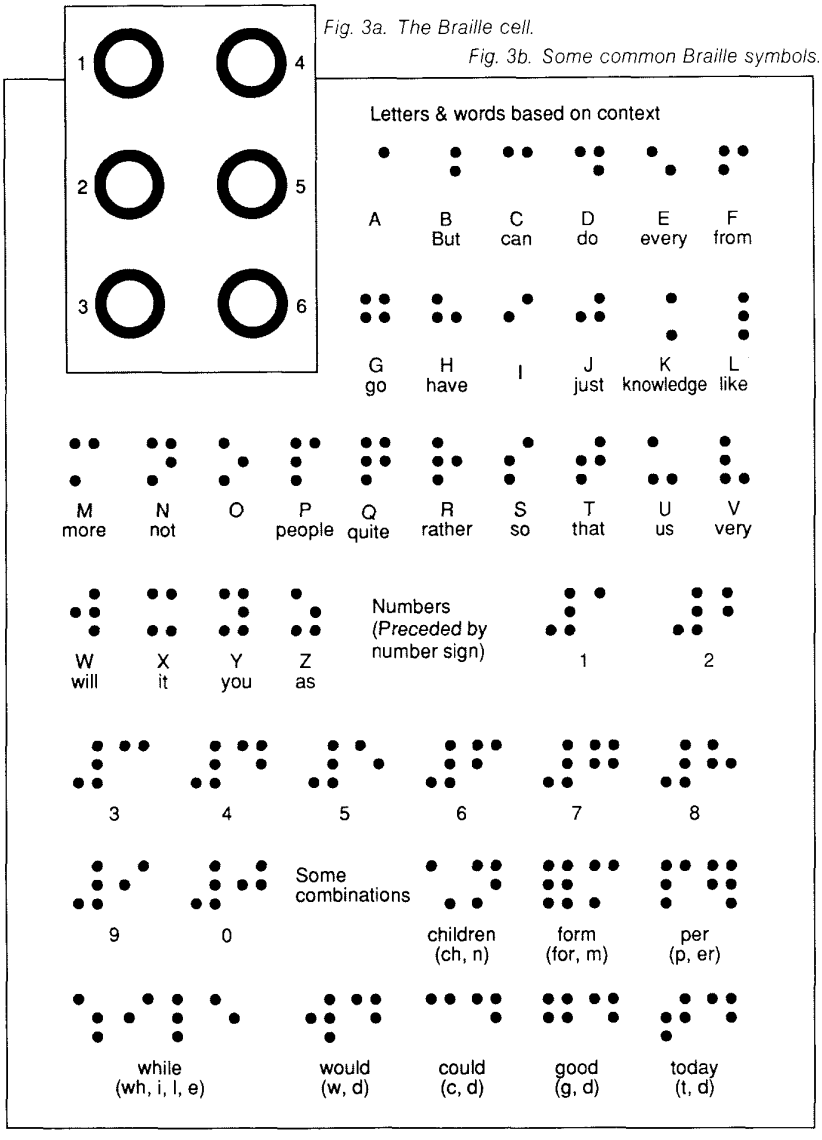


Fig. 3a. The Braille cell.

Fig. 3b. Some common Braille symbols.

other text on the screen is largely noise and ignored by the eye. For vocalization of text, the reduction of the noise words on the screen is equally, if not more, important than for visual display.

The process of moving through a file of spoken text is called screen review. Screen review functions are basically the same as the cursor movement keys in an editor. Some packages provide the screen review function integrated with the editor while some packages provide it independently. Clearly, the prior approach is advantageous if for no other reason than to reduce the number of keys that need be remembered. Still other systems solve this problem by providing a separate keypad for the screen reader functions. In any case, the functionality required is the ability to move the focus of what is spoken up a line, down a line, right, left, and so forth.

Windows present an interesting problem for screen review functionality. Clearly the currently active window should be the primary focus. However, what about activity that occurs asynchronously in other windows? Should the current screen review be interrupted to service another window? If so, how do we do this in a way that is not distracting? Furthermore, how does a visually impaired user switch between windows? These are unanswered questions.

Perhaps by now you expect every output solution to require some specialized hardware and software. For the partially sighted user though, simply using large print may suffice for output. Most type is 10 or 12 point. Large print may be 18 or 20 point. It is relatively easy using most graphics hardware to simply produce enlarged characters on the CRT screen. Alternatively, a fresnel lens may be placed over a standard CRT screen. Still, another possibility is to use a closed circuit TV that can read both CRT screens as well as print. Modern desktop laser printers can produce arbitrarily large text using standard word processing software. A further advantage is that when a document has been proofread and is in its final form, a simple printing control change reduces the output down to "acceptable size" for the non-visually impaired reader. (Indeed, this document was printed in point size 20 during its development with the final version printed in typewriter font for transmittal to the publisher.)

Existing aids

There has been a virtual explosion in products for the visually impaired. The availability of low-cost hardware has resulted in a ballooning cottage industry of packages and hardware. Rather than attempt to provide a taxonomy of devices and packages, we list several sources for further information. The book *Computer Equipment & Aids for the Blind and Visually Impaired* by CCVI Publishing, New York, 1985 provides just such a taxonomy, as of 1985, of devices and software for the visually impaired. IBM maintains the National Support Center for Persons with Disabilities as a clearinghouse for both IBM and non-IBM technologies available to the physically challenged. They can be reached at 1-(800)-IBM-2133. On the USENET electronic news system, the group misc.handicap has ongoing discussions on the suitability of various technologies. Ask your system manager how to access this bulletin board.

About the authors

Bruce McMillin is assistant professor of computer science at the University of Missouri at Rolla. His current research interests include fault-tolerance, multiprocessing systems, parallel algorithms, software engineering, and distributed systems theory. He is a member of IEEE, ACM, and SIAM.

Patricia McMillin is completing a B.A. from Michigan State University and will pursue a master's degree in education for the visually impaired. She also has extensive experience with numerous aids for the visually impaired. □