ORGANISATION EUROPÉENNE POUR LA RECHERCHE NUCLÉAIRE
CERN EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

PROGRAMMING FOR HPD AND OTHER FLYING SPOT DEVICES

held at the Collège de France, Paris
on 21 - 23 August, 1963

(organised jointly by the Laboratoire de Physique Nucléaire,
Collège de France, Paris and Data Handling Division, CERN, Geneva)

PROCEEDINGS
edited by
J.M. Howie
S.J. Mccarroll
B.W. Powell
A. Wilson

GENEVA
1963
Propriété littéraire et scientifique réservée pour
tous les pays du monde. Ce document ne peut
être reproduit ou traduit en tout ou en partie
sans l’autorisation écrite du Directeur général
du CERN, titulaire du droit d’auteur. Dans
les cas appropriés, et s’il s’agit d’utiliser le
document à des fins non commerciales, cette
autorisation sera volontiers accordée.
Le CERN ne revendique pas la propriété des
inventions brevetables et dessins ou modèles
susceptibles de dépôt qui pourraient être décrits
dans le présent document ; ceux-ci peuvent être
librement utilisés par les instituts de recherche,
les industriels et autres intéressés. Cependant,
le CERN se réserve le droit de s’opposer à
toute revendication qu’un usager pourrait faire
de la propriété scientifique ou industrielle de
toute invention et tout dessin ou modèle dé-
crits dans le présent document.

© Copyright CERN, Genève, 1963

Literary and scientific copyrights reserved in
all countries of the world. This report, or
any part of it, may not be reprinted or trans-
lated without written permission of the
copyright holder, the Director-General of
CERN. However, permission will be freely
granted for appropriate non-commercial use.
If any patentable invention or registrable
design is described in the report, CERN makes
no claim to property rights in it but offers it
for the free use of research institutions, manu-
facturers and others. CERN, however, may
oppose any attempt by a user to claim any
proprietary or patent rights in such inventions
or designs as may be described in the present
document.
<table>
<thead>
<tr>
<th>TABLE OF CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Preface</strong></td>
</tr>
<tr>
<td>Acknowledgments</td>
</tr>
<tr>
<td>Introduction</td>
</tr>
<tr>
<td><strong>L. Kowarski</strong></td>
</tr>
<tr>
<td>A general explanation of HAZE</td>
</tr>
<tr>
<td>including differences between</td>
</tr>
<tr>
<td>LRL, BNL, and CERN</td>
</tr>
<tr>
<td><strong>DISCUSSION</strong></td>
</tr>
<tr>
<td>FSD operating experiences at</td>
</tr>
<tr>
<td>Berkeley</td>
</tr>
<tr>
<td><strong>DISCUSSION</strong></td>
</tr>
<tr>
<td>A method of filtering HPD</td>
</tr>
<tr>
<td>data using &quot;Track following&quot;</td>
</tr>
<tr>
<td><strong>DISCUSSION</strong></td>
</tr>
<tr>
<td>Programming for the SMP</td>
</tr>
<tr>
<td><strong>DISCUSSION</strong></td>
</tr>
<tr>
<td>Off line treatment of HPD</td>
</tr>
<tr>
<td>digitizings</td>
</tr>
<tr>
<td><strong>DISCUSSION</strong></td>
</tr>
<tr>
<td>Performance of the fiducial-finding subroutine H-10</td>
</tr>
<tr>
<td><strong>DISCUSSION</strong></td>
</tr>
</tbody>
</table>

7164/mh
<table>
<thead>
<tr>
<th>Topic</th>
<th>Author(s)</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controlled filter and smoothing</td>
<td>D. Burd, N. Weber</td>
<td>77</td>
</tr>
<tr>
<td>DISCUSSION</td>
<td></td>
<td>83</td>
</tr>
<tr>
<td>Adaptation of FOG and CLOUDY to HAZE</td>
<td>D. Hall</td>
<td>85</td>
</tr>
<tr>
<td>DISCUSSION</td>
<td></td>
<td>91</td>
</tr>
<tr>
<td>The CERN version of MIST</td>
<td>S.J. McC Carroll</td>
<td>95</td>
</tr>
<tr>
<td>DISCUSSION</td>
<td></td>
<td>107</td>
</tr>
<tr>
<td>Adaptation of THRESH and GRIND for HPD</td>
<td>J.M. Howie</td>
<td>111</td>
</tr>
<tr>
<td>DISCUSSION</td>
<td></td>
<td>123</td>
</tr>
<tr>
<td>Future plans for CERN HAZE</td>
<td>W.G. Moorhead, W. Krischer</td>
<td>125</td>
</tr>
<tr>
<td>DISCUSSION</td>
<td></td>
<td>131</td>
</tr>
<tr>
<td>Attempts to obtain high rate, high quality measurement of events despite FILTER II imperfections</td>
<td>P.V.C. Hough</td>
<td>133</td>
</tr>
<tr>
<td>DISCUSSION</td>
<td></td>
<td>137</td>
</tr>
<tr>
<td>The automatic analysis of spark chamber pictures using a flying spot digitiser</td>
<td>P.N. Blackall, G.R. Macleod, P. Zanella</td>
<td>141</td>
</tr>
<tr>
<td>DISCUSSION</td>
<td></td>
<td>159</td>
</tr>
<tr>
<td>DISCUSSION</td>
<td></td>
<td>169</td>
</tr>
</tbody>
</table>
A spark chamber film reading system
W. F. Baker
171

DISCUSSION
183

SASS
D. Hall
185

DISCUSSION
189

A mathematician looks at bubble and spark chamber data processing
J. W. Calkin
191

DISCUSSION
197

Current status of the digital pattern recognition program at BNL
W. J. Beard, R. R. Harr
201

G. Rabinowitz, B. Weeser

(Abstract)

PEPR—Programming progress
W. Humphrey
203

DISCUSSION
207

DAPR
D. Hall
209

DISCUSSION
213

Computer association of tracks in different views
D. Burd
215

P. L. Connolly

DISCUSSION
221

PAN 3—The HPD diagnostic program
R. Glasser
223

S. J. McCarroll

DISCUSSION
229

PIP: A photo-interpretive program for the analysis of spark chamber data
H. Rudloe
231

T. Marill

N. Deutsch

(Abstract)

DISCUSSION
233
<table>
<thead>
<tr>
<th>Section</th>
<th>Author(s)</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concluding remarks</td>
<td>L. Kowarski</td>
<td>237</td>
</tr>
<tr>
<td>Appendix 1: Current status of the digital pattern recognition program at BNL</td>
<td>G. Rabinowitz</td>
<td>243</td>
</tr>
<tr>
<td>DISCUSSION</td>
<td></td>
<td>253</td>
</tr>
<tr>
<td>Appendix 2: A programming system for scanning digitized bubble chamber negatives (Abstract)</td>
<td>R. Narasimhan</td>
<td>255</td>
</tr>
<tr>
<td>Appendix 3: Luciole</td>
<td>H. Anders, T. Lingjaerde, J.A. Wilson, T. Wiskott</td>
<td>259</td>
</tr>
<tr>
<td>List of participants</td>
<td></td>
<td>265</td>
</tr>
</tbody>
</table>

---

7184/mh
A few explanations are needed since these proceedings do not correspond exactly to what was said at the meeting. With regard to the discussion after each paper, it must be emphasized that in the majority of cases, speakers have not had the opportunity to check the final version of the text. We have thought it desirable to contract and re-word parts of the discussion and therefore all inaccuracies should be attributed to us and not to the speakers themselves.

Only an abstract for the paper by Rabinowitz was received and since this paper was of considerable interest we have transcribed the tape recording and this is reproduced in appendix 1 together with the discussion which followed it. Also, we have included in appendix 3 a paper by Anders et al. on Lucicle which was not given at the meeting due to lack of time.

No manuscript was received for the paper on the SPASS system by Rudloe but since this closely followed the article by Rudloe et al. entitled: "PIP: A Photo-Interpretive Program for the Analysis of Spark Chamber Data (communications of the ACM, 332, 5, 1963), we have printed the abstract from that article and the discussion which followed Rudloe's talk at the meeting.

Finally, we have included the abstract of the paper submitted by Dr. R. Narasimhan entitled "A Programming System for Scanning Digitised Bubble Chamber Negatives" in appendix 2. Some copies of this report were distributed during the meeting and further copies are available from the Digital Computer Laboratory, University of Illinois, Urbana, Illinois, U.S.A.

B.W. Powell
Acknowledgements

It is a pleasure to record the cooperation of M. Bloch and M. Schiff of the Laboratoire de Physique Nucléaire, Collège de France in organising this meeting. Due to them and their assistants at the Collège de France, the meeting took place informally but with a minimum of difficulties.

In preparing these proceedings for publication, the work of editing has been shared between J. Howie, Miss S.J. McCarroll, A. Wilson and myself. A.Wilson who is a member of Rutherford Laboratory programming group, was a visitor at CERN during this period and kindly agreed to help us in this work. The general organisation for producing this report was done for us by Mrs. G. Andréossi.

We are also most grateful to E. Bissa for the tape-recording of the proceedings, to Miss M. Hutin for preparing drafts and stencils for this report and to Mrs B. Powell for her help with the figures.

B.W. Powell
21 August 1963

I. MORNING SESSION

Chairman: P.V.C. Hough
INTRODUCTION

L. KOWARSKI
CERN, Geneva

A ce début de notre conférence, il paraît opportun qu’au nom de tous les visiteurs j’exprime notre reconnaissance aux chercheurs du Collège de France qui nous accueillent ici. Je le fais dans ma double capacité de visiteur et d’ancien résident; assez ancien, en effet, car c’est il y a un quart de siècle que Frédéric Joliot m’initiait ici aux pratiques de la recherche avancée en physique nucléaire. Goldschmidt-Clermont, lui non plus, n’est pas étranger à cette maison où nous nous sommes rencontrés pour la première fois il y a 16 ans; et il nous est très agréable de constater, en revenant ici, que la tradition de cette maison continue et que la recherche avancée, si vivante dans le domaine qui nous intéresse aujourd’hui, y a pris racine et porte fruit - nous le verrons en détail au cours de ces journées.

This is - as you know - the fifth meeting in the series devoted to automatic devices for the evaluation of track chamber pictures. The series was started informally in November 1960 in Brookhaven where the collaboration on the flying spot digitizer of the Hough-Powell type was decided for the first time. The second meeting took place in CERN in May 1961, when the first pictures could be digitized at CERN thanks to that collaboration. The third meeting was held in January 1962 at Berkeley; the first results and already the first difficulties were frankly discussed. The fourth was part of the Informal Meeting on Track Data Processing which took place in Geneva a little over a year ago. The difficulties by that time had become more definite and it has taken us this time a little longer to sort them out and to get ready to discuss them squarely. After some 13 months, here we are again. The object of the meeting is now more precise than previously and significantly the question of programming is in the foreground. Thus it can be seen how this succession of meetings has mirrored the evolution of our art.

Let me start with a few controversial and personal statements. It will be fun to have them on record and then to see how they get invalidated by the results of this meeting; this will give us an opportunity to say something quite different at the end of the meeting. A few historical definitions will be useful to define my terms; I see four stages in this development. The first stage was that of the human observer unaided by any machine except perhaps a good inch rule. That
was the great line of the cloud chambers work; in the most exciting experiments there would be possibly one interesting picture a week—maybe I am exaggerating, anyhow the production was slow enough to give the physicist every time in the world to look at his picture from all angles, to measure and to make conclusions. The question of the speed of measurement did not arise at all.

The second stage was inaugurated around 1956 at Berkeley under the forceful impulsion given by Alvarez who was probably the first to see the importance of track chamber photographs to be taken in numbers which at the time appeared fantastically high. He saw quite clearly that one of the problems was to set up measuring procedures capable of dealing with these pictures at a vastly increased speed. So started the second stage—the human assisted by the machine; as far as working physics goes, we are still in it. The human observer sets the pace of the measurement and proceeds at his human speed; the auxiliary mechanical device is usually the digitizing projector which has been developed independently—or nearly independently—in Berkeley and at CERN under the respective names of Frischenstein and IEP. This device enabled the physicist to process pictorial events at one single laboratory in the numbers of tens of thousands a year with 100,000 or slightly over 100,000 as the ultimate goal beyond which it is understood that the system cannot and should not go. We all know how, in America, under the impulsion given by the Alvarez school and in Europe starting out from the CERN school led by Goldschmidt-Clermont, this system came to be used everywhere and although it is realized that another transition to a higher speed is necessary, we are not quite there yet.

The next stage aimed at transcending the limit of 100,000 events per year; say, a factor 10 more if possible, but to achieve this, mechanization had to be undertaken on principles where the machine would impose its own speed. For that, the best idea would be to do away with the human altogether but nobody knew yet precisely how to do that. Paul Hough was, and I think that was fundamentally his merit, the first to realize that instead of dreaming of completely mechanized systems, one had first to see whether human discrimination could not be used in a way which would not slow down the proper pace of the measuring machine. Thus the Hough-Powell system was born, and with it the third stage—the machine assisted by a human. Thinking on somewhat different lines, Alvarez produced another principle and we have now the two extremes with, possibly, some intermediates in between. On one end, the Hough-Powell system segregates the machine from the human; it puts the human operation in its allotted corner and lets most of the machine operation work without the human hampering it. On the contrary, in the
Alvarez system, the human operator still is in the center of things but the machine is arranged so that it speeds up the human operation, and corrects its inherent lack of accuracy. The two systems - the HPD, now increasingly called FSD, and the SMF system of Alvarez are now just beginning to do physics. They came into useful physical operation not as quickly as we could expect; in addition to their obvious hardware aspect there is also the programming or software side and the physicists who originated all these systems, although realizing that software is important, did not have it in their blood (if I may mix my metaphors). Though himself started out as a pure hardware man and became addicted to software rather recently. The hardware has already been working for some time in several centers; it works as most of the physicist-designed hardware works, that is well enough to generate the overwhelming impulse to take it to pieces only once in every few months - but now it is on the way to being finally engineered by professionals. It looks then - forgive me for being so optimistic - as if the hardware problem has been solved. But on the software side we are - frankly speaking - still in a jam and that is why we are assembled here: to swap our heaps of unfortunate experience and to point out some happier glimmerings.

It is interesting to speculate on what would have happened if this system had been developed originally not in physics laboratories but in data processing laboratories proper, for example by IBM. Software problems would then probably have been attacked far earlier, most likely in a more workmanlike way, and possibly they would have been solved earlier. It is also possible that since physicists would have to do less with this development, the system, as developed, would be finally of a rather remote interest to physics; such things do happen. The remedy against that might have been a very close contact between the data-processing specialists and the physicists; this is difficult to achieve mainly because the data-processing specialist should really be in the saddle and few physicists like to work when somebody else is in the saddle. In this respect an interesting lesson can be derived from what happened in Berkeley, where awareness of data-processing questions was from the start definitely higher than that obtaining in most other physics laboratories. It was in Berkeley that the physicists have collaborated with the data-processing experts in the most fruitful way and therefore it is in Berkeley that the FSD system at present is working best, as we shall hear. The words "Working system" have to be used with that bit of indulgence which is proper for a pioneering effort in a very new field, but when speaking of Berkeley it seems that soon we may do with less and less indulgence. Elsewhere the physicist and the hardware developers are painfully
learning that software is important; eventually the time lag - we hope so - between hardware and software will be gradually overcome and working systems will come into existence in other laboratories as well. Meanwhile, the other system, the one in which the human being is in the center, has been put into operation in Berkeley again, and this is again significant. One can already see - what I had once said on an earlier occasion - that FSD is fundamentally more machine-minded and will develop in a more purely machine-minded direction, whereas SMP is more human-minded and therefore it seems likely that SMP will remain important in the kinds of experiments where human intervention is very essential - that is, for more variegated events of a less predictable kind in which more human decision on the spot is necessary, while the FSD will develop towards the handling of very high numbers of standardized and predictable events, where the essential physics comes from the statistics more than from the individual features of an event.

Let me venture here a controversial statement with which perhaps the SMP people may not agree: it seems that the FSD in principle is better suited for this last transition to the fourth stage, where there is only a machine and no human on line. If the mixed system - the third stage, machine plus human - is already capable of supplying something up to a million events per year, one may ask why go any further. Well, there will be kinds of physics - I feel sure of that - in which there will be more than one million events per year needed, and this is especially true of spark chamber physics. We shall then need a system which can dispose of a picture - bubble or spark - in something like a second, which leaves no time for any human intervention. The fourth stage, the entirely mechanized system is well under way. As in the third stage, but in a different sense, we observe here a curious parting of the ways: first, there is the philosophy which wants to do the job by using a tricky kind of hardware. For example, the flying spot itself can be subjected to control by the computer, through a program adapted to the particular kind of physics we want to be done. The spot will then extract from the photograph only the elements relevant to that particular physical problem. This is what I might call the tricky-hardware solution. The other school wants to transfer everything even remotely significant from the picture first into the computer and apply all the necessary tricks strictly to the software, leaving the photograph alone or even doing away with the photograph altogether by using the acoustic techniques, vidicons and so on. Well, it can be said, almost as an abstract theorem, that if any physical requirement is expressible in terms of any trick which is incorporated in the hardware, then the same trick logic must be applicable to the software alone. I would like some logician to prove this.
theorem; to me it appears as self-evident as sometimes fallacies are, and I am rashly inclined to take it for granted. We assume, then, that any scanning and measuring system which uses a special flying spot, a special controlling unit and in which the memory is constituted by the photograph — any such system can be duplicated by using a blind serial scan — TV fashion — and then applying the same kind of logic strictly to the software. In one case you need a connecting electronic system and a small computer which works according to a commanding program and controls the hardware in a rather sophisticated mode. In the second system you need a simple, sturdy, reliable digitizer and then a computer with a big memory. It is interesting to observe how these two schools of our "fourth stage" are developing at present. Here again we see the phenomenon that these systems — at least in high energy physics — are mostly handled by the physicists and not by the data processing specialists. The physicists, as I said before, are more at home with hardware than with software; therefore most of the systems at present being developed (at least in the Western world, I don't know much of what happens in the East) — belong to the Tricky Hardware school. The first automatic device which has done physics at all is a spark chamber system developed by Deutsch at MIT. It was quite fitting for this home of many ideas not only on physics but also on data processing, and for this very versatile physicist and developer, to be at the origin of this first system to come into successful operation. At present, if I may be permitted a somewhat jocular appreciation, this system still needs a Deutsch on line, but there is little doubt that, with proper development, it will be able to free itself from this expensive servitude and will become a true automatic and non-human process. There have been proposals and developments aiming at handling bubble pictures in the same way; in particular a very definite effort is going on at Argonne in this direction, starting with sparks but aiming also at bubbles.

The other school, that of the Tricky Software, has so far led, to my knowledge, to one physical application which I shall presently mention; like its Hardware colleague, and so far with a somewhat less impressive record of achievement, it has been applied to a spark chamber problem. On bubbles, the Tricky Software development was started, I think, by Pasta in 1961 at Brookhaven. This system shows promise; it is being actively worked upon both in Brookhaven and in Berkeley, but (if I may venture another controversial statement) it seems that the pure in the soul physicists look at it as yet with some misgivings.

* called glamorous DAPR at Berkeley. The Brookhaven system, according to the names of its three developers, could be called Pastrani, but a reference to this somewhat simple delicacy may appear uninspiring.
Another interesting development is going on under McCormick at the same laboratory where Pasta is now (University of Illinois). Eventually it will require a piece of very special hardware - the McCormick pattern-recognizing computer - but there are also some ideas, of which we know so far rather little, of using the 7090 as a kind of first stage in the development of the system. This seems to put it definitely in the Tricky Software class. Other tricky-software systems are being studied specifically in connection with the spark chambers, in particular in Brookhaven (with a mechanical digitizer) and also in CERN, where a cathode-ray scanner is being developed under the name of Lucile. The trickiness of its software is at present being studied with the provisional help of the Hough-Powell digitizer which was originally developed for bubble chambers. We have obtained some results in CERN with this provisional system; some real physics events have very recently been processed in this way. Later on, most probably, the Lucile Scanner will replace HPD for these purposes, but the latter may have to be retained for some classes of spark chamber experiments presenting special requirements of accuracy or format.

To sum up, it can be said that the Fourth Stage, or the entirely non-human scanning and measuring, has definitely been achieved in some physics done with spark chambers, and this goes both for the Tricky Hardware and the Tricky Software approach.

To wind up my introduction, I will say this again: if data-processing specialists had taken a greater interest in these problems and had managed to enroll the support of the high-energy physicists, the software both for the third and the fourth stage would be today in a better shape.

Recently I was asked whether, for these professionals, it was too late to pitch in. Well, I wish I could say it is too late. I hope that all of you here present have by now carried the development to so high a level that it is really no longer rewarding for a great body of data-processing interests to get involved in its turn. I hope so but I am not quite sure of it. This meeting in particular should enable us to see again whether and to what extent the data-processing experts proper should take a hand and how the very necessary collaboration with working physicists could be ensured.
A GENERAL EXPLANATION OF HAZE INCLUDING DIFFERENCES BETWEEN LRL, BNL AND CERN

D. HALL
Lawrence Radiation Laboratory, Berkeley

Before plunging directly into a discussion of the HAZE program, it might be well to first review the overall operation of the FSD system.

The FSD system is designed to produce a large volume of very high precision measurements of bubble chamber photographs. The system consists of the following four basic elements (figure 1).

1. The FSD itself.
2. A digitizing scan-table.
3. An IBM 709 or 7094 computer.
4. The HAZE program.

The system operates as follows:

1. A roll of film is placed on the scanning table. At Berkeley all three views of an event are contained on the same roll of film. At Brookhaven and CERN, the individual views are on separate rolls so that three rolls of film are mounted simultaneously.

2. When the scanner has found an event, he indicates the rough position of the tracks by first digitizing the fiducials and then digitizing three points on each track. The scanner also records the picture or frame number, the experiment and subexperiment number, the Assignment List or event type number (or in the case of CERN, the GRIND library number). He may optionally record the particle code of the event, whether or not a range momentum measurement may be used and chain linkage information. There is also provision for 80 characters of comments which are entered through the Flexowriter. This procedure is carried out for each of the views of the event. This information is recorded on magnetic tape or on IBM cards to be used by the HAZE program at the time of measurement.

3. At the end of a 24 hour scanning period, the scanning data are processed through an editing program which removes error records and provides detailed scanning statistics. At CERN, this program also converts the scan-table digitizings to rectangular coordinates. This program produces a magnetic tape for each of the rolls of film scanned during that period.
4. The edited scan tape is mounted on the computer and the corresponding roll of film is mounted on the FSD. The HAZE program is loaded into the computer and measuring begins.

5. The HAZE program produces 3 output tapes: one for each view. At CERN since the views are on separate rolls of film, the number is kept down to one by merging the view currently being measured with a tape containing the previously measured views. This is not done at Berkeley because of the way in which views and frames are interlaced on the film. Sorting them by advancing and reversing the film on the FSD so that all the views of an event were measured together would cause intolerable delays.

Running FOG simultaneously with HAZE would require 9 such tapes. Since these tapes are essentially scratch tapes anyway, we plan at Berkeley to use disc storage instead.

6. The first operation is to run FOG and CLOUDY, or in the case of CERN, THRESH and GRIND, on the HAZE output. Eventually these programs will be multiprogrammed to run concurrently with HAZE.

With these introducing remarks, we turn now to the organisation of HAZE itself.

The dispatcher portion of the HAZE program (Figure 2) is responsible for synchronizing all of the pieces of the system. The first two sections of the dispatcher take care of initializing all of the parts of the system. This includes:

1) Connecting the FSD to the Direct Data Channel.
2) Enabling traps.
3) Initializing buffers.
4) Labeling all output tapes.
5) Checking labels on all input tapes.
6) Deriving, from the basic constants of the hardware, constants for the various transformations.

Sections 03 and 04 of the HAZE program initialize for the processing of a view. This includes:

1) Finding the next view on the scan tape.
2) If a new scan tape or a new roll of film is required a statement to that effect is printed online.
Section 05 starts the FSD machinery into motion. The following operations occur:

1) Working storage for GATE and FILTER is initialized.
2) Working storage for the road fitting routine is initialized.
3) Fiducials and roads for the entire view are input and converted to an orthogonal micron coordinate system.
4) Circles are fitted to the road points, and the type of scan (normal, orthogonal or both) is determined. If a track is too short, it may be necessary to use the every point mode of weeding. This is also determined at this time.
5) Finally, the order to begin measuring is sent to the FSD together with the mode, speed and density.

There are two 2000 word buffers (called BUF A) in HAZE for FSD input. During a measuring scan, one buffer is used for input while the other buffer is being worked on. When the first buffer is full, the roles of the two buffers are interchanged and the second buffer is filled while the first buffer is being worked on.

The program now enters section 06 which sends an IOCT BUF A*) 2000 to the data channel. The channel D traps are enabled and the program enters the pause loop. This section of the program consists of a commutator loop which executes the non-priority programs occupying the memory with HAZE.

When the word count in channel D has been reduced to zero a trap occurs causing the program to enter section 07. This section of the program tests the channel for transmission errors, and for an End of File. An End of File sends the program to section 16. Otherwise control goes to section 06. If the road fiducials have been found, control goes to section 13. Otherwise the program searches for the road fiducials via H-10. During the search it may be necessary to wait for more data from the FSD. In this case, control returns to the pause loop.

*)
BUF A means the appropriate BUF A in the sense of the one ready for data.
When the road fiducials have been found, the roads are transformed into the FSD coordinate system. At Berkeley this transformation uses two fiducials so that the transformation is a translation, a rotation and a magnification. At BNL, only one fiducial is used so that the transformation is a simple translation with a constant rotation.

Section 11 tests for the existence of any normal mode tracks. If no tracks require a normal scan, control goes to section 15. Otherwise control continues to section 12.

Section 12 assigns BUF B, BUF C and BUF D storage blocks for each track being scanned in this mode, then transfers control to the gating subroutine.

The gating subroutine examines every point in the current BUF A against the road edges of each of the assigned tracks. If a point falls inside the road edges for a given track, the point is stored in the corresponding BUF B and a point tally is incremented. When the point tally reaches 20, the filtering routine is called for that track. A 20 point segment of track is called a byte.

The filtering routine (figure 3) generates a histogram of the 20 points and searches the histogram for pulses. There are two thresholds involved in this search, the noise threshold $T_N$ and the pulse threshold $T_P$. A pulse is defined as follows:

Consider a set $P = (b_{i+1}, \ldots, b_{i+n})$ of $n$ consecutive histogram bins such that:

$$C(b_{i+j}) \geq T_N \quad \text{for} \quad j = 1, \ldots, n$$

$$C(b_{i}) < T_N \quad \text{and} \quad C(b_{i+n+1}) < T_N$$

Then $P$ is called a pulse if

$$\sum_{j=1}^{n} C(b_{i+j}) \geq T_P$$

In the initialization phase, FILTER searches the entire histogram for pulses. Having found a pulse, or at most two pulses, FILTER begins following the track, (or tracks). In the track following phase, a much smaller search area is used. This area is determined from the position of the preceding pulse by adding an increment $I_P$ to the preceding region. Thus, if the previous pulse were in cells $i + 1$, $\ldots$, $i + n$, the new search area could be from $b(i + 1) - I_P$ to $b(i + n) + I_P$.

The points in the pulse are normally averaged to form master points. The master points are given to the WEED subroutine in FILTER which keeps the number of master points taken from exceeding a prespecified number. If the point was not weeded, it is stored in the appropriate
BUF C. If a track is very short, however, it may not be possible to get enough master points. In this case, each individual digitizing is treated like a master point in the weeding process.

A number of special cases arise:

1. The pulse is too wide. (At Berkeley more than 3 bins). In this case no master point is taken, but the pulse is used to predict the next search area.

2. There are no pulses in the search area. In this case the subroutine COAST is called. COAST keeps a tally of the number of consecutive bytes which have had no pulse in the histogram. When this tally exceeds 3 the track is rejected. If the tally has not exceeded 3 the old search area is maintained and FILTER waits for the next byte.

3. There is more than one pulse in the search area.
   a) If only one track was previously being followed, FILTER assumes that a fork has occurred and sets up storage for another track to be followed.
   b) If two tracks were already being followed FILTER attempts to coast past the region of difficulty.

When all of the points in BUF A have been gated and filtered, the program searches for internal fiducials and returns control to the pause loop. When the entire view has been scanned, an End of File is sent by the FSD and control transfers to section 16. Here the gating and filtering operations are carried out on the last buffer, final internal fiducial searching is completed, and control transfers to section 18.

In section 18 the filtered points are prepared for output. This includes converting the points to microns, and correctly re-joining tracks which were scanned in both modes.

The program now asks if there are more scans to be done. If there are, control is transferred to section 15 where the next set of BUF B and BUF C are prepared. The appropriate mode, speed, density and position are sent to the FSD and control returns to the pause loop. If there are no more scans to be performed, the output is written out onto the magnetic tape appropriate to that view.

If sense switch 3 is up, the program automatically processes the next view by transferring to section 03. If sense switch 3 is down the operator has requested an interrupt. In this case, final summaries are generated and control returns to the FOG monitor system.
This is the general flow of the program, and is nearly the same at all three laboratories. There are some additional features to the program which we find quite useful at Berkeley for diagnostic purposes.

1. Sense switch 2

Depressing sense switch 2 causes the HAZE program to dump each of the histograms together with the corresponding BUF B and BUF C points onto A2 in a binary format. These tapes are then prepared for printing by another program which displays the histogram with the filtered pulses indicated by an equals sign.

2. Sense switch 4

Depressing sense switch 4 causes the program to stop just before the event is written out. If SSW4 is left down and START is pressed, the program will reprocess the same view. If SSW 4 is put up and START is pressed, the program will write the output and process the next view.

We find this to be a useful way to keep trivial errors from spoiling an event.

3. Sense switch 6

Depressing sense switch 6 causes the HAZE program to print the view summary online. This information consists of only about 6 - 10 lines per view and provides a relatively cheap method to monitor the quality of the measurements. In practice we find that putting SSW 6 down about every 5 - 10 minutes gives a good statistical sampling of the events.
Figure captions

Fig. 1  The data flow in the PSD system.

Fig. 2  The summary flow chart for the HAZE program.

Fig. 3  The flow chart for the FILTER routine.
Figure 1

film

Scanning table

Scan tape A

(7094) Editing program

Scan tape B

FSD

(7094) HAZE program

view 1  view 2  view 3

(7094) Reconstruction Kinematics

Output
Figure 2
Filter

Compute Histogram

Track initializes

N

How many pulses

V

Initialize appropriately

GATE

Search entire histogram for pulses

Tally and try again

Rej

Y

GATE

How many pulses

2

Y

Assume fork. Set up to follow two tracks

Rej

N

Pulse region

N

Compute Master point

COAST

Rej

Y

Compute Master pt. Track 1

Compute Master pt. Track 2

Regions overlap

N

Any pulse in both regions

Y

How many pulses in both regions

0 or > 1

Compute Master pt. Track 1

Rej

COAST

How many pulses in both regions

Y

Compute Master pt. Track 2

COAST

Rej

N

How many pulses in union

2

Coast t=1

Coast t=2

No

Have they crossed

Y

Turn off. Carry on both

Cross them

Rej

Rej

Compute Master pt. Track 1

Compute Master pt. Track 2

Figure 3
DISCUSSION

LOYAL: Do you cross tracks as soon as you have two pulses in the same road?

HALL: Not in a road. If your regions overlap or if one pulse is in both regions, or if there are two pulses within epsilon of each other, where epsilon is a constant which defines the overlap, then you have a cross.

EURD: Is it possible to have two tracks merging?

HALL: Unless they come apart again at the end, one is crossed out.

BURDEN: What is the size of histogram bin you use, what are the values at $T_N$ and $T_P$, and do you put any restrictions on the width of a bump to call it a pulse?

HALL: The bin width is 16 least counts of 1 $\mu$, and the same width is used at Brookhaven. We take $T_N = 2$, $T_P = 5$ - that is one point in a bin is rejected but 2 is accepted, and there is a minimum of 5 in the total pulse. The width of the pulse must be $\leq 3$ for it to be accepted.

HOUGH: If you are following two tracks in a road, then you have 10 points per track, and yet you have to have 5 in the total pulse. This is what drives you down to low thresholds.
FSD OPERATING EXPERIENCES AT BERKELEY

D. HALL
Lawrence Radiation Laboratory, Berkeley

Volume and type of events

At the time of the report, the Data Handling Group at Berkeley had measured 4200 events on two separate experiments.

For the Trilling-Goldhaber Group there were 1050 events consisting of 90% 4 prongs and 10% strange particle production.

For the Powell-Birge Group there were 3150 events consisting of 95% 2 prongs and 5% 4 prongs.

Measurement quality

The quality of measurement has been compared with Franckenstein data as follows. A sample of 65 events from the Goldhaber experiment were measured on the FSD and also measured on the Trilling-Goldhaber Franckenstein and reduced through the PACKAGE programs. The following table shows the comparisons:

<table>
<thead>
<tr>
<th></th>
<th>Beam</th>
<th>Non-Beam</th>
<th>Dip</th>
<th>Azimuth</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\sigma_{P_{FOG} - P_{PACK}^b}$</td>
<td>$\sigma_{P_{FoG} - P_{PACK}^B}$</td>
<td>$\sigma_{\gamma_{FOG} - \gamma_{PACK}^b}$</td>
<td>$\sigma_{\gamma_{FOG} - \gamma_{PACK}^B}$</td>
</tr>
<tr>
<td>unconstrained</td>
<td>.045 GeV</td>
<td>.030 GeV</td>
<td>.20°</td>
<td>.15°</td>
</tr>
<tr>
<td>constrained</td>
<td>.014 GeV</td>
<td>.004 GeV</td>
<td>.08°</td>
<td>.15°</td>
</tr>
</tbody>
</table>

The somewhat larger $\beta$ spread is probably explained by a slight rotation between the two coordinate systems in the $x - y$ plane. The same mass hypothesis was picked on every event.

Speed. The number of seconds per event per program is given in the following tables:

<table>
<thead>
<tr>
<th></th>
<th>HAZE</th>
<th>FOG</th>
<th>CLOUDY</th>
<th>FAIR</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>T - G Expt</td>
<td>30.9</td>
<td>0.5</td>
<td>2.0</td>
<td>2.6</td>
<td>36.0</td>
</tr>
<tr>
<td>P - E Expt</td>
<td>22.2</td>
<td>0.5</td>
<td>2.0</td>
<td>2.6</td>
<td>27.3</td>
</tr>
</tbody>
</table>

7184/*8
By combining HAZE, FOG and CLOUDY in a multiprogrammed way, the total time would reduce to 31 seconds for the 4 prongs and to 22 seconds for the 2 prongs.

These figures yield the following rough estimate of timing.

\[ T_E = 4M_s + 3M_v \]

where \( T_E \) = total elapsed time in seconds for HAZE
\( M_s \) = total number of scans
\( M_v \) = total number of views processed

We note that the 3 second film advance time would be significantly reduced by the addition of a tandem FSD. The time per event is then simply \( 4M_s \).

**Sustained rate**

For the past 2\( \frac{1}{2} \) weeks the FSD has been turning out 2500 events per week, which corresponds to 100 events per hour of computer time. We expect to increase this to at least 120 events per hour with some experience.

**Volume of scanning**

We find that our scanners are able to scan at the rate of 12 events/hour. We are now using one scan table 84 hours per week producing 1000 events per week. By the end of this week the second scan table should be operational. We plan to use both scan tables a total of 300 hours per week to produce 3600 events per week. The third scan table is scheduled for installation on the 1st of September, and the 4th scan table is due on the 15th of September.

**Percentage successful**

For the two prong events we observe that a total of 83\% of the events scanned generate acceptable FAIR output. For the 4 prong events we find that 61\% of the events have successful FAIR output. A breakdown of the errors for the 2 prong events is given in the following table.
A similar breakdown for the 4 prong events has not yet been performed because of the extremely time consuming nature of the task. I would assume that the general levels are about the same and that the higher rejection rate is due to the larger number of tracks per event.
Scan Table  
(i.e. Hardware. Chiefly parity errors and scalers going out)  
13%  

Scanner  
(Failure to digitize fiducials, failure to digitize all three views, etc.)  
2%  

FSD  
(Repeated wrong positioning at the film, Ferranti errors etc.)  
2%  

HAZE (not FILTER)  
5%  

FILTER  
51%  

Fiducials  
(Repeated failure to find a fiducial. Or finding the fiducial in the wrong place)  
15%  

FOG  
Track directions in the two views are reversed  
10%  

Most of the fiducial rejects are due to the fact that there are two very close fiducials in the following pattern:

\[ \times \times \]

Since the current search area somewhat overlaps into the next fiducial, the E-10 subroutine may misidentify one or both of the fiducial arms. Since the film positioning accuracy is much better than the present search area indicates, we are confident that reducing the search area will eliminate this problem.
Breakdown of FILTER rejects

We have found the following patterns to be the most typical of the FILTER rejects (see figure 1).

Problem I is clearly an initialization problem. It would seem that a slightly more sophisticated procedure, such as continuing to search for new tracks for the first 3 bytes, would solve this problem.

Problem II is also soluble since the pattern is predictable. This pattern results from tracks which strongly violate the circle hypothesis; i.e., tracks which curve through many degrees of arc. The pattern arises as follows:

In principle one can predict the direction and magnitude of the shift, and although this is probably not the best method from a programming standpoint, A simpler procedure would be to allow $I_r$, defined earlier, to vary as a function of the number of degrees of arc through which the particle travels.

Problem III is more difficult, but also soluble. There is a routine in FILTER which rejects wide pulses from the master point calculation. P.V.C. Hough has suggested that a "stiffness" criterion should be added to this routine. That is, if on the basis of the preceding pulses one knows that a radical departure is erroneous, then the pulse should be rejected not only from the master point calculation, but also from the calculation of the next search area. This procedure would therefore also solve problem IV.

It remains to define an adequate "stiffness" criterion. Several proposals have been made, but none have been finally adopted.

1. Use the same approach as problem II.

This may in fact reduce the frequency with which problem III occurs, but it could never solve the following case.
2. Predict the position of the pulse based on a least squares fit to the preceding master points.

This method would certainly work, but it would also require a great deal of time per byte.

3. Estimate the standard deviation of the preceding pulses about zero and compare this distance to the deviation of the current pulse. The chief objection to this method would be its failure to detect the following case:

\[ \text{[Image: Diagram of pulse deviation]} \]

If one included the slope, the method would essentially be the same as 2.

4. Allow FOG to reject the bad points as is done in the case of Frankenstein measurements. This method has the advantage that it would be easy to solve, and it would certainly solve the problem. However, it does nothing for problem IV.

5. Use the preceding 3 points with parabolic extrapolation to predict the true value of \( W \) given \( X \). This method is quite time consuming. It would require 2 divisions, 3 multiplications and 5 subtractions for each master point. This does, however, get around the problem of using a parabola over the entire length of track.
Figure caption

Fig. 1  A schematic representation of the various faults encountered with the present FILTER program.
Figure 1

O = pulse
Q = filtered pulse
DISCUSSION

HOUGH: So there is nothing in common between your run and the measurements made with Franckenstein and PACKAGE?

HALL: That is right. The only thing in common is the 7094.

MULLER: Is this dispersion an instrumental error or just a dispersion of the beam?

HALL: They were the same beam tracks each time, so it is an instrumental error.

MULLER: In the 80 cm chamber in Geneva, with conventional IEPs, we can get 1%, you have a chamber twice the size and yet you get about the same precision.

BLOCH: What momentum is this beam?

HALL: K⁺ at 4.0 GeV/c.

BLOCH: Then you should get 1/4%, but perhaps the 72" chamber does not give such good measurements.

HALL: That is possible. One of our early problems was that of distortion. There seemed to be a small random distortion from picture to picture of the order of 20 μ. We found at first that we did not get any agreement at all, but this was connected with using a bad corner of the chamber for reconstruction. Then we chose a different fiducial region and got this agreement.

HOUGH: We measured some tracks with the new system and on comparison with the conventional method we found that the curvatures agree to less than the standard error in either. There is a difference in the external errors, and we have not gone back to the external error formulae to relate multiple scattering to the precision of the measurement.

GOLDSCHMIDT-CLERMONT: I understand these are the r.m.s. differences between the two systems? If this is a systematic effect, you should also look at the difference in the means.
HALL: I agree, but we have not yet had time.

EDMONDS: Could you give some idea of the total time per event taken by the program?

HALL: For 4 prong events, HAZE takes about 31 seconds, and for 2 prong events, 22 seconds. This time includes the sweeps for both modes for three views, and the time to advance the film. I took the total time for a trial run and divided by the number of events (about 500). Also, about 80% of the 4 prong events needed an abnormal scan and 10% of the 2 prong events. The 31 seconds breaks down into 4 secs/scan plus 3 seconds/view film advance time.

LORD: Were you having to advance long distances between pictures?

HALL: We are getting about 400 events/roll out of a total of about 500 frames.

BURREN: At what angle do you decide a track needs an abnormal scan?

HALL: 45° and a track which has even 1° in the abnormal mode costs us an abnormal scan.

HOUGH: 2 prong events on 2.2 GeV/c K^- film have about 50% requiring abnormal scans, so we have 10%, 50% and 80% for different kinds of experiment.

BLOCH: Why do you chose 45°?

HALL: We can probably adjust this to about 60°. There is also a provision which Moorhead has written, which we shall try, which will attempt to do a track in the normal mode even if it is partly abnormal.

SCHIFF: Can you give an estimate of the program time for HAZE, since what you give here is real time?

HALL: Some time in the 31 seconds is wasted, but I do not know how much.

SCHIFF: Do you time share HAZE?

HALL: No. What are the times at Brookhaven where less time is spent advancing the film?
HOUGH: Not allowing for overcovering the picture, we have 3 seconds of actual scanning time and a second for stage retrace and film advance. Generally the film advance beats the stage. This is with 40 \( \mu \) line spacing.

MULLER: What is the price of your scanning table?

HALL: About \$ 30,000.

HOUGH: Our basic scanner costs \$ 18,000 and the Datex unit which does the sequential punch about \$ 9,000. The rough digitiser costs about \$ 3,000.

POWELL: We reckon about \$ 15,000 for the complete machine.

BURREN: Does your geometry program have a points-out-of-line check?

HALL: The one which runs with HAZE output does not, it only tells us how well we fit the track.
A METHOD OF FILTERING HPD DATA USING "TRACK FOLLOWING"

J.W. BURREN and M.J. MITCHELL

N.I.R.N.S., Rutherford Laboratory, Chilton

(presented by J.W. Burren)

1. Choice of slice length

For the purposes of filtering, tracks must be divided into segments. We first discuss the three main factors affecting the choice of the length of these segments, which we call "slices".

i) There must be enough points on the track to distinguish the track from background by a histogram. The number of points depends on

a) the number of scan lines in the segment,
b) the bubble density of the track.

ii) For simplicity and speed of calculation, the track should not curve significantly within the slice. Curvature gives rise to two difficulties, firstly an error in the coordinates of the filtered point, though this is made smaller by the use of curved road edges as discussed in the next section, and secondly an increase in the width of a bump in a histogram.

Figure 1
The normal error in the coordinate is given by $\Delta S$ as shown in figure 1 and

$$\Delta S \sim \frac{4-6 \frac{L^2}{R}}{\text{microns}}$$

where $L$ is in mm and $R$ in cm. The factor 4-6 depends on the distribution of points along the curve and the exact method of calculating the filtered point. This error should really be less than one micron, but it could be larger if we apply the correction to the coordinates.

The width of the histogram is given by

$$\sim \frac{12 \frac{L^2}{R}}{\text{microns}}$$

and we should like this to be less than 20 microns, that is, less than a track width.

iii) We must obtain enough filtered points for the geometrical reconstruction program.

We have decided on a system of dividing the picture into slices of 32 scan-lines and filtering all roads after each slice. The length of segment is then in the region 1 - 2 mm long and certainly satisfies criteria (i) and (iii). We allow double length slices for sparse tracks. Criterion (ii) is sensitive to $L$, the length of the slice. For $L = 1$ mm, curvature can be ignored for tracks with radii greater than about 2 cm on the film.

2. Getting one slice

Road regions are defined for a slice by three parameters. The region is a parallelogram with a base made by the road edges $YR_0$ and $YR_0'$ on the first scan-line of the slice. The edge makes an angle $\theta$ with the x-axis. (See figure 2).

![Figure 2](image-url)
Gating consists of finding all the points in the roads and then "histogramming" these points. To make histograms, the road is divided into strips parallel to the road edge, as shown in figure 2. For each strip we make four counts.

i) The total number of points in the strip = $\Sigma 1$

ii) The sum of the x-coordinates of the points in the strip = $\Sigma x$

iii) The sum of the y-coordinates of the points in the strip = $\Sigma y$

iv) A 32-bit word for each strip to show on which scan-lines points in the strip have been found. This is used for finding vertices, and possibly for "gap" measurements.

The coordinates of points found in the road are not stored but are added immediately into the histograms for the road.

The gating process is governed by four parameters:

a) $YR_0$ = position of the road

b) $YR'_0 - YR_0$ = width of the road

c) $\tan \theta$ = slope of the road edge

d) The width of the histogram strip - the histogram block size.

A further modification has been made recently namely to introduce curved road edges. Instead of allowing $\tan \theta$ to remain constant in the slice, if $\Delta \tan \theta$ is the change in $\tan \theta$ along the slice, the slope after the nth scan-line is taken to be $\tan \theta - \frac{1}{2} \frac{\Delta \tan \theta}{64}$

and this allows the road edge to follow the track more closely. This can be achieved with the addition of only 5 cycles/scan-line/road.

3. Filtering

At the end of each slice, filtering is carried out. The operation of filtering consists of two main parts. Firstly, the histograms for the road are examined to try to find a filtered point for the slice, and secondly, the four "gating" parameters are adjusted for the next slice.

To find the filtered point, the point count histogram is scanned. All blocks with less points than a given threshold are ignored and consecutive blocks above the threshold are added together to form a "bump".
There may, of course, be more than one bump in the histogram. For a bump to be good enough for a point, it must have an area (number of points) greater than a set value and a width (number of blocks) less than a set value. From the other histograms we find $\frac{\sum x_i}{N}$, $\frac{\sum y_i}{n}$ for the points in the bump, where $N$ is the number of points in the bump. The point

$$x = \frac{\sum x_i}{N}, \quad y = \frac{\sum y_i}{N}$$

is a point on the straight line fit to the points in the bump and this is taken as the filtered point. The special cases of no good bump or more than one good bump will be dealt with later.

Next, the four gating parameters are adjusted for the next slice.

1) The position of the road edge is shifted so that, allowing for the curvature, the last filtered point would have been in the middle of the road.

ii) The width of the road in the next slice is found from the width of the bump in the last slice. We take three times the width of the bump plus a constant width. The justification for this is shown by figure 4.

```
new road edge
constant extra width
possible directions of track
```

Figure 4
iii) We have to predict a value of \( \tan \theta \) for the next slice. There is a predicted value for the current slice, and if we have found a filtered point, this can be used to obtain the tangent to the filtered track. The predicted \( \tan \theta \) and the \( \tan \theta \) from the filtered points are weighted together to give a corrected \( \tan \theta \) for the slice.

Thus, if \( \tan \theta_{\text{pred}} \) is the \( \tan \theta \) which was predicted for the present slice, then \( \tan \theta_{\text{cor}} \), the corrected \( \tan \theta \) is taken to be

\[
\tan \theta_{\text{cor}} = \tan \theta_{\text{pred}} + \frac{1}{a + bn} (\tan \theta_{\text{actual}} - \tan \theta_{\text{pred}})
\]

where \( \tan \theta_{\text{actual}} \) is obtained from the filtered points.

The weight used is of the form \( \frac{1}{a + bn} \) where \( a \) and \( b \) are constant and \( n \) is the number of filtered points found, so that increasing weight is given to the predicted tangent.

The difference \( \tan \theta_{\text{actual}} - \tan \theta_{\text{pred}} \) can also be considered as an error in \( d\tan \theta \) - the change in \( \tan \theta \) along a slice.

So we have

\[
d\tan \theta_{\text{cor}} = d\tan \theta_{\text{pred}} + \frac{1}{c + dn} (\tan \theta_{\text{actual}} - \tan \theta_{\text{pred}})
\]

in the same way as the corrected value for \( d\tan \theta \). This is called \( d\tan \theta'_{\text{cor}} \) because a further correction has to be made to allow for the change in \( \theta \) along the track.

If the curve is circular, \( d\tan \theta \) would vary as \( \frac{1}{R} \sec^2 \theta \), which is, allowing for the difference in \( x \) and \( y \) scales

\[
\frac{1}{R} (1 + \tan^2 \theta) \frac{3}{2}, \quad \text{where} \quad \gamma \text{ is the ratio of the scales.}
\]

Thus the variation in \( \theta \) is allowed for by making a further correction to give \( d\tan \theta_{\text{cor}} \) as

\[
d\tan \theta_{\text{cor}} = d\tan \theta'_{\text{cor}} \left( \frac{\gamma^2 + \tan^2 \theta (\theta + \delta \theta)}{\gamma^2 + \tan^2 \theta} \right)^{3/2}
\]

where \( \delta \theta \) is the change in \( \theta \) between slices.
Linearising this, we obtain

\[
\frac{\text{d} \tan \theta}{\text{cor}} = \frac{\text{d} \tan \theta}{\text{cor}} \left(1 + \frac{2 \tan \theta \text{d} \tan \theta}{\sigma^2 + \tan^2 \theta}\right)
\]

and the predicted value of \( \tan \theta \) for the next slice is taken as

\[
\tan \theta = \tan \theta_{\text{cor}} + \text{d} \tan \theta_{\text{cor}}.
\]

iv) Finally, we adjust the block width for the next slice. We allow for three sizes of block width - fine, medium and coarse. We always start tracks with the coarse block size and try to change this as rapidly as possible to the fine, via the medium. The width of the bump is used to decide whether to change to a finer status (i.e., block width) for the road. If the width of the bump is less than a set value (at present we use \( \leq 2 \) blocks), a change is made to the next finer status.

In the case where there is no good bump, the four gating parameters are changed as follows:

i) The centre of the road remains unaltered.

ii) The road is widened by an amount equal to the width of the last good bump obtained - that is, figure 4 is carried into the next slice.

iii) No correction can be made to \( \tan \theta \) but \( \text{d} \tan \theta \) can be corrected for curvature as previously and this \( \text{d} \tan \theta \) is added to \( \tan \theta \) to give the slope for the next slice.

iv) The number of filtered points found on the current status has been counted and if this is greater than a set value we say that the track is established on that status. When a track is not established and no good bump is found, a change to the next coarser status is made immediately.

The total width of the road is restricted to 20 blocks and when the road has widened to this limit, a change is also made to the next coarser status.

Finally, we have the case of more than one good bump in the histogram. If it happens on fine status, we reject the bumps and say no good bump is found. For coarse or medium status, we set up a new road for each of the bumps except the first. These new roads are identical.
to the original road, having the same parameters and any filtered points already found. These new roads are then treated in exactly the same way as normal roads.

4. Experience with the program

The program has been written in FORTRAN except for GATE and one or two small routines which are in FAP. The program was originally debugged using a small number of tracks that were not from real bubble chamber events. In the last few weeks further extensive tests have been carried out on all views of four pictures containing actual events. An off-line run with the HPD program is preceded by the Rutherford Laboratory equivalent of MIST which reads MILADY cards and produces an input tape for the HPD program, so that the present runs have been carried out with genuine rough digitizings. One event has been passed through the geometrical reconstruction program.

The four events each had five tracks, so, allowing for 3 views, the program was tested on 60 tracks altogether. The track-following mechanism has worked extremely well since all the final corrections to \( \tan \theta \) and \( \delta \tan \theta \) described above have been added. Also, all tracks have started successfully. If a track has a vertex at one end, and no point is found in the first slice (which is possible since the track may only start in, say, the last quarter of the slice), then a double slice is allowed, and this technique always works. There was an almost 100% record in successful track following, though exceptions must be made of two short tracks which had lengths varying between one slice and three slices on various views. It is clear that in this case, the digitizings will have to be stored and some special action taken.

The parameters used at present in the program are assigned as follows. The possible block widths are 16, 8 and 4 least counts for coarse, medium and fine respectively. The threshold in histogram scanning for bumps is \( \geq 4 \) points on coarse and medium, and \( \geq 3 \) points on fine. The minimum area for a good bump is 10 points, for each status, and the maximum width is 5 blocks on coarse and medium, and 4 blocks on fine.

Further improvements will be made in the future, particularly when advantage can be taken of parts of the program having to be rewritten for the ORION computer at the Rutherford Laboratory. It is hoped to start tracks at the beginning of a slice, and possibly to allow the number of scan-lines per slice to vary from track to track according to bubble density or other criteria. All points can be stored for short tracks and special action taken, and it may be useful to do
this for the first one or two slices anyway. Finally, in the case of failures, it is hoped to be able to rescan for the failed track and to take special action. It should be emphasised, however, that all these are refinements and the program at present operates successfully without them. Some actual roads which followed two tracks of a $V^0$ are shown in figure 5.

Acknowledgements

The authors would like to thank D. Lord, W.G. Moorhead, Miss B.J. Compston and A.G. Wilson for many useful discussions and assistance at various times. They would also like to thank Data Handling Division at CERN for their hospitality in the last few weeks and making available facilities for more extensive tests on the programs.
Figure captions

Figs. 1, 2, 3 and 4 are inserted in the text.

Fig. 5 An Example of the narrowing down of the roads is shown for the secondaries of a $\pi^0$ particle.
DISCUSSION

SCHIFF: Could you give some details of the gating procedure for several tracks at a time?

BURREN: For gating purposes, we list the roads in order of increasing y, that is their near road edge. The order is determined at the beginning of a slice and we only re-order every slice. In some slices where two roads cross, one shields the other. This can be minimized by searching first for the far road edge and then working backwards, and then starting from where you have got to forwards again, so that if two roads cross you only miss starting points in one of them when the other has completely crossed over. Otherwise, we just do one road after the other. When one road is completed, we start from where we have got to for the next road, since we know they are correctly ordered, so a digitising is only looked at once unless it is in more than one road.

SCHIFF: What is the computer time to treat one view?

BURREN: We have not processed enough events to answer this. The program will treat a maximum of between four and five roads at a time on the 709.

HOUGH: How many cycles per scan line do you have?

BURREN: 600

HOUGH: There are 1000 scan lines on the 7090, so you would be better off by a ratio of 3 to 2 and you could do 8 or 9 roads.

BURREN: These times just show that the program is not ridiculous from the real time point of view. The times should not be very different from the Berkeley figures, since much of it is spent doing similar operations.

HOWIE: If two roads are introduced for the same track, what criteria are used to finally select the correct road?

BURREN: If a track has several roads, they are treated independently. At the end we have a routine called TIDY which checks that everything goes through the rough digitisations. We also look at vertices
so that if one of two roads goes beyond the vertex we would reject that.

HOWIE: What is the maximum road width?

BURREN: 20 blocks of 16 counts, which is just over 500μ.
OFF LINE TREATMENT OF HPD DIGITIZINGS

M. BLOCH and M. SCHIFF
Collège de France, Paris

(presented by M. Schiff)

Introduction

This morning, Dr. Kowarski said that we were here to talk about our unfortunate experiences with HPD programs. We don't have any unfortunate experience to communicate; in fact we don't have any experience at all. All we can contribute so far are ideas.

The work I am going to report on has been done mainly by M. Bloch and myself. Mr. Leblond, who recently joined the HPD group, has started to work on problems of the small computer.

The main difference between the system we plan to use and other HPD systems is that we shall treat the HPD digitizings off-line; the HPD will be operated by a small computer which, essentially, will only serve to control the HPD and to transfer the digitizings onto magnetic tape. Another important difference is that we plan to treat the tracks as a whole, instead of slice by slice.

My talk will be divided into two parts:

I. The HPD system.

II. The processing of HPD digitizings.

Before starting, I want to say that we have received considerable help from W.G. Moorhead and from J. Burren. I hope that they will go on helping us. I also hope that we will some day be in a position to return the favour.
I. Outline of our plans concerning an HPD system

We have already described our system in a report, which some of you have received. So, I will be rather brief.\footnote{1}

Figure 1 shows a sketch of the system. One of our first concerns was to find a name for it. We chose F.A.D.A.. These initials can stand for French Automatic Data Analyser; or Fully Automatic Data Analyser. In French, Fada is the slang for crazy.

On the left part of figure 1, is the hardware, on the right part, is the software. From the point of view of the programmer, there are three phases in the system:

a) Use of the premeasurement.

b) The measurement.

c) The off-line treatment.

a) Use of the premeasurement

Our program will serve more or less the same purpose as MIST, except for one important addition; we will use the roads to prepare a rough digital mask. This mask will be used during the next phase to reduce the number of HPD digitizings.

b) The measurement

During the measurement, the CDC 160 A will be used on-line, and exclusively by the HPD. It will serve four functions:

- Control of the HPD.
- Input of the digitizings through the buffer channel.
- Use of a rough digital mask to eliminate the digitizings that are far from the roads and far from the fiducials.
- Write on magnetic tape the acceptable digitizings (through the normal, unbuffered channel).

Let me say a few words about the preagating. First let me say that it is not essential to our system; it is only convenient. If it is done on the small computer it saves some time on the large computer by reducing the time needed to read the tapes. It also saves memory space in the CDC 3600. We expect to gain a factor 2 for sparse pictures and a factor 3 or 4 for crowded pictures.
We think that we can handle the following rates of digitizings coming from the HFD:

without pregating: 603 tapes: \( \sim 8,000 \) / second
606 tapes: \( \sim 15,000 \) “

with pregating: 603 tapes: \( \sim 3,000 \) / second
606 tapes: \( \sim 4,000 \) “

For the purpose of comparison: the rate produced by a disk turning at 1000 rpm, with 20 W's per scan line is 2800 digitizings per second.

If the rate of digitizings is too high, so that we cannot do the pregating with the small computer, we shall do the pregating with the large computer. This way, we shall still save memory space.

\( \text{c) The off-line treatment} \)

The third phase is the off-line treatment of the tapes that have been produced by the small computer. These tapes will be read in and treated by a CDC 3600.

The two main advantages of the off-line treatment are:

- No problems of simultaneity with other programs.
- Plenty of memory space.

We will make use of the available memory (64 K words) to treat the tracks one after the other, each one as a whole. If nothing else, this has the advantage of simplicity.

Tape handling will cause no trouble. Our estimate of the amount of tapes produced is as follows:

At 1000 rpm, one tape / hour (with pregating; one tape every 2 to 4 hours)
At 3000 rpm, two tapes / hour

At first we will use manual handling. We will then use switches manufactured by Control Data, that allow one to connect tapes either to the small or to the large computer.

\( \text{II. The off-line treatment} \)

We shall take advantage of being off-line to treat tracks as a whole. If you take a slice of track within a road, the background may be as important as the track; but, if you consider the track as a
whole, it will usually predominate over the background. So we will proceed as follows: first we will take sample points spreading over the whole track (or over a large portion thereof), which will serve to narrow down the road; then we will treat the content of this narrow road. In both phases, we make use of the fact that tracks usually describe a rather smooth curve: the whole track, or at least elements of track, can be approximated by a second degree curve.

1) ENTER: The digitizings corresponding to one picture are read into the large computer (at present an IBM 7090, but later a CDC 3600) and organized as follows:

\[
\begin{align*}
\text{XPHOTO} (L) &= \text{abcissa of scan line } L \text{ (from the Ferranti)} \\
\text{WPHOTO} (K) &= \text{HPD digitizings (W's) packed in a list} \\
\text{KPHOTO} (L) &= \text{index of last W of line } L
\end{align*}
\]

2) GATE: At present we treat the tracks one after the other, first with GATE then with something like FILTER. For economic reasons, we may later be forced to gate several tracks at a time. But we shall definitely keep GATE and FILTER separate, treating a track only after GATE has been applied to the whole track.

The content of a road, as given by GATE is in the following form:

\[
\begin{align*}
\text{WROAD} (KR) &= \text{W's of the road packed in a list} \\
\text{KROAD} (L) &= \text{Index of last W of line } L \text{ (within the road)}
\end{align*}
\]

For the whole picture, the abcissa are given by:

\[
\text{XPHOTO} (L) = \text{abcissa of line } L
\]

By keeping the line structure, we make it easier to take sample points of the roads, at regular-intervals.

3) SAMPLE: This is used to divide the track into elements. Each element contains a fixed number of more or less equidistant sample points, say 10 points per element. We obtain:

1 element of rank 1, spanning the whole track
2 elements of rank 2, each spanning 1/2 of the track
4 elements of rank 3, each spanning 1/4 of the track
and so on....
4) FILTER: The purpose of FILTER is to eliminate the background by narrowing down the roads. The main work of FILTER is done by 3 subroutines:

- FITPAR which makes a least squares fit to a parabola
- CHOOSE which eliminates a fixed number of "bad" points after a fit, say 4 points out of 10.
- FINEGATE, which is used to narrow down the road.

The fitting allows us to locate the track. This will often be possible on the whole track. Otherwise the track will be located on one half, one fourth,... of the road.

Once the track has been found on one element, FINEGATE is used to extrapolate it into the other elements, from one element into the next.

This has some analogy with the procedure developed at Harwell, but there are two improvements. First, we can use large elements because we have quadratic expressions.

Second, we don't have to start at one end, and we can extrapolate in both directions. So, in principle, for the method to work, it is sufficient if one single element of the original road is "good". Good means here that the ratio of track points/background points is higher than say 1.5.

The only troublesome treatment will probably be that of the beam track. If we use wide roads (which we will probably do), we may have to develop a special procedure for the beam track, starting from the vertex.

5) Summary points
Once we have fitted the smallest elements, we are ready to get the results, which are IEP-like points. Each point is obtained by the intersection of a straight line $X = \text{constant}$, with the parabola fitting a small element.

One gets a summary point, and also an idea of its accuracy. The weight of the point will depend on the fit of the element used and also on the number of points used to build the element.

6) State of the program

Let me first emphasize that what I have just described is only a prototype program, designed to quickly test our ideas about tracks. For the time being, simplicity and speed of coding are our main concern. Therefore we use FORTRAN, with as few interloops as possible between the various parts of the program.

The program has been coded up to FILTER. The subroutines have been assembled and the main program is being assembled. The part that has been coded contains about 500 FORTRAN statements. By the time we will have included the procedure to get summary points, an economical GATE, a procedure to find the vertex, one for the stopping points etc... the program for the off-line treatment will probably be 3 times that size. So, we are far from home, to say nothing of the coding on the small computer.

Conclusion

I have given you a brief outline of our future HPD system. The small computer will act mainly to control the HPD and to write the data onto magnetic tape. This data will be treated by a large computer, off-line. The advantages of the off-line treatment are: simplicity of programming and plenty of memory space.

I then described our prototype program for the treatment of tracks. Taking advantage of the memory space available, we treat tracks as a whole, or as made of large elements. Quadratic fits are used to narrow down the road. We will extrapolate from one element of the track into the next. Simplicity of coding has been our main concern up to now.

Reference

Figure caption

Fig. 1 The three phases of the HPD system FADA.
a) Premeasurement:

MILADY → CONTROL

b) Measurement:

H.P.D. from Sogenique → C.D.C. 160 A

c) Off-line treatment:

C.D.C. 3600

CONTROL ENTER PREGATE TRANSFER

ENTER GATE SAMPLE, FILTER, RESULT VERTEX EXIT

Figure 1
DISCUSSION

HOUGH: Do you have any results for the squeezing down process? Have you been able to treat any tracks as a whole?

SCHIFF: I have gone through one picture numerically, I have not had the courage to go through FITPAR since it requires ten decimal places. I have gone up to SIMPLE and I feel that it will work, but I cannot say anything about the quality of the elimination of background. The difficulties are numerical ones, for example, due to the short IBM word which corresponds to 8 or 9 decimal digits. This causes trouble if the coordinate system is not relative to the track, and this is what we are doing. I do not see how this procedure can fail where others don't, but it may take more time.

HOUGH: You have to be careful with tracks that are not very good circles or parabolas.

SCHIFF: That is why we have included a provision for dividing tracks into elements.

KOWARSKI: How small is your small computer?

SCHIFF: It is a CDC 160A. The minimum system includes a typewriter, 2 magnetic tape decks and one tape control and this costs about as much to buy as for 1/4 years rental of one IBM channel with DDX. We actually have a bigger system because we shall also use it for input-output. The IEP people are using it.

KOWARSKI: How big is the memory and what is the price?

SCHIFF: 8,000 words of 12 bits - two banks of 4000 words each. The basic price is $ 90,000. The tape units are extra and form the expensive part.

EDMONDS: I would like to ask about the decision to buy the CDC 160A. Many of the small computers on the market have word lengths of 20 bits, which fits rather better to the requirements of this problem. The cost is not much greater than that of the CDC machine. Time could be saved either in transferring to magnetic tape or in processing on the big computer because one could do so much more in the small one.
SCHIFF: The main purpose for us is to put data onto magnetic tapes. We looked for the best possible tapes and decided on CDC. Other people also want to use the input-output facilities for card to tape and tape to card conversion. We also considered the PDF 1 and ASI 210, but neither were likely to be serviced in Europe. Pregating would be a nice addition for us on the small computer, and it has actually been coded in CDC machine language, but if it happens to fail, the system will still work.

LEBOY: It also gives IBM compatible tapes cheaply.

CALKIN: There is no computer of larger word length which has the input-output capabilities of this machine. Also, they supply their own tapes rather than someone else's.

LEBOY: It also has a fully buffered channel, so you can get output on magnetic tape while calculating simultaneously.

BLOCH: One of the most important factors in our decision was the hope of getting a CDC 3600 and if we do, we need a 160A anyway, so we cannot lose by this decision.

SCHIFF: 100 - 150 without pregating. If you take into account a factor of 2 or 3 for pregating, plus very high density tapes, you get a few hundred events.
PERFORMANCE OF THE FIDUCIAL-FINDING SUBROUTINE H-10*

P.V.C. HOUGH
Brookhaven National Laboratory, Upton L.I.

1. Program authors

N.W. Webre has been largely responsible for the work on HAZE at Brookhaven, with extensive contributions by L. Berlent, W.M. Thompson, S. Kaplan, and D. Burd (Columbia). The subroutine H-10 itself was written by D. Hall and W.M. Thompson at Berkeley, and the study of the performance of H-10 on fiducials from the BNL 20" chamber has been carried out by W.M. Thompson at Brookhaven.

2. Introduction

The hardware used for the studies reported here is the Mark I (35 mm) FSD measuring 20" chamber film on-line to a 7094. A general view of the Mark I is shown in figure 1.

In figure 2 is reproduced part of a 20" chamber photograph showing the production and decay of a pair of cascade particles. The X-fiducials appearing are 2 of 3 which are scribed on the inside front glass of the chamber. A square just containing each interior fiducial is on film \(\sim 2.1\) mm on a side. The two road fiducials are about 30% smaller and appear on film just outside the illuminated region on the down-beam side.

3. Rules for preparation of fiducials

A relatively rich history of needless difficulty in computer recognition and measurement of fiducials has led us to formulate these rules for preparation of fiducial marks at the chamber:

a) The fiducial X's should fit in a square 2-3 mm on a side. For the usual 30-50\(\mu\) line spacing this rule will insure that the fiducials are found reliably, in the presence of what is sometimes formidable track and electron–spiral background, and

*) Work performed under the auspices of the United States Atomic Energy Commission.
despite occasional bad picture quality.

b) There should be more than 3 internal fiducials, probably 4 or 5. The point here is to provide enough redundancy that a pair of error-free reconstruction fiducials can be selected on the basis of internal evidence alone in the presence of slight film rotation and 1-2% real variation in magnification. The latter variation seems to arise through defective operation of the vacuum hold-down in the cameras at the chamber.

c) All fiducials, read and internal, should be fixed relative to each other, and of fixed arm angles, for all time. The change of fiducial positions from roll to roll and from bubble chamber run to bubble chamber run has been an annoying deterrent to extending high-rate measurement runs from a few dozen events to a few hundred. The provision of the display diagnostic described below and, most recently automatic adjustment of fiducial box position (under operator request) have largely overcome this difficulty.

d) Fiducial arms should be of width about equal to that of a dense track in the chamber. Thicker arms will digitize with a systematically too small flying spot coordinate (using the center of area track-center circuit) or with an excessive scatter (using the delay-difference track-center circuit).

4. Precision required of the fiducial finding subroutine

Consider two of the 20" chamber reconstruction fiducials separated by \( \sim 15,000 \mu \). If in both views used for reconstructing a particular track the downstream fiducial were measured with a transverse error of \( (+ \text{ and } -) 15 \mu \), a spurious projected angle of \( 1/1000 \text{ rad} = 1/20^\circ \) is introduced, corresponding for the 20" chamber stereo angle to a spurious dip angle of \( \sim 1/5^\circ \). This spurious dip angle is comparable to the error usually to be assigned a dip angle under the assumption of no fiducial error, so an occurrence of this magnitude of fiducial error is about the maximum tolerable, and less is desirable.

5. Mode of operation of H-10

a) H-10 recognizes fiducials from the subset of points contained in a rectangular box whose width \( (X) \) and height \( (\text{flying-spot coordinate } W) \) are program parameters typically set to 3 mm and 2 mm respectively.
Given the view number from the scan tape, H-10 looks up in a table the road-fiducial (1) box position, given "absolutely", i.e., assuming that the film is positioned sufficiently accurately relative to the clock track and hence road fiducial. Once the road fiducial is found, a table look-up of $\Delta X'$s and $\Delta W'$s gives the box positions for fiducials 2,3,4. No subsequent box position depends on finding any fiducial except the road fiducial.

b) H-10 next gates the fiducial box.

c) Next H-10 projects each point onto the W axis, once for each arm-angle, using angles given in a table and appropriate to the particular view and fiducial number.

d) H-10 then histograms according to the value of the W-intercept using bin widths which at present are $32 \mu$ and which will probably be made $16 \mu$.

e) The two histograms are then searched and all bins containing more than 8 points are tallied to select "pulses" of histogram area more than 16 points. Typically 2 bins contribute, often 1, occasionally 3. Typical pulse areas are 60 points. In recent detailed examination of $\sim 100$ histograms no cases were found with more than one pulse per histogram.

f) A least squares straight-line fit to the points in the pulse for each projection is made with the constraint of the slope used in the projection itself. The intersection of the two lines is H-10's output of fiducial position.

6. Diagnostics

a) Our most valuable diagnostic is an on-line CRT display of the contents of each fiducial box which appears for a fraction of a second as the measurement scan passes each fiducial. An example is shown in Fig. 3. The superposed vertical and horizontal reticle show the fiducial coordinate as found by H-10, with a spacing of $\sim 100 \mu$ between adjacent dots so that H-10 errors of $\sim 20 \mu$ are evident to the observer. Bad box positioning is also immediately evident, and so are any defects in
digitizing by the FSD. This "window" on overall machine performance we find as valuable in monitoring fiducial measurement as our road window is for monitoring track measurement. However, we use it normally only at the beginning of a run and then switch it off.

b) For more detailed investigation of H-10 operation we occasionally dump the H-10 histograms on tape in real time and later list for analysis.

c) Our final criterion for satisfactory performance of H-10 is a check of fiducial separations. If a good grouping of separations is obtained (with standard error $\sim 5\mu$) the fiducial performance is accepted. In the presence of real variations of up to $200\mu$ we have compared with separations measured on the NUL measuring projectors. A rough check in FOG 135 detects gross errors in any single fiducial and automatically selects the remaining two for space reconstruction. A small FORTRAN program is being constructed to read HAZE output tapes and provide summaries of H-10 performance for large numbers of cases.

7. Remaining problems

Although H-10 by now rarely fails to find a fiducial, and also rarely makes a gross error, it still makes a sprinkling of errors in the range 10–90\mu. These are difficult to track down in the presence of real variations in fiducial separation of comparable magnitude and we have not yet found the reason for the errors.

The automatic adjustment of fiducial box position is not in our current production (AM) assembly of HAZE, so until the AM assembly is ready we are faced with somewhat halting re-adjustment of box constants in going e.g. from $\nu^-$ film exposed in September 1962 to $K^-$ film run in April 1963, and minor adjustment even from roll to roll.
Figure captions

Fig. 1  A general view of the Brookhaven Mark I (35 mr) FSD.

Fig. 2  Part of a BNL 20" chamber photograph showing the production and decay of a cascade and anticascade particles. The \( x \)-fiducials appearing are 2 of 3 which are scribed on the inside front glass of the chamber.

Fig. 3  A CRT display of the contents of a fiducial box. The superposed vertical and horizontal reticle show the fiducial coordinates as found by H-10.
DISCUSSION

MOORHEAD: What is the average number of digitising on a fiducial arm in the Berkeley chamber?

HILL: 20 points per arm.

MOORHEAD: Have you tried any other technique for finding fiducials?

HOUGH: Not on these fiducials. Last December we tried two routines of our own on the smaller fiducials with the old film format of the 20" chamber. We were much less successful than with H-10, so we have no inclination to go back.

MOORHEAD: What number of digitising did those fiducials give?

HOUGH: About 30 per arm.

MOORHEAD: That you had difficulty even than with the technique of following I think is relevant to pattern recognition.

HOUGH: Yes. I think it is instructive how powerful the histogram technique is for initiation.

MOORHEAD: This relies on your having the right angle.

HOUGH: Here you do know the angles. It is a fortunate case so that you are able to use the most powerful line element recognition technique available.

MOORHEAD: I have one comment, really on behalf of Blackall, who is doing this, on the technique of looking one point ahead. We had trouble because there may only be three digitisings on the first half of the arm and then the look ahead did not work through the difficult region in the centre of the fiducial because you had to extrapolate on too little information. The technique ought to work with 30 digitisings on each arm.

HOUGH: An integration over the entire length of the arm is much simpler and more powerful.
MOORHEAD: It is faster machine-wise to look ahead by one. I understand why this does not work for our fiducials, but with 30 points surely something similar to the Rabinowitz look-ahead must work.

HOUGH: There are two reasons why it may not. Firstly there are many crossing tracks, and secondly there are occasional gaps in the fiducial arms. In a real life situation, a point by point look ahead is far weaker than an integration over an extended length. This is the reason why I believe in initiation, if you can afford it, either by being out of real time or only handling special cases that way, or through special purpose hardware. H-10 type integration is probably the recognition technique for initiation.

EDMONDS: How long does the H-10 routine take to locate a fiducial?

HALL: The routine was written in about two weeks, so no attempt was made to make it particularly fast.

HOUGH: When we have buffer overflow, it is during fiducial search. That is, when you are gating, filtering and looking for fiducials, so the time is not negligible compared to the time to gate and filter.

HALL: Moorhead has already indicated an improvement in coding which should reduce the time.

SCHIFF: In your method of histogramming with a table look-up, do you look up in X and W for a given S. Are you avoiding formulae?

HOUGH: The table look-up is only to find the tangent of the angle to make the projection. Then you look how far downstream from the edge of the block the point is and subtract an amount which is proportional to that.

BLOCH: How do you determine the gating regions for the different fiducials?

HOUGH: For the first fiducial, you have, for a given view, an absolute stage coordinate and a flying spot coordinate, which means the film must actually be in the film gate at a known position. Then you simply search for the road fiducial. Once this is found, the coordinate system is established on the film to within small rotations which we don't take account of here. Then the remaining fiducial gating regions are put down relative to the road fiducial. Since they are only relative, failure to find another fiducial does not inhibit looking for later ones.
BLOCH: For an HPD controlled by a small computer which cannot do much calculation, would it be easier if the read fiducial had a horizontal arm? You want to position a 2\(\times\) 2\(\mathrm{mm}\) box, and you may not need a 20\(\mu\)m accuracy on the read fiducial, so possibly just a horizontal line without a crossing arm would give you an accuracy of about one scan line separation.

HOUGH: That is a little fragile. There are a number of other marks which might simulate that. However, if you had two of these, with a known separation, it might work well.
CONTROLLED FILTER AND SMOOTHING

D. BURD
Columbia University, Irvington-on-Hudson

N. WEBRE
Brookhaven National Laboratory, Upton L.I.

(presented by D. Burd)

CONTROLLED FILTER was designed to meet two basic problems which we encountered in the HAZE system. The first was to provide a bypass around FILTER II, which was producing neither the low rejection rate nor the high quality of data needed for production, without going so far as to write a third version of FILTER, which we estimated would take several months. The second problem was that no convenient means existed for studying the difficulties that FILTER was having.

The need for an editing feature such as the present smoothing program became apparent as soon as the first results were available from CONTROLLED FILTER. Small undetected errors, such as the usual one or two bad master points, could be corrected by fitting a smooth curve to the data and rejecting those points whose deviations were greater than some maximum amount. Large errors in filtering could be detected by the fact that the master points did not actually fit a smooth curve, or that a large section of the track had no master points.

1. CONTROLLED FILTER display

CONTROLLED FILTER displays the contents of each of the rejected roads on the CRT. This, of course, means that all the gated points for one event must be stored. The format of the display is as follows (see fig. 1): event identification is at the bottom of the screen; the road stretches from left to right across the screen, its vertical extent being defined by the mid-road dots. The digitized points are plotted as dots and the master points are superimposed as X's. As filter can process two tracks, the master points of the auxiliary track (if one exists) are superimposed as + 's. The alphabetic codes in the upper right hand section have the meanings:

E - every point node
R - rejected
A - abnormal node
In the upper left are the three rough digizer points inside of a chamber outline.

2. CONTROLLED FILTER Method

Let us suppose that a rejected road has been displayed on the CRT screen. At this point the operator may decide that the measurements are perfectly good, or at least that the imperfections are slight and probably cannot be eliminated. In this case he simply can override the rejection from the console and continue on to the next track. If there seems to be an error in the displayed road, he can proceed as follows: He chooses the track which he believes to be the correct one (this is usually obvious, but in any case can be decided from a sketch made at the scan table) and enters its CRT division at the beginning, middle and end of the road into the keys. Since the road covers 8 large divisions vertically, these can be entered as the numbers 0-7 and the divisions on the external display at Brookhaven are so labeled. The operator then pushes a button which interrupts the CRT display and returns control to the program. The information in the keys is now used to make a parabolic fit to the reduced track and the filtering process is repeated. This time, however, the look ahead region is determined only by the parabola. A byte will be filtered if it has one and only one pulse in the parabolic region. The rfiltered track is then displayed once more and the operator is free to try again if he is still unsatisfied.

Figures 2 and 3 illustrate the use of CONTROLLED FILTER in a case where FILTER has strayed onto a crossing track and left the road.

Figures 4 and 5 illustrate a case where FILTER has initiated the wrong track and CONTROLLED FILTER sets it right.

3. Smoothing method

Bad points are rejected on the basis of a series of parabolic least squares fits to the master points. Each fit results in a single point rejection if the maximum deviation exceeds a calculated tolerance. The tolerance here depends on the turning angle of the measured portion of the track. If three or fewer point rejections are required, the track passes the smoothness test and the longitudinal distribution is checked. Here the master points are required to be separated by not more than a fixed fraction of the total road length. Likewise the end measurements are required to be reasonably close to the road ends. If either of these tests is not met, the entire track is rejected, hence displayed.

Figures 6 and 7 show a filtered road before and after smoothing.
4. **Sequencing of smoothing and CONTROLLED FILTER**

Immediately after the final call to GATING and FILTER II, smoothing is applied to all tracks which are not rejected by FILTER II. At this point, each track has skipped smoothing, been smoothed, or been rejected during the smoothing process. Control is then returned to DISPATCHER, which immediately calls CONTROLLED FILTER. Each of the rejected tracks is displayed. If a track is manually refiltered, the new master points are subjected to smoothing. This means that output from any point in the system will have been smoothed.

5. **Production and rejection rates**

The statistics from a batch of 100 events will give some idea of the present performance of CONTROLLED FILTER.

<table>
<thead>
<tr>
<th># of tracks</th>
<th># of displays</th>
<th># of rejections</th>
</tr>
</thead>
<tbody>
<tr>
<td>935</td>
<td>242</td>
<td>84</td>
</tr>
</tbody>
</table>

This indicates that we had to look at about one track in four, which is misleadingly high. 51 of the tracks displayed were missing from the road completely. This is due to a bug in the system and is undoubtedly curable. 17 more were displayed because of an error in the program causing all tracks to be rejected when both secondaries were in the abnormal mode. If we subtract these out, we get the following set of numbers:

<table>
<thead>
<tr>
<th># of tracks</th>
<th># of displays</th>
<th># of rejections</th>
</tr>
</thead>
<tbody>
<tr>
<td>935</td>
<td>174</td>
<td>16</td>
</tr>
</tbody>
</table>

This puts the number of displays at the 1 in 5 or 6 range. It is clear already that the tolerances we are using now in smoothing are much too conservative. Many of the displayed tracks seem perfectly good. Loosening these tolerances should significantly reduce the number of displays without affecting the quality of the data. We would like to, and think we soon can, reach the 1 in 10 region.

Our rate with this sample was about 50 events per hour or about 400 events/8 hour shift. If we reach the 1 in 10 region on displays and use 400 ft. rolls of film instead of the 100 ft. rolls used to get the above results, we believe we can reach 600 events/shift with no difficulty. Any substantial increase in this rate will probably require a basic revision of FILTER II.
Figure captions

Fig. 1 An example of the CONTROLLED FILTER display. The dots represent gated digitisations, the X's are the computed master points. In the upper left of the picture is shown the outline of the chamber and the positions of the three rough digitisations for the track which is displayed.

Figs. 2 These show a typical example where FILTER follows a crossing track (2) and is corrected by the use of CONTROLLED FILTER (3).

Figs. 4 Here FILTER fails due to starting on the wrong track (4) and is corrected (5).

Figs. 6 These show the effect of smoothing (7) on a track where some bad points are initially present (6).
DISCUSSION

TYCKO: Are most of the rejections due to the smoothing criteria?

BURD: Nearly everything.

TYCKO: If you increased the limit to $6\mu$, would they all go away?

BURD: I hope so.

TYCKO: If it were $6\mu$, you would then be doing like a Frankenstein and you could compare with that.

MACLEOD: If one track in nine has to be looked at, how much of the overall time do you spend thinking about this one track?

HROUGH: The time is raised by a factor of 2.

MACLEOD: You are processing 600 events per shift, and you are spending 4 hours of this shift thinking about 10% of the tracks. Why is this faster than taking a dump for the 10% and looking at them off-line later?

BURD: The main reason is to see what FILTER is doing.

MACLEOD: Will it be part of the production system?

BURD: If it degenerates to a small percentage, it will be taken away.

CALKIN: Does the 600 events per shift include processing through FOG-CLOUDY-FAIR?

BURD: No, just HAZE, and then out onto tape.
ADAPTATION OF FOG AND CLOUDY TO HAZE

D. HALL
Lawrence Radiation Laboratory, Berkeley

The FSD versions of the FOG and CLOUDY programs will remain essentially the same as far as the geometry and kinematics are concerned. The only differences will be the administrative changes associated with multiprogramming, together with the additional programming required for making use of ionization information.

Multiprogramming

It should be emphasized from the beginning that all of these remarks are highly tentative until work actually begins. In fact, the problem of multiprogramming will almost certainly be solved differently at each of the three laboratories. This is because of the following fundamental differences.

In the first place the Berkeley 7094 has a 65 K memory. This will tend to ease the space problem for CLOUDY and allow a less radical solution to the problem initially.

In the second place the Berkeley 7094 is equipped with a disc file. This will provide a very convenient scratch storage as we shall see later.

Finally, it is highly unlikely that HAZE with controlled FILTER will have any room left over for a secondary program inside 32 K.

For these reasons, this discussion will be limited to the Berkeley solution of the multiprogramming problem, which will most likely be attained in two phases.

The first phase will consist of running HAZE in the lower 32 K and FOG in the upper 32 K. This configuration should be fairly easy to attain and will yield invaluable experience for the more difficult second phase. In fact, the only programming changes needed to achieve the first phase will be to provide for loading programs into the upper memory and allowing the dispatcher portion of HAZE to call a program in the second memory.
This configuration might be represented as shown in figure 1. The nine scratch tapes are necessary because of the Alvarez film format (Appendix A). When this phase is debugged, work will probably begin on converting the nine scratch tapes to disc storage.

The second phase will consist of running HAZE and FOG in the lower 32 K with the two halves of CLOUDY being multiplexed in and out of the upper 32 K.

To do this, approximately 8,000 words will have to be trimmed from the sum of the current HAZE and FOG programs. However, this should not be too difficult since no particular effort has been made in the current HAZE program to make it small.

It may also be worthwhile to assemble a short version of CLOUDY for events with fewer than $n$ tracks (where $n$ is yet to be determined), and which will fit into one 32 K module. This would allow nearly the same version of the DISPATCHER to be used at the beginning of the second phase as was used in the first phase, and should significantly reduce the debugging phase.

For complete generality however, the DISPATCHER will have to be extended to include calling each of the two halves of CLOUDY together with the appropriate adjustable constants.

This configuration can be represented as shown in figure 2.

**Ionization**

The current FILTER program now generates the following output:

1. Number of hits
2. Number of misses
3. Number of cells

Getting this output onto the CLOUDY library should be quite easy. In fact, all that is required is a slightly longer buffer in HAZE and FOG to hold the information; the addition of a trivial calculation in the FOG program which gives the bubble density in space; and a slight change to the formats in the CLOUDY program 208.

The CLOUDY program 208 operates as follows:

Each mathematically possible mass permutation is generated in the memory. It is then tested for the following appropriate acceptance requirements:
1. Nucleon conservation
2. Lepton conservation
3. Strangeness conservation
4. Strangeness non-conservation

If the mass permutation passes these tests, it is output onto the CLOUDY library.

R.C. Strand of BNL has written a subroutine for program 208 which constrains the three overdetermined parameters: momentum, mass, and ionization for each charged track. Once the ionization information is available to program 208, the subroutine with a $\chi^2$ cutoff could be used as a further acceptance requirement on a mass permutation. If the test failed, a short form of output could be generated giving the value of $\chi^2$ and the mass assignment.

It has been further suggested that after kinematical constraints have been applied that there still may be remaining ambiguities. Since the momentum is much better known after kinematical constraints, one possibility would be to re-apply the ionization constraint using the fitted moments in the hope that this procedure would remove any remaining ambiguities. However, this last procedure has been by no means finally adopted.

Appendix A

Alvarez film format:

All three views of an event are contained on the same roll of film as shown in figure 3. Letters indicate the frame number and numbers indicate the corresponding view.

Because of the time involved in advancing the film, it is mandatory that the views be measured in the order that they appear on the film. However, for reconstruction purposes it is necessary to have all three views of an event together. Thus a sort is indicated, whereby all of the view 1's appear on one tape, the views 2's on another tape and the view 3's are on a third tape.

Furthermore, if one is to be running FOG concurrently, it is necessary to be buffered ahead by 15 events. Thus, three sets of three tapes each are required. One set of three is being used by FOG for reconstruction, while the other two sets of three are being generated. When FOG is finished, it releases the current set of three for more measurement and begins processing the next set of three.
Figure captions

Fig. 1 The node of operation for Phase 1 with 64 K memory but only time sharing with FOG.

Fig. 2 Here Phase 2 is shown where both FOG and CLOUDY are being time shared with HAZE.

Fig. 3 The alvarez film format. The letters indicate the frame and the numbers 1, 2 and 3 represent the corresponding views.
DISCUSSION

HAGOPIAN: What is CLOUDY A and CLOUDY B?

HALL: Because of memory space they have been divided, and CLOUDY A is the abstraction portion of the program; essentially all it does is to take a FORTRAN library and generate a CLOUDY library with all of the mass permutations on it; CLOUDY B contains all of the fitting, and what we call derived quantities. We calculate Q values, centre of mass transformations etc. in CLOUDY and not in PAIR.

MOORHEAD: Referring to the fits, I would have thought you could include the ionization into the fits to begin with.

HALL: You could do that, but the idea of this is to reduce the number of hypotheses that are tested.

CONNOLLY: The point is that nobody wants to combine the ionization and kinematics in a single fit, because of the question of ionization and momentum. How do you handle the errors and $\chi^2$? They are related quantities. If you do an ionization and measured curvature fit first, you will get rid of the most ridiculous mass permutations, and not prejudice the kinematics fit. The kinematics fit then is unbiased, but now after the kinematics fit you will have the improved momentum estimates, and with that finer information you may be lucky and get only one fit left. It is essentially a 3 step process. I am not so sure that the final ionization-momentum fit is going to really help so much. The real work is done in the first fitting process.

MOORHEAD: What is the advantage of using discs since you are not having random access; you can probably get magnetic tapes which are faster to read in?

HALL: They are expensive whereas we already have a disc file. We are going to use discs for all of this. It will originally be done on tape. HAZE goes onto tape at the present time and will go onto tape under Phase 1. But as soon as we can, we will get disc routines and change it. We have got 9 tape units tied up otherwise.

MOORHEAD: I think the no-disc arrangement could be done with less than 9 tape units.

HALL: Not with our film format. I agree it can be done at CERN and at Brookhaven without these.
SCHIFF: What line separation are you using for the ionization measurements in production?

HALL: The line separation we are using is 60μ. We have done essentially no experiments to decide where the optimum is.

ROUGH: About 3 months ago we ran about 100 tracks all the way through the ionization routine, to obtain the mean bubble density from the "hits", "misses", and total number of gaps — those are the parameters the program uses at the HAZE level. There is a break point in the quality of the ionization data, probably at the line spacing for which you completely cover the picture. This, for us, is a 40μ line spacing. This equals the sum of the bubble diameter and spot diameter so you detect all the bubbles in the picture. For greater separations, the quality of the ionization goes down, probably rather slowly. How best to use the information, we are not quite sure. The most straightforward statistical answer may not be the one that is most independent of chamber operating parameters, that is the effect of the location of the tracks in different parts of the chamber. Strand is going to take this up again now that we have the prospect of a large number of well filtered tracks.

CALKIN: Have you had similar difficulties with HAZE or FILTER at Berkeley?

HALL: Yes, but with a lower frequency. We have a figure of 83% success in the best case and I think it is largely due to the film — and the fact that we have more programmers.

CALKIN: If HAZE output goes onto tape, this is then no longer a real time business. What is the advantage of doing FOG and CLOUDY right away? Is it not better to just do HAZE with a small computer as they propose here in Paris and then do the rest on a big machine later?

ROUGH: There is a big difference between going onto tape with pre-filtered coordinates as Schiff proposes, with some modest reduction in the volume of data, and going onto tape with the final HAZE output. That part of HAZE is the main measuring machine problem. It seems to me we are using the full facilities of a 7094 to do gating and filtering.

HALL: You have a faster film advance than we do. The 3 second film advance is just about right to do FOG and CLOUDY if HAZE takes all the time during the sweep.

BLOCH: We will use the small computer only to control the HPD and to transfer the digitizations onto magnetic tape. It may be possible to do PRIMATE as well but that is all. Then, off-line, we will do HAZE and the rest with a large machine.
HOWIE: Returning to ionization measurement, when you take your hits and misses in HAZE, do you keep the bit patterns?

HALL: No, we don't keep them. The number of misses is computed by dividing by the scan line spacing, which supposes a constant stage speed.

HOWIE: The point is that the information you are keeping is based on the results of Strand. If one wanted to find the ionization by another method, then you might not have enough information without the bit patterns themselves. For instance if you wanted to find the gap lengths, you would then have to know the number of consecutive misses.

HALL: We do actually count the number of consecutive misses in each cell, but we only keep the integrated number.

HOUGH: The method used is supposed to be optimum statistically, but there are doubts as to whether it takes account adequately of the real variations in the chamber.
THE CERN VERSION OF MIST

S.J. McCARROLL
CERN, Geneva

MIST is a FAP coded 709 program that runs with the FORTRAN monitor. The present program is a modified version of one designed and written at Brookhaven National Laboratory by J.M. Friedman. *

MIST processes the data punched on the CERN scan tables (Milady I and II) and produces a tape for input to HAZE for use with the HPD, and the programs THRESH and GRIND. (See figures 1, 2, 3 and 4).

The Milady is used to punch three points for each track plus information used to classify the tracks and events at a later time. It also punches the coordinates of a pair of fiducials and information to identify the picture and event.

The HPD scans a picture along parallel lines on a rectangular grid and digitizes points as it finds them. The HAZE program uses the rough digitisations from Milady to construct a gating region for each track of interest to aid in sorting the data from the HPD.

The program

MIST makes three major changes in the format of the data and checks for errors in punching and in sequence.

The first change is in the order of the data. Milady processes the three views of an experiment in parallel, that is all views of one event are punched together. HAZE processes the views in series, that is all events of one view are processed together. MIST stores the data for each view from the Milady on a different tape. At the end of a roll MIST will either leave the data on separate tapes or transfer all views to the tape for view one depending on the value of an input parameter.

* See MIST Bubble Chamber Data Reduction Program by Jerry M. Friedman, Brookhaven National Laboratory Report AMD 309, February 1963.
Secondly, MIST converts the Milady coordinates to a standard rectangular system. A point on the Milady is given by two radial coordinates and a slice number which refers to one of four divisions of the picture. Each radial coordinate is a four digit number of which the high order digit is an integer from 0 to 9 and the three low order digits are cyclic decimal.

The third change that MIST makes is to add track numbers to all tracks and to find the primary vertex of a connecting track. A vertex number is punched by the Milady on each track card; on the incoming track of a charged vertex, this number has an overpunch. The program looks at the vertex number of a track; if it is an uncharged vertex, that is if it is a new number with no overpunch, the track is given the number 1. If the vertex is charged, i.e. the number has an overpunch, the track is given the number 0. If the track has the same vertex number as the previous track, it is given a track number of 1 plus the previous track number. If a track must be punched in more than one segment (see description of input), all segments are given the same track number.

To find the primary vertex for a secondary vertex, the program makes up a table of the coordinates of all vertices. For an uncharged vertex, it stores as the coordinates of the vertex, the \( X_1, Y_1 \) of the first track with the vertex number, and for a charged vertex, it takes the \( X_2, Y_2 \) of the incoming track. Then when a charged vertex is found, the program compares the \( X_1, Y_1 \) of the incoming track with the coordinates of all the vertices in the table and calls the closest vertex to \( X_1, Y_1 \) the primary vertex.

Finally, MIST checks the data for the proper sequence, and for errors in punching. It makes no decisions; if an event is not correct, MIST discards it.

**INPUT**

MIST recognises six different cards by the following code in column 30:

- 0 Fiducial
- 1 Track card
- 6 View error card
- 7 Event error card
- 8 Beginning of roll card
- 9 Comment card
The deck of cards from the Milady must be in the following order:

- **Beginning of Roll**
- **Frame 1, Event 1**
- **View 1 Fiducial**
  
  Track
  
  .
  
  .
  
  **View 2 Fiducial**
  
  Track
  
  .
  
  .
  
  **View 3 Fiducial**
  
  Track
  
  .
  
  .
  
  **Frame 1, Event 2**
  
  **Fiducial**
  
  Track
  
  .
  
  .
  
  **Beginning of Roll**

Some conventions have been adopted to simplify testing for MIST and processing by other programs. Pictures and events must be measured in ascending order; views can be measured in any order. Three points are measured along a track in the direction of motion. If a track is curved through more than 90°, it can be punched in several segments by punching a "continue" indicator. All segments but the last should have the continue indicator punched. A connecting track is given the vertex number of the secondary vertex and must be the first track punched with that vertex number. The primary vertex from which it comes must be punched before it. Outgoing tracks may be punched in any order, but they must be punched in the same order in all views.
Data cards

Two data cards are input before the deck of Milady data. The first contains the code to place all views on one tape and a fiducial separation and tolerance for each view (see detailed information on fiducials). The second has a date, a tape number and an identifying comment if desired. The tape number should be the number of the output tape; this information is used to put a label record at the beginning of the output tape.

All the input is stored on the FORTRAN Monitor System Input tape to be executed as a Monitor Job.

Detailed program operation

Test cards

An overpunch *) in column 1 indicates a test card. The program ignores all such cards but counts the ones before and after the Beginning of Roll card.

Fiducial cards

a) Error checks

1. The frame number is checked for non-numeric characters

2. If there is no overpunch in column 10, the frame number is checked for sequence **) and the event number for the proper sequence within frames.

3. The view number must be equal to 1, 2, 3, or 4 and there must not be more than one fiducial card for each view.

4. The fiducial coordinates are checked for non-numeric characters, converted to the rectangular system and the separation between the fiducials checked. The separation constants and tolerances can be input on the Fixed Data card, but if they are blank on this card, the values set in the program are used.

*) An overpunch is an 11 punch.

**) If there is an overpunch in col. 10 and if the number in col. 10 is 0, then the 1401 stores the octal character 52 instead of 40. The program checks for this possibility.
5. There must not be two fiducial cards in a row.
6. Each fiducial card is checked to establish that only two sets of coordinates are present.

b) Processing

1. If all tests are passed, the fiducial card is stored for the appropriate view. Other information on the card is transferred without checking.
2. A just found fiducial indicator is set.
3. A fiducial for this view is set.

Track cards

a) Error checks

1. The first track must be preceded by a fiducial.
2. The viewpoint frame number on each track must agree with the preceding fiducial.
3. A track coordinate or frame number may not have any non-numeric character.

b) Processing

1. A count is kept of the number of tracks in each view; all views must have the same number of tracks.
2. The quality code is tested: 0 = good, 1 = poor, and 2 = non-existent in that view. If the track is non-existent, it is not stored for output, but it is counted in the total number of tracks for that view, and the track number and vertex number are treated as for an existing track.
3. If a track exists, its coordinates are converted, scanners number, scanning table and scan date - all from the preceding fiducial are inserted, and a track number assigned. It is then stored for output. If the track is an incoming track to a charged vertex, the connecting vertex is found, and its number inserted.
4. If the track has a new vertex number, the coordinates of the vertex are stored in the table of vertices.* The maximum number of vertices is 10.

Comment cards

Comment cards can be inserted anywhere, but the frame number and view number must agree with the preceding fiducial. Only the last comment card input for a view will be output, but inserting more than one comment per view will not cause an error.

Event error and view error cards

If the program finds one of these cards, it will delete the present event or view and cancel any error indicators that have been set. The program counts the number of each that occur and outputs it in the Roll Summary.

Event tally

When the program finds a fiducial card with a new event or picture number it tallies the old event. It tests all error indicators; if none are set, it checks to see that all views have the same number of tracks and that there are at least two correct views. Then it writes each view on a separate tape and returns to process the next event.

If the program finds an error in a view, it clears the indicators for that view and looks for the next view in that event. If the error found is an event error, it writes an error message, adds to the tally of errors, clears all indicators and looks for the next event. If either a view or event error card is found, it adds to the count, clears the indicators and proceeds to the next card.

OUTPUT

Scan tape

The first record of the tape has a BOL in columns 1 - 3, the tape number in columns 7 - 12, the number 152,004 in columns 31 - 36 and any comment input on the second data card in the remaining columns.

*) Since the vertex numbers start with 1, the combination of an over-punch and the number 0 is impossible. (See note on fiducial card).
This label record is used for identification.

Then the Milady data in the following order:

View 1

Frame 1 Event 1  Fiducial
                              Track
                              .
                              .
                              .

Frame 1 Event 2  Fiducial
                              Track
                              .
                              .
                              .
                              .
                              .
                              .
                              .
                              .
                              E.O.R.

View 2

Frame 1 Event 1  Fiducial
                              .
                              .
                              .
                              .
                              .

Frame 2 Event 2
                              .
                              .
                              .
                              E.O.R.
The scan data is followed by an End of file, another label record identical to the initial label except for a Z in column 1 instead of a B, and another End of file.

System output tape

For each event rejected or conditionally accepted the program prints a line giving frame and event number, scan date, scanner, table, the first X and Y coordinates of the event, and a code indicating the reason for rejection.

At the end of each roll, it prints a Roll Summary which lists the initial and final frame processed, the number of cards with test node overpunch after non-test data, the number of view and event error cards, and the number of rejected events by category.

At the end of a run, a Run Summary gives the views, if any, not present, the number of accepted and rejected events for scan table, scanner and remeasure number, and a histogram of the hundreds, tens and units digits of the X and Y coordinates to check the operation of the scan tables.
### CARD PUNCHING FORMATS

<table>
<thead>
<tr>
<th>Column</th>
<th>Contents</th>
<th>Card</th>
<th>Fid</th>
<th>Trk</th>
<th>View</th>
<th>Eve</th>
<th>End</th>
<th>Conn</th>
<th>Err,</th>
<th>Err,</th>
<th>Roll</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0/P</td>
<td>&quot;Test&quot; Mode</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 - 2</td>
<td>Experiment No.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 - 4</td>
<td>Skip x 2</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 - 6</td>
<td>Roll No.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 - 10</td>
<td>Frame No.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 O/P</td>
<td>Roll, Frame and View Nos., from switches</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>View, No.</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Vertex No.</td>
<td>o</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 O/P</td>
<td>Charged Vertex</td>
<td>o</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13 - 14</td>
<td>Skip x 2</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15 - 16</td>
<td>Scan Table No. (wired in)</td>
<td>x</td>
<td>o</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Skip x 1</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18 - 22</td>
<td>Scan Date</td>
<td>x</td>
<td>o</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Event No.</td>
<td>x</td>
<td>o</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24 - 27</td>
<td>Skip x 4</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28 - 29</td>
<td>Scanners No.</td>
<td>x</td>
<td>o</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Card Format No. (auto.inserted)</td>
<td>o</td>
<td>1</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31 - 33</td>
<td>Skip x 3</td>
<td>d</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Sub-experiment No.</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35 - 40</td>
<td>Skip x 6</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Fiducial Code</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>Reomasure No.</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>Unassigned</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>Code for mass of particle</td>
<td>o</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45 O/P</td>
<td>Range</td>
<td>o</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>Poor, gives 1, impossible, 2 otherwise skip</td>
<td>o</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>Continue, gives 1, otherwise skip</td>
<td>o</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>Skip</td>
<td>o</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>Slice 1</td>
<td>x</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50 - 57</td>
<td>r1, r2</td>
<td>x</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>Slice 2</td>
<td>x</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59 - 66</td>
<td>r2, r3</td>
<td>x</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>Slice 3</td>
<td>o</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>68 - 75</td>
<td>r3, r4</td>
<td>o</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>76 - 77</td>
<td>Skip x 2</td>
<td>o</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>Ionization</td>
<td>o</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>79 - 80</td>
<td>Skip</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

x = character  
o = no character punched  
0/P = overpunch row 11
MIST OUTPUT FORMAT
(Identical with Scan Tape B)

<table>
<thead>
<tr>
<th>Column</th>
<th>Contents</th>
<th>Fid</th>
<th>Track</th>
<th>End of Roll</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 2</td>
<td>Experiment No.</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>3 - 10</td>
<td>Frame No. (3, 4 zero)</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>10</td>
<td>(X-overpunch) Do not believe frame no. on film</td>
<td>x</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>View</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>12</td>
<td>Vertex</td>
<td>0</td>
<td>x</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>13 - 14</td>
<td>Track</td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>15 - 16</td>
<td>Scan Table No.</td>
<td>x</td>
<td>x*</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>17</td>
<td>Connecting vertex No.</td>
<td>0</td>
<td>x</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>18 - 22</td>
<td>Scan Date</td>
<td>x</td>
<td>x*</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>23</td>
<td>Event No.</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>24 - 26</td>
<td>Unassigned</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>27</td>
<td>Beginning of roll put = 1</td>
<td>0</td>
<td></td>
<td>x</td>
<td>0</td>
</tr>
<tr>
<td>28 - 29</td>
<td>Scanner No.</td>
<td>x</td>
<td>x*</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>30</td>
<td>Card Format No.</td>
<td></td>
<td></td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>31 - 33</td>
<td>Unassigned</td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>34</td>
<td>Sub-experiment No.</td>
<td>x</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>35 - 40</td>
<td>Unassigned</td>
<td>o</td>
<td></td>
<td>x</td>
<td>0</td>
</tr>
<tr>
<td>41</td>
<td>Financial Code</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>42</td>
<td>Remeasure No.</td>
<td>x</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>43</td>
<td>Unassigned</td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>44</td>
<td>Code for mass of particle</td>
<td>0</td>
<td>x</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>Sign of charge</td>
<td>0</td>
<td>x</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>Range Symbol (X-overpunch)</td>
<td>o</td>
<td>x</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>46</td>
<td>Quality Estimate (0, 1 or 2)</td>
<td>o</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>47 - 48</td>
<td>Unassigned</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>49 - 75</td>
<td>Coordinates</td>
<td>x</td>
<td>x</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>76 - 77</td>
<td>Unassigned</td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>78</td>
<td>Ionization</td>
<td>0</td>
<td>x</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>79 - 80</td>
<td>Unassigned</td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

x = character
o = no character punched
* = not necessary, i.e. can be omitted without causing trouble

7184/mh
Figure captions

Fig. 1  Block diagram of the main program.
Fig. 2  Fiducial routine.
Fig. 3  Track routine.
Fig. 4  Error routines and tally routine.
Figure 1
Figure 2
Figure 3
TALLY

Check indicators for event and view errors

Check # of good views, and # of tracks in each view

Add the statistics for rejected views

Add the statistics for accepted views

Print error message

Write fiducial, comment (if any), and tracks on tape for each view

Reset all indicators except sequence error indicators

RETURN

FE1

Search for next picture

Count view error cards that occur

If an event error card is not followed by a card with new picture #, add to event error count and continue

When new picture # is found tally last event

RETURN

FE2

Search for next event

Count view error cards that occur

When new picture #, event #, or event error card is found tally last event

RETURN

FE4

Search for next view

If view error card is found, reset all view counts and indicators

If event error card, new picture #, or event # is found, tally last event

RETURN
DISCUSSION

HAGOPIAN: What were the units you were using and secondly how long does the MIST program take per event on the computer?

POWELL: The units correspond to 20 $\mu$ least counts on the film.

McCARROLL: So far we haven't processed many events together and the only timing estimate I have is for a set of 6 events on two rolls. These took 1.7 minutes on the 709.

SCHIFF: I would like to ask a question about rejects; what is your policy, do you abandon them as far as the FSD is concerned and use IEPG, or do you make a second premeasurement and wait to merge it with the good ones that went through MIST?

McCARROLL: Our policy at the moment is to re-measure. The most frequent error has been multiple punches in one column. In this case the 1401 won't accept the card as a BCD card, and it has to be discarded; the program then has insufficient information in one view and MIST rejects the event.

HOUGH: The MIST programs at CERN and Brookhaven are still related, so some of the errors we have found might still be applicable. One error arises from the fact that the view error card has a view number on it. Now, the most frequent occurrence of view error cards is when you are measuring one view and find you have forgotten something in the previous view, and we found that when a view error card was punched, it was fairly random as to whether the view number punched was the view being processed or the previous one, and MIST was not capable of distinguishing between the two cases. To correct this error we made it a rule that when a view error card occurred it cancelled the previous view, irrespective of what view number was on the card.

Another point: it sometimes occurs that by pushing the "impossible" button for a track on one view, one necessarily rejects the whole event. For example, with beam tracks, one of the cameras will generally be necessary for the reconstruction and if you push the impossible button on that view, then you can't possibly reconstruct the event. Therefore, one's policy should be that if you are going to call the track impossible, then one does not consider the event at all.

Finally, we have a fairly good diagnostic on the rejects, but we omitted to make a summary of the successful ones.
CONNOLLY: I would like to make a point about the usefulness of tables versus constructing the event type from the geometry of the tracks. Your scheme requires the scanner to push buttons to indicate whether it is a charged or neutral vertex. We have taken that out of the Brookhaven scheme, so that once a scanner has put in the event identification at the beginning of the event, she has no further buttons to push except those which will transfer coordinates to the card. In our early runs a fairly frequent mistake the scanner made was to forget to punch the vertex number.

BLOCH: Are you making a check on corresponding points for vertices?

McCARROLL: Not at the moment, but we intend to do so.

BLOCH: I have another question on the hardware of the scanning table. You mentioned that you divide the picture into slices; that seems to present two difficulties. Firstly, precision, you have to be sure when you use this additional motion that you know the relation between the sets of coordinates and I think that is not a very easy problem.

Secondly, to make a fast premeasurement, if you have to do an extra manual operation to change slices while measuring it must take more time. Why do you choose this system which might be needed with very large chambers, but might not be useful right now?

WISKOTT: With the existing chambers we find that the arms of our operators are not sufficiently long. The region on the table where you can easily do measurements is limited to a length of about 50 cm from the operator. Pictures are typically 100 cm long so you have to have the possibility to move the picture toward the operator. If you put her on the long side of the table, I think it is still unsatisfactory on account of not being able to look along the tracks.

SCHIFF: I was going to mention that. For existing chambers if you have to be able to measure all at one time, you have to turn the thing sideways and measure 50 cm deep and one meter wide. We have been doing that at the Ecole Polytechnique now for three years and we didn't find any difficulty. We might have some difficulty in scanning at very high energies even though in that case we scan with two operators. But for measuring and pre-measuring where the scanning is supposed to have been done already, it may not be too difficult.
HALL: We find an equally important task of MIST is the production of a detailed summary of each of the events that were scanned together with the efficiency of the scanner and exactly what happened in every event.

McCARROLL: We are going to put into MIST the provision to output a comment for each event. We will have a comment for each event and this then will be output on a tape as a record of what happened.

MOORHEAD: I can explain this slight difference between Berkeley and CERN. At Berkeley you keep a library system on the basis of numerical order of events. We are keeping with the library an index tape. This index tape will be based on the comment records from MIST which contain the information required. This tape will then be used to locate events on the GRIND library; it can also be used to print out a record of events as they are processed through the system.
ADAPTATION OF THRESH AND GRIND FOR HPD

J.M. HOWIE

CERN, Geneva

There are three major changes which need to be made to the THRESH-GRIND system of programs in order that the data from the HPD might be processed satisfactorily. These modifications will be described under the following general headings.

1. Input to THRESH
2. Reconstruction of the Vertices
3. The Labelling System

THRESH is the program which makes the geometrical reconstruction of the tracks in space, and GRIND is the kinematics program. In order to distinguish between the two versions of THRESH, i.e. the versions before and after modification, I will refer to the unmodified version as "THRESH", and to the modified version for the HPD as "HPD THRESH".

1. Input

The input to THRESH can be split up into two parts. Firstly there is the basic geometry data necessary for any reconstruction in space eg. camera coordinates, fiducial coordinates, refractive indices etc. This data is contained in TITLE 1 and is read in at the beginning of THRESH. (TITLES 2, 3, 4 and 5 are other sets of basic data which are used in GRIND). Since this data depends on the particular experiment and chamber used, it will not normally have to be changed during a THRESH run.

The main part of the input to THRESH is of course the measurement data for each event. This is in the form of a series of BCD records on magnetic tape, the first record of an event being an identification record containing the word 444444 (i.e. 4's record). The last event on the input tape is terminated by a 7's record (i.e. the BCD word 777777). The data for one event is read into THRESH by a Fortran Subroutine (EVENT 2), and then processed completely by THRESH, before the data for the next event is read into store.

For HPD THRESH, the TITLE 1 data is read in as before, but the input data, which is the output tape from HAZE is in an entirely
different format from the normal input to THRESH. Therefore a new input subroutine (EVENT 2) has been written for HPD THRESH. This has been written in PAP because there is a large amount of unpacking of words to be done. The input tape for HPD THRESH has the following form:

```
LABEL RECORD (BCD)
E.O.F.
10,000 RECORD (BINARY)
11,000 RECORD view 3 (BINARY)
11,000 RECORD view 2 (BINARY)
11,000 RECORD view 1 (BINARY)
   etc.
```

The 10,000 and 11,000 records are so called because the first words of these records are \(10000\) and \(11000\) octal respectively. The 10,000 record contains a few identification words, but these are not used at all by THRESH. A 10,000 record is written by HAZE every time the roll of film is changed. The reason why the views are given in the order 3,2, 1 is because the views for each event have to be merged together within HAZE, and this is the final order when the merging process is completed. Because of the new input format for HPD THRESH, one or two additional small changes have had to be made.

a) Coordinates

In THRESH, all coordinates (i.e. of points on tracks, or fiducials or vertices) are read into a two-dimensional array \(BSTR (2,1500)\) i.e. \(BSTR = (X_1 \ X_2 \ X_3 \ X_4 \ \ldots \)

\[(Y_1 \ Y_2 \ Y_3 \ Y_4 \ \ldots)\]

In the HPD THRESH input, the X and Y coordinates of a point are packed into one word.

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>35</td>
<td></td>
</tr>
</tbody>
</table>

These are stored in the array \(BSTR\) as if it were a one dimensional array

\[\text{i.e. } BSTR = (X_1 \ Y_1 \ X_3 \ Y_3 \ X_5 \ Y_5 \ \ldots)

\[(X_2 \ Y_2 \ X_4 \ Y_4 \ X_6 \ Y_6 \ \ldots)\]
To make the HPD THRESH as compatible as possible with
THRESH, a function FNBSTR has been introduced, where
FNBSTR (I, J) is defined as follows:
If I = 1, the function unpacks the first half of the Jth
word of BSTR, and if I = 2 it unpacks the second half of
the word.
Hence one merely has to replace BSTR (I,J) by FNBSTR (I,J)
everywhere it appears in the program.

b) Fiducials

The labelling rotation used in the THRESH-GRIND system is
such that track vertices are denoted by letters and fiducial
marks by numbers. Allowance is made for up to 10 fiducial
marks and these are numbered 0,1,...,9. In the
normal THRESH input, fiducials are recognised by their
numbers, and the next four numbers following the fiducial
number give the measurements for this fiducial in views
1,2,3,4.

In the output from HAZE there are facilities for the measure-
ments of 6 fiducials on each view, but no indication is
given of what the number of each fiducial is. With HPD THRESH
this information is now provided to the program through TITLE 1.
In TITLE 1, four words of 6 BCD characters each are read into
store, where the words denote the views 1,2,3,4 respectively
and the 6 characters (which are all numbers) denote the order
in which the fiducials are measured in each view.

e.g. suppose the third word of this block is 247389, then
this means that, in view 3

the first fiducial measured is fiducial 2
the second " " " 4
the third " " " 7

etc.

c) Track labels

In THRESH, tracks are labelled by two characters, either two
different letters or else a letter and a number. Two dif-
ferent letters, labels of the type AB, specify a track which
connects the two vertices A and B. Vertex A will be the
beginning point of the track and vertex B will either be a point of secondary scattering, a decay point or else the stopping point in the chamber of this track. Labels of the type A2 specify a track starting at vertex A, which either passes out of the chamber, or else the end of the track is of no importance.

e.g. the network of tracks shown below could be labelled as indicated.

With the HPD the vertices and tracks are given numbers at the scan table, a vertex being given a number between 1 and 9, and a track being given a number between 0 and 99. On the HAZE output the label of each track is given by a SCD word \((V_2 - V_1 T T)\).

where \(V_1\) is the vertex of the track,
\(V_2\) is the connecting vertex if it exists
and \(T T\) are the two digits of the track number

In HPD THRESH the vertex numbers are converted to letters by adding \((20)_{octal}\) to the number in store

e.g. \(1 = (01)_o \rightarrow (21)_o = A\)
\(9 = (11)_o \rightarrow (31)_o = 1\)
The track is then labelled as in normal THRESH, i.e., by the two letters if there is a connecting vertex, the track number in this case being ignored, or else by a letter and a number, the number being the track number already given (T T).

d) Incoming tracks

The output of HAZE is such that measurements on a track are given in the order of the direction of motion of the particle. Therefore a track coming into a vertex, (A say), will have the vertex point A at the end of the track, whilst all the outgoing tracks will have the vertex A at the beginning of the track. For the geometrical reconstruction, THRESH expects to have all tracks starting at the vertex point, therefore for HPD THRESH all incoming tracks have to be reversed. There is no difficulty in identifying these tracks because on the scan table all incoming tracks are given a track number of zero.

2. Reconstruction of the vertex

In the input to THRESH, the track vertices are usually given as measured points on each view and hence are easily reconstructed. THRESH then fits helices to all tracks and finally transfers the geometrical information so determined into GRIND; i.e., the coordinates of the vertices and the parameters of the track helices. For the case when a vertex is not given as a measured point, THRESH, for each of the tracks leading from this vertex, reconstructs the first measured point on one of the views as the vertex of the track. Hence a separate determination of the vertex is made for each track leaving the vertex. THRESH then fits helices to all the tracks as before, but does not attempt to make a more accurate determination of any unmeasured vertices. This is done in GRIND using linear extrapolation. As an example we consider the case of just two tracks.

Let $V_1$ be the determination of the vertex on the first track and $V_2$ be the determination of the vertex on the second track.
The actual vertex V of the two tracks is found by linearly extrapolating backwards from the approximate vertices \( V_1 \) and \( V_2 \), and solving by least squares to determine the best "point of intersection" of the two lines. The fact that only linear extrapolation is used to construct the vertex point in this case is rather unsatisfactory, but on the whole not too serious, since THRESH was primarily expected to deal with measured vertices, and it is usual practice to measure all relevant vertices at the IEP.

With the HFD though, it is not possible to measure the track vertices satisfactorily and hence no vertices are given as measured points. In HFD THRESH therefore, a better method for constructing the vertex points has been incorporated, which will provide more accurate determinations of the vertices.

The difference between the two methods lies in the final reconstruction of the vertex. We assume we have reached the stage where an approximate vertex has been constructed for each track, and also a helix has been fitted to each track. To reconstruct a particular vertex, we now attempt to find the best "point of intersection" of all the track helices which emit from this vertex, i.e. we extrapolate back along the helices instead of doing a simple linear extrapolation.

In the explanation of the method I shall use the following notation:

1) The basic system of coordinates is \( x, y, z \).
2) The axes of the helices are all parallel to the \( z \)-axis.
3) The coordinates of the approximate vertex on a track are \( A, B, C \).
4) The equation of a helix is given with respect to an axis system \( x', y', z' \), where the origin of these coordinates is transferred to the vertex point \( (A, B, C) \), and the axes are rotated through an angle \( \theta \) in the \( (x, y) \)-plane, so that the \( x' \)-axis is pointing outwards along the normal to the helix, i.e.
the equation of the helix is

\[ x' = \rho (\cos \theta - 1) \]
\[ y' = \rho \sin \theta \]
\[ z' = \rho \theta \tan \alpha \]

where \( \rho \) is the radius and \( \alpha \) is the dip angle. \( \theta \) is the parameter which defines the points on the helix; \( \theta = 0 \) being the origin, i.e. the point \((A, B, C)\).

Referred to the basic coordinate system, a point on the helix is given by

\[ x = A + x' \cos \beta - y' \sin \beta \]
\[ y = B + x' \sin \beta + y' \cos \beta \]
\[ z = C + z' \]

Since \( \theta \) is the only parameter of the helix which varies, we may express a point on the helix as a function of \( \theta \), i.e.

\[ x = x(\theta) \]
\[ y = y(\theta) \]
\[ z = z(\theta) \]

For the sake of simplicity we consider now the case of one vertex and just two tracks emitting from the vertex. Let quantities with suffix 1 refer to one track and those with suffix 2 refer to the second track.
We suppose that $\theta_1 = \theta_1^*$ and $\theta_2 = \theta_2^*$ are approximations for the true vortex point on tracks 1 and 2 respectively, and that the true values of $\theta_1$ and $\theta_2$ are

$$
\theta_1 = \theta_1^* + \Delta \theta_1 \quad \theta_2 = \theta_2^* + \Delta \theta_2
$$

By equating the $x$, $y$, and $z$ coordinates of the two points on the respective helices we have from Eq (3) that

$$
\begin{align*}
x_1(\theta_1^* + \Delta \theta_1) &= x_2(\theta_2^* + \Delta \theta_2) \\
y_1(\theta_1^* + \Delta \theta_1) &= y_2(\theta_2^* + \Delta \theta_2) \\
z_1(\theta_1^* + \Delta \theta_1) &= z_2(\theta_2^* + \Delta \theta_2)
\end{align*}
$$

(4)

Now by expanding the left and right hand sides of the above equations in Taylor series about $\theta_1$ and $\theta_2$, respectively, and ignoring powers of $\Delta \theta_1$ and $\Delta \theta_2$ greater than the first we arrive at a linear set of equations for the unknown $\Delta \theta_1$ and $\Delta \theta_2$, i.e.

$$
\begin{align*}
\frac{dx_1}{d\theta}(\theta_1^*) \Delta \theta_1 &= \frac{dx_2}{d\theta}(\theta_2^*) \Delta \theta_2 \\
\frac{dy_1}{d\theta}(\theta_1^*) \Delta \theta_1 &= \frac{dy_2}{d\theta}(\theta_2^*) \Delta \theta_2 \\
\frac{dz_1}{d\theta}(\theta_1^*) \Delta \theta_1 &= \frac{dz_2}{d\theta}(\theta_2^*) \Delta \theta_2
\end{align*}
$$

(5)

The coefficients in the above equations are determined from equations (1) and (2) and the system solved by least squares for the unknowns $\Delta \theta_1$ and $\Delta \theta_2$. We now replace $\theta_1^*$ by $\theta_1^* + \Delta \theta_1$ and $\theta_2^*$ by $\theta_2^* + \Delta \theta_2$ and repeat the process to determine further increments $\Delta \theta_1$ and $\Delta \theta_2$. The iteration converges when the values of $\Delta \theta_1$ and $\Delta \theta_2$ become small enough. (For the first approximation we put $\theta_1^* = \theta_2^* = 0$.

When the iteration converges, we then have two approximations for the vertex given by the final value of $\theta_1^*$ for track 1 and the final value of $\theta_2^*$ for track 2. The true vertex point is therefore taken to be the mean of these two approximations.
The above analysis is only for the simplified model of two tracks, but the method can be extended in an obvious way to deal with more than two tracks. In this case we would have to consider all possible intersections of tracks, which would unfortunately mean a large increase in the number of equations. More precisely if there were \( n \) tracks, then we would have a system of

\[
\frac{3 \times n \times (n-1)}{2}
\]

Because of this large increase in the number of equations, (e.g., for 6 tracks we would have 45 equations), not all combinations are considered in the HPD THRESH routine. Instead, combinations are taken such that, if there are more than two tracks, then each track is only used twice. The true vertex point is, of course taken to be the average of all the separate determinations. At the moment no weighting is used in the calculation of this average, but it is intended to prescribe weights depending on the angles between the tracks at the vertex. The implication is that an intersection point, determined by two tracks which make an angle of about 90°, will be more reliable than one determined by two tracks with only a small angle between them.

A further point concerning the above analysis is that we have made the implicit assumption that all the parameters occurring in the equations (i.e., \( \rho, \alpha, \beta, A, B, C \)) have been determined exactly and are without error. This is of course not true, since for each track, the values of \( \rho, \alpha, \beta, A, B \) and \( C \) are themselves determined from least squares equations and all have errors. Therefore in our equations (5), we also have 6 measured variables for each helix as well as the two unknowns \( \Delta \theta_1 \) and \( \Delta \theta_2 \). The solution of these equations can be determined by using a more generalized method of least squares. Details of the method can be found in various publications on statistics or least squares methods, but for a compact account see Böck (1960) \(^1\). By introducing these errors on the measured variables, one in fact does weight the solution in favour of the good measurements, but the analysis becomes much more complicated. So far at CERN we have not made any elaborate analysis of this sort. To safeguard against using tracks which have been badly measured though, we only use tracks which have converged in the least squares helix fitting procedure.

3. The labelling system

The third major modification which has to be made to THRESH is caused by the labelling system. In normal THRESH, the labels which are given to the vertices and tracks at the scan table are not allocated
arbitrarily, but are used to carry important information from the measuring machines into THRESH and GRIND. Depending on the different kinds of interactions between which one wishes to distinguish, one sets up a series of interaction classes. To the classes are allocated letters and numbers which may be used as labels for vertices and tracks of a particular interaction. In this way one can, for instance, distinguish between a decay point of a neutral particle or a point of secondary scattering. There are also other classes which, for example, define a vertex as an end point of a stopping track, or a track as a straight track. GRIND uses these label classes to give each vertex and each track a number which is called the "Nature" of the point or track. The numerical value of the "nature" of the point or track is equivalent to defining its class or type, and hence one sees that the labelling system is just a means of indicating this "nature".

With the HPD, the labels are prescribed at the Milady scan table, and depend entirely on the order in which the points and tracks are measured. Since the labelling system is now to a large extent a function of the measuring machine rather than the operator, the labels cannot be used in the same manner as before. The "nature" of the points and tracks must therefore be determined in some other way. This in fact is not too difficult, as some of the information which used to be transmitted to the programs through the labelling system is now contained directly in the input data to HPD THRESH. This additional information arises from decisions which are made at the scan table and is as follows:

1) whether a track is straight or not,
2) whether a track stops within the chamber,
3) whether a track should be considered to have zero range,
4) a mass code and a charge code are also given, by which the scanner may indicate either the mass or charge of a particle (or both), if he really wishes.

We now give a list of all the properties which are used to define the "nature" of a point or a track and we also indicate how each property is determined in HPD THRESH.

a) Tracks

1) stopping track : specified in HPD THRESH input
2) straight track : specified in HPD THRESH input
3) zero-range track : specified in HPD THRESH input,
4) beam track : the beam track will always be the first track measured and will also have a
track number of zero. If there was no beam track then the first track would not have a label zero.

5) connecting track: i.e. a track which has a vertex at both ends; specified in HPD THRESH input.

b) Points

The "nature" of the points can not be determined quite so directly as that of the tracks. From the input data to HPD THRESH we know whether a vertex is charged or uncharged (i.e. if there is an incoming track or not), and we can also determine the total number of measured tracks leaving each vertex, from the labelling system.

By using this information we can distinguish between the various interaction classes used for points. The classes generally used in GRIND are as follows:

1) a) Points of beam interactions (with an odd number of measured tracks).
   b) Points of beam interactions (with an even number of measured tracks). This point will always be the first measured point anyway.

2) Decay points of charged particles (with two measured tracks).
3) Decay points of neutral particles (with two measured tracks).
4) Secondary scattering of particles (with three measured tracks).

Reference

   CERN 60-30.
DISCUSSION

BUDDE: I would like to ask whether the extrapolation to the apex isn't a dangerous procedure, especially for high energy $V^0$s. If you find the apex by extrapolation, as long as it is on the line of flight, the kinematics is still alright, but if now you want to compute decay corrections, you need to know the actual path length which the $V^0$ travelled. Now I think the error introduced by extrapolation will be a systematic one, and this means the path lengths you find will have a tendency to be too short.

HOWIE: Yes, it was this kind of phenomena which I indicated could be accounted for by weighting but obviously if one has only the two secondaries of an energetic $V^0$, the method would be unsatisfactory. Therefore in particular cases this procedure would have to be modified. Speaking generally though, I feel that this would be a good way of determining the vertex, because one doesn't have to know the beginning point. One just takes a good point on the helix and if one has a good fit, then one should obtain a good fit for the vertex.

BURD: I take it that since you didn't mention anything about it, you don't have time sharing problems?

HOWIE: At the moment we are not considering this in detail. I think we can cut down the space taken by THRESH a fair amount. There are certain routines we can leave out, for example, there is a routine which constructs vertex points from actual measured points, and another routine which orders the measurements on a track (with the HPD we can assume that the measurements will be given in the correct order).

BURD: And you think you will share time on the 709* between HAZE and THRESH?

HOWIE: We hope so when we have the system running on the 7090.

TYCKO: You say you have a routine in THRESH which reconstructs vertex points from measured points. Why don't you continue to use that technique. Have you considered reconstructing unmeasured vertex points from tracks using the measurements themselves on each view and then going into space?

HOWIE: This is an alternative method, but I feel a spatial reconstruction will be generally more accurate.

* CERN will have a 7090 as from October 1963.
BURD: It is a much simpler alternative. But you say it is better to do it in space.

HOWIE: If you can get good fits to the helices, then, except for these cases where you only have 2 tracks at a very small angle, I feel you must get a good vertex fit.

CAIKIN: I don't see that there can be a real difference. You are probably looking at the same thing in a different coordinate system.

HUMPHREY: There might be a danger in trying to reconstruct the point on the film in that for strongly dipping tracks you can get very peculiar shapes on the film which may be very difficult to extrapolate to the vertex. Dick Hartung has developed a method of fitting all tracks simultaneously in which the vertex point automatically falls out but it does involve solving for more variables. I was wondering if you had looked into this method?

HOWIE: I have discussed this approach with a statistician at CERN, but we haven't gone into any details. He thought it was feasible. I think you have to solve the problem by the method of maximum likelihood.

HUMPHREY: I think Dick Hartung has actually written a program which does this.

HALL: Could you say what your least squares criterion is with the present scheme? In other words what are you minimising?

HOWIE: We have an overdetermined set of linear equations and we simply minimize the residuals. This is, physically equivalent to finding the values of $\theta_1$ and $\theta_2$ which give you the shortest distance between the two helices.
FUTURE PLANS FOR CERN HAZE

W.G. MOORHEAD and W. KRISCHER
CERN, Geneva

(presented by W.G. Moorhead)

The main flow of HAZE has been described by D. Hall. He has also mentioned some of the differences in the program at the three centres - LRL, BNL and CERN.

CERN HAZE now occupies half the 709 store. It is hoped to keep it about that size so as to "share time" with THRESH. The program now works and includes the 90° scan for those tracks which require it.

Unfortunately it gives a high rejection rate due to FILTER. FILTER is discussed later. Some changes thought desirable in the main program, (sometimes called the "DISPATCHER") are:

1) Procedures when error signals are received from the HPD are incomplete. In particular, a roll of film cannot be restarted in the middle after a breakdown.

2) If there is a very large number of tracks so that more than one 60° sweep is required, the tracks to be analysed in each sweep, should be judiciously chosen so as to avoid gating and filtering too many in one scan-line (not merely to take the first eight).

3) Finding the "road" fiducials requires much more program work-space than finding the "glass" fiducials because of the greater uncertainty of their positions. But tracks are not being gated at the beginning of the picture; therefore the track gating work-space should be used.

4) More of the program could be transferred into MIST if there is difficulty in fitting THRESH into store at the same time.

We have processed only about ten events using HAZE. However, we have encountered all (except one) of the cases of failure by FILTER described in other papers. We are particularly worried about "average points" being found which are in some cases a considerable distance from

7184/mh
the track.

The GATE and FILTER routines of HAZE need considerable modification. In fact we feel that FILTER can never work properly unless it is modified so as to converge towards a track-following system which makes use of points already found on the track.

There are two basic reasons why FILTER II does not work:

1) The road edge is always computed from the circle fitted through the scan-table points. This means that a coarse histogram (about a track width per bin) has to be used throughout due to possible misalignments of the road and the track. This gives very poor resolution of the interesting track from the background.

2) FILTER is called each time when twenty points in the road have been accumulated by GATE. These twenty points can represent a very widely varying length of track. When there are nearby or crossing tracks the twenty points are shared by several tracks, thus further making resolution difficult. (This problem may be partly solved by adding successive histograms together but this is at best a palliative).

At CERN we have decided to rewrite the GATE and FILTER part of HAZE. For this an earlier collaboration with NIRNS has been renewed. The basic method has been described by J. Burren. Gating is over a fixed number of scan-lines (called a slice) at a time. At first the scan-table road slope is used, then the slope, road width and histogram bin width are changed on the basis of extrapolation from previously found average points.

It may be noted in connection with the question of histograms that if the track can be accurately extrapolated, then a very narrow road (say 2\(\sigma\) wide) can be used. This means that histograms are unnecessary because nearly all the digitizings found in the road must belong to the track. However, it may be better to keep a slightly wider road in order to see, in advance, tracks intersecting at a small angle.

It has been shown (in CERN 61-31) that the RMS deviation of digitizings from a good track is about 3 or 4\(\sigma\). If the noise can be rejected completely the average points should ultimately be accurate to \(\frac{3\sigma}{\sqrt{N}}\) where \(N\) is the number of digitizings used in the averaging. However, the least count of the output is the same as the least count of the digitizers viz. 1.6 \(\mu\) and this is certainly good enough.

The GATE and FILTER being programmed for CERN HAZE (figure 1) differs in some respects from the NIRNS experimental program. (The NIRNS final program will differ from this also, of course !)
1) The number of scan-lines per slice is a program variable. It is hoped to find the best value by experiment.

2) Scan-lines are counted separately for each track from its beginning. This avoids having only a short piece of track in the first slice. Gating will also cease immediately at the end of a track so as to avoid picking up digitizings further along which do not belong to it.

3) There is a subroutine in HAZE to compute the road edge and tangent from the circle through the scan-table points. FILTER will use this to help to predict ahead with decreasing weight as the track becomes "established". It will also be used to check if the track leaves the road.

4) There are also many differences in detail of FILTER operation such as that no intermediate histogram block size is used - only 16 least counts and 4 least counts.

5) Parabolic road edges will be used over the slice for all tracks.

6) The main difficulty with any track-following scheme lies in getting started. The digitizings in the wide road in the first two slices will be kept and dealt with afterwards (or concurrently in a second priority time-shared program). There are several reasons for this:

   a) To clean up the first two average points of the track which have come from a coarse histogram.
   
   b) To find the very first digitizing of the track using a narrower road extrapolated back.
   
   c) To treat short tracks (less than two slices long) by a special procedure.
   
   d) If a track does not get started, to examine the digitizings of the first two slices more carefully to see if anything can be picked up.

7) In any case, if no track has been started in a road it is proposed to return the stage to the beginning of that track and deal with it separately in a special routine.
Figure caption

Fig. 1 Provisional flow chart for the new GATE routine.
Figure 1
HOUGH: If you develop 2 or more pulses in the first slice, will you follow each of them?

MOORHEAD: Yes, we do. I would be grateful if anyone could tell us of a better way of initiating than a course histogram.

HALL: We don't really believe the filtering problem is solved at Berkeley. However, we do believe it can be solved in some sense within the existing framework. I mean that finally less than 50% of the errors will be due to Filter. So in effect this is what the future plans of a FILTER at LRL will be.

BLOCH: Would you care to make an estimate of the time needed to carry through that program?

MOORHEAD: This GATE, of which I've shown you the flow chart, I am in the middle of writing. The only part which is missing really is the GATE ROAD block in the bottom right hand corner of the flow chart. It is not writing the thing which counts but getting it to work. I hope that the writing will be complete in a couple of months. I feel sure we can get this working in a far shorter time than we can clear up even one of the troubles in the other version.

EDMONDS: Could you tell us what programming language you are writing this in, and also how far these various techniques you are using are IBM oriented and special to that type of computer?

MOORHEAD: For the first question, we are using FAP. For the second, I don't think they are very much IBM oriented. There are IBM features which I have never been able to make use of, like the indicators; but I think these techniques of linked addresses and using erasable blocks should be available on any machine.

TYCKO: Have you had any experience so far that has made you take what seems to be a much more difficult or at least a more extreme approach than Brookhaven or Berkeley?

MOORHEAD: We have not had much experience, based on about 10 events, but that was enough to show us that there were things which would never be satisfactory for production, e.g. a track getting lost
in a not very difficult region, just because of the 2C points and a coarse histogram. We had a vested interest in trying to make this work because it is much easier to use someone else's program. But we had enough experience to see all the difficulties already mentioned except possibly the sinusoidal curve inside the road. To my mind the most disturbing one is that of getting average points which are wrong. The smoothing should be done when you have the HPD points; it is a very powerful device with all the accuracy you can want. One should do the smoothing when you have got the full accuracy of the machine.

HALL: Could you explain why the 3x1000 word blocks are better than 2x2000 word blocks?

BURREN: The actual size of the buffer that you require is the size necessary to even out the flow of the data because your program has to filter every now and again. You could perhaps get away with a buffer of only 50 words but FILTER takes rather a long time, so you have to have a big enough buffer to smooth out while doing filtering. That gives the smallest size of buffer that you could get away with. But you don't want to keep on trapping, so you spread a few traps down a bigger buffer. We actually use 4 buffers of 250 words.

HALL: It is essentially a space saving device.

BURREN: Yes, you work with the smallest buffer you can, so that you don't waste space on a buffer.
ATTEMPTS TO OBTAIN HIGH RATE, HIGH QUALITY MEASUREMENT OF EVENTS DESPITE FILTER II IMPERFECTIONS

P.V.C. HOUGH
Brookhaven National Laboratory, Upton, L.I.

1. Basic philosophy

For several months we have been quite conscious of the weaknesses of FILTER II. As used initially, for about 20% of the tracks and therefore for a large fraction of the events, FILTER II gave erroneous master points. This performance effectively precluded running in the system and so prevented finding and curing any other difficulties which lay in the path of production measurement. So we decided to work around FILTER II and force good filter output in order to get a system, then return to the filter problem with a sound idea of the changes necessary. CONTROLLED FILTER was invented and got running by N.W. Webre and D. Burd, along the lines described elsewhere in the conference proceedings by D. Burd.

2. Benefits derived to date

Perhaps the most valuable single ingredient in CONTROLLED FILTER has been the CRT display of all digitizings in a read, plus HAZE output master points overplotted as x's. Illustrations of this diagnostic display are given by D. Burd.

The CRT display immediately enabled us to separate clearly road-fitting problems from FILTER problems, a separation which is less clear in a diagnostic organized around the 20 point bites of FILTER II (such as the 421 histogram summary of Berkeley). Next the display revealed the "micro-errors" in FILTER II, i.e. the generation of master points some tens of microns off the track at points where crossing tracks occur at a gap. Finally we acquired in a few weeks a rather clear appreciation of the kinds of failure in FILTER II. The two most common, initiation on a wrong track which soon leaves the road, and carrying off on a crossing track, we believe can be cured within the

* Work performed under the auspices of the United States Atomic Energy Commission.
context of FILTER II, although we have not yet tackled the problem. A residue of genuinely difficult filter cases were found to occur much less frequently and can, we believe, be either force-filtered or permitted to form an accepted rejection category for remeasurement by conventional measuring projector.

To remove micro-errors, a re-cycled least squares parabolic fit of the master points was introduced, and this smoothing routine was allowed to remove up to 3 of the 15-20 points normally output by HAZE. Now for the first time our HAZE output data was sound enough to present in quantity to FOG 135 and we began to get FOG-CLOUDY-FAIR output which fairly meant something.

Two important problems having nothing to do with filtering could also be attacked. The first is reliable measurement of fiducials, and is discussed in detail in a separate report to this conference. The second is the introduction of a number of small improvements in both hardware and software to get smooth on-line operation. In this category, for example, is the development of procedures for re-run of a given event without spoiling the output tape for the run, for restart at an arbitrary picture number, for changing film rapidly, for control of Ferranti errors, and the like. Without such development each small error is able to ruin an entire run.

3. A production system using CONTROLLED FILTER

A by-product of the smoothing routine described above is "computer acceptance criterion". If, after removal of up to 3 bad points, the remainder deviate from a parabola by less than $E$ microns, and if in addition no fraction greater than $r$ of the road length is free of master points the computer accepts the output data and makes no CRT display. (We have worked with $E = \frac{3}{\sqrt{2}}$ and $r = 0.2$, a somewhat tighter criterion than necessary, and $E = \frac{6}{\sqrt{2}}$ and $r = 0.3$, which may be a bit loose). Tracks which fail the computer test are displayed, and the operator may take several actions: 1) he may acquiesce in the rejection, press the direct data interrupt button and so return control to the computer; 2) he may force acceptance by depressing Key # 22 and then the direct data interrupt; 3) he may "manually filter" the track as described in the paper by D. Baird.

The point of a production system with CONTROLLED FILTER is this: present filter program techniques and also those expected to be available in the near future will err in a fraction of cases ranging from 1 in 10 to 1 in 50. For 2-prongs this track error rate will lead to an event rejection rate ranging from 30% to 6% and for 5-track events ranging from 50% to 10%. These rejection percentages can be largely eliminated with CONTROLLED FILTER and the only question is whether it can be justified economically. A recent test on 125 2-prongs
for which 1 track in 10 had to be displayed suggests that the answer is yes. The 125 events were measured in 1-1/2 hours for a rate of 80 events per hour, or about 2/3 the design rate for the system. Considering the sequence of 125 frames for a single view, 40 CRT displays were obtained with an average interrupt time of 7-1/2 seconds per display, for an integrated interrupt time of 5 minutes out of the 30 minutes total. So manual filtering reduced the production rate by only 15%. (The remaining deficit in rate was due to an abnormally close line spacing for which the stage controller happened to be set for this run).

Of course, the basic feedback from a CRT diagnostic to an improved FILTER program should not be interrupted and is not being interrupted. But there is perhaps a real justification for maintaining CONTROLLED FILTER even at a track rejection level of 1 in 50. The justification is computer efficiency, since there seems hardly a more economical time to dispose of a difficulty than immediately, while the film is in place for measurement.

4. Remaining problems

The basic system problem which must be solved at Brookhaven before physics results can be obtained in production is road-fitting. In addition a careful comparison of the output data with results from the conventional analysis system must be completed.

With regard to road-fitting, two event samples of \( \sim 100 \) events and \( \sim 20 \) events showed \( \sim 15\% \) event rejection due to bad road fit. Two other samples of \( \sim 40 \) events and \( \sim 20 \) events showed much worse fitting. The difficulty is believed not due to operator error at the scan table or to lack of precision in the rough digitizer, but rather to a failure to transform correctly from the scan table to FSD coordinate system. A rough digitizing of the internal fiducial most distant from the road fiducial, which is presently taken but not used, will be transformed in HAZE along with the usual road coordinates, and by comparison with this internal fiducial as found by the FSD, feedback will be obtained to correct the transformation. Aberrations in the scan table coordinate system itself have been studied but seem insufficient to explain the effect.

With regard to study of the output data, a sample of some 200 2-pronged from an exposure of the 20" chamber to 2.3 BeV/c \( K^- \) have been analyzed by the conventional system. The comparative study of FSD and conventional results separates into a study of fiducials, a study of track angles and vertex position in the chamber, and a study of track curvatures.
DISCUSSION

GOLDSCHMIDT-CLERMONT: How do you keep the information about the tracks after you have made your master points. Do you retrace the stage or do you keep the gated points?

HOUGH: We store the entire contents of every read.

SCHIFF: Suppose you get the system so that you have 3% rejects, do you feel it would be dangerous to just drop these rejects - would you then have systematic errors with FILTER rejects?

HOUGH: I don't think anyone would object; 3% - 15% has been suggested. One reason for keeping manual FILTER is that there are some events which you do want to measure at all costs.

MOORHEAD: Do you not feel worried about using a 7090 to do what a Frankenstein could do?

HOUGH: That is a question of whether or not you wish to clean up the small percentage.

MOORHEAD: Do you not believe that a program can be written taking only half the store which will do the problem and give a low reject rate?

HOUGH: I don't worry about using the entire 32K - I don't see any advantage in putting FOG or CLOUDY in.

MOORHEAD: Not necessarily FOG and CLOUDY, but some other program - you are using a very small percentage of CPU time to do something which a Frankenstein could do and the rest of the 7094 is standing idle. The problem is only worth solving if it is done in real time.

HOUGH: Let's distinguish two things: the real time versus the non-real time - if you are not working in real time, there is an efficient way of using the 7094, which is to do only GATE in real time to speed up the disc and stage by a factor 2. The overall cycle time is the same, but FILTER is done after.

MOORHEAD: You are still proposing to use the whole of core - even for reading onto tape which is still an inefficient use of the 7094. Our program is going to be kept in half the core.
HOUGH: What are you going to do with the other 16 K?

MOORFIELD: Well, we have got to use it for something or we shall be wasting an enormous amount of money. At least, with half the core free, it is possible to time share, but if your HAZE fills nearly the whole 32 K, you are obliged to let much of the machine stand idle.

CONNOLLY: One group is proposing to time share with 2 separate programs. Hough is proposing to speed up the mechanical rate for the HPD - in the end if both plans work out, both are using the CPU at nearly 100% efficiency.

CALKIN: Surely to use a 7094 with manual FILTER is a very inefficient use of a 7094?

HOUGH: There are 2 questions: whether to GATE in real time and FILTER afterwards, and whether to use manual FILTER at all. To provide 1000 events/shift requires 5 scan tables 2 shifts or 5 SMP's 2 shifts. So it's roughly equivalent to a full SMP system.

CALKIN: Do you think you can do 500 to 1000 events/shift with manual FILTER and do POF too?

HOUGH: The initial stage will be a reflexive one with more and more work being transferred to the computer.

CALKIN: Even if manual FILTER is one event in ten or twenty, you are comparing human reaction times with 7094 times.

HOUGH: The comparison has to be between human reaction times and the time a 7094 takes for a whole picture. You compare the time for some recognition by humans with a time of the order of 10 seconds. That is the time for processing 3 views automatically.

CALKIN: Why cannot the computer take the decisions?

HOUGH: We shall always be trying and modifying the program to do this, but meanwhile go on using manual FILTER.

GOLDSCHMIDT-CLERMONT: It may be desirable to avoid having to put the film back on the HPD for the events which give difficulties, and I wonder if 7094 speeds would allow the following type of operation. First, to write systematically all the measured coordinates on a scratch tape, working asynchronously while the measuring process goes on. Second, to copy the scratch tape onto a library tape which is kept, whenever an event gives difficulties. The logistics of tape handling in this way may be better than that of film handling.
HOUGH: It may be possible to save the road coordinates.

GOLDSCHMIDT-CLERMONT: It could be a mistake of the scanning table for instance. Only if you save all the coordinates, can you hope to save some manipulation with the film?

HOUGH: It would be necessary to do some editing of the number of tapes involved. We have 70 events/tape. If we can find a way of knowing that they are failures and then put them on tape it would be worth considering.

MACLEOD: This morning the figure of 600 events/shift was given and 10% of them held you up and reduced the overall rate by 50% or so. This means that you are spending half of one shift on 60 events, i.e. about 4 minutes/event. What is going on in each 4 minutes. How does this compare with the decision times of the order of 10 seconds?

HOUGH: The only time we have reached a rejection rate of 10% is after manually filtering. The residual is due to road fitting. It is not the 60 out of 600 on the CRT - you have to go back to the scan table for these. According to Berkeley, between 10 and 20% fail in FILTER. We look at the moment at 1 track in 6. This may end up at 1 in 20. This is a question of computer acceptance criteria. Only until the rate is established, can the reflexive procedure be sensibly defended. But it seems to me you only have to take a 20% cut in rate to get them all.

MACLEOD: You look at 1 track in 6. What happens when you get a bad track on the screen?

HOUGH: At present it's every 2 pictures or so. 2 out of 3 times it's OK - the computer has been too restrictive. Depress key 22 and go ahead. The residuals are usually road fit problems. The residual, when that has been solved too is about 1 track in 20.

MACLEOD: When you look at something systematically, what goes on?

HOUGH: 1) In 2 out of 3 cases, press key 22 and the interrupt button and its goes on.

2) In the case where you have to do something set up 3 total digits and press interrupt. It redispays and if it's satisfactory you go on.
THE AUTOMATIC ANALYSIS OF SPARK CHAMBER PICTURES USING A FLYING SPOT DIGITISER

P.M. BLACKALL, G.R. MACLEOD and P. ZANELLA

CERN, Geneva

(presented by P.M. Blackall)

This paper describes an IBM 709/7090 program for the automatic analysis of spark chamber pictures using a flying spot digitiser on-line with the computer. It has been written primarily for the analysis of pictures from a $\gamma^- - p$ elastic scattering experiment run at the CERN PS in January 1963. The program controls the digitiser and carries out all the operations necessary for automatic scanning of the film and measurement and space reconstruction of true elastic scattering events. Whilst a picture is being digitised the central processor time is shared between the program scanning the previous picture and any other compatible independent secondary program.

The program has been constructed from a set of FORTRAN/FAP subroutines of which the majority are independent of the format of the pictures being analysed.

1. Spark chamber experiment

1.1 Experimental arrangement

The experiment was run at the CERN PS in January 1963 by Caldwell et al.\cite{1}; some 250,000 pictures were taken of which 150,000 showed elastic scattering events. The aim of the experiment is to measure the differential cross-section for $\gamma^- - p$ and $\gamma^+ - p$ elastic scattering at angles between $7^\circ$ and $40^\circ$ in the centre-of-mass system. It is also planned to study the behaviour of $\gamma^- - p$ diffraction scattering at very high energies and to search for a possible shrinking of this peak similar to that found with protons.

A schematic diagram of the experimental arrangement is shown in figure 1. Chambers 1, 2, 3 and 4 and the first bending magnet define the path and momentum of the incident pion. Chambers 5, 6, 7 and 8 and the second bending magnet
define the path and momentum of the scattered pion. Chamber 9, which is in fact two 6 gap chambers separated by an air gap, records the path of the recoil proton. All chambers are of parallel plate design.

When the counter selection circuits signal a possible event the camera is triggered and a photograph is taken of all nine spark chambers in two stereoscopic views on a single 24 x 36 mm film frame. This is accomplished by a mirror-lens system for each chamber and view. The two views, called the normal and the stereo, are taken in two perpendicular directions. The normal view is taken from above the chamber and shows a projection in the horizontal plane; the stereo view is taken from the side of the chamber perpendicular to the beam direction and shows a projection in the vertical plane. Cylindrical field lenses placed at the two photographed faces of each chamber ensure that the two views represent orthogonal rather than conical projections of the chambers. The field of both magnets is in the vertical direction.

1.2 Frame format

A typical frame is shown in figure 2. The format has been designed with both manual and automatic scanning in mind. Each view has two fiducial grids of strokes and V-marks. The larger chambers have an extensive array of fiducial marks to allow compensations to be made for distortions in the mirror-lens system. To facilitate the manual scanning, the corresponding views of adjacent pairs of chambers have been placed so that the images of tracks passing through both are co-linear on the film. In the centre of the frame is the data box information which includes the time, date, frame number, roll number and incident pion momentum as well as an array of lights giving a binary representation of the relative frame number. The three large fiducial crosses are used by the scanning program to locate chamber positions within the digitised coordinate system. Figure 3 shows a schematic representation of the chamber view positions on the frame.

A prism mounted in front of one gap in the stereo views of chambers 1 to 8 causes a displacement of the image of any spark in that gap by a distance proportional to its depth in the chamber. This displacement is used to resolve ambiguities arising in correlating tracks in two stereoscopic views when there is more than one track per chamber. The prism position for each chamber is given in table 1.
Table 1

<table>
<thead>
<tr>
<th>CHAMBER</th>
<th>No. OF GAPS</th>
<th>PRISM GAP No.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(ROLL 142)</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>10*</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>6+6*</td>
<td>11</td>
</tr>
<tr>
<td>8</td>
<td>6*</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>6+6*</td>
<td>-</td>
</tr>
</tbody>
</table>

*) Chambers 7 and 9 are double chambers with an air gap.

1.3 Digitising

The flying spot digitiser (HPD) developed by Hough and Powell for the analysis of bubble chamber pictures is used to digitise the spark chamber pictures.

A mechanical system causes a spot of light of diameter 15 microns to scan the film in a television-like raster. The scanning lines are parallel to the length of the film and the images of the chamber plates. To scan one line takes 7.5 milliseconds so that for 500 lines/frame (i.e. a scan line separation of 60 to 70 microns) the total measuring time per frame including film advancement is 5 seconds. Whenever a photomultiplier detects the passage of the spot across a dark image its position is digitised and a "spot coordinate" is transmitted to the computer memory via the Direct Data Connection. This coordinate gives the distance along the scan line of the mid-point of the dark image, and has a word length of 15 bits with a least count equal to 1.6 microns on the film. At the end of every scan line, during the dead time of the spot, a "scan-line coordinate" is transmitted to the computer. This coordinate gives the position of the scan-line within the scanning raster and has a word length of 18 bits with a least count equal to 2.54 microns. The most significant bit is used as a marker bit to distinguish a scan-line coordinate from a spot coordinate.

The computer controls the digitiser by transmitting to the HPD, through the Direct Data Connection, binary bit
patterns corresponding to the control instructions such as film advance, start digitising etc. Immediately after instructing the HPD to begin digitising a frame, the computer sets up the Direct Data Connection to read the digitisations directly into core storage from the HPD. In order to digitise very faint sparks, a suitable level of digitisation for these pictures has been found to be 4,000 to 5,000 digitisations per picture. Two 5,000 word storage buffers are allocated to contain the digitisations from two frames. While the contents of one buffer are being processed by the scanning program the other buffer is being filled with digitisations from the next frame. Each time the digitising of a new frame is begun the roles of the two storage buffers are interchanged. When the HPD has completed the digitising of a frame it sends an "End of File" signal to the Direct Data Connection. Fault conditions in the HPD may be detected by the program testing the input sense lines.

2. General organisation of the program

2.1 Time-sharing

To obtain good definition of the fiducial crosses (12-14 digitisations on each arm) and sparks (2-3 digitisations), a scan line spacing of 60-70 microns is required. This results in a total measuring time of 5 seconds. The complete processing of an event as far as the space reconstruction of all tracks and obtaining an approximation to the space coordinates of the scattering vertex, takes about 5-10 seconds on the IBM 709. (A precise time has not yet been determined due to the large amount of intermediate output taken for testing purposes).

On average only about one frame in two shows a good (i.e. true elastic scattering) event. As it is possible to detect the frames showing bad events or no events in much less than the 5 seconds required by the HPD to digitise a frame, it is desirable to utilise the remaining central processor time. This will be even more imperative when the processing is done on the IBM 7090, since the mechanical speed of the HPD will remain the same whilst the internal speed of the computer is six times faster, and the processing of even the good events will be completed in a time much shorter than the digitising time per frame.
The simplest method of time-sharing the central processor unit (CPU) is to keep in the 709 memory a second program which can run independently of the scanning program. The second program can then be interrupted at will whenever the HPD needs attention, or whenever digitisings are ready to be processed, and serves as a base load to use up all spare CPU time. A natural choice for the second program is SCRAP\(^5\), the geometrical optimisation and kinematics program for this experiment. SCRAP will take as input the binary output tape from the scanning program, containing reconstructed space coordinates for each good event analysed. Thus as the scanning program is producing a binary output tape of measured good events SCRAP will be reading in a binary output tape from an earlier scanning run.

2.2 Main control program

The flow diagram of the main program controlling the two time-sharing subprograms is shown in figure 4. The scanning program is referred to as CALC 1, and the kinematics program, SCRAP, is referred to as CALC 2. CALC 1 and CALC 2 can either time-share together or one of them can run independently.

Sense switch 3 controls the entry of CALC 1. If the HPD is not operational at the time of starting the run, the program can detect this from the sense input lines and will print on-line an interruption procedure to be used when the HPD is ready. It proceeds to CALC 2.

CALC 2 is controlled by sense switch 2. The program cycles once per event through the main control loop permitting both temporary and final termination. Sense switch 1 controls the final termination of both CALC 1 and CALC 2.

The programs contained within the broken-line rectangle are interrupted at the end of the measurement of the first and succeeding frames and control transfers to TRAP, the control program for data channel trap interruption, which initiates the measurement of the next frame and enters CALC 1 to scan the current frame.

There are two possible kinds of data channel interrupt. An "End of File" interrupt which occurs at the end of a measurement and a "buffer overflow" interrupt which occurs when 5,000 digitisings have been read into core storage.
An additional interrupt, the direct data interrupt, is used for communication from the HPD operator to the computer to indicate either a new roll of film has been loaded and the HPD is ready or the HPD is operational again after a breakdown.

2.3 Shared subroutines

As CALC 1 and CALC 2 are written in FORTRAN/FAP and are intended to occupy together 32,000 words of core storage it is desirable that only one copy of all the common subroutines be in core storage. However, an interrupt may occur whilst the second program is using one of these stored subroutines, and the scanning program may, after this interrupt, want to use the same subroutine. It is necessary therefore to arrange that if an interrupt occurs during execution of one of these routines the routine should be completed before the scanning program is entered after the interrupt.

The following FORTRAN II library routines have been modified to allow them to be used by two independent programs.

I/O routines: BST, EFT, RWT, IOR, IOH

Arithmetic routines: XP1, XP2, XP3, XFT, LOG, ATN, SCN, TNH, SQR

Other routines: FPT, CLOCK (CERN installation routine)

The modification involves the use of four indicator flags situated in the first four locations of COMMON storage, listed in Table 2.

<table>
<thead>
<tr>
<th>Octal location</th>
<th>Name</th>
<th>Contents</th>
<th>Purpose If Non-Zero:</th>
</tr>
</thead>
<tbody>
<tr>
<td>77461</td>
<td>DDIFLG</td>
<td>Non Zero/Zero</td>
<td>Indicates a direct data interrupt has occurred</td>
</tr>
<tr>
<td>77460</td>
<td>SUBFLG</td>
<td>Non Zero/Zero</td>
<td>Indicates a shared subroutine has been entered</td>
</tr>
<tr>
<td>77457</td>
<td>TRPFLG</td>
<td>Non Zero/Zero</td>
<td>Indicates a data channel interrupt has occurred</td>
</tr>
<tr>
<td>77456</td>
<td>FPTFLG</td>
<td>Non Zero/Zero</td>
<td>Indicates a floating point trap has occurred</td>
</tr>
</tbody>
</table>
The entry of any shared subroutine sets SUBFILG non-zero. Should an interrupt occur during execution of one of the shared subroutines, the state of SUBFILG indicates this to the interrupt routine (see next paragraph, TRAP) which can then arrange for the subroutine to be completed before attending to the interrupt condition. At the point of exit it resets SUBFILG to zero but first tests whether the flags DDIFILG and TRIFTLG are non-zero. If either flag is non-zero, showing that an interrupt had occurred during the execution of the routine, the address of the point of interrupt is set to the address of the exit point of the shared subroutine, and a transfer is made to the appropriate interrupt program.

2.4 TRAP program

The flow diagram of TRAP is shown in Figure 5. TRAP is entered either by trapping on the End of File signal at the end of a measurement or by trapping on the completion of a channel command whilst measuring — this corresponds to the allocated core storage buffer running full.

TRAP ensures that the following requirements are met:

1. That the HPD is measuring whenever possible.
2. That the last digitised frame (the current frame) is processed whilst the HPD digitises the next frame.
3. That any spare CPU time is utilised by returning to the interrupted program, CALC 2.

TRAP initially sets TRIFTLG non-zero. If TRIFTLG and SUBFILG are tested in turn to see whether any outstanding floating point trap or shared subroutine entry requires completion. These are completed if necessary and TRIFTLG is reset to zero preventing the shared subroutines re-entering again at the beginning of TRAP if used within TRAP. All registers are saved and sense switch 1 is tested for termination of the run. If termination is not required the type of trap is examined. At the end of every measurement a test is made for a blank frame; this is detected if the total number of digitisations, KDIGIT, is less than a parameter KMINID. If the frame is not blank, the storage buffer addresses are switched and the next measurement is initiated, provided the end of the roll has not been reached. CALC 1 is entered to process the current frame. Upon exit the arithmetic registers are restored, channel traps are restored and a return is made to the point of interrupt.
If a trap on buffer overflow occurs, a search is made within the last 50 digitisations for a scan line coordinate which is compared with a parameter KMINX. If it is equal to or greater than KMINX, enough of the frame has been digitised for the scanning program to attempt to process it. Otherwise the frame is rejected as un-measurable.

3. The automatic scanning program

3.1 Coordinate system

The digitisations are made in an arbitrary cartesian frame of reference defined by the HPD. The fiducial grids photographed on each view of each chamber define points and a fiducial plane whose coordinates in space are known from measurements made during the experiment. For each roll of film, measurements are made on the first frame, the calibration frame, using a digitised projector to determine the relative positions of the images of fiducial crosses and the fiducial grids of the various chambers. This information is read from cards by the program whenever a new film is mounted on the HPD.

The HPD film transport positions each frame with an uncertainty of 0.5 mm. For each frame digitised, the program searches for digitisations within certain predetermined regions for the fiducial crosses. These regions are large (5 mm × 2.5 mm) enough to contain the crosses even allowing for the uncertainty in the film positioning.

Once the positions, in the HPD reference frame, of two of the three fiducial crosses have been found, the program determines a transformation deduced from the measurements of the calibration frame, which allows it to define for each frame:

a) exactly where on the frame in HPD coordinates the chamber views lie, and

b) the transformation from HPD coordinates to space coordinates for each chamber.

The relative positions on the film of the chamber images and the fiducial crosses gradually change over a period of several hours due to instabilities in the mirror-lens system. Measurements showed that over the period of 1-2 hours represented by one roll of film these changes are smaller than the errors of measurement;
hence one calibration frame is measured for each roll of film.

3.2 Scanning criteria

An elastic scattering event is recognised by the following geometrical criteria.

1. There is only one incident particle and its track must not be inclined more than 1° with the axis of any of the beam chambers 1, 2, 3, 4. However one of the four chambers may also contain a second track. The incident track must be visible in both views of all the beam chambers.

2. The scattered particle must pass through the correct region of chamber 8 for its incident momentum; its track must not be inclined more than 10° with the chamber axis. The scattered particle track must be visible in both views of chambers 5, 6, 7, 8.

3. Chamber 9 must contain at least one recoil proton track visible in both views coming from the target volume.

4. The incident particle track, the scattered particle track and the recoil proton track must all extrapolate to a scattering vertex within the target volume.

For events satisfying these geometrical criteria, the program outputs on a binary magnetic tape space coordinates for a point on the appropriate track in each of chambers 1 to 8, two points on the track in chamber 9, and approximate space coordinates for the scattering vertex.

3.3 General flow

As a frame is stored as a one dimensional array of digitisations it is desirable to minimise any repetitive scanning of the array. Thus the logical order of operations given by the geometrical criteria has been modified to take into account the positions of chambers on the film. After scanning the region of a chamber or of a fiducial cross, the position in the array corresponding to the end of the search region is saved. These positions are subsequently used to move from chamber to chamber across the film without any further repetitive scanning.
The sequence of operations carried out for each frame showing an elastic scattering event is listed below. If at any point in this sequence the necessary criteria are not met, the frame is rejected as not containing an event of interest. An appropriate error output is made, and control passed to the second program CALC 2.

1. Search for the fiducial crosses.

2. Calculate the transformation coefficients from the calibration reference frame to that of the HPD.

3. Search for a track of a scattered particle in both views of chamber 7.

4. Search for a track of an incident particle in both views of chamber 1.

5. Extrapolate within the HPD reference system the track of the scattered particle found in chamber 7 into both views of chamber 8 and search for its continuation. (Extrapolating the particle in this direction is usually more successful than the reverse direction as the track in chamber 7 is better defined viz. 12 sparks as against 6 sparks).

6. Check that the track is in the correct region of chamber 8 for its incident momentum and that its inclination with the chamber axis is less than 10°.

7. Search for a track of an incident particle in both views of chamber 2.

8. Test if the tracks in both views of chambers 1 and 2 correspond to the same incident particle.

9. Search for a track of a recoil proton coming from the target volume in both views of chamber 9.

10. Extrapolate the track of the scattered particle found in chamber 7 into the stereo view of chamber 6 and search for its continuation. The bending magnets have negligible effect in the vertical plane.

11. Search in the normal view of chamber 6 for the correlated track of that found in the stereo view.

12. Extrapolate the track of the scattered particle found in chamber 6 into both views of chamber 5 and search for its continuation.

13. Extrapolate the track of the incident particle found in chamber 2 into the stereo view of chamber 3.
14. Search in the normal view of chamber 3 for the correlated track of that found in the stereo view.

15. Search for a track of an incident particle in both views of chamber 4.

16. Test if the tracks in both views of chambers 3 and 4 correspond to the same incident particle.

17. Reconstruct into space all recognised tracks and test that the incident particle track, the scattered particle track and the recoil proton track extrapolate to a scattering vertex within the target volume.

One point on the track in each chamber is reconstructed into space. This point is the intersection of the track with one of the fiducial planes of the chamber. The position of this fiducial plane is known in space and its image is indicated in both views by a fiducial grid. For each view the intersection of the track image with the fiducial grid line is transformed to space by linearly interpolating between the two nearest chamber fiducials. The intersections obtained from both views give two of the space coordinates; the third coordinate, that of the fiducial plane, is known.

3.4 Basic subroutines

There are four basic subroutines:

1. FIDUC A subroutine for finding a fiducial cross

2. SPARK A subroutine for finding sparks in a view

3. TRACK A subroutine for finding linear tracks from sparks within a view

4. CORREL A subroutine for correlating tracks in two stereoscopic views

3.4.1 FIDUC subroutine

The input data to this routine are the coordinates of a rectangular searching region and the tangents of the angles made by the two arms of the cross and the scan line direction. Figure 6 shows a plot of the digitisations obtained from a fiducial cross; the rectangular region represents approximately the search region used (2.5 mm × 5 mm on the film) where the long dimension is in the direction of film motion.
The subroutine searches within the boundaries of the rectangular region for those digitisations corresponding to a fiducial cross. The fiducial is recognised by the upper arms converging to within a distance of 400 microns. Once this condition is satisfied the subroutine searches for digitisations on the lower arms. The end of the search region terminates the procedure and a least-squares fit is made to the digitisations of both arms to give the intersection point.

3.4.2 SPARK subroutine

The input data to this routine are the number of gaps and the coordinate positions of each gap. The subroutine assumes that the plates of the chamber are parallel to the scan direction. SPARK searches independently within each gap to associate digitisations from one scan line to the next. Digitisations must differ by less than half a spark width (40 microns) to be associated to the same spark.

If there are at least two digitisations per spark length, a spark is recognised with coordinates of the mean of the digitisations and of the mid-point of the gap. Figures 7, 8 and 9 show plots of digitisations of tracks from figure 2.

3.4.3 TRACK subroutine

The subroutine uses the following algorithm for finding linear tracks:

1. Ignoring gaps with no sparks, select two gaps which first contain the least number of sparks and second are most distant from one another.

2. Generate all possible tracks from the combinations of the sparks from one gap with those of the other, which fulfill the conditions of track slope for the chamber. These spark combinations are termed "generating pairs".

3. For each generating pair of sparks calculate the intercept of the line joining then on the remaining gaps.

4. Within a region of one spark width (80 microns on the film) either side of the intercept select the nearest spark to the intercept for each gap.
5. Accept the track if the total number of sparks selected is greater than the threshold for the chamber (usually equal to half the total number of gaps).

6. If the track is accepted cancel the accepted sparks from each gap.

If no track is found from the generating pairs of the two selected gaps, one gap is held selected whilst the next beat possibility is chosen for the other. When all possibilities have been exhausted for the second gap, the sparks from the first gap are erased. This erasure together with the cancellation of accepted sparks ensures convergence of the procedure.

3.4.4 CORREL subroutine

It was originally planned to correlate tracks in the two stereoscopic views using the prism displacement effect. Unfortunately this feature was not very reliable due either to the non-firing of the gap containing the prism or to the displacement being outside the field of view. It was therefore decided to correlate tracks by their spark patterns and to use the displacement information only when this method failed.

The correlation of tracks by spark patterns is based on the efficiencies of the chambers. Generally when there is more than one track in a chamber not all gaps fire for all tracks. Moreover, the presence or absence of a spark in a gap is observed in both stereoscopic views. A track can be characterised by a binary pattern where each bit corresponds to a gap and a 1 represents the presence of a spark and a 0 the absence of a spark. Thus ideally for two tracks to be correlated the binary patterns will be identical. In actual practice very thin weak sparks can sometimes be digitised by the HPD in one view and not in the others. A correlation of tracks is therefore accepted if there are not more than two inconsistencies in the binary patterns.

3.5 Storage requirements

A total of 23,000 words are at present used by the scanning program, including 10,000 for the input buffers and 3,000 for library routines. With removal of various printing and debugging routines at present included, some 10,000 to 11,000 words will be available for the CALC 2 program, excluding its library routines which will be shared with those already used in the scanning program.
4. Preliminary results

Tests on the complete program have been under way since the end of June 1963, and results so far obtained are given below.

The on-line control of the HPD, and the time-sharing features of the program operate satisfactorily. Tests have been carried out on film previously measured by hand on digitised scanning tables and the results from the two methods compared. A total of 390 pictures have been digitised, known to contain 187 elastic scattering events. The program has recognised 136 i.e. 72% of these. It has therefore rejected 27% of the scattering events which it should have found. On the other hand it has not accepted any events at all which should have been rejected.

The incorrect rejections fall into three categories:

a) Errors in track recognition when two tracks are very close to one another.

b) Incorrect track extrapolation from one chamber to another.

c) Failure to correlate the correct tracks between two stereo views when there is more than one track in a chamber.

Several modifications are being made to reduce the number of these rejects.

The first two categories will be reduced shortly when the HPD track centre circuit is changed. At present if a spark is wider than 30 microns the digitising does not correspond to the centre of the spark but to a position 15 to 20 microns from one edge. Thus for a track containing both fat and thin sparks (a condition often seen) an error is introduced into the measurement of spark positions. For closely adjacent tracks this may cause the wrong sparks to be grouped into a track. For a single track an error in slope is introduced, which when magnified by extrapolation to the next chamber can cause the continuation of the track in this chamber to be completely missed.

The extrapolation of a track from a view of a chamber to the corresponding view of an adjacent chamber is made within the HPD reference frame, neglecting the small variations in chamber demagnifications. A correction to the alignment of the two chamber axes is obtained from the relative positions of chamber fiducials. A complete transformation into a common reference frame for both chambers will be made to improve track extrapolation.
A variety of possibilities also exist for logical modifications to the program which will be explored; for example the order of selection of chambers can be changed, tracks can be followed in the reverse direction, difficult track correlations can be avoided by following the correlated tracks found in an adjacent chamber into both views of the chamber, etc.

References


5. B. Zacharov, unpublished work.
Figure captions

Fig. 1  Schematic Diagram of \( \gamma \pm - p \) elastic scattering experiment.

Fig. 2  Typical picture showing positions of chamber views.

Fig. 3  Schematic Diagram of frame format.

Fig. 4  Main control Program.

Fig. 5  Data Channel Trap Program.

Fig. 6  Digitising of fiducial cross No. 2 of figure 2.

Fig. 7  Digitising of an inclined track.  View 9/1 of figure 2.

Fig. 8  Digitising of a track with a spurious spark view 7/1 of figure 2.

Fig. 9  Digitising of a track with a displaced spark view 7/2 of figure 2.
Figure 1

SC 1 - SC 9  spark chambers
C        coincidence counters
A        anticoincidence counters
Figure 4
FIDUCIAL CROSS 2

Figure 6

CHAMBER 9/1

Figure 7
Figure 8

Figure 9
DISCUSSION

TYCKO: What is the road width?

BLACKALL: The road width is about 160 microns, i.e., about twice the width of a spark.

BAKER: What changes have to be made to the HPD hardware to process spark chamber pictures and how long does it take?

POWELL: Not very much. It means taking off one film transport and putting on another, which is easy on our HPD. You then have to re-focus the track channel. The last time we did it, it was at most half a day before we were ready again.

BAKER: Do you position the film for measurement by means of the perforations?

BLACKALL: Yes.

BAKER: When you are fitting a track to the sparks, what point on the track do you take?

BLACKALL: In the direction parallel to the plate we take the average of the digitisings, and in the other direction we take the mid-point of the gap.

BAKER: Have you tried to take pictures of a general grid, and then digitise it to find out what distortions you had during the run - e.g., movements of the mirrors, etc.

MACLEOD: We measure a calibration frame once per roll and we find that the reproducibility is better than the precision of our digitisings. The fiducials are measured on the film to something like 10⁻⁶.

POWELL: Is there any danger of losing digitisations on the track due to premature rejection when the program tries an incorrect pair?

BLACKALL: A spark is rejected only after all possibilities of forming tracks with other sparks have been tried unsuccessfully.
THE NEVIS SPARK CHAMBER AUTOMATIC MEASURING MACHINE (SAMM)

M. ARM, D. BROWN, and A. JACOBSON
Columbia University, Electronics Research Laboratory, Irvington-on-Hudson
S. ADLER and D.H. TYCKO
Columbia University, Nevis Laboratories, Irvington-on-Hudson

(presented by D.H. TYCKO)

1. Introduction

We have recently completed the construction of a flying-spot digitizer to be used in the automatic analysis of spark chamber photographs. The design of the system was derived from certain general properties of spark chamber data.

First, the geometries of most conceivable spark chambers are such that the locations of the gaps relative to the fiducial positions are known and are easily represented. In the case of parallel flat plate chambers, for example, one may choose a cartesian coordinate direction perpendicular to the plates and specify the positions of the gap center lines by a table of coordinate values versus gap number. The position of a spark center is then specified by its location along the gap center line and the gap number. Thus, a measuring device need only measure distances along lines parallel to the gaps.

The second relevant property of spark chamber data is the relatively small number of spark images per frame. Representing each spark by the precise coordinate of its center (or even by the two coordinates of its ends), it is feasible to store in a computer the complete memory image of a photograph. Taking an extreme case of 200 gaps and 10 tracks, we would still require only 4000 words to store both stereo views. Therefore, it is practical to transcribe the complete images from film onto paper or magnetic tape for subsequent processing by the scanning programs.

2. Film format and machine functions

The flying-spot digitizer we have built is designed to accomplish this transcribing process. It is presently best suited.
to film formats in which all the gaps are parallel to each other. Fig. 1 is a drawing illustrating the type of film format the system requires. Black to white (or white to black) transitions on the rough grating define the vertical, or Y, positions of the gaps at which horizontal digitizing scans take place. Two modes of operation are possible. In one mode, horizontal scans are made at all lines of the rough grating with up to 100 such lines allowed. In the other mode, any 20 lines may be used, the selection being made at the console and stored in a special relay memory of the machine. Irregular spacing of these scan positions may be used. Digitization in the horizontal, or X, direction is accomplished by splitting the light beam from a CRT spot generator and scanning an optical grating and the film simultaneously. The horizontal scans proceed from left to right in the figure and the pair of fiducial lines on the left are used to reset the grating pulse counter. A delayed coincidence between the two lines is made to eliminate spurious reset pulses. These lines are not digitized. The third fiducial line on the right is digitized. The large square in the upper left hand corner defines the frame and is detected by the film advance control system.

The digitizing of a frame goes as follows. The rest position of the spot is above the rough grating. When the scan is initiated the spot unblanks and moves down to the first-selected vertical level. The vertical position is then held while a horizontal sweep proceeds. When the spot has traversed a spark image the counting is stopped. The spot completes the sweep and returns to its rest position while the position of the trailing edge of the spark, X, and the spark width, W, are read out onto paper tape. The position of the trailing edge is retained in the machine. On completion of the read out, the spot moves down to the same vertical level and makes another horizontal sweep. After the spot passes the position of the previous spark the machine is free to digitize again and records the width and position of the next spark. This process is repeated until all sparks within a preset length of scan are digitized. A gap number and end of record mark are then punched. The gap number is stepped and the same procedure is gone through on the next selected vertical level, etc. When a preset number of vertical levels have been scanned, the film advances to the next frame and the process is repeated.

3. Description of the measuring machine

Figure 2 shows the spot generator, optics and film transport section of the machine. The spot is produced by a CBS high resolution CRT. The lens assembly consists of a 7" F.L. Aero Ektar back to back with a 12" F.3. Aero Ektar. A mirror type beam splitter is shown in the picture but a pellicle beam splitter is presently being installed. The grating, located beneath the mirror, has a 100 micron pitch.
Both light-to-dark and dark-to-light transitions are counted giving a grating count every 50 microns. A keyed oscillator is used to divide each grating interval into 10 subintervals, yielding a least count of 5 microns. This system has a resolution of about 60 \( \mu \) when scanning normal spark chamber film and about 50 \( \mu \) when scanning high contrast, black background film. The precision, measured by the RMS scatter of forty digitizations along a straight line, is 20 \( \mu \). When scanning normal spark chamber film, spark images which have widths of at least 25 \( \mu \) are detected with high efficiency. The film transport is designed for 35 mm perforated film.

Figure 3 shows the electronics rack and the Teletype paper tape punch. The paper tape punch limits the system to a digitizing speed of about 80 nsec per spark image. In the near future we shall write on magnetic tape via the Nevis 1401 which is equipped with a serial I/O adapter. This will increase the speed to about 50 nsec per digitizing. The machine has been christened SIMM.

4. Programming

The system was originally intended to be used with the Nevis IBM 1620 computer. It is now clear that the Columbia 7090 will be more suitable for production scanning programs but the very accessible 1620 is extremely useful in the present testing phase of the project and a 1620 program, SPARKS, designed for this purpose is in operation.

The program is set up for a particular experiment by loading the table of \( Y \) values corresponding to the gaps and certain parameters which serve as preliminary scanning instructions to the program. These parameters now include the number of gaps in the chamber, the specification of the gaps in which tracks may or may not begin, the number of consecutive missing sparks allowed in a track, the minimum number of sparks allowed per track, the tolerance used in the track following procedure, and the limits on the fiducial coordinates.

Each record on the data tapes contains the \( X \) and \( W \) values (four and two decimal digits, respectively) for each spark in a gap and the gap number. These data constitute the contents of a Primary Gap Bank. The \( X \) coordinates of the centers of all the sparks are calculated immediately and these are stored in Secondary Gap Banks (SGB). In case a third (digitized) fiducial line is present on the left side of the frame (see Fig. 4), the spark center coordinates are referred to it before being stored in the Secondary Gap Banks.

After the data are read and the Gap Banks are filled the program sorts the sparks into tracks. Letting \( X(i, j) \) be the spark
center coordinate in the \( J^{th} \) gap associated with the \( I^{th} \) track, the general procedure used is to predict a spark center coordinate in the \((J + 1)^{th}\) gap according to the formula

\[
X_{\text{predicted}} = X(I, J) + \frac{X(I, J) - X(I, J-1)}{Y(J) - Y(J-1)} \cdot \sqrt{Y(J+1) - Y(J)}
\]

and to then search the \((J+1)^{th}\) SGB for an \( X \) which agrees with \( X_{\text{predicted}} \) within the tolerance. If no \( X \) in the \((J+1)^{th}\) SGB agrees with the prediction, a zero is placed in the \((J+1)^{th}\) field of the \( I^{th}\) Track Bank, a new prediction is made for the \((J+2)^{th}\) gap, etc. As sparks are associated with tracks they are stored in Track Banks and flagged in the Secondary Gap Banks.

Typically, we have allowed two consecutive missing sparks, have restricted initial sparks to the first two gaps, and have demanded at least three sparks before setting up a Track Bank.

SPARKS fits each track to both a straight line and a parabola and calculates the RMS scatters and \( \chi^2 \) values for each fit.

The program runs at a speed of approximately one second per track for pictures having eight sparks per track. This is not including the dumping of the Gap Banks and Track Banks onto punched cards.

A complex system of optional monitor punching and tracing is built into the program. It is controlled by Sense Switches and allows a detailed examination of the progress of a calculation if it is needed.

5. Results

The complete system, SAMM plus SPARKS, has been applied to both the artificial spark chamber picture shown in Fig. 1 and to the pictures obtained using an eight gap thin plate chamber exposed to cosmic rays. Figure 4 is a sample of the latter.

About 50 different frames of the artificial picture were processed. The spark widths average about 100\( \mu \) on film. Linagraph Shellburst film was used giving excellent contrast and the spark images were white against a dark background, contrary to actual spark chamber film. With this idealized film the system functioned perfectly, producing the five Track Banks per frame correctly. The prediction
tolerance was set at ± 150 μ. No confusions occurred at the intersection of the crossing tracks and missing sparks were no trouble.

A roll of 600 cosmic ray pictures has been processed and the detailed study of the results is now in progress. In these pictures, spark images average about 45 μ in width. Plus-X film was used to provide sensitivity to weak sparks. The film was given no special treatment as far as cleanliness was concerned and the pictures were of normal quality. Examination of the results from the first 50 frames indicates that SANN produced spurious digitizations about once per frame. This low noise level has no effect on the ability of SPARKS to follow tracks. Most of the frames contain single track events and these tracks were followed in every case. Double track events which do not exceed the resolution and sensitivity limits stated above are also successfully processed.

In those cases in which two sparks in the same gap are not resolved, an abnormally large width is recorded. This will probably be useful in interpreting events having two diverging tracks, e.g., a gamma ray conversion.

6. Plans for the future

Besides the new beam splitter and the magnetic tape output mentioned above, we are planning to replace our lens system with a Wray HPD lens and to install a CRT with better resolution. Two physics experiments are being designed at Nevis with SANN in mind and production scanning programs will be written for them.

7. Acknowledgements

We would like to thank Mr. Garry Schulze for his very able and generous help with construction and operation of SANN.

This work is supported in part by the United States Atomic Energy Commission.
Figure captions

Fig. 1 Illustration showing type of film format required by SAMM.

Fig. 2 Photograph showing the CRT, optics and film transport section of SAMM.

Fig. 3 Photograph showing electronics racks and paper tape punch of SAMM.

Fig. 4 Example of double track cosmic ray event from test film.
DISCUSSION

KASHA: What is the spot diameter on the film?

TYCKO: We scan a 100μ pitch grating and get 50% modulation.

KASHA: Do you have any difficulty in making the spot keep along the center of the gaps and not stray away from the centre.

TYCKO: We minimise the pin cushion distortion by using standard linearity correction coils made by CEICO.

KASHA: What is your demagnification factor between the chamber and the film?

TYCKO: The demagnification is about 30 and the average spark width is 45μ.

BLOCH: Do you scan white sparks on a black background or vice versa?

TYCKO: Either.
A SPARK CHAMBER FILM READING SYSTEM

W.F. BAKER

Brookhaven National Laboratory, Upton L.I.

In preparing a system for the reduction of spark chamber photographs several factors must be kept in mind which would distinguish it from a scanner for bubble chamber photographs. Firstly, spark chamber photographs contain much less information and essentially no background. Secondly, the entire spark chamber format may change from experiment to experiment due to the relatively low cost of spark chambers. These factors in conjunction with the desire to utilize as much existing equipment as possible and to obtain good resolution have led to the construction of the film reading system which will now be described.

This device digitizes the coordinates of all information which is on the film. It records this on magnetic tape which is later processed in an IBM 7094 computer where sparks and tracks are reconstructed and kinematical analysis of the event is made. To match different experiments various spot sizes can be used as well as various spacings of the scanning lines.

The flying spot scanner

The layout of the optical components of this scanner is shown isometrically in figure 1. The principal difference between this system and the Hough-Powell device for raster generation is that here a fixed illuminated pinhole is demagnified by about a factor of ten onto the film, and the notion of the lens itself produces the scan line of the spot on the film. For this purpose a microscope objective is used which has a large numerical aperture resulting in an intense spot of light. Sixteen such microscope objectives are placed around the circumference of a rotating disk with their optical axes parallel to and 145 mm from the axis of the disk. As long as an objective is passing through the cone of light emanating from the pinhole, it focuses a spot of light onto the film. The success of this system depends on the fact that the lenses used show very small aberrations for off axis rays.

There are three independent optical channels as shown in figure 1. Two provide the normal and orthogonal scans of the film; the third channel focuses onto a transmission grating which provides...
the X coordinate of the scan by digitizing the azimuthal position of the disk. A high pressure mercury arc lamp is the source for all channels. In each, the light is collected and made parallel by a 210 mm f/3.5 camera lens. These are followed by 150 mm f/2.3 lenses which converge the light onto the pinholes and demagnify the 300 micron square arc to about 200 microns which illuminates the largest size pinhole we expect to use.

The two scanning positions are located so that the microscope objective has rotated 45° from the horizontal diameter of the disk when it is passing over the film. This results in the scan lines being at an angle of 45° to the edge of the film and in their being perpendicular to each other at the two positions giving a normal and an orthogonal scan.

Y motion is obtained from a precision stage to which the film is clamped by a vacuum platen. It is digitized by a Ferranti Moiré fringe grating system. The combination of the X and Y motions produces a cycloidal trajectory of the spot on the film; see figure 15. This data is transformed into more convenient rectangular coordinates later in the computer.

Although the azimuthal positioning of the objectives on the disk is not critical, the radial position is. This adjustment is provided by mounting the objectives eccentrically on carriers which can be rotated in their mounting holes in the disk. The objectives have, in practice, been set to the same radius to ± 1.5 micron.

Whereas the X motion is measured using a different objective, the start of digitization for a particular scan must be signalled by the spot which makes the scan. To do this, just before the spot starts onto the film it is interrupted by a 45° silvered prism which deflects the light through a second ten power microscope objective which remagnifies the spot back up to the original pinhole size. The spot then passes over a slit generating the start of scan pulse. Light intensity is sensed in all channels by ten stage RCA 6199 photomultiplier tubes.

The disk is driven through a belt by a synchronous motor at 520 RPM. This generates one scan line every 7 milliseconds with the spot actually scanning for a period of 4 milliseconds.

The film transport system which is shown schematically in figure 2, handles 35 mm perforated film which is the type used by the fastest cameras suitable for spark chamber work. The supply and take up spools are buffered from the rest of the system by
vacuum columns. These spools are driven by continuously running motors which are clutched in and out on signal from pressure switches on the columns. Both spools can be driven only in one direction; the 45 cm long vacuum columns provide adequate buffering for the film motion. Between the columns the film is driven by a sprocketed drive capstan which in turn is driven by a reversible motor through an electrical brake-clutch.

The film motion sequence starts when a frame passes through the projector and registration marks are sensed by a matrix of photodiodes on the screen beyond. At this time metering of film is commenced using the rotary encoder mounted above the drive capstan. The frame is advanced a predetermined distance to the normal window where it is clamped to the stage and the normal scan of the frame is made. When that is done the film is released and advanced to the orthogonal window where it is again clamped and the orthogonal scan is made while the stage returns to its original position. The frame is then returned to the projector and the film is advanced until the next frame is registered and the process is repeated. Since it is intended to scan every frame, normal forward speed is only 20 cm/sec and reverse speed is 43 cm/sec. In the search mode forward speed is also 43 cm/sec.

The format for a typical spark chamber photograph is shown in figure 3. The two squares in the upper right hand corner form the registration mark. The coding at the bottom is the film and frame numbers in the binary coded decimal system. These are read by photodiodes, also located on the projection screen, immediately after the registration marks are sensed.

The circuitry used to sense the frame registration marks is given in figure 4. The relative positions of the photodiodes can be adjusted to match any overall magnification. The registration marks, frame numbering and fiducials are all produced by argon filled photographic flash lamps. This high light intensity is necessary because of the small lens openings and fast cycling cameras used in spark chamber work.

Tests of the film transport and metering system show that film can be stopped repeatedly to within ± 1/4 mm of the expected point. This precision is adequate for the computer analysis where the fiducial marks on the film must be located and recognized unambiguously.

Photographs of the completed film reader are figures 5 through 8.
The signals directly at the outputs of the various photomultipliers are shown by the oscilloscope traces in figure 9. The parameters of the reader for these traces were as follows: disk speed 520 RPM, all pinholes 55 microns (this yielded a measured spot size of 7.5 microns) and a stage speed which gave a 60 micron line spacing (20,200 and 500 microns are also obtainable). The X grating used had a 17 micron cycle, with the clear and opaque portion approximately equal at 8.5 microns.

The upper left hand oscilloscope pattern is the X grating signal at two different sweep speeds. The upper shows the entire passage of one spot across the grating. The individual crossings of the grating lines are not visible, but their envelope shows a light modulation of about 75%. The shrinkage of light intensity as one moves away to either side of center is caused by vignetting of the light by the microscope lenses and their holders and is not due to defocusing of the light spot. The lower trace is just a portion of the upper one with a sweep speed sufficient to show the individual crossings. The next pattern down is again the X grating but with the scans produced by all sixteen objectives showing. These first two patterns were taken with 85% of the light removed by a uniform neutral density filter placed over the photomultiplier.

To remove the effect of vignetting a nonuniform filter was placed over the photomultiplier to replace the uniform one. This was produced by placing unexposed photographic film over the tube and exposing it to the light from the wheel. With this non-uniform filter in place the X grating signal shown in the lower left of figure 9 was obtained.

In the upper right trace the start of scan pulse is shown just below the X grating pulse. The next pattern down again shows the start of scan pulse and above it the output of the photomultiplier behind the film platen when no film is in place. The last pattern was taken with film in place and contains one data point on that scan.

The data processing unit

The function of this unit is to transform the pulses from the photomultipliers into useful coordinates on magnetic tape. A photograph of the components is shown in figure 8. An abbreviated block diagram of the logic is given in figure 10.

The Moiré fringe system is permitted to start driving the Y scaler when a start of frame signal is picked up from a flasher mounted on the stage. This start signal is used to reset the
Y scaler after a delay of time adequate to complete the frame. As it is now adjusted, one Y count represents 20 microns of stage movement. Similarly the X grating is allowed to drive the X scaler after the start of scan signal has opened a gate. This start signal is also used to reset the X scaler after a time delay sufficient to complete one scan line. Initially one X count corresponds to one cycle of the grating. At present we are doubling this frequency using a time delay and an adder, and this gives a least count of 8.5 microns. The size of the least count can be reduced by at least another factor of two.

An X coordinate is recorded whenever a point is read from the film, but the Y coordinate is recorded only once at the end of these scan lines in which data have been sensed. The data pulse does not appear at the moment the spark is crossed, but instead at some constant distance beyond its center. The method by which the center of the spark is located is explained in figure 11. An independent track centering scaler is used, and when it is full the overflow pulse causes the data point to be read. It is fed at one half the normal frequency while the spot is traversing the spark and at normal frequency thereafter, thus making the distance from spark center to data point independent of spark width. There is, of course, a maximum width which can be so treated as well as a dead time which corresponds to half the width. The capacity of the track centering scaler can be adjusted to match the maximum that is expected. Should a spark width exceed the capacity of the scaler, no data are read.

Under the above conditions, the contents of the X and Y scalers are read into a 256 word core memory. This memory serves as a buffer between flying spot scanner and the tape unit enabling the reading of more closely spaced points on the film. When the memory is half full it dumps 128 words onto tape leaving the other half of the memory to accept data. The interlace network prevents simultaneous reading and writing in the memory. A read request always takes preference, and the write information is held until the reading of the word is completed.

Since each word in the memory contains 18 bits, it must be reduced to three words of 6 bits each for use with the 7094 computer. The tape buffer and ring counter do this and also manufacture parity bits as needed. The tape unit itself records 200 bits/inch at a tape speed of 150 cm/sec.

Figure 12 is the first event we attempted to scan. Only the normal window position was used with the 45° scan line running generally from lower right to upper left. In figure 13 is shown the face of the cathode ray tube on the 7094 which displayed the event after it had been scanned and transformed from the cycloidal system into the rectangular coordinate system.
The computer programs

A program, called STEAM, has been written for the IBM 7094 computer to analyse events in a cylindrical spark chamber. The flow chart for this is shown in figure 14. The times given are for the processing of both the normal and orthogonal scans of one frame, each containing 2,000 data points. This program specifically analyses K + p elastic scatterings to obtain the differential cross section as a function of scattering angle.

One file on the tape, which corresponds to the normal and orthogonal scans of one frame on the film, is read into the core memory of the 7094 computer. The first thing that must be done before these two scans can be superimposed in the same coordinate system is to locate the fiducial crosses on both scans.

Figure 15 shows diagrammatically how this is done. The areas containing the fiducial data are approximately known since the position of the frames on the vacuum platen repeats within a few tenths of a millimeter. A binary search is made of the memory to locate the data in these areas which are candidates for fiducials. In one of these areas the smallest value of Y is sought along which scan there are two data points, that is X values. The next larger value of Y is sought again having two data points with the requirement that $\Delta X$ for these be less than the $\Delta X$ for the previous scan. Continuing this process, pairs of data points are accepted until the crossover point is reached after which it is required that $\Delta X$ for the two points increase with each succeeding scan line. When all data arising from this selection process have been obtained, those points read by the scanner first on each scan line while $\Delta X$ is decreasing are matched with those read second while $\Delta X$ is increasing. Likewise the points read second before the crossover are matched with those read first after the crossover.

These two groups of data are now transformed from the cycloidal coordinate system of the scanner to rectangular coordinates. With this done, straight line least squares fits are made to the two sets and the intersection of the two lines is the fiducial point. This process is repeated for each of the four fiducial marks in both scans.

The sequence then proceeds into the main program where the data from both the normal and the orthogonal scans are unpacked to give one coordinate per 36 bit word. The data are then transformed from the cycloidal into a rectangular coordinate system. This transformation is performed through a combination of table look up and calculation so as to minimize the time needed. As indicated in the flow chart of figure 14, this transformation requires approximately 0.15 seconds for 2,000 data points.
In the routine TRFARM the data from the orthogonal scan is translated and rotated so that the fiducial points in both views coincide. It is required that the matrix of this transformation be a simple translation plus a rotation, otherwise one of the scans is stretched or shrunk and the event is rejected. All the data is now known with respect to the center of the spark chamber.

The spark chamber is divided into 128 sectors which, since there are 10 gaps, yield 1280 areas into which the data can be located. This is shown in figure 16. All data that do not lie in these sectors cannot be candidates for sparks. The size of the sectors was chosen so that ordinarily not more than one spark lies in a sector and so that one spark will usually lie entirely within one sector. SPRT classifies each datum according to its gap and sector with the result that an ordered table is obtained having for each segment of area the number of data points in it, the sum of the X and the sum of the Y coordinates of those points in the rectangular system, and the first and last points picked up by the scanner in the area.

Upon entering the TRACK routine a table, SPARK, is set up containing in each grouping, information relating to one spark, such as the first and last points on a spark, the centroid of the spark, and some identification. A spark must be entirely within one or two area segments, and it is assumed that all information in one area belongs to one spark. Assignment of data to specific sparks is shown in figure 16. The first point in one sector must be within some predetermined distance from the last point in the adjacent sector of the same gap for the two sets of data to be assigned to one spark. It is also required that there be between two and fifteen points for a spark.

Now that the SPARK table has been set up, these sparks are assigned to tracks before a least squares fit is made to their centroids to find the equations of the tracks. As this assignment is made, the sparks are ordered in a new table, called A, in the same sequence as they occur along the track.

Starting with some spark, the computer examines the adjacent gap in the same sector and in the two adjoining sectors. For each spark so found the minimum distance, d, between either of its end points and either of the original spark's end points is found. For a spark to be a candidate, d must be less than some preselected value, D. Of the sparks found in the adjacent gap, that one having the smallest d is assigned to the track. A similar search is then made into the third gap, if an acceptable candidate is found the track is started; if not, the computer returns to the second best spark in the second gap and seeks a compatible spark in the third gap.
With the track thus started, the sequence is continued through succeeding gaps, with the modification that the most acceptable next spark is that whose minimum distance, \( d \), is nearest to the \( d \) of the preceding spark and not necessarily the minimum. It is also required that changes in sector number along the track be monotonic.

In the event that a gap of the spark chamber has failed to fire, a search is made into the next gap, with the requirement that \( d \) now be less than \( D + g \) where \( g \) is the gap width. Similarly, if two successive gaps fail to fire, \( d \) must be less than \( D + 2g \). If more than two successive gaps fail, the track is assumed to have terminated.

With Table A set up, all sparks have been assigned to tracks or have been rejected for assignment. It is now required that there be six tracks as are obtained from an elastic scattering event. A stray background particle passing through the chamber would yield four more tracks which, although not permitted at present, would not unduly complicate the programming. It is also required that there be at least \( n \) sparks in the track, where \( n \) is some number between 6 and 10, inclusive.

Straight line least squares fits are now made to the centroids of the assigned sparks resulting in six equations of the form \( ax + by + c = 0 \).

The direct images are distinguished from the reflected images by the fact that the former must intersect in a common vertex as do the tracks of the scattering particles in real space. The number of combinations of tracks which must be taken to make this identification is reduced by the fact that the reflected images are always located either clockwise or counterclockwise from the direct images. The track of the incoming particle is known from the layout of the experiment.

The equations for the tracks are next fed into the kinematical analysis program for \( K + p \rightarrow K + p \) elastic scattering. Here the two possible identifications of the outgoing tracks are tested and if one combination satisfies the constraints for kaon-nucleon elastic scattering, within an allowable error, the event is accepted and the desired quantities therefrom are printed out.

One notices that throughout the program whenever one criterion is not satisfied, the event is discarded. This is permissible since only one particular type of event is being sought.
The program in its entirely has been tested with hand digitized events from the K + p elastic scattering experiment.

Acknowledgments

Many people at Brookhaven contributed to this work, and it is a pleasure to thank them all; unfortunately space does not permit acknowledging them individually. In particular the contributions of Dr. A.L. Read in the programming problems, G. Schwender in electronics and control systems and J.A.G. Russell in the early phases of the project were essential.
Figure captions

Fig. 1 Film reader optical system. Relative positions of components have been distorted to show all elements.

Fig. 2 Film transport system. In reality the normal and orthogonal windows are the same size and are 45° above the horizontal diameter.

Fig. 3 Typical spark chamber event showing coding method. Registration mark is in upper right corner with frame number at the bottom. K meson entering from the top is scattered in liquid hydrogen contained within fiducial crosses. The scattered meson and recoil proton are both visible. A secondary image of each spark located counterclockwise from the original gives depth of spark in chamber. This photograph was made in the ten gap cylindrical chamber made by T.F. Kycia and K.F. Riley at Brookhaven.

Fig. 4 Circuitry for sensing frame registration marks.

Fig. 5 Overall view of flying spot scanner. Mercury lamp housing is to the right, rotating disk is in the center and the film transport system is to the left.

Fig. 6 Close-up of platen area showing microscope objectives mounted in wheel. Pinholes are mounted on three mutually perpendicular slides to the right. 45° prism for start of scan is mounted on renormifying lens.

Fig. 7 Film transport system. Drive capstan and metering encoder are mounted just to right of the projector. Photodiode matrices for registration marks and frame numbering are rock in this photograph.

Fig. 8 Rack containing data processing unit. To the left is the tape transport and to the right the bins containing the logic components and the core memory. Controls for the scanning table are to the far right.

Fig. 9 Signals from photomultipliers. See text for descriptions of individual traces.
Fig. 10  Block diagram of data processing unit. For the X count signal from the grating to drive the X scaler a start of scan signal is required. The start of scan signal is also used to reset the X scaler after a delay of time sufficient to complete the scan.

Fig. 11  Method of finding the center of a spark along the scan line.

Fig. 12  Photograph of a K + p scattering in a cylindrical spark chamber.

Fig. 13  Photograph of 7094 cathode ray tube after scanning the above frame. Only the normal scan was made giving a poor fidelity for sparks oriented along the scan line.

Fig. 14  Flow chart for analysis of cylindrical spark chamber photographs. Times given are for a normal and orthogonal scan of 2,000 data points. The name of the programmer of each routine is given. Data tape is normally read on a separate data channel while the final analysis is being executed.

Fig. 15  Reading sequence of fiducial marks.

Fig. 16  Track reconstruction. At the left, data points within area segments are shown with end points that are scanned first and last encircled, and spark centroids shown by crosses. In the center assignment of sparks to tracks is made, and on the right a straight track is fitted to spark centroids.
Figure 10

Figure 11
Figure 15

Figure 16
DISCUSSION

BLOCH: We were wondering whether such a device could be used for bubble chamber work where the precision requirements are much greater. Could you repeat the parameters of your wheel and especially what is the precision of alignment of the microscope objectives. Do you have any trouble due to centrifugal forces?

BAKER: The rotating disk contains 16 microscope objectives mounted on a circle of 14.5 cm radius. Focusing and radial alignment tests were made with the disk at operating speed. Radial alignment was made possible by mounting the objectives eccentrically in carriers which could be rotated in their mounting holes in the disk. With 75 μ eccentricity in the carriers, alignment was possible to ± 1.5μ. The force on the objectives is less than twice gravity.

We are using an X grating with 17μ pitch. Earlier tests with a larger pitch grating gave essentially 100% modulation.

LEBOY: Could you speed up the measuring process by doing the orthogonal scan on a successive view instead of going back and forth on the film transport? You have to measure the fiducials for each scan anyway.

BAKER: Yes, but this requires some sort of memory of which frame is where.

LEBOY: You put it on tape anyway.

BAKER: Then you would have to sense the frame number at the platens.
SASS

D. HALL

Lawrence Radiation Laboratory, Berkeley

SASS stands for Spark Chamber Automatic Scanning System. This system is conceived as a general purpose spark chamber film reading and data reduction system, and at the present time is still in the planning stage. The purpose of this report is to indicate the current plans for SASS at Berkeley, and is based on a memo from Leroy Kerth and Dennis Keefe to a group of interested physicists, data processors, and engineers.

The general approach is as follows: a spot of light can be positioned on a CRT in any one of 4096 positions in x and 4096 positions in y. The light passes through the film and a field lens onto a photomultiplier which returns a signal to the computer when a differential signal is detected.

There are five basic components of the system:

1. The computer

The computer will consist of a DDP-24 which has been ordered from the Computer Control Company. It will include the following general specifications.

   i) Three magnetic tape drives
   ii) A parallel input/output channel
   iii) A direct memory access channel
   iv) A typewriter
   v) Paper tape input/output
   vi) A 10 microsecond cycle time

This computer was chosen as the best overall choice of currently available computers which could easily be adapted to connection with a CRT programmed spot.

2. The precision CRT

No firm decision has yet been made on a CRT tube; however, the following general specifications will apply:
a) The positioning accuracy must be good to 1 part in 4096 in both x and y.

b) This accuracy must be reproducible over a substantial length of time. (On the order of 30 minutes).

At the present time, Ferranti is the most probable choice for a tube.

3. The controller

Because much of the spot display takes place in an ordered but time consuming way, it is planned to build a separate logic unit called the controller which carries out the iterative features of the display and interrupts the computer only for new starting conditions or when something of interest has been found.

The controller will consist of these parts:

a) The CRT plotting section

This section will consist of 5 registers as follows:

i) x register 22 bits
ii) y register 22 bits
iii) Δx register 13 bits plus sign
iv) Δy register 13 bits plus sign
v) n register 12 bits

The computer will send this information to the controller which will then function as follows: The 12 most significant bits of x and y are plotted and n is tested for zero. If n is not zero, x and y are modified by adding x and y to the least significant 13 bits, and n is reduced by 1. When n has been reduced to zero, the computer is interrupted.

The controller will be able to accept any number of words up to 5 so that one could make use of the precision CRT as a point plotter by simply setting n = 0, initially and sending only x and y subsequently.

b) The photomultiplier logic

No method has yet been finally decided for detecting sparks. One possibility would be a differential technique based on the output of the n\textsuperscript{th} and the (n-1)\textsuperscript{th} photomultiplier signals.
In any case, when a change is detected, both x and y will be sent to the computer through the DMA with a tag bit to indicate whether the transition was from black to white or from white to black.

c) Control of the controller

There will be at least two and probably several modes of operation of the controller:

i) Plot until \( n = 0 \)
ii) Plot until \( n = 0 \) or until a signal is detected.

Control will be through the computer probably as an OCP (output control pulse) command. The address of this instruction indicates the destination of the signal. One possibility here would be to have the signal indicate to the controller that the next word from the DMA would be an instruction. Thus we would be able to have a full 24-bit instruction in the controller.

4. The film transport mechanism

A film transport for 35 mm film has been produced for Bruce McCormick at the University of Illinois by Flight Research Company. The film is advanced in about 100 ms and located by pin registration to better than .001 inches. At the present time, this film transport mechanism is the most probable choice.

5. The software

No planning has yet been started on the SASS program; however, the following two requirements seem almost axiomatic.

a) Special functions must be written as subroutines to allow each to be called separately. A partial list of these functions follows.

i) Fiducial searching
ii) Scan line adjustment for film position
iii) Searching gaps for sparks
iv) Following sparks

b) The program must be easily adaptable to new scanning criteria. A typical scanning procedure might proceed as follows:
I) Find the fiducials and adjust for film movement with respect to the known gap positions.

II) Search critical areas for sparks. If there are not enough, or if there are too many sparks, skip to the next event. If the correct number of sparks are present, localize the search region and begin track following.

III) If there are gaps with missing sparks, it may be desirable to re-examine those gaps with a different intensity spot since this kind of information may be useful for matching stereo views.

IV) Match the images of the tracks in each of the stereo views.

V) Fit straight lines to the tracks.

VI) Determine whether or not the event satisfies the scanning conditions. For example, in a search for elastic scatters, if track A dips down, track B should dip up, or tracks A and B in plain view should straddle the incoming track.
DISCUSSION

TYCKO: Is it not a better idea, for spark chamber work, to just create a computer image of the film rather than using a programmed spot approach?

HALL: You are asking the question of tricky hardware versus tricky programming. I think it makes the programming slightly easier with the programmed spot or at least faster. The point is that we are aiming for a time of 1.5 seconds per picture for any type of picture.

TYCKO: Time is not at all independent of the picture.

HALL: I know that is true. Whereas for the Luciole type device it is constant. We felt that the most general approach would be as I described. This would help us, for example, if we wanted to search for sparks at fairly steep angles to the scan as in the pictures that Baker had.
A MATHEMATICIAN LOOKS AT BUBBLE AND SPARK CHAMBER DATA PROCESSING

J.W. CALKIN
Brookhaven National Laboratory, Upton, L.I.

I would like to begin with my own definition of the area of discussion. I do this not with the presumption to inform or instruct you in an area where you are all far more expert than I, but simply to make clear the framework within which I speak. It lies between two limits, HIGH ENERGY ACCELERATORS and UNDERSTANDING PHYSICS. Moreover, insofar as we here are concerned, our information from accelerators comes to us in the form of bubble chamber and spark chamber film. The study of the physical universe around us runs through cycles in which there are periods during which the primary task is the accurate accumulation and recording of observed physical data, and then periods of almost blinding theoretical "insight" during which these accumulations of information fall into a mathematical formulation which is - if nothing else - in some obscure way, spiritually comforting. In a number of areas these periods of course overlap. Nevertheless, I take it as given, that in the area of interest to this meeting, we must all agree that in this decade, however humiliating and frustrating it may be, we find ourselves on the whole in a period of observation and collection of data, as opposed to one of deep theoretical insight. There is of course a merging here; we interpret our physical observations in terms of current theoretical understanding, and arrive at the same sort of contradictions which our predecessors have encountered, survived and overcome. Let us hope we may do as well. If what I have said makes sense so far, then our objective is clear: to preserve within the framework of our own limited understanding, the observations which we have made and our interpretation of them.

Within the context of this view, therefore, I have been very happy at this meeting to have heard several times the phrase "to do physics" used. That is our objective, and at previous meetings I have sometimes felt that the means to this end had in fact become an end in itself. For whatever it is worth, I must admit that all previous meetings I have attended have been in the United States. Whether the change which I believe I detect is more space-dependent than time-dependent, or vice versa, I cannot judge. Certainly, in any case, it is salutary.
Now let me get down to specifics, but always with the above broad objectives in mind. For the purposes of discussion, I will restrict my comments to bubble chambers as distinguished from spark chambers, but I assure you without prejudice in one direction or the other. Most of what I say applies equally to either, with qualifying clauses which any of you can easily supply. There is first, the simple acquisition of data, which in its unpleasantly raw form means simply the recording of the information on film in numerical form. There is second, the measure of the information which we are sure is irrelevant and uninteresting — to us or to our successors. There is next the translation of what we believe to be relevant information into the Cartesian-Newtonian-Einsteinian language in which we speak scientifically, the identification of those events which we believe we understand in terms of current theory and the separation of those which we do not so understand with a clear statement to that effect.

Having said this, let me now come to something which all of you know. We go through the above procedure with a number of systems, and we plan to go through them in the future with more sophisticated systems. One hope is of course increasingly to cut down the amount of human intervention and so, among other things, to create more technological unemployment. The more important hope is to save time and (possibly?) money. But, whatever is used now, or may be used in the future at the front end of one or another system, it is abundantly clear that at the other end must be a general-purpose computer, and it is, in fact, my own experience in this field that gives me the right — however questionable — to appear before you at all.

I will return to the use of this device, but I cannot conveniently discuss it without first talking about the special-purpose hardware at the front end of the system. In this area, the three devices which have been mentioned here are PEPR, SMP, and HPD or FSD, as you prefer. There are undoubtedly other approaches, certainly at least one which is well under way of which we all know — the machine being constructed at the University of Illinois by Professor McCormick. With respect to PEPR, as well as with respect to the Illinois machine, although either or both, in their completed state, will certainly change our views drastically, they seem on the basis of my own weighted synthesis of current opinion — in an area where I have necessarily to rely, on the judgement of others — considerably farther from completion than the admittedly simpler devices, SMP and FSD. Therefore, I here restrict my attention to these two.

With respect to the FSD, I think it is not only fair, but necessary, to separate it into two phases. The first is the phase in
which it requires the preliminary use of rough digitizing equipment which I shall call FSD-RD, and the second is that in which the rough digitizing equipment with the associated human operators is replaced by computer programs - or possibly in the longer run by computer programs and further special-purpose hardware - to do what is somewhat loosely called pattern recognition. This phase I shall call FSD-PR. (I would like to point out that the PR here stands for pattern recognition, and not Pasta-Rabinowitz, to the exclusion of Dr. Marr, who after the initiation of this work in the summer of 1961 by Pasta, joined forces with Rabinowitz in the intensive study of this problem. I say this of course because the project on which they are working has come, in some circles, to be known by the initials FR). Marr and Rabinowitz have achieved, and with the help of others in the Applied Mathematics Department at Brookhaven, are achieving results in which I, as Chairman of that Department, take great pride - both personally and professionally.

Let me now return to FSD-RD - that is the first phase. It seems to me that the SMP is very probably highly competitive with this - to the extent in fact that if there were no hope for FSD-PR, it might be very hard to justify the construction of FSD's. I have to qualify this statement with the observation that it is based on inadequate statistics, but it is my personal best judgment on the strength of available information, and also that of a number of others much more highly qualified than I, all of whom are well-known to you.

Let me now turn to FSD-PR. The great argument for this system is of course the elimination of human intervention. The importance of this varies from place to place and is, in effect, a function of the available labor market. I take it, nevertheless, that on the whole we regard it as desirable, uncertainties in the labor supply for such work being what they are. In fairness, I should add that our motivations in pursuing this work at Brookhaven were mixed - certainly, anyway, insofar as I and others in the Mathematics Department at Brookhaven are concerned. The terms of reference under which we operate include a directive toward development of new uses of computers. One such use is in the area of pattern recognition generally, and while there are a number of "blue sky" projects in that domain, I have always personally felt that a more fruitful approach is through specific attacks on concrete problems. This work of Marr and Rabinowitz represents one such attack, and it will have - in fact, is already beginning to have repercussions in other and quite disparate directions. As a matter of historical fact, furthermore, this study was undertaken long after the decision to build FSD's in a number of places was taken.

Let me now come to the matter of FSD-PR as distinguished from
the SMP approach. It has been argued by some that the latter is
still superior; the argument is based on dollar costs. 'I regard it
as fallacious, if for no other reason than that it is based necessarily
on the present state of computer technology, and this is advancing so
rapidly as to render such comparisons irrelevant. What to me is rele-
vant is that these two systems will, with no doubt whatsoever, be
found to complement one another. This applies to FSD-RD as well as to
FSD-FR. For it seems true, without question, that either of the latter
will for a long time to come have non-trivial rejection rates, no matter
what set of spatial reconstruction and kinematic (more properly "dynamic")
fitting programs is used in conjunction with them. Here I remind you
that when one speaks of processing hundreds of thousands, or millions
of events per year, a rejection rate measured even per mil may be
non-trivial. And to my knowledge, there just does not exist, nor has
anyone proposed, a better device than the SMP for handling these re-
jections. It has also to be said that when the number of "interesting"
events per one hundred trials is "small", the SMP on a small computer
with appropriate interrupt features has obvious economic advantages.

Let me next turn to one other aspect of this matter, and only
to state an opinion about a subject that has already been discussed
here. I firmly believe that it makes much more sense to attach either
an FSD or an SMP to a small or medium size fixed point computer than
to a large general-purpose computer. There are a number of the former
available which are certainly adequate for the filtering programs used
with either system. One could certainly also do a part of the kinema-
tic fitting on such a machine. The information from these programs
could be transmitted to magnetic tape, or to a disc or drum shared with
the larger general-purpose computer which is of course necessary for
the kinematic fitting and abstracting and editing programs. As has
already been pointed out here, this has great advantages with respect
to the FSD; it has similar advantages with respect to the SMP.

Now let me turn in conclusion to some remarks on the kinematic
fitting programs. When I first came into contact with this subject some
two and a half years ago, I subscribed to, and publicly stated, two
views (somewhat embarrassingly, to some extent in writing) which I
herewith publicly renounce. One was to the effect that we should settle
down on one set of "universal" programs, and the other was to the effect
that programming of this kind should be done in machine language. Both
of these views were, as all of you will realize, in the interests of
efficiency, which I have now come to regard as to some extent a false
god. In according it the respect which I did, I ignored certain facts
of primary importance. The first is that any set of programs of this
order of complexity, written completely in machine language, will be
of necessity, to a very unpleasant extent, a tremendous black box to the

7184/ah
physicist who has to use it. Second, and partially as a corollary of the above, it is extremely awkward and difficult to modify such a set of programs, as new experience dictates. Third, it seems to me now virtually certain that no such thing as a "universal" set of programs will ever exist. In contradistinction to my previously expressed views, therefore, I have now to say that I regard it as of transcendent importance that the physicist who is concerned with the analysis of experiments be able to use whatever set of programs he prefers and feels most comfortable with, and that these programs be easy to cut into and modify. To that end, I think they should be written for the most part in FORTRAN or ALGOL — which is chosen will undoubtedly be more geography-dependent than anything else.

There is after all no intrinsic reason why the FSD or SMP, or any other device, should be tied to one or another set of subsequent programs, and I think it is part of our job in this field to provide such program interfaces as are necessary between whatever hardware is used, and the various programs that are popular.
DISCUSSION

HOUGH: One thing on Fortran programs that I am very strongly for is that they become machine independent. I think we have a serious situation at Brookhaven in that we have an IBM machine which we may wish to change for a CDC machine and we are faced with a great many PAP programs. For instance: the FOG-CLOUDY-FaIR system is not machine independent. Some of the KICK programs, particularly those written at Brookhaven, and now in the Alvarez group too, are very largely in Fortran. The main point for us as users is to have machine independent programs, if we want to take advantage of advances in the field. My other point is in favour of the small computer connected to a flying spot digitizer. It's the only way of being able to use FSD's around the clock. Otherwise, we will have two very large quite expensive FSD's sitting there; the sum of which can be used at most for one shift on the 7094 and this I think over a range of years would be intolerable.

CALKIN: The machines one might consider are the SDS 9300, the PDP-6, or the stripped down CDC 3600, there are a number of possibilities.

HOUGH: The barrier there is of course programming.

CALKIN: This comes back to the question of just how much of the FSD programs one tries to run on the small machine and this will necessarily be machine dependent, but it is not an insoluble problem and it is a direction in which I think we should go. I think it's better to do it than to become dependent on one manufacturer.

NIEDERER: With respect to this Fortran type of language, it really hasn't settled down. To a certain extent, you are going to be devoting a lot of time to changing compilers so that the language stays stable. For instance, you can't go between the 7040 and the 7090 because most installations haven't made up their minds whether to use Fortran II or Fortran IV.

CALKIN: I think that there will be any number of different machines each with its own Fortran. All of these Fortrans will have a certain amount in common. What we have to hope for is that the common part gets big and the remaining parts get small. There will never be a universal Fortran simply because different machines have different capabilities and different features. The 7040 has much less power than the 7090, so if you try to write a Fortran which is completely identical for both machines, you won't use the full capabilities of the 7090.
MACLEOD: In this matter of small computers I quite agree that for small institutes, universities and regional laboratories that the attachment of things like FSD's onto small computers looks as if it's going to be a necessary development over the next few years. But for large institutes, where within the next say three years one will begin to install large machines designed from a software point of view for extensive time sharing, do you still think that the advantages of a separate computer for on-line operation of this kind of device will still hold?

CALKIN: I think that what I said will remain true, but there will be a technological change. The CDC 6600, for example is really eleven computers. That is a big central computer with ten peripheral computers. One might have a small computer with a shared disk or a shared drum with one of these peripheral computers on the 6600. One might attach an HPD or an SMP in this way but I don't think we will get there in a hurry because of the difficulty of this multi-programming problem for a very diversified institution, like Brookhaven. Whether CDC will provide an adequate executive program to satisfy everyone - anymore than IBM now supplies a monitor that does, I doubt very much.

MACLEOD: Nevertheless everyone uses the IBM monitor and makes their own alterations. Either the computer manufacturer or the individual users have to lay down certain rules by which the game can be played; this doesn't satisfy everybody but one manages without a great deal of inconvenience.

ROUGH: My main doubt is how real is time sharing from the programming point of view. What is the reality status of executive programs that would allow time sharing?

MACLEOD: I think from what we know of the CDC 6600 and the Atlas system, it should be practically feasible in about 3 years time from the hardware point of view to handle quite efficiently the miscellany of jobs which places like Brookhaven have as their general background computing load. From the programming point of view, it doesn't seem to be impossible, provided you have a machine with 65 K or 128 K store. One could time share even with things like HPD which use programs of 32 K.

CALKIN: I think we shall learn a lot when Livermore gets its CDC 6600 in February 1964. They do fairly diversified work there and plan to use the machine on a time shared basis.
HALL: I would like to point out, that neither the SMP nor the HAZE programs could be written in Fortran, at least Fortran as it exists now. FOG, CLOUDY, FAIR could be.

MOORHEAD: I believe that one could have written HAZE in Fortran except for some sub-routines such as GATE.

SCHIFF: I think the separation between the small and large computer should be made before FILTER. It is true that you do not need floating point functions, but it is extremely useful to have a lot of memory, especially because it allows a nice transition to eliminating the reads—putting the whole picture on core memory. It seems to me that a good way of putting data into a computer is through magnetic tape. You can put in about 100,000 digitings per second if you use 3 magnetic tapes and that is easy time sharing. If you have experts who not only understand time sharing but also do some of it, you should use them to solve some problems related to physics.

CALKIN: There is another undefined term here and that is the word "small". I was thinking of something of the order of a 16000 word 24 bit memory. I don’t see why FILTER can’t be done in that size machine.

BLOCH: I think in a large laboratory which would acquire a very large computer there would be some spatial considerations. It is quite true if all the HPD’s or all the hardware could be located right on the premises of the large computer, then maybe no small computer would be needed because you can transmit the data at a high rate over a distance of a few meters. I believe much work is going on for transmission of data at a very high rate – but I don’t know how successful it is. For many experiments the hardware would be located far from the main computer and unless reliable high speed transmission were available one would need small computers locally.

EDMONDS: We are already faced by this problem of time sharing. The London University Atlas will be working fairly early next year. When you talk of time sharing there are two aspects. The Atlas, in addition to being a computer in which you can time share different programs, is rather like 7 1401’s and 7 7090’s rolled into one, so that in addition you are time sharing input-output operations and computing, so we will be faced with making decisions about a small computer, which only exists in a notional sense inside Atlas. We will be interrupting Atlas operation to carry out fairly simple procedures using very much the same philosophy as Bloch and Schiff. Inside Atlas, we will take this information we have processed in a simple way and push it away.
into the drums or into some part of the core store and then onto magnetic tape, until in a main control program we can process it in more detail later. Now one can take a pessimistic point of view and say that with a big computer like the Atlas or 6600, one is using time sharing in a difficult and rather complicated way which will introduce enormous problems, or one can take the optimistic point of view that life is made very much easier because of time sharing, because we don't need to worry, for example, about what is happening while we are shifting the film on; somebody's program will come in automatically.

BURRETT: People at our laboratory have succeeded in writing a Fortran compiler in Fortran and that seems to me a good deal harder than writing HAZE in Fortran. Moreover, on the 7090 it compiles from Fortran into our type of machine code faster than the Fortran compiler compiles the same Fortran into IBM machine code.

BUDDE: I just wish to give some figures which may be helpful in the arguments about economics. It takes about equal amounts of money to take a picture and to analyse an event. Therefore how much you have to care about economy will depend on whether you analyse 1 event per picture, or whether you analyse about 1 event in 10 pictures which is more usual at the moment. Therefore I think for still some time we can allow ourselves the luxury of not caring too much about economy over the analysis and in this way we know that physics comes out better and very often more rapidly.

CALKIN: It seems to me that there is a number $N$ of events of interest per number of film frames where an SMT, for example, becomes a much more reasonable machine to use than an FSD. I don't know the value of this number and only experience over the next couple of years perhaps will determine it.
CURRENT STATUS OF THE DIGITAL PATTERN RECOGNITION
PROGRAM AT BNL. *

by

W.J. BEARD, R.B. MARR, G. RABINOWITZ and B. WENESER
Brookhaven National Laboratory, Upton, L.I.

ABSTRACT

A software approach to the problem of automatic scanning
of bubble chamber photographs, starting with raw data from the
Hough-Powell flying spot digitizer, is described. The programs
currently being developed deal exclusively with the processing of
a single view, and are divided into three distinct phases. The
first, or track reconstruction phase, operates in real time during
the HPD scan and attempts to reduce the mass of data to a list of
parameters pertaining to complete tracks. The second phase checks
the validity of the first phase output, and finishes the recon-
struction of complete tracks from any partial segments which may
have been produced. The final vertex finding phase, uses the com-
pleted track information in an attempt to list all the possible
event candidates recognizable from a single view. The organization
and performance capabilities of the track reconstruction phase,
which is in the most advanced stage of development, are discussed
in some detail, and the second and third phases are treated in a
more qualitative manner.

+ A transcription of the talk given by Rabinowitz is reproduced
  in Appendix I.

* Work performed under the auspices of the United States Atomic
  Energy Commission.
First of all, let me make it clear that the remarks I will have to make about the PEFR system are those of a spectator. Although I have followed the progress of the PEFR system fairly carefully, I have contributed in no way to the PEFR development effort. What I have to report is based on conversations with Irwin Pless (MIT), Horace Taft (Yale) and Margaret Alston, Art Rosenfeld, and Frank Solnitzi at Berkeley. One final opening remark I have been requested by the above people to make it clear that the PEFR system is not "nearly in production". In short, both the hardware and the software are in an early stage of development, compared to some of the systems described at this meeting.

I will describe some of the properties of the hardware that are relevant to the programming; I will outline the proposed program structure for PEFR, and I will discuss in more detail the programming problems which are currently in the most advanced state of development.

The PEFR hardware itself is in three sections. There is a high precision CRT which scans the film, there is a "controller" which governs the detailed behaviour of the CRT scan, and there is the computer which sends commands to the controller and accepts data back from the controller for processing.

The high precision CRT tube which is currently being used at MIT is a 5" Dumont tube with a 25 µ spot size. (There is a better tube at MIT, but the Dumont is being used because it is commercially available, complete with deflection system). In addition to the usual deflection system, there is a biquadrupole "focusing" system which is capable of causing the normal CRT spot to be extended to a line element with a length of up to 2.8 mm, oriented in any direction. The CRT is being used with a minimum of film transport equipment in as much as the current stage of development requires only grids of lines for testing.

The controller causes the spot or line to be deflected in either of two directions (active or inactive scan, corresponding to normal-abnormal scans of HPD) over a distance of up to 2 mm. In the course of the sweep, up to 6 line elements may be recorded and stored.
(If there are more than 6, a second sweep initiated by the computer can pick up the remainder). There are 100 least counts of 18μ along this sweep, and the sweep pattern can be positioned in increments of 25μ. (CRT positioning is stable to about 5μ over a period of hours). The shape of a track pulse with a slit scan is fairly well determined and the strobe circuit takes advantage of this by specifying the shape of accepted pulses on a point by point basis. Presently three points are used, that is signals from delay lines are compared and a "strobe" occurs when the amplitude at the center of a pulse is more than twice the magnitude of the amplitudes on the edges of the pulse. This method is insensitive to light level variations over a range of about 10. It is proposed that 5 pulse levels be used to further specify the pulse shape in a later version of this circuit. The present circuit will detect a test line oriented ± 30° to the scan line and will detect 25μ lines separated by 25μ. Only line patterns have (officially) been looked at so far, track studies are scheduled for September.

The controller receives (as an instruction from the computer) the (x,y) of the center of the scan pattern, the initial and final angles of the scan (incremented in 1° intervals) and the initial and final count to designate the part of the scan during which the pulse finder circuit will be gated on. The controller returns to the computer information as to the position of up to 6 pulses for each line sweep. The computer is a 12000 word PDP-1, built by DEC, who also fabricated the controller.

The program structure for the PEPR system is as follows:

In the PDP1 -

1) Area scan

look at the entire area of the film in 2 cm squares trying to identify line elements and connect them together to form track segments.

2) Track following

using area scan information, clean up track connections, etc, using techniques which might be too time consuming for area scan.

3) Precision encode

using sagitizing gratings and a spot scan, determine high precision points on track segments.
4) Output results onto tape
angles at ends, end points, length, precision points, fiducials.

In a 7094 (or equivalent) -

5) Associate tracks in all views (TRIAD)
match vertices, match tracks, organize output at a rate of ~2000 words/frame.

6) Scan
selectively produce event types for processing by PACKAGE, etc.

Of these sections, only numbers 1 and 5 are actually substantially coded. Briefly, the procedure for 1 involves two passes such as with the HPD. Rather than take line segments of all angles, it has been found convenient to do a pass using an active scan and then return to do an inactive scan.

This way, one doesn't have the annoying problems of following two tracks which eventually turn out to be the same track, or having to predict a track back into the same line of scan cells. The area scan procedure is a simple one in which you use two cells and their angles to predict the next cell and its angle. Some of the trickier problems involve identifying tracks in the angle - line number plot when there are two close tracks. Between the area scan and track following phase, it is proposed to have a general "clean up" phase where through tracks are accounted for, and track segments from the active and inactive scan are glued together, etc.

Now, skipping to the other more highly developed portion of the programming, we come to the track matching problem. The matching program TRIAD first identifies vertices of the same type. Then vertices of the same type are associated in the various views by checking their consistency with stereo reconstruction into space. Any tracks connecting identified vertices are, of course, easily taken care of. Then comes the hard part. Given a vertex in each view, identify each of the corresponding N tracks in each view. This is done using an elimination procedure. An \( N \times N \times N \) three dimensional matrix is set up, corresponding to track labelling in each of the three views. Then each pair of views is subjected to the following tests for all pairs of tracks:
1) Slope test

Is slope with respect to stereo axis of different sign?

2) Volume test

Does one point spatial reconstruction give rise to an unallowed space point?

3) Sign of curvature

Is charge inconsistent?

4) Turn over test

Is track (extrapolated) tangent to stereo axis?

These are the tests presently carried out. If the answers to any one is yes, then all elements of that view pair and track are zeroed in the matrix. (Initially all elements are 1). One then hopes to find that after all track pairs in all view pairs have been tested, that there are only N non-zero elements in the matrix, which identifies each physical track with a track label in each of the three views. In practice, this procedure is 95% effective on 4 prong events. (110 events gave 4 ambiguous cases, and one with no solution. Two of the ambiguous events were hard by eye. Two failed as a result of a bug for short straight tracks. The no-solution event is not yet understood). A proposed additional test is on depth which requires a nonotonic increase in depth on the basis of a simple film point test.
DISCUSSION

LORD: Where do they say "Yes we have an event on this frame" or "no, we don't"?

HUMPHREY: They put out a record of each track and how it is connected to any other track. They then have to identify tracks in different views to combine the various views. SCAN then scans through this TRIAD output tape with some topology in mind and this gives a list of events which have this event type.

LORD: You have to expend quite a bit of computer time before you can make a decision, so do you have any idea what density of events on a reel they would need to make this economic?

HUMPHREY: The numbers were big enough to make them think of using the PEPR machine as a high speed memory where you record $4 \times 10^6$ bits of information on a 35 mm frame by recording spots which can then be read by the PEPR machine at a rate which is 10 times faster than a tape unit and where the storage space is much less. This is a very promising idea, possibly for other library problems too.

LEBOY: Would you be using this in connection with bubble chamber film analysis, or do you mean for the general problem of storing information?

HUMPHREY: You might decide to use PEPR for only 2 shifts a day and use it during the other hours to do library work on scanning tapes, etc.

BLOCH: What do they feel about the adequacy of the PDP-I to perform the jobs ABCD?

HUMPHREY: They seem happy with it and even optimistic about the programming problems.

POWELL: Isn't it risky to use end points as a means of rejecting possible correspondence of two tracks in a chamber where you have non-uniform illumination?

HUMPHREY: They ask the question: "Is this end point physically meaningful?" If the end point is in a region with no light for instance, they recognise this as impossible and throw out that combination of 2 tracks.
GOLDSCHMIDT-CLERMONT: I did not understand your matrix. Is your method based on the fact that usually, if you reconstruct the track in space from 2 of the views it doesn't correspond to any of the tracks on the third view?

HUMPHREY: This matrix will represent a large number of elements, many of which are 1. As you eliminate various columns and shoot through shafts of zero when you find a combination between A and D for instance, then you should end up with intersecting shafts obliterating all but n elements in this matrix. Part of the idea of the whole scheme was to look at 2 views instead of 3 to simplify the logic. By taking pairs of views, one pair at a time, they found it a lot easier.
DAPR

D. HALL
Lawrence Radiation Laboratory, Berkeley

DAPR stands for Digital Automatic Pattern Recognition. The prototype programs for this system were originally obtained from Pasta, Marr and Rabinowitz at Brookhaven. In fact, the M Technique and DIRECTED LIST Methods were developed at Brookhaven and are used in the DAPR program. The DAPR system is currently being developed at Berkeley by Charles Dickens and Mary Downton.

Because of the similarity between the DAPR system and its counterpart at BNL, and because of the thorough description given by Dr. Rabinowitz in the morning meeting, it will be the purpose of this discussion to only point out some of the differences between the two systems.

The first difference lies in the beam track following phase. The Berkeley program uses the known beam radius and direction to create a shear table for transforming the tracks into straight lines. Allowance for energy loss is also made in the table. The method has proven to be very effective for following beam tracks. Furthermore, because of the table look up method, it is somewhat faster in execution than the Brookhaven method.

The second difference concerns non-beam tracks. Here the method is to predict a slope based on the preceding points rather than on the circular chords described by Dr. Rabinowitz. This method finds all of the non-beam tracks in the picture. However, they do not appear as complete track entities but rather as a group of shorter track segments (figure 1). The track collecting phase then groups the track segments into tracks.

The third difference concerns the track collecting phase. We observe at Berkeley that angle correlations are also quite useful in grouping track segments. Use of this information has been included in this program. This program has been run, but is not entirely debugged.

The vertex searching program has been flow charted, and partially coded. The aim here is to write a program to find two prong events as a first step. It is believed that this can be accomplished within two or three months.

7184/nh
Figure caption

Fig. 1 In the figure a picture from the 72 inch chamber which has been digitised on the FSD and then put through the program is shown. Each segment found by the program is identified by a letter, those with primes indicate beam tracks. Tracks whose slope changes rapidly along their length tend to be divided into several segments due to the current choice of program parameters.
DISCUSSION

BLOCH: Do you mean to run on-line with an HPD?

HALL: No. We expect to find events from a tape. Nothing is running on-line at present. It has been assembled so that it can run with a high density tape and this is approximately the same speed as an FSD.

RABINOWITZ: Could you indicate how the non-beam tracks are followed?

HALL: This is strictly a histogramming method. There is no shearing and no curve fitting, which is why there are so many segments on the picture (figure 1).
COMPUTER ASSOCIATION OF TRACKS IN DIFFERENT VIEWS

D. BURD
Columbia University, Irvington-on-Hudson

P.L. CONNOLLY
Brookhaven National Laboratory, Upton L.I.

(presented by P.L. Connolly)

In current production systems matching or association of
tracks in different views is performed by human scanners who have
available to them all the detailed information present in 3 or 4 film
images (figures 1, 2, 3). However, it appears that sign of curvature
and crude angle and ionization estimates generally serve as the means
of association. In difficult cases, drift-rays or the detailed gap-blob
structure of the track can be used to resolve ambiguities. The human
scanner appears to use overall properties of the track first and to
use detailed information only if there are remaining problems.

In automatic track recognition schemes, the same general
information will be available to the computer. However, quantities
like projected angle and curvature will be much more accurate than
the scanners estimates. Ionization information will be present but
might be complicated by questions of different illumination in certain
areas of the chamber. Detailed track structure by long gaps or blobs
is complicated by the fact that different sections of the track may
be obscured by crossing tracks in different views. Attemps to find
corresponding points on tracks in different views are open to essential-
ly the same difficulties. We are then left with the fact that the most
reliable information available from a track recognition scheme is the
angle at the vertex and the overall curvature of the track. We have
been attempting to develop an algorithm for track association based
on these two quantities.

* Research carried out under auspices of the U.S. Atomic Energy
Commission.
Method

We shall assume then that the recognition program gives output in three views:

a) Fiducials
b) Angle, curvature and vertex point for each track

Using the vertex points from 2 (or even 3 views) we first reconstruct the vertex point \((A, B, C)\) in the chamber using the full refraction equations. All refraction correction terms used in the remaining calculations are those evaluated at the vertex.

The parametric equations of a straight line with direction cosines \((L, M, N)\) which pass through the point \((ABC)\) in the chamber are

\[
X = \frac{L}{N} Z + A - \frac{L}{N} C
\]
\[
Y = \frac{M}{N} Z + B - \frac{M}{N} C .
\]

The image of this line on the projection plane is given by \(y = px + q\)

where

\[
p = \frac{M}{N} \left[ 1 + \frac{\delta}{v_1} - \frac{C}{v_o} \right] + \frac{B - Q}{v_o}
\]

\[
\frac{L}{N} \left[ 1 + \frac{\delta}{v_1} - \frac{C}{v_o} \right] + \frac{A - P}{v_o}
\]

and \(P, Q, R\) are the camera coordinates; \(\delta\) is the window thickness, and \(v_o, v_1\) are quantities containing the refraction corrections. The latter quantities are insensitive to \(x\) and \(y\).

The measured values of \(p\) in two views are sufficient to determine the direction of the track in space, which in turn predicts the value of \(p\) in the third view.
A prototype program has been written in FORTRAN which uses all possible image pairs in the first two views and scans the measurements in the third view. If the pair chosen in views 1, 2 is a correct match, there should be a corresponding projected angle in the third view. Based on a small sample of events from the Brookhaven 20" Chamber using conventional measurements, predictions on correct matches are accurate to better than $1/4^\circ$. It should be noted here that although the number of predictions and tests that must be made can be rather high, each individual calculation is extremely simple, and no square roots or extrapolations are involved.

There will be cases where incorrect matches in the first two views will produce a prediction which is satisfied in the third view and also cases where there will be two tracks in view three which will satisfy the prediction. Projected curvature is a considerable help in resolving these. Knowing the position, and direction at the vertex, and a projected curvature on views 1 and 2 the projected curvature in view 3 must also match. The errors in the curvature measurement appear to be of the same order as the refraction corrections so at least for the present we are ignoring the corrections. Normally the correct pair gives a prediction accurate to around 10%.

The program has been tested on a sample of events which gave the human scanners difficulty. To make things harder for the program no information was given to the program about the sign of curvature. This test was successful if the results were analyzed in the following manner.

Use all possible view 1, 2 pairs to find matches by method of tangents. This should produce all the right matches and possibly a few extra. If a view 1 image appears in more than one match apply method of curvatures to each of them. If for the track in view 1 there is only a single track in views 2 and 3 which match, call this a clear match. The remaining ambiguities will be cases in which a view 2 or a view 3 image is repeated. Reject all of these which repeat a track image involved in a clear match and check for missing or rejected tracks.

We hope to test this procedure on a large sample of events in the near future.
References


Figure Caption

Fig. 1 The two 4-prong events shown in three cameras illustrate the range of complexity which must be handled. The downstream event is a severe test of any track matching procedure.
DISCUSSION

HALL: What are you using for input?

CONNOLLY: We use the conventional Frankenstein type input - we had to produce the angles artificially since they are not directly measured. In a track recognition scheme the difficulty of producing these projected angles at the vertex is not great and in some schemes it might be there anyway.

MACLEOD: Connolly has given CERN 60-12 as a reference - I must give a warning. The equations for the determination of radius of curvature contain approximations which are valid only for the small bubble chambers (5" - 10" diameter) we were using at CERN at that time.
PAN 3 - THE HPD DIAGNOSTIC PROGRAM

R. GLASSER* and S.J. McCARROLL
CERN, Geneva

PAN 3 is a program used for testing the HPD system for some of the more common faults which have occurred. The program consists of FORTRAN and FAP coded subroutines loaded with the FORTRAN BSS loader.

Before describing the program, I will describe briefly the operation of the HPD from the point of view of the data presented to the computer. The film is fixed on a stage which moves along either the X-axis or the W-axis. In the following, the position of this stage, regardless of the direction of scan is referred to as the X-coordinate and the non-measuring coordinate as W. An X-coordinate has a marker bit in the 18th position and may have up to 17 bits for the value.

The HPD mechanical flying spot traces a line across the film perpendicular to the direction of the stage. Every time a point is found, the position of the flying spot is digitized as the X-coordinate in a 15 bit number. The 11 high-order bits of this number are from a counter whose units correspond to a 25.4μ spacing, and the 4 low-order bits are from an interpolation by 16 between counts.

Before each scan line, the HPD transmits the X-coordinate n times to the read-out buffer. The number, n, of these sequential X-values is fixed on the HPD at 1, 2, 4 or 8 and should always be the same (usually 8 for bubble chamber film). n is determined by the expected number of the Y-coordinates and is used in other programs to scan the incoming data to find the next X-coordinate.

The first set of X's is followed by the W-coordinate which gives the position of the stage relative to the flying spot. This coordinate also has the 18 bit marker and is transmitted only once. Then an unknown number of Y's are transmitted followed by the full grating count which gives a measure of the length of the flying spot path and should be constant.

* Present address: Code 7232, US Naval Research Laboratory, Washington 25 D.C.
The program (see Flow charts) checks the number and value of the X's, the order of the Y's, and the grating count. It keeps a histogram of the least significant hexadecimal digit in the Y-coordinates, and it checks that the X-increment lies between 20 and 120, a minimum and maximum that are set in the program and can be changed.

The program simultaneously reads from the HPD and processes the data until an End of File signal is received signaling the end of one picture. It then prints out the parameters for the run and any error messages it has found plus detailed information if desired. The main program is a FORTRAN program which calls the FAP subroutines F3D and DDGSET. F3D does all the checking of the data and saves all information for printing. It then calls FSOUT, a FORTRAN program, to print all the results. DDGSET sets the required transfer vector in the trap cells and deals with the trapping.

Four FAP subroutines are used for various conversions. ADRF converts the address and DBCF the decrement of the argument to an integer in the entire word. XLDRF converts the address and XDBCF the decrement of the argument to an integer in the decrement of the word.

The output of PAN 3 for each picture gives the total number of scan lines, the total number of digitizations, the W-coordinate, the initial X-coordinate, the initial number of X-coordinates for each line, and the value of the initial full grating count. Then it lists the average value of the X-increment (the X-translation between lines) and the r.m.s. deviation of this number and the minimum and maximum X-increment which occur within the prescribed limits. This is followed by a histogram of the LSD in hexadecimal for the Y-coordinates. All output is in octal except the line count, the number of X-values in a sequence, the r.m.s. deviation of the X-increments, and the numbers in the histograms.

Following the heading is a list of all errors found or the statement that none were found. For each error the printout contains the line number at which the error occurred, the X-coordinate of that line, and the error message. The errors looked for and the output are:

1. *No OF X-S CHANGES followed by the new number
2. X-VALUES DIFFER followed by the new one
3. Y OUT OF ORDER followed by the two Y values out of order
4. DEL-X TOO BIG followed by the value of the offending X-increment
5. *GRATING COUNT BAD followed by the new grating count
6. LSD = 17 followed by the Y-coordinate
The two errors indicated by the stars in the list above change the standard of comparison. Thus if a single line has a bad full grating count the error message is printed twice, once when the error occurs and once when the correct value is reestablished. The Y OUT OF ORDER error is also given for equal values of Y.

As a further aid to tracing the source of errors one can print out the contents of the lines containing errors. For the Y out of order only the line in question is saved, for all other errors the line containing the error, the preceding line and the following line are all saved.

To operate PAN 3 one loads the program deck with a BSS loader and FORTRAN I/O subroutines into the card reader, followed by a set of cards giving the orders to be transmitted to the HPD. Press the "LOAD CARDS" button on the 709 console. There will be a halt after the program is loaded. Press START and the program writes the message "the 709 is ready for the HPD" on the on-line printer. The computer then reads a card from the card reader and if sense line 7 has been activated sends the HPD its orders and starts to read in coordinates. If not, the computer delays until the sense line is on. The data from a single picture is read in blocks of 2000 words and checked before any output is initiated. At the conclusion of the output for one frame the program recycles and reads in the next card.

The program normally reads the data from the HPD on-line and prints the output on-line. By depressing sense switch 3 it will read tape B1 for input and write on tape A3. The data on tape B1 must be in records of 2000 words and there is no limit to the number of records in the file.

Detailed information on the errors found is printed out when sense switch 1 is depressed. If errors are found on any frame and sense switch 1 is not depressed the program stops after printing the message "Depress sense switch 1 for fuller print".

The orders to the HPD are input to the program on cards identical with those used by the 411 program. The cards are standard absolute row binary cards. The decrement of the first word (9 left) contains the word count (number of orders to be transmitted). The remainder of the 9 row and 8 left word are ignored and the orders are read starting from the 8 right word on the card. These are numbers in the first half of the word as detailed in DD/DEV/63/1. If data is read from tape, sense switch 3 down, there is no input at all.
Changes to the standard procedure: since the computing for the above checks takes too much time on the 709 we cannot keep up if the HPD operates at full density and low speed (its normal operating conditions). It is possible to suppress the histogram check on the L.S.D. by depressing sense switch 2. This will probably become obsolete as soon as the 7090 arrives.
Figure captions

Fig. 1  Flow chart of main processing subroutine "FSD".
Fig. 2  Flow chart of FSD exit and error saving routines.
Fig. 3  Flow chart of MAIN LOOP of FSD subroutine.
INFO
Calling sequence:
AC -N = # of lines back.
+ N = # of words back to start.
MQ M = lines to save.

INFO
Save XR's

Store line number and X value in Error table.

Increment addresses in buffer.

Is Error table full?

Y \rightarrow PRINT

Is N + or -?

- \rightarrow

Search for beginning of "N" line back.

Search for "N"+1 word back.

Move one point from working buffer to Error buffer.

Increment addresses.

Is Error buffer full?

Y \rightarrow PRINT

N \rightarrow RETURN

N \rightarrow RETURN

Figure 2
DISCUSSION

HALL: How is it that the X-coordinate can't be one count different on a scan line? Our X's can sometimes differ by one least count.

POWELL: We strobe the contents of the X-counter into a register and then copy it 8 times into the memory.

HALL: Ours is strobed 8 times directly from the Ferranti scaler.

POWELL: We really have had all these errors occurring. I would like to emphasize that in my opinion the computer is the ideal way of checking for these errors - so why not use it? I think this could be one reason why on-line devices will do better than off-line film scanning devices.
PIP: A PHOTO-INTERPRETIVE PROGRAM FOR THE ANALYSIS
OF SPARK CHAMBER DATA *

H. RUDLOF and T. MARILL
Bolt, Beranek and Newman Inc.

M. DEUTSCH
M.I.T., Cambridge 39

ABSTRACT *

"An operating computer program that processes photographically recorded data is described. The input to the program consists of spark chamber photographs on which tracks of high-energy particles are recorded. The program automatically scans, measures and performs the preliminary interpretation of these photographs. In continuous operation a processing rate of 5,000 photographic frames per hour is achieved."

* Work performed under the auspices of the United States Atomic Energy Commission.

DISCUSSION

ROUGH: Could you just give a sample of the sequencing of the spot movements in the course of measuring one frame in one film? Where is there human intervention?

RUDLOF: To begin with, there is no human intervention. The human intervention stage comes at calibration time where, using a number of sample photographs, you construct a calibration with pertinent information for all views and all frames of the experiment as a whole. After that, the only human intervention is changing the film. The sequence on the monitor scope is as follows: You will see first a vertical slash to find the fiducial arm, then a much smaller slash to find the other fiducial arm. When one of these slashes hits something you'll see a little activity while it finds the object size. The next thing you'll see is a bunch of little sweeps where it's going after the marker lights. It will read all the marker lights. Next you'll see it checking the secondary fiducials for tipping and distortion. Then you'll see it sweep the entire gap of a given view picking up the track origins; you'll see it follow the track. Then it will go on to the next view, check the secondary fiducials, find track origins, and follow out the tracks. If it does not find something in a gap, it may try the next one, or it may try all the way down. The thing is arranged so that as many things as possible are parameters. For instance, how many gaps it will look in for track origins; another parameter is if it misses, it will look a little bit more carefully by looking a little above and a little below the position of the first scan.

ROUGH: So, if you hit the spark you take one pass. In other words, you take as your point on the spark, the center of the spark between the two plates.

RUDLOF: Yes, that is right.

ROUGH: You mentioned that you have analysed several experiments, were these all done with the same spark chambers?

RUDLOF: No.

ROUGH: All done with parallel plate chambers?

RUDLOF: Yes, that is correct.
HOUGH: If you were to go over to analysing pictures from a parallel plate chamber experiment where you have an array of them some of which were vertical and some of which were horizontal on your picture. How much of your programming could you use?

RUDLOB: Practically all of it. At least for PIP-2, there is no fixed program there are simply a number of subroutines so chosen that you can fit them together to fit any experiment within a rather broad range. So that somebody who was getting up an experiment would have to be somewhat familiar with what these subroutines are. We hook them together for every new issue that comes up. There is no problem between vertical and horizontal because we have versions of this routine that will go in any direction; we simply use which ever one is appropriate for a particular view. This again is a view parameter.

POWELL: Can you give a figure for the failure rate that you have had with these pictures?

RUDLOB: You mean pictures that were wrongly rejected?

POWELL: Yes, either events that were found that were not real events, or real events that were missed.

RUDLOB: Prof. Deutsch gave a figure of 85% when he tried some test figures. 85 or 90% of things came out the way they should.

POWELL: Do you have any trouble in detecting faint sparks? We had quite a lot of trouble with this.

RUDLOB: Either it finds sparks or it doesn't. As a rule most of our things have been fairly big and black, and moreover they have to be more than several scope points in extent. There is a substantial background of objects of one or two scope points in size, so we usually expect our sparks to be three scope points wide.

LEBOY: What is your horizontal least count and the stability on your sweep?

RUDLOB: We use ten bits so that is one part in a thousand.

MOORHEAD: The experiment, I believe is set up for two separate cameras. Is that right?

RUDLOB: There are two film advances. The way we handle this is that when one is interrogating the film, one can connect either of two
flip-flops to the computer. So the way it works is that an event will be on two films; most of our events have three frames so there will be two frames on one film and one frame on the other. While it is scanning one frame, the other film will be advancing so as not to lose any time.

WISKOTT: What is the time taken to measure each event?

RUDLOF: The average rate per event is about three to five seconds depending on how complicated they are. It started out going much faster, but as we added more and more improvements to the program so that it would catch all the various cases, it got slower and slower. In a lot of cases, if the program wants to reject a track we make it look back or try again or look more carefully or try other criteria and this slows it down a bit.
CONCLUDING REMARKS

L. KOWARSKI
CERN, Geneva

There will be no time to deploy a stately logic and I shall have to rely on the feelings I have accumulated during these three days. Feelings are apt to be strong, full of prejudice, and not at all accurate. Those who find themselves on the receiving end of my prejudices please remember that no accuracy is claimed and, therefore, no definite slander is intended.

My main feeling suggests a rough classification into two kinds of speakers. Among the authors and the participants in discussions, many gave the impression of a subdued mood. This does not necessarily mean that they were pessimistic; some of them expressed very optimistic views, and yet they spoke in a minor key. Other speakers, however, were cheerful and exuberant. The general correlation, with a few exceptions, is that those who have attempted to do real physics belong to the first class, while the cheerful ones are mostly those who are about to do physics or believe they are about to do physics. There are one or two significant exceptions which we shall mention.

Now this correlation between the attempted physics and the degree of good cheer is worth going into. For this purpose I shall use some of the concepts mentioned in my Introduction the day before yesterday, and draw a chart which refers to that conceptual space. I spoke of two main controversies, one relating to the "Third stage", that of the human intervention in the machine work, and the other relating to the "Fourth stage", in which the human intervention has been eliminated or is about to be eliminated in order to achieve speed. Here they are on the chart (fig.1) and let us consider first the Third stage. We have two main philosophies, in one of which the human and the machine are cheerfully combined, and in the other they are segregated. On the combined side, the Berkeley SMP wears a happy smile (which, no doubt, accounts for its name), because it has already reached the stage of actual use for physics; I shall underline those projects which have progressed that far. On the other side we gave the Berkeley experience, the Brookhaven experience, and the CERN experience. Paul Hough will recognize here his
logarithmic ordinate. As we know, these three foci of development of HPD have reached very different degrees, logarithmically different degrees, of success in physics. It is partly due, I am sure, to their respective depth of insight, but partly also to the fact that those who have some success have made their task slightly easier. There are at least two immediate ways of doing things the easier way, and we shall see them in other projects. One is to select the experiments, to postpone, so to speak, the universality of application, which ultimately will of course be required. To begin with, one chooses an experiment which will be particularly easy to tackle with this particular machine. It seems that Berkeley did choose this way and was rewarded by a definite success in physics; we have again to underline Berkeley. I think it has been mentioned here that the events with which they are at present successful are fairly simple. Quite possibly, at the present stage, with more complicated events, the percentage of success would be somewhat lower. We may have observed that the Berkeley success has been reported here with a note of caution, recognizing that there is considerable room for improvement.

The other way of alleviation is to sacrifice provisionally not the universality, but the automaticity. Brookhaven still has a human on line, so I am not sure it rates a full underlining; let's do it with a dotted line. Somebody told me privately, that as long as that human on line is there, the HPD as it now works at Brookhaven is just a very expensive SMP. I think that as in all outrageous statements made sincerely, there is a small hint of truth in it - a very small hint. We shall point our dotted line in the direction of SMP, bearing firmly in mind that both the pointer and the dottedness are short-term features.

So Berkeley lacks universality, Brookhaven lacks automaticity, and CERN which proudly refuses to sacrifice either universality or automaticity, so far lacks successful achievement in bubble-chamber physics. As we shall see in a moment, it has fared better in sparks.

So much for this Third level, and now we go to the more ambitious Fourth-machine only. Here we have again, but in a very different way, a split into two approaches which I call - and I am glad to hear it has already caught at this meeting - the approach of the tricky hardware and that of the tricky software. Here we also have two levels of ambition: the deep-reaching ambition to process bubbles and the more superficial ambition aiming at sparks.

It has been explained why the spark ambition is the smaller one: there is less information in each picture, it is somewhat less diverse, easier to handle.
For sparks, tricky hardware really means a programmed spot: one system of this kind has achieved a very definite success and this is Deutsch's SPASS. For bubbles, on the side of tricky hardware we have PEPR. For tricky software we may start, on the sparks level, by putting down that particular use of EPD (at CERN). Then we have there the Brookhaven system described by Baker, and the beginnings of a CERN cathode-ray system, based again on the universal TV-type serial scan, which we call Luciole and on which we had no time to report. For tricky-software bubbles we have the Pastrami-DAPR system *. Unfortunately there was no one to report here on the present work done under McCormick, where I think they tried to simulate the future McCormick computer on the 7090, using a program called PAX. Perhaps, when McCormick completes his very special computer, we will have to transfer him to the Hardware side, but for the time being, the tricks are definitely of the software kind.

How do we stand with physics? SPASS and EPD-CERN have to be underlined, no other Fourth stage system as yet. We heard interesting news today on the existence of spark-processing systems intermediate between the Tricky Hardware and the Tricky Software. For instance, I would put the Nuvis system closer to the Software side because the discrimination by hardware intervenes only for one coordinate and in a rather simple way. There seems to exist an interesting uncertainty concerning SASS.

It looks similar to SPASS, but the authors are not sure yet whether or how much they are going to use the programmed spot. No doubt this uncertainty accounts for the disappearance of the letter P from the system's name. And there is the project Chloe of Argonne which was not mentioned here at all, and which in the first approximation looks similar to SASS - also a programmed spot and also some uncertainty about how it will be used. Well, it seems that the chart now contains everything that has been discussed here. One more remark about these two successes in physics which after all deserve the greatest measure of our immediate interest: SPASS is far more successful than any other pure machine system but I think that

* It is worth noting in this connection that the approach used by Burren and Moorhead for the solution of the filtering problems in the HAZE program for bubbles may be seen as a promising Third Stage prefiguration of the Tricky Software philosophy.
it has been leaning on both the alleviations which I mentioned before. So far, there has been usually a physicist on line, to wit Deutsch, and on the other hand we heard a moment ago that so far only the PIP-1 program, which was devised for a special experiment, has been in full operation. I have no doubt that gradually these two limitations will be removed. As regards the CERN system one can say that Blackall's report on it reflected the same subdued mood which we could perceive in Hall's report from Berkeley. It pointed out the present percentage of rejects, the expectation to reduce it, and the fairly ambitious scope of the project as regards full automaticity and universality. In fact, the universality, that is the possibility of using the same piece of equipment for a wide class of experiments, is, I think, a very desirable quality because it makes it easier to build a stable and reliable hardware. This point perhaps has not yet been quite appreciated, but I notice that it was mentioned by one speaker today.

A look at the chart may remind us of the question of a proper collaboration between the physicists and the specialists of data processing. One strong hint comes from the left upper corner: the SMF admittedly must always have a physicist present. With this presence it was quite natural that the system in question was able to achieve its clear successes in physics. Here, as in SPASS, the physicists have scored their successes, starting from their customary methods of approach. Since the data processing specialists have been more slow to organize their intervention and their methods of collaboration with the physicists, it is again natural to expect that the systems, in which the track-chamber physicist has played a lesser role, have so far been less successful in physics. A conclusion from that might be drawn that physics should better be left to physicists and let them achieve their successes. I would like to mention here at least three arguments against this conclusion. One is that physicists are, on the whole, interested in the kind of physics they know, and they adapt their machinery to the kind of physics they know, whereas the data processing expert can very well prepare the machinery for a kind of physics which does not quite exist yet. It is obvious that the kind of physics that is going on at present is better handled with the full participation of physicists and that the HFD, for example, either in its human-containing mode or its totally inhuman mode, should be used (Dr. Calkin mentioned it today) in a kind of physics where there are millions of events to be processed, and where zoology plays only a small role.

As physicists turn to this latter kind of experiment, (and I am not the only one to think that this turn will come), it will become urgent to develop the corresponding processing techniques, and
here the active intervention of data-processing specialists may well become necessary. Let us keep in mind the lesson of Berkeley, where data processing, as such, has been given more attention than elsewhere, with conspicuous results.

The second consideration which tends to be forgotten, but was again mentioned today by one of the speakers, is that other sciences are getting interested in this kind of apparatus. Molt populations are being photographed and counted; we hear also from Argonne that their Chloe is already in use for some biological research, prior to the intended use by high-energy physicists. Obviously, if the physicists continue to take a very preponderant part, they will not develop a machine universal enough to be of use to other sciences - and other sciences have to be served too.

The third consideration is the very probable existence of classified work aiming at the development of visual data processing for defence purposes. It will be quite normal and in keeping with past history, if this new tool for fundamental scientific investigation comes out in its perfected and serviceable form as a by-product of defence research. Such research means a lot of money, crash programs, and everything it takes to arouse the interest of big corporations. Under the influence of this third factor and after a suitable time lag to let the secrecy abate, it may well happen that fundamental science will yet profit from gadgets developed for defence purposes.

One such gadget appears to have been nearly completed by one very big data processing corporation. It will offer a full programmed spot system for handling film images; I was even asked whether it would be of interest to bubble chamber work, but their fundamental raster is 4000 x 4000 and this is probably not enough for bubbles. It would certainly be quite enough for sparks, and the spark people should be glad to know that this instrument already exists in a virtually complete and, of course, suitably expensive shape.

The combined operation of these three factors amounts to a sort of prognosis which I am now venturing to express, and this will be my last concluding remark.
THIRD STAGE
(Bubbles)  Human ( Machine
Berkeley

FOURTH STAGE
(Sparks) SPASS  Chloé  Nevis  Brookhaven (Baker)
SASS  HPD-CERN  Luciole

(Bubbles) PEPR  Pastrami - DAPR  PAK ?
CURRENT STATUS OF THE DIGITAL PATTERN RECOGNITION
PROGRAM AT BNL *

G. RABINOWITZ
Brookhaven National Laboratory, Upton, L.I.

Outlined in Fig. 1 is a possible approach to the problem we are considering. The first phase is what we call Track Reconstruction, which is done in real time and in which we attempt to deal with all the digitisations coming into the computer, i.e., reduce this data into information about track segments. I want to emphasize that this phase, is done in real time and therefore must be done rapidly and efficiently. No matter how well the track reconstruction program functions, there will always be instances of failures and of error and these we hope to handle in the second phase, which we call Track Editing. This phase will scan the collection of track segments, dismember segments which do not belong together, link those which do but have not been joined by the program, and perform a general clean up. The output of this is then passed on to the third phase, the Vertex Finding program. I'd like to emphasize that all the work we have done so far has been limited to these aspects of the problem and also that all considerations have been with respect to just one view. Following this one might try to recognize selected types of events and go on into the kinematics programs.

I might mention that the track reconstruction program has been worked on principally by Bob Marr and myself and the other two phases have been worked on by Bill Beard and Betty Veneser. I think that in terms of describing where these things stand, the first is in the most advanced stage of development; programs exist for the other two, but to date there has not been enough experience with real data to know how well they will function—it's only recently that we were able to get fairly reliable track reconstruction. The second and third programs were tested out against some dummy data but this does not really get you very far.

In the remainder of my talk, I am going to concentrate on the Track Reconstruction problem. I'd like to begin by discussing the input format—we assume HPD data in standard format coming in either from tape or from the DDC. The data is put into a buffer of total size 4K which is divided up into 8 sub-buffers, and after the first 512 word

* This is a transcription made from the recording of Rabinowitz's talk. It has been edited only where it seemed absolutely necessary.
buffer is filled up we then start processing the data. These buffers are filled cyclically, and we have provision for a trap in case the input overtakes the processing. Now let me mention what the output of this program is - in the upper memory we have blocks of storage set aside corresponding to each track that has been reconstructed. The length of each block is roughly proportional to the length of track under consideration. The structure of these blocks is as follows - we have at the beginning of a block a heading which consists of 4 words and contains general information about the track such as the density, the track type - whether it’s a beam track or non-beam track and whether the track started at the top of the picture and went out through the bottom or ended at the side etc. Following this heading information are stored the first three digitisations along the track. After that we have a list of averaged points accumulated along the track where each average point represents either 4, 8 or 16 digitisations. Following the averaged points are the digitisations at the end of the track. In addition to this type of information which really refers to just the track segments, we have a block of residual data. The residue consists of noise points, beginnings of tracks, etc., and is stored in a buffer in the same order in which it is found by the program in the form of a W coordinate and the corresponding scan line number. Heavily represented will be things like nearly horizontal tracks, (i.e. tracks almost parallel to the scan direction), electron spirals, etc. I mentioned that the beginning of a track is always represented in the residue - the fact that these points are on a track is indicated by the fact that these words will be flagged so that it is possible to distinguish between true residue and track beginnings. In addition to these two categories of data, we have a table of X grating counts versus scan line number and some program statistics, the number of beam tracks, total number of tracks, running time etc. We keep track of the running time using the clock on the 7094.

The program is organised about the use of 50 or 60 identical blocks of coding, each of which at any given time will correspond to a track which is being reconstructed and in this block of coding are contained parameters about the track and various instructions which essentially expedite the handling of the data relating to that track. Fig. 2 gives the overall organization of the track reconstruction program.

Aside from the track blocks themselves there are three types of functions that go on in the program and we term those sorting, tracking and control. Sorting is obvious - we want to assign data to track banks as quickly as possible - there is interaction between the track banks and the tracking section which controls the extrapolation of tracks, the monitoring to make sure that tracks continue to pick up data at a reasonable rate for that track, the performing of data reduction and track initialisation.
There are two principle techniques used in the sorting section. The first is the "directed list" technique in which we utilise the fact that the input data as they arrive are nonotonically ordered on a given scan line. The track banks are also kept ordered across a scan line by the use of pointer words; therefore a single pass through all the data on one line is adequate to determine whether or not the data belongs to any of the track banks present. Now this clearly has some logical complications because it is possible for tracks to inter-change their relative position on a scan line which means that some book-keeping has to be done to take this into account – some re-ordering is necessary from time to time. The second method is the "pseudo-register" method which involves setting up a block of words in storage which essentially constitute a one dimensional map across a scan line; the words in this one dimensional map either contain zero which indicates that there is no track in that region or else they contain a pointer to the appropriate track block. Now if all tracks were vertical this would be an ideal method of handling the sorting problem because the structure would never have to be changed and one could simply imagine everything raining into the appropriate bucket and going very neatly – but of course things are not that simple. There is a category of tracks for which the pseudo-register technique is useful, and this is the beam track. Even though they are not vertical they have the property of being well collimated and we can adapt this one dimensional map to the process by applying the appropriate transformation to make them vertical. So we handle the beam tracks using the pseudo-register technique and we handle the non-beam tracks using the directed list technique.

Fig. 3 illustrates the basic processing loop. The program exists in three separate levels, the beam track processing program, the non-beam track processing program, and a program to process true residue and tracks which are just starting. Each incoming point is first tested to see if it lies on some beam track, because of this one dimensional map this can be done very quickly, then it is tested against a directed list track. If it is found that geometrically it lies between two tracks in the directed list, then it drops through to the initialising scheme. We have found that this division into these distinct categories is extremely powerful. Let me point out that this has the effect of eliminating the beam tracks in a picture from the view of the non-beam tracks thus removing much of the confusion which does arise if you treat all the data on the same basis. The second point is that we are able to use search regions which are appropriate to the tracks under consideration. At the beginning of a track, we use very wide search regions when we are trying to get things linked up just roughly. When a track is well established we use quite narrow roads, say of the order of 50 microns or so. In between we use roads of variable width depending upon the length of the track segment that we have. I also want to mention that since most of the digitisations do come from beam tracks, it is most efficient to order the sequence of tests in this manner.
I would now like to mention the initialisation procedures. When a point drops through these first two levels, it is immediately entered in the residue. We have a second "one dimensional map" of unassigned points which has a somewhat coarser structure than the beam-map I mentioned earlier, and the W coordinate of the unassigned point is used to reference three consecutive cells of this second map. If no entries are found there, a pointer to the storage position in the residue is inserted into the map and that is the end of the processing for that point. If a pointer is found in one of the three cells which are referenced (let me just mention the scale involved here - each of the cells corresponds initially to 16 least counts, or 32 microns, so we use a total search width of 96 microns), a test is made to check how many scan lines ago the last entry was encountered - if there is too large a scan line gap we do not associate the points, the old entry is destroyed and we just put the new entry in and continue processing. If the test is passed, then these two points are associated together and again we continue. The number of points to be linked in this very rough manner is a program parameter and so far we have used 4 hits as the requirement. After we get 4 hits linked together this is then considered to be the beginning of a track and we compute a rough slope and position for this track and this then is inserted into the directed list scheme.

The way the program functions can be understood by referring to the figures which follow.

A line printer has been used to plot this information, one line of print corresponds to a scan line; a character printed in a line represents the position of a digitising on that line. The same character on successive lines shows the points assigned by the program to the same track.

Let us turn to Fig. 4. The numbers on the left hand represent scan line numbers - each character printed here corresponds to a digitising. The asterisks at the top refer to points in the unassigned category, once the first four have been linked together they are considered to be part of a track segment.

At the beginning of the processing of a picture, we would like to pick up the beam tracks as quickly as possible, since this seems to be the most efficient way to proceed. We have made provision for using the beam direction - if known - in this initialising procedure. I mentioned earlier that initially the cells represented 16 least counts - this is the case until the beam direction is determined - after this each cell represents 32 least counts.
Let us now concentrate on the history of a single track as it develops. We have computed a rough slope from the first 4 hits - whenever we compute a slope for these initialising tracks it is done by taking some number of hits which will be 4, 8 or 16 and dividing them into two groups and computing the slope from the averages based on the two groups of hits.

The numbers printed to indicate the digitisations represent the tracking level at which the data was handled - after collecting the next 4 hits, which are printed as the character 1, the program then uses the 8 hits to compute a new slope until all the hits printed as 2 have been collected and it goes on in this manner. The number of digitisations entering into a slope computation is computed by referring to a two dimensional table which is a function of both the number of scan lines elapsed on the track and the number of hits on the track. It's clear that for a very sparse track you would not want to use as many digitisations as on a very dense track, for instance. Essentially, you would like to have the data entering this computation represent a fixed length and this is one way of accomplishing it.

The program proceeds in this manner - recomputing slopes in the described manner until the 64th scan line is reached; this is an arbitrary number which we have used as the dividing line between established tracks and non-established tracks. First I would like to describe how non-beam tracks are handled. Several things happen at this point. First of all, a decision is made about the density of this track, then data reduction is performed - we take all the digitisations up to that point and collapse them into 4, 8 or 16 hit average points. For the remainder of this track's history extrapolation is performed by circle fitting. Initially, we use the first three average points we have, fit a circle to them and use that to set up a road for the track. The roads themselves are set up by using a polygonal approximation to a circle. It might be appropriate to mention in this respect that the computation of the circle parameters take of the order of 500 cycles and the re-computation of successive edges of the polygon take about 100 cycles. If the maximum permissible deviation of one of the edges of the polygon from the circle is $S_{\text{max}}$ we find that we can extrapolate a length $4\sqrt{S_{\text{max}}}R$, where $R$ is the radius of the circle. This length in most cases is of the order of one or two hundred scan lines so that the polygonal up-dating occurs relatively infrequently. One further point is that if at a given point in a track reconstruction we do a circular fit - this fit is only used to extend the track over 50% of its accumulated length. The effect of this is to make the standard deviation of the extrapolation error comparable to the standard deviation of the distance of the individual digitisations from the track.
Now I'd like to mention something about the treatment of beam tracks. Periodically the program scans all the track banks and examines the slopes of the tracks it finds. If the number of track banks exceeds some minimum (which is a program parameter), then a histogram is set up based on the slopes of the track segments and if a large enough pulse is found then the tracks entering into this pulse are said to determine the beam direction. At this point each of the tracks which have been accepted as a beam track is entered into the beam map mentioned earlier, and a pointer is set in the appropriate cell to the assigned track block.

Figs. 5 and 6 represent the output from an earlier program but the principles still apply. Given this structure of words in a beam map you have the problem of essentially manipulating these words so that they are always centered about the track under consideration. I don't know how visible this is but there are overprints here which represent where the digitizations actually fell and the three letters represent the position of the words in the beam map. The centering is performed by using a histogram of the hit distribution among the three cells — when the shape becomes lopsided enough, a shift in one direction or the other is made. Although this seems like a very simple minded procedure it seems to work quite well. I think the reason is that in setting up the transformation to make the beam tracks vertical one is using data from all the beam tracks and this seems to be a very powerful device. In addition to being recognized as a beam track at the very top of the picture when the beam map is first set up, each track at some phase of its development is also tested for being a beam track. If its slope is close enough to the correct beam slope then it is included in the beam track scheme.

The next thing I'd like to mention is how we determine when we have lost a track or run out of data on a track. At each stage of a track's history we use the density which has been observed up to date to reference a set of tables which are based on the binomial distribution which tells us how many scan lines we should reasonably wait for the number of hits we want to accumulate before exiting to the next tracking routine. The tables have been set up so that the probability of being in error is less than 0.01. If we fail to get enough hits twice in succession then the track is terminated and the corresponding track block is returned to a rotating list of track blocks which are available for further use. This list of track blocks is kept in the form of a stack of plates, when you want one you take one off the top, when you want to put one back, you put it back at the bottom of the pile.
Figs. 7, 8 and 9 represent a continuation of Fig. 4. In Fig. 7 all three of these tracks are still considered as initialising tracks because they have not yet gone 64 scan lines. The criterion is not 64 scan lines but a fixed length — the inclination is referenced in making this decision. As soon as the decision is made about the density of the track, a different kind of labelling occurs. A permanent label is assigned to the track, the letters A-T being used for established beam tracks, the characters +, -, =, $\neq$, ( ) being used for non-beam.

Fig. 10 is an overall view of the picture we have been looking at. The top picture represents the CRT display of digitisations of the original photograph. This is the event we were looking at, this is the track that crossed from one side and these were the two arms. The second picture represents all the data which in the printout was flagged with asterisks — this essentially is the residue. It also includes the front ends of tracks — the initialising parts of the tracks — so that at the top the beam tracks are heavily represented. Electron spirals show up very clearly in the residue.

On the lower right of Fig. 11 is a CRT plot of all the tracks the program has determined are beam tracks. Alongside is another CRT plot of all the established tracks which are not beam tracks and above the third picture is a reunion of these two photographs. Now I'd like to make some qualitative comments on this picture. The original photograph represents about 17,000 digitisations, the total number of track segments is of the order of 130 and there are roughly 60 established track segments. This gives an idea of the data reduction which has been accomplished.

Figs. 12 and 13 are corresponding displays for a second photograph.

I'd like to comment on the processing speed. The speed depends very strongly on the kind of photograph — for instance if the picture is very clear and with well organised data, then of course the program runs most efficiently. In a messy picture when many track segments are starting up and then being terminated, there is a lot of book-keeping which is time consuming. The range seems to be — and this is really based on very little experience — between 8 and 10 thousand digitisations/sec.

Now I'd just like to comment briefly on the second and third boxes which I displayed initially, namely the track editing and vertex finding programs. As I said, these programs have been coded — as a matter of fact there is a Brookhaven report being written entitled "A Digital Scanning Program for Single View Bubble Chamber Photographs"
and it should be available before too long. I will first describe
the track editing aspect. A one dimensional map of end points is set
up on the basis of scan line number and in addition a list is made
of track segments ordered according to length. The program starts
with the largest segment in this list and attempts to extrapolate this
segment in both directions using circular fits. If matching segments
are found they are incorporated and this procedure isiterated until
nothing more can be added on. When this stage is reached a pass is also
made through the residue with what has been put together so far. Our
objective is to clear things up as much as we possibly can. After a
single track is completely reconstructed, we then go and process the
next longest track and this process is again iterated.

The end points mentioned are not the individual digitisations
but the first and last average points which seemed like a more reliable
thing to use. One part of the processing is simply a test for kinks.
If a kink is found the track is dismembered. After the tracks have
been maximally linked there is a program which attempts to use a simple
area criterion to locate vertices. These programs are still in a pre-
liminary form.
Figure captions

Fig. 1 A schematic representation of the general approach discussed in this article.

Fig. 2 A schematic representation of the Track Reconstruction program.

Fig. 3 The sorting of the incoming data.

Fig. 4 The numbers running down the left hand margin are the scan-line numbers. The labelling of the digitisations indicates the stage of development of a track. Each print position corresponds to $12\mu$ on film. Data not assigned to a track bank is marked by an asterisk. The integer labels indicate the number of times the track has been examined by one of the tracking routines. The track corresponding to the four nearly vertical asterisks is too sparse to be considered a track candidate at this stage.

Fig. 5 Each beam track is represented by three consecutive words in the pseudoregister — corresponding here to the same character appearing in three consecutive print positions. The program shifts the words left or right to keep the track centered within the three positions.

Fig. 6 Because of the narrow road width and dependence on the collective behavior of the beam, the following of individual tracks is relatively insensitive to crossing tracks and background noise.

Fig. 7 This is a continuation of Fig. 4. The letters U-Z are used for initializing beam tracks, A-T are used for established beam tracks. The sparse track (to right of track labeled B) is dense enough to be initialized, i.e., entered in a track bank.

Fig. 8 Track B leads into a vertex, with two tracks emerging. The track on the left is initialized and followed immediately. The track going off to the right is both wider angled and sparse and is not followed nearly as well.

Fig. 9 The track angling off to the left is the continuation of the left hand track of Fig. 6. After approximately $1.5\text{ mm}$ along a non-beam track, linear extrapolation is replaced by circular extrapolation, indicated by the changeover to the + character.
Fig. 10 CRT Displays of FSD data from a single 20" bubble chamber photograph, segregated according to its treatment by the track-following program. The top display represents all the data. The lower left display represents the residual points, which were not assigned to track segments. On the lower right are displayed those points assigned to track banks in the linear fit mode of extrapolation.

Fig. 11 Continuation of Fig. 10. At the lower left are displayed those points assigned to tracks in the circle fit mode, while the lower right represents the beam tracks. The top display is the union of the two lower ones.

Figs. 12 and 13 CRT displays of FSD data from a different photograph. The format coincides with that in the previous two figures.
Figure 1
Begin scan line

Re-entry after assignment of point

End of Scan line

Yes → Exit to set up for next scan line

No → Fetch next input point

On beam track? (Test against B. T. map)

Yes → Exit to beam track bank for assignment of point

No → On another track? (Directed list scheme)

Yes → Exit to track bank

No
DISCUSSION

LORD: When do you decide you will need an orthogonal scan? Do you decide you have an event before you call for it?

RABINOWITZ: This question has not yet been faced. We hope that by intensively searching both through the residue and the track segments, we can at least determine whether one is called for or not.

MACLEOD: Could you say how much use you have made of the on-line CRT in developing this sort of program?

RABINOWITZ: We haven't, not on line. Of course we take pictures and we find these useful. Actually what we have found most useful, have been printer plots for diagnostic purposes. Because we really want to see in detail how these programs are functioning on a microscopic level. You can't tell very much from the overall pictures which I showed at the end.

HALL: Did you say that when you were matching up the track segments that you found the end point matching the most useful? Or did you say it was matching the angles?

RABINOWITZ: The program as it is presently written essentially uses just nearness of end points. I'd like to emphasize that my concluding remarks were really all very tentative.

HOUGH: Do you have any comment on the use of these techniques within roads.

RABINOWITZ: I think that some of the extrapolation techniques might be relevant, but I don't think the method as a whole would be very useful because it really is so big and complicated.

BURD: Could you go into some detail about the kink detection procedure?

RABINOWITZ: Essentially we take some sub-set of the average points along a track, make a circular fit and look at the deviations.

SCHIFF: Does your one dimensional map allow you to take into account crossing beam tracks? I ask because we plan to use your method for our gating procedure.
RABINOWITZ: What we do now is sort of a makeshift. Namely if beam tracks come close, we stop them both at that point. However, this is not the right thing to do, it's just that there are other things to take care of first. You always know when beam tracks are going to cross, because two entries will occupy the same cell. You obviously can do something much more sophisticated than just to stop them.
A PROGRAMMING SYSTEM FOR SCANNING DIGITIZED BUBBLE-CHAMBER NEGATIVES

R. NARASIMHAN
University of Illinois, Urbana

ABSTRACT

This paper describes the functional characteristics of a program for scanning digitized bubble chamber negatives which is in an advanced stage of realization in this Laboratory using an IBM 7090 computer. The program is set up to scan the negatives one view at a time. Comparative analysis between separate views is explicitly relegated to a higher-level post-editing program; enough peripheral information is generated while scanning the single views to make this post-editingsimple and efficient.

The present scanning program starts with a suitably preprocessed, digitized, image of a negative and generates two output lists: (1) a TRACK LIST and (2) a VERTEX LIST; these list the names of tracks and vertices (with their associated tracks) identified in the input negative. Each track name and vertex name carries with it other subsidiary information for further classification of these (at the post-editing level) into V-type vertices, stars, bean tracks, spirals and so on. The tracks are identified spatially by listing the rough x,y coordinates of a string of points which lie on them. This coordinate information will be made use of in the subsequent measuring phase to obtain the precise coordinate measurements of the points which constitute the tracks.

The schematic of Fig. 1 shows how the processing carried out by the scanning program fits into the over-all scheme for the automatic analysis of bubble-chamber data, as envisaged at present. The details of the various parts of this program are adequately documented in the references listed below:

* Report No. 139, Digital Computer Laboratory, Univ. of Illinois, Urbana; (June 1963).
References


2. B.H. Mayoh: Bubble Chamber Scanning Program: Syntax Table for the Compilation Phase of the MAIN Program; DCL File No. 538 (May 1963).

3. R.K. Rice and R. Narasimhan: Bubble-Chamber Scanning Program: 1. LABEL, 2. SEARCH (Stage 1); DCL File No. 542 (June 1963).

Figure caption

Fig. 1  The analysis scheme for bubble chamber data.
Figure 1

Film input

One view at a time

Digitizing

Preprocessing

Scanning

Post-editing
(event classification based on scanned output from three views)

Analysis

Measuring

Output
LUCIOLE

H. ANDERS, T. LINGJAERDE, J.A. WILSON and D. WISKOTT

CERN, Geneva

1. Introduction

Luciole is an on-line flying spot scanning device designed for the analysis of spark chamber photographs. The first on-line tests using the IBM 7090 are due to take place in October 1963. The device uses a high resolution CRT to produce a continuously running TV-like scan with a spot size on the film not exceeding 30 µm diameter over 36 x 36 mm².

2. Approach

Luciole is, according to Kowarski's terminology, an example of the "tricky software solution" to the automatic processing of spark chamber pictures. It has evolved from the HPD and is thought to be better adapted to the particular problem of spark chamber pictures. While less precise than the HPD, the time to scan a picture is less and the cost is relatively low. Particular attention has been devoted to the stability and reliability of the device 1).

3. The flying spot generator

A Ferranti type 5/71 AM CRT is used together with CELCO focusing and deflection coils (fig.1). The raster is produced over a 60 x 60 mm² area within which the spot size is always less than 50 µm diameter *. This is demagnified by 1 : 0.6 onto the film by means of a pair of lenses face to face (Schneider Xenar and Componon).

The time for the complete scan is 1.14 sec and this is followed by a flyback time of 0.38 sec. A scan contains a total of 768 lines approximately 50 µm apart. The least count along the scan time is 22.5 µm.

* A much finer spot can be obtained at the center of the screen; even over the whole scanning area, a more stringent requirement could be met. We have preferred to chose a value that can be maintained over several hours without re-adjustment.
The pin cushion distortion has been roughly corrected for by an array of permanent magnets. The residual distortion will then be corrected for by the computer on the basis of a calibration made every few hours by scanning a suitable test pattern.

A reproducibility has been obtained in the "fast" direction (F) along the scan lines of better than $\pm 10^{-4}$ rms and in the "slow" direction (X) perpendicular to the scan lines of better than $\pm 3 \times 10^{-4}$ rms. Slow general shifts of the raster in either the F or X direction do not affect the measurement process.

4. The digitising process

A method of digitising ("Sampled Analog Digitising") has been chosen which does not necessitate a large buffer memory. At regular intervals (corresponding to the least event in F) the photo-multiplier signal corresponding to the scan across the film is sampled. A sequence of zeros and ones is generated where the ones correspond to opaque regions on the film. This sequence is fed into a shift register of 17 bits.

After every 17 clock pulses, its contents are transferred into the lower half of a one word buffer provided that the register does not contain only zeros (figure 2). Subsequent half words go alternately into the upper and the lower halves of the word. Such a sequence of half words ends when an upper half containing only zeros occurs. Then this is replaced by the true F-coordinate of the last sampled point expressed as the total number of clock pulses counted from the beginning of that scan-line. The X-coordinate (number of the scan line) is read out at the end of each scan-line (except when this has been completely blank).

The timing of the load-unload procedure of the buffer is such that each completed word can be read into the 7090 before the next load operation is due.

5. Speed

The speed of the device is limited essentially by the maximum rate of the DDC (estimated to be one transfer every 17.6 $\mu$sec. for our operating conditions and by the film transport (about 0.12 sec. actual advance time plus settling time). Because the raster runs continuously, the maximum speed will not be reached unless the computer can complete its processing before the scan of the next picture starts. If it should fail to do this, no data is read in and that picture will be scanned again by the following scan. In a later version of the device, this loss can be eliminated.
6. Programming

Programming for Lucile falls basically into two categories:

a) programs designed to test hardware,

b) programs to process experimental photographs

The present programming effort is concentrated on the category a), for which programs are already well advanced.

It is the aim of the programs in the latter category to read in Lucile information from one frame, write out processed data from a frame two previous and process the previous frame simultaneously, the complete cycle taking a maximum of 1.5 sec. It is proposed to process each frame in two passes of the input data. Pass (1) to locate the fiducial marks (and hence the chambers), and pass (2) to locate sparks and associate them into tracks, possibly in space coordinates. Data from each frame will be written on tape with a record to identify it, and it is hoped to apply some simple criteria to eliminate frames which are of no interest.

For this work the experience gained on the HPD is useful, although the existing programs are not directly applicable due to the format of the data. In the computer the data appears as a string of binary digits representing dark and light areas on the frame; in other words a bit pattern is assembled in the machine, and it is proposed to utilise this information directly by means of pattern recognition programs. The object of such programs is to "mask-out" the binary digits corresponding to a cross for example, and then to test on the remainder of the search area for zero. In figure 5 an area of memory has been depicted as an array on which the program will operate, a perfect fiducial cross may appear as indicated. If a mask is found which will zero the outlined word, then this mask will predict the masks necessary for the two words immediately below and to the right of this word. The complete area will then be scanned and a check on the number of zero words made. Background noise may be dealt with by accepting a number of non-zero words. For example, consider a cross that has been detected by twelve consecutive scan lines, if twelve lines were found to be initially non-zero, and after masking three or four lines still remained non-zero, then it may be accepted as a cross. Such criteria can only be established by experimentation - a stage that the present programs have not yet attained. It is intended to convert information to coordinates only when the necessary pattern has been recognised, thus conserving computing time.

Considerable effort is being concentrated on the problems associated with crosses; it is essential that this work precedes work
on spark recognition since the sparks are located from data derived from the fiducial marks. Clearly the method of pattern recognition described above may be extended for use with sparks, but it is probable that simpler methods will be devised in this case.

7. Further Development

A new version of Lucicle will be developed when the first one, described above, is successfully put into operation. It will incorporate one or several of the following improvements:

a) triggering of the beginning of the frame scan by the computer,
b) preset switching to select one or several scan line densities, corresponding to the information density on the picture,
c) positioning of the scan line, under computer control,
d) computer controlled selection of "active elements" on the scan line,
e) scanning in two perpendicular directions.

With these modifications the computer, the scanner and the experiment to be evaluated can be matched together most efficiently.

Reference:

Figure captions

Fig. 1  Schematic block diagram of the Lucile system.

Fig. 2  Format of data transferred to the IBM 7090.

Fig. 3  Schematic information array in the core store of the computer, derived from a perfect fiducial cross and prepared for application of the pattern recognition program.
### X-word

| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x₁ | x₂ | ... | ... | x₁₀ | x₁₁ |

- **X-Flag**: Number of scan line (X-coordinate)

### F/S-word

| 0 | 1 | 0 | 0 | 0 | 0 | f₁ | f₂ | ... | ... | f₁₀ | f₁₁ | s₁₇ | s₁₆ | ... | ... | s₂ | s₁ |

- **F-Flag**: True F-coordinate
- **S/S-word**: Content of shift register stored in a "Lower half"

### S/S-word

| 0 | 0 | s₁₇ | s₁₆ | ... | ... | s₂ | s₁ | s₁₇ | s₁₆ | ... | ... | s₂ | s₁ |

- **S-Flag**: Content of shift register stored in "Upper half" (always > 0)

- **S₁₇ ... S₁**: Stored in a "Lower half" is necessarily > 0 at the beginning of a "batch", > 0 otherwise.

- The "batch" ends with an F/S-word containing the true F-coordinate of the point whose content is stored in S₁₇ (location 19) + 17.

#### Example

<table>
<thead>
<tr>
<th>First word</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>425</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Second word</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 0 0 0 0 0 0 0 1 1 1 0 0 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>459</td>
</tr>
</tbody>
</table>

(End of the batch)

---

**Figure 2**
Figure 3
LIST OF PARTICIPANTS

Lawrence Radiation Laboratory, Berkeley

D. HALL
W. HUMPHREY

Brookhaven National Laboratory, Upton L.I., N.Y.

W.F. BAKER
J.W. CAKIN
P.L. CONNOLLY
P.V.C. HOUGH
G. RABINOWITZ

Pennsylvania University, Philadelphia

V. HAGOPIAN
E. LEBYO
J. NIEDERER

Columbia University, Irvington

D. BURD
D.H. TYCKO

Princeton University, Princeton, N.J.

J. BENOT

M.I.T., Cambridge 39

H. RUDLOE

N.I.R.H.S., Chilton

J.W. BURREN
D. LORD
M.J. MITCHELL
J.A. WILSON

Imperial College, London

A. EDMONDS

Bologna University, Bologna

M.L. LUVISETTO
M. MAESTEIl
E. VACCARI
Max-Planck-Institut, Munich
N. SCHMITZ
G. WOLF

C.E.N., Saclay
M. GRANDIDIERS
H. KASHA
A. LEVEQUE
J.P. MERLO

Laboratoire de Physique Nucléaire, Orsay
C. OUANNES

Collège de France, Paris
M. BLOCH
M. CROZON
P. LEBLOND
C. PICARD
M. SCHIFF

C.E.R.N., Geneva
P.M. BLACKALL
R. BUDDE
Y. GOULDSMIDT-CLERKONT
J.M. HOWIE
L. KOWARSKI
W. KRISCHER
G.R. MACLEOD
S.J. McCARROLL
G.W. MOORHEAD
F. MULLER
B.W. POWELL
M. ROSENBLUM
D. WISKOTT