1 copy
6-26-80
MMB

MASTER

ANL/AMD/TM-64

AROMA-AIRWICK
A CHLOE/CDC-3600 SYSTEM
for the
AUTOMATIC IDENTIFICATION OF SPARK IMAGES AND
THEIR ASSOCIATION INTO TRACKS

by

Robert K. Clark

U of C-AUA-USDOE

ARGONNE NATIONAL LABORATORY, ARGONNE, ILLINOIS

Operated under Contract W-31-109-Eng-38 for the

U. S. DEPARTMENT OF ENERGY

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

fly

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

The following pages are an exact
representation of what is in the original
document folder.

The facilities of Argonne National Laboratory are owned by the United States Government. Under the terms of a contract (W-31-109-Eng-38) among the U. S. Department of Energy, Argonne Universities Association and The University of Chicago, the University employs the staff and operates the Laboratory in accordance with policies and programs formulated, approved and reviewed by the Association.

MEMBERS OF ARGONNE UNIVERSITIES ASSOCIATION

The University of Arizona
Carnegie-Mellon University
Case Western Reserve University
The University of Chicago
University of Cincinnati
Illinois Institute of Technology
University of Illinois
Indiana University
The University of Iowa
Iowa State University

The University of Kansas
Kansas State University
Loyola University of Chicago
Marquette University
The University of Michigan
Michigan State University
University of Minnesota
University of Missouri
Northwestern University
University of Notre Dame

The Ohio State University
Ohio University
The Pennsylvania State University
Purdue University
Saint Louis University
Southern Illinois University
The University of Texas at Austin
Washington University
Wayne State University
The University of Wisconsin-Madison

NOTICE

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use or the results of such use of any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights. Mention of commercial products, their manufacturers, or their suppliers in this publication does not imply or connote approval or disapproval of the product by Argonne National Laboratory or the United States Government.

TABLE OF CONTENTS

	<u>Page</u>
SECTION I - Introduction.	1
SECTION II - The CHLOE Film Scanning Equipment	6
SECTION III - The CDC-3600 Computer	11
SECTION IV - AROMA	12
SECTION V - AIRWICK	25
SECTION VI - PREPROS	28
SECTION VII - PAIR.	29
SECTION VIII - LINK.	40
SECTION IX - References.	46

LIST OF FIGURES

<u>No.</u>	<u>Title</u>	<u>Page</u>
1A	View 1 of a stereo pair taken by the Argonne group at CERN . . .	2
1B	View 2 of a stereo pair taken by the Argonne group at CERN . . .	3
2	Format of film to be taken in Ke_2 experiment	4
3	Format of film to be taken by Argonne Magnetic Spark Chamber Group.	5
4	Schematic of the CHLOE System.	8
5	Schematic of the CRT Scanner	9
6	Photograph of the CHLOE Equipment.	10
7	CDC-3600 System at Argonne National Laboratory	11
8	Block diagram of AROMA	12
9	Block diagram of AIRWICK	27
10	Linking when NTREE = 1	44
11	Extrapolation of Links	45

DISCLAIMER

This book was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

FIGURE 1a

FIGURE 1b

FIGURE 2

FIGURE 3

SECTION II

THE CHLOE FILM SCANNING EQUIPMENT

CHLOE is an automatic film scanning equipment built by the Applied Mathematics Division for digitizing information from sprocketed 35 mm. film. The equipment consists of two main components: a digital computer and an optical scanner controlled by the computer. The computer is the commercially available Advanced Scientific Instruments ASI-210, a 21-bit word length computer having an 8K memory with a 2-microsecond cycle time and provided with two buffered input/output channels. The optical scanner was built at ANL. See Figure 4, Figure 5, Figure 6, and references 9, 10, 11, and 12.

The optical scanner makes use of two cathode ray tubes as light sources. The spot from the CRT is projected onto the film, and light transmitted through the film is viewed by a photomultiplier, from which a decision is made concerning the density of the film in question. A given scan covers a rectangular portion of a 4096 x 4096 raster which is $1\frac{1}{4}$ x $1\frac{1}{4}$ inches in size. The transmission of data to the memory of the computer is fully buffered.

The programmer has the following options:

1. He can specify the x-low, x-high, y-low, and y-high coordinates of the rectangular area to be scanned. This area can be squeezed in one direction to a line and in two directions to a point.
2. He can specify which of the two scanners is to be set in operation.
3. He can specify the mode of the scanning:
 - a. Upward scans from left to right.
 - b. Rightward scans from top to bottom.
4. He can specify the number of points (in either direction) to be skipped between scanned points. This number can be $2^k - 1$ where $k = 0, 1, 2, 3, 4, 5, 6$, or 7.
5. He can specify whether information is to be found as black on a white background or as white on a black background.
6. He can specify any of eight predetermined photomultiplier response levels which must be attained or dropped below before a point is transmitted to the computer.
7. He can start the scanner.
8. He can allow the scanner to begin inputting data.
9. He can display the results of the scan on monitoring scopes and can draw points on the faces of these scopes. He can program the gain of one of them.
10. He can move either of two film transports forward or backward an integral number of $\frac{1}{4}$ -sprocket holes.

Information is reported to the computer only when a boundary has been passed. In scanning, the first point across a black and white boundary will be reported. Therefore, if we have a white circular region on a black background, and a scanning line intersecting the region at the points A and B, only these two points will be reported. At the end of each scanning sweep, the setting is forced to that of the background. The x and y coordinates of the point are reported along with the intensity level at this point. Points in the rectangular area are investigated at the rate of 4 microseconds per point. Approximately 60 microseconds are required to initiate each scanning action. The entire 16,000,000-point raster can be scanned in 64 seconds, but in practice only small portions of this raster are actually scanned.

Interrupts will occur when:

1. A film transport has moved correctly.
2. A film transport has not moved correctly.
3. The scanner has scanned the rectangular area.
4. The input buffer is filled.

Along with the standard peripheral equipments of a magnetic tape unit, a typewriter, a paper tape reader, and a paper tape punch, the ASI-210 is connected by a 24-bit direct data channel to the CDC-3600. The channel is a standard CDC-3607 data channel, while the interface was built at ANL. Data transmission is buffered in both computers and transmission is at the rate of 350,000 characters per second. The programmer in either computer can:

1. Check the readiness of the other computer.
2. Change his own condition of readiness.
3. Read information from or write information to the other computer and be informed
 - a. of the quality of transmission (i.e., whether there was a parity error);
 - b. if the I/O instruction has been rejected.
4. Set status flags which can be interrogated by the other computer.

FIGURE 4
SCHEMATIC OF THE CHLOE SYSTEM

FIGURE 5
SCHEMATIC OF THE CRT SCANNER

FIGURE 6

SECTION III

THE CDC-3600 COMPUTER

The Control Data CDC-3600 is a large-scale general-purpose digital computer with an extremely good repertoire of arithmetic and logical instructions. It has a fast (1.4 microsecond cycle time), 65,536-word random access memory. Each storage word contains 48 data bits and there are six 15-bit index registers. Information on the CDC-3600 is contained in references 13a, 13b, 13c, 13d, 13e, and 13f. Figure 7 shows the CDC-3600 system at Argonne.

FIGURE 7

SECTION IV

AROMA

The AROMA portion of the AROMA-AIRWICK system is designed for the ASI-210. It controls the movement of film and the digitization of the photographs, identifies sparks and fiducials and transmits their CHLOE coordinates to magnetic tape or to the CDC-3600. AROMA can best be described as a quasi-monitor which can furnish subroutines to:

1. Move the film transports.
2. Control the scanner and photomultiplier settings.
3. Start the scanners and allow them to input data.
4. Communicate with the CDC-3600.
5. Communicate with the standard peripheral equipment.
6. Execute the AROMA cell-construction algorithm.

A block diagram of AROMA is given in Figure 8.

Halts: 0000 - Cannot Reach Scanners
 0001 - Unexpected Interrupt
 0002 - Divide Fault
 0003 - Add Overflow
 0004 - Index Overflow

 8191 - Program Finished

Figure 8

Detailed descriptions of AROMA subroutines, especially BUILD A and BUILD B which execute the cell-construction algorithm, will be given in this section. Since many of the other subroutines are of the type which are available on the CDC-3600 library tape, the later discussion of AIRWICK will not go into this in much detail.

AROMA Cell-Construction Algorithm

The subroutine executing this algorithm sequences through the input area from the scanner, sorting line segments into cells. The programmer may specify the maximum number of cells (which must be two or more) that can be constructed. Upon exiting from the subroutine, the cells will have been sorted by descending area and the A register will contain the actual number of active cells. Two types of cell growth are allowed: Type 1, which allows 8 words per cell, and Type 2, which allows 11 words per cell.

When an attempt is made to insert a line segment into a cell in the cell growth area, a search is made for a cell with a non-zero area. When one is found, the close-out flag is tested to see if any additional line segments can be accepted by the cell. If not, the search for a new non-empty cell continues. If the cell can accept a new line segment, the y-value of the new segment is compared with that of the last segment placed in the cell. If more than two sweeps have passed since this segment was inserted, the close-out flag is set and a return is made to searching for another non-empty cell. If the cell can still accept a segment, the segment is tested for maximum length, and if it is short enough, a test is made to see if it overlaps the last segment in the cell. If overlap exists, the segment is inserted and the cell data is updated. A check is then made to see if the maximum number of segments is in the cell. If this is the case, the close-out flag is set. If overlap does not exist, the search for another non-empty cell continues. If the line segment will not fit into any cell and room is left in the cell growth area, a new cell will be started. If the line segment will not fit into any cell and no room is left in the cell growth area, the cell with the smallest area is removed and a new cell is started in its place.

This algorithm will be employed in extracting information from the CERN film and from film to be taken in the Ke₂ and the 70-ton spark chamber experiments. It has been tested on data plotted from CHLOE digitizations of the CERN pictures. The subroutines for executing the algorithm have been written and are in the process of being debugged.

The contents of each word in a cell are as follows:

<u>Word</u>	<u>Meaning</u>	<u>Type 1</u>	<u>Type 2</u>
1	Area	X	X
2	Sum of x-values	X	X
3	Sum of y-values	X	X
4	Number of line segments in cell	X	X
5	Last y-value	X	X
6	Last left x-value	X	X
7	Last right x-value	X	X
8	Close-out flag	X	X
9	Sum of x ² -values		X
10	Sum of y ² -values		X
11	Sum of xy-values		X

Because images on film taken in spark chambers have relatively simple shapes, when the cell-construction algorithm has been applied to all line segments transmitted by the scanner in scanning a rectangular area of the film, we can readily have the following information on the Nth largest connected (within the limitations imposed by MAXX and MAXY) figures in the area:

1. The area.
2. The centroid ($\bar{x}, \bar{y}$).
3. The height (number of line segments in the cell).
4. The average width of the cell.
5. The coordinates of the last segment in the cell.

We can also obtain:

6. The variances σ_x^2 and σ_y^2 .
7. The covariance μ_{xy} .

There are two entry points into the subroutine, BUILDA and BUILDDB. The first has the calling sequence:

```
L1   RTN   BUILDA
      JMP   BUILDA+1
      PAR   MAXY
      PAR   MAXX
      PAR   ADD1
      PAR   ADD2
L2   .....
```

Upon entry, the contents of the A and E registers are interrogated. If (A) is less than zero, cell growth will be of Type 2. If (A) is greater than zero, cell growth will be of Type 1. (A) will be the number of cells which can be grown. If (E) is less than zero, a scanning mode of 1 will be assumed. If (E) is greater than zero, a scanning mode of 0 will be assumed. (E) will be the number of words in the input area. Memory size forces a limit of 1024 cells of Type 1 and 744 cells of Type 2. The programmer must be responsible for preventing overflow from the cell growth and input areas. MAXY contains the maximum number of line segments that can be placed in a cell. MAXX contains the maximum allowable line segment length. ADD1 is the address of the first word in the cell growth area. ADD2 is the address of the first word in the input area. If upon entry, (A) = 0 or (E) = 0, then (A) will be zero upon exit. Control information is established, the cell growth area is initialized to zero, the entry flag is set and control is transferred to BUILDDB.

The second entry point has the calling sequence:

```
L1   RTN   BUILDDB
      JMP   BUILDDB+1
L2   .....
```

Upon entry, the entry flag is cleared and control is transferred to BUILDDB. BUILDA is then to be used in sorting the first block of data from a scanned area and BUILDDB is to be used in sorting additional blocks. If the programmer uses the subroutine INPUT, he will know when to terminate his calls to BUILDDB. If there is a possibility of an add overflow in calculating sum-x , sum-y , sum-x^2 , sum-y^2 , sum-xy , or the area, it will be the programmer's responsibility to take precautionary measures.

If an area is being scanned which is noise-free, and if it contains only sparks or only fiducials, Type 1 growth should be sufficient to determine all necessary information concerning sparks and to establish fiducials. Approximately

$$[(4/3)N] + 2$$

cells should be allowed if N is the maximum number of sparks or fiducials that can appear in the area.

If the area contains both sparks and fiducials, Type 2 growth provides second-order moments which can be used to do some crude pattern recognition. If the fiducials are relatively large symmetric figures, they can easily be separated

from sparks. If the area contains noise, such as illuminated plate edges, the number of cells should be approximately

$$2(N + 1) ,$$

where N is the maximum number of sparks and/or fiducials that can appear in the area. With judicious selections for MAXY and MAXX and for the boundary of the area, the noise can be effectively eliminated. Scanning the area in the other mode can yield increased information about the cells. As an example, a criterion that seems to be effective in establishing the existence of a spark is to force a cell to pass the following tests:

1. $(1/2)\bar{A} \leq \text{Area} \leq (3/2)\bar{A}$, where $\bar{A}$ is the average area of the active cells.
2. $(1/2)\bar{h} \leq \text{Height} \leq (3/2)\bar{h}$, where $\bar{h}$ is the average height of the active cells.
3. $|\bar{W} - \delta| \leq .25\bar{W}$, where $\bar{W}$ is the average width of the cell and δ is the length of the last line segment.
4. $|\rho| > .10$, where ρ is the correlation coefficient.

In processing the CERN film, we assume the presence of six rectangular areas to be scanned. Each of these will correspond to a chamber; three chambers are above the center separation and three are below it. Since noise is present, we will allow

$$2(6M + 1)$$

cells per area, where M is the number of gaps present in the chamber. Fiducial areas will be scanned separately and after using the algorithm, additional checking is made to ensure the presence of "├" and "┐" shapes.

In processing film from the Ke_2 experiment, we establish the location of fiducials by scanning a predetermined area (using 10 cells) and checking the largest cell to see if it represents a fiducial. A narrow band is scanned in each chamber. If sparks are detected, a shorter, wider rectangular region centered about their location is scanned. In the first case 10 cells are used and in the second case 20 cells. As an example of some additional techniques in utilizing the cell information from these pictures, we give the following tests:

1. For first two fiducials. The two largest cells must have areas between 1800 and 4500 raster points (on the 4096 x 4096 raster). If their centroids are $(\bar{x}_1, \bar{y}_1)$ and $(\bar{x}_2, \bar{y}_2)$ we must have

$$|\bar{x}_1 - \bar{x}_2| \leq 50$$

$$1140 \leq |\bar{y}_1 - \bar{y}_2| \leq 1280$$

2. For additional fiducials. The largest cell must have an area between 1800 and 4500 raster points.
3. For existence of sparks in first scan down the chamber. The area of the largest cell must be greater than 20 raster points, and if there is more than one cell, the distance between the centroids and the two largest must be less than $(200/975)D$, where D is distance between the first two fiducials.
4. For existence of sparks and fiducials in second scan. All cells meeting the following criteria

$$|\bar{x}_i - \bar{x}_j| \leq \left(\frac{20}{975}\right)D, \quad \left(\frac{150}{975}\right)D \leq |\bar{y}_i - \bar{y}_j| \leq \left(\frac{180}{975}\right)D$$

or

$$\left(\frac{180}{975}\right)D \leq |\bar{x}_i - \bar{x}_j| \leq \left(\frac{220}{975}\right)D, \quad |\bar{y}_i - \bar{y}_j| \leq \left(\frac{20}{975}\right)D$$

or

$$\left(\frac{180}{975}\right)D \leq |\bar{x}_i - \bar{x}_j| \leq \left(\frac{220}{975}\right)D, \quad \left(\frac{150}{975}\right)D \leq |\bar{y}_i - \bar{y}_j| \leq \left(\frac{180}{975}\right)D,$$

where i and j represent active cells, are accepted as fiducials. If there are at least four of these, the other four largest remaining cells are accepted as sparks. Three of these must have non-zero area.

Flowcharts of Cell-Construction Process

The cell growth area will be referred to as CELLS(I,J), where $I_{\max}$ is the number of cells and $J_{\max}$ is 8 or 11. The input area will be referred to as INPT(K), where $K_{\max}$ is the number of words in the input area. $K_{\max}$ must be a multiple of 4.

SORT

Sort Cells by Decreasing Area
 Place Number of Active Cells in A
 Is Entry Flag Set?
 Jump to BUILDB

yes → BUILDA

UPDATE

Zero Cell of Smallest Area
 Place New Data in This Cell
 Jump to BUILDC+1

BUILD C

BUILD C+1

no

K = -3
 K = K+4
 Is K greater than $K_{\max}$?
 Is Mode = 0?
 x-left = INPT(K)
 y-left = INPT(K+1)
 x-right = INPT(K+2)
 y-right = INPT(K+3)
 Jump
 x-left = INPT(K+1)
 y-left = INPT(K)
 x-right = INPT(K+3)
 y-right = INPT(K+2)
 Shift data in x-left, y-left, x-right, y-right
 Is x-left less than zero?
 Is y-left less than zero?
 Is x-right less than zero?
 Is y-right less than zero?
 Is x-left greater than x-right?
 Is y-left equal to y-right?
 I = 0
 I = I+1
 Is I greater than $I_{\max}$?
 Jump to BUILD E

yes → SORT

yes → SORT

yes → SORT

yes → SORT

yes → BUILD C+1

no → BUILD C+1

UPDATE

BUILDE

Is CELLS(I,1) greater than zero?
 Is CELLS(I,5)+3 greater than y-left?
 Is x-left greater than CELLS(I,7)?
 Is CELLS(I,6) greater than x-right?
 Jump to BUILDG

no

BUILDD

no

BUILDF

yes

BUILDD

yes

BUILDD

BUILDF

CELLS(I,8) = 1
 Jump to BUILDD

BUILDG

DELT = x-right - x-left + 1
 Is DELT greater than MAXX?
 Incorporate All Data Pertaining to This
 Line Segment into
 CELLS(I,1),.....,CELLS(I,7)
 If Type 2 growth is allowed, incorporate
 all data pertaining to this line segment
 into CELLS(I,9), CELLS(I,10) and
 CELLS(I,11)
 Does CELLS(I,4) exceed MAXY?
 Jump to BUILDG+1
 CELLS(I,8) = 1
 Jump to BUILDG+1

yes

BUILDG+1

yes

Listing of Available Subroutines

The following subroutines are available to the programmer in the AROMA portion of the AROMA-AIRWICK system:

BCDRD	Magnetic Tape Routine
BCDWRT	Magnetic Tape Routine
BCKGND	Photomultiplier Control Routine
BEGIN	CDC-3600 Routine
BINRD	Magnetic Tape Routine
BINWRT	Magnetic Tape Routine
BUILD A	Cell-Construction Routine
BUILDB	Cell-Construction Routine
CONVT	Data-Handling Routine
INPUT	Scanner Input Routine
LAST	CDC-3600 Routine
LEVEL	Photomultiplier Control Routine
MODE	Scanner Control Routine
MVFT	Film Transport Routine
NREADY	CDC-3600 Routine
OCTER	Data-Handling Routine
POWX	Scanner Control Routine
POWY	Scanner Control Routine
PTIN	Paper Tape Reader Routine
QUERY	CDC-3600 Routine
READ	CDC-3600 Routine
READY	CDC-3600 Routine
REWIND	Magnetic Tape Routine
SCANNER	Scanner Control Routine
SET	CDC-3600 Routine
SPACE	Magnetic Tape Routine
START	Scanner Input Routine
TEST	CDC-3600 Routine
TESTR	CDC-3600 Routine
TIN	Typewriter Routine
TOUT	Typewriter Routine
WEOF	Magnetic Tape Routine
WRITE	CDC-3600 Routine
XSET	Scanner Control Routine
YSET	Scanner Control Routine

Film Transport Movement

The calling sequence for this subroutine is

```

L1  RTN  MVFT
    JMP  MVFT+1
L2  .....

```

Upon entering MVFT, the contents of the A and E registers are interrogated. If (E) = 0, film transport 2 is to be moved. If (E) ≠ 0, film transport is to be moved. If (A) is less than zero, the film transport is to be moved backward. If (A) is greater than zero, the film transport is to be moved forward. If (A) = 0, control is returned to the main program at location L2 with (A) = 0. |(A)| gives the number of quarter sprocket holes that the film is to be moved. When the film transport has completed movement, control is returned to L2 with (A) = 0. If the film transport fails to move properly, a message "FT1FT" or "FT2FT" is typed out and control is returned to L2 with (A) ≠ 0.

Scanner and Photomultiplier Settings

The subroutines XSET, YSET, POWX, POWY, MODE, LEVEL, SCANNER, and BCKGND fall into this category. The calling sequence for each subroutine is of the form:

```
L1  RTN  XXXXX
      JMP  XXXXX+1
L2  .....
```

Upon completion of the function, control is returned to the main program at location L2.

- (1) Upon entering XSET, the contents of the A and E registers are interrogated. (A) contains the x-low setting and (E) contains the x-high setting. The x-high register will be set to

$\max(1, \min((E), 4095))$.

The x-low register will be set to

$\max(0, \min((A), (x\text{-high} - 1)))$.

- (2) Upon entering YSET, the contents of the A and E registers are interrogated. (A) contains the y-low setting and (E) contains the y-high setting. The y-high register will be set to

$\max(1, \min((E), 4095))$.

The y-low register will be set to

$\max(0, \min((A), (y\text{-high} - 1)))$.

- (3) Upon entering POWX, the contents of the A register is interrogated. (A) contains the power-of-x setting. The power-of-x will be set to

$\max(5, \min(12, (A)))$,

provided $(x\text{-high}) - (x\text{-low})$ is divisible by 2^k , where $k = 12 - (\text{power-of-x})$; otherwise the power-of-x will be set to 12.

- (4) Upon entering POWY, the contents of the A register is interrogated. (A) contains the power-of-y setting. The power-of-y will be set to

$\max(5, \min(12, (A)))$,

provided $(y\text{-high}) - (y\text{-low})$ is divisible by 2^k , where $k = 12 - (\text{power-of-y})$; otherwise the power-of-y will be set to 12.

- (5) Upon entering MODE, the contents of the A register is interrogated. The scanning mode is set to 1 if $(A) \neq 0$. The scanning mode is set to 0 if $(A) = 0$.

- (6) Upon entering LEVEL, the contents of the A register is interrogated. (A) contains the photomultiplier level setting. The photomultiplier response level will be set to

$\max(1, \min(8, (A)))$.

- (7) Upon entering SCANNER, the contents of the A register is interrogated. If (A) $\neq$ 0, scanner 2 is to be used. If (A) = 0, scanner 1 is to be used.
- (8) Upon entering BCKGND, the contents of the A register is interrogated. If (A) $\neq$ 0, white-on-black film is to be scanned. If (A) = 0, black-on-white film is to be scanned.

Scanner Control

The subroutines START and INPUT fall into this category. The calling sequence for each subroutine is of the form:

```
L1   RTN   XXXXX
      JMP   XXXXX+1
L2   .....
```

Upon completion of the function, control is returned to the main program at location L2.

- (1) The subroutine START executes the ED control words established by the earlier subroutine calls to XSET, YSET, POWX, POWY, MODE, LEVEL, SCANNER, and/or BCKGND and starts the scanner. Up to 5000 attempts are made to accomplish the desired function. If this cannot be done, the message "NOSCN" is typed out and a halt at location 0 occurs.
- (2) Upon entry into INPUT the contents of the A and E registers are interrogated. A contains the ASR control word giving the first word of the input area. E contains the ASR control word giving the last word of the input area. Up to 5000 attempts are made to input data from the scanner. If this cannot be done, the message "NOIPT" is typed out and control is returned to L2 with (A) = 0. If incorrect control information has been given to the subroutine (incorrect ASR control words or a word count which is not a multiple of 4), the message "BAD1" is typed out and control is returned to L2 with (A) = 1. The contents of each word in the input area is initially set to -1 (7777776₈) and the scanner is allowed to input data. When the input area is filled, or when the predetermined area has been scanned, control is returned to L2. (A) will be set to 2 if the area has not been completely scanned and will be set to 3 if it has. If the programmer wishes to buffer the scanner input, he must write his own scanning subroutines.

Communication with the CDC-3600

The subroutines READY, NREADY, READ, WRITE, SET, QUERY, TEST, BEGIN, LAST, and TESTR which perform these functions are described in reference 11.

Communication with the Standard Peripheral Equipment

(1) Typewriter Subroutines

The calling sequence for each of the subroutines TOUT and TIN is

```
L1   RTN   XXXXX
      JMP   XXXXX+1
L2   .....
```

When the function has been completed, control is returned to the main program at location L2.

- a. TOUT is used to type out a typewriter message. Upon entering TOUT the contents of the A and E registers are interrogated. A contains the ASR control word giving the first word of the output area. E contains the ASR control word giving the last word of the output area.
- b. TIN is used to input one character from the typewriter. The character will appear in bits 1-6 of the A register upon return to L2.

(2) Data Rearranging Subroutines

- a. The calling sequence for the subroutine OCTER is

```
L1   RTN   OCTER
      JMP   OCTER+1
L2   .....
```

OCTER is used to rearrange octal numbers. Upon entering OCTER the contents of the A register is interrogated. (A), which will be PQRSTU₈, is rearranged and occupies (AE) in the form OPOQOROSOTOUOV₈ upon return to L2.

- b. The calling sequence for the subroutine CONVT is

```
L1   RTN   CONVT
      JMP   CONVT+1
      PAR   ADD1
L2   .....
```

CONVT will interpret bits 1-6 of the six words ADD1, ADD1+1, ..., ADD1+5 as BCD representations of decimal integers and convert this 6-digit number into a binary number which will be in the A register upon return to L2. ADD1 will contain the most significant digit. Any difficulties in the conversion will result in setting (A) = 0 upon return.

(3) Paper Tape Subroutines

There is only one subroutine in this category. Its calling sequence is:

```
L1   RTN   PTIN
      JMP   PTIN+1
L2   .....
```

Upon entering PTIN, the contents of the A and E registers are interrogated. A contains the ASR control word giving the first word of the input area. E contains the ASR control word giving the last word of the input area. The characters will appear in bits 1-6 of the words in the input area when control is returned to L2. If incorrect control information is furnished or if a parity error is detected, a message "PTERR" is typed out and (A) $\neq$ 0 when control is returned to L2. If the function has been successfully completed, (A) = 0 at L2.

(4) Magnetic Tape Subroutines

The subroutines in this category are BINRD, BINWRT, BCDRD, BCDWRT, REWIND, SPACE, and WEOF. Their calling sequences are of the form:

```
L1   RTN   XXXX
      JMP   XXXX+1
L2   .....
```

Control will be returned to the main program at location L2 with one of the following settings in the A register:

- 0 - Function successfully completed
- 1 - Incorrect control information
- 2 - 5 successive parity errors in reading
- 3 - 5 successive parity errors in writing
- 4 - Encountered end-of-tape
- 5 - Encountered end-of-file while reading
- 6 - Other tape failure

If the contents of A is to be 1, 2, 3, 4, 5, or 6, the message "TPERR" will be typed out prior to returning to L2.

- a. In BINRD, BINWRT, BCDRD, and BCDWRT the contents of the A and E registers are interrogated upon entry. A will contain the ASR control word giving the first word of the input (or output) area. E will contain the ASR control word giving the last word of the input (or output) area. The functions of the subroutines are:

```
BINRD - Read Binary Tape
BINWRT - Write Binary Tape
BCDRD - Read BCE Tape
BCDWRT - Write BCD Tape
```

- b. REWIND rewinds the magnetic tape.
- c. Upon entry into SPACE the contents of the A register is interrogated. If (A) is less than zero the tape will be moved backward. If (A) is greater than zero, the tape will be moved forward. (A) is the number of records over which the tape is to be moved.
- d. WEOF writes an end-of-file mark on the magnetic tape.

SECTION V

AIRWICK

While AROMA was written with spark chamber photograph digitization in mind, it is general enough to be useful on many classes of photographs. AIRWICK has a more specialized objective, that of furnishing a uniform input into high energy physics geometry and kinematics programs. Much thought has been given to making it useful for a large variety of spark chamber data processing problems. Figure 9 gives a block diagram of the AIRWICK concept.

AIRWICK may be divided into three parts, PREPROS, PAIR, and LINK. Data will be processed in order through these segments and each segment will have an increasing amount of generality in the system.

1. PREPROS. This section of the system is the least general and will probably have to be rewritten as the problem and the input formats change. It is responsible for furnishing input to PAIR or LINK.
2. PAIR. With all the problems inherent in the automatic processing of bubble chamber photographs, there is one distinct advantage that they possess over photographs taken of spark chambers. This is the continuity of the track or at least large segments of the track. By this we mean that if a point lies on a track we can almost always assume that its nearest neighbor also belongs to the track. The problem of linking points into tracks is therefore a minor one when compared to the problem of filtering (removing beam tracks) or to the problem of combining tracks into curves in space.

Since individual bubbles cannot be readily identified, it is necessary to complete the linking process first. As far as we know, all efforts to process spark chamber photographs have also adopted this approach. However, as the complexity of the photograph increases, i.e., as the number of tracks increases, as curvatures vary, when sheeting becomes more common, when gaps have different widths and orientations, this problem becomes much more difficult. It was realized that if sparks could be reconstructed in space with no a priori knowledge of the picture's topology, the situation would be much better. Tracks which intersect would then almost always intersect because of a vertex. The physical separation of the sparks would always be at least as great as in the picture and in most cases would be significantly greater, an effect which tends to normalize distances between sparks.

The most important feature of the AIRWICK concept has been that of reconstruction prior to linking and the techniques we have used in the reconstruction process have been surprisingly successful. The purpose of PAIR is then to resolve the ambiguities concerning spark location, to reconstruct the spark in real space and to furnish LINK with a list of real-space coordinate triples. It is applicable to information received from two stereo views from photographs or vidicon systems and is therefore the next most general part of AIRWICK.

3. LINK. The problem of linking, which is now carried out in three dimensions, is common to all spark chamber data processing. LINK accepts a random group of real-space coordinates, links them together into tracks and provides a "tree"-structured input into the geometry and kinematics programs. This is the most general part of the system and is also applicable to data taken from sonic or wire chambers. In approaching the problem of linking, we have made as few assumptions as possible concerning track behavior.

AROMA is itself a system, designed for the CHLOE Film Scanning Equipment. For other data sources, scanning will be necessary if the expense of completely processing an "uninteresting" event is prohibitive.

THE AIRWICK CONCEPT

Figure 9

SECTION VI

PREPROS

AIRWICK has been designed to be run under the operation of the CDC-3600's SCOPE operating system. The ability to converse with the ASI-210 has been imbedded in the monitor and the programmer need not concern himself with unusual input/output procedures.

The PREPROS portion of AIRWICK has been written for the pictures taken by the Argonne Group at CERN. Here the functions of PREPROS are to:

1. Receive data in a specified format from the ASI-210.
2. Rearrange this information and place it in a form more acceptable to the CDC-3600.
3. Remove the optical distortions from the data and transform the CHLOE coordinates of spark and fiducial centroids into real-space coordinates on the top surface of the chamber.*
4. Sort sparks into gaps and chambers.

*These are the coordinates of the intersections with the top surface of the chamber, of straight lines joining the sparks to the camera lenses.

SECTION VII

PAIR

PAIR Input

In writing PAIR, we have made several assumptions regarding the coordinate axes and the shape of the chambers. It is assumed that the chambers are rectangular parallelepipeds whose outer boundaries can be effectively delineated by x-low, x-high, y-low, y-high, z-low, and z-high values, which are real-space coordinates, i.e., the real-space coordinate axes are parallel or perpendicular to the outer surfaces of the chamber. If this cannot be assumed, PREPOS must make the necessary transformations and LINKFIX, which is at present a dummy subroutine, must be furnished by the programmer to follow LINK and re-establish the real space coordinate system for the geometry program. Sparks in up to ten different chambers may be handled by PAIR and LINK. Gap numbers can range from 1 to 4095, chamber numbers can range from 1 to 10.

It is assumed that each camera is photographing only one surface of the chamber; however, this surface may be the same for both cameras (as is the case in the CERN photographs). It is also assumed that in each chamber, gaps will be numbered consecutively from 1 to N and gap 1 will lie on the entry end of the chamber. A maximum of 300 sparks in one view can be handled by PAIR. PAIR must be furnished with the following:

1. A subroutine THREED which can reconstruct a coordinate triple (x,y,z) from the two coordinate pairs. This subroutine will be a function of camera locations and will change as the experiment changes. When a call is made to THREED, the following information is furnished by PAIR:
 - a. XVIEW1 contains the "x"-coordinate from PRIN(I,1,1)
 - b. XVIEW2 contains the "x"-coordinate from PRIN(I,1,2)
 - c. YVIEW1 contains the "y"-coordinate from PRIN(I,2,1)
 - d. YVIEW2 contains the "y"-coordinate from PRIN(I,2,2)
 - e. NCHAMB contains the chamber number
 - f. NGAP contains the gap number
2. A table BOUNDARY(10,7) which contains the boundaries for the chambers. For the Ith chamber:
 - a. BOUNDARY(I,1) contains the low "x"-value
 - b. BOUNDARY(I,2) contains the high "x"-value
 - c. BOUNDARY(I,3) contains the low "y"-value
 - d. BOUNDARY(I,4) contains the high "y"-value
 - e. BOUNDARY(I,5) contains the low "z"-value
 - f. BOUNDARY(I,6) contains the high "z"-value
 - g. BOUNDARY(I,7) = 1 if the entry surface is parallel to the xy-plane
 = 2 if the entry surface is parallel to the xz-plane
 = 3 if the entry surface is parallel to the yz-plane

All quantities except BOUNDARY(I,7) are floating point numbers.

3. NPLANE, which is 1 or 2 depending upon the number of surfaces being photographed.
4. AWIDTH, AMETRIC, APROB, and AFUZ, which are weighting values to be used in evaluating the PAIR decision function. They must be non-negative and sum to 100.
5. A table PRIN(300,4,2) which contains the following spark information for the Ith spark in the Jth view:
 - a. PRIN(I,1,K) contains the "x"-coordinate which has been assigned to the centroid of this spark.
 - b. PRIN(I,2,K) contains the "y"-coordinate which has been assigned to the centroid of this spark.
 - c. PRIN(I,3,K) contains the width of the spark.
 - d. PRIN(I,4,K) contains the gap number in bits 23-0 and the chamber number in bits 47-24.

All quantities except PRIN(I,4,K) are floating point quantities.

Unused space in PRIN(300,4,2) and BOUNDARY(10,7) should be cleared prior to entry into PAIR.

PAIR Processing

For each view, sparks are sorted by chamber, and within each chamber are sorted by gap. Within each gap, sparks are sorted by increasing $x^2 + y^2$. Only the first 25 sparks in each gap are accepted. Since one of the coordinates, x or y, will be relatively constant, the sparks will in fact be ordered. We then sequence through each chamber and gap. In the remainder of the discussion, we concern ourselves only with sparks which appear in the same gap.

If, in either view, we find the gap devoid of sparks, we return to the sequencing operation. If there is at least one spark in the gap in each view, we can visualize the situation in an $m \times n$ matrix, where m and n are both between 1 and 25. The m sparks in view 1 represent the rows and the n sparks in view 2 represent the columns. Since the sparks in both views have been ordered, the row and column indices will reflect the positions of the sparks in that order.

An $m \times n$ incidence matrix is constructed in the following manner:

- a. If NPLANE = 2, each element is set to 1.
- b. If NPLANE = 1, spark i in view 1 and spark j in view 2 represent a possible pairing $P(i,j)$. The coordinates of these sparks are transmitted to THREED for spatial reconstruction. If the reconstructed spark lies within the chamber, the ij th element of the matrix $M(i,j)$ is set to 1. If the spark is not in the chamber, $M(i,j)$ is set to 0. It is clear that as the stereo angle decreases, the number of 1's in the incidence matrix will decrease because of fewer ambiguities in pairing. We will say that sparks have been paired if their reconstruction lies within the chamber and that the pairing has been accepted if this reconstruction is entered into the LINK input table LKIN(300,12).

2. Step B. Suppose we discover that there is less than a 10 percent difference in an $r \times s$ block. We calculate $A(i,j)$, where

$$A(i,j) = \sum_k D(i,k) + \sum_e D(e,j) - 2D(i,j)$$

if $D(i,j)$ is non-zero;

$$A(i,j) = 10^6$$

if $D(i,j)$ is zero. In the sum, k is summed over the r remaining rows and e is summed over the s remaining columns. $A(i,j)$ will be a measure of the degree to which all other sparks in view 1 could pair with spark j and all other sparks in view 2 could pair with spark i . We will say that a value of $A(i,j)$ is small if it is less than 10^6 . If the minimum small $A(i,j)$ in the block is 10 percent less than the next smallest value, $P(i,j)$ is accepted as a pairing and row i and column j are removed from the block. If there is a row or column containing two or more small values of $A(i,j)$, the process is continued. If for any small $A(i,j)$, all other values in the i th row and j th column are not small, $P(i,j)$ is accepted as a pairing and the row and column is removed. There can be at most $\min(r,s)$ of these steps.

It is still possible that in some $u \times v$ block this criterion fails. We must go on, but before we do, we give the following summary of the sizes of blocks with which we have dealt:

where

$$1 \leq u \leq r \leq p \leq m \leq 25$$

$$1 \leq v \leq s \leq q \leq n \leq 25$$

$$\max(u,v) > 1$$

3. Step C. We return to what is left of our original $p \times q$ matrix of $D(i,j)$ values and select the largest one as a pairing $P(i,j)$, delete that row and column and return to Step A. If there are several maximal values of $D(i,j)$, we recalculate the $A(i,j)$, select the pairing $P(i,j)$ with the smallest value of $A(i,j)$, delete that row and column and return to Step A.

If there are several minimal values of $A(i,j)$, we select the pairing $P(i,j)$ which minimizes $i^2 + j^2$, delete that row and column and return to Step A. If there is more than one minimal value for $i^2 + j^2$, we select the pairing $P(i,j)$ with the smallest i , delete that row and column and return to Step A.

PAIR Decision Function

The values A, B, C, and D in the PAIR decision function correspond to weights AWIDTH, AMETRIC, APROB, and AFUZ, respectively, which have been input into PAIR. These weights are non-negative and sum to 100. The values of the functions f_1 , f_2 , f_3 , and f_4 are all in the range from 0 to 1. The value of the function is 0 if the incidence matrix entry $M(i,j)$ is 0. We therefore need consider only those pairings whose reconstruction lies within the chamber.

1. The width function $f_1(i,j)$

This function is evaluated as

$$f_1(i,j) = 1 - |w'(i) - w'(j)| ,$$

where $w'(i)$ is the normalized width of the i th spark in view 1 and $w'(j)$ is the normalized width of the j th spark in view 2. The normalized width w' is calculated by

$$w'(k) = (w(k)) / \left(\sum_e w(e) \right) ,$$

where $w(k)$ is the width of the spark and e ranges over all sparks in the block which are in the same view. $f_1(i,j)$ will therefore be 1 if the normalized widths are the same and will approach 0 as the difference in the normalized widths increases.

2. The metric function $f_2(i,j)$

This function is evaluated as

$$f_2(i,j) = 1 - |(i-j)| / (\max(p,q) - 1) ,$$

where i and j are the row and column indices of the pairing and the block size is $p \times q$. $f_2(i,j)$ will therefore be 1 if the pairing lies on the diagonal, i.e., if sparks have the same order in both views, and will approach 0 as the pairing moves from the diagonal.

3. The position function $f_3(i,j)$

The boundary of this chamber can be extracted from BOUNDARY(10,7). Let M, N, P, Q, R, and S represent the x-low, x-high, y-low, y-high, z-low, and z-high boundaries of the chamber. The gaps in the chamber will be numbered from 1 to T, with gap number 1 on the entry side of the chamber. If, for example, the entry surface is parallel to the xz-plane, we start with the line through the chamber which has length Q-P and passes through

$$(E,F) = ((M+N)/2, (R+S)/2) .$$

We calculate the quantities

$$L = \min((N-M)/2, (S-R)/2) ,$$

$$K = WL/T ,$$

where W is the gap number. K will be the radius of a conical slice for this gap. The cone has radius L/T in the first gap and L in the last gap. We then calculate the distance D of the reconstructed spark from the center line (E,F) and evaluate $f_3(i,j)$ as

$$f_3(i,j) = \max(0, t/5),$$

where t is an integer $5, 4, \dots$ such that

$$(5-t)K \leq D < (6-t)K.$$

$f_3(i,j)$ gives an approximate measure of the probability of the position of the reconstruction. If the pairing lies inside this cone, the value of the function is 1.

4. The "fuzziness" function $f_4(i,j)$

This function is a measure of the isolation of the sparks in the pairing. It approaches 1 as there are fewer pairings in the block which could interfere with this pairing and approaches 0 as the number of interfering pairings increases. It is evaluated by

$$f_4(i,j) = (1 - (S(i)-1)/P) (1 - (T(j)-1)/P) ,$$

where $P = \max(p,q)$, $S(i)$ is the number of pairings in the i th row of the block and $T(j)$ is the number of pairings in the j th column of the block.

Values of the Weights in the Decision Function

If $NPLANE = 2$, we have a case where all pairings exist and order will serve no purpose in our decision function. In this case, $AMETRIC$ should be set to 0. If the widths of all sparks are the same, $AWIDTH$ should be set to 0. $AFUZ$ and $APROB$ will always yield some information on the pairing. On the CERN film, it has been useful to use all of these values, and the best weights that we have been able to assign to them are:

1. $AWIDTH = 60$
2. $AMETRIC = 10$
3. $APROB = 10$
4. $AFUZ = 20$

If there have been no spark mergings in a view, the values of the decision function have, in every case, chosen the pairings that a human scanner would have made.

Example

As a rather simple example, we take the following case.

```

View 1:   Spark 1:   x = 1365
              y = 21
              w = 21
           Spark 2:   x = 1670
              y = 21
              w = 90
           Spark 3:   x = 1887
              y = 21
              w = 12

View 2:   Spark 1:   x = 1318
              y = 21
              w = 20
           Spark 2:   x = 1709
              y = 21
              w = 80
           Spark 3:   x = 1857
              y = 21
              x = 15
  
```

From the incidence matrix, we obtain

```

1 0 0
0 1 1
0 1 1
  
```

The pairing $P(1,1)$ is accepted and we need only consider the second block.

AWIDTH = 100, AMETRIC = 0, APROB = 0, AFUZ = 0

```

96.0  27.6
27.6  96.0
  
```

Pairings accepted are $P(2,2)$ and $P(3,3)$.

AWIDTH = 0, AMETRIC = 100, APROB = 0, AFUZ = 0

```

100.0  0.0
0.0  100.0
  
```

Pairings accepted are $P(2,2)$ and $P(3,3)$.

AWIDTH = 0, AMETRIC = 0, APROB = 100, AFUZ = 0

```

100.0  100.0
100.0  100.0
  
```

Pairings accepted are $P(2,2)$ and $P(3,3)$.

AWIDTH = 0, AMETRIC = 0, APROB = 0, AFUZ = 100

25.0 25.0
25.0 25.0

Pairings accepted are P(2,2) and P(3,3).

AWIDTH = 25, AMETRIC = 25, APROB = 25, AFUZ = 25

80.2 38.1
38.1 80.2

Pairings accepted are P(2,2) and P(3,3).

AWIDTH = 40, AMETRIC = 20, APROB = 30, AFUZ = 10

90.9 41.5
41.5 90.9

Pairings accepted are P(2,2) and P(3,3).

Another case, which has the following incidence matrix

```

1 1 0 0 0 0
1 1 0 0 0 0
1 1 1 1 1 0
1 1 1 1 1 0
1 1 1 1 1 0
1 1 1 1 1 0
1 1 1 1 1 0
0 0 0 0 0 1

```

has the following values of the decision function for AWIDTH = 50, AMETRIC = 50, APROB = 0, AFUZ = 0, for the first block

95.60	93.38	00.00	00.00	00.00	188.98
88.42	99.35	00.00	00.00	00.00	187.77
75.60	83.38	98.84	88.94	74.49	421.25
65.61	73.38	88.84	98.94	84.49	411.26
59.33	67.11	75.11	87.33	98.23	387.11
45.61	53.38	68.85	78.94	84.49	331.26
430.17	469.98	331.63	354.15	341.70	

Since the largest value of $D(i,j)$, 99.35, is less than 10 percent greater than the next largest value, 98.94, we obtain the following matrix of $A(i,j)$ values

427.95	472.20	10 ⁶	10 ⁶	10 ⁶
441.10	459.05	10 ⁶	10 ⁶	10 ⁶
700.22	724.47	.	.	.
710.21	734.48	.	.	.
698.62	722.87	.	.	.
670.21	694.48	.	.	.

The minimum small value, 427.95, is not 10 percent less than the next value, 441.10. We therefore accept the pairing P(2,2) and process the following matrix of $D(i,j)$ values:

95.60	00.00	00.00	00.00	95.60
75.60	98.84	88.94	74.49	337.87
65.61	88.84	98.94	84.49	377.88
59.33	75.11	87.33	98.23	320.00
45.61	68.84	78.94	84.49	277.88
341.75	331.63	354.15	341.70	

Again we must look at the $A(i,j)$ values, where we find

246.15	10^6	10^6	10^6
528.84			
.....			
.....			
.....			

Because 246.15 is more than 10 percent less than the next small value, the pairing $P(1,1)$ is accepted and we process the following matrix of $D(i,j)$ values:

98.84	88.94	74.49	262.27
88.84	98.94	84.49	272.27
75.11	87.33	98.23	260.67
68.84	78.94	84.49	232.27
331.63	354.15	341.70	

Here, we must look at the $A(i,j)$ values

396.22	438.54	454.99
426.22	428.54	444.99
442.08	440.16	405.91
426.22	428.54	404.99

The minimum value fails to satisfy our 10 percent criterion; we therefore accept the pairing $P(4,4)$ and process the following matrix of $D(i,j)$ values:

98.84	74.49
75.11	98.23
68.84	84.49

Our 10 percent criterion on the $D(i,j)$ values is not satisfied and we process the following matrix of $A(i,j)$ values:

218.44	281.56
265.91	231.09
258.44	241.56

The criterion is not satisfied and after accepting the pairing $P(3,3)$ we consider the matrix of $D(i,j)$ values

98.23
84.49

Here $D(5,5)$ is more than 10 percent greater than $D(5,6)$. We accept the pairing $P(5,5)$ and exhaust the block.

Many-to-One Pairings

Because sparks in one view may merge or be lost for various reasons, we can not, with positive assurance, accept only one-to-one pairings. The procedure outlined earlier for doing this has been very successful. When this is done, LINK must be relied upon to discard bad pairings. A modification of the pairing procedure which allows many-to-one pairings has been written and is outlined below:

In Step C, we select the maximal non-zero value(s) of $D(i,j)$ and all values within 10 percent of the maximal value. If this set has only one element, we accept the pairing, delete the row and column and return to Step A.

If the set has more than one element and all elements lie in the same row or in the same column, we accept the corresponding $P(i,j)$ as a many-to-one pairing, delete all affected rows and columns and return to Step A. For example, suppose our set is

$$\{D(i_1, j_0), \dots, D(i_n, j_0)\} .$$

We accept the pairings

$$P(i_1, j_0), \dots, P(i_n, j_0)$$

and delete column j_0 and rows $i_1, \dots, i_n$.

If the set can be divided into two subsets, each having at least two elements, one subset lying in row i and one subset lying in column j , and if $D(i,j)$ is not in the original set, we accept two many-to-one pairings, delete all affected rows and columns and return to Step A. For example, suppose our set is

$$\{D(i_1, j_0), \dots, D(i_n, j_0), D(i_0, j_1), \dots, D(i_0, j_m)\}$$

and

$$i_k \neq i_0 \text{ for } k = 1, \dots, n ,$$

$$j_r \neq j_0 \text{ for } r = 1, \dots, m ;$$

we accept the corresponding pairings and delete columns $j_0, j_1, \dots, j_m$ and rows $i_0, i_1, \dots, i_n$.

If the set of maximal values does not fall into any of these categories, but has a single maximal value, we proceed as before, accepting the corresponding pairing, deleting the row and column and returning to Step A. If there is more than one maximal value, we accept the pairing which minimizes $i^2 + j^2$. If there is more than one of these pairings, we accept the one which minimizes i . After a pairing $P(i,j)$ has been accepted, we delete the row and column and return to Step A.

It can be seen that this procedure did not alter any of the choices in the example.

Generality of PAIR

Since the pairing choices in a block are interdependent, if we are to be completely general, we must make a first choice of all sparks, a next best choice, and so on, transmitting all choices to LINK, which must choose between them. The problem of making these choices can be formulated as the k th best policy problem, whose solution is non-trivial. The versions of LINK and PAIR presented here take the more simplified view, a view which still has much merit. Work is being done on formulations of LINK and PAIR which deal with this k th best policy problem.

SECTION VIII

LINK

While the version of PAIR described earlier has been written and tested and can be said to furnish excellent results, our claims concerning LINK cannot be quite so broad. Much of it has been written, some of it has been tested, but parts are lacking in the system design.

Our approaches to the problem of linking have been:

1. global
 - a. the interpretation of moment invariants;
 - b. the construction of continuous functions whose "ridges" can be traced to delineate the tracks.
2. local
 - a. the use of various schemes to link sparks to their neighbors and to build these segment links into tracks.

Global techniques, which would be relatively easy to implement, are very difficult to formulate. Local techniques involve much combinatorial gymnastics and it is difficult to insure convergence to an adequate solution of the linking problem.

Theoretically, moment invariants are sufficient to completely separate the different classes of tracks. They are easy to calculate, but the difficulty comes in placing a geometric interpretation on them. J. W. Butler of ANL has suggested an approach which may be more applicable to linking if it is done before the pairing process. Here, continuous surfaces are defined on the plane of the photographs. Their "ridges" may be traced by methods of "least" descent. To visualize this, suppose rods are erected on the sparks and a plastic sheet is draped over them. The ridges remaining will tend to follow the tracks. This approach may also be applicable to linking after pairing, in which case "tubes" are formed along the tracks. No definite progress can be reported on either of these techniques.

Therefore we have formulated LINK to realize local techniques, assembling sparks into track segments and track segments into tracks. It should be remembered that we are concerned not only with single straight or relatively straight tracks, but with multiple tracks whose curvatures vary over wide ranges.

LINK Input

LINK input is furnished in the table LKIN(300,12), where for the Ith spark

- LKIN(I,1) contains the x-coordinate of the reconstructed spark.
- LKIN(I,2) contains the y-coordinate of the reconstructed spark.
- LKIN(I,3) contains the z-coordinate of the reconstructed spark.
- LKIN(I,4) contains the gap and chamber numbers for the spark. It has the same structure as PRIN(I,4,K).

All quantities except LKIN(I,4) are floating point quantities. Unused space in LKIN(300,12) should be cleared. In addition, LINK is furnished with the table CHAMBER(10), which contains the average gap width of the gaps in the chambers, and NTREE, which contains the maximum number of trees.

Organization of LINK Output

A tree is normally defined to be a graph which contains no circuits and which has at most one branch entering each node. This concept can be extended to a structure in which each node is represented by an array (in our case a 1 x 12 array); however, our structure is only ideally a tree since it is possible for circuits to exist and for more than one branch to enter a node (or spark).

This structure gives a reasonable description of the actual physical event taking place in the linking region and is amenable to certain generalized data processing techniques.

LINK Output

The main output from LINK is given in LKIN(300,12). If I represents the Ith spark,

LKIN(I,1) same as before
 LKIN(I,2) same as before
 LKIN(I,3) same as before
 LKIN(I,4) same as before
 LKIN(I,5) contains the vertex order
 LKIN(I,6) contains the index (in LKIN(300,12)) of the 1st connecting spark
 LKIN(I,7) contains the index (in LKIN(300,12)) of the 2nd connecting spark

 LKIN(I,12) contains the index (in LKIN(300,12)) of the 7th connecting spark

LKIN(I,5) through LKIN(I,12) are fixed point quantities which must be between 0 and 300. The order of the vertex can be 0, 1, 2, 3, 4, 5, 6, 7, or 8. It designates the number of sparks that are connected to this spark. If the vertex order is M ($M \geq 1$), LKIN(I,6) through LKIN(I,6 + M-1) are used and LKIN(I,6 + M) through LKIN(I,12) are cleared. If the vertex order is 0, LKIN(I,6) through LKIN(I,12) are cleared.

A vertex order 0 indicates an isolated spark, 1 indicates an endpoint, 2 indicates a spark interior to a track or the vertex of a "V". A vertex order $N \geq 3$ indicates the vertex of an (N-1)-pronged "Y" or an N-pronged "V". If the vertex order is greater than 8, information will be lost.

In addition to LKIN(300,12), the following tables are furnished:

1. MAINPT(1) gives the index I of the first spark in the main tree.
2. SECDPT(30) gives the indices I of the first sparks in up to 30 secondary trees.

MAINPT will be set and as many cells of SECDPT(30) as needed are used. If there are more than 30 secondary trees, the spark data will appear in LKIN(300,12) but there will be no directory of these trees. If a tree closes on itself, an arbitrary first spark will be chosen. Every effort is made in LINK to link track segments, and in the ideal case there will be only one tree furnished to the geometry program for each group of connected tracks in the chamber.

This tree structure seems reasonably well adapted for input into later programs. Most vertices of order 2 will be sparks in tracks rather than vertices of a "V". The cases can be distinguished by looking at the angles between the two line segments joining the three sparks. The vertices which are of interest to the physicist will lie within some small sphere about the vertices in the tree but, in general, these vertices will not coincide.

Tree Searching Subroutine

The tree searching subroutines TREESRCH and TREESORT are used repeatedly in LINK. The index of a spark is placed into INTOTREE. We call TREESRCH and the unique indices of sparks in the same tree (including INTOTREE) are placed in NTAB(300). The number of these indices is stored in NINDEX. These subroutines are used to investigate the status of trees of linked sparks. They give us the number of sparks in the tree and their indices.

We enter INTOTREE into NTAB(1) and set NINDEX to 1. We then check the non-zero indices of LKIN(INTOTREE,6) through LKIN(INTOTREE,12), entering their values into NTAB(300) and incrementing NINDEX by 1 each time one is entered. If none are entered, we exit with NTAB(1) = INTOTREE and NINDEX = 1. If some are entered, we set NTOT equal to 1, and sequence through NTAB(NTOT+1) through NTAB(NINDEX). If M = NTAB(I), we check the non-zero values of LKIN(M,6) through LKIN(M,12), making an entry into NTAB(300) whenever we encounter an index which has not previously been inserted. The process terminates when a pass through NTAB(NTOT+1) to NTAB(NINDEX) fails to encounter new entries. Flow charts of this process are given below.

TREESRCH

TREESORT

LINK Processing

From the order that sparks were placed in LKIN(300,12) by PAIR, we see that ordering by chamber and by gap is still preserved. To insure this order if input is not furnished by PAIR, we sort into chamber and gap all sparks in LKIN(300,12). For each chamber, we take the average gap width W from CHAMBER(10) and proceed as follows:

1. Step A. Locate the first gap.
2. Step B. If there is more than 1 spark in the gap, we compute the distances between the sparks in the gap. If the distance between any two sparks given by indices I and J in LKIN(300,12) is $\leq W$, we establish a link between them in the following manner:

```

LKIN(I,5) = LKIN(I,5) + 1
LKIN(J,5) = LKIN(J,5) + 1
M = LKIN(I,5)
N = LKIN(J,5)
LKIN(I, 5+N) = J if N ≤ 7   (step not done otherwise)
LKIN(J, 5+M) = I if M ≤ 7   (step not done otherwise)
  
```


3. Step C. If the next gap exists and there are sparks in both gaps, we compute the distances between the sparks in the first gap and the sparks in the second gap and link in the same way that we did in Step B, except that if no links occur with a distance of W , tests are also made with a distance of $2W$. If the next gap exists and contains sparks but there were no sparks in the first gap, we increment the gap counter and return to Step B. If the next gap exists but contains no sparks, we increment the gap counter and return to Step B. If the next gap does not exist, we exit and repeat the process for all chambers until we have sequenced through all of them.

If $NTREE = 1$, and there are sparks of positive vertex order, calls to TREESRCH will identify the subtrees in which they are located. If there is one subtree and no other sparks, the first spark of vertex order 1 is entered in MAINPT and we exit from LINK. If there is one subtree and other sparks or more than one subtree (with or without other sparks), each subtree must have two sparks of vertex order 1 or processing is terminated. If this condition is met, the endpoints are joined or are joined to isolated sparks by minimum distance. If all vertex orders are 0, we link sparks together which are separated by minimum distance. Linking is done in the same manner as before.

Figure 10

This process links all sparks and the index of the first spark of vertex order 1 is placed in MAINPT and we exit from LINK. If all vertices are of order 2, an arbitrary spark index is placed in MAINPT. If there can be at most one track in the linking region, $NTREE$ should be set to 1. This will force linking of all sparks. If there is any possibility that two or more tracks can be present, $NTREE$ should be greater than 1, since no discrimination is made of a link's validity.

If $NTRACK$ is greater than 1, we at present simplify matters by sequencing through LKIN(300,12). If any vertex order exceeds 3, processing is abandoned. If there are no sparks of vertex order 1, processing is abandoned. If there are, all of these cases are flagged and we begin to sequence through them in the following manner, extrapolating ahead to look for the next linkable spark.

Figure 11

In each case, there will be two sparks involved. We look ahead on a line determined by the points B and C. If no sparks are found in a cylinder of radius $2M$ (where M is the maximum value from `CHAMBER(10)`), we terminate our search. The first spark we find in this cylinder is linked to spark C. When this has been done for all cases, we place an index to the longest track into `MAINPT` and use as many cells of `SECDPT(30)` as necessary to index other track segments.

Comments

It is at this point that this description of the linking process must stop. We would like to be able to remove the restrictions we have placed on the vertex order and we would like to have some way of insuring termination of this rather agonizing step-by-step procedure of linking. Current and future work on `LINK` is toward solving these problems. We have hopefully linked many of the sparks in the chambers and by using `TREESRCH` repeatedly, we can keep track of our progress.

SECTION IX

REFERENCES

On Spark Chambers

1. Argonne Symposium on Spark Chambers, The Review of Scientific Instruments, Vol. 32, No. 5, May 1961, pp. 481-531.
2. Proceedings of the 1962 CERN Conference on Instrumentation for High-Energy Physics, Nuclear Instruments and Methods, Vol. 20, January 1963, pp. 1-520.
3. M. I. Daion, G. A. Leskin, "Spark Detectors for Charged Particles," Soviet Physics Uspekhi, Vol. 6, No. 3, November 1963, pp. 428-458.

On Automatic Processing

4. Harry Rudloe, Martin Deutsch, Thomas Marill, "PIP: A Photo-Interpretive Program for the Analysis of Spark Chamber Data," Communications of the ACM, Vol. 6, No. 6, June 1963, pp. 332-335.
- 5a. Arthur H. Rosenfeld, William E. Humphrey, "Analysis of Bubble Chamber Data," Advances in Nuclear Science, 1963, Pergamon Press, 1963, pp. 103-144.
- 5b. J. M. Howie, et al., "Programming for HPD and Other Flying Spot Devices," Rept. CERN 63-34, Data Handling Division, CERN, October 1963.

On Photographs Taken by the Argonne Group at CERN

6. Three papers which appear in Reference 2, G. R. Burleson, et al.
 - a. "Design and Construction of a Large Magnetic-Field Spark Chamber," pp. 180-184.
 - b. "Initial Operation and Performance of a Large Magnetic-Field Spark Chamber," pp. 185-192.
 - c. "Optical System and Data Processing for a Large Spark Chamber System in a Magnetic Field," pp. 448-451.

On Photographs to be Taken in the Ke_2 Experiment

7. P. I. P. Kalmus, T. A. Romanowski, Y. S. Kim, K. Reibel, P. Phillips, "Proposal for an Experiment at the ZGS: Search for the $K^+ \rightarrow e^+ + \nu$ Decay Mode and the Determination of the $Ke_2/K\mu_2$ Branching Ratio," Argonne National Laboratory, July 1963.

On Photographs to be Taken by the Argonne Magnetic Spark Chamber Group

8. A. Roberts, D. McCloud, G. Yurka, "Optical System for the New 70-Ton Spark Chamber System," Argonne National Laboratory, December 1963.

On the CHLOE Film Scanning Equipment

9. Donald Hodges, "CHLOE, An Automatic Film Scanning Equipment, Hardware Reference Manual," AMD Technical Memorandum No. 61, Argonne National Laboratory, November 1963.
10. Robert Clark, Donald Hodges, "ASI-210/CDC-3600 Communication Link," AMD Technical Memorandum No. 63, Argonne National Laboratory, November 1963.
11. Robert Clark, "Programming for the ASI-210/CDC-3600 Communication Link," AMD Technical Memorandum No. 58, Argonne National Laboratory, November 1963.
12. Advanced Scientific Instruments Division of Electromechanical Research, Inc., Programming Manual for the ASI-210 Computer System, Publ. 210-3, Minneapolis, Minnesota, April 1963.

On the CDC-3600 Computer

13. Six manuals provided by the Control Data Corporation, Minneapolis, Minnesota.
 - a. Control Data 3600 Computer System Reference Manual, Publ. 213b, November 1963.
 - b. COMPASS Reference Manual, Publ. 525a, August 1963.
 - c. SCOPE General Information Manual, Publ. 532, October 1963.
 - d. SCOPE Reference Manual, Publ. 533, August 1963.
 - e. 3600 FORTRAN General Information Manual, Publ. 534, October 1963.
 - f. Preliminary 3600 FORTRAN Reference Manual, December 1963.

On AROMA

14. J. W. Butler, Margaret K. Butler, Agnes Stroud, "Automatic Classification of Chromosomes," AMD Technical Memorandum No. 60, Argonne National Laboratory, October 1963.

On PAIR

15. Claude Berge, The Theory of Graphs and Its Applications, John Wiley, 1962.
16. Claude Berge, Topological Spaces, Oliver & Boyd, 1963.
17. Oystein Ore, Graphs and Their Uses, Random House, 1963.
18. Oystein Ore, Theory of Graphs, American Mathematical Society Colloquium Publications, Vol. XXXVIII, 1962.
19. W. H. Highleyman, "Linear Decision Functions with Application to Pattern Recognition," Doctoral Dissertation, Dept. of Electrical Engineering, Polytechnic Institute of Brooklyn, June 1961.
20. George S. Sebestyen, Decision-Making Processes in Pattern Recognition, ACM Monograph Series, Macmillan, 1962.
21. S. Vajda, Mathematical Programming, Addison-Wesley, 1961.
22. R. Bellman, R. Kalaba, "On k^{th} Best Policies," Journ. Soc. Ind. and Appl. Math., Vol. 8, 1960, pp. 582-588.

On LINK

23. R. Narasimhan, "A Linguistic Approach to Pattern Recognition," Digital Computer Laboratory Report No. 121, University of Illinois, July 1962.
24. Robert Clark, "Pattern Moments," AMD Program Library 1703/HEP 130, Argonne National Laboratory, March 1963.
25. Min-Juei Hu, "Visual Pattern Recognition by Moment Invariants," IRE Transactions on Information Theory, February 1962, pp. 179-187.
26. Franz L. Alt, "Digital Pattern Recognition by Moments," Journal of the ACM, Vol. 9, No. 2, April 1962, pp. 240-258.
27. John Riordan, An Introduction to Combinatorial Analysis, John Wiley, 1958.
28. H. J. Ryser, Combinatorial Mathematics, Carus Mathematical Monograph No. 14, John Wiley, 1963.
29. Kenneth E. Iverson, A Programming Language, John Wiley, 1962.
30. John M. H. Olmsted, Solid Analytic Geometry, Appleton-Century, 1947.