interpreted as `(A/B)*C`. In all languages, parentheses may always be used to override the precedence rules.

The operator precedence rules together with the rule just stated about parentheses are used to enable the compilation of a program written in a high-level language into machine executable code. This procedure is referred to as the *arithmetic scan*, in which the first step is a transformation that converts normal infix form (i.e. the form in which the operator is placed between its operands as in the examples above) to a Polish form, in which there exist no parentheses and the order of execution of the operators is specified by their positioning. Such a transformation is required because of the difficulty of associating operands with operators in infix notation. As an example, consider the expression

```
(A*X+B)/(C*X-D)                (1)
```

which, because of the precedence relations discussed above, is to be interpreted as

```
(((A*X)+B)/((C*X)-D)).         (2)
```

By use of a classical algorithm that scans across the string in expression (1) from left to right just once, this string can be converted to the Polish postfix string

```
AX*B+CX*D-/                    (3)
```

which, without a need for parentheses or precedence relations, has the unique interpretation of expression (2). With just one more scan across the string, it can be compiled into machine code.

The arithmetic scan described here is a special case of a general syntactic analyzer that uses precedence relationships (*see* COMPILER).

Anthony Ralston

OPTICAL CHARACTER RECOGNITION (OCR)

For articles on related subjects *see* IMAGE PROCESSING; PATTERN RECOGNITION; PERCEPTRON; and UNIVERSAL PRODUCT CODE.

Optical character recognition (OCR) is performed by optical character readers which are automated electronic systems. OCR may be defined as the process of converting images of machine printed or handwritten numerals, letters, and symbols into a computer-processable format. The long history of research in this area, commercial success, and the continuing need and ability to handle less restricted forms of text make OCR the most important application area in machine perception to date.

Two types of automated reading equipment, distinct from optical character readers, are optical mark readers (OMRs) and magnetic ink character readers (MICRs). OMRs characteristically read nontextual input such as bar codes. Examples of OMRs are grocery store bar code readers that read the Universal Product Code (UPC) and the United States Postal Service's wide-area bar code reader that reads ZIP codes encoded in the PostNet code. MICRs classify alphanumeric characters by sensing the pattern corresponding to the magnetic field generated by character ink. One common MICR is a bank check reader that reads account numbers on the bottom of checks.

Commercial OCR predominantly handles machine-printed text. Although neatly printed handwriting is accepted by OCR systems, the technology for handwriting recognition is generally distinct from OCR technology for machine-printed text. Handwriting recognition systems can be divided into two types: online and offline. Online systems allow recognition of characters and words as they are written on a surface. The interactive nature of these systems enables recognition algorithms to use information about how characters are written. Many online systems allow interactive correction of misrecognized characters. Examples of applications using online recognition include the Apple Newton, the 3Com PalmPilot and Cross's Crosspad. Offline systems recognize characters that have been previously written on a document. Therefore no information is available on the writing style or the implement path used to create the character strokes. Examples include address reading machines used by post offices and check amount reading machines.

The following discussion explores the structure, technological attributes, application areas and availability of commercial OCR.

OCR Systems

A typical OCR system (*see* Fig. 1) contains three logical components:

1. Image scanner
2. OCR software and hardware
3. Output interface

The image scanner optically captures text images to be recognized. Text images are processed with OCR software and hardware. The process involves three operations: document analysis (extracting individual character images), recognizing these images (based on shape), and contextual processing (either to correct misclassifications made by the recognition algorithm

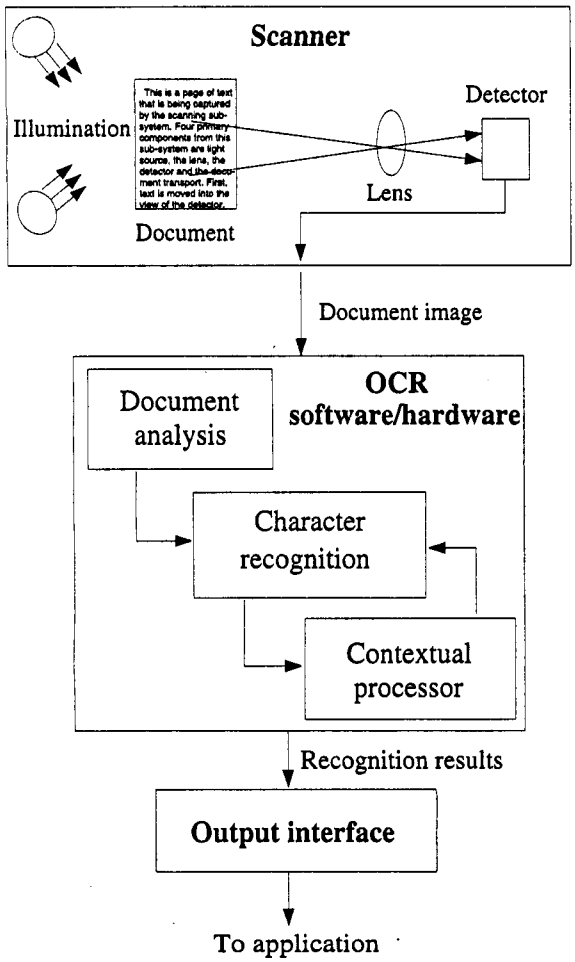

Figure 1. General structure of an OCR system.

or to limit recognition choices). The output interface is responsible for communication of OCR system results to the outside world.

IMAGE SCANNER

Four basic building blocks form functional image scanners: a detector (and associated electronics), an illumination source, a scan lens, and a document transport. The document transport places the document in the scanning field, the light source floods the object with illumination, and the lens forms the object's image on the detector. The detector consists of an array of elements each of which converts incident light into a charge, or analog signal. These analog signals are then converted into an image. Scanning is performed by the detector and the motion of the text object with respect to the detector. After an image is captured, the document transport removes the document from the scanning field.

Recent advances in scanner technology have made available resolution in the range of 600 pixels per inch (ppi) to 1200 ppi. Recognition methods that use features (as opposed to template matching) use resolutions in the range of 200 ppi to 400 ppi, and careful consideration of gray scale. Lower resolutions and simple thresholding tend to break thin lines or fill gaps, thus *invalidating* features.

OCR SOFTWARE AND HARDWARE

Document analysis. Text is extracted from the document image in a process known as *document analysis*. Reliable character segmentation and recognition depend upon both original document quality and registered image quality. Processes that attempt to compensate for poor quality originals or poor quality scanning include image enhancement, underline removal, and noise removal. Image enhancement methods emphasize character versus non-character discrimination. Underline removal erases printed guidelines and other lines which may touch characters and interfere with character recognition, and noise removal erases portions of the image that are not part of the characters.

Prior to character recognition it is necessary to isolate individual characters from the text image. Many OCR systems use connected components for this process. For those connected components that represent multiple or partial characters, more sophisticated algorithms are used. In low-quality or nonuniform text images these sophisticated algorithms may not correctly extract characters, and thus recognition errors may occur. Recognition of unconstrained handwritten text can be very difficult because characters cannot be reliably isolated, especially when the text is cursive handwriting.

Character recognition. Two essential components in a character recognition algorithm are the *feature extractor* and the *classifier*. Feature analysis determines the descriptors, or feature set, used to describe all characters. Given a character image, the feature extractor derives the features that the character possesses. The derived features are then used as input to the character classifier.

Template matching, or *matrix matching*, is one of the most common classification methods. In template matching, individual image pixels are used as features. Classification is performed by comparing an input character image with a set of templates (or prototypes) from each character class. Each comparison results in a similarity measure between the input character and the template. One measure increases the amount of similarity when a pixel in the observed character is identical to the same pixel in the template image. If the pixels differ, the measure of similarity may be decreased. After all templates have been compared with

the observed character image, the character's identity *is assigned* as the identity of the most similar template.

Template matching is a trainable process because template characters may be changed. In many commercial systems, PROMs (programmable read-only memory) store templates containing single fonts. To retrain the algorithm, the current PROMs are replaced with PROMs that contain images of a new font. Thus, if a suitable PROM exists for a font, template matching can be trained to recognize that font. The similarity measure of template matching may also be modified, but commercial OCR systems typically do not allow this.

Structural classification methods use structural features and decision rules to classify characters. Structural features may be defined in terms of character strokes, character holes, or other character attributes such as concavities. For instance, the letter "P" may be described as a vertical stroke with a loop attached to the upper right side. For a character image input, the structural features are extracted and a rule-based system is applied to classify the character. Structural methods are also trainable, but construction of a good feature set and a good rule-base can be time-consuming.

Many character recognizers are based on mathematical formalisms that minimize a measure of misclassification. These recognizers may use pixel-based features or structural features. Some examples are discriminant function classifiers, Bayesian classifiers, artificial neural networks (ANNs) (*see* NEURAL NETWORKS), and template matchers. Discriminant function classifiers use hypersurfaces to separate the featural description of characters from different semantic classes and in the process reduce the mean-squared error. Bayesian methods seek to minimize the loss function associated with misclassification through the use of probability theory. ANNs, which are closer to theories of human perception, employ mathematical minimization techniques. Both discriminant functions and ANNs are used in commercial OCR systems.

Character misclassifications stem from two main sources: poor-quality character images and poor discriminatory ability. Poor document quality, image scanning, and preprocessing can all degrade performance by yielding poor-quality characters. On the other hand, the character recognition method may not have been trained for a proper response on the character causing the error. This type of error source is difficult to overcome because the recognition method may have limitations and all possible character images cannot possibly be considered in training the classifier. Recognition rates for machine-printed characters can reach over 99%, but handwritten character recognition rates are typically lower because every person

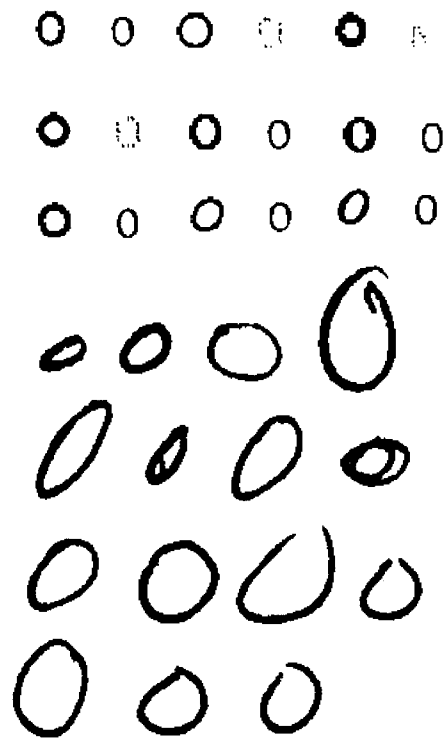

Figure 2. Machine-printed and handwritten capital "O"s.

writes differently. This random nature often results in misclassifications. Fig. 2 shows several examples of machine printed and handwritten capital "O"s. Each capital O can be easily confused with the numeral 0 and the number of different styles of capital "O"s demonstrates the difficulties recognizers must cope with.

ICR. *Intelligent character recognition* refers to the reading of handwritten characters. The term "ICR" was first coined by Kurzweil Computer Products, Inc. in the late 1970s to highlight its own "omni-font" OCR system, which was trainable and could learn new fonts. In the late 1980s this term was usurped to describe the systems developed to read handwritten characters. By its very nature, the variations in handwriting are limitless, so any algorithm which attempts to read handwritten characters will have to be more intelligent than, say, an OCR algorithm which reads only a limited set of fonts. Hence the term "ICR."

A typical ICR algorithm starts by slant-correcting the input image. Unlike machine-printed characters, handwriting is often written with a certain slant that can vary with factors such as the writing instrument, position of the writer, the surface, etc. Slant is corrected by calculating the average angle of slant of the major vertical strokes, and it is corrected by shearing the image.

The next step is size-normalizing the input image. Handwritten characters in general are much bigger than machine-printed characters, as can be seen in Fig. 2. In this step, the input image is reduced (or in the unlikely event that it is smaller than the target size, it is enlarged) to fit a target size, usually something on the order of 32×32 pixels. This entire process is called "preprocessing."

Once the input has been normalized in this fashion, some of the traditional pattern recognition techniques can be applied to recognize the resulting image. ANNs are a common choice for classification, but they suffer from the drawback that sometimes, when the input image is hopelessly complex (for example, when a writer tries to correct a mistake and writes over another character), the network may choose an incorrect classification with high confidence.

Human handwriting is not limited to isolated digits and characters. Quite often ICR algorithms have to cope with touching digits or characters. For example, it is common to write two 0s with a ligature joining them. Some ICR algorithms include modules to segment (separate) touching characters. When such intelligence is built into an ICR system, it is said to be able to read "natural handwriting."

The other extreme in ICR is cursive word recognition. Here, the input is a word (or a phrase) written cursively and naturally, along with a "lexicon" of possibilities. The aim is to find the closest match in the lexicon to the given word image.

There are some systems that claim to perform cursive word recognition. Most of these systems start by performing preprocessing operations like slant normalization. However, sometimes the base of the cursive writing is skewed and that has to be corrected using a skew-removal algorithm.

Some of these cursive word recognition systems work by looking for ligatures joining the individual characters and separating the individual characters that make up the word at these ligatures. The characters are then processed, and the closest match in the lexicon is found.

Other algorithms take a more "holistic" approach, based on the observation that when writing cursive words, people are sloppy and the individual characters cannot be isolated with great accuracy (as shown in Fig. 3). Due to this fact, they try to look for holistic features in the input image, like ascenders (tops of "b," "d," "l," etc.), descenders (bottoms of "g," "p," "q," etc.), loops (like those in "o," "a," etc.), and match the presence or absence and the location of these features with the input lexicon. The closest lexicon entry is the most likely choice.

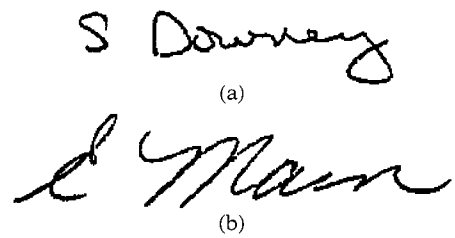

Figure 3. Two cursive words.

Contextual Processing. Contextual information can be used in recognition. The number of word choices for a given field can be limited by knowing the content of another field, e.g. to recognize the street name in an address, first recognize the ZIP code, and then the street name choices can be limited to a lexicon. Alternatively, the result of recognition can be post-processed to correct the recognition errors. One method used to post-process character recognition results is to apply a spelling checker (*q.v.*) to verify word spelling. Similarly, other post-processing methods use lexicons to verify word results or recognition results may be verified interactively with the user. Additional methods to correct or prevent errors using contextual knowledge are state-of-the-art and should appear in commercial systems shortly.

Non-Roman character recognition. Recognition of scripts other than Roman has worldwide interest. There are some 26 different scripts in use today. Some of the scripts have had little work done on their recognition, e.g. Kannada, while a significant amount of work has been done on others, e.g. Japanese. In addition to letters and numerals, Japanese text uses Kanji characters (Chinese ideographs) and Kana (Japanese syllables). Therefore, it is more difficult to recognize Japanese text because of the size of the character set (more than 3,300 characters) and the complexity and similarity of the Kanji character structures (*see* Fig. 4). Low data quality is an additional problem in all OCR. A Japanese OCR system is usually composed of two individual classifiers (pre-classifier and secondary classifier) in a cascade structure. The pre-classifier first performs a fast coarse classification to reduce the character set to a short candidate list (usually containing no more than 100 candidates). The secondary classifier then uses more complex features to determine which candidate in the list has the closest match to the test pattern.

OUTPUT INTERFACE

The output interface allows character recognition results to be electronically transferred into the domain that uses the results. For example, many commercial systems allow recognition results to be placed directly

(a)

(b)

(c)

Figure 4. Difficulties in Japanese character recognition: (a) presence of complex Kanji characters; (b) many characters share the same lexicographical element; and (c) diverse print qualities (each row repeats the same character).

into spreadsheets (*q.v.*), databases, and word processors. Other commercial systems use recognition results directly in further automated processing, and when the processing is complete, the recognition results are discarded. In any event, the output interface, while simple, is vital to the commercial success of OCR systems because it communicates results to the world outside of the OCR system.

Historical Perspective

An ideal model OCR system uses the human eye as the scanner and the human brain as the character recognizer. Accordingly, many of the early developments in OCR technology stemmed from attempts to help visually impaired people to read (*see* DISABLED, COMPUTERS AND THE). According to H. F. Schantz, the first steps towards OCR systems were made in 1809 when the first patents for reading devices to aid the blind were awarded. The first retinal scanner was developed by C. R. Carey in 1870 by using a mosaic of photocells to scan characters. In 1890, P. Nipkow developed a scanning disk that many consider a forerunner of modern television cameras, and in 1912, E. Goldberg converted scanned text into Morse code to be sent over a telegraph line.

Modern OCR technology was born in 1951 with David Shepard's invention, GISMO—A Robot Reader-Writer. In 1954, J. Rabinow developed a prototype machine that was able to read upper case typewritten output at the "fantastic" speed of one character per minute. Several companies, including IBM, Recognition Equipment, Inc., Farrington, Control Data, and Optical Scanning Corporation, marketed OCR systems by 1967. During the late 1960s, the technology underwent many dramatic developments, but OCR systems were considered exotic and futuristic, being used only by government agencies or large corporations. Systems that cost one million dollars were not uncommon.

In the early years of OCR many standards (*q.v.*) were developed to help guide automatic document processing. These standards included:

- Character Set for Optical Character Recognition (OCR-A). ANSI X3.17-81.
- Character Set for Optical Character Recognition (OCR-B). ANSI X3.49-75.
- Paper Used in Optical Character Recognition Systems. ANSI X3.62-87.
- Optical Character Recognition (OCR) Inks. ANSI X3.86-80.
- Optical Character Recognition (OCR) Character Position. ANSI X3.93-81.

The two ANSI-standard machine-printed fonts, along with the ANSI-standard handwritten font, are pictured in Fig. 5.

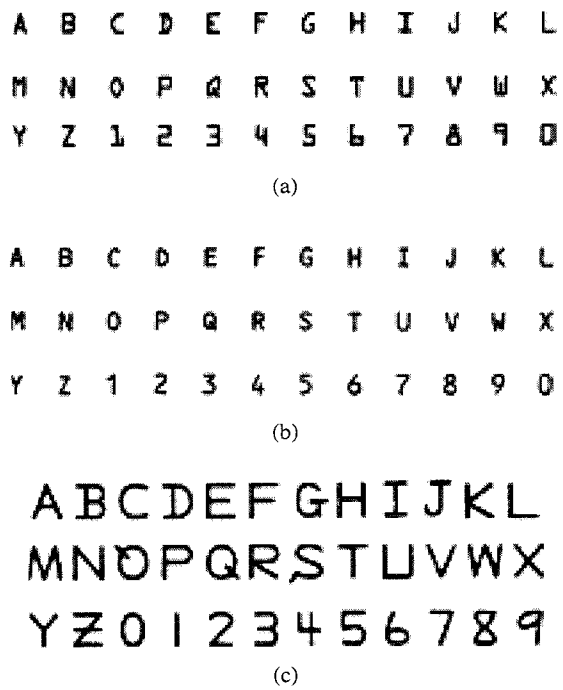

Figure 5. Standardized fonts: (a) OCR-A font; (b) OCR-B font; and (c) handwritten font.

Current OCR systems are cheaper, faster, and more reliable. It is not uncommon to find PC-based OCR packages for under $200 capable of recognizing several hundred characters per second. More fonts than ever can be recognized with today's OCR systems and some systems advertise themselves as omnifont—able to read any machine printed font. Less expensive electronic components and extensive research have paved the way for these new systems. With continued commercial demand for OCR systems, these trends will continue. Increased productivity by reducing human intervention and the ability to store text efficiently are two major selling points.

Current research areas in OCR include handwriting recognition and form "reading." Reliable recognition of handwritten cursive script is now under intense investigation. In addition, research is being conducted into "reading" forms, that is, using all available information to formulate an interpretation of the document. For instance, some United States Postal Service research focuses on assigning ZIP codes to letter images which may not contain any ZIP code. Such an assignment can be made by understanding the various address fields. The use of contextual information in both handwriting recognition and form reading is essential.

Commercial Applications

Hundreds of OCR systems have been developed since the 1950s and many are commercially available today. Commercial OCR systems can largely be grouped into two categories: *task-specific readers* and *general-purpose page readers*. A task-specific reader handles only specific document types. Some of the most common task-specific readers read bank checks, letter mail, or credit card slips. These readers usually use custom-made image lift-hardware that captures only a few predefined document regions. For example, a bank check reader may just scan the courtesy amount field and a postal OCR system may just scan the address block on a mail piece. Such systems emphasize high throughput rates and low error rates. Applications such as letter mail reading have throughput (*q.v.*) rates of 12 letters per second with error rates less than 2%. The character recognizer in many task-specific readers is able to recognize both handwritten and machine-printed text.

General-purpose page readers are designed to handle a broader range of documents such as business letters, technical writings, and newspapers. These systems capture an image of a document page and separate the page into text regions and nontext regions. Nontext regions such as graphics and line drawings are often saved separately from the text and associated recognition results. Text regions are segmented into lines, words, and characters and the characters are passed to the recognizer. Recognition results are output in a format that can be postprocessed by application software. Most of these page readers can read machine-written text, but only a few can read hand-printed alphanumerics.

Task-Specific Readers

Task-specific readers are used primarily for high-volume applications which require high system throughput. Since high throughput rates are desired, handling only the fields of interest helps to alleviate time constraints. Since similar documents possess similar size and layout structure, it is straightforward for the image scanner to focus on those fields where the desired information lies. This approach can considerably reduce the image processing and text recognition time. Some application areas to which task-specific readers have been applied include:

- Assigning ZIP codes to letter mail
- Reading data entered in forms, e.g. tax forms
- Automatic accounting procedures used in processing utility bills
- Verification of account numbers and courtesy amounts on bank checks
- Automatic accounting of airline passenger tickets
- Automatic validation of passports

Address readers. The address reader in a postal mail sorter locates the destination address block on a mail piece and reads the ZIP code in this address block. If additional fields in the address block are read with high confidence the system may generate a nine-digit ZIP code for the piece. The resulting ZIP code is used to generate a bar code which is sprayed on the envelope.

The *Multiline Optical Character Reader* (MLOCR) used by the United States Postal Service (USPS) locates the address block on a mail piece, reads the whole address, identifies the ZIP+4 code, generates 9- or 11-digit bar code, and sorts the mail to the correct stacker. The character classifier recognizes up to 400 fonts and the system can process up to 45,000 mail pieces per hour. The system is shown in Fig. 6.

Form readers. A form reading system needs to discriminate between preprinted form instructions and filled-in data. The system is first trained with a blank form. The system registers those areas on the form where the data should be printed. During the form recognition phase, the system uses the spatial information obtained from training to scan the regions that should be filled with data. Some readers read hand-printed data as well as various machine-written text. They can read data on a form without being confused by the form instructions. Some systems can process forms at a rate of many thousands of forms per hour.

Check readers. A check reader captures check images and recognizes courtesy amounts and account information on the checks. Some readers also recognize the legal amount on checks and use the information in both fields to cross-check the recognition results. An operator can correct misclassified characters by cross-validating the recognition results with the check image that appears on a system console.

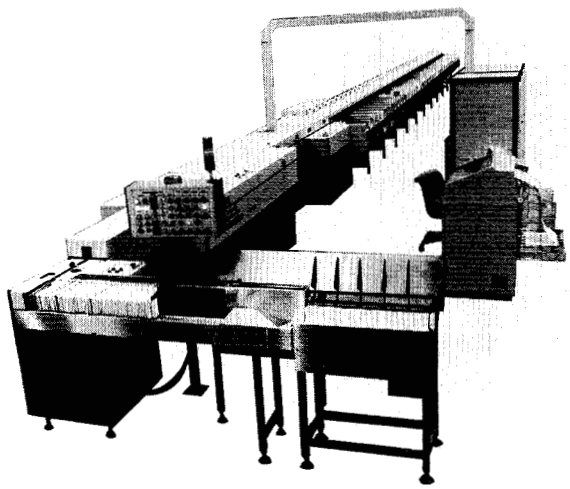

Figure 6. A Multiline Optical Character Reader used by the USPS.

Bill processing systems. In general, a bill processing system is used to read payment slips, utility bills and inventory documents. The system focuses on certain regions on a document where the expected information is located, e.g. account number and payment value.

Airline ticket readers. In order to claim revenue from a airline passenger ticket, an airline needs to have three records matched: reservation record, the travel agent record, and the passenger ticket. However, it is impossible to match all three records for every ticket sold. Current methods which use manual random sampling of tickets is far from accurate in claiming the maximal amount of revenue.

Several airlines are using a passenger revenue accounting system to account accurately for passenger revenues. The system reads the ticket number on a passenger ticket and matches it with the one in the airline reservation database. It scans up to 260,000 tickets per day and achieves a sorting rate of 17 tickets per second.

Passport readers. An automated passport reader is used to speed returning US passengers through custom inspections. The reader reads a traveler's name, date of birth, and passport number on the passport and checks these against the database records that contain information on fugitive felons and known smugglers.

GENERAL-PURPOSE PAGE READERS

There are two general categories of page readers: *high-end page readers* and *low-end page readers*. High-end page readers are more advanced in recognition capability and higher data throughput than the low-end page readers. High-end readers cost about $5,000 or more, and the more expensive ones are bundled into one hardware + software solution.

Low-end page readers usually do not come with a scanner and are compatible with many flat-bed scanners. They are mostly used in an office environment with desktop workstations (*q.v.*), which are less demanding on system throughput. Since they are designed to handle a broader range of documents, a sacrifice of recognition accuracy has to be made. Many scanners today come bundled with free low-end OCR software.

As the speeds of CPUs rise, the distinction between "high-end" and "low-end" machines tends to blur.

Most OCR vendors offer a wide range of solutions to fit every need. In 1999, a Caere Corporation low-end system, the Omnipage Pro, cost about $99, and a high-end Forms/Free Text toolkit, about $5,000. Most of the other vendors, like Expervision, Mitek, NewSoft, and ScanSoft, offer a similar range of products.

Voting systems. A new trend today is to integrate OCR engines from diverse vendors, and perform recognition independently using each of them. The recognition results are then combined ("voted") to yield the final result. The advantage of this approach is that the multiple engines are able to compensate for each other's weaknesses, resulting in a significantly higher recognition rate. PrimeOCR is one such vendor whose main product is a voting OCR system.

Bibliography

1982. Schantz, H. *The History of OCR.* Manchester Center, VT: Recognition Technologies Users Association. (The history of OCR is related from its inauspicious beginnings up to its current commercial success.)

1985. Smith, J. W., and Merali, Z. *Optical Character Recognition: The Technology and its Application in Information Units and Libraries.* The British Library. (This report is intended for use by anyone who is considering OCR in an information or library context. Since minimal knowledge of OCR is assumed, general background material is abundant.)

1990. Adams, R. *Sourcebook of Automatic Identification and Data Collection.* New York: Van Nostrand Reinhold. (This book is a good general reference for OCR. It also considers a number of commercially available OCR systems. Names, addresses, and phone numbers of many OCR vendors are given.)

1999. Rice, S. V., Nagy, G., and Nartker, T. A. *Optical Character Recognition: An Illustrated Guide to the Frontier.* Boston: Kluwer.

Sargur N. Srihari, Ajay Shekhawat, and Stephen W. Lam

OPTICAL COMPUTING

For articles on related subjects *see* FIBER OPTICS; OPTICAL CHARACTER RECOGNITION; OPTICAL STORAGE; and UNIVERSAL PRODUCT CODE.

Digital computing with the use of optical components was considered at least as early as the 1940s by John von Neumann (*q.v.*), a pioneer in electronic computing. If lasers had been available at the time, the first digital computers might well have used optics. Historically, optical technology has found a few special purpose uses as an adjunct technology to electronics for analog and digital computing. Starting in the early 1960s, optical technology has been used for computing (fast) Fourier transforms (FFT—*q.v.*) of military images in matched filtering operations (McAulay, 1991). A simple lens setup realizes a Fourier transform, which maps a two-dimensional image from the space domain to the frequency domain. Aerial views of isolated objects are scanned by an optical/electronic setup that identifies features of interest in the frequency domain. Synthetic aperture radar (SAR) signal processing is an optical pattern recognition application that matches images in stored photographic form with input images at a very high rate. Spectrum analysis is another application that is performed with acousto-optic signal processing. These applications as well as others are performed optically when the need for high bandwidth (*q.v.*) exceeds electronic capability.

There was renewed interest in optical information processing in the late 1970s as advances were made in optical transmission and optically nonlinear materials. The limitations of electronic digital circuits grew increasingly severe as the need for communication bandwidth increased, and attention returned to digital optical computing.

The fastest transistors switch on the order of 5 picoseconds (ps), but the fastest computers have cycle times of the order of 1 nanosecond (ns), 1/200th as fast. This disparity arises from a number of problems related to conventional electronics which include:

- electromagnetic interference at high speed
- distorted edge transitions
- complexity of metal connections
- drive requirements for pins
- large peak power levels
- impedance matching effects

Electromagnetic interference arises because the inductances of two current-carrying wires are coupled. Sharp edge transitions must be maintained for proper switching, but higher frequencies are attenuated more than lower frequencies as an electrical pulse travels through a wire, resulting in sloppy edges at high speeds. The complexity of metal connections on chips, on circuit boards, and between system components affects connection topology and introduces complex fields and unequal path lengths. This translates to signal skews that are overcome by slowing system speeds so that signals overlap sufficiently in time. Large peak power levels are needed to overcome residual capacitances, and impedance matching effects at connections require high currents, which are generated by driver circuits that increase delays between integrated circuits (ICs).

A technology based on optics offers solutions to these problems if the advantages of optics are exploited without introducing new complexities or limitations that render their use ineffective. Advantages of optics include:

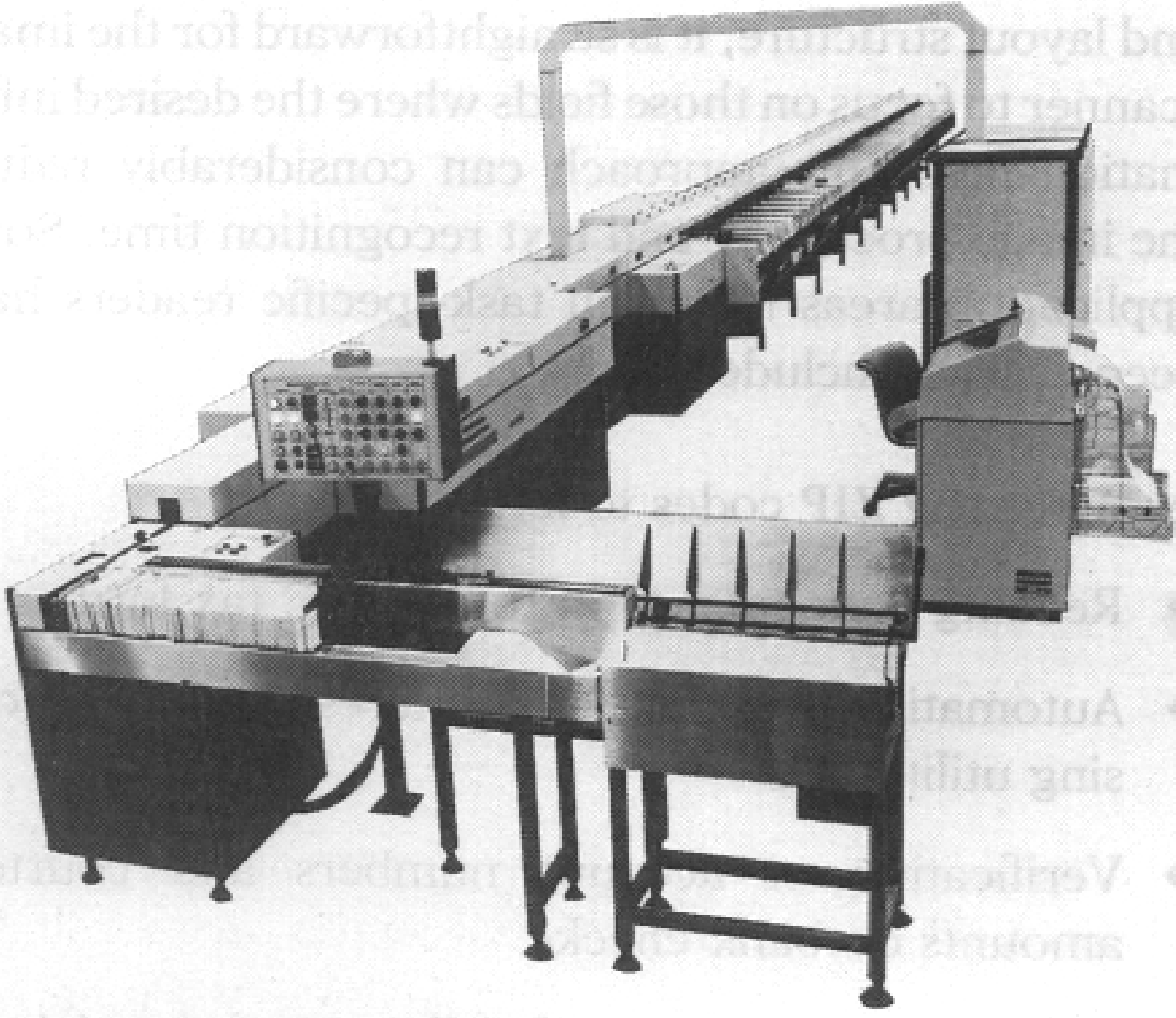

Figure 6. A Multiline Optical Character Reader used by the USPS.