The GRAFIX I image processing system

by ARNOLD K. GRIFFITH

Information International Inc.
Los Angeles, California

INTRODUCTION

The GRAFIX I system was developed in the late 1960's as a fast flexible system for processing and analyzing filmed images, particularly of material which is essentially binary (black and white), such as printed text, line drawings, certain biomedical images, fingerprints, etc. It incorporates a large scale general purpose time shared computer to provide the facilities for the efficient development of algorithms necessary to perform various image processing and analysis tasks. In addition it contains a fast, high resolution flying-spot film scanner and a powerful and rather general slave processor (the binary image processor, or BIP) which provide data collection and manipulation facilities adequate to perform image processing and analysis tasks at commercially practical speeds. So far it has been successfully employed in a commercial environment to the reading of printed multiform text in complex page formats, and to the reading of Cyrillic, Greek and even handprinted text. At present we are considering future applications beyond the area of optical character recognition, particularly the analysis of engineering drawings, as well as the automatic analysis and classification of fingerprints, the analysis of biomedical images such as chromosomes, the analysis of x-ray images, and the analysis of satellite imagery data, among others.

THE GRAFIX I SYSTEM

The main processor in the GRAFIX I is a large, third generation computer with 144,000 words of 36-bit two microsecond memory. It operates under a standard time sharing monitor and incorporates the usual range of text editors, file handlers, assemblers and compilers. Most of the developmental programming is carried out in TRIP, our own interactive on-line compiler language. In addition to the standard peripherals such as magnetic tape drives, "Microtape" drives, disk drive and a high speed line printer, the hardware includes a sonic data tablet for the input of graphical information, a number of teletypes, and five terminals with full keyboards and CRT's which display both images and characters.

The flying spot scanner reads rectangular rasters of density values from filmed images at a rate of up to 500 points per millisecond. The number and spacing of points in both axes, the position and orientation of the scan raster, the dwell time of each point, and the spot size are all under program control. Points on an image are specified in a 15-bit per-axis coordinate system which allows a very precise angular and incremental control of the position of the raster scan relative to the image. The scanner has extremely high positional linearity and repeatability, and resolves about 6000 points across the CRT face. Density values are measured on a range of 0 to 2.5 (100 percent to 0.5 percent transmission) on a logarithmic scale of 512 values.

The slave processor, the Binary Image Processor (BIP), performs a wide variety of processes and measurements on images at core memory limited speeds. It will be described in detail in the next section.

THE BINARY IMAGE PROCESSOR

The GRAFIX I system is designed principally to operate on "binary images," that is, images which are essentially black or white, such as engineering drawings, printed pages, fingerprints or waveform photographs; or images such as bubble, chamber, cell, or chromosome photographs, which for many purposes may be reduced to black and white images without significant loss of information. These images are represented in the GRAFIX I system as arrays of zero- and one-bits in the 36 bit words of the system's main memory. They are manipulated by a special high speed slave processor, the binary image processor (BIP), which has direct access to any part of the main computer memory.

Before discussing the capabilities of the BIP in detail, I shall present a simple paradigm of what is meant by binary image processing in the present sense: Consider a set of $n$ sequential 36-bit words of core memory and let the bits of the $i$th word be indexed by $j$ from 1 to 36, so that the value of the $j$th bit of the $i$th word is represented by $a_{i,j}$. These values form a matrix $[a_{i,j} | i = 1 \ldots n; j = 1 \ldots 36]$ of ones and zeroes. Consider two such matrices $[a_{i,j}]$ and $[b_{i,j}]$, and consider the matrix $[c_{i,j}]$ where:

$$c_{i,j} = F(a_{i,j}, b_{i,j}, b_{i+1,j}, b_{i+1,j+1}, b_{i+1,j-1}, b_{i,j+1}, b_{i-1,j}, b_{i-1,j-1}, b_{i-1,j+1})$$

for some function $F$. Clearly this forms a binary image with...
the same dimensions as those represented by $|a_{i,j}|$ and $|b_{i,j}|$, provided of course that some ad hoc value is assigned to such values as $b_{i-1,j}$, etc. Considered as a transformation or combination of images, the production of this third image from the first two is local in the sense that the value of some point in the result image is dependent only on the corresponding point in the first image and the "local neighborhood" (of radius 1) of the point in the second image. Now if this transformation were purely local in the sense of a neighborhood combination of images, the production of this third image would be quite uninteresting, since actually there would be only sixteen possible functions corresponding to "or-ing," "and-ing," negating the first image irrespective of the second, etc. However, the inclusion of neighborhood bits of the second image as arguments to a possible image processing function allows a total of $2^{32}$ possible functions!

The BIP is in most respects more general than the paradigm just described. It has addressing features which allow either image to occupy only a predetermined subset of contiguous bits within an image; and it is relatively easily programmed to allow the image height to be greater than 36 bits. The major respect in which the BIP is not as general as the paradigm discussed is in that there are only 33 bits in the control field specifying the function, and hence only $2^{33}$ possible functions. The set of functions is by no means an arbitrary subset of all possible functions. It was in fact chosen on the basis of a rather extensive theory and practical considerations worked out by Gray and others, for the purpose of providing a rich variety of functions, as will be seen later in this section.

A second major area of BIP function not mentioned by the paradigm is that of global measurements. In the process of generating a result image the BIP can compute the area, i.e., number of ones; together with the number of configurations of subsets of the result image of the form $c_{i,i}$, $e_{i,i+1}$, $c_{i+1,i}$, $c_{i+1,i+1}$ (i.e., subsquares of the result image) in which all four bits are one, in which three bits are one, in which non-diagonal pairs of bits are one, and in which only one bit is a one. From these values and others, again according to the theory worked out by Gray, the Euler number (number of objects minus the number of holes), the approximate line width, as well as a number of other global features, such as average slant of lines, may be calculated. A final global measurement of this sort is the area of the "exclusive-or" of two images, i.e., the area in which they are not alike, which is a measure of similarity. A single pass over a pair of images not only computes this area of dissimilarity but also simultaneously computes eight similar values corresponding to one of the images being displaced relative to the other by one unit in each of eight possible directions.

An additional feature of the BIP is a limited capability to work with arrays of integral values, not just one or zero. The two images to be processed consist of six bit bytes, packed six per word, and would produce a binary image with ones where one of the corresponding pair of values exceeds the other, and zeroes elsewhere. If one image is an array of constant values, then this process amounts to thresholding the other image at that value. Having one image of smoothly varying value allows position dependent thresholding.

The purpose of the BIP is to serve as a special purpose slave processor to perform inner loop tasks in image processing operations at high speeds (40 MHz), leaving system control, decision making tasks, etc., to the general purpose computer operating the system. The BIP operates at speeds of up to 1000 times that possible on a standard high speed computer. Its use of modern integrated circuits and pipeline construction allows it to run at speeds of 25 nanoseconds per image point. A process operating on an array image of 36 bits high runs at memory limited speeds even in a one microsecond memory. The speed of complex processes is proportional to the area of the image and to the number of passes required to perform it.

**MAN COMPUTER SYMBIOSIS IN THE READING OF COMPLEX PRINTED PAGES**

Recently the GRAFIX I has been employed to perform a number of major commercial text conversion (OCR) tasks. It has been our experience that "real-world" text conversion does not amount simply to the character by character or word by word recognition of printed text. Rather it has included the necessity of recognizing certain typographical features of the pages such as the location of text relative to illustrations or the structure of text within tables. In addition such tasks often require that the recognized text be broken into fields according to content and sometimes suitably reformatted in instances when the converted text is subsequently manipulated by an information retrieval system, or output and reformatted by a computer typesetting device. Generally it is beyond the state of the art in artificial intelligence to perform all of these tasks automatically.

We have structured our text conversion systems to perform the various format recognition, character recognition and output formatting tasks as automatically as possible. These text processing systems, however, are programmed to be "aware" of their own limitations, and to ask for human help when unable to cope with the complexities of the data with which they are presented. We have found this technique of "man computer symbiosis" to be an extremely powerful one, providing as it does an efficient division of labor between man and computer.

Conversion by GRAFIX I of technical manuals affords a good illustration of a text conversion process in which typographical features of the text must be taken into account. Figure 1 illustrates a typical page which the system converted, in its entirety, in a recent procurement benchmark test. Information critical to successful reading of the page included the location of the heading and page numbers at the top and bottom left, the locations of the various rules of the table at the top of the page, the locations of the tab stops in the three passages of tabulated material in the running text columns, and the locations of the portions of
TABLE XIII. VERTICAL GYRO TROUBLESHOOTING (Cont)

<table>
<thead>
<tr>
<th>Step</th>
<th>Trouble</th>
<th>Probable Cause</th>
<th>Remedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Rundown time is Less Than Minimum Value.</td>
<td>Defective gyro motor</td>
<td>Check motor windings and circuit leads. Replace defective motor or repair wiring.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incorrect end play in motor shaft</td>
<td>Check and adjust end play.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unbalanced flywheel</td>
<td>Check balance of wheel and rotor assembly; balance flywheel.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Damaged motor shaft bearings</td>
<td>Check bearings; replace motor.</td>
</tr>
</tbody>
</table>

### AC Meter Selector

<table>
<thead>
<tr>
<th>AC Meter Selector</th>
<th>Other</th>
<th>AC Meter</th>
<th>DC Voltmeter Tolerance (volts dc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRECESSION MODE SELECTOR:</td>
<td>Platform Angular Displacement</td>
<td>Minimum</td>
<td>Maximum</td>
</tr>
<tr>
<td>&quot;PITCH&quot; NU ND</td>
<td>12.6</td>
<td>12.6</td>
<td>Null output N/A 100 mv</td>
</tr>
<tr>
<td>&quot;ROLL&quot; RWU RWD</td>
<td>12.6</td>
<td>12.6</td>
<td>30 minutes 2.0 4.5</td>
</tr>
</tbody>
</table>

### Switch Position

<table>
<thead>
<tr>
<th>Switch</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC METER SELECTOR</td>
<td>115V ADJ.</td>
</tr>
<tr>
<td>MODE SELECTOR</td>
<td>AUTO</td>
</tr>
<tr>
<td>PRECESSION MODE SELECTOR</td>
<td>OFF</td>
</tr>
<tr>
<td>DC METER SELECTOR</td>
<td>PITCH GYRO OUTPUT</td>
</tr>
</tbody>
</table>

11-8. PITCH OUTPUT TEST. To check the pitch output voltage of the gyro for the nose up and nose down attitudes, proceed as follows:

a. Set the DC METER SELECTOR switch to "PITCH" position.

b. Rotate the gyro mounting plate until the oscillator connector of the gyro faces due north.

c. Adjust angular setting dial until the dc meter indicates null voltage (minimum). The gyro mounting plate must not be out of level by more than 30 minutes.

d. Utilizing the platform angle adjusting block and angular setting dial, displace the gyro to the indicated attitudes, and check voltage outputs, as follows:

e. Rotate gyro mounting plate until the oscillator connector of the gyro faces in a south direction.

f. Repeat steps c. and d.

g. Setting the gyro in an NU or RWD direction shall indicate a positive output voltage. Setting the gyro in an ND or RWU direction shall indicate a negative output voltage.

h. Re-null the gyro after each set of voltage output readings.

i. Changing the gyro angle settings from NU to ND or from RWU to RWD is accomplished by rotating the gyro 180 degrees.

11-9. ROLL OUTPUT TEST. To check the output voltage of the gyro for the right wing up and right wing down attitudes, proceed as follows:

a. Set DC METER SELECTOR switch to "ROLL" position.

b. Rotate gyro mounting plate until the oscillator connector of the gyro faces in a west direction.

c. Repeat procedure outlined in paragraph 11-8, steps c. and d.
HEURISTICS

The exploitation of the technique of man-computer symbiosis is a direct consequence of our use of a full sized computer with time sharing and extensive system software. The development of the last application, for example, took only three man months of programming. Besides facilitating the development of these procedures, the full scale system was necessary for their post-development implementation.

It should not be inferred from the previous section that the only way "intelligent" behavior is manifested by the various GRAFIX I OCR programs is by receiving the appropriate instructions from a human operator. On the contrary, many of the various OCR sub-processes operate autonomously, performing quite intricate analyses with no provision whatever for human interaction and prompting. These procedures are carried out by elaborate heuristic programs developed by the staff which are constantly undergoing further development and refinement. The use of extensive heuristic software was a conscious part of the original design of GRAFIX I; and the rapid design and development of such programs is made possible by the use of a time shared computer with extensive software support.

An example of our use of heuristic procedures is the hand-printed character recognizer currently under development by the present author.  Most handprinted character recognition procedures currently described in the literature tend to be more "algorithmic" than heuristic. Typically a set of numerical "features" or some sort of formal description is derived from a character image by an analysis which performs a large amount of calculation independently of the nature of the character image. This feature vector, or formal description, is then given to a decision algorithm which decides on the identity of the character (or possibly rejects it as unrecognizable) by means of a linear decision procedure or a feature vector table lookup process, or a formal linguistic analysis in the case of a formal description. Such procedures often have the disadvantage, when applied to data as variable as handprinted characters, to occasionally yield wildly unpredictable results. In addition, attempts to modify such algorithms to perform better in certain respects often lead to a decrease in performance in other respects. (This, and other arguments are particularly applicable to the linear decision or "perceptron" technique, see Minsky and Papert.)

Although a number of systems organized along these lines have produced respectable results, the present system, in an attempt to circumvent some of these difficulties, has been designed and organized along purely heuristic lines. In particular it is composed of a large number of simple, carefully designed, and well-understood sub-tests. These are arranged into a hierarchy in such a fashion that tests at various levels are usually only performed when relevant or "necessary," and so that the whole structure may be quite simply and selectively modified to improve in a particular respect without impairing its performance in other respects. On one sample of about 13,000 mixed alphanumeric characters (26 letters and 10 numerals), printed by untrained clerks, it achieved a reject rate (unable to recognize) of around 4 percent, and a substitution rate (misidentification) of around 0.3 percent.

Numerous other examples of heuristic procedures are embodied in the GRAFIX I machine printed text reading system. Two examples discussed in the previous section, although involving human interaction, include quite elaborate heuristics to perform those portions of their respective tasks that are performed independently of the human operator. The procedure employed in the technical manual conversion process to read text in tabular form, after receiving information from a human operator as to the location of the rules in the table, must re-find these rules more exactly on the
filmed image, and employs a rather elaborate line and character finding procedure to properly read the material within each ruled box. In the case of the fielding and formatting of patents, a number of heuristics are used to tentatively locate each item in the converted patent text before the human operator is consulted. In most cases the heuristics produced a correct result; and the operator's only task was to indicate correctness. A number of other procedures from the basic OCR system, such as "page finding," "line finding" and character finding similarly rely heavily on heuristics. Some of these procedures will be discussed in the next section in connection with the concept of heterarchical organization.

HETE RARCHY AND HIERARCHY

The OCR system currently implemented on the GRAFIX I system is principally hierarchical in structure. The first step in processing a frame of text consists of finding the outline of the page image on the film. The text on a page is considered to be composed of a number of "fields" consisting of one or a number of lines of text, a title, an entry in a piece of tabulated material, etc. Information about the fields of a particular page is supplied from a data file of "descriptors" entered either manually or with a data tablet. Each field is read line by line and each line is read character by character. Each line is first found, then read; and within a line each character is first found and then read.

The procedure just described is a prototypical hierarchy, where each step of the processing depends on the accuracy of decisions of the previous step. For example once a field is found, it is assumed that the finding procedure is correct, and the success or the failure of the finding and reading of the individual lines is dependent on the accuracy of the field finding data. No attempt is currently made to re-find the field if attempts to find the constituent lines within it meet with failure.

Recent research in the analysis of complex scenes (e.g., [10,11]) has explored a more general approach than strictly hierarchical organization in dealing with real world or complex images. Clearly it is an essential limitation for information pass from level to level in one direction only, as for example in the case of scene analysis, from local edge detection to edge line and contour detection, to detection of simple forms, to detection and recognition of complex objects. In the case of optical character recognition of text on elaborately structured pages, this limitation might take the form, as previously mentioned, of making a "one-shot" decision as to the location of a field of text and then forcing the line and character finder to accept this without complaint, without letting a failure in line finding pass back to the field finder, if necessary, to force a re-find of the field. Some of these limitations are circumvented by a more general approach to program organization, termed "heterarchical," which allows a two way interaction between levels to occur. In effect the program interacts with the data at hand instead of just analyzing it step by step to higher levels of abstraction or in greater levels of detail.

Heterarchical program organization has been employed to advantage in a number of sub-processes of the present GRAFIX I OCR system:

1. The image of a line of text is scanned from left to right in a series of "segments" of about seven characters in length. Given the proper location of the first segment of a line, the approximate locations of subsequent segments for that line may be inferred using information as to the orientation of the page and the orientation of other lines of text on the page. However lines of text are sometimes not exactly straight nor are they always oriented exactly parallel to each other. Furthermore, in the case of the first line on a page, the only information is the orientation of the borders of the page, which is a poor predictor of the orientation of the lines of text on it. Consequently it is generally impossible to scan a line of text segment by segment using only global orientation information and the correct location of the left end without risking cutting off the tops or bottoms of characters within certain segments. Our approach to this problem is as follows: In the course of recognizing the characters from a particular segment, the positions of the baselines of these characters relative to the bottom of the segment are computed for each character recognized. The baseline is a function not of, e.g., the bottom of the character image, but is computed after recognition on the basis of a fiducial line within the character (often at the bottom, but not always) whose location is dependent on the identity of the character. The difference between some predetermined constant and this baseline-to-segment-bottom distance is then fed back to the scanner and used to define the exact vertical location of the next scan segment.

2. The process of reducing a gray level image as obtained from the scanner into a binary image to present to the recognition procedure is accomplished by thresholding the gray level image at some clip level. Because of blurring and other effects, the appearance of the resulting binary image, and hence the recognizability of the characters it contains, is affected by this clip level. Due to variabilities of the image resulting from such factors as variations in the reflectance of the ink and paper of the original document, it is in general impossible to choose a single threshold which will be appropriate for an entire page image. Our approach has been to initially threshold a new image at the same threshold which was successful for the preceding segment. In the course of subsequent analysis, the apparent linewidth of the characters in the segment is calculated. If the width is outside certain tolerances, no further recognition analysis is performed on the binary image of the segment. Instead the information regarding the degree to which linewidth is outside
From the collection of the Computer History Museum (www.computerhistory.org)

tolerances is fed back to the thresholding procedure, and the gray level image of the segment is re-thresholded accordingly. This procedure may be repeated several times.

(3) In certain instances a line of text may be entirely unrecognizable due to being in an unexpected font, or being systematically degraded along its entire length. Such situations are generally impossible to predict a priori; but when they are encountered it would be a waste of effort to apply the recognition process to every character in the line. This is especially the case because with unrecognizable material, the recognition procedure has "tried everything" before it gives up and may expend ten times as much effort on a rejected character as on a recognized one. To overcome this problem the GRAFIX I OCR system has a provision whereby if a certain number of rejects is encountered in a line, the line is re-read using a different set of character masks, or is abandoned if no other sets of masks remain to be tried. To see how this procedure may be considered to be hierarchically organized, consider the task of the recognition of all the characters in a particular line of text. The restart of this task with a new "font," or the abandonment of the task if all fonts have been tried, is essentially a higher level decision based on information (the failure of the lower level process of recognition) being passed up to it, which in turn affects other lower level processes (the further recognition of characters within the line).

FUTURE DIRECTIONS

The GRAFIX I was conceived as an image processing system of considerable versatility. Its application to date principally as an optical character recognition (OCR) device has been prompted by the commercial promise of this area. Having realized a substantial portion of the potential of the GRAFIX I system in OCR, we are considering further applications, particularly in the areas of the processing of essentially binary images. An area in which we hope to become involved in the near future is the intelligent reading of engineering drawings and technical diagrams. Recent work of the present author10-12 and others at Stanford,13 SRI and the University of Edinburgh in the analysis of scenes consisting of prismatic solids, appear to be adaptable to the problem of efficiently extracting the component lines from these drawings so as to represent them in a reasonably compact and updatable form. Other possible areas include: biomedical image processing such as cell counting, chromosome analysis, etc.; high volume metallurgical image processing such as fibre measurement, inclusion counts, grain size, etc.; automatic inspection of x-ray photographs and photographs of manufactured parts for defect analysis; the analysis of fingerprints; and others.

ACKNOWLEDGMENTS

The author is particularly indebted to Stephen Gray for aid in the preparation of this paper. In addition the following, among others, have significantly contributed to the design and implementation of the GRAFIX I: Russell Ham, Dick Martin, Dan Forsyth, and Ed Fredkin.

REFERENCES


From the collection of the Computer History Museum (www.computerhistory.org)