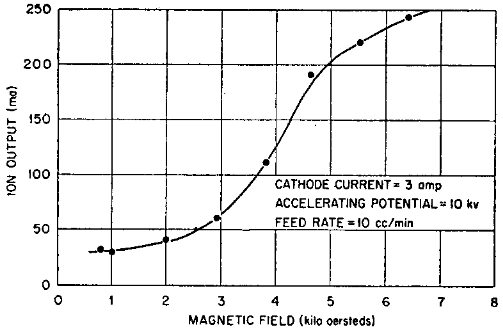

FIG. 6. Ion output *vs* magnetic field.

The data as shown on the curves and described above are consistent and reproducible to within 10 percent. Greater variation may be expected if the geometry of the cathode position is changed from one assembly to the next.

There have been no equipment failures in test operations of normal length. In a stability and life test, the source was operated continuously for 65 hr at an ion output of 175 to 200 ma. Failure at the end of this period was caused by discharge from the cathode support to the carbon chamber, which resulted in sputtering of the support. Very few adjustments were required for the entire 65-hr period. All insulation and other materials were in perfect condition after the test.

To determine the relative magnitude of plasma oscillations, a voltage signal was obtained from a resistor placed in series with the cathode supply. Under normal operating conditions the "hash" characteristic amplitude was only 4 percent of the total voltage signal.

CONCLUSION

This high intensity ion source has operated satisfactorily with sufficient proton output and efficiency. Since the present design yields an ion output exceeding immediate needs, no further attempt has been made to explore the limits of output obtainable. The ruggedness of construction and stability of operation make it a dependable ion source.

ACKNOWLEDGMENTS

This work was prompted and supported by R. S. Livingston. G. Leichsenring and R. Reinhardt assisted materially in the designing and testing of the ion source.

THE REVIEW OF SCIENTIFIC INSTRUMENTS VOLUME 25 NUMBER 12 DECEMBER, 1954

Programming a Digital Computer for Cell Counting and Sizing*

WALTER WELKOWITZ†
Bioacoustics Laboratory, University of Illinois, Urbana, Illinois
(Received June 10, 1954)

A method is presented for programming a digital computer, used in conjunction with a television scanning system, to count and size solidly and contrastingly stained cells of different shapes and sizes as found on slides prepared for histological study. When the operating times of the program are considered, it is found that machine counting and sizing is very much faster than a manual performance of the same operations. A program prepared in accordance with the method presented operated satisfactorily on the Illiac digital computer.

INTRODUCTION

IN recent years there have been a number of successful attempts to apply automatic methods to microscope counting operations.[1,2] In these approaches a television scanning apparatus is incorporated into the microscope system and, with appropriate counting circuitry, counts of cells or particles of uniform shapes and sizes can be obtained at a very rapid rate. The methods used encounter difficulties when it is desired to count and find the size distribution of cells of greatly irregular shapes. In this paper a method is proposed of operating on the output obtained from a television "flying spot" system with a digital computer to obtain accurate information regarding the number and size distribution of cells in the microscope field regardless of the shape of the cells.

PROGRAM CONSTRUCTION

For the purpose of this discussion, a field in the microscope is considered which is populated with cells stained with sufficient contrast with respect to the background to produce a light intensity difference that can be resolved by a photomultiplier tube. Nerve cells in such a field can cover the size range from about 3 microns to about 30 microns in largest diameter. It is assumed that the scanning light beam, after transmission through the microscope lens system, is less than half the size of the smallest cell (about 1 micron

* This work was supported in part by the Physiology Branch of the Office of Naval Research under contract Nonr 336 (00)Bio-075.
† Present address: Acoustics Laboratory, Columbia University, New York, New York.
[1] F. Roberts, and J. Z. Young, Proc. Inst. Elec. Engrs. 99, 747 (1952).
[2] L. E. Flory and W. S. Pike, RCA Rev. 14, 546 (1953).

is reasonable for the beam diameter) and that it scans across the microscope field line by line. It is further assumed that the output of the physical system consists of a series of pulses of uniform height appearing timewise when the scanning beam is interrupted by a cell. For the purposes of analysis each scan line is considered broken up into 300 equal size spots. If ten scan lines are analyzed, we have a matrix of spot areas 300 by 10. If each spot area is considered to be one micron square, then the field corresponds roughly to a region in a microscope slide section of the gray matter of a cat brain which contains 100 nerve cells. The programming method involves keeping track of all the 300 spots scanned on a line and appropriately relating the ones that contain parts of cells. After taking the computer memory size into account it was decided to make the analysis continuous rather than a "batch" process and to consider only two lines of scanned information at a time. The first program was set up to obtain the number of cells in each of five area groups which were chosen as the following multiples of the beam area: >135, 135–45, 45–15, 15–5, and 5–1. Both the group sizes and the number of lines in a field are readily altered parameters of the program.

With the preliminary information established, it is possible to analyze the problem in detail. It breaks down essentially into making decisions on contiguity of spots. Consider Fig. 1, in which a cell and some scan spots on two lines of scan are diagrammed. Scanning takes place from left to right on each line, with the lines scanned from top to bottom. Spot x is first examined to see if it contains a 0 or 1. (The method used to feed in blank or filled spots into the computer memory consists in putting either a 0 or 1 in the appropriate memory location for each spot.) This decision can be notated as

$$x\cdot 1=1$$

$$\text{or}\quad x\cdot 1=0.$$

If $x\cdot 1=0$, the next spot is considered. If $x\cdot 1=1$, then $x\cdot a$ is examined. This can also be 0 or 1. If $x\cdot a=1$, we extract the location of the cell counter from the 40 bit word in the memory characterizing spot "a." We then place this counter address of "a" into the "x" word, increase the counter by 1 and go on to $(x+1)$.

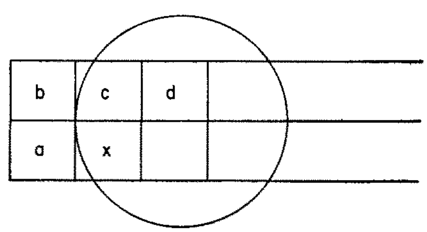

FIG. 1. Diagram of a cell intersected by two scan lines.

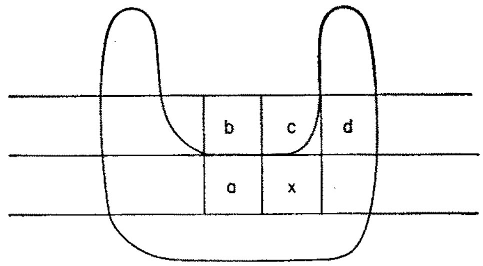

FIG. 2. Diagram of a concave cell intersected by two scan lines.

If $x\cdot a=0$, the process is repeated with "b." If

$$x\cdot a=0$$
$$x\cdot b=0,$$

we proceed to "c," and if

$$x\cdot a=0$$
$$x\cdot b=0$$
$$x\cdot c=0,$$

we proceed to "d." If

$$x\cdot 1=1,$$

$$\text{but}\quad x\cdot a=0$$
$$x\cdot b=0$$
$$x\cdot c=0$$
$$x\cdot d=0,$$

then x is considered to be the start of a new cell. A new cell counter number is obtained from the bin of cell counter numbers, its address is placed in the "x" word, and its count is started off with a 1.

The process described seemed reasonable until cells were considered which were concave upward or contained deep indentations on the left side. As is apparent from Fig. 2, the computer will start counting such cells as two cells—a very undesirable state of affairs. In order to coalesce the two cells, the following modification of the programming procedure is required. If $x\cdot 1=1$ and $x\cdot a=1$, then $x\cdot d$ is examined. If $x\cdot d=1$, then the cell counter addresses in the "a" word and the "d" word are compared. If they are the same, the previous procedure is followed. If they are different, though, the address of the counter in the "d" word is changed to that in the "a" word, the contents of the "d" word counter are added to those of the "a" word counter and a 0 is placed in the "d" word counter. The "a" word counter is then increased by 1 and its address placed in the "x" word. Spot $(x+1)$ is then considered. If

$$x\cdot 1=1$$
$$x\cdot a=0$$
$$x\cdot b=1$$

the same modification is used. The modification is unnecessary if $x\cdot b=0$, or if $x\cdot c=0$,

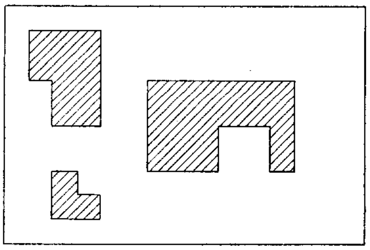

FIG. 3. Pattern used to test the computer program. This pattern when used was repeated 20 times horizontally. Twenty-two percent of the area is covered by the sixty cells.

After operating on the 300 spots in a line, a new line is fed into the computer memory. In order to avoid time consuming line shifting inside the machine, the new line is fed into the position of line $(n-2)$. To take care of the needed address changes in the program that this entails, a duplicate program is used. Line ending spots are keyed by having 1's in the appropriate memory positions. This avoids the necessity of including line ending considerations in blank spot considerations.

When a complete field of lines (ten in the problem under consideration) has been investigated, the desired partial output information is obtained by considering each size group in turn and sorting through all the cell counters. A one is totaled in that size group for each cell of the appropriate size. The five size group contents are then printed out.

PROGRAM OPERATING TIMES

It is apparent that the system described in the discussion is useful only if the operating time is short compared to the time required to do the same job manually. An estimate of the time required for analysis of a ten line field was carried out using the order execution times of the Illiac computer. This time is made up of the individual spot operation time, line ending and line feeding time, and field ending and print out time. The estimate made is based on the consideration of a field with 20 percent of the area covered with cells. It is further assumed that the filled spots are equally divided among the five classes of spot operations considered. On this basis, for electronic feed into the computer, operation on a field is estimated to take 10 seconds. If tape input is substituted for electronic input about 20 seconds is required.

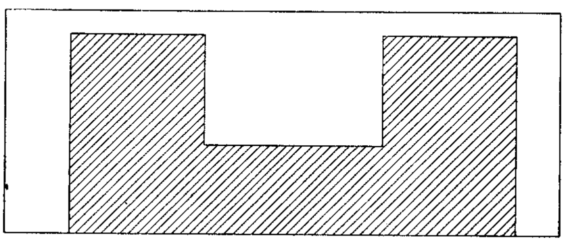

FIG. 4. Pattern used to test the computer program. This pattern when used was repeated 12 times horizontally. Fifty-six percent of the area is covered by the twelve cells.

A check of the operation of such a program has been obtained by feeding dummy patterns into the Illiac computer on tapes. As previously suggested, these patterns were made up of fields of ten lines of 300 spots each. The patterns that were used are indicated in Figs. 3 and 4. The machine required about 30 seconds to take in the problem and type out the answer in both cases. This is considered to be in fairly good agreement with the estimated value of about 20 seconds.

Counts that have been made on Marchi stained sections of nerve fibers of the pyramidal tract of a cat spinal cord can be used to obtain a value for the time involved in manually counting particles on slides. A slide with about 900 degenerate fibers (stained jet black) covering about 15 percent of the area required one-half day of intensive work to obtain a complete count. This count included all stained fibers above a certain minimum diameter with no attempt made to size sections in any other fashion. It is estimated that by using the Illiac computer with an electronic input, one could count and size all these fibers in about 4 minutes. If the human operator were to carry out as complete an analysis, it is apparent that the machine would readily operate at more than 100 times the manual rate. Machine operation is thus practical from the point of view of the operating times involved.

CONCLUSIONS

A method has been presented for programming a digital computer used in conjunction with a television scanning system, to count and size solidly (no holes) and contrastingly stained cells or particles of different shapes and sizes as found on slides prepared for histological study. This program is useful when one considers the operating times involved. A "flying spot" microscope system as described by Roberts and Young[1] might be used as the physical apparatus for scanning. The output of such a system, suitably chopped into pulses, can be fed electronically into the digital computer. Such a composite system would be of great value in obtaining quantitative data about nerve cell sizes and their distribution in the central nervous system.

ACKNOWLEDGMENT

The author thanks Professor W. J. Fry for suggesting this problem and the members of the Computer Laboratory at the University of Illinois for help in testing the program on the Illiac computer.