ERASME - Automatic Processing of Bubble Chamber Photographs

W. Jank
CERN, Geneva

Contents

1. Introduction

ERASME is a system for the automatic processing of photographs of bubble chamber events (ref. 1). It represents a new approach in that it fuses into a single facility the usually independent steps such as scanning and premeasuring, measuring, spatial reconstruction and rescuing.

1.1 Bubble Chambers

Before discussing the system in detail I would like to describe briefly the use of the bubble chamber in high energy physics.

The bubble chamber is an instrument which visualizes the trajectories of fast charged particles; its operation relies on the formation of tiny (0.1 - 1 mm) bubbles along the path of the particles in a superheated liquid. These bubbles provide a "track" which can be recorded in stereoscopic photographs, and show with high accuracy where the particle went. The chamber is immersed in the field of a large electromagnet to provide a means of measuring momentum from track curvature.

The use of a bubble chamber makes it possible to observe interaction of incident particles with the nuclei of the liquid (free protons in the case of the hydrogen bubble chamber). A beam of 10 - 20 particles enters the chamber, and some of these particles interact with the protons of the liquid; for the study of these "events" one needs to determine the curvature and angles in space of all the tracks at the point of interaction (fig. 1,2). Bubble chambers can take one or two photographs per accelerator pulse; as an example, the CERN 2m HBC takes 4-5 million stereopictures (3 views) per year and therefore experiments with several hundred thousand events are not uncommon with this chamber.

1.2 Event Processing Chain

Typical bubble chamber film measuring systems distinguish four consecutive phases during the processing of the film. In the first one, called the scanning phase, an operator inspects the film for interesting events; he then selects those interactions which are of interest. This selection is normally made on the basis of some rather simple topological criteria and very little help, if any, is given by a computer. Attempts to automize this phase were only successful for very simple experiments, but in general the human eye and brain have a much more efficient pattern recognition capability and are much more economic.

In the second phase, called the measuring phase, the selected events are measured by giving precisely some 10 to 20 points along each track image belonging to the event. From this one obtains the necessary information to parameterize the tracks. There are many different systems for doing these measurements, ranging from rather simple ones, which are hand operated, to fully automatic machines, which make very precise measurements at high speed.

Most of these automatic systems use a light beam which is projected onto the film and can be moved very precisely in a controlled way. The light which passes through the film is detected by a photo multiplier tube. Thus a signal is obtained when the light beam crosses the image of a bubble which can be used to trigger the read out of counters giving the bubble position. Therefore, one can convert the interesting events contained in the picture into a stream of digitizings, which can be processed by a computer. The distinction between background (noise, other tracks, etc.) and the interesting information (tracks belonging to the event) is made at several levels. At the lowest level, much of the background noise having different characteristics from tracks is removed by discrimination in hardware. Further reduction of the data has to be done by a computer or special purpose hardware, preferably in real time to minimize off-line data handling. The complex control of all the different units of one machine can again be done by either a computer or hardware. A computer is more flexible but a very fast response is needed in order to control the machine and process the outcoming data in real time.

The way this light beam is generated gives rise to many different machines, where either spots or line elements are generated and moved by mechanical or electronic means using high intensity lamps, lasers or cathode ray tubes (CRT) as the light source.

In the third phase, the spatial reconstruction of the event is performed in order to parameterise the tracks in a spatial coordinate system and check that the measurements done in the different stereoscopic views are consistent when projected back into space. This involves the massive usage of fitting procedures and least squares techniques as well as large matrix calculation, so fast floating point arithmetic and a large memory to store the data are necessary on the computer used for these reconstruction programs.

All the programs involved up to this stage make checks for consistency while processing the measurements. Nevertheless, a certain number of events normally remain, in which one or several tracks were not measured or reconstructed properly. In order to save these events, one normally tries to rescue them in a forth phase by either passing the film through the system again or trying to patch up the event on an interactive graphics system.

After that, the measurement of an event is normally finished and data describing it completely are written onto a file for further processing. This processing includes the kinematic analysis of the interaction and the statistical evaluation of the whole data of an experiment.

2. General Design of ERASME

Any bubble chamber film measuring system has to deal with the problem of completely different computing requirements during the phases described above. In the scan phase virtually no central processor (CP) power is required but this phase lasts a significant length of time. On the other hand, during the measuring phase one needs rather little CP-power but response in real time to satisfy the time critical functions of the measuring unit. On the contrary, during the reconstruction phase, massive allocation of CP power is necessary. The rescuing is then a mixture of scanning and measuring. In addition it was felt necessary for ERASME to provide a good time sharing service for program development in parallel with production.

It was thought that the best solution for a new system would be to have a system in which all these different phases could be combined (ref.1,2); e.g. the operator scans the film for the interesting events, measures them and geometrical reconstruction can be done immediately. Whenever an event fails, one can immediately rescue it. In this way, the film has to pass through the system only once and only acceptable data from the reconstruction phase has to be output. In this way one avoids some of the problems present in traditional systems, where all these different phases have been kept seperated. To pass an event through such a system may take several weeks. Data handling is very complex and management of large amounts of data has to be organised.

The ERASME system consists of several scan and measure tables (or S/M units) each controlled by a minicomputer and all connected into a medium size computer. For a few, special tasks which need very fast data processing, a micro programmed special processor (called ESOP for ERASME Special Online Processor) is placed between the S/M unit and the control computer. The operators sitting at the tables are given several means to interact efficiently with the system, such as displays, keyboards and a 'track ball'. This gives a certain structuring of the components in the system (fig. 6). The computing power can be thought of as being distributed vertically on three different levels, while the S/M units form a horizontal extension of the system. This scheme allows good matching of the CP power and real-time response required in the different parts of the system. The vertical structure is well suited for software development. Thus when one starts programming a certain algorithm, one can code it in a high level language (FORTRAN) and run and test it in the main computer. Once established, this code or part of it can be shifted one level down into the small computer, where it has to be coded in assembly language and runs more efficiently. When necessary, it can be shifted even further down into the micro programmed processor, where it runs faster again, but needs tedious coding with micro instructions. In this way an optimal balance between response time and computer usage can be achieved. It is clear that the less complex a task is and the more frequently it needs to be executed, the lower should be the level at which it operates.

3. Hardware

3.1 Scan and Measure (S/M) Unit

Each S/M-unit is made up of the following parts (fig. 3, ref. 5):

- the optical and mechanical part (fig. 4)
- the precision CRT and track detection unit
- the digitizing logic and scan control
- the stage and film transport system with its control
- the operator's table with displays, keyboards and a track ball.

3.1.1 Optical and Mechanical Part

The main structure carries a lower stage on which four filmgates and a calibration grid are mounted. This lower stage is used to move the views under either the measuring or projection channel. Within the base are mounted the condensor lens and photo multiplier tube for the measuring channel as well as the light source for the projection channel. A bridge structure over the base carries:

- the CRT-unit with coils, their mounts, shielding and four reference photomultipliers;
- the large aperture, high precision lens with a mirror to bend the axis of the measuring channel into the vertical;
- the x and y-stages which carry two lenses for the projection channel (small and large magnification);
- the first mirror of the projection channel.

A second, much larger mirror, to reflect an image of the film onto the operator's table, is suspended from the ceiling.

3.1.3 Precision CRT and Track Detection Unit

The CRT-unit is based on previous developments at CERN. It is used to generate a light spot whose position is defined at any moment by the contents of the deflection counters in the digital scan control unit. Very precise 16-bit D/A converters in the deflection control unit convert the digitally defined positions into voltages which are inputs for the deflection drivers. These are highly stable voltage-current converters that control the currents in the deflection coils. Dynamic focus and astigmatism correction is used to maintain the spot to a size of about 15 microns over the whole scanning area of the screen. The necessary correction currents are generated by the spot shape control (SSC) unit. The values for the correction at 81 points are stored in 9 x 9 matrices of potentiometers and 2-dimensional linear interpolation is used between these values.

The spot on the screen of the CRT is focussed by a large aperture high precision lens onto the film. A condensor lens projects the light which traversed the film onto a photomultiplier. The signal from this photomultiplier is treated in the video amplifier and track detector. Inputs from four reference photomultipliers are used to compensate for the variations of the light output of the phosphor. Normalized track pulses are fed to the digitizing logic.

3.1.3 Digitizing Logic and Scan Control

The measurements are made by moving the CRT-spot rapidly along a series of straight parallel lines covering a small rectangular area. This is called a "slice scan" (fig. 5). All parameters of this slice scan, such as origin, orientation, number of lines, length of lines, line spacing, spot speed, etc. can be set by the control computer in registers of the scan control logic. The digitizing logic then generates from normalized track pulses the position and width of a hit in a local coordinate system. These values are written into the control computer's private memory by direct memory access (DMA) or are picked up directly by the microprogrammed processor.

3.1.4 Stage and Film Transport System with its Control

The film transport system has been designed for low speed only. It needs approximately one second to move the film one frame. This is adequate for the planned mode of operation, since scanning implies that the operator has to examine the film, frame by frame, spending a significant time to examine it each time the film stops. The stopping accuracy is of the order of 0.5 mm and either 50 mm unperforated film from the CERN 2m HBC or 70 mm perforated film from BEBC or Mirabelle can be used. There are three servo-controlled stages (x, y and lower) on the machine. The lower stage carries four film gates for up to four stereoscopic views and a special gate for a calibration grid, which is mounted all the time. Film transport and all stages can be controlled by the computer. In another mode of operation, the x and y-stage positions can be controlled in a non-linear fashion from the track ball. For the film transport manual control is provided for loading and unloading the film.

3.1.5 Operator's Table with Displays, Keyboards and Trackball

The operator station consists firstly of a large table, onto which the film can be projected, together with a reference mark in the form of a bright cross. A track ball allows the operator to move the projected image and to point with the light cross to different items on the picture. For convenience two different magnifications are available which can be selected by means of a button. There is a function keyboard with a numeric keyboard incorporated, as well as an alphanumeric keyboard. To inform the operator how far the measurement of an event is advanced, to show him the result and to inform him about problems found by the system, there are two storage scope displays. On these, messages with different character size in both normal and bright characters can be written as well as pictures consisting of points and vectors. Tracks found by the program can be plotted on top of digitizings, which allows the operator to check all the automatic measurements. A cursor can be connected to the track ball and its position can be read out when any keyboard button is pushed. The buttons on the functional keyboard can be individually enabled by the program and illuminated to show the operator the available choice of actions.

3.2 Computer System

3.2.1 Main Computer (DEC System 10)

A DEC-system 10 was chosen as the central computer (fig. 6). At present it consists of a KI-10 central processor with paging hardware, 256 K 36-bit word core memory (1 μs) which can be up to 4 ways interleaved, three 5-million words disc pack drives, two magnetic tape units, three DEC-tape units and a fast line printer. To provide access for the users a line multiplexor at present connects some seven display terminals, five Teletypes and one special line per S/M unit.

3.2.2 Control Computer (PDP-11)

These minicomputers are either PDP-11/20's or PDP-11/45's with 8 K 16-bit word memory each, but no standard peripherals of their own. On the PDP-11/20's, only core memory is used, whereas on the PDP-11/45's faster MOS memory is used. To interface the different hardware controls, a special bus is connected to the standard UNIBUS of the PDP-11's.

The PDP-11 has a single bus, the UNIBUS, which connects to both memory and peripherals and allows them to be accessed in an identical way by the CP. Interrupts may be on one of four hardware levels and are directly vectored through the first 1000 bytes of memory. In this way the vector addresses and the priority on which the handler routine will run may be set by software.

3.2.3 Microprogrammed Special Processor (ESOP)

This processor, called ESOP (ERASME Special On-line Processor) was completely developed and built at CERN (fig. 7). Its initial design goal was to perform the histogramming of the individual digitizings in real time. This means it has to be fast enough to update the four quantities stored per histogram bin in the time normally taken to transfer the coordinates to the PDP-11 (about two microseconds). It consists of four main units:

i) a data memory of 256 16-bit words which can be made to appear like ordinary PDP-11 memory for communication purposes. When used by the microprocessor it has a read/write cycle time of 45 nanoseconds.

ii) an instruction unit which has its own memory of 256 words, each 48 bits wide (both memories have their own arithmetic unit for address calculations).

iii) an arithmetic unit which will perform arithmetic and logical operations on the contents of two 16-bit registers and is also able to perform logical shifts or rotates on either of these registers on its own.

iv) a multiply and divide unit which will perform 32-bit operations in about 2 microseconds.

The cycle time of the machine, except for unit four, is 70 nanoseconds. There are two main buses: a data bus which conveys all information between the data memory and the arithmetic unit and also carries all incoming and outgoing data. The instruction bus carries all control and command functions for each cycle. To obtain the necessary speed of operation the processor was designed to execute several functions concurrently; the four main units will normally all work at the same time, and the address calculations in the memory units overlap with the read or write of the previous cycle. A conditional skip or jump system makes four way branching on a single loop instruction possible. External quasi interrupt flags are included to initiate a routine or signal the end of the incoming data to be processed.

The next task to be implemented on the microprocessor is a histogram scanning routine which will calculate master points in a local coordinate system. Further tasks may be performed in this processor, such as scaling and coordinate transformation for the on-line displays or execution of short time critical routines.

Programming of this processor has to be done on a very basic level. Each instruction is made up from 48 bits which must be separately set up. As many bits have up to four or five different functions, there are numerous situations where clashes occur. To ease the programming, an instruction set, consisting of 31 mnemonic operation codes, has been defined which encompasses all functions of the processor. About half of these OP-codes are used to set up bits which control data paths, the rest normally have associated parameters to specify the way in which the data is actually processed. Each complete 48-bit instruction is then made up from sets of these OP codes. A cross-assembler running on the DEC System 10 has been written which produces 48-bit micro-code directly from programs written in these OP-codes. It also makes checks for instruction compatibility, clashes and timing.

Via the cross assembler, a set of programs have been written to test microprocessor functions and a control package has been written for the PDP-11 to load and run the microprocessor. From experience gained on the prototype, which is now able to histogram digitizings, a number of improvements and additions will be made to the second and subsequent machines.

3.2.4 Special Interface, DEC System 10/PDP-11

The PDP-11's are connected to the DEC System 10 through parallel interfaces which have been designed to meet the specific needs of the ERASME system (ref. 3). Each of these interfaces allows the connected PDP-11 to map its addressing space into DEC System 10 memory (fig. 8). This is achieved in the following way. In each interface two sets of address switches define lower boundaries of two UNIBUS address ranges (windows) to which the interface will respond. The sizes of these windows and their positions in the PDP-10 memory are defined by two registers loaded from the DEC System 10, so that the two memory windows can be set independently for each PDP-11 to point to arbitrary memory areas in the DEC System 10. One of the windows is used as ordinary read/write memory, whereas the second window appears as read-only and can be used to share code for the different PDP-11's (this code then exists only once in DEC System 10 memory). A 16-bit

PDP-11 word is packed in each half of the DEC System 10 36-bit word.

The actual communication between the two processors is done by first depositing data as task parameters in the communication window and then sending an interrupt request to each PDP-11 on any of its four bus request levels and for any vector address 0 to 744_8 to trigger a specific task in the PDP-11. The PDP-11 then in turn can interrupt the DEC System 10 processor on an appropriate PI-level, provided the DEC System 10 has enabled it first.

4. Software

4.1. General

As already described, a certain structure of the system hardware was adopted which must be reflected in the software. There are several, sometimes even contradictory requirements, which have to be met by the software. For example, during the scanning phase, which is rather long, virtually no CP time is required but the operator wants a fast response on any action he takes. This means that this particular part of the program has to be available in the system in a way that it responds quickly to any requests without using the resources (e.g. CPU, memory channels) of the computers more than necessary. During the measuring phase the program typically needs only relatively little CPU time, but the program doing the pattern recognition and data reduction is usually rather big (typically some 30K words, including buffers) and has to respond in real time to the requests of the measuring machine in order to finish the measurement of an event as quickly as possible. During the reconstruction phase, a very big program (40-50 K words) needs a lot of CP time for all the floating point calculations. During the rescue phase, the load of the computer looks like a mixture of the scanning and measuring phase, where the operator looks at the results for a while and then wants to measure a single track and so on.

Because software exists on three different levels, namely the main computer, the control computer and the micro processor, one has to optimize the distribution of the code as well as the communication between the machines to achieve maximal use of the available hardware and operators. The software also has to allow the simultaneous operation of all S/M-units for different experiments requiring different modes of operation. Because of the long time scale of the whole project, one also has to allow for production and development work to go on concurrently.

4.2 System organisation

The main computer (fig. 6.) is provided with a terminal oriented time-sharing operating system which also has extensive file handling capabilities and allows a privileged user to handle his own real-time devices. A scheduler allocates the system resources such as CP, memory, channels and I/O devices. It also does swapping of programs onto discs when memory space is needed for other users. A program is brought back into memory when either space is available again or the scheduler decides to run it (ref. 4.). The CPU contains paging hardware which the current operating system (OS) uses to allocate single pages of user programs to arbitrary physical memory locations in blocks of 512 words. Every user job consists of one or two logical segments, the low and the high or shared segment. The latter can be write-protected and therefore can be shared between several users.

Most of the system software makes use of this feature. For example, there need only be one copy of a compiler working as a shared high segment for several users at the same time. Each of these has his own low segment, where all private data are kept. The OS also allows one to keep one or both segments of a user job in the same place in memory, called "locking" it in core. In hardware, communication between user programs and the OS is done by means of unimplemented operation codes, so called monitor UUO's which transfer program control to a fixed address in the monitor addressing space.

In the control computer there is no monitor in the usual sense. Each basic task is directly triggered by a vectored interrupt from the DEC System 10 and also responds to interrupts from the ERASME hardware. Priorities are treated by the natural hardware interrupt levels in the PDP-11. A real time clock is used to time out hardware malfunctions and retrigger repetetive tasks. A task sequencer allows the chaining of basic tasks to perform more complex functions.

At present there are about 20 tasks implemented, ranging from very simple ones like setting up slice scan parameters or transferring title values from the DEC System 10 to rather complex ones like track segment following or fiducial finding.

4.2.1 Current implementation

To meet the special needs of the different phases described above, and to allow normal time-sharing together with production, it seemed necessary to split the production program into suitable modules, each of which is dedicated to a special function. In such a scheme, easy communication between different modules is absolutely necessary. In addition in order to avoid problems with standard software and OS updates no significant OS changes should be necessary. This was very easily done under the OS available on the main computer. By making all the different modules user jobs, it is possible to arrange communication between each of them by a shared high segment containing all data particular to one table. Control is passed from one job to another by a simple mechanism in the standard OS called "wake" and "hibernate" - UUO's. By this, a user job can simply issue a wake request to another job and hibernate itself. The scheduler in the OS passes control to the woken job, which can then attach itself to the high segment of a table and work with the data found there. When it has finished its tasks it in turn can wake the next job and hibernate.

The modules are mainly written in FORTRAN, except for a few routines either for setting up data for PDP-11 tasks, where packing has to be done, or for short tasks which run frequently. These are written in the assembler language. As all the table dependent data are contained in one high segment per table, these code modules, which are low segments, may be used for several tables. This is possible only in a sequential fashion as the current FORTRAN compiler does not produce reentrant code. This means that one module can serve different tables but only at different times.

To do this allocation of modules to the S/M units, some tables describing the present status and availability of each module have been added to the OS. This mechanism allows the setting of a "busy" bit in an interlock table when one particular module is scheduled for an S/M unit. When the module returns control the bit is cleared and the module becomes available again. In addition there are sequence tables allowing a particular S/M unit to queue for a sequence of modules. All requests for service from a single module or a sequence of modules are entered into these tables from control jobs by means of OS calls. These tables also allow the provision of several identical copies of one particular module. In case one module is already busy for another S/M unit the next free module of the same kind can be allocated to the requesting S/M unit. By these means it is possible to avoid bottlenecks in the system for either time critical modules or modules which are used for a rather long time by one particular S/M unit.

As already mentioned there is one control job per S/M unit, which determines the mode of operation of this unit. So by changing the sequence of module calls one can adapt the mode of operation easily to what is needed for the experiment being processed on this table. As the program flow is often very similar for different experiments, the operation of a module is steered by data read into the high segment private to each S/M unit. Modules particular to one experiment or to one S/M unit can be declared private for that unit so when an experiment requires special techniques a module dedicated for this can be provided. When the programmer is debugging new routines, he can just "plug in" his changed module for one S/M unit and debug it there. He also has to reload only a small part of the programs, namely the changed module, which saves time for the programmer and on the computer.

To better define the interfaces between the different modules, the HYDRA system (ref. talk at this school .6.7.), developed at CERN was used. For the pattern recognition part, only the titles-package and the memory manager were used. This allows the definition of a dynamic data structure as input to and output from every module (which is something like a processor in HYDRA terms). So the programmer can change the code in a module as much as he wants and can still use the other modules as long as he does not change these defined input and output structures. This gives him a great deal of flexibility and modularity.

The pattern recognition and data reduction part of the programs (fig. 9.) is split in this way into eight different modules, namely a control job which mainly contains the steering of the S/M unit, an initialisation job which initialises data in the high segment, and then six functional modules which are a scan job, a fiducial job, a track initialise and follow job, a vertex job, an interaction job and a job to write the output. The geometrical reconstruction program can be split into up to five different modules, where the initialisation part is always split up. The rest can be split in a point match and reconstruction a track match job and a final fit job. The functioning and the methods used in the individual parts will be discussed later on.

There is one additional job per S/M unit which controls the interface to the PDP-11 of this unit. This job, the "Link Job", which is locked in core, contains the code to service all program interrupts arriving from the PDP-11. It first initialises the interface to the PDP-11 by setting up the two relocation and protection registers which point to a data communication area (DCA) and a common code area (CCA) respectively. Then the interface is enabled to interrupt both the PDP-11 and DEC System 10 and the PDP-11 private core is loaded by putting the code read from a disc file into the DCA, where a simple, absolute loader running in the PDP-11 can pick it up and transfer it into PDP-11 private memory. In addition, it can contain the CCA loaded with special code for the PDP-11 of this unit. When a module wants to "talk" to a PDP-11, it first puts input data for the task to run in the PDP-11 into the DCA, then sends an interrupt to the appropriate interrupt vector in the PDP-11 and hibernates. The PDP-11 executes the task, puts its output into the DCA and sends back an interrupt to the DEC System 10, which activates the interface control job. This job handles the received interrupt and passes control back to the module originally requesting the execution of the task by sending a wake UUO to it. The module finds the result of the task in the DCA and can proceed to use it.

It was already mentioned that the PDP-11's can share code which is loaded in the DEC System 10 memory. This is implemented by two more jobs locked in core, one job for the PDP-11/20's and one for the PDP-11/45's. The Link Jobs for PDP-11's using this code set the interface relocation register for the CCA to point to this area in DEC System 10 memory. If a particular PDP-11 does not use this shared code, additional code can be held in a private CCA by the Link Job.

To handle the large number of jobs which are present in this sort of system when all S/M units are running, a modified version EROPS of a standard system program OPSER is used. All necessary console commands to start the total system or one particular S/M unit can be put in a file. The file is read by EROPS which then executes the commands.

4.2.2 Future Plans

It is clear that one gets a certain overhead in the OS due to the large number of jobs necessary to run all S/M units in production. In addition, the fact that the code is not properly reentrant leads to the duplication of certain parts of it, which occupy additional memory and have to be swapped in and out.

It was felt that the system could be made more efficient and easier to handle if reentrant code could be used. In fact there is a new FORTRAN-compiler available from the manufacturer which produces reentrant code, again using the two segment feature of the DEC System 10. The code and space for local data and variables are generated in such a way that they can be loaded into different segments. So, the code can be loaded into a high (pure) segment and be write protected. This segment can then be shared between several users. All the data, like constants, variables, buffers, etc. can be loaded into a low (impure) segment, different for each user. Another feature, available with the next OS release will be user demand paging which will be very useful for our application. This means that a user can himself specify at which moment he wants certain pages of his program to stay in core or to be swapped out. Hence parts of a program (data or code) can be swapped out from memory.

If we proceed with these ideas we will reverse things with respect to what we have now. That means that all code of the pattern recognition, data reduction and reconstruction programs, which will become reentrant, can be loaded into one high segment, shared between the S/M units. All data, buffers, title values, etc. and the code which determines the flow of the program (the old control job) which are different for the individual S/M units can be loaded into a low segment, together with the DCA and the code to handle the interrupts from the PDP-11's. In addition code to be debugged can be loaded into the low segment which is private for that S/M unit. In this way there is only one job per S/M unit. If priority should be given to any particular S/M-unit, one would only have to raise the priority of the corresponding job in the DEC System 10. Buffers of sometimes considerable size, not being used through a certain phase, can be swapped by the user demand paging and the memory freed for other programs. However, every single program modification would imply reloading the whole program, which is certainly a disadvantage.

4.3 Application Software

Many of the methods and algorithms used in the ERASME system are well established and have proved effective in other systems. The programs for the pattern recognition and data reduction have been especially written for the ERASME system, whereas only modifications have been made to the CERN standard reconstruction programs to run them in an on-line fashion.

4.3.1 Calibration

The position of the spot on the screen of the CRT is not a linear function of the currents applied to the deflection coils, the main effect being a pincushion distortion. A calibration is used to remove this non-linearity from the measurements done with the CRT. This is done by measuring crosses on a very accurate glass grid with the CRT. By mapping these measurements onto the known positions the coefficients of a 5th order polynomial in x and y are calculated. A typical distribution of residuals of such a fit peaks at around one micron with a maximum value around three microns (fig. 10).

4.3.2 Pattern Recognition and Data Reduction

There is one basic method, called histogramming, which is very often used for the pattern recognition in the ERASME System. It consists of projecting a set of digitizings onto a line at a given angle, which is divided into equal sections, the "bins". Counting the digitizings in all bins gives a distribution, the histogram. A "pulse" is defined as a series of consecutive bins with contents greater than a threshold and a total contents not less than a given value, the width not exceeding a given limit. This is a very powerful method of identifying a string of digitizings forming a straight line, removing most of the background. It is simple enough to run even in the ESOP.

At present the system is based on vertex guidance (fig. 11). That is a human operator feeds the positions of the primary and all secondary vertices into the machine by measuring them on the scanning table. When the machine comes to measure an event, an operator has already indicated that there is an event on the frame and where it is located. It is up to the machine now to identify and measure this event as well as possible. On ERASME the operator can also give additional help to the programs by pointing in a clear region at tracks belonging to the event (crutch point), by pointing to tracks which should be deleted if found (anticrutch point), or by giving end points of tracks. An end point means pointing to the end of a track or the place where the image of the track becomes confused with other track images. For difficult tracks the operator can even give two or more points on the table, thus defining a curve along which the track will be precisely measured later. For all these measurements the projected image of the film is moved by the track ball relative to an illuminated cross.

When the operator has finished all operations on the optical projection, the view is moved under the measurement channel by displacing the lower stage. The stopping position of this stage can be read from a precision encoder and so all the measurements in the optical channel can be transformed into the CRT-coordinate system. Firstly, the system needs to measure reference marks, so called fiducials. Because these marks are engraved in the chamber glasses or glued on the chamber walls, the positions of these marks do not change and they are known to a very great precision. They are used later to transform all measurements back into a reference plane in chamber space.

The fiducial finding program now tries to find one reference fiducial in a fairly large search region. Having found this first mark the position of the other marks can be predicted fairly well. To measure one fiducial mark, the program makes one slice scan with the scan lines approximately bisecting the larger angle between the fiducial arms. The digitizings from this scan are then histogrammed twice at the angles of both fiducial arms. A straight line fit through the digitizings falling into the nearest pulse to the prediction is then made, giving an improved angle for a second histogram. A final fit, of second order when the fiducial arms are curved, is made and the intersection of the two straight lines or parabolas give the final fiducial position. When any problems are encountered by the program, for example not enough fiducials are found, it immediately becomes interactive, showing the operator what it has found and leaving the choice to him of what to do next, such as restarting everything or measuring the missing fiducials by hand.

Having finished with the fiducials, the program proceeds to measure the tracks. From the measurements on the projection table, the approximate positions of the vertices are known and the program first tries to identify and initialize the tracks emanating from it. This is done by using a search pattern of very narrow slice scans in the form of full or partial octagon pairs around each vertex (fig. 12). The scan lines are parallel to the octagon sides so that the intersection with tracks coming from the vertex is approximately normal. The digitizings of these slice scans are histogrammed and masterpoints are calculated as the centre of gravity of the histogram pulses. Masterpoints from the two different octagons whose distance does not exceed a given value are considered to define possible track candidates if their connecting line points roughly to the vertex.

It is checked whether a track lying close to this track candidate was already followed. If not, track following is started towards the vertex and only when this track comes relatively close to the vertex is it also followed outwards. This procedure of initializing tracks can be repeated at up to five different radii so that the chance to pick up even a confused track is very much increased.

Track following consists of making slice scans at positions calculated from points already measured along the track. The digitizings from every such slice scan are then histogrammed. A centre of gravity of the digitizings in each histogram pulse is calculated and that nearest to the predicted track position is taken as a measurement and is called a masterpoint (ref. 11).

We are convinced that a very high percentage of track segments are easy to follow and do not need very sophisticated algorithms. In particular no floating point arithmetic is needed. Only when this simple track follower fails for any reason should a bigger effort be made using a more complex algorithm. Thus it is possible to implement the relatively simple track segment follower in the small control computer and only to give additional help to this by more powerful methods programmed in the main computer.

The track segment follower contains two different methods of predicting the next slice scan. As starting values two points and a direction are required. It first makes a straight line extrapolation and puts a slice scan not far from the second point. If no master point is found by this, the slice scan is shifted onwards half a slice length at a time up to four times. If this still does not give any acceptable master point, the track candidate is given up, otherwise the new masterpoint is used to predict the next slice scan along a circle through the last points. This is repeated along the track and, with increasing confidence, the skip distance between the slice scans is increased for every new master point. The prediction along the circle is done with an iteration formula, using two points and the two tangents at these points (ref. 12). This track segment following is done until it is stopped by:

i) reaching a given end point;

ii) needing to switch the coordinate system in which the predictions are made. This is done at about 45^{0} to avoid problems with tangents of angles;

iii) finding no more master points in four consecutive slice scans.

In case (i) the segment is possibly added to already existing segments and stored away as a track measurement. Case (ii) just needs switching of the coordinate system in the main computer, then track following can be resumed. Only in case (iii) is the higher level track follower in the main computer involved. One might be either at the end of the track or the track follower may have made a wrong prediction because of one or more wrong points. In either case the last points are thrown away and a circular least squares fit is made through the remaining last six points. With this an extrapolation beyond the dropped points is made and track following is reinitialised. If this does not find any new points after four slice scans, the track is terminated and stored away. With this restart feature, many tracks are saved and also the end of a track is defined much better. Tracks can be followed through any turning angle until a given maximum azimuth or track length is reached.

After a complete track is followed, an attempt is made to remove bad measurements and to detect kinks. The program also tries to match special points given by the operator, such as charged vees, stop, end, crutch and anticrutch points to the track. Whenever a charged vee, stop or end point is matched or a kink is detected, the track is cut. When an anticrutch point is matched, the whole track is deleted, but the track parameters are kept so that the track is not reinitialized. The number of master points is reduced to a given value by weeding out points in such a way as to give as uniform a distribution along the track as possible.

If there are any unmatched crutch points left when track initialization from the octagons is finished, the program tries to initialize tracks from these points. Four complete "octagons" are made around each point and a track candidate is chosen by putting the master points from these octagons into "roads". The road taken is the one with a maximum number of master points within a minimum distance of the original point.

Having finished automatic track initialization and following, checks are made to eliminate identical tracks. This is made by sliding a parabola along a track, checking at the same time the distance of points of any other track which has approximately the same azimuth angle at the vertex. When a certain number of points is found to fall within a minimum distance, the tracks are declared to be identical and the shorter one, or the one without a crutch point is deleted.

With the remaining tracks the program tries to improve the vertex position (ref. 9). Intersection points of circle fits through master points near the vertex are put into classes. One class is formed by the set of intersection points where the distance of the points to each other does not exceed a given value. The new vertex is calculated as the average point of all points in this class including the manual measurement of the vertex. The distances of all tracks from this vertex are checked and the tracks too far from it are deleted.

The remaining tracks are now shown to the operator on one of the display units (fig. 13). He can request displays of the result at different magnifications. He can add missing tracks by giving crutch points on one of the display units. He can add one or several points to already existing tracks. He can also delete complete tracks, parts of a track or just single points. The operator has to check and, if necessary, to patch up the measurements. The program always tries to help him by writing messages or warnings on the display units.

In this way all vertices in a view and all views are measured in turn. At the end of this phase the event is presented to an on-line geometry program which performs the spatial reconstruction of the event.

4.3.3 Geometrical Reconstruction

The main tasks of the geometrical reconstruction (ref. 7) of an event are:

i) fiducial handling;

ii) point match and point reconstruction;

iii) track match and track reconstruction with the possibility of mass dependent fits.

For each view, vertex and track, measurements must be transformed to a reference coordinate system, defined by the expected positions of a number of fiducial marks on a plane parallel to the film plane. The transformation is linear and is determined by matching measurements of the fiducial marks to their expected positions.

Given a number of views where vertices have been measured, point match tries to find the vertex associations, i.e. to identify the vertices on different views which are images of the same space point. Where there are points close together it may be necessary to resort to a final point fit in order to resolve the ambiguities which arise. For this reason, and because it requires relatively little computing time, the final fit, with minimization in the film plane, is performed for each space point candidate. With the approximate coordinates of the position of a point in space ($X_1X_2X_3$) a least squares fit is made by minimizing, with respect to $X_1X_2X_3$, the sum of squares of deviations in the film reference planes between the measurements and the projected points.

It is the task of track match to associate those images at one vertex which correspond to the same track in space. Furthermore it should eliminate any spurious tracks not concerned with the interaction or decay and should make sure that the resulting set of tracks in space is unambiguous.

The solution to the problem can be rather trivial when the topology is simple and when the measurements are clean. It may, however, become quite complex for high multiplicity events where not all tracks have been measured, or where some spurious tracks are present. The basic steps are the following:

- Lists of track images to be considered are set up and crude film plane parameters (circle fit) are computed for each image.
- The space to film transformation near the vertex are used to derive candidate multiples, together with a first estimate of their space parameters.
- This first approximation is employed in the reconstruction of near-corresponding points along the first section of the track. These points, in turn, are used to improve the track parameters so that, when necessary, more near-corresponding points can be reliably computed further along the track.
- Ambiguities are diagnosed and resolved from the results of the near-corresponding point computation.

After that, a final fit to the track trajectory, usually mass dependent, can be made. A steering processor decides which masses are required for the track and judges the success of the fits. Track parameters for a helix fit and the various mass fits are output.

5. Status

The ERASME project was started towards the end of 1970. The main computer with a KA10 central processor and 96 K words of core memory was installed by August 1971. The first S/M unit was completed in May 1972.

By now three S/M units are ready. Two of them are being used to measure film from the CERN 2m HBC and BEBC, while the third one is used for program and hardware development. The main computer has been upgraded as planned and a KI-10 central processor with 256 K words core memory was installed. The construction of SM4 and SM5 is progressing on schedule. SM4 will be completed in autumn and SM5 by the end of the year.

References

1. D. Lord and E. Quercigh, "The ERASME Project Summary", CERN-DD/DH/70/20, September 1970.

2. The ERASME Group, "ERASME: A Scanning and Measuring System for Large Bubble Chamber Photographs". CERN-DD/71/19, October 1971.

3. J. Bettels et al., "A High Performance PDP-10/PDP-11 Link", Proc. of the 8th DECUS Europe Seminar, September 1972.

4. The ERASME Group, "The Use of a Standard Time Sharing Operating System to Control a Complex of Real-Time Processes". CERN-DD 72/26, October 1972. Proc. of DATAFAIR 73, Vol. II pp. 364 - 369. April 1973.

5. The ERASME Group, "Description and Status Report of the ERASME System". Proc. of the Oxford Conference on Computer Scanning, April 1974 (in print).

6. The HYDRA-System manual. Available from CERN/TC Division.

7. The HYDRA-Application Manual. Available form CERN/TC Division.

8. P. Zanella, "Machine Recognition of Patterns in Particle Physiscs", Proc. of the First European Conference on Computational Physics, April 1972, 63-74.

9. W.W.M. Allison, F. Beck and J.G. Loken, "POLLY II: A Complete System Incorporating Facilities for Measuring Bubble Chamber Film without Prescanning", Vol. I, II. Argonne National Laboratory, Argonne, Illinois, ANL/HEP 6916.

10. J.P. Berge et al., "Oxford PEPR System", Proc. of the Int. Conference on Data Handling Systems in High-Energy Physics, Cambridge (U.K.) 1970, CERN 70-21, pp. 61-68.

11. W. Blair et al., "Experience with the CERN HPD Road Guidance System", Proc. of the 1967 Int. Conference on Programming for Flying Spot Devices, Munich, January 1967.

12. M. Ferran et al., "Progress with a Minimum Guidance Program at CERN and RHEL", Proc. of the 1967 Int. Conference on Programming for Flying Spot Devices, Munich, January 1967.

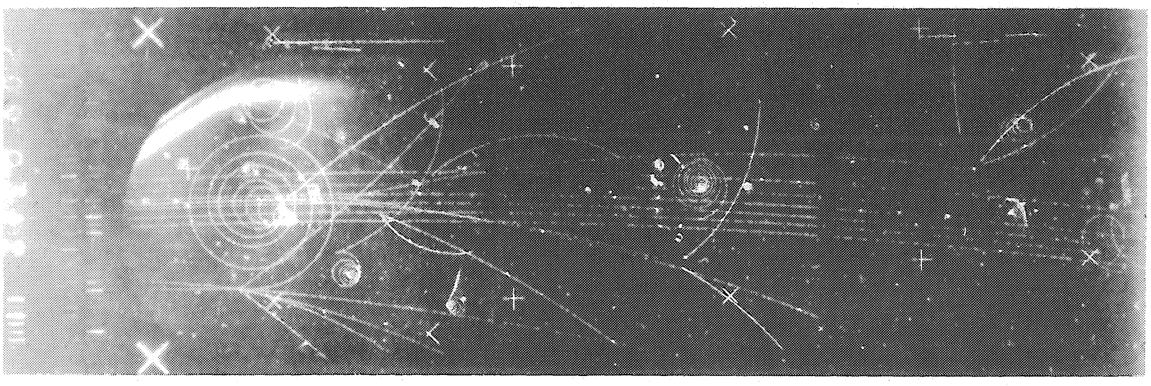

Fig. 1. Picture of CERN 2m Hydrogen Bubble Chamber.

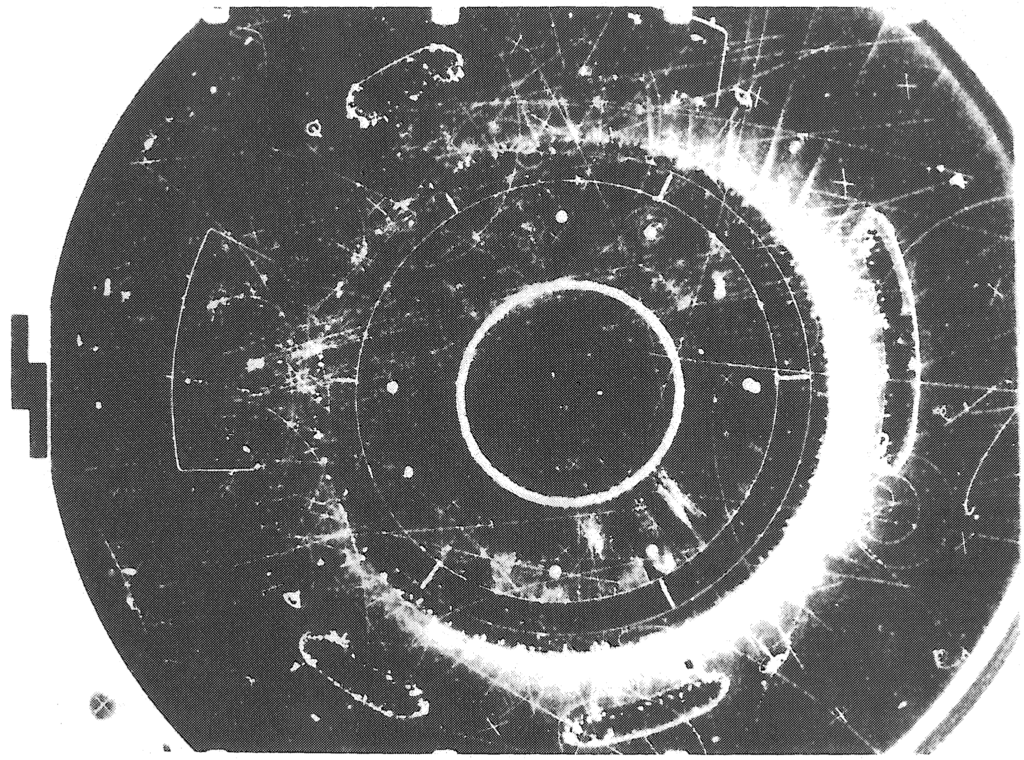

Fig. 2. Picture of BEBC.

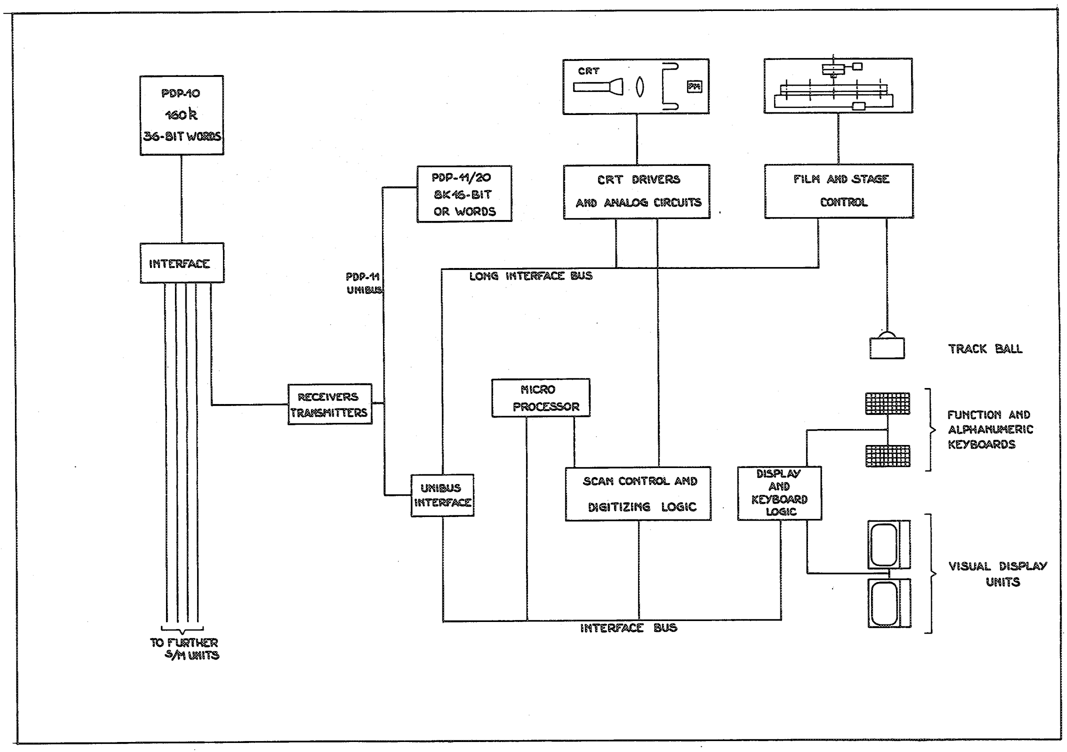

Fig. 3. Block Diagram of an S/M-unit.

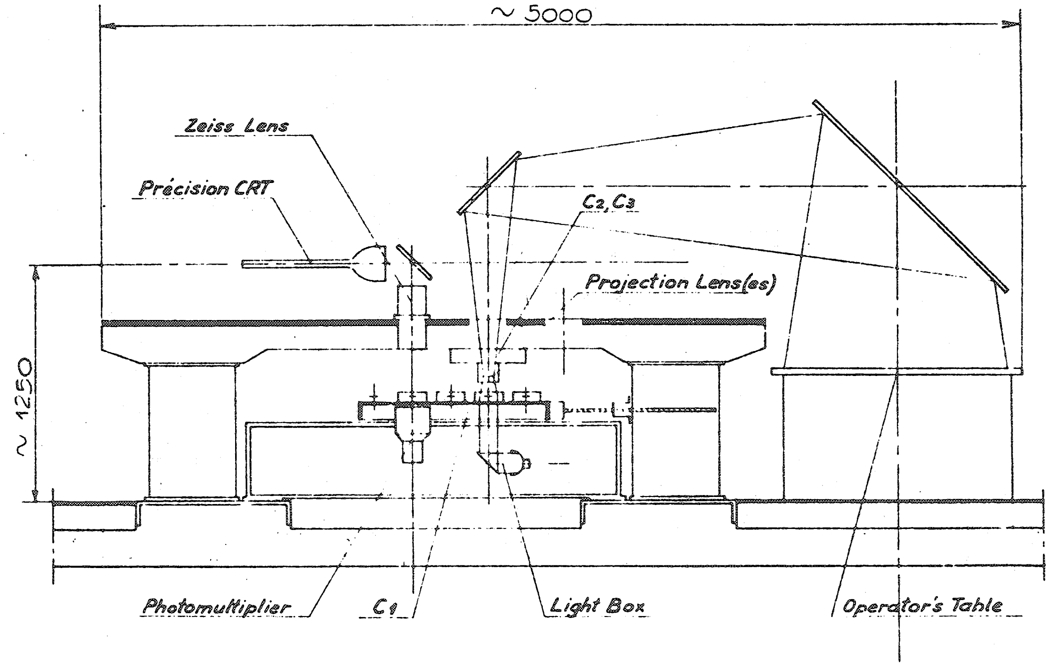

Fig. 4. ERASME, optical and mechanical part.

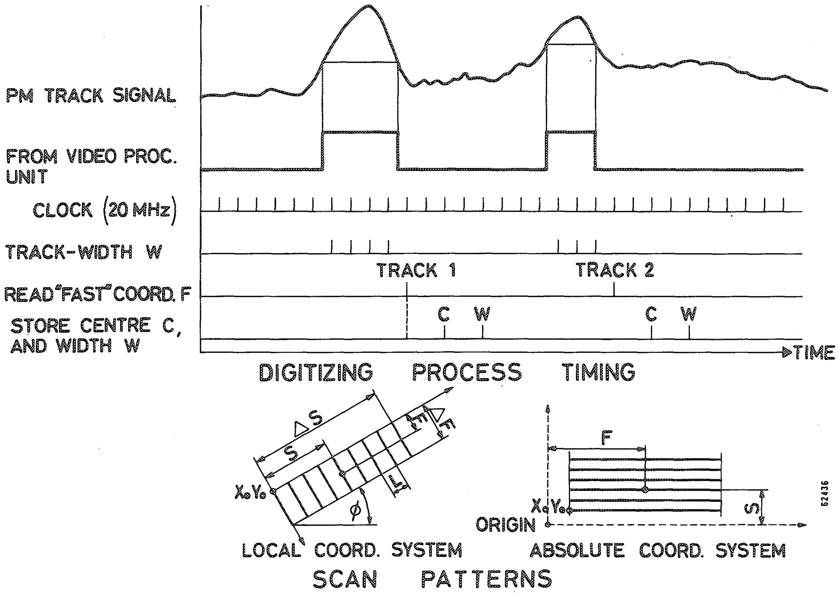

Fig. 5. Slice scan.

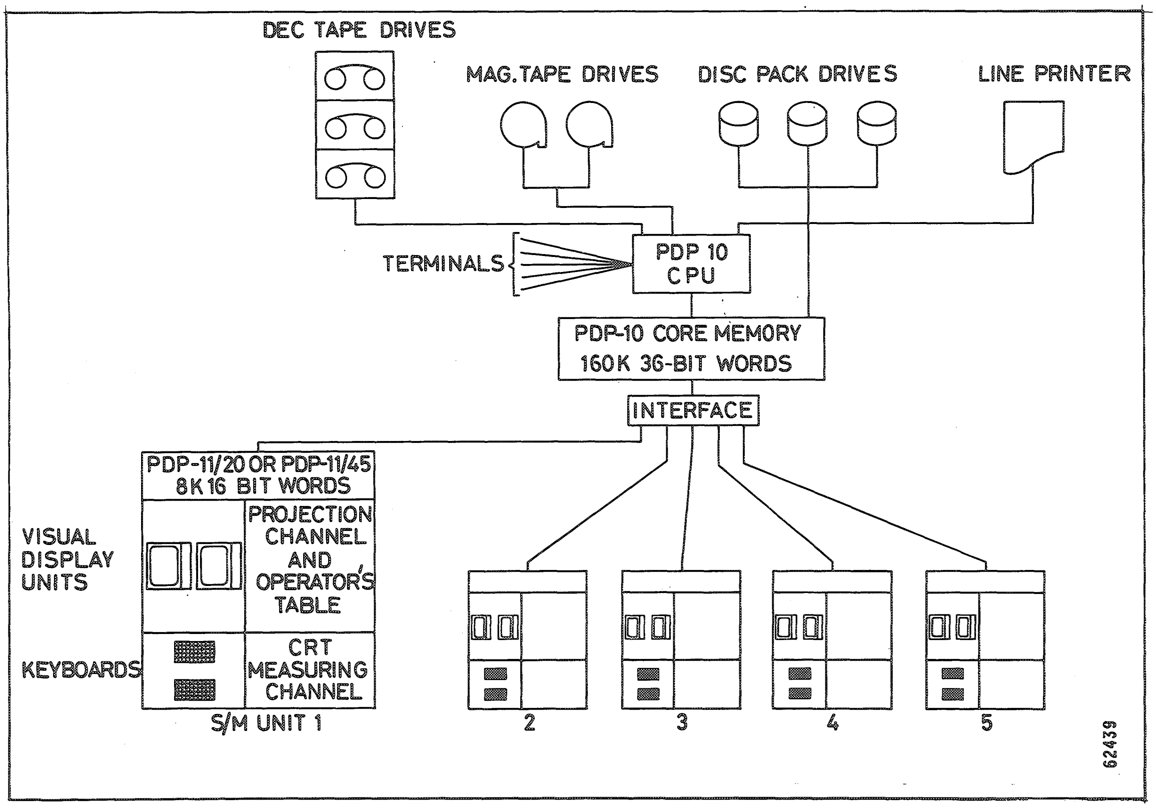

Fig. 6. General lay-out of the ERASME-system.

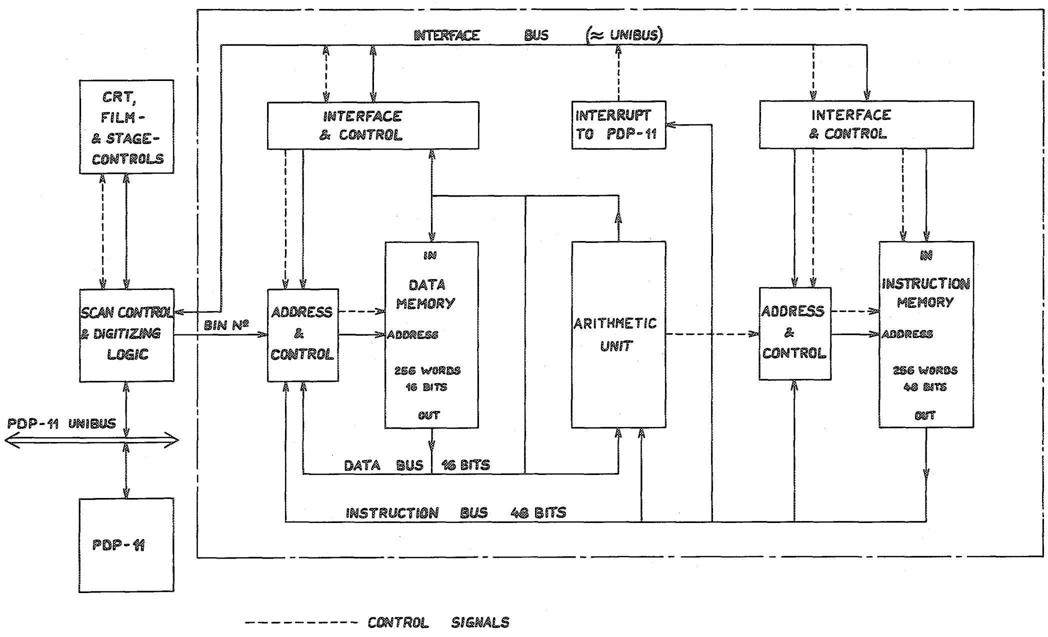

Fig. 7. ESOP (ERASME Special On-line Processor).

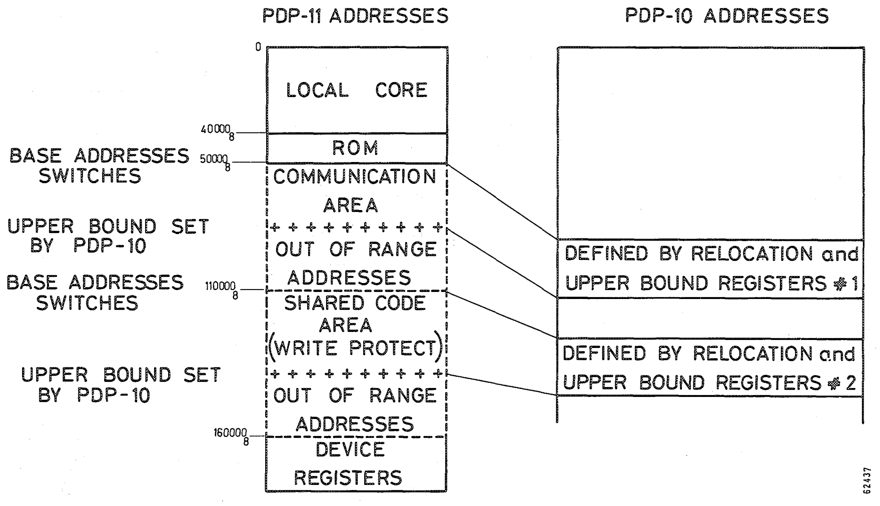

Fig. 8. DEC System 10/PDP-11 interface address mapping.

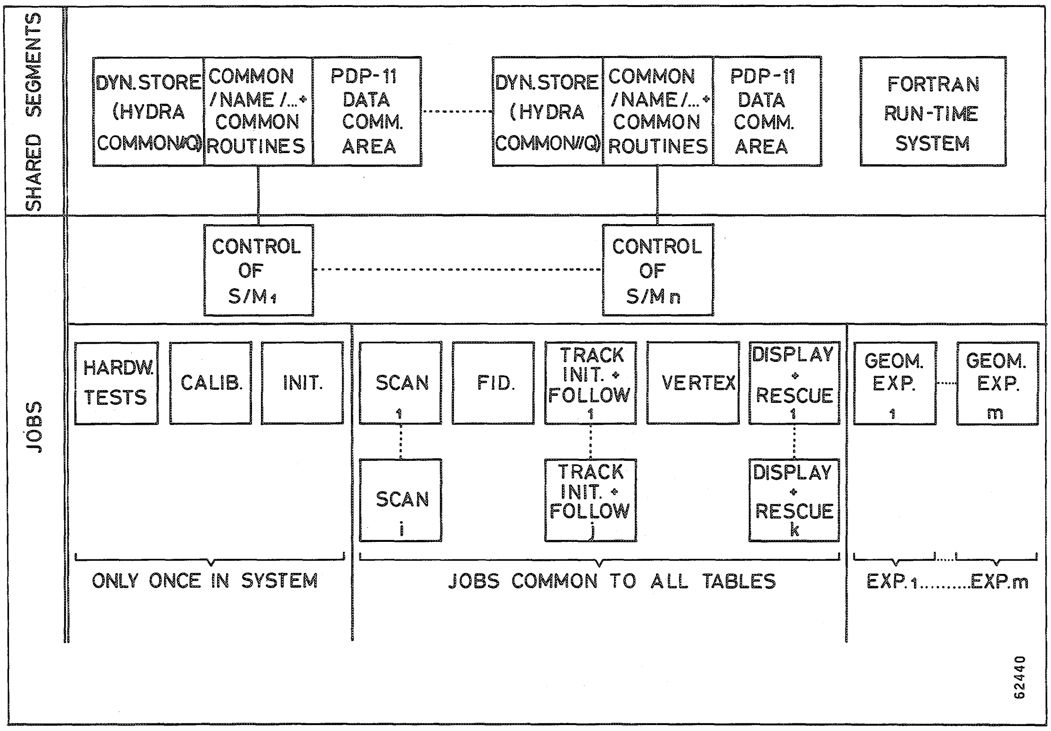

Fig. 9. Software structure.

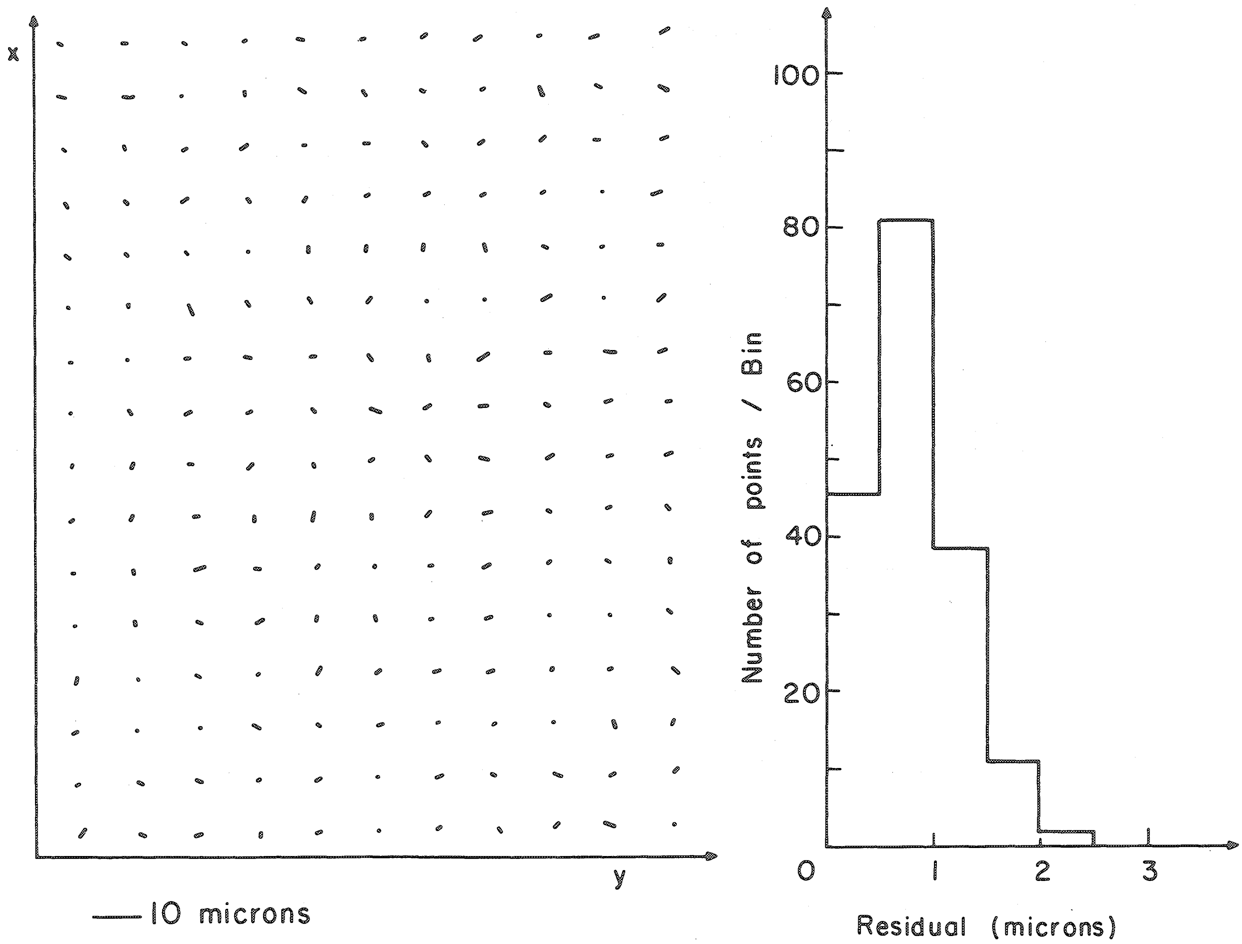

Fig. 10. Calibration: a) Residuals, b) Distribution of residuals.

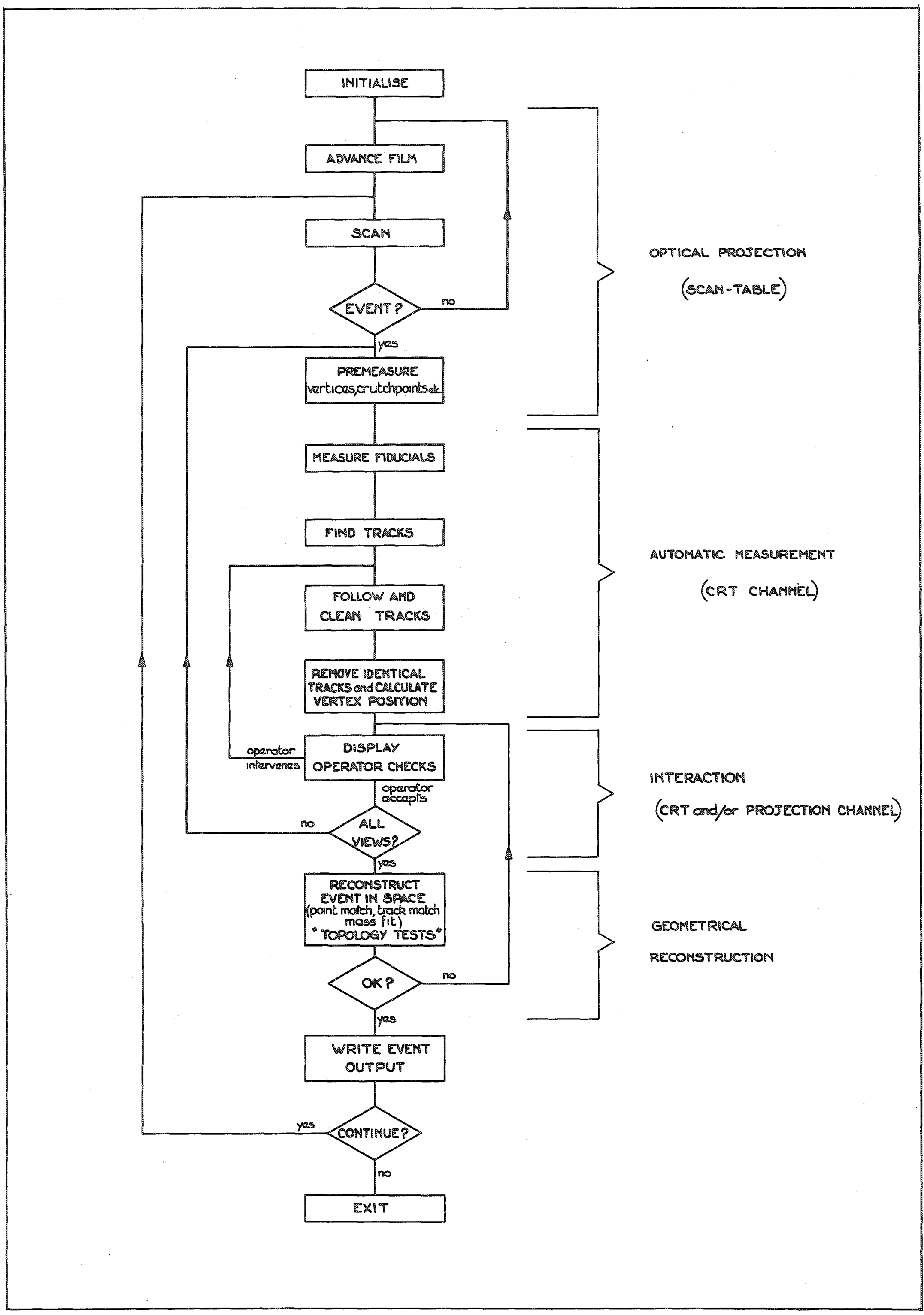

Fig. 11. Sequence for event processing on ERASME.

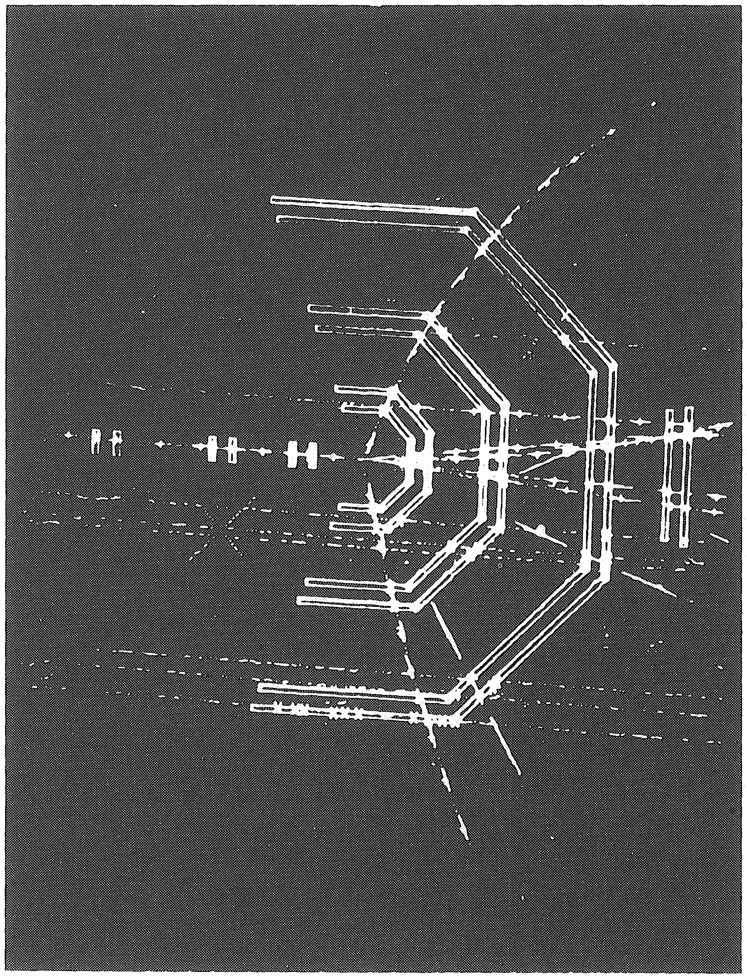

Fig. 12. Track initialisation with partial octagons.

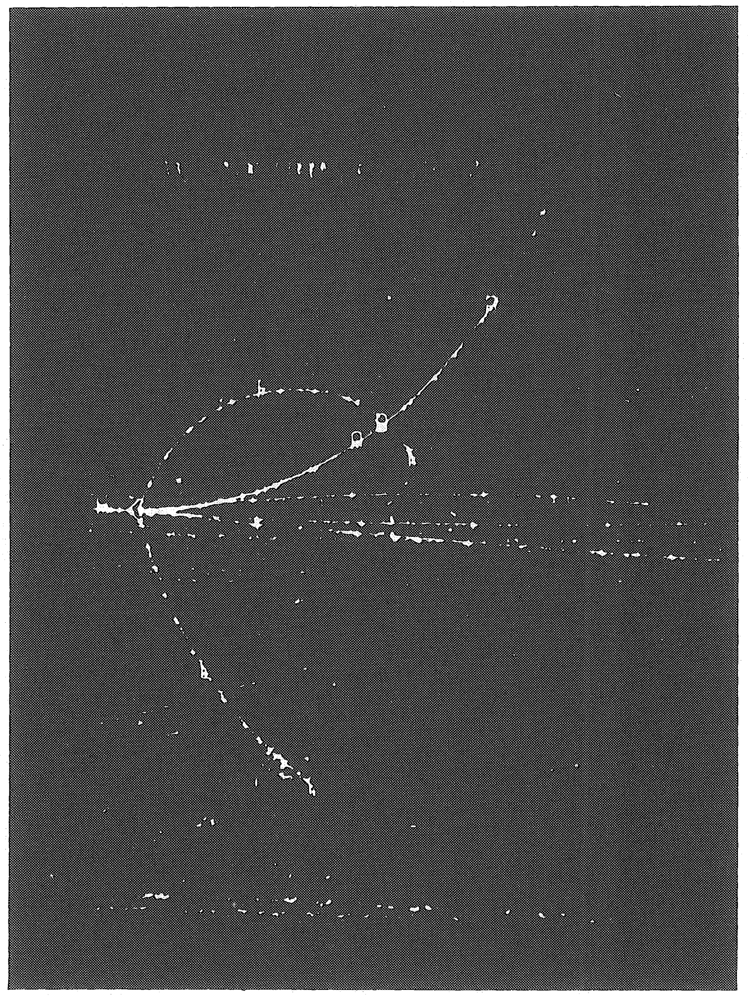

Fig. 13. Digitisings and measured tracks plotted on display unit.