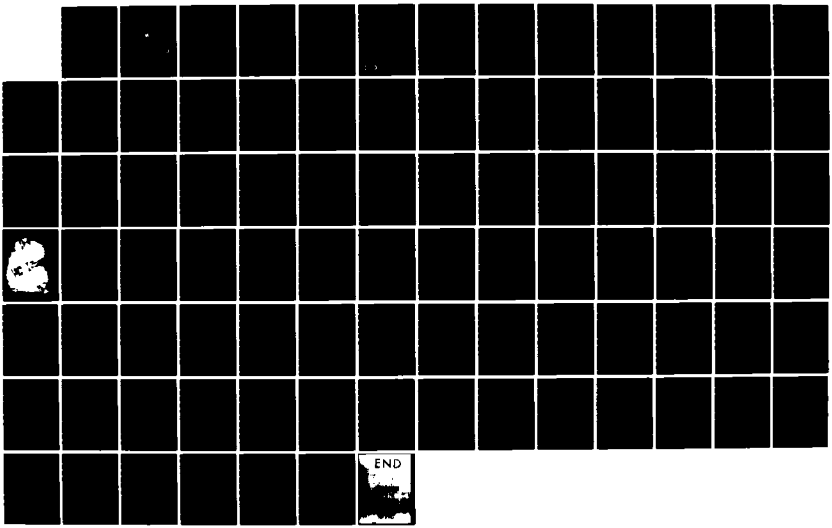
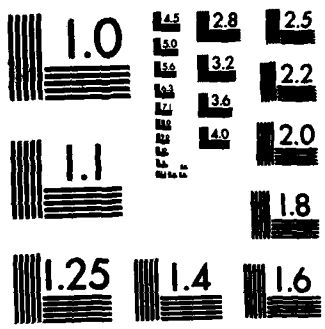
AD-A144 537 A METHOD OF PHOTO DATA REDUCTION WITH DESIGN CONSIDERATIONS FOR THE NOVA..(U) NAVAL BIODYNAMICS LAB NEW ORLEANS LA J J LAMBERT JAN 84 NBDL-84R001 1/1

UNCLASSIFIED F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

NBDL - 84R001

(12)

A METHOD OF PHOTO DATA REDUCTION, WITH DESIGN CONSIDERATIONS FOR THE NOVA 800® AND UNIVAC 1100/83® COMPUTERS

James J. Lambert

January 1984

NAVAL BIODYNAMICS LABORATORY
New Orleans, Louisiana

Approved for public release. Distribution unlimited.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NBDL-84R001	2. GOVT ACCESSION NO. AD-A144537	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Method of Photo Data Reduction, with Design Considerations for the NOVA 800® and UNIVAC 1100/83® Computers		5. TYPE OF REPORT & PERIOD COVERED Research Report 6. PERFORMING ORG. REPORT NUMBER NBDL-84R001
7. AUTHOR(s) James J. Lambert		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Biodynamics Laboratory P. O. Box 29407 New Orleans, LA 70189		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS M0097PN001-5001 63706N
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Medical Research & Development Command Bethesda, MD 20814		12. REPORT DATE January 1984 13. NUMBER OF PAGES 86
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

data analysis, computer program, photo data reduction, photo digitizing system (PDS), high speed camera, biodynamic experiments, inertial environments, impact injury models

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Digitized photo data acquired during impact experiments must be scaled and converted to a format suitable for analysis. Once converted, this data requires comprehensive graphical presentation for efficient interpretation. This report presents a detailed description of the software developed to accomplish such tasks in both a production and an interactive environment. Procedures utilizing the design presented have been instituted at NBDL and found to be effective.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

NBDL - 84R001

A METHOD OF PHOTO DATA REDUCTION, WITH DESIGN CONSIDERATIONS FOR THE NOVA 800® AND UNIVAC 1100/83® COMPUTERS

James J. Lambert

January 1984

Naval Medical Research and Development Command
Research Work Unit No. M0097PN001-5001

Approved by

J. C. Guignard
Chairman, Editorial Review Board

Released by

Captain L. E. WILLIAMS MC USN
Commanding Officer

Naval Biodynamics Laboratory
P. O. Box 29407
New Orleans, LA 70189

SUMMARY PAGE

THE PROBLEM

Digitized photo data acquired during impact experiments must be scaled and converted to a format suitable for analysis. Once converted, these data require comprehensive graphical presentation for efficient interpretation. This report presents a detailed description of the software developed to accomplish such tasks in both a production and an interactive environment.

FINDINGS

Procedures utilizing the design presented have been instituted at NBDL and found to be effective.

RECOMMENDATIONS

In an environment requiring data reduction processing of photo data, the design presented herein should be considered.

ACKNOWLEDGEMENT

This research was sponsored by the Naval Medical Research and Development Command and was performed under Navy Work Unit No. M0097PN001-5001. The author acknowledges Mr. William R. Anderson for valuable technical collaboration proving useful in the presentation of findings of this research. Miss Judy B. Johnson expended much time and patience in creating and maintaining the report text using NBDL word processing equipment. Additionally, Mr. Arthur M. Prell and Miss Margaret M. Harbeson were responsible for the creation of photographic reproductions (of illustrations) and the manuscript editing functions, respectively.

Trade names of materials or products of commercial or nongovernment organizations are cited only where essential for precision in describing research procedures or evaluation of results. Their use does not constitute official endorsement or approval of the use of such commercial hardware or software.

TABLE OF CONTENTS

A METHOD OF PHOTO DATA REDUCTION, WITH DESIGN CONSIDERATIONS FOR THE NOVA 800® AND UNIVAC 1100/83® COMPUTERS

INTRODUCTION

The Naval Biodynamics Laboratory (NBDL), located in New Orleans, LA, is an internationally recognized laboratory which performs experimental research to determine the effects of aircraft crashes, ship motion, vibration, aircraft ejection and parachute opening forces on the health and performance of Navy personnel. On-going research programs use high speed instrumentation cameras to record the motion of test subjects during biodynamic experiments. The films are digitized and the 3-dimensional motion is reconstructed and analyzed.

The procedures and programs used to convert the digitized photo data into a format compatible with a large-scale general purpose computer (UNIVAC 1100®), to plot the data for identification of data errors, and to prepare the data for subsequent processing have recently undergone major revision. The objectives of the revision were to:

1. Modify existing software to be compatible with announced changes in the UNIVAC operating system.

2. Use new graphics support capabilities to improve error detection procedures, reduce turn-around time of analysis, and produce output compatible with all available graphics devices.

3. Improve efficiency of operations personnel.

4. Standardize data formats and I/O access methods to be compatible with existing archival data.

5. Develop a modular organization of software which provides greater operational flexibility.

This report documents the new procedures and programs.

DESCRIPTION OF SOFTWARE

Preparation of digitized photo data for subsequent reduction and analysis requires a process consisting of several tasks performed under computer control. Software supporting each task conforms to the principles of modular design so that task sequences may be selected to meet specific needs of the data being processed.

In order to process digitized photo data through a data reduction flow, major tasks are selected and merged in a single runstream. The three available programs are:

1. Convert photo data from PDS (Photo Digitizing System) NOVA 800® format to UNIVAC 1100/83® format for subsequent analysis.

2. Produce film frame plots and x-y trajectory plots of target data in order to provide graphical presentation for error analysis.

3. Convert data digitized manually into the format of data tracked under computer control.

Each of these components is documented in detail in the body of this presentation. It is assumed that the reader is familiar with the operating system of the UNIVAC 1100/83® computer (see reference 3) and with the UNIVAC ASCII FORTRAN programming language (see reference 4). Although the software presented herein is intended for specific computers, plotting devices and software support packages, it has been designed to minimize the conversion effort for other systems.

1. Photo Data Conversion Program

A. Introduction. This section describes a UNIVAC 1100/83® program which converts photo data from PDS (Photo Digitizing System) data tapes to UNIVAC 1100/83® format. In converting data to a format compatible with a large-scale general purpose computer (UNIVAC 1100/83®), the photo data conversion program provides a suitable environment for efficient data reduction and graphical presentation.

B. Discussion. Input to the conversion program is a PDS data tape. It contains digitized photographic data from at least one experiment, or run. For a particular run, as many as three cameras are used to photograph targets on anatomical mounts strapped to the experimental subject. Each reel of film generated from these cameras during the run is digitized (on the NOVA 800® system), with the output being written to a PDS data tape. Data for several runs may be written to a single PDS tape. This tape contains four record types. The first is a header record, which contains identification information for film from a particular camera used during a run. This record includes such items as run number, PDS record type, camera site number (1, 2 or 3), julian date the film was digitized and time-of-day for the first film frame (all character data on the PDS tape is in ASCII format and numeric data is in twos complement integer format). The second record type is the time record, which identifies time-of-day for each film frame, or data point. The time data values originate from LED displays on the film frames themselves. There are 250 data points on a time record (the first point being the time at first motion of the subject). Identification information for the reel of film the time record describes is also included in the record, i.e., run number, PDS record type and camera site number. The final two record types are the X and Y data records, respectively. Each contains 250 data points measuring film plane displacement (inches) from a stationary origin on the film. An X and Y data record exists for every photo target tracked during the run. Each data record contains (in addition to displacement values) run number, PDS record type, camera site number and a target number. The target number contains information describing the anatomical mount type on which it is located. For instance, there are head mounts, neck mounts, mouth mounts and pelvic mounts - all designed to track motion of a particular segment of the body.

Data on a PDS output tape is organized by reels of film. That is, all data for a particular reel of film (with its unique header, time and data records) are grouped together into a file. The order in which the records appear is always the same: header record, time record, target 1 (x values), target 1 (y values) ··· target N (x values), target N (y values), where N = number targets digitized on film. In some experiments more than the standard 250 frames are digitized. In this instance the data records for the extra frames (always a multiple of 250) are written to the PDS tape immediately after the final data record containing data for the previous 250 frames. An end-of-file separates each group of data records for a particular set of 250 frames. A double end-of-file signals the end of data on the tape.

The conversion program reads the NOVA 800® input tape and stores it in an input buffer. The PDS characters are in 8-bit ASCII format and the numeric data is either in 16-bit or 32-bit integer format. The conversion program determines the format of the information to be converted from the format specifications of the PDS input tape (see Figures 1b-1d). All character data is converted from 8-bit NOVA ASCII to 9-bit UNIVAC ASCII format. Numeric data is either converted to 36-bit UNIVAC floating point format or 36-bit UNIVAC integer format, depending on the format type of the information being converted. For instance, PDS record type is converted from NOVA 16-bit integer to UNIVAC 36-bit integer, while displacement (data) values are converted from NOVA 32-bit integer to UNIVAC 36-bit floating point format. All converted displacement values are then scaled from inches to meters. This is done to provide data compatibility with a group of data analysis programs collectively called EASYFLOW*.

The converted and scaled photo data must be output in a standard file format suitable to subsequent processing; in addition this data must be output in FIELDATA format, which is the NBDL archival standard. This second requirement presents difficulties in an ASCII FORTRAN programming environment. In order to provide for both needs an I/O access package (see Reference 1) has been developed to (1) provide standard I/O access methods for an established photo data file format and (2) maintain data format compatibility with the NBDL archives. The I/O package provides directory maintenance for the user. The conversion program utilizes this capability to output the standardized direct access record. This directory contains unique run identification numbers (for all runs on the output tape) and accompanying sector locations for both header and time records for the run. The directory provides means for rapid direct access of photo data for a given run number.

* EASYFLOW is an NBDL term representing a data reduction and analysis system requiring input data of a specific format. EASYFLOW uses converted photo data to derive position coordinates in 3-dimensional space of targets affixed to anatomical mounts (which are strapped on to the experimental subject). These data are subsequently used to compute and plot displacement, linear (and angular) velocity and linear (and angular) acceleration of the subjects.

There are four record types, in addition to the directory, which the conversion program outputs to a UNIVAC mass storage file. The first of these is the header record. There exists a unique header record for each camera site used in a run. There are no more than three and no less than two cameras used in a run, each assigned a site identification number of 1, 2 or 3. This site ID and the run number in concert make the header record unique. A keyword is stored in the header record along with identification information such as the run number and site ID. The keyword identifies the record as a header. The second type of output record is the time record. It contains the time-of-day information for each of 250 film frames. In addition the time record includes the run number, the site ID and a keyword for identification. The third record type is a photo data record. It contains displacement values (from a fixed point on the film plane) in meters of a particular photo target on the reel of film. Also included are the run number and site ID of the film, along with a keyword identifying the record type. This keyword also contains information identifying the anatomical mount (on which the photo target resides) and the axis type of the data (X or Y). The combination of header record, time record and data records (for all targets on the reel of film) is written to disk, in the order listed, by the conversion program. Data for each reel of film with a camera site ID of 1 are grouped together on the output file and separated by double ends-of-file*. An additional end-of-file is written after the final data record of the final reel of film for site ID 1. This signifies end-of-data. However, the direct access table follows this final end-of-file. It is located at the end of the file so as not to limit its size. In order to provide access to the directory a "locator table" is written as the initial output record. It contains the sector address of the directory.

The above output file structure is duplicated for camera site 2 (and 3 if necessary). A separate UNIVAC disk file is thus created for each site ID. Each of these files is in a standardized format, and serves as input to a UNIVAC runstream designed to write them to a UNIVAC 1100/83® data tape.

The conversion program is designed to correct three common situations in which PDS operator error has occurred. These three special cases are:

(1) The run number of a header record for a reel of film is entered incorrectly by the PDS operator and written to the PDS data tape.

(2) A reel of film is re-digitized by the PDS operator and output to the end of the same PDS data tape as data from the original reel of film.

(3) The camera site ID of a header record for a reel of film is entered incorrectly by the PDS operator and written to the PDS data tape.

* If more than 250 frames of data were digitized on a reel of film, each group of data records for the extra frames (always a multiple of 250) is written in succession and separated by single end-of-file.

In cases (1) and (3) the conversion program uses correction fields in card image input in order to scan the header record for the incorrect entry and then correct it. This is done prior to writing the header to disk. In case (2) the conversion program scans for the first occurrence of the data file for a reel of film. Once found, this data file (header, time and data records) is bypassed and thus not converted and output. However, the second occurrence of a data file for the reel of film is converted and output in the usual manner. The photo conversion program accomplishes the following objectives:

(1) Converts photo data to a format compatible with a large-scale general purpose computer (UNIVAC 1100/83®).

(2) Scales displacement data values from inches to meters in order to provide data compatibility for subsequent processing.

(3) Provides for correction of common PDS operator errors.

(4) Accomplishes output of data with standardized I/O access methods.

(5) Maintains data format compatibility with the NBDL archives.

C. Main Program and Subroutine Descriptions:

(1) MAIN (see Appendix 1b for listing) - This is the driver program which controls all program operations. All correction cards are read by this program and used to rectify the three special cases earlier noted. MAIN then calls subroutine CONVT in order to read and convert the first PDS input record (see Figures 1e-1g for input record formats). The rest of the input records (non-header) for the run are read and converted by virtue of calls to subroutine CNVT1. Here the converted records are written to mass storage. Once a run has finished processing, control returns to MAIN and the process repeats itself until all runs have been processed. At this point a triple end-of-file, denoting end-of-data, is written to the output file.

(2) CNVT1 (see Appendix 1c for listing) - This subroutine controls the conversion and output of all data for each camera. The main program reads the header records and transfers control to this subroutine for reading and conversion of the position and time records. The position data is converted from inches to meters*. Each header, time and position record is written to mass storage by a call to the photo IO subroutine PUTRCD (see Appendix 4c). Data for each site ID is written to a separate mass storage unit (see Figure 2a). As mentioned earlier, the necessary data directories are maintained and written by the photo IO subroutines. After processing an entire run, CNVT1 writes a double end-of-file for each output unit and returns control to the calling program.**

* During the scaling process all data values of 0.0 (points not digitized) are set to 999.0

** If more than 250 frames were digitized for each target in a run, only a single EOF (End-of-file) is written after the final data record for each set of 250 frames.

(3) CONVT (see Appendix 1d for listing) - This subroutine is called by CNVT1 to perform the actual conversion of data from PDS format to UNIVAC 1100/83® format. It is called after each read of a PDS input record. All character data is converted from 8-bit ASCII to 9-bit ASCII format. Numeric data is converted from 32-bit binary integer to UNIVAC 1100/83® 36-bit binary floating point format (see Figures 2b-2f for output record formats).

In this subroutine the sign bit is extended for negative numbers; these negative values are data entered manually by the PDS operator by using the X,Y crosshair input.

(4) SPHAVG (see Appendix 1e for listing) - This subroutine calculates the X and Y position of a stationary reference target relative to the film sprocket hole for each camera used in a run. Sprocket hole values for five consecutive frames are averaged, the result being subtracted from the averaged sled (chair) displacements for the same five frames in order to produce the final value. The X and Y sprocket hole averages are printed for each camera and in those cases where there is no sprocket hole information on the PDS tape, the values can be read from the film by the PDS operator and calculated by hand. Execution of EASYFLOW is expedited by providing sprocket hole averages at the time of conversion. This information is required for an internal consistency check on the data collection and system.

(5) ASC8T9 (see Appendix 1f for listing) - This subroutine converts a string of 8-bit ASCII characters to a UNIVAC 1100/83® compatible 9-bit ASCII character string. All character data read from the PDS input tape is converted by the subroutine. The high order bit of a 9-bit ASCII character is always off, this bit being the only difference between an 8-bit ASCII character and a UNIVAC 1100/82 9-bit character.

2. Plot Program for Photo Data

A. Introduction. This section describes an ASCII FORTRAN UNIVAC 1100/83® program which plots UNIVAC-compatible photo data as (a) X-Y film frame plots and (b) X-Y trajectory plots. These plots provide a method for error detection of faulty photo data, allowing for correction of the data before any further processing takes place.

B. Discussion. The plot program generates both x-y film frame plots and x-y trajectory plots. The film frame plots (see Figure 5a) plot the position of each photo target (on a reel of film) relative to the film frame origin. Each individual frame of film is plotted, and all target positions in the frame are connected by straight lines; this aids in visual recognition of relative changes in target positions from frame to frame. There is sometimes more than one anatomical mount (see Figure 6 for mount example) visible to a camera during a run. It is for the reason that a separate set of film frame plots are generated for each anatomical mount found on a reel of film. For instance, if two mounts are visible to a particular camera and only one mount is visible to a second camera, three sets of film frame plots are generated for the experiment. A total of 250 film frames are plotted for each set, which consists of ten pages of twenty-five plots each. On top of each page the plot scale (meters) is listed; this measures the greatest target displacement from the first to last frame on the page. In addition all target positions which were manually digitized (not automatically tracked) by the PDS operator are circled for identification. The film frame plots are used to identify errors which may exist in the data. For instance, a target may have been identified incorrectly by the PDS operator; this problem is visually exposed in the film frame plots. Subsequently the data file is corrected.

The x-y trajectory plots (see Figure 5b) identify the path each target (on a reel of film) travels in the film plane during a run. Each individual point on the plot measures target displacement from a fixed film frame origin. A total of 250 data points are plotted, with the first coinciding with time of first motion of the experimental subject. Since more than one anatomical mount may appear on a reel of film, a separate x-y trajectory plot is generated for each mount on the film. The path of every photo target on a mount (visible to the camera) is plotted, each identified by its own target number. The x-y trajectory plot provides at a glance a visual presentation of target movement (for a given mount) for the entire run duration; this allows for quick detection of data discontinuities. For instance, if the pattern recognition algorithms (used in the photo digitizing process) found the wrong target, a detectable discontinuity results in the x-y trajectory plot.

The plot program for photo data possesses versatile plot output capabilities, made available through the use of DISSPLA® plot software (a product of ISSCO*). All plot commands are output to a catalogued UNIVAC disk file (via DISSPLA® software). This "compressed" plot file can then be used as input to a variety of DISSPLA® post-processor programs, which "decompress" the plot commands and direct them to a plotting device. This disk file may be utilized several times on a variety of plotting devices without once repeating execution of the actual plot program. DISSPLA® post-processor programs exist for such plotting devices as the TEKTRONIX 4014®, FR80® and HP 7220® (all available to NBDL).

There are advantages in using any of these post-processors. For instance, the FR80® post-processor is used in a production environment to generate microfiche and high quality hardcopy. The TEKTRONIX 4014® post-processor can be used to view selected plots on the TEKTRONIX 4014® scope immediately after execution of the photo data plot program. This quick turnaround is of benefit in expedient data analysis. The FR80® is not located on-site, so in order to combine the benefits of quick turnaround and high quality hardcopy an HP 7220® plotter is to be installed on-site. This plotter in addition affords a multicolor plotting capability.

* Integrated Software Systems Corporation

The input to the plot program is UNIVAC-compatible photo data. Character data is in 9-bit UNIVAC ASCII format and numeric data is either in 36-bit UNIVAC integer format or 36-bit UNIVAC floating point format (depending on format specifications). As noted in section 1.B data is organized on the input files by film reel, with a separate input file for each camera site. A directory exists in each input file, containing the run numbers (each run number identifies a unique film reel) and accompanying sector locations of the header and time records. The data records for a run (or film reel) immediately follow the time record, and a "locator table" for the directory precedes all other records on the file. Input is accomplished with a general purpose I/O package (see Reference 1) which reads FIELDATA (NBDL archival standard) in an ASCII FORTRAN programming environment. Plots are output for each input file (or camera site). Within each file, plots are output by run number (or film reel) and within each run plots are output by anatomical mount.

The PDS operator is responsible for entering time-of-day values (LED displays) for each frame of digitized film. To insure their legitimacy the plot program performs tolerance checks for the time values. The results (see Figure 5c), called a time record analysis, are output to a line printer*.

The plot program for photo data accomplishes the following:

(1) Provides error detection methods for photo data by generating:

(a) x-y film frame plots.

(b) x-y trajectory plots.

(2) Performs confidence checks for time-of-day values for film frames and prints the results.

(3) Provides plot flexibility features through the use of post-processor programs.

(a) Plots can be generated repetitively without repeat execution of the plot program.

(b) Plots can be viewed immediately upon execution on a demand terminal scope.

(c) Quality hardcopy and microfiche of plots are generated.

(4) Accomplishes input of data with standardized I/O access methods.

* This analysis may be output to a plotting device, but it is slower and more expensive to do so.

C. Main Program and Subroutine Descriptions:

(1) PLOT (see Appendix 2b for listing) - This is the driver program which controls all program operations. A single input card identifies (a) the number of input files, (b) the time record analysis option, (c) the X-Y contour plot option, (d) the frame plot option, (e) the frame plot format option, (f) the lowest run number to be processed and (g) the highest run number to be processed (see Appendix 2a for card format). Once the input card is read, DISSPLA® is called to initialize the compressor. Plotting is accomplished by virtue of a loop which controls the input and subsequent plotting of photo data for each combination of run/camera site/anatomical mount (see Figures 2b-2f for input record formats).

(2) TIMER (see Appendix 2c for listing) - This subroutine generates a time record analysis for a given run number and site ID. The time record is input via the photo data I/O program (see reference 1). A time interval between frames is approximated by averaging total elapsed time over the 250 frames of data. This increment is used to calculate an "expected time" for each of 250 frames. These times are compared to the actual time record data points in order to test for excessive error (a difference between actual and "expected" time greater than .001 second). Any such discrepancies are annotated with an asterisk (*) on the time record analysis (see Figure 5c).

(3) LDATA (see Appendix 2d for listing) - This subroutine loads photo displacement data into common for access by plot subroutines. This routine is called once for each combination of run/camera/mount. The photo data I/O package is used to accomplish input (in order to read FIELDATA within an ASCII FORTRAN program).

(4) MIXY (see Appendix 2e for listing) - This subroutine determines minimum and maximum X and Y values for a page (twenty-five frames) of frame plots. These values are placed in common for scaling use by the plot subroutines. The minimum and maximum values are selected so that the abscissa and ordinate axes are equal lengths in data units (meters). This allows the viewer of the plots to determine at a glance in which direction the targets were moving fastest. For example, if the targets moved from the far left of the initial frame on the page to the far right of the final (25th) frame, the X filmplane displacement was changing more rapidly than the Y displacement.

(5) PLOTD (see Appendix 2f for listing) - This subroutine consists of two entry points. All plotting in both entry points is done with DISSPLA® software.

A. PLOT1 - This entry point plots the title information for each page of frame plots. Included in the title information are the (i) run number, (ii) camera number, (iii) plot scale (meters), (iv) anatomical mount type and (v) page number. All plotting is done with DISSPLA® software.

B. PLOT2 - This entry point plots all frame plots. It is called once for every frame of photo data to be plotted. Two-hundred fifty frames are plotted for each mount, twenty-five frames/page, ten pages/mount. The target number for all targets tracked (by the PDS operator) in a particular frame are printed directly at their digitized position. Those targets digitized manually are circled for identification. All targets in a particular frame are connected by vectors in order to present the image of motion as the targets begin to change position from frame to frame. Targets may have been deleted or added as the frames advance. Deleted target numbers are printed in the bottom right corner of the frame in which the target was dropped. Added targets are printed in the bottom left corner of the frame in which the target was added. If chair data are available for the current run/camera, a 'C' is printed in the top left corner of each frame. The frame number is printed in the top right corner of the frame.

(6) XYPLOT (see Appendix 2g for listing) - This subroutine plots X vs. Y filmplane displacement values for a given run/camera/anatomical mount. All targets digitized for the mount are plotted on the same graph. A maximum of 250 data points are plotted (and connected) for each target, with the target number appearing coincident with the first point for a curve. The maximum and minimum values of the abscissa and ordinate are printed on the plot. In addition, titling information is listed which includes (i) run number, (ii) camera number, (iii) anatomical mount type and (iv) current date. The X-Y contour plot affords the viewer a quick method of isolating data incongruities.

3. Reformatting (sign change) Program for Photo Data

A. Introduction. This section describes an ASCII FORTRAN UNIVAC 1100/83® program which converts UNIVAC-compatible photo data, digitized manually, into the format of data tracked under computer control. This prepares the data for subsequent computation of 3-dimensional trajectories of test subjects.

B. Discussion. This program is designed to read all target position data for a given range of run numbers and scan for negative data values. Each negative value represents an entry made manually by the PDS (Photo Digitizing System) operator using the x-y crosshairs. All such values are multiplied by -1 and rewritten to the data file. An archival data tape is later generated (with the reformatted data records) by an independent runstream. This tape serves as input to the EASYFLOW runstream.

Input to the reformatting program is UNIVAC-compatible photo data. Character data are in 9-bit UNIVAC ASCII format and numeric data are either in 36-bit UNIVAC integer format or 36-bit UNIVAC floating point format (depending on format specifications). Data are organized on the input files by film reel (or run number), with a separate input file for each camera site (see section 1.B for a more detailed description). Direct access tables accompany each input file. Input/output is accomplished with a general purpose I/O package (reference 1) which reads/writes FIELDATA (NBDL archival standard) within the context of an ASCII FORTRAN programming environment. Output file formats for the reformatting program are identical to the input file formats, the only difference being that all values in the output data records are positive.

The reformatting program accomplishes the following:

(1) Prepares UNIVAC-compatible photo data for subsequent computation of three-dimensional trajectories (of test subjects) by converting manually digitized photo data into the format of computer-tracked photo data.

(2) Accomplishes input/output of data with standardized I/O access methods.

C. Main Program Description:

SGFLIP (see Appendix 3b for listing) - This is the driver program which controls all program operations. The only subroutines utilized are those belonging to the photo data ASCII FORTRAN I/O program. A single input card identifies (a) the number of input files, (b) the lowest run number to be processed and (c) the highest run number to be processed (see Appendix 3a for card format). The number of input files and the run number range (from the input card) are used to index a loop which systematically (a) reads data records, (b) scans them for negative data points and (c) changes negative values to positive values. Once a data record is completely processed it is rewritten to the input file unless no negative values were found (see Figures 2b-2f for input/output record formats).

4. Operation of Photo Data Reduction System

A. Introduction. This section describes the actual UNIVAC 1100/83® production runstream which controls data reduction processing of NBDL photo data prior to its input to the EASYFLOW runstream.

B. Discussion. As previously noted, there are three main processing steps in the photo data reduction process: conversion, plotting and reformatting. Each step performs an integral function independent of the other steps. This modular organization of steps provides several benefits:

(1) Any step can be performed independently from the others; i.e., each step has stand-alone capability.

(2) Modifications to steps can be effected with:

(a) Complete non-interference with other steps.

(b) Expediency in that all necessary changes take place in a single module.

(3) Input/output may be examined at any point between steps.

(4) Additional steps may be added to the overall process without altering any existing steps, thus providing a desirable environment for enhancements to the system.

Each step, or module, shall heretofore be referred to as an element. Element is a UNIVAC term describing a collection of computer commands. When it is desired to execute an element, or step, the computer is instructed to carry out all commands in that element. The production runstream described in this section executes a collection of three such elements:

(1) XQTCVT is the conversion element.

(2) XQTPLT is the plot element.

(3) XQTSGN is the reformatting (sign change) element.

The first element to be executed is the conversion element XQTCVT (see Appendix 1a). The main function of this element is to execute the photo conversion program. Additionally, the output of the conversion program is written to a UNIVAC data tape. In sequence, the conversion element (1) copies a PDS data tape onto disk, (2) executes the photo conversion program and (3) writes the output (from the conversion program) to a UNIVAC data tape. The UNIVAC tape is saved for 10 days as a precautionary measure. The key output from the conversion element is the converted photo data written to disk by the conversion program.

If it is necessary to correct for PDS operator error (see section 1.B.) the conversion element is altered to include correction options. These options are input to the element in card image via the UNIVAC EDIT processor (see reference 5). The EDIT processor commands are part of the production runstream. Thus the runstream actually edits the conversion element before executing it. The correction options allow for rectification (by the conversion program) of instances where the PDS operator:

(1) Enters incorrect run numbers on PDS header records.

(2) Outputs data to a PDS tape for two film reels with the same run number and camera site ID (on the header record).

(3) Enters incorrect camera site ID's on PDS header records.

The first card image of options (input with the EDIT processor) indicates the number of film reels to be corrected for each of the three categories of problem cases. A card image then follows for each occurrence of a problem case (1); it specifies the run number and camera site ID of (the header record for) the problem film reel as well as the occurrence (of this unique combination) to be corrected. It also specifies the correct run number to replace the incorrect field in the header record. After the last correction card image for case (1), card images follow for case (2). Each of these specifies the run number and camera site ID of the problem reel. The first occurrence of this combination of run number and site ID (in a PDS header record) is bypassed and only the second occurrence (of PDS records with this header identification) is converted to UNIVAC format. Lastly, the correction card images for case (3) follow. Each specifies the run number, camera site ID and occurrence index for the problem reel. Additionally, it contains the correct camera site ID. If no corrections are necessary on the PDS input tape, only one card image is input. It indicates quantity zero for each of the three problem cases.

The EDIT processor not only controls input of all correction options to the conversion element, but it also is used to enter the reel number of the PDS data tape to be used as input to the photo conversion program. As with the EDIT commands for correction options, these commands also constitute a portion of the overall production runstream.

The second element of the runstream to be executed is the plot element XQTPLT. Its primary function is to execute the plot program for photo data. Additionally, the disk file of DISSPLA® plot commands (output from the plot program) is used as input to the FR80® post-processor program to produce production quality microfiche and hardcopy of all plots. In sequence, the plot element (1) executes the plot program for photo data (the plot program uses photo data residing on disk as input), (2) executes the FR80® post-processor program and (3) writes the disk file of plot commands to a catalogued disk file for public access. As previously noted the plot program uses output from the conversion program as its input. This input consists of disk files of photo data, one for each camera site (1, 2 or 3) used in a series of runs. Output from the plot program, a disk file of DISSPLA® plot commands, serves as input to the FR80® post-processor. The post-processor interprets the plot commands in order to produce microfiche and hardcopy of all plots generated by the plot program (these include the x-y film frame plots and x-y trajectory plots). These plots are used in error detection analysis and are maintained on-site.

Certain plot options exist and are input to the plot element via the UNIVAC EDIT processor. The EDIT commands constitute part of the production runstream (see Figure 8); they are used to edit the plot element before executing it. A single card image contains all the plot options. These options include:

(1) Number of input files.

(2) Time record analysis option.

  (a) Output to line printer.

  (b) Output DISSPLA® plot commands to disk file.

  (c) Do not output to any device.

(3) X-Y trajectory plot option.

  (a) Output DISSPLA® plot commands to disk file.

  (b) Do not output to any device.

(4) X-Y film frame plot option.

(a) Output DISSPLA® commands to disk file.

(b) Do not output to any device.

(5) Plot direction option for x-y film frame plots*.

(a) Plot five columns to a page with frame number increasing in each column (plot direction always downward).

(b) Plot five columns to a page with frame number increasing in columns 1, 3 and 5 (plot direction downward) and decreasing in columns 2 and 4 (plot direction upward).

(6) Lowest run number to be plotted (7) Highest run number to be plotted	In an input file each run number identifies the header record for a single film reel; photo data is plotted for this range of run numbers (or film reels).

Once the plot program has been executed, the plot element executes the FR80® post-processor program to produce microfiche and hardcopy of all generated plots. The plot (disk) file is then copied to a catalogued disk file.

The third and final element of the production runstream is the reformatting element. The primary function of this element is execution of the reformatting (sign change) program. The element also generates a UNIVAC data tape containing output from the reformatting program. This data tape is saved for 180 days and is used as input to the EASYFLOW data reduction and analysis system. In sequence, the reformatting element (1) executes the reformatting program and (2) saves the output on a UNIVAC data tape. The reformatting program uses photo data residing on disk (output from the conversion program) as input. These same files, after being reformatted to reflect automatically digitized photo data, are output to the UNIVAC data tape.

The UNIVAC EDIT processor is utilized (in the production runstream) to input (1) reformatting program options and (2) the output tape file name. The input options include:

(1) Number of input files.

(2) Lowest run number to be processed (3) Highest run number to be processed	In an input file each run number identifies the header record for a single film reel; photo data is reformatted for this range of run numbers (or film reels).

* This option is used only in the event x-y film frame plots are generated.

C. Production Runstream. The photo data reduction runstream is a collection of computer commands designed to execute the three steps (elements) in the data reduction process. The runstream consists of (1) UNIVAC Operating System commands (see reference 3) and (2) UNIVAC EDIT processor commands (see reference 5). All three elements to be executed are stored in a single program file. The runstream (1) copies the program file onto a scratch disk file, (2) enters the input options for each element and (3) executes each element. The only input to the runstream is a single PDS data tape containing digitized photo data. Output from the runstream includes:

(1) A 10-day UNIVAC data tape of converted photo data.

(2) Catalogued disk file containing DISSPLA® plot commands for all generated plots.

(3) Microfiche and hardcopy of all generated plots.

(4) A 180-day UNIVAC data tape of converted and reformatted photo data.

(5) Line printer listing containing:

(a) Time record analyses.

(b) Operating system diagnostics.

(c) User-generated diagnostics.

(d) Operating system accounting information.

Several considerations entered into the design of the photo data production runstream. The more significant of these are noted as follows:

(1) In storing plot commands on a catalogued disk file, plots can be generated as many times as necessary (and on several plotting devices) without once repeating execution of the actual plot program.

(2) In some instances (a film reel of) photo data is redigitized in order to improve data quality. It is important (in computing 3-dimensional position of the test subject) that a minimum total of three targets (not all from the same camera site) are digitized successfully for a given test run; these targets may be tracked from any of the available film reels (camera sites) for the run. If the PDS computer cannot successfully locate enough targets (through computer-controlled tracking), the PDS operator generates another data tape in which (s)he manually digitizes (with the x-y crosshairs) the necessary targets.

The production runstream is designed to reprocess the new PDS data tape with a minimum of operator preparation. It is also equipped with the option to process redigitized data residing on the same PDS tape as the initial data (for the same film reel). The production runstream also creates UNIVAC data tapes directly after (a) conversion and (b) reformatting in order to make data available for further analysis, error correction, and reprocessing.

(3) The individual cameras (used in the experiments) are calibrated to provide mathematical camera constants vital in the computation of three-dimensional position (of the test subject). Examples of camera constants are the camera nodal point coordinates (x, y and z) and the lens distortion parameter. These values and other camera constants are generated by a camera-calibration program which uses as input converted data from a camera-calibration film reel. The photo data production runstream is designed to, with slight modification, execute the camera-calibration conversion program. This altered runstream is included as an element in the program file containing all photo data reduction programs.

(4) The modular organization of the steps (elements) in the photo data reduction process allows for efficient and expedient expansion of functions. New elements can be added without changing any existing elements. This feature affords the system a desirable flexibility. Standardized data formats and I/O access methods provide further flexibility. Possible additional functional elements are listed below:

(a) Error editing elements which correct improper data values in the data file.

(b) Packing elements which combine data files (and accompanying directories) from multiple tapes onto a single tape (with one all-inclusive directory).

(c) Plot elements which allow plot selection of individual photo variables residing on a converted photo data file.

(d) Special analysis elements which extract the data easily through the use of standardized data file formats and standardized I/O access methods.

Operational particulars of the runstream are as follows:

(1) <u>Program File</u>:

DATASYSTEMS*PHOTOSTREAM.

(2) <u>Programming Languages</u>:

(a) UNIVAC ASCII FORTRAN Level 10R1

(b) UNIVAC Executive System EXEC Level 38R5

(c) UNIVAC Text Editor Level 16R1

CONCLUSION

The photo data reduction and analysis system has been revised to successfully meet all proposed objectives:

1. All constituent programs are now compatible with announced changes in the UNIVAC operating system. These changes incorporate an ASCII FORTRAN programming environment.

2. A new graphics support capability is now utilized to:

   a. Improve error detection procedures by providing additional plot presentations.

   b. Produce output compatible with a variety of graphics devices (such as the Tektronix 4014® and the Hewlett Packard 7220®).

   c. Reduce turn-around time of analysis by providing for immediate review of plots.

3. A stand-alone runstream providing for greater operator efficiency is now in effect. In addition, all component programs exist in a single program file.

4. Standardized data formats and I/O access methods, which in unison provide compatibility with stored archival data, are now incorporated into photo data reduction processing.

5. A modular organization of software is now incorporated in order to provide for greater operational flexibility.

PDS OUTPUT TAPE FORMAT

RECORD SEQUENCE

BOT

Header
Time
Target 1, x values
" y values
Target 2, x values
" y values

First 250 frames

Target N, x values
" y values

Repeat above format if more than 250 frames are being digitized

Complete Date for One Film

End-of-File

Repeat for each film

End-of-File
End-of-File (double EOF:end of data on tape)

Figure 1a

PHOTO HEADER RECORD FORMAT

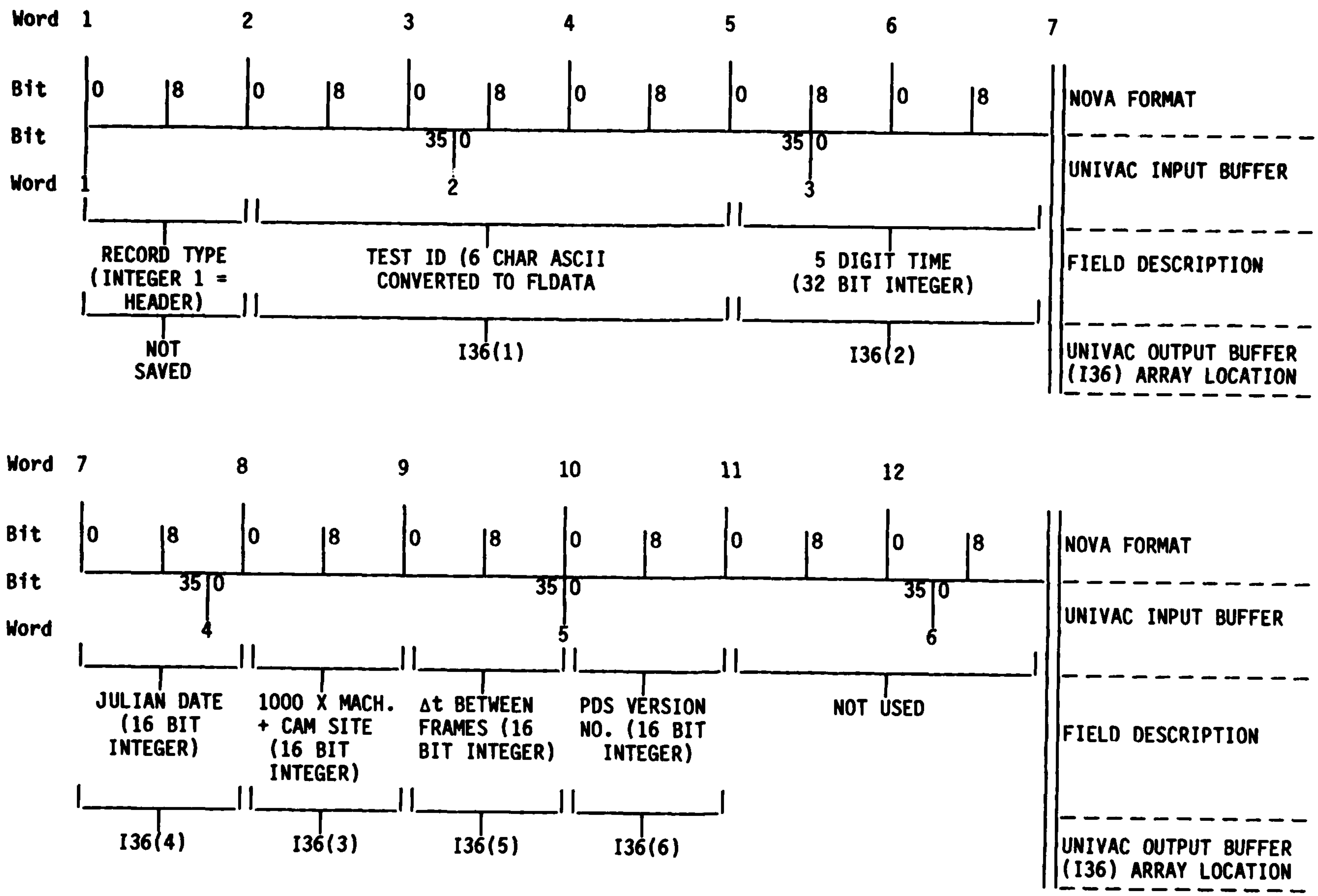


FIGURE 1b

28 Mar 79

PHOTO TIME DATA RECORD FORMAT

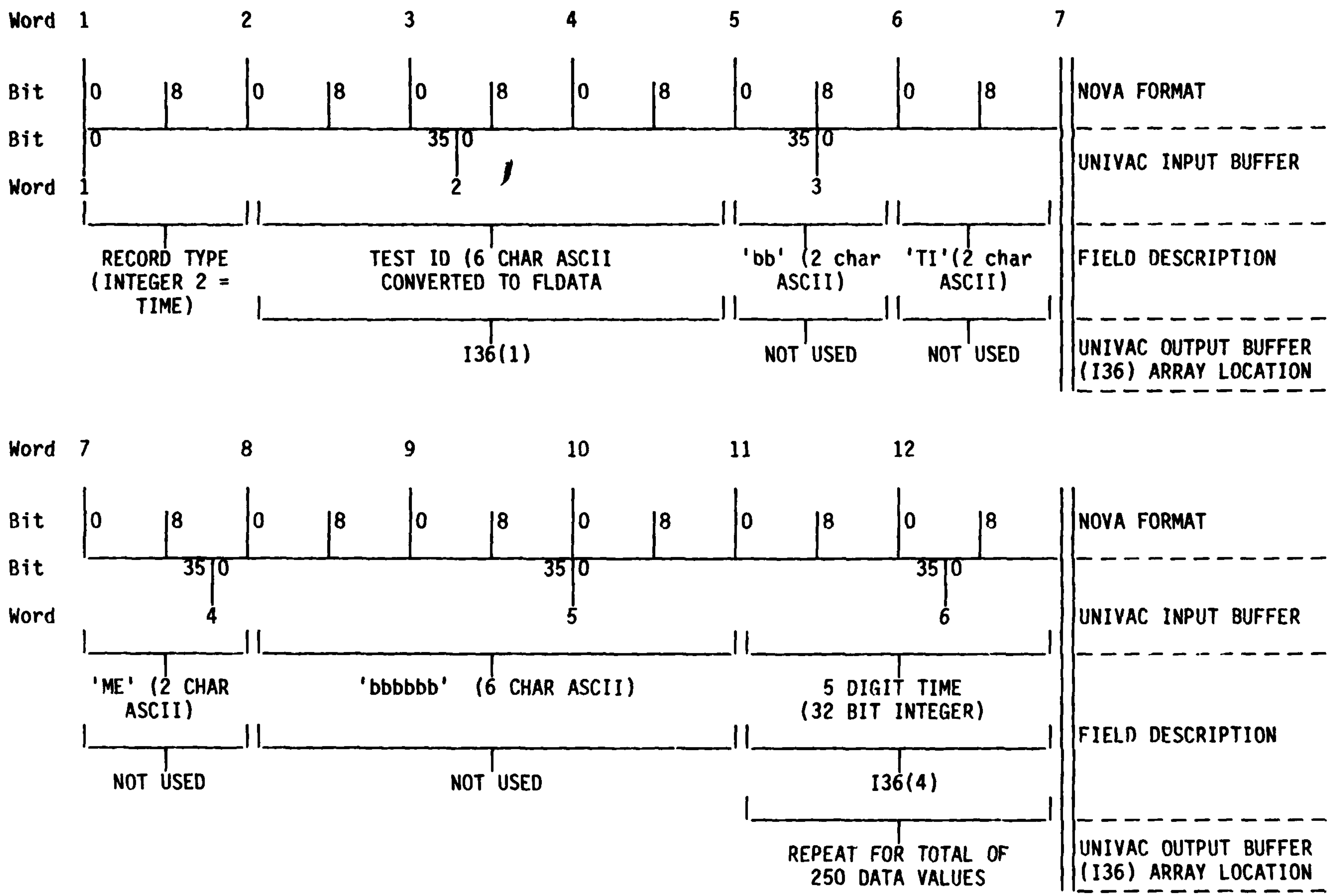


FIGURE 1c

28 Mar 79

PHOTO X (OR Y) DATA RECORD FORMAT

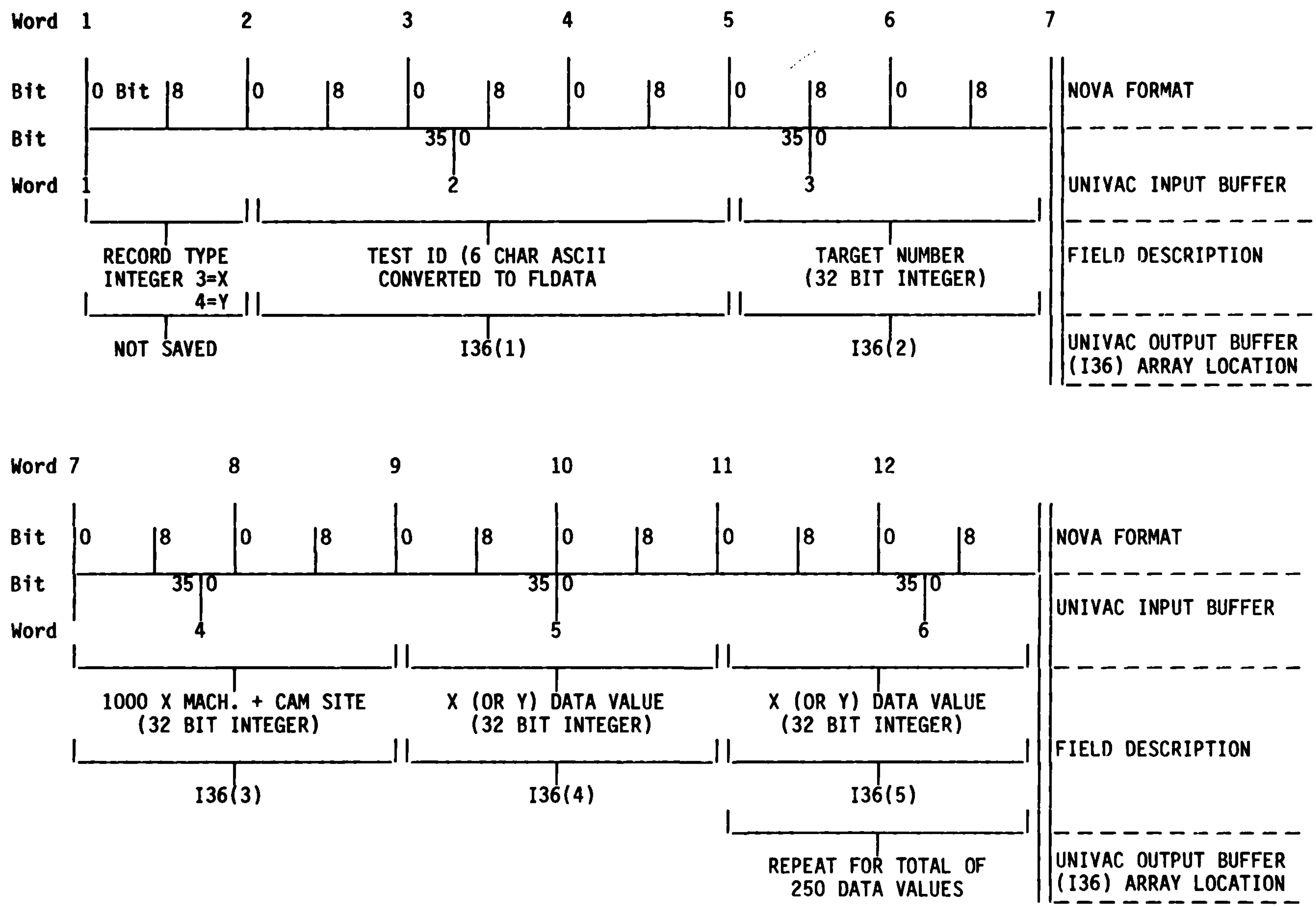


FIGURE 1d

28 Mar 79

HEADER FORMAT (PDS DATA)

16 Bit Word	Contents
1	Record type: 1 (16 bit integer)
2, 3, 4	Test ID (6 character ASCII)
5, 6	Time of first frame in units of the least significant LED digit (32 bit integer)
7	Julian date digitized: DDDY (16 bit integer)
8	1000 X machine # + camera site # (16 bit integer)
9	Δt: expected time between successive frames, in units of the least significant LED digit (16 bit integer)
10	Program version number (16 bit integer)
11	Not used
12	Not used

March 29, 1979 Figure 1e

TIME RECORD FORMAT (PDS DATA)

16 Bit Word	Contents
1	Record type: 2 (16 bit integer)
2 3 4	Test ID (6 character ASCII)
5 6 7 8 9 10	'b̸b̸ TIME b̸b̸b̸b̸b̸b̸' (12 character ASCII)
11 12	Time of frame 1 in units of the least significant LED digit (32 bit integer)
. . .	
509 510	Time of frame 250
512	Not used
513	Not used (fixed length record)

March 29, 1979 Figure 1f

X (OR Y) COORDINATE RECORD FORMAT (PDS DATA)

16 Bit Word	Contents
1	Record type: 3=x values (16 bit integer) 4=y values
2–4	Test ID (6 character ASCII)
5–6	Target # (32 bit integer)
7–8	Julian date on film: DDDY (16 bit integer)
9–10	x (or y) value for target in frame 1: displacement from a fixed location in the filmplane in units of .00001 inches. Negative value indicates manual entry was made using the crosshairs. Zero indicates location of target was not defined for this frame. (32 bit integer)
.	
.	
.	
509–510	x (or y) value for target in frame 250
511	Not used
512	Not used (fixed length record)

March 29, 1979 Figure 1g

FILE FORMAT FOR UNIVAC

COMPATIBLE PHOTO DATA

RECORD SEQUENCE

BOT

Location of random Access table — 5 words

Header Time — 7 words

Target 1, x values
" y values
Target 2, x values
" y values

Target N, x values
" y values — 253 words (each record)

End-of-file

Repeat above format if more than 250 words were digitized

.
.
.

End-of-file
End-of-file

Complete data for one run

Repeat for each run

.
.
.

End-of-file
End-of-file
End-of-file (Triple EOF: end of data)
Random access table

SITE* ONE

* Above block of data copied from mass storage to tape using @ copy. There are 3 blocks of site one data copied to tape followed by 3 blocks of site 2 followed by 3 blocks of site 3.

May 5, 1982 Figure 2a

RECORD FORMAT FOR LOCATION OF RANDOM ACCESS TABLE FOR UNIVAC COMPATIBLE PHOTO DATA

WORD	CONTENTS
1	Sector address of random access table (integer)
2	Number of runs in this block of data (integer)
3	Site ID (integer)
4	Not used
5	Not used

May 5, 1982

Figure 2b

HEADER RECORD FORMAT FOR UNIVAC

COMPATIBLE PHOTO DATA

FIELD	CONTENTS
1	Run number (alpha - 6 characters) FLDATA
2	The word 'PHDATA' (alpha) FLDATA
3	Site ID (integer)
4	Time of first frame - 5 most significant digits (real)
5	Julian date: DDDY (alpha-6 char) Date digitized
6	Δt: expected time between frames (real)
7	Program version number (alpha-6 char)

May 5, 1982 Figure 2c

TIME RECORD FORMAT FOR UNIVAC

COMPATIBLE PHOTO DATA

FIELD	CONTENTS
1	Run number (alpha - 6 characters) FLDATA
2	The word 'CTSbbb' (alpha) FLDATA*
3	Site ID (integer)
4	Time of first frame, 5 least significant digits (real)
.	
.	
.	
.	
.	
.	
.	
.	
.	
253	Time of 250th frame

* C is the camera site ID (1, 2, or 3)

May 5, 1982

Figure 2d

DATA RECORD FORMAT FOR UNIVAC

COMPATIBLE PHOTO DATA

WORD	CONTENTS
1	Run number (alpha - 6 characters)
2	Displacement ID (alpha - 6 characters) (see Figure 4)
3	Site ID (integer)
4	X (or Y) value for target in frame 1: Displacement from a fixed location in meters. Negative value indicates manual entry was made using the crosshairs. A value of 999.0 indicates location of target was undefined for this frame. (REAL)
.	
.	
.	
.	
253	X (or Y) value for target in frame 250.

May 5, 1982

Figure 2e

RANDOM ACCESS DIRECTORY FORMAT

FOR UNIVAC COMPATIBLE PHOTO DATA

FIELD	CONTENTS
0	Words following in this record (integer)
1	RN(1) Run number for first run (alpha - 6 characters) FLDATA
2	SH(1) Sector address of header for first run (integer)
3	NR(1) Number of physical records of 253 words required to make one logical record for each target for first run. This occurs when more than 250 frames are digitized for a run. Output is written in multiples of 253 word physical records (integer)
4	ST(1) Sector address of time record of first physical record for first run (integer)
.	.
.	.
.	.
.	.
NR(1)+3	ST(1,NR(1)) Sector address of time record of NR physical records for first run (integer)

For J number of runs where X is the number of words used by all previous runs

FIELD	CONTENTS
1+X	RN(J) Run number for Jth run FLDATA
2+X	SH(J) Sector address of header for Jth run
3+X	NR(J) Number of physical records for Jth run
4+X	ST(J,1) Sector address of time record of 1st physical record for Jth run
.	.
.	.
.	.
.	.
NR(J)+X+3	ST(J,NR(J)) Sector address of time record of NR(J)th physical record for Jth run

See Figure 3 for example

May 5, 1982 Figure 2f

EXAMPLE OF RANDOM ACCESS TABLE WITH

SINGLE PHYSICAL RECORD FOR EACH TARGET

RUN NO.	SH	NR	ST(1)
LX3036	1	1	2
LX3037	134	1	135
LX3038	267	1	268
LX3039	400	1	401
LX3041	533	1	534
LX3042	666	1	667
LX3045	799	1	800
LX3047	932	1	933
LX3048	1065	1	1066
LX3050	1198	1	1199
LX3051	1331	1	1332
LX3053	1504	1	1505
LX3049	1637	1	1638

Figure 3

DISPLACEMENT ID (EZFLOW FORMAT)

CDMTTb 6 char ID (Field data)

where

C: camera site ID (1, 2, or 3)

D: coordinate component (X or Y) of data

M: mount system

- 1: neck*
- 2: mouth
- 3: top of head
- 4: pelvis
- 5: other

- 9: sprocket hole**

TT: target ID (01, 02, 03, ...)

* The chair reference target has the special mount/target ID of 100
** The sprocket hole data has the special mount/target ID of 900

Figure 4

FILM FRAME PLOTS

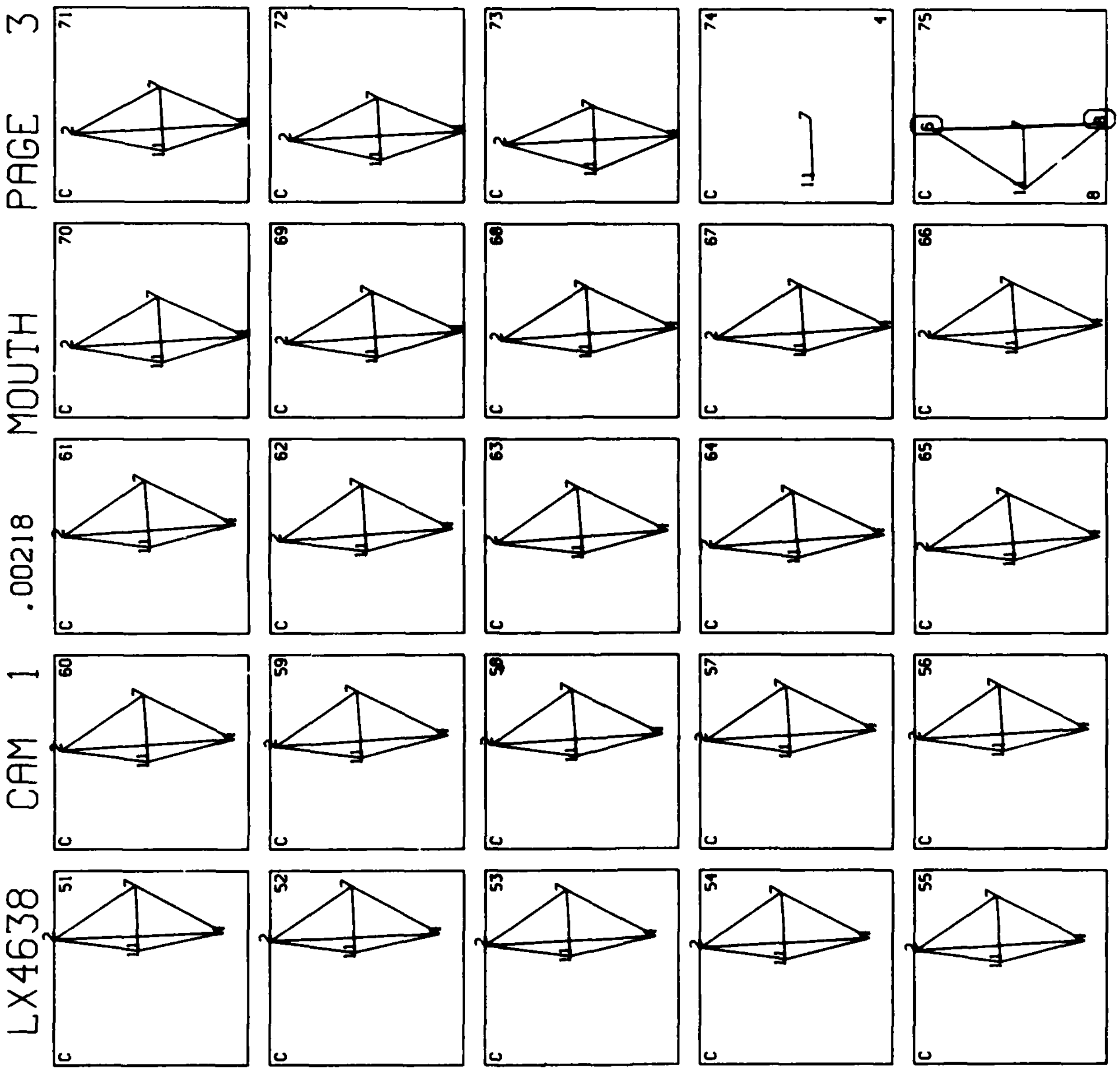


Figure 5a

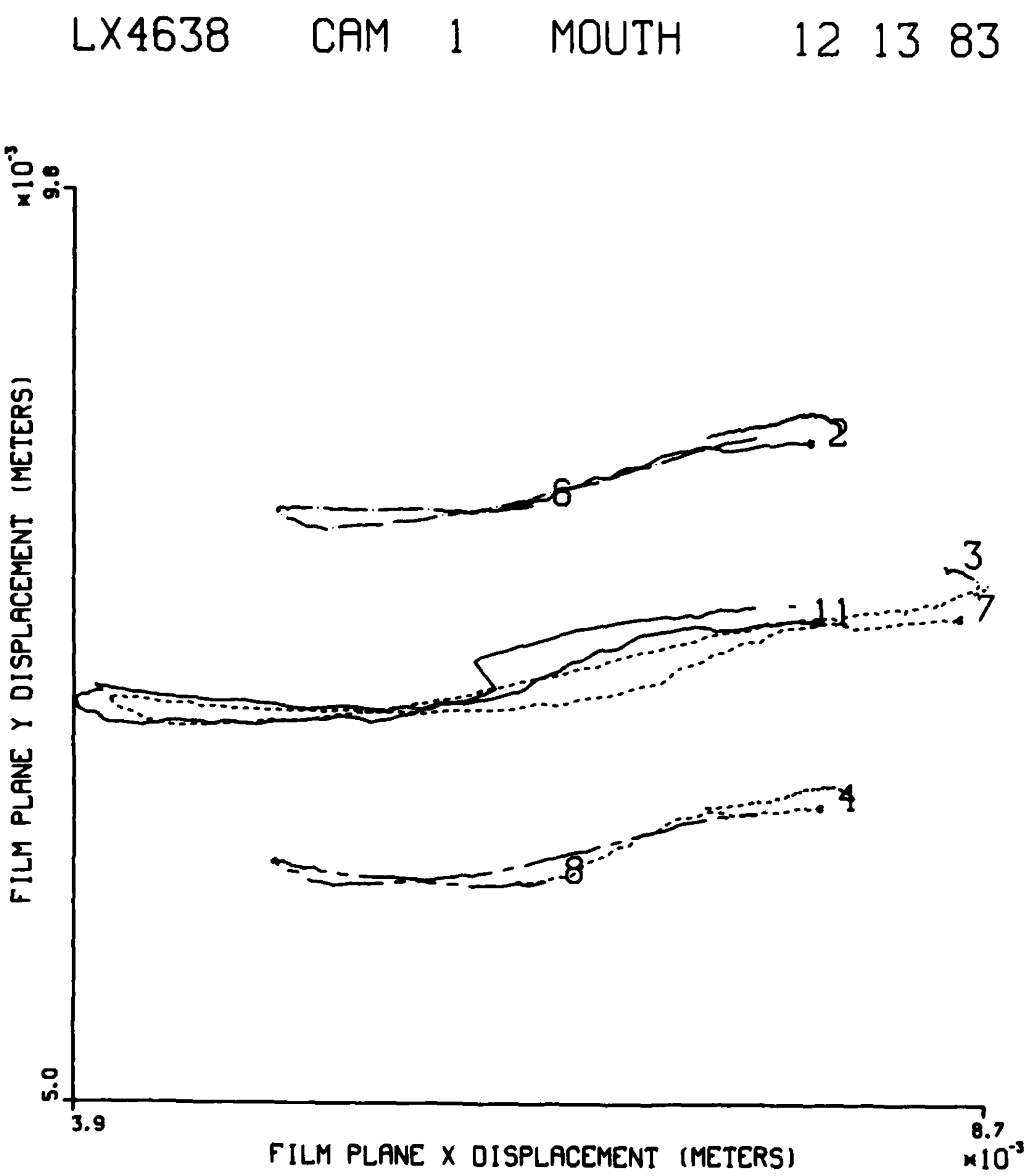
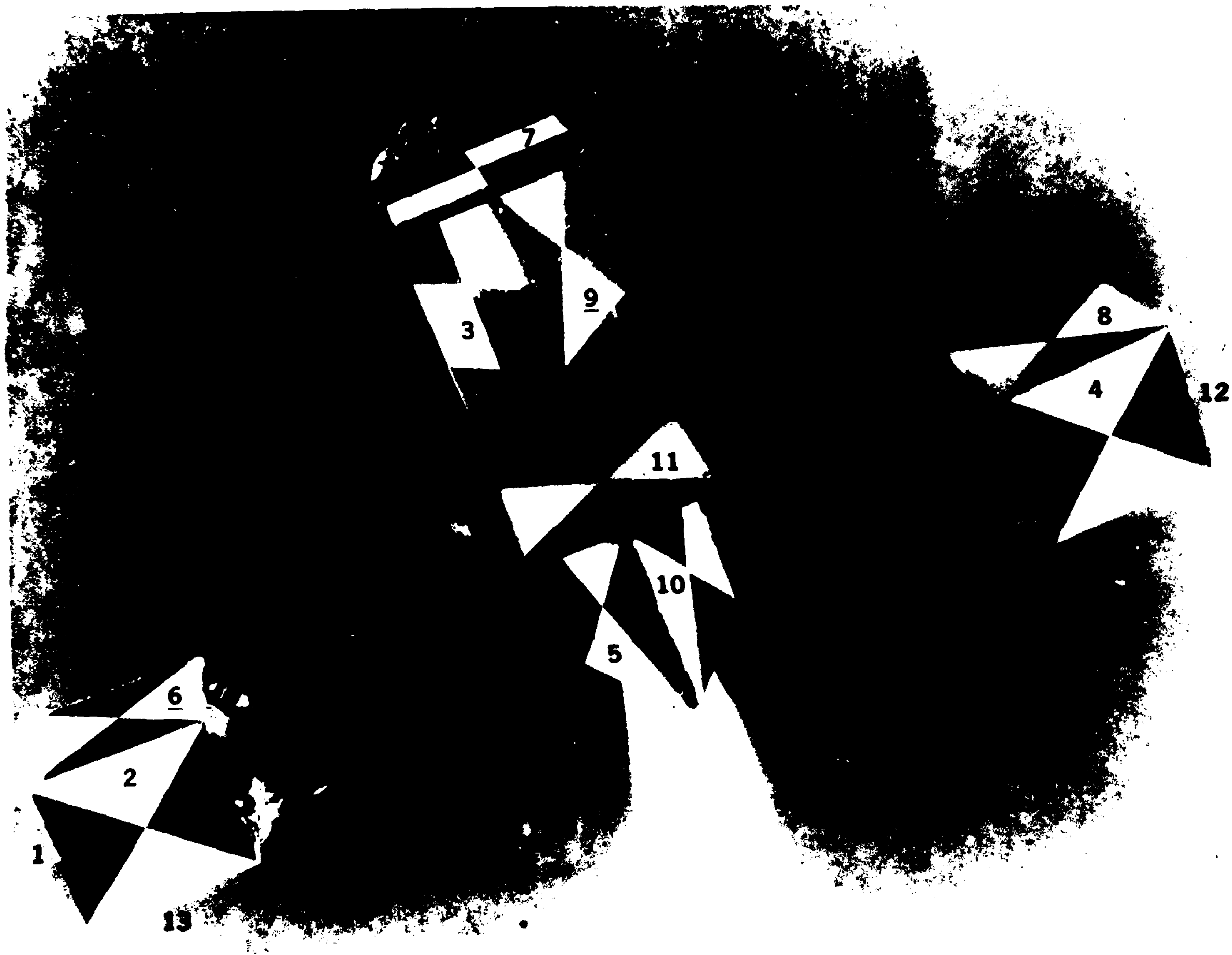

LX4638 CAM 1 MOUTH 12 13 83
FILM PLANE Y DISPLACEMENT (METERS)
$\times 10^{-3}$
9.8
5.0
3.9
8.7
FILM PLANE X DISPLACEMENT (METERS)
2
6
3
7
11
4
8

FIGURE 5b

RUN LX4642 CAMERA 1 PREDICTED RATE .002[illegible]

FRAME	D TIME	P TIME	FRAME	D TIME	P TIME	FRAME	D TIME	P TIME	FRAME	D TIME	P TIME	FRAME	D TIME	P TIME
1	5.37180	5.37180	51	5.47170	5.47166	101	5.57170	5.57152	151	5.67150	5.67138	201	5.77130	5.77123
2	5.37380	5.37380	52	5.47370	5.47366	102	5.57570	5.57351*	152	5.67360	5.67337	202	5.77330	5.77323
3	5.37560	5.37579	53	5.47510	5.47565	103	5.57570	5.57551	153	5.67550	5.67537	203	5.77530	5.77523
4	5.37760	5.37779	54	5.47110	5.47765*	104	5.57770	5.57751	154	5.67750	5.67737	204	5.77730	5.77723
5	5.37480	5.37979*	55	5.47410	5.47965*	105	5.57470	5.57951*	155	5.67450	5.67936*	205	5.11430	5.77922*
6	5.30180	5.38179*	56	5.48170	5.48164	106	5.58170	5.58150	156	5.68150	5.68136	206	5.78130	5.78122
7	5.38360	5.38378	57	5.48370	5.48364	107	5.58370	5.58350	157	5.68340	5.68336	207	5.78320	5.78322
8	6.38580	5.38578*	58	5.48570	5.48564	108	5.58570	5.58550	158	5.68540	5.68536	208	5.78520	5.78521
9	5.30780	5.38778*	59	5.49770	5.48764*	109	5.58770	5.58749	159	5.68740	5.68735	2[illegible]	5.78720	5.78721
10	5.38380	5.38977*	60	5.48970	5.48963	110	5.58470	5.58949*	160	5.68440	5.68935*	210	5.78430	5.78921*
11	5.34160	5.39177*	61	5.44170	5.49163*	111	5.54170	5.59149*	161	5.64140	5.69135*	211	5.74120	5.79121*
12	5.31180	5.39377*	62	5.44370	5.49363*	112	5.54370	5.59349*	162	5.64340	5.69334*	212	5.74120	5.79320*
13	5.19580	5.39577*	63	5.44570	5.49562*	113	5.54570	5.59548*	163	5.64540	5.69534*	213	5.14520	5.79520*
14	5.34780	5.39776*	64	5.44710	5.49762*	114	5.54760	5.59748*	164	5.64740	5.69734*	214	5.74720	5.79720*
15	5.34480	5.39976*	65	5.44470	5.49962*	115	5.54460	5.59948*	165	5.64440	5.69934*	215	5.74420	5.79919*
16	9.40180	5.40176*	66	5.50170	5.50162	116	5.60160	5.60147	166	5.70140	5.70133	216	5.80120	5.80119
17	5.40380	5.40375	67	5.50370	5.50361	117	5.60360	5.60347	167	5.10340	5.70333*	217	5.80320	5.80319
18	5.40500	5.40575	68	5.50570	5.50561	118	5.60560	5.60547	168	5.70540	5.70533	218	5.80520	5.80519
19	5.40180	5.40775*	69	5.50370	5.50761*	119	5.60160	5.60747*	169	5.10740	5.70732*	219	5.80720	5.80718
20	5.40980	5.40975	70	5.50970	5.50960	120	5.60460	5.60946*	170	5.70940	5.70932	220	5.80420	5.80918*
21	5.41180	5.41174	71	5.51170	5.51160	121	5.61160	5.61146	171	5.71140	5.71132	221	5.81120	5.81118
22	5.41380	5.41374	72	5.51370	5.51360	122	5.61360	5.61346	172	5.71340	5.71332	222	5.81320	5.81318
23	5.41580	5.41574	73	5.51570	5.51560	123	5.61560	5.61546	173	5.71540	5.71531	223	5.81520	5.81517
24	5.41780	5.41773	74	5.51770	5.51759	124	5.61760	5.61745	174	5.71740	5.71731	224	5.81720	5.81717
25	5.41480	5.41973*	75	5.51430	5.51959*	125	5.61460	5.61945*	175	5.71440	5.71931*	225	5.81420	5.81917*
26	5.42180	5.42173	76	5.52170	5.52159	126	5.62160	5.62145	176	5.72140	5.72131	226	5.82120	5.82116
27	5.42380	5.42373	77	5.52310	5.52359	127	5.62360	5.62344	177	5.12340	5.72330*	227	5.82320	5.82316
28	5.42580	5.42572	78	5.52570	5.52558	128	5.62560	5.62544	178	5.72540	5.72530	228	5.82520	5.82516
29	5.42780	5.42772	79	5.52170	5.52758*	129	5.62160	5.62744*	179	5.12130	5.72730*	229	5.82720	5.82716
30	5.42480	5.42972*	80	5.52470	5.52958*	130	5.62460	5.62944*	180	5.72930	5.72929	230	5.82420	5.82915*
31	5.13180	5.43172*	81	5.53170	5.53157	131	5.63160	5.63143	181	5.73130	5.73129	231	5.83120	5.83115
32	5.43380	5.43371	82	5.11370	5.53357*	132	5.63360	5.63343	182	5.73330	5.73329	232	5.83320	5.83315
33	5.43580	5.43571	83	5.53570	5.53557	133	5.63560	5.63543	183	5.73530	5.73529	233	5.83520	5.83514
34	5.43780	5.43771	84	5.53770	5.53757	134	5.63760	5.63742	184	5.73730	5.73728	234	5.83720	5.83714
35	5.43480	5.43970*	85	5.53410	5.53956*	135	5.63460	5.63942*	185	5.73930	5.73928	235	5.83420	5.83914*
36	5.44180	5.44170	86	5.54110	5.54156	136	5.64150	5.64142	186	5.74130	5.74128	236	5.84120	5.84114
37	5.44380	5.44370	87	5.54310	5.54356	137	5.64350	5.64342	187	5.14330	5.74327*	237	5.84320	5.84313
38	5.44580	5.44570	88	5.54570	5.54555	138	5.64550	5.64541	188	5.11530	5.74527*	238	5.84520	5.84513
39	5.14780	5.44769*	89	5.54710	5.54755	139	5.64750	5.64741	189	5.14[illegible]	5.74727*	239	5.84720	5.84713
40	5.41480	5.44969*	90	5.54410	5.54955*	140	5.64950	5.64941	190	5.1443[illegible]	5.74927*	240	5.84420	5.84912*
41	5.45180	5.45169	91	5.55170	5.55155	141	5.65150	5.65140	191	5.15130	5.75126*	241	5.85120	5.85112
42	5.45380	5.45368	92	5.55310	5.55354	142	5.65350	5.65340	192	5.75330	5.75326	242	5.85320	5.85312
43	5.45580	5.45568	93	5.55510	5.55554	143	5.65550	5.65540	193	5.75530	5.75526	243	5.85520	5.85512
44	5.45180	5.45768*	94	5.55770	5.55754	144	5.65750	5.65740	194	5.75730	5.75725	244	5.85720	5.85711
45	5.45980	5.45968	95	5.55470	5.55953*	145	5.65450	5.65939*	195	5.75430	5.75925*	245	5.85420	5.85911*
46	5.46180	5.46167	96	5.56170	5.56153	146	5.66150	5.66139	196	5.76130	5.76125	246	5.86120	5.86111
47	5.46380	5.46367	97	5.56370	5.56353	147	5.66350	5.66339	197	5.76330	5.76325	247	5.86310	5.86310
48	5.46570	5.46567	98	5.56570	5.56553	148	5.66550	5.66538	198	5.76530	5.76524	248	5.86510	5.86510
49	5.46770	5.46766	99	5.56770	5.56752	149	5.66750	5.66738	199	5.16730	5.76724*	249	5.86710	5.86710
50	5.46470	5.46966*	100	5.56410	5.56952*	150	5.66950	5.66938	200	5.76430	5.76924*	250	5.86910	5.869[illegible]0

FIGURE 5c



ANATOMICAL (MOUTH) MOUNT TARGET LABELS

Figure 6

PHOTO DATA REDUCTION RUNSTREAM

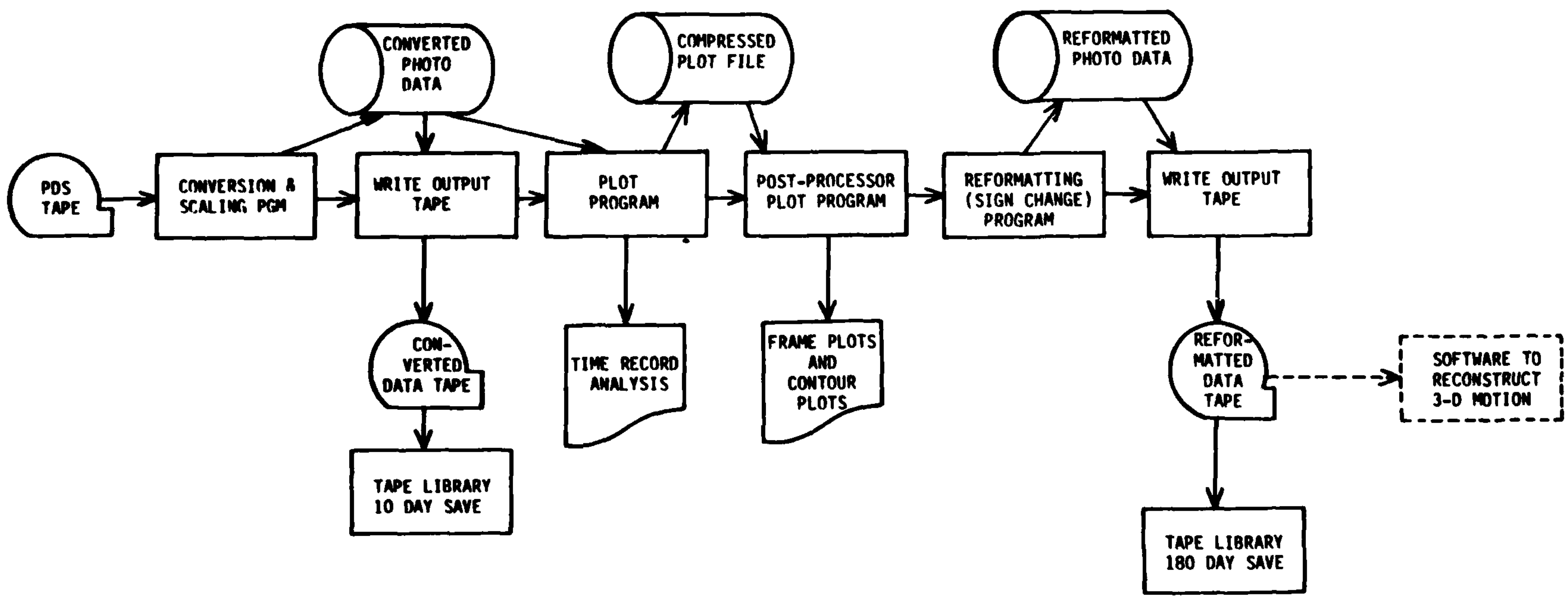


FIGURE 7

PHOTO DATA REDUCTION RUNSTREAM ELEMENT

```
@RUN
@PASSWD
@ASG,A DATASYSTEMS*PHOTOSTREAM.
@COPY DATASYSTEMS*PHOTOSTREAM.
@FREE DATASYSTEMS*PHOTOSTREAM.
@ED,U .XQTCVT                                 ENTER EDIT PROCESSOR
C /REELN/XXXXX/A                              WHERE XXXXX=CURRENT PDS REEL NUMBER
FIND@XQT                                      LOCATE PLACE TO INSERT CORRECTIONS
INPUT                                         INSERT CORRECTIONS (IF ANY) AS PER ELT .XQTCVT (APPENDIX 1a)
(    0    0    0)                             (INSERT ZEROES IF NO CORRECTIONS REQUIRED)
@EOF                                          EXIT INPUT MODE AND EDIT PROCESSOR
@ADD,EP .XQTCVT                               EXECUTE CONVERSION PROGRAM
@ED,U .XQTPLT                                 ENTER EDIT PROCESSOR
FIND@XQT                                      LOCATE PLACE TO INSERT PLOT OPTIONS
INPUT                                         INSERT PLOT OPTIONS AS PER ELT .XQTPLT (APPENDIX 2a)
   3    0    0    0    0      LX1111   LX9999 REFER TO .XQTPLT FOR OPTION DEFINITIONS
@EOF                                          EXIT INPUT MODE AND EDIT PROCESSOR
@ADD,EP .XQTPLT                               EXECUTE FRAME PLOT PROGRAM
@ED,U .XQTSGN                                 ENTER EDIT PROCESSOR
C /XXXXXXXX/11119999/A                        WHERE 11119999=FILE NAME OF 180-DAY UNIVAC OUTPUT DATA TAPE
FIND@XQT                                      LOCATE PLACE TO INSERT INPUT OPTIONS
INPUT                                         INSERT INPUT OPTIONS AS PER ELT .XQTSGN (APPENDIX 3a)
   3       LX1111   LX9999                    REFER TO .XQTSGN FOR OPTION DEFINITIONS
@EOF                                          EXIT INPUT MODE AND EDIT PROCESSOR
@ADD,EP .XQTSGN                               EXECUTE REFORMATTING PROGRAM
@FIN
```

FIGURE 8

REFERENCES

1. J. J. Lambert, "Development of ASCII FORTRAN I/O Program for Photo Data", (NBDL:72B:jbj, 5230, 1 June 1983), New Orleans, LA, 1983.

2. Integrated Software Systems Corporation, "DISSPLA User's Manual", San Diego, CA, 1981.

3. Sperry Corporation, "Sperry UNIVAC 1100 Executive System EXEC Level 38R5 Programmer Reference", Blue Bell, PA, 1982.

4. Sperry Corporation, "Sperry UNIVAC Series 1100 FORTRAN (ASCII) Level 10R1 Programmer Reference", Blue Bell, PA, 1982.

5. Sperry Corporation, "Sperry UNIVAC 1100 Series Text Editor (ED Processor) Level 16R1", Blue Bell, PA, 1980.

```
LAMBER1*TPF$(0).XQTCVT(26)
        @ASG,TJ     1.,U9H,REELN   .   INPUT DATA FROM DIGITIZER
        @ASG      9.,F///300
        @ASG      10.,F///300
        @ASG      11.,F///300
        @.
        @XQT      MP0051/PHOTOC
        12345678901234567890123456789012345678901234567890
        @.
        @.  ALL CONTROL STATEMENTS IMMEDIATELY AFTER @XQT STATEMENT
        @.  1ST CONTROL STATEMENT
        @.      POS 5   - NRD = NO. OF RUNS FOR WHICH TO CHANGE RUN NOS. MAX OF 6
        @.      POS 10  - NRB = NO. OF RUNS TO BYPASS ON THIS TAPE. MAX OF 6
        @.                       (SEE NOTE IN PGM MAIN)
        @.      POS 15  - NRC = NO. OF RUNS REQUIRING A CHANGE IN  CAM NO. MAX OF 6
        @.
        @.  2ND CONTROL STATEMENT (1 REQUIRED FOR EACH MAGNITUDE OF NRD)
        @.      POS 5-10   - LR = OLD RUN NUMBER
        @.      POS 15-20  - NR = NEW RUN NUMBER
        @.      POS 25     - IC = CAMERA NO. FOR WHICH RUN NO. IS TO BE CHANGED
        @.      POS 30     - IP = OCCURRENCE OF RUN NO. TO BE CHANGED, 1ST, 2ND, ETC.
        @.    MAY BE MORE THAN ONE OF ABOVE CONTROL STATEMENT
        @.
        @.  NEXT CONTROL STATEMENT (1 REQUIRED FOR EACH MAGNITUDE OF NRB)
        @.      POS 5-10   - IRN = RUN NUMBER TO BYPASS
        @.      POS 15     - ICN = CAMERA NO. FOR WHICH TO BYPASS RUN
        @.    MAY BE MORE THAN ONE OF ABOVE CONTROL STATEMENT
        @.
        @.  NEXT CONTROL STATEMENT (1 REQUIRED FOR EACH MAGNITUDE OF NRC)
        @.      POS 5-10   - LRN = RUN NUMBER
        @.      POS 15     - LCN = OLD CAMERA NUMBER
        @.      POS 20     - NCN = NEW CAMERA NUMBER
        @.      POS 25     - NOC = OCCURRENCE OF CAMERA TO CHANGE, 1ST, 2ND, ETC.
        @.    MAY BE MORE THAN ONE OF ABOVE CONTROL STATEMENT
        @.
        @FREE     1.
        @TEST     TNE/O/S6
        @JUMP     OUT
        @ASG,TF      A.,U9S,SAVE01   .   PREPLOTPHOTODATA
        @COPY,GM     9.,A.
        @TEST2:
        @TEST      TG/1/S6
        @JUMP      TEST3
        @COPY,GM     10.,A.
        @TEST3:
        @TEST      TG/2/S6
        @JUMP      OUT
        @COPY,GM     11.,A.
        @OUT:
        @FREE    A.
        @.  EXITED FROM .XQTCVT

@PRT,S .MAIN
```

```
LAMBERT*TPF$(O).MAIN(37)
      C   THIS PROGRAM WILL READ DIGITIZED PHOTODATA IN PDS OUTPUT
      C   FORMAT AND CONVERT TO UNIVAC 1182 FIELDATA FORMAT.  A
      C   SEPARATE FILE IS GENERATED FOR EACH CAMERA SITE ID.
      C
      C  NRD = NUMBER OF RUNS REQUIRING A CHANGE IN THE RUN NUMBER.
      C        THE RUN NUMBER MAY OR MAY NOT BE DUPLICATED.
      C  LR = OLD RUN NUMBER
      C  NR = NEW RUN NUMBER
      C  IC = CAMERA NUMBER FOR WHICH A RUN NUMBER IS TO BE CORRECTED
      C  IP = OCCURRENCE OF RUN NUMBER TO BE CHANGED, 1ST, 2ND, ETC.
      C
      C
      C  NRB = NUMBER OF RUNS TO BYPASS ON THIS TAPE
      C  IRN = RUN NUMBER TO SKIP
      C  ICN = CAMERA NUMBER FOR WHICH TO SKIP RUN
      C
      C         IT IS ASSUMED THAT WHEN A RUN IS TO BE BYPASSED, IT HAS
      C         BEEN PREVIOUSLY DIGITIZED AND PLACED IN THE FILE, AND
      C         LATER RE-DIGITIZED AND PLACED IN THE FILE AFTER A GROSS
      C         ERROR WAS DISCOVERED.  THE FIRST OCCURRENCE OF THE RUN IS
      C         ALWAYS BYPASSED.
      C
      C
      C   NRC = NUMBER OF RUNS REQUIRING A CHANGE IN CAMERA NUMBER
      C   LRN = RUN NUMBER
      C   LCN = OLD CAMERA NUMBER
      C   NCN = NEW CAMERA NUMBER
      C   NOC = OCCURRENCE OF CAMERA NUMBER TO BE CHANGED, 1ST, 2ND, ETC
      C
      C
            DIMENSION LCN(6), NCN(6), NOC(6), ITRY(6), ICCD(6)
            DIMENSION IC(6), IP(6), ICN(6), IPAS(6)
            DIMENSION ITABLE(5),JTABLE(4000),ICAM(3)
            COMMON/PHTHDR/RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER
            COMMON/I36/IOUT(3),DATA(250),IDL,IUNIT,ITARG
            CHARACTER*6 RUNID,VARID,LR(6),NR(6),IRN(6),FD2ASC,
           1LRN(6),JDTDIG,IPGVER,RUN(100,3)
      C
      C
            DATA IOUT/9,10,11/
       68   FORMAT(1X,I4)
            IERR=0
            IUNIT=1
      C
      C   READ CORRECTION INFORMATION
      C
            READ (5,230) NRD,NRB,NRC
            PRINT 230, NRD,NRB,NRC
      C
            IF (NRD.EQ.0) GO TO 20
            DO 10 J=1,NRD
              READ (5,240) LR(J),NR(J),IC(J),IP(J)
              PRINT 240, LR(J),NR(J),IC(J),IP(J)
      10    CONTINUE
      C
      20    IF (NRB.EQ.0) GO TO 40
            DO 30 J=1,NRB
              READ (5,250) IRN(J),ICN(J)
              PRINT 250, IRN(J),ICN(J)
```

```
 30    CONTINUE
 C
 40    IF (NRC.EQ.O) GO TO 60
       DO 50 J=1,NRC
         READ (5,250) LRN(J),LCN(J),NCN(J),NOC(J)
         PRINT 250, LRN(J),LCN(J),NCN(J),NOC(J)
 50    CONTINUE
 C
 60    CONTINUE
 C
       NC=O
 C
 70    CONTINUE
 C
 C     READ RECORD FROM PDS TAPE
 C
       CALL CONVT(ITYPE,IST)
       IF (IST.NE.4) GO TO 210
 C
       IF (ITYPE.NE.1) GO TO 70
       IF (NRD.EQ.O) GO TO 100
       DO 90 J=1,NRD
         IF (RUNID.EQ.LR(J).AND.ISITID.EQ.IC(J)) ITRY(J)=ITRY(J)+1
         IF (RUNID.EQ.LR(J).AND.ISITID.EQ.IC(J).AND.ITRY(J).EQ.IP(J))
      1  RUNID=NR(J)
 90    CONTINUE
 C
 100   IF (NRB.EQ.O) GO TO 120
       DO 110 J=1,NRB
         IF (RUNID.EQ.IRN(J).AND.ISITID.EQ.ICN(J).AND.IPAS(J).EQ.O)
      1  GO TO 190
 110   CONTINUE
 C
 C     PROCESS RUNS CONTAINING INCORRECT CAMERA NUMBERS
 C
 120   IF (NRC.EQ.O) GO TO 140
       DO 130 J=1,NRC
         IF(RUNID.EQ.LRN(J).AND.ISITID.EQ.LCN(J))
      1  ICCD(J)=ICCD(J)+1
         IF (RUNID.EQ.LRN(J).AND.ISITID.EQ.LCN(J).AND.NOC(J).EQ.
      1  ICCD(J)) ISITID=NCN(J)
 130   CONTINUE
 C
 140   CONTINUE
 C
       ISW=ISITID
       IF (ISW.LE.O.OR.ISW.GT.3) GO TO 150
 C
 C     DETERMINE WHICH CAMERA NUMBER TO PROCESS
 C
       GO TO (160,170,180), ISW
 C
 150   CONTINUE
       PRINT 270, RUNID,ISITID,ISW
       STOP 'CAM NO. ERROR'
 C
 160   CONTINUE
 C
 C     CONVERT DATA FOR CAMERA 1
 C
```

```
      CALL CNVT1(ITYPE,IST)
      IF (NC.LT.2) NC=1
      GO TO 70
C
170   CONTINUE
C
C     CONVERT DATA FOR CAMERA 2
C
      CALL CNVT1(ITYPE,IST)
      IF (NC.LT.3) NC=2
      GO TO 70
C
180   CONTINUE
C
C     CONVERT DATA FOR CAMERA 3
C
      CALL CNVT1(ITYPE,IST)
      NC=3
      GO TO 70
C
190   CONTINUE
      IPAS(J)=1
C
C     BYPASS CURRENT RUN
C
 200  CALL CONVT(ITYPE,IST)
      IF (IST.EQ.4) GO TO 200
      GO TO 70
C
210   CONTINUE
C
C     WRITE TRIPLE END-OF -FILE AT END OF DATA
C
      DO 220,I=1,NC
      DO 225,J=1,3
      CALL EOFPUT(IOUT(I),IER)
      IF(IER.NE.O)THEN
        PRINT 13,IER
        STOP 'ABORT IN MAIN'
      END IF
 225  CONTINUE
 220  CONTINUE
C
C     SET CONDITION WORD TO NUMBER OF CAMERAS
C
      CALL FCOND(ICOND)
      BITS(ICOND,25,12)=BITS(NC,25,12)
      CALL FSETC(ICOND)
C
C     PRINT DIRECT ACCESS TABLES
C
      PRINT 251
      MAX=4000
      DO 300,I=1,NC
      CALL GETDIR(IOUT(I),MAX,O,ITABLE,IER)
      IF(IER.NE.O)STOP 'DIRECTORY ERR'
      ICAM(I)=ITABLE(3)
      NRUNS=ITABLE(2)
      ISECT=ITABLE(1)
      PRINT 211,ICAM(I)
```

```
      CALL GETDIR(IOUT(I),MAX,ISECT,JTABLE,IER)
      IF(IER.NE.O)STOP 'DIRECTORY ERR'
      IPRINT=2
      DO 296,J=1,NRUNS
      RUN(NRUNS,ICAM(I))=FD2ASC(JTABLE(IPRINT),6)
      PRINT 221,RUN(NRUNS,ICAM(I)),(JTABLE(K),K=IPRINT+1,IPRINT+1+
     1JTABLE(IPRINT+2)+1)
      IPRINT=IPRINT+3+JTABLE(IPRINT+2)
  296 CONTINUE
      PRINT 231,(ITABLE(K),K=1,5)
  300 CONTINUE
C
      STOP
C
   13 FORMAT(1X,'STATUS ERROR = ',O3)
  230 FORMAT (3I5)
  240 FORMAT (2(4X,A6),2I5)
  250 FORMAT (4X,A6,3I5)
  260 FORMAT (' READ ERROR, STATUS = ',I3)
  270 FORMAT (' CAMERA NUMBER ERROR, RUN = ',A6,2I10)
  251 FORMAT('1')
  211 FORMAT(' DIRECT ACCESS TABLE FOR CAMERA ',I2)
  231 FORMAT(1X,5(I6,4X)//)
  221 FORMAT(1X,A6,2X,13I8)
C
      END
```

@PRT,S .CONVT

```
LAMBERT*TPF$(O).CNVT1(7)
            SUBROUTINE CNVT1(ITYPE,IST)
      C
      C     FUNCTION:
      C       CONTROL (1) READ OF DIGITIZED PHOTO DATA FROM PDS TAPE
      C       (2) CONVERSION TO UNIVAC COMPATIBLE FORMAT AND (3) SCALING
      C       OF DATA
      C
      C     ARGUMENT DEFINITIONS:
      C       ITYPE-RETURNED PDS RECORD TYPE WHERE
      C         1=HEADER RECORD
      C         2=TIME RECORD
      C         3=X DATA RECORD
      C         4=Y DATA RECORD
      C       IST-RETURNED ERROR STATUS WORD FROM ADTIO GENERAL
      C           PURPOSE I/O ROUTINE
      C
            COMMON/PHTHDR/RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER
            COMMON/I36/IOUT(3),DAT(250),IDL,IUNIT,ITARG
            CHARACTER*6 VAR1C1(28), VAR2C1(28), VAR3C1(28), VARCC1(28)
            CHARACTER*6 VAR1C2(28), VAR2C2(28), VAR3C2(28), VARCC2(28)
            CHARACTER*6 VAR3C3(28)
            CHARACTER*6 RUNID,VARID,JDTDIG,IPGVER,B(28),TARNUM*3,CAMNO*1
           1,OLDRUN
      C
      C     DEFINE MOUNT/TARGET ID'S
      C
            DATA VAR1C1 /'1X100','1Y100','1X102','1Y102',
           1'1X103','1Y103','1X101','1Y101','1X201','1Y201',
           2'1X204','1Y204','1X203','1Y203','1X308','1Y308',12*' '/
            DATA VAR2C1 /'1X100','1Y100','1X103','1Y103','1X105',
           1'1Y105','1X104','1Y104','1X101','1Y101','1X201','1Y201','1X204',
           2'1Y204','1X205','1Y205','1X203','1Y203','1X310','1Y310',8*' '/
            DATA VAR3C1 /'1X100','1Y100','1X212','1Y212','1X204',
           1'1Y204','1X208','1Y208','1X210','1Y210','1X202','1Y202','1X211',
           2'1Y211','1X209','1Y209','1X207','1Y207','1X010','1Y010',8*' '/
            DATA VARCC1 /'1X100','1Y100','1X002','1Y002','1X003',
           1'1Y003','1X004','1Y004','1X005','1Y005','1X201','1Y201','1X204',
           2'1Y204','1X205','1Y205','1X203','1Y203','1X310','1Y310',8*' '/
            DATA VAR1C2 /'2X100','2Y100','2X203','2Y203',
           1'2X204','2Y204','2X202','2Y202',20*' '/
            DATA VAR2C2 /'2X100','2Y100','2X205','2Y205','2X204',
           1'2Y204','2X202','2Y202','2X203','2Y203',18*' '/
            DATA VAR3C2 /'2X100','2Y100','2X103','2Y103',
           1'2X105','2Y105','2X101','2Y101','2X102',
           2'2Y102','2X201','2Y201','2X202','2Y202','2X205','2Y205','2X204',
           3'2Y204','2X203','2Y203','2X011','2Y011','2X012','2Y012',4*' '/
            DATA VARCC2 /'2X100','2Y100','2X103','2Y103','2X105','2Y105',
           1'2X101','2Y101','2X102','2Y102','2X201','2Y201','2X202',
           2'2Y202','2X205','2Y205','2X204','2Y204','2X203','2Y203',8*' '/
            DATA VAR3C3 /'3X100','3Y100','3X106','3Y106','3X111','3Y111',
           1'3X102','3Y102','3X101','3Y101','3X103','3Y103','3X107','3Y107',
           2'3X108','3Y108','3X105','3Y105','3X104','3Y104','3X109','3Y109',
           3'3X110','3Y110',4*' '/
      C
            OLDRUN='      '
      C
      C
      C     CONVERTS AND SCALES PHOTO DATA
      C
```

```
      DO 90 KP=1,1000
        DO 10 J=1,250
   10   DAT(J)=0.0
        IF (ITYPE.NE.1) GO TO 20
C
C
C     CHECK FOR MORE THAN 250 FRAMES OF DATA.
C
C
        IF (RUNID.EQ.OLDRUN) GO TO 30
        OLDRUN=RUNID
        VARID='PHDATA'
C
C     WRITE HEADER
C
        CALL PUTRCD(IOUT(ISITID),RUNID,VARID,ISITID,IDL,DAT,IER)
        IF(IER.NE.0)GO TO 112
        PRINT 190,RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER,ITYPE
C
C     READ NEXT PDS RECORD
C
   20   CALL CONVT(ITYPE,IST)
        IF (IST.NE.4) GO TO 100
        IF(ITYPE.EQ.1)GO TO 90
        GO TO 40
C
C
C     HANDLES EOF'S FOR MORE THAN 250 FRAMES
C
   30   CALL EOFPUT(IOUT(ISITID),IER)
        IF(IER.NE.0)GO TO 112
        CALL CONVT(ITYPE,IST)
        IF (IST.NE.4) GO TO 100
C
C
C     DETERMINE VARIABLE NAMES FOR A GIVEN CAMERA.
C
C
   40   CONTINUE
C     CHANGE DATA TO METERS & SET ZERO VALUES IN NON-TIME RECORDS EQUAL
        VARID='      '
        ENCODE (1,160,CAMNO) ISITID
        SUBSTR(VARID,1,1)=CAMNO
        IF(ITYPE.EQ.2)SUBSTR(VARID,2,2)='TS'
        IF (ITYPE.EQ.2) GO TO 70
        IF (BITS(ITARG,5,16).NE.0) GO TO 60
        IF (RUNID.LT.'LX0253') NDT=1
        IF (RUNID.LT.'LX1410') NDT=2
        IF (RUNID.GE.'LX1410') NDT=3
        IF ((RUNID.EQ.'LX1234').OR.(RUNID.EQ.'LX1235').OR.(RUNID.EQ.
     1'LX1236').OR.(RUNID.EQ.'LX1237')) NDT=4
        DO 50 J=1,28
          IF ((ISITID.EQ.1).AND.(NDT.EQ.1)) B(J)=VAR1C1(J)
          IF ((ISITID.EQ.1).AND.(NDT.EQ.2)) B(J)=VAR2C1(J)
          IF ((ISITID.EQ.1).AND.(NDT.EQ.3)) B(J)=VAR3C1(J)
          IF ((ISITID.EQ.1).AND.(NDT.EQ.4)) B(J)=VARCC1(J)
          IF ((ISITID.EQ.2).AND.(NDT.EQ.1)) B(J)=VAR1C2(J)
          IF ((ISITID.EQ.2).AND.(NDT.EQ.2)) B(J)=VAR2C2(J)
          IF ((ISITID.EQ.2).AND.(NDT.EQ.3)) B(J)=VAR3C2(J)
          IF ((ISITID.EQ.2).AND.(NDT.EQ.4)) B(J)=VARCC2(J)
```

```
          IF ((ISITID.EQ.3).AND.(NOT.EQ.3)) B(J)=VAR3C3(J)
 50     CONTINUE
        ITT=2*ITARG
        IF (ITYPE.EQ.3) VARID=B(ITT-1)
        IF (ITYPE.EQ.4) VARID=B(ITT)
        GO TO 70
 60     CONTINUE
        IF(ITYPE.EQ.3)SUBSTR(VARID,2,1)='X'
        IF(ITYPE.EQ.4)SUBSTR(VARID,2,1)='Y'
        NTC=BITS(ITARG,5,16)
        ENCODE (3,170,TARNUM) NTC
        SUBSTR(VARID,3,3)=TARNUM
 70     CONTINUE
        DO 80 J=1,250
          IF (ITYPE.EQ.2) DAT(J)=DAT(J)/10000.0
          IF (ITYPE.GT.2) DAT(J)=(DAT(J)/100000.0)*.0254
          IF ((ITYPE.GT.2).AND.(DAT(J).EQ.0.0)) DAT(J)=999.0
 80     CONTINUE
        IF ((RUNID.GE.'LX0253').AND.(VARID.EQ.'      ')) GO TO 90
C
C   CALL SPROCKET HOLE SUBROUTINE IF 900 OR 100 RECORD
C
        IF (SUBSTR(VARID,2,5).EQ.'X900 ') CALL SPH9X (DAT)
        IF (SUBSTR(VARID,2,5).EQ.'Y900 ') CALL SPH9Y (DAT)
        IF (SUBSTR(VARID,2,5).EQ.'X100 ') CALL SPH1X (DAT)
        IF (SUBSTR(VARID,2,5).EQ.'Y100 ') CALL SPH1Y (DAT)
C
C
C     WRITE TIME OR DISPLACEMENT RECORD
C
        CALL PUTRCD(IOUT(ISITID),RUNID,VARID,ISITID,IDL,DAT,IER)
        IF(IER.NE.0)GO TO 112
        PRINT 200,RUNID,VARID,ISITID,ITYPE,(DAT(J1),J1=1,10)
  90  CONTINUE
  100 CONTINUE
C
C     WRITE DOUBLE END-OF-FILE AT END OF RUN (END OF
C     DATA FOR 1 FILM)
C
      CALL EOFPUT(IOUT(ISITID),IER)
      IF(IER.NE.0)GO TO 112
      CALL EOFPUT(IOUT(ISITID),IER)
      IF(IER.NE.0)GO TO 112
C
C     CALCULATE SPROCKET HOLE AVERAGES
C
      CALL SPHAVG (SPHX,SPHY)
      PRINT 180, SPHX,SPHY
C
      RETURN
C
C     OUTPUT ABORT ROUTINE
C
  112 PRINT 12,IER
  12  FORMAT(1X,'STATUS ERROR = ',O3)
      STOP 'ABORT'
C
 160  FORMAT (I1)
 170  FORMAT (I3)
 180  FORMAT (/' SPROCKET HOLE AVERAGES, X = ',F10.6,'    Y = ',F10.6)
```

```
190   FORMAT ('1',1X,2(A6,4X),I6,4X,F10.3,4X,A6,4X,F6.0,4X,A6,4X,I6)
200   FORMAT (1X,2(A6,4X),I6,4X,I6,4X/1X,10(F11.8,2X))
C
      END

@PRT,S .SPHAVG
```

```
LAMBERT*TPF$(0).CONVT(34)
      SUBROUTINE CONVT(ITYPE,IST)
C
C     FUNCTION:
C       (1) READ PDS PHOTO DATA FROM TAPE AND (2) CONVERT
C       DIGITIZED PHOTO DATA TO UHNIVAC COMPATIBLE FORMAT
C
C     ARGUMENT DEFINITIONS:
C       ITYPE-RETURNED PDS RECORD TYPE WHERE
C         1=HEADER RECORD
C         2=TIME RECORD
C         3=X DATA RECORD
C         4=Y DATA RECORD
C       IST-RETURNED ERROR STATUS WORD FROM ADTIO GENERAL
C           PURPOSE I/O ROUTINE
C
      DIMENSION I32(300)
      COMMON/PHTHDR/RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER
      COMMON/I36/IOUT(3),DAT(250),IDL,IUNIT,ITARG
      CHARACTER*6 RUNID,VARID,JDTDIG,IPGVER
C
C     DATA FOR EXTENDING THE SIGN BIT.
C
      DATA NZ /0740000000000/ ,SIGN /0020000000000/
C
      NW=255
      CALL ATRD(IUNIT,I32,NW,IST)
      IF (IST.EQ.1) RETURN
C
C   ZERO OUT RECEIVING ARRAY.
C
      DO 10 I=1,250
        DAT(I)=0.
10    CONTINUE
C
C     CONVERT RUN NUMBER FROM NOVA ASCII TO UNIVAC ASCII
C
      CALL ASC8T9(I32,1,17,6,RUNID)
C
      ITYPE=BITS(I32(1),1,16)
      IF (ITYPE.GT.1) GO TO 60
C
C     EXTRACT HEADER RECORD AND CONVERT TO UNIVAC WORDS
C
      BITS(ITIME,5,8)=BITS(I32(2),29,8)
      BITS(ITIME,13,24)=BITS(I32(3),1,24)
      F1TIME=FLOAT(ITIME)
      BITS(ISITID,21,16)=BITS(I32(4),5,16)
      BITS(IDATE,21,12)=BITS(I32(3),25,12)
      BITS(IDATE,33,4)=BITS(I32(4),1,4)
      ENCODE(6,151,JDTDIG)IDATE
      BITS(IDELTA,33,4)=BITS(I32(5),1,4)
      DELTA=FLOAT(IDELTA)
      BITS(IVER,25,12)=BITS(I32(5),5,12)
      ENCODE(6,151,IPGVER)IVER
  151 FORMAT(I6)
      IF(IDATE.EQ.0)JDTDIG='      '
      IF(IVER.EQ.0)IPGVER='      '
C
      RETURN
```

```
60    IF (ITYPE.GT.2) GO TO 70
      GO TO 80
C
C     EXTRACT TARGET NUMBERS FROM INPUT BUFFER
C
 70   BITS(ITARG,5,8)=BITS(I32(2),29,8)
      BITS(ITARG,13,24)=BITS(I32(3),1,24)
      BITS(ISITID,5,12)=BITS(I32(3),25,12)
      BITS(ISITID,17,20)=BITS(I32(4),1,20)
C
C     EXTRACT DATA VALUES FOR DATA RECORD (OR TIME RECORD)
C
      IW=4
      IB=21
      GO TO 90
80    IW=5
      IB=17
90    DO 110 I=1,250
        ITEMP=0
        IS=5
        IL=37-IB
        IF (IL.GT.32) IL=32
        BITS(ITEMP,IS,IL)=BITS(I32(IW),IB,IL)
        IF (IL.EQ.32) GO TO 100
        IW=IW+1
        IB=1
        IS=4+IL+1
        IL=37-IS
        BITS(ITEMP,IS,IL)=BITS(I32(IW),IB,IL)
100     IB=IB+IL
        IF (IB.GT.36) IW=IW+1
C
C     EXTEND THE SIGN BIT FOR NEGATIVE NUMBERS AND CHANGE
C     2'S COMPLEMENT TO 1'S COMPLEMENT
C
        IF (AND(SIGN,ITEMP).NE.O) ITEMP=OR(ITEMP,NZ)-1
        IF (IB.GT.36) IB=1
        DAT(I)=FLOAT(ITEMP)
110   CONTINUE
      IDL=250
C
      RETURN
C
      END
```

@PRT,S .CNVT1

```
LAMBERT*TPF$(O).SPHAVG(11)
             SUBROUTINE SPHAVG (SPHX,SPHY)
             DIMENSION X(1), X1(250), X9(250), Y1(250), Y9(250)
      C
      C   THIS SUBROUTINE CALCULATES SPROCKET HOLE AVERAGES FOR EACH CAMERA
      C   FROM 5 CONTINUOUS FRAMES IN THE PHOTO DATA USING THE CHAIR OR SLED
      C   DISPLACEMENTS AND THE SPROCKET HOLE DISPLACEMENT.
      C   (THESE ARE THE 900 AND 100 RECORDS)
      C
      C     ARGUMENT DEFINITIONS:
      C       SPHX-RETURNED SPROCKET HOLE AVG. FOR X VALUES
      C       SPHY-RETURNED SPROCKET HOLE AVG. FOR Y VALUES
      C
             GO TO 50
      C
      C
      C   SAVE X AND Y DISPLACEMENTS
      C
             ENTRY SPH9X (X)
             DO 10 J=1,250
       10    X9(J)=X(J)
             ICX=ICX+1
             RETURN
      C
      C
             ENTRY SPH9Y (X)
             DO 20 J=1,250
       20    Y9(J)=X(J)
             ICY=ICY+1
             RETURN
      C
      C
             ENTRY SPH1X (X)
             DO 30 J=1,250
       30    X1(J)=X(J)
             ICX=ICX+1
             RETURN
      C
      C
             ENTRY SPH1Y (X)
             DO 40 J=1,250
       40    Y1(J)=X(J)
             ICY=ICY+1
             RETURN
      C
      C
       50    CONTINUE
      C
      C   CALCULATE DIFFERENCES IN SPROCKET HOLE AND SLED OR CHAIR
      C   TARGET DISPLACEMENTS FOR 5 CONSECUTIVE FRAMES,
      C   AND SUM THESE DIFFERENCES (X VALUES)
      C
             IBG=0
             IF (ICX.LT.2) GO TO 80
      C
             SUMX=0
             DO 70 J=1,250
               IF (X1(J).GT.900..OR.X9(J).GT.900.) GO TO 60
               SUMX=SUMX+(ABS(X1(J))-ABS(X9(J)))
               IF (IBG.EQ.0) IBG=J
```

```
              IF ((J-IBG).GE.4) GO TO 90
              GO TO 70
   60         SUMX=0
              IBG=0
   70       CONTINUE
      C
            IF (IBG.GE.250) IBG=0
      C
   80       SPHX=0
            GO TO 100
   90       CONTINUE
      C
      C
      C   CALCULATE SPROCKET HOLE AVERAGES AND CONVERT TO PDS UNITS
      C   FROM CENTIMETERS (X VALUE)
      C
            SPHX=(SUMX/5.)*(1000./(1.884765*.0254))
      C
  100       IF (ICY.LT.2) GO TO 120
      C
      C
      C   CALCULATE DIFFERENCES IN SPROCKET HOLE AND SLED OR CHAIR
      C   DISPLACEMENTS FOR 5 CONSECUTIVE FRAMES AND SUM THESE VALUES.
      C   USE THE SAME 5 FRAMES USED IN CALCULATING THE X VALUES ABOVE
      C    (Y VALUE)
      C
            IST=4
            IF (IBG.NE.0) IST=IBG
            IND=250
            IF (IBG.NE.0) IND=IST+4
      C
            SUMY=0
            DO 110 J=IST,IND
              IF (Y1(J).GT.900..OR.Y9(J).GT.900.) GO TO 120
              SUMY=SUMY+(ABS(Y1(J))-ABS(Y9(J)))
  110       CONTINUE
      C
            GO TO 130
      C
  120       SPHY=0
            GO TO 140
      C
  130       CONTINUE
      C
      C
      C   CALCULATE SPROCKET HOLE AVERAGE AND CONVERT TO PDS UNITS
      C   FROM CENTIMETERS (Y VALUE)
      C
            SPHY=(SUMY/5.)*(1000./(1.884765*.0254))
      C
  140       ICX=0
            ICY=0
      C
            RETURN
      C
            END
```

@PRT,S .ASC8T9

```
LAMBERT*TPF$(O).ASC8T9(17)
          SUBROUTINE ASC8T9(INPUTA,IWD,IBIT,NCHAR,CSTRNG)
C
C         FUNCTION:
C           CONVERT STRING OF 8-BIT ASCII CHARACTERS TO
C           UNIVAC COMPATIBLE 9-BIT ASCII CHARACTER STRING
C
C         ARGUMENT DEFINITIONS:
C           INPUTA-INTEGER INPUT ARRAY OF 8-BIT CHARACTER DATA
C           IWD-STARTING WORD IN INPUTA OF 8-BIT ASCII CHARACTER STRING
C           IBIT-BEGINNING BIT IN IWD OF INPUT 8-BIT ASCII CHARACTER STRING
C           NCHAR-NUMBER ASCII CHARACTERS TO BE CONVERTED TO UNIVAC FORMAT
C           CSTRNG-RETURNED OUTPUT STRING OF UNIVAC 9-BIT ASCII CHARACTERS
C
C         PROGRAMMER:  J LAMBERT  20 MAY 1983
C
          INTEGER INPUTA(1)
C
C         LENGTH OF CHARACTER STRING MUST BE INITIALIZED IN CALLING PROGRAM
C
          CHARACTER*(*) CSTRNG,ITEMP*1
          DATA IL1,IL2/2*0/
          IB=IBIT
          IW=IWD
C
C         1ST BIT OF UNIVAC ASCII CHARACTER ALWAYS O(ZERO)
C
          BITS(ITEMP,1,1)=0
C
C         CONVERT EACH 8-BIT CHARACTER TO 9-BITS
C
          DO 20,I=1,NCHAR
          IF((IB+7).LT.36)THEN
            BITS(ITEMP,2,8)=BITS(INPUTA(IW),IB,8)
          ELSE
            IL1=37-IB
            IL2=8-IL1
            BITS(ITEMP,2,IL1)=BITS(INPUTA(IW),IB,IL1)
            BITS(ITEMP,2+IL1,IL2)=BITS(INPUTA(IW+1),1,IL2)
          END IF
          SUBSTR(CSTRNG,I,1)=SUBSTR(ITEMP,1,1)
          IB=IB+8
          IF(IB.GT.36)THEN
            IW=IW+1
            IB=IB-36
          END IF
   20     CONTINUE
          RETURN
          END

@PRT,S .XQTPLT
```

```
LAMBERT*TPF$(O).XQTPLT(11)
        @XQT MPOO51/PLOT
        @.
        @. CONTROL STATEMENT IMMEDIATELY AFTER @XQT STATEMENT
        @.      POS 5 - NO. INPUT UNITS (1, 2 OR 3)
        @.      POS 10 - TIME RECORD ANALYSIS PLOT OPTION
        @.               O = OUTPUT TO LINE PRINTER
        @.               1 = OUTPUT TO COMPRESSED PLOT FILE
        @.               NOT O OR 1 = NO TIME RECORD ANALYSIS
        @.      POS 15 - XY CONTOUR PLOT OPTION
        @.               O = PLOT XY CONTOUR FOR ALL MOUNTS/SITES/RUNS
        @.               NOT O = NO XY CONTOUR PLOTS
        @.      POS 20 - FRAME PLOT OPTION
        @.               O = FRAME PLOTS FOR ALL MOUNTS/SITES/RUNS
        @.               NOT O = NO FRAME PLOTS
        @.      POS 25 - FORMAT OPTION FOR FRAME PLOTS
        @.               O = COL'S 2 & 4 UP -- COL'S 1, 3 & 5 DOWN
        @.               NOT O = ALL COL'S DOWN
        @.      POS 35 - LOWEST RUN NO. TO BE PROCESSED
        @.      POS 45 - HIGHEST RUN NO. TO BE PROCESSED
        @XQT SCC$*INTLIB.FR8OPOP
        MO4O
        DATASYSTEMS
        5-4890
        FRAME PLOTS
        FRAME PLOTS
        1 HARDCOPY  1 MICROFICHE


        DRAW=1-END$
        @DELETE,C COMPRESS.
        @FREE COMPRESS.
        @ASG,CP COMPRESS.,F///1000
        @COPY 17.,COMPRESS.
        @FREE COMPRESS.
        @. EXITED FROM .XQTPLT

@PRT,S .PLOT
```

```
LAMBERT*TPF$(O).PLOT(9)
      C
      C     FUNCTION:
      C       THIS ASCII FORTRAN DRIVER PROGRAM CONTROLS THE GENERATION
      C       OF MOUNT PLOTS FOR A RANGE OF RUNS SPECIFIED BY THE USER.
      C       TWENTY-FIVE FRAMES (DATA PTS.) ARE PLOTTED ON EACH OF 10
      C       PAGES FOR A GIVEN MOUNT AND SITE ID.  THE DISSPLA PLOT
      C       PACKAGE IS USED FOR ALL PLOTTING.
      C
            COMMON/LOAD/XMIN,XMAX,IPGE,YMIN,YMAX,TITLE,IST,ILT,IX,XOR,
           *YOR,VARX,VARY,IERR,STEP,DC(250)
            COMMON/PHTHDR/RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER
            CHARACTER*6 RUNID,VARID,JDTDIG,IPGVER,TITLE(9),VARX(12),VARY(12)
            CHARACTER*6 FRUN,LRUN,RUNS(200),FIRST,LAST,ENC*4,SITE*1
            DATA MAX/280/
      C
      C     READ USER OPTIONS IN CARD IMAGE
      C
            READ(5,6) IFILE,ITIM,IXY,IMTPLT,IWAY,FRUN,LRUN
       6    FORMAT(5I5,9X,2(A6,4X))
      C
      C     WHERE:
      C       IFILE: NO. OF INPUT FILES
      C       ITIM: TIME RECORD ANALYSIS PLOT OPTION
      C             O=OUTPUT TO LINE PRINTER
      C             1=OUTPUT TO COMPRESSED PLOT FILE
      C             NOT O AND NOT 1 = NO TIME RECORD ANALYSIS
      C       IXY: XY CONTOUR PLOT OPTION
      C            O=PLOT XY CONTOUR FOR EACH MOUNT/SITE/RUN
      C            NOT O = NO XY CONTOURS
      C       IMTPLT: MOUNT PLOT OPTION
      C               O=FRAME PLOTS REQUIRED FOR ALL DATA
      C               NOT O = NO FRAME PLOTS
      C       IWAY: PLOT DIRECTION OPTION
      C              O=BOTTOM TO TOP COLUMNS 2 AND 4
      C              NOT O = TOP TO BOTTOM ALL COLUMNS
      C       FRUN: LOW ORDER VALUE OF RUN NO. RANGE
      C       LRUN: HIGH ORDER VALUE OF RUN NO. RANGE
      C
      C
      C     SET COMPRESSOR FOR PLOTTING
      C
            CALL COMPRS
            CALL SETDEV(6,0)
            CALL GRACE(0.)
            CALL PAGE(14./1.02,[illegible]/1.02)
            CALL NOBRDR
            CALL ERASE
      C
      C     ONE UNIT = 10.5/1.02 INCHES
      C
            CALL UNITS(10.5/1.02)
      C
      C     COMPUTE RUN RANGE
      C
            FIRST=SUBSTR(FRUN,3,4)
            LAST=SUBSTR(LRUN,3,4)
            DECODE(4,FIRST)IFIRST
            DECODE(4,LAST)ILAST
       4    FORMAT(I4)
```

```
C
C     COMPUTE INPUT UNIT ASSIGNMENTS
C
      ILIM=IFILE+8
      IU=9
    1 CONTINUE
      PRINT 3,IU,ILIM,ITIM,IXY,IMTPLT,IWAY,FRUN,LRUN
    3 FORMAT(6I10,2(4X,A6))
C
C     GENERATE TITLE FOR EACH PAGE
C
      TITLE(2)='    CAM'
      TITLE(5)='       '
      TITLE(6)='    PAG'
      TITLE(7)='E      '
      TITLE(9)='       '
      TITLE(3)='       '
      TITLE(4)='       '
      TITLE(8)='       '
C
C     PLOT DATA FOR EACH RUN
C
      IRN=IABS(ILAST-IFIRST)+1
      INC=IFIRST
      DO 500,I=1,IRN
C
C     CREATE RUN ARRAY TO BE SEARCHED
C
      SUBSTR(RUNS(I),1,2)='LX'
      ENCODE(17,ENC)INC
   17 FORMAT(I4)
      SUBSTR(RUNS(I),3,4)=ENC
      INC=INC+1
C
C     ATTEMPT TO SET SECTOR TO HEADER FOR RUN
C
      CALL SETSCT(IU,RUNS(I),'PHDATA',1,ISITE,NREC,IER)
      IF(IER.NE.0)GO TO 500
      CALL GETRCD(IU,MAX,RUNID,VARID,ISITE,IDL,DC,IER)
      IF(IER.NE.0)THEN
        PRINT 13,IER
   13   FORMAT(1X,'STATUS=',O3)
        STOP 'HEADER ERROR'
      END IF
      IF(VARID.NE.'PHDATA')STOP 'HEADER ERROR'
C
C     GENERATE TIME RECORD ANALYSIS AND WRITE ON FICHE
C
      ENCODE(10,SITE)ISITID
   10 FORMAT(I1)
      SUBSTR(TITLE(3),3,1)=SITE
      TITLE(1)=RUNID
      IF(ITIM.EQ.0 .OR. ITIM.EQ.1)THEN
        CALL TIMER(IU,ITIM)
        IF(IERR.NE.0)GO TO 500
      END IF
C
C     PLOT DATA FOR EACH MOUNT FOR THIS RUN
C
      DO 400,L=1,4
```

```
C
C     LOAD DATA
C
      CALL LDATA(IU,L)
      IF(IX.NE.0)THEN
        IST=1
        IFM=1
      ELSE
        GO TO 400
      END IF
C
C     GENERATE TITLE INFORMATION FOR THIS MOUNT
C
      IF(L.EQ.1)THEN
        TITLE(5)='NECK   '
      ELSE IF(L.EQ.2)THEN
        TITLE(5)='MOUTH  '
      ELSE IF(L.EQ.3)THEN
        TITLE(5)='HEAD   '
      ELSE IF(L.EQ.4)THEN
        TITLE(5)='PELVIS'
      END IF
C
C     LOOP FOR EACH PAGE OF PLOTS (FRAMES)
      IF(IMTPLT.NE.0)GO TO 300
C
      DO 55,IPGE=1,10
C
C     DEFINE PHYSICAL ORIGIN FOR FIRST FRAME ON PAGE
C
      XOR=.16
      YOR=.81
C
C     DETERMINE MAX AND MIN VALUES FOR THIS PAGE OF PLOTS
C
      ILT=IST+24
      CALL MIXY
C
C     PRINT TITLE AT TOP OF PAGE
C
      CALL PLOT1
C
C     LOOP TO CONTROL COLUMN OF PLOTS
C
      DO 50,K=1,5
      IC=MOD(K,2)
C
C     PLOT DATA 5 FRAMES FOR EACH OF 5 COLUMNS
C
      DO 25,J=1,5
      CALL PLOT2(IFM)
C
C     CHECK FOR DIRECTION OF PLOTS (UP OR DOWN)
C
      IF(IWAY.NE.0)THEN
C
C     PLOT TOP TO BOTTOM ALL COLUMNS
C
        IF(J.NE.5)YOR=YOR-.2
      ELSE
```

```
C
C     PLOT BOTTOM TO TOP COLUMNS 2 AND 4
C
        IF(J.NE.5)THEN
          IF(IC.NE.O)THEN
            YOR=YOR-.2
          ELSE
            YOR=YOR+.2
          END IF
        END IF
      END IF
C
C     CHECK FOR LAST DATA POINT
C
      IF(IST.LT.250)THEN
        IST=IST+1
        IFM=IFM+1
      ELSE
        CALL ENDPL(O)
        GO TO 55
      END IF
   25 CONTINUE
C
C     MODIFY PHYSICAL ORIGIN FOR NEXT COLUMN
C
      IF(IWAY.NE.O)THEN
        XOR=XOR+.2
        YOR=.81
      ELSE
        XOR=XOR+.2
      END IF
   50 CONTINUE
      CALL ENDPL(O)
   55 CONTINUE
C
C     PLOT X VS Y ALL FRAMES
C
  300 IF(IXY.EQ.O)CALL XYPLOT
  400 CONTINUE
  500 CONTINUE
      IF(IU.LT.ILIM)THEN
        IU=IU+1
        GO TO 1
      END IF
      CALL DONEPL
      STOP
      END
```

@PRT,S .PLOTD

```
LAMBERT*TPF$(O).TIMER(33)
            SUBROUTINE TIMER(IU,ITIM)
      C
      C     FUNCTION:
      C       GENERATE AND PLOT TIME RECORD ANALYSIS FOR A GIVEN RUN
      C
      C     ARGUMENT DEFINITIONS:
      C       IU: LOGICAL UNIT NO. OF INPUT FILE
      C       ITIM: OPTION TO PLOT TIME RECORD ANALYSIS WHERE
      C             O = OUTPUT TO LINE PRINTER
      C             1 = OUTPUT TO COMPRESSED PLOT FILE
      C
            COMMON/LOAD/XMIN,XMAX,IPGE,YMIN,YMAX,TITLE,IST,ILT,IX,XOR,
           *YOR,VARX,VARY,IERR,STEP,DC(250)
            COMMON/PHTHDR/RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER
            CHARACTER*6 RUNID,VARID,JDTDIG,IPGVER,TITLE(9),VARX(12),VARY(12)
            CHARACTER*6 SYM*1(250),LTINT,LDD*9(250),LETIME*9(250)
            CHARACTER*8 LINDEX*3(250),LPDST,LEXPT,LFRNO*7
            CHARACTER*128 LINE,LHEAD(2)
            DIMENSION ETIME(250),DD(250)
            DATA LPDST,LEXPT,LFRNO/'D TIME','P TIME','FRAME '/
            IERR=1
            CALL HEIGHT(.018)
      C
      C     SET SECTOR FOR TIME RECORD
      C
            VARID=' TS   '
            SUBSTR(VARID,1,1)=SUBSTR(TITLE(3),3,1)
            CALL SETSCT(IU,RUNID,VARID,1,ISIT,NREC,IER)
            IF(IER.NE.O)THEN
              PRINT 13,IER
       13     FORMAT(1X,'STATUS=',O3)
              GO TO 100
            END IF
      C
      C     READ TIME RECORD
      C
            MAX=280
            CALL GETRCD(IU,MAX,RUNID,VARID,ISIT,IDL,DD,IER)
            IF(IER.NE.O)THEN
              IF(IER.EQ.1)THEN
                GO TO 100
              ELSE
                PRINT 13,IER
                STOP 'TIME ERROR'
              END IF
            END IF
      C
      C     CHECK FOR TIME RECORD IDENTIFICATION
      C
            IF(SUBSTR(VARID,2,2).NE.'TS')THEN
              PRINT 1,RUNID
        1     FORMAT(1X,'COULD NOT FIND TIME RECORD FOR RUN ',A6)
              STOP 'TIME ERROR'
            ELSE
              IERR=O
            END IF
      C
      C     COMPUTE TIME INTERVAL BETWEEN FRAMES
      C
```

```
      DO 15,K=250,1,-1
      IF(DD(K).LE.O)THEN
        CONTINUE
      ELSE
        DO 21,J=1,250
        IF(DD(J).LE.O)THEN
          CONTINUE
        ELSE
          IF(DD(K).LT.DD(J))DD(K)=DD(K)+10.
          TINT=(DD(K)-DD(J))/(K-J)
          GO TO 32
        END IF
 21     CONTINUE
        GO TO 100
      END IF
 15   CONTINUE
      PRINT 16,RUNID
 16   FORMAT(1X,'TIME RECORD FOR RUN ',A6,'WAS ALL ZEROES')
      GO TO 100
 32   ETIME(1)=DD(1)
      SYM(1)=' '
C
C     COMPUTE EXPECTED TIMES
C
      DO 35,JJ=2,250
      ETIME(JJ)=ETIME(JJ-1)+TINT
      SYM(JJ)=' '
      IF(DD(JJ) .LT. .000001)THEN
        CONTINUE
      ELSE
        DIF=ETIME(JJ)-DD(JJ)
        IF(ABS(DIF) .GT. .001)SYM(JJ)='*'
      END IF
 35   CONTINUE
C
C     WRITE HEADER INFO FOR TIME RECORD ANALYSIS
C
      ENCODE(11,LTINT)TINT
 11   FORMAT(F6.5)
      IF(ITIM.EQ.1)THEN
        CALL TABLET('CENTER','LONG')
      ELSE IF(ITIM.EQ.O)THEN
        PRINT 56
 56     FORMAT('1')
      END IF
      SUBSTR(LHEAD(1),1,4)='RUN '
      SUBSTR(LHEAD(1),5,6)=RUNID
      SUBSTR(LHEAD(1),11,11)=' '
      SUBSTR(LHEAD(1),22,7)='CAMERA '
      SUBSTR(LHEAD(1),29,3)=SUBSTR(TITLE(3),1,3)
      SUBSTR(LHEAD(1),32,11)=' '
      SUBSTR(LHEAD(1),43,15)='PREDICTED RATE '
      SUBSTR(LHEAD(1),58,6)=LTINT
      IF(ITIM.EQ.1)THEN
        SUBSTR(LHEAD(1),64,1)='$'
        CALL CTLINE(LHEAD(1))
        CALL CTLINE(' $')
      ELSE IF(ITIM.EQ.O)THEN
        PRINT 14,LHEAD(1)
 14     FORMAT(21X,A63/)
```

```
      END IF
C
C     WRITE HEADER INFO. FOR INDIVIDUAL COLUMNS
C
      CALL HEIGHT(.01)
      DO 23,IREP=1,5
      SUBSTR(LHEAD(2),(((IREP-1)*23)+1),7)=LFRNO
      SUBSTR(LHEAD(2),(((IREP-1)*23)+8),8)=LPDST
      SUBSTR(LHEAD(2),(((IREP-1)*23)+16),8)=LEXPT
 23   CONTINUE
      IF(ITIM.EQ.1)THEN
        SUBSTR(LHEAD(2),116,1)='$'
        CALL CTLINE(LHEAD(2))
        CALL CTLINE(' $')
      ELSE IF(ITIM.EQ.0)THEN
        PRINT 17,LHEAD(2)
 17     FORMAT(1X,A115/)
      END IF
C
C     PRINT TIME RECORD ANALYSIS
C
      DO 60,KK=1,50
      DO 71,IENC=1,5
      INDEX=(((IENC-1)*50)+KK)
      ENCODE(8,LINDEX(INDEX))INDEX
      ENCODE(9,LDD(INDEX)) DD(INDEX)
      ENCODE(9,LETIME(INDEX)) ETIME(INDEX)
 9    FORMAT(F9.5)
 8    FORMAT(I3)
      SUBSTR(LINE,(((IENC-1)*23)+1),1)=' '
      SUBSTR(LINE,(((IENC-1)*23)+2),3)=LINDEX(INDEX)
      SUBSTR(LINE,(((IENC-1)*23)+5),9)=LDD(INDEX)
      SUBSTR(LINE,(((IENC-1)*23)+14),9)=LETIME(INDEX)
      SUBSTR(LINE,(((IENC-1)*23)+23),1)=SYM(INDEX)
 71   CONTINUE
      IF(ITIM.EQ.1)THEN
        SUBSTR(LINE,116,1)='$'
      CALL CTLINE(LINE)
      ELSE IF(ITIM.EQ.0)THEN
        PRINT 18,LINE
 18     FORMAT(1X,A115)
      END IF
 60   CONTINUE
      IF(ITIM.EQ.0)THEN
        PRINT 78
 78     FORMAT('1')
      END IF
 100  CONTINUE
      CALL RESET('HEIGHT')
      IF(ITIM.EQ.1)THEN
        CALL ENDTAB(0)
        CALL ENDPL(0)
      END IF
      RETURN
      END
```

@PRT,S .XOTSGN

LAMBERT*TPF$(O).LDATA(13)

```
      SUBROUTINE LDATA(IU,MNT)
C
C     FUNCTION:
C       LOAD PHOTO DISPLACEMENT DATA INTO COMMON FOR
C       ACCESS BY PLOT ROUTINES
C
C     ARGUMENT DEFINTIONS:
C       IU: LOGICAL UNIT NO. OF INPUT DATA FILE
C       MNT: MOUNT INDEX
C
      COMMON/LOAD/XMIN,XMAX,IPGE,YMIN,YMAX,TITLE,IST,ILT,IX,XOR,
     *YOR,VARX,VARY,IERR,STEP,DC(250),DX(250,12),DY(250,12)
      COMMON/PHTHDR/RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER
      DIMENSION DD(250)
      CHARACTER*6 RUNID,VARID,JDTDIG,IPGVER,TITLE(9),VARX(12),VARY(12)
      CHARACTER*2 ONE/'01'/,TGCHK/'00'/,MTCHK*1/'0'/
      CHARACTER*3 TARGET,OLDRUN*6
      DATA MAX/280/
      IX=0
      IY=0
C
C     INITIALIZE CHAIR DATA FLAG
C
      IF(OLDRUN.NE.RUNID)ICHAIR=0
C
C     SET SECTOR TO TIME RECORD
C
      VARID=' TS   '
      SUBSTR(VARID,1,1)=SUBSTR(TITLE(3),3,1)
      CALL SETSCT(IU,RUNID,VARID,1,ISIT,NREC,IER)
      IF(IER.NE.0)THEN
        PRINT 13,IER
   13   FORMAT(1X,'STATUS=',O3)
        STOP 'LDATA'
      END IF
C
C     GET NEXT DATA RECORD
C
    8 CALL GETRCD(IU,MAX,RUNID,VARID,ISIT,IDL,DD,IER)
      IF(IER.NE.0)THEN
        IF(IER.EQ.1)THEN
          GO TO 56
        ELSE
          PRINT 13,IER
          GO TO 72
        END IF
      END IF
C
C     CHECK FOR CHAIR DATA AND STORE
C
      IF(ICHAIR.EQ.1)THEN
        CONTINUE
      ELSE
        TARGET=SUBSTR(VARID,3,3)
        IF(TARGET.EQ.'100')THEN
          DO 16,J=1,250
   16     DC(J)=DD(J)
          PRINT 606,VARID,RUNID
  606     FORMAT(' CHAIR DATA, VARIABLE',A6,' LOADED FOR RUN ',A6)
```

```
              ICHAIR=1
              GO TO 8
            END IF
          END IF
C
C         PROCESS DATA RECORD
C
          TGCHK=SUBSTR(VARID,4,2)
          IF(TGCHK.LT.ONE)THEN
            CONTINUE
          ELSE
            MTCHK=SUBSTR(VARID,3,1)
            DECODE(10,MTCHK)IMTCHK
   10       FORMAT(I1)
C
C         CHECK FOR CURRENT MOUNT
C
            IF(IMTCHK.NE.MNT)THEN
              CONTINUE
            ELSE
              IF(SUBSTR(VARID,2,1).EQ.'X')THEN
                IX=IX+1
                DO 22,J=1,250
   22           DX(J,IX)=DD(J)
                VARX(IX)=VARID
              ELSE IF(SUBSTR(VARID,2,1).EQ.'Y')THEN
                IY=IY+1
                DO 32,J=1,250
   32           DY(J,IY)=DD(J)
                VARY(IY)=VARID
              END IF
            END IF
          END IF
C
C         READ ANOTHER DATA RECORD
C
          GO TO 8
   56     CONTINUE
          IF(IY.EQ.0)THEN
            IX=0
          ELSE IF(IX.EQ.0)THEN
            CONTINUE
          ELSE
            PRINT 58,IX,IY,MNT,RUNID
   58       FORMAT(1X,I2,2X,'X DISP AND ',I2,2X,'Y DISP',
         *' LOADED FOR MOUNT ',I2,5X,'RUN ',A6)
            PRINT 66,(VARX(I),I=1,IX),(VARY(I),I=1,IY)
   66       FORMAT(12(2X,A6))
          END IF
C
C         IF NO CHAIR DATA STORE 999'S IN ARRAY
C
          IF(ICHAIR.EQ.0)THEN
            DO 71,J=1,250
   71       DC(J)=999.
          END IF
   72     CONTINUE
          OLDRUN=RUNID
          RETURN
          END
```

```
LAMBERT*TPF$(O).MIXY(30)
            SUBROUTINE MIXY
      C
      C     FUNCTION:
      C       DETERMINE MINIMUM AND MAXIMUM X AND Y VALUES FOR
      C       A PAGE OF MOUNT PLOTS
      C
            COMMON/LOAD/XMIN,XMAX,IPGE,YMIN,YMAX,TITLE,IST,ILT,IX,XOR,
           *YOR,VARX,VARY,IERR,STEP,DC(250),DX(250,12),DY(250,12)
            COMMON/PHTHDR/RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER
            CHARACTER*6 RUNID,VARID,JDTDIG,IPGVER,TITLE(9),VARX(12),VARY(12)
            XMIN,YMIN=999.
            XMAX,YMAX=O.
      C
      C     INITIALIZE COUNTERS
      C
            ISTNEW=IST
            ILTNEW=ILT
      C
      C     CHECK IF ALL PTS TO BE PLOTTED
      C
            IF(IERR.EQ.999)THEN
              ISTNEW=1
              ILTNEW=250
            END IF
      C
      C     EXAMINE DATA PTS. FOR ALL TARGETS
      C
            DO 20,JK=ISTNEW,ILTNEW
            DO 20,JL=1,IX
            DXX=ABS(DX(JK,JL))
            DYY=ABS(DY(JK,JL))
            IF(DXX.LE.990. .AND. DYY.LE.990.)THEN
              XMIN=AMIN1(DXX,XMIN)
              YMIN=AMIN1(DYY,YMIN)
              XMAX=AMAX1(DXX,XMAX)
              YMAX=AMAX1(DYY,YMAX)
            END IF
       20   CONTINUE
      C
      C     EQUATE X AND Y AXES
      C
            XDIF=XMAX-XMIN
            YDIF=YMAX-YMIN
            IF(XDIF.LE.YDIF)THEN
              XMAX=XMAX+(YDIF-XDIF)/2.
              XMIN=XMIN-(YDIF-XDIF)/2.
            ELSE
              YMAX=YMAX+(XDIF-YDIF)/2.
              YMIN=YMIN-(XDIF-YDIF)/2.
            END IF
      C
      C     COMPUTE STEP INTERVAL FOR PLOT
      C
            IF(IERR.NE.999)THEN
              STEP=(XMAX-XMIN)/.18
            ELSE
              STEP=(XMAX-XMIN)/.8
            END IF
            RETURN
```

```
   60          END
@PRT,S .LDATA
```

```
LAMBERT*TPF$(O).PLOTD(46)
      SUBROUTINE PLOTD
C
C     FUNCTION:
C       CONTROLS (1) GENERATION OF TITLE FOR EACH PAGE OF MOUNT
C       PLOTS AND (2) GENERATION OF MOUNT PLOTS WITH DISSPLA
C       PLOT PACKAGE
C
      COMMON/LOAD/XMIN,XMAX,IPGE,YMIN,YMAX,TITLE,IST,ILT,IX,XOR,
     *YOR,VARX,VARY,IERR,STEP,DC(250),DX(250,12),DY(250,12)
      COMMON/PHTHDR/RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER
      CHARACTER*6 RUNID,VARID,JDTDIG,IPGVER,TITLE(9),VARX(12),VARY(12)
     *,PAG*3
      CHARACTER*2 TGT(12),TADD,TDROP,TNEW(12),TOLD(12),SSCL*8,SC*7
      CHARACTER*4 FRAM/'    '/,TEMP*2/'  '/
      DIMENSION DAT(12,2),NPT(12)
C
C     PLOT TITLE AT TOP OF FRAME
C
C
      ENTRY PLOT1
C
C     FUNCTION:
C       GENERATE TITLE INFO. FOR PAGE OF MOUNT PLOTS
C
      CALL RESET('PHYSOR')
      CALL AREA2D(1.,.8)
      CALL HEIGHT(.018)
C
C     PRINT X-SCALE VALUE FOR THIS PAGE OF PLOTS (METERS)
C
      SCAL=XMAX-XMIN
      ENCODE(11,SC)SCAL
   11 FORMAT(F7.6)
      SUBSTR(SSCL,1,7)=SC
      SUBSTR(SSCL,7,1)='$'
      CALL MESSAG(SSCL,100,.39,.925)
C
C     PLOT TITLE
C
      ENCODE(3,PAG)IPGE
      SUBSTR(TITLE(7),3,3)=PAG
    3 FORMAT(I3)
      CALL HEIGHT(.027)
      CALL MESSAG(TITLE,44,0.,.92)
      CALL ENDGR(0)
      CALL RESET('HEIGHT')
      RETURN
C
C
      ENTRY PLOT2(IFM)
C
C     FUNCTION:
C       GENERATE MOUNT PLOT FOR A FRAME OF DATA
C     ARGUMENT DEFINITION:
C       IFM: CURRENT FRAME COUNT
C
C     PLOT DATA POINTS
C
      CALL HEIGHT(.01)
```

```
      CALL PHYSOR(XOR,YOR)
      CALL AREA2D(.18,.18)
C
C     WRITE BLOCK NO. CORRESPONDING TO DATA PT.
C
      XFM=.153
      YFM=.165
      ENCODE(7,FRAM)IFM
 7    FORMAT(I3)
      SUBSTR(FRAM,4,1)='$'
      CALL MESSAG(FRAM,100,XFM,YFM)
C
C     PLOT SYMBOL FOR CHAIR DATA (IF CHAIR DATA AVAILABLE)
C
      IF(DC(IST).GT.990)THEN
        CONTINUE
      ELSE
      XCH=.005
      YCH=.165
        CALL MESSAG('C',1,XCH,YCH)
      END IF
C
C     SCALE DATA TO PRESENT PLOTTING AREA
C
      CALL GRAF(XMIN,STEP,XMAX,YMIN,STEP,YMAX)
C
C     DRAW BLOCK AROUND SUBPLOT
C
      CALL RLVEC(XMIN,YMIN,XMIN,YMAX,0000)
      CALL RLVEC(XMAX,YMAX,XMIN,YMAX,0000)
      CALL RLVEC(XMAX,YMAX,XMAX,YMIN,0000)
      CALL RLVEC(XMIN,YMIN,XMAX,YMIN,0000)
      NTHIS=0
      DO 50,J=1,IX
      NPT(J)=0
      IF(DX(IST,J).GT.990.OR.DY(IST,J).GT.990)THEN
        CONTINUE
      ELSE
        NTHIS=NTHIS+1
C
C     LOAD DATA FOR ALL TARGETS IN THIS FRAME
C
        DAT(NTHIS,1)=ABS(DX(IST,J))
        DAT(NTHIS,2)=ABS(DY(IST,J))
C
C     IDENTIFY MANUAL ENTRIES
C
        IF(DX(IST,J).LT.0.)NPT(NTHIS)=1
        TGT(NTHIS)=SUBSTR(VARX(J),4,2)
        IF(SUBSTR(VARX(J),4,2).NE.SUBSTR(VARY(J),4,2))THEN
          PRINT 52
 52       FORMAT(' X + Y DISP. TARGETS DO NOT AGREE ')
          STOP 'PLOTD'
        END IF
      END IF
 50   CONTINUE
      IF(NTHIS.NE.0)THEN
        DO 60,J=1,NTHIS
        X=XPOSN(DAT(J,1),DAT(J,2))
        Y=YPOSN(DAT(J,1),DAT(J,2))
```

```
              IF(TGT(J).LE.'09')THEN
                XT=X
                YT=Y
                XM=X-.004
                YM=Y-.009
                SUBSTR(TEMP,1,1)=SUBSTR(TGT(J),2,1)
                IT=1
              ELSE
                XT=X-.006
                YT=Y
                XM=X-.007
                YM=Y-.009
                TEMP=TGT(J)
                IT=2
              END IF
C
C     PLOT TARGET
C
              CALL MESSAG(TEMP,IT,XT,YT)
C
C     PLOT CIRCLE AROUND TARGET IF MANUAL ENTRY
C
              IF(NPT(J).NE.0)THEN
                CALL HEIGHT(.029)
                CALL INTNO(0,XM,YM)
                CALL HEIGHT(.01)
              END IF
   60         CONTINUE
              INS=NTHIS-1
C
C     CONNECT TARGETS WITH LINES
C
              DO 68,J=1,INS
              J1=J+1
              DO 68,J2=J1,NTHIS
              CALL RLVEC(DAT(J,1),DAT(J,2),DAT(J2,1),DAT(J2,2),0000)
   68         CONTINUE
            END IF
            TADD='  '
            TDROP='  '
C
C     DETERMINE WHICH TARGETS HAVE BEEN ADDED OR DELETED FROM LAST FRAME
C
            IF(IST.NE.1)THEN
              IF(NTHIS.NE.0.OR.NLAST.NE.0)THEN
                IF(NTHIS.NE.0)THEN
                  DO 112,J=1,NTHIS
  112             TNEW(J)=TGT(J)
                  IF(NLAST.NE.0)THEN
                    DO 113,J=1,NLAST
                    DO 113,K=1,NTHIS
                    IBOTH=0
                    IF(TOLD(J).EQ.TNEW(K))IBOTH=1
                    IF(IBOTH.EQ.1)THEN
                      TOLD(J)='  '
                      TNEW(K)='  '
                    END IF
  113               CONTINUE
                    DO 114,J=1,NLAST
  114               IF(TOLD(J).NE.'  ')TDROP=TOLD(J)
```

```
                  END IF
                  DO 115,K=1,NTHIS
  115             IF(TNEW(K).NE.'  ')TADD=TNEW(K)
                ELSE IF(NTHIS.NE.0)THEN
                  IF(NLAST.EQ.0)THEN
                    DO 214,J=1,NLAST
  214               IF(TOLD(J).NE.'  ')TDROP=TOLD(J)
                  END IF
                END IF
              ELSE IF(NTHIS.EQ.0 .AND. NLAST.EQ.0)THEN
                GO TO 111
              END IF
            ELSE IF(IST .EQ. 1)THEN
              IF(NTHIS.EQ.0)GO TO 111
            END IF
C
C           STORE TARGET NOS. FOR THIS FRAME
C
            DO 116,J=1,NTHIS
  116       TOLD(J)=TGT(J)
            NLAST=NTHIS
C
C           PLOT SYMBOL IF NEW TARGET WAS VISIBLE IN THIS FRAME
C
            XNEW=.005
            YNEW=.007
            IF(TADD.LE.'09')THEN
              SUBSTR(TEMP,1,1)=SUBSTR(TADD,2,1)
              IA=1
            ELSE
              TEMP=TADD
              IA=2
            END IF
            CALL MESSAG(TEMP,IA,XNEW,YNEW)
C
C           PLOT SYMBOL IF A TARGET WAS DROPPED IN THIS FRAME
C
            XDROP=.159
            YDROP=.007
            IF(SUBSTR(TDROP,1,1).EQ.'0')SUBSTR(TDROP,1,1)=' '
            CALL MESSAG(TDROP,2,XDROP,YDROP)
  111       CONTINUE
C
C           END SUBPLOT
C
            CALL RESET('HEIGHT')
            CALL ENDGR(0)
            RETURN
            END

@PRT,S .MIXY
```

```
LAMBERT*IPF$(O).XYPLOT(22)
      SUBROUTINE XYPLOT
C
C     FUNCTION:
C        PLOT X AND Y DISPLACEMENT VALUES FOR GIVEN RUN AND ANATOMICAL
C        MOUNT; EACH PHOTO TARGET PLOTTED ON SAME GRAPH
C
      DIMENSION DXARY(250),DYARY(250)
      COMMON/LOAD/XMIN,XMAX,IPGE,YMIN,YMAX,TITLE,IST,ILT,IX,XOR,
     *YOR,VARX,VARY,IERR,STEP,DC(250),DX(250,12),DY(250,12)
      COMMON/PHTHDR/RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER
      CHARACTER*6 RUNID,VARID,JDTDIG,IPGVER,TITLE(9),VARX(12),VARY(12)
      CHARACTER*6 XYTIT(9),TAR*2,TIME*8,DATE*8
      DATA NPTS,ITAR/2*0/
C
C     LOAD TITLE INFO.
C
      DO 12,I=1,9
      XYTIT(I)=TITLE(I)
   12 CONTINUE
      XYTIT(5)=' '
      XYTIT(6)=' '
      XYTIT(7)=' '
      XYTIT(8)='      '
      XYTIT(9)='      '
      CALL ADATE(DATE,TIME)
      XYTIT(4)=TITLE(5)
      SUBSTR(XYTIT(5),4,2)=SUBSTR(DATE,1,2)
      SUBSTR(XYTIT(6),1,2)=SUBSTR(DATE,3,2)
      SUBSTR(XYTIT(6),4,2)=SUBSTR(DATE,5,2)
C
C     SUPPRESS LISTING FOR OUT OF RANGE POINTS
C
      CALL NOCHEK
      CALL RESET('PHYSOR')
C
C     LABEL AXES
C
      CALL AREA2D(.8,.8)
      CALL XNAME(' FILM PLANE X DISPLACEMENT (METERS)',35)
      CALL YNAME(' FILM PLANE Y DISPLACEMENT (METERS)',35)
C
C     PRINT HEADING
C
      CALL HEIGHT(.027)
      CALL MESSAG(XYTIT,50,0...92)
      CALL HEIGHT(.016)
C
C     COMPUTE MIN AND MAX VALUES FOR X AND Y
C
      IERR=999
      CALL MIXY
      IERR=0
C
C     SCALE DATA FOR PLOTS
C
      CALL GRAF(XMIN,STEP,XMAX,YMIN,STEP,YMAX)
C
C     PLOT X-Y CONTOUR FOR EACH TARGET
C
```

```
      CALL HEIGHT(.022)
      DO 10,I=1,IX
      LINE=MOD(I,5)
      IF(LINE.EQ.O)THEN
        CALL DOT
      ELSE IF(LINE.EQ.1)THEN
        CALL RESET('DOT')
      ELSE IF(LINE.EQ.2)THEN
        CALL CHNDOT
      ELSE IF(LINE.EQ.3)THEN
        CALL DASH
      ELSE IF(LINE.EQ.4)THEN
        CALL CHNDSH
      END IF
C
C     LOAD DATA INTO PLOT ARRAY AND CHANGE NEGATIVE SIGNS
C
      DO 30,J=1,250
      DXARY(J)=DX(J,I)
      DYARY(J)=DY(J,I)
      IF(DXARY(J).LT.O.)THEN
        DXARY(J)=-1.*DXARY(J)
      END IF
      IF(DYARY(J).LT.O.)THEN
        DYARY(J)=-1.*DYARY(J)
      END IF
   30 CONTINUE
C
C     PLOT DATA (ONLY NON-ZERO PTS)
C
      IPOS=1
      ITAR=O
    5 DO 35,J=IPOS,250
      IF(DXARY(J).GT.990..OR.DYARY(J).GT.990.)THEN
        CONTINUE
      ELSE
        NPTS=O
        IPLOT=J
        GO TO 15
      END IF
   35 CONTINUE
      GO TO 10
   15 DO 40,K=IPLOT,250
      IF(DXARY(K).GT.990..OR.DYARY(K).GT.990.)THEN
        IF(ITAR.EQ.O)THEN
          SX=XPOSN(DXARY(IPLOT),DYARY(IPLOT))
          SY=YPOSN(DXARY(IPLOT),DYARY(IPLOT))
          TAR=SUBSTR(VARX(I),4,2)
          IF(SUBSTR(TAR,1,1).EQ.'O')SUBSTR(TAR,1,1)=' '
          CALL MESSAG(TAR,2,SX,SY)
          ITAR=1
        END IF
        CALL CURVE(DXARY(IPLOT),DYARY(IPLOT),NPTS,O)
        IPOS=K+1
        IF(K.LT.250)GO TO 5
      ELSE
        NPTS=NPTS+1
        IF(K.EQ.250)THEN
          IF(ITAR.EQ.O)THEN
            SX=XPOSN(DXARY(IPLOT),DYARY(IPLOT))
```

```
                  SY=YPOSN(DXARY(IPLOT),DYARY(IPLOT))
                  TAR=SUBSTR(VARX(I),4,2)
                  IF(SUBSTR(TAR,1,1).EQ.'0')SUBSTR(TAR,1,1)=' '
                  CALL MESSAG(TAR,2,SX,SY)
                  ITAR=1
                END IF
              CALL CURVE(DXARY(IPLOT),DYARY(IPLOT),NPTS,0)
              END IF
            END IF
   40       CONTINUE
   10       CONTINUE
            CALL RESET('HEIGHT')
            CALL RESET('XNAME')
            CALL RESET('YNAME')
            CALL RESET('DOT')
            CALL ENDPL(0)
            RETURN
            END
```

@PRT,S .TIMER

```
LAMBERT*TPF$(O).XQTSGN(3)
@XQT MPO051/SIGN
@.
@.  CONTROL STATEMENT IMMEDIATELY AFTER @XQT STATEMENT
@.       POS 5 - NO. INPUT UNITS (1, 2 OR 3)
@.       POS 10 - LOWEST RUN NO. TO BE PROCESSED
@.       POS 20 - HIGHEST RUN NO. TO BE PROCESSED
@TEST TNE/O/S6
@JUMP OUT
@ASG,TF PCSXXXXXXXX.,U9S,SAVEO5 . PCSXXXXXXXX
@COPY,GM 9.,PCSXXXXXXXX.
@COPY,GM 9.,PCSXXXXXXXX.
@COPY,GM 9.,PCSXXXXXXXX.
@TEST2:
@TEST TG/1/S6
@JUMP TEST3
@COPY,GM 10.,PCSXXXXXXXX.
@COPY,GM 10.,PCSXXXXXXXX.
@COPY,GM 10.,PCSXXXXXXXX.
@TEST3:
@TEST TG/2/S6
@JUMP OUT
@COPY,GM 11.,PCSXXXXXXXX.
@COPY,GM 11.,PCSXXXXXXXX.
@COPY,GM 11.,PCSXXXXXXXX.
@OUT:
@FREE,I PCSXXXXXXXX.
@. EXITED FROM .XQTSGN

@PRT,S .SGFLIP
```

```
LAMBERT*TPF$(O).SGFLIP(61)
C
C     FUNCTION:
C       THIS ASCII FORTRAN DRIVER PROGRAM PERFORMS
C       (1) SIGN CHANGE OF NEGATIVE DATA PTS. (MANUAL ENTRIES)
C       TO POSITIVE AND (2) WRITES REFORMATTED DATA RECORD TO MASS STORAGE
C
      DIMENSION DC(250)
      COMMON/PHTHDR/RUNID,VARID,ISITID,F1TIME,JDTDIG,DELTA,IPGVER
      CHARACTER*6 RUNID,VARID,JDTDIG,IPGVER,FRUN,LRUN,RUNS(200)
     *,FIRST,LAST,ENC*4,SITE*1
C
C     READ USER PARAMETERS IN CARD IMAGE
C
      READ(5,6)IFILE,FRUN,LRUN
    6 FORMAT(I5,9X,2(A6,4X))
      DATA MAX/320/
C
C     COMPUTE RUN RANGE
C
      FIRST=SUBSTR(FRUN,3,4)
      LAST=SUBSTR(LRUN,3,4)
      DECODE(4,FIRST)IFIRST
      DECODE(4,LAST)ILAST
    4 FORMAT(I4)
C
C     COMPUTE INPUT UNIT ASSIGNMENTS
C
      ILIM=IFILE+8
      IU=9
    1 CONTINUE
      PRINT 3,IU,ILIM,FRUN,LRUN
    3 FORMAT(2I10,2(4X,A6))
      NIU=-1*IU
C
C     BEGIN RUN LOOP
C
      IRN=IABS(ILAST-IFIRST)+1
      INC=IFIRST
      DO 100,I=1,IRN
      SUBSTR(RUNS(I),1,2)='LX'
      ENCODE(17,ENC)INC
   17 FORMAT(I4)
      SUBSTR(RUNS(I),3,4)=ENC
      INC=INC+1
C
C     ATTEMPT TO SET SECTOR TO HEADER FOR RUN
C
      CALL SETSCT(IU,RUNS(I),'PHDATA',1,ISITE,NREC,IER)
      IF(IER.NE.0)GO TO 100
      CALL GETRCD(IU,MAX,RUNID,VARID,ISITE,IDL,DC,IER)
      IF(IER.NE.0)THEN
        PRINT 13,IER
   13   FORMAT(1X,'STATUS=',O3)
        STOP 'HEADER ERROR'
      END IF
C
C     SET SECTOR FOR TIME RECORD
C
      VARID=' TS   '
```

```
      ENCODE(11,SITE)ISITID
   11 FORMAT(I1)
      SUBSTR(VARID,1,1)=SITE
      CALL SETSCT(IU,RUNID,VARID,1,ISIT,NREC,IER)
      IF(IER.NE.O)GO TO 110
C
C     READ NEXT RECORD
C
      DO 10,K=1,100
      CALL SECTOR(IU,ISEC)
      CALL GETRCD(IU,MAX,RUNID,VARID,ISIT,IDL,DC,IER)
      IF(IER.NE.O)THEN
        IF(IER.EQ.1)THEN
          GO TO 100
        ELSE
          GO TO 110
        END IF
      END IF
      INEG=O
C
C     CHANGE SIGNS OF NEGATIVE NOS.
C
      DO 8,M=1,250
      IF(DC(M).LT.O)THEN
        INEG=1
        DC(M)=-DC(M)
      END IF
    8 CONTINUE
C
C     WRITE RECORD BACK TO MASS STORAGE
C
      IF(INEG.NE.O)THEN
        CALL SETADR(IU,ISEC)
        CALL PUTRCD(NIU,RUNID,VARID,ISIT,IDL,DC,IER)
        IF(IER.NE.O)GO TO 110
      END IF
   10 CONTINUE
C
C     ERROR ROUTINE
C
  110 PRINT 111,RUNID,IER
  111 FORMAT(' ERROR IN SIGN FLIP ROUTINE FOR RUN ',A6/' STATUS=',O3)
  100 CONTINUE
      IF(IU.LT.ILIM)THEN
        IU=IU+1
        GO TO 1
      END IF
      END
```

SUBROUTINE SETSCT

a. Function: Subroutine SETSCT is an ASCII FORTRAN program designed to control the positioning of sector address for either a photo header record or a photo time record. The calling program must provide the logical unit number of the photo data file and the ASCII variable RUNID; these arguments are used to locate the desired run in the random access directory. In addition, the user provides the variable name (see Calling Sequence) and the desired occurrence of the time record as arguments. There may be more than one PDS (Photo Digitizing System) physical record/target in a run, hence it is necessary to designate which one is desired (even if there is only one PDS physical record/target).*

Entry DATSCT of subroutine DIRBFR is called in order to retrieve the sector address of the desired record. This address is used in a subsequent call to Entry SETADR of subroutine FORTIO in order to correctly position the data file.

This subroutine returns the site ID of the data file as argument ISITE. Additionally, the number of PDS physical records required per target for the desired run is returned in argument NREC.

b. Subroutines (see reference 1):

DIRBFR/Entry DATSCT, FORTIO/Entry SETADR

c. Calling Sequence:

CALL SETSCT (IUNIT, RUNID, VARID, NPHYS, ISITE, NREC, IER)

* In the event more than 250 frames of photo data were digitized for a target, an extra PDS physical record is required for each additional 250 frames.

SUBROUTINE SETSCT (Continued)

where:

ARGUMENT	DESCRIPTION	TYPE
IUNIT	Logical unit number of photo directory and data file	Integer
RUNID	Run number for requested sector positioning	6 char
VARID*	Variable name for requested sector positioning	6 char
NPHYS	Desired occurrence of time record (Nphys=1 unless more than 250 frames were digitized)	Integer
ISITE	Returned site ID of data file	Integer
NREC	Returned number PDS physical records required for each photo target	Integer
IER	Returned error status code (See Appendix 4e for code definitions)	Octal

* This argument identifies whether the sector is to be set for a header record (VARID='PHDATA') or a time record (VARID='1TS   ' for site 1).

SUBROUTINE GETRCD

a. Function: This subroutine is an ASCII FORTRAN program designed to control the execution of sequential binary reads of FORTRAN V photo (1) header records, (2) time records and (3) displacement records. GETRCD utilizes Entry READIO of subroutine FORTIO to retrieve the data record from mass storage (the data record is stored in named common IOBUFF). It then extracts the run number (RUNID) and variable name (VARID) from the data record and converts each from FIELDATA to ASCII with system function subprogram FD2ASC; these converted character variables are returned to the calling program. The site ID (ISITE) of the data record is also returned as an argument.

If the retrieved record is either a time record or a displacement record, the word length of the data array (IDL) and the data array itself (DATARY) are returned to the calling program. If the retrieved record is a header these arguments have no meaning. However, when a header record is read GETRCD calls subroutine DECHED in order to (1) convert all alphanumeric data in the header to ASCII and (2) store all header information in named common PHTHDR.

The user specifies MAXLEN as the maximum word length of the data record being read from mass storage. GETRCD utilizes function subprogram EXPAND to insure that the user has specified a maximum length large enough to accommodate those sentinels and checksums which will be read from mass storage as part of the input record. If MAXLEN is too small, an appropriate error status code is returned.

b. Subroutines (see reference 1):

FORTIO/Entry READIO, FD2ASC, ASC2FD, DECHED, EXPAND

c. Calling Sequence:

Call GETRCD (IUNIT, MAXLEN, RUNID, VARID, ISITE, IDL, DATARY, IER)

SUBROUTINE GETRCD (Continued)

where:

ARGUMENT	DESCRIPTION	TYPE
IUNIT	Logical unit number of photo data and directory file	Integer
MAXLEN	Maximum word length of input record	Integer
RUNID	Returned run number of input record	6 char
VARID*	Returned variable name of input record	6 char
ISITE	Returned site ID	Integer
IDL	Returned word length of data array	Integer
DATARY	Returned data array	Real
IER	Returned error status code (see Appendix 4e for code definitions)	Octal

d. Common Block Usage:

COMMON/IOBUFF/IOBUFF(4100)

where:

VARIABLE	DESCRIPTION	TYPE
IOBUFF(4100)	Input array containing FORTRAN (FIELDATA) photo data record read from mass storage (see Figures 2c-2e for data record formats	Integer

The contents of this array are passed as an argument to the header decoding subroutine (DECHED). Alternatively, if the photo input record is either a time record or a displacement record, the data array is extracted from IOBUFF for return to the calling program.

* VARID='PHDATA' for a photo header record
VARID='1TS   ', '2TS   ' or '3TS   ' for a photo time record (first character represents site ID)
See Figure 4 for possible values of VARID for a photo displacement record

SUBROUTINE PUTRCD

a. Function: Subroutine PUTRCD is an ASCII FORTRAN program designed to control the execution of sequential binary writes of FORTRAN (FIELDATA) photo (1) header records, (2) time records, and (3) displacement records. This routine provides the user with complete directory maintenance. The user may utilize PUTRCD to (1) create a photo data output file with accompanying directory or (2) perform sequential output of photo records without directory maintenance. If the user desires to perform sequential writes without building a corresponding directory, (s)he must transmit the negative of the logical unit number (of the photo data file) as the value of the argument IUNIT in the call statement.

In the event a photo header record is to be written, it is the user's responsibility to provide the ASCII version of the header in named common PHTHDR (see Figure 2c for format). However, if the user utilizes input subroutine GETRCD to read an existing header record, its ASCII version is automatically stored in common. This ASCII version of the header must be converted to FIELDATA; this is accomplished by a call to subroutine ENCHED.

The user must provide PURTCD with the (1) logical unit number of the photo data output file (IUNIT), (2) run number of the output record (RUNID), (3) variable name of the output record (VARID*), (4) site ID of the output record (ISITE), (5) word length of the output data array (IDL) and (6) output data array (DATARY). Values for items (5) and (6) must be assigned for all output records except header records. Items (1) through (4) are passed by *PUTRCD* to subroutine DIRCTY where the necessary directory entries are created.

Once the directory has been updated, PUTRCD uses the word length of the output array (IDL) and the output array itself (DATARY) to construct the output record. Entry WRITIO of subroutine FORTIO is then called to perform the sequential write. The output record is passed down in an argument array (IOBUFF), and is also stored in named common IOBUFF.

b. Subroutines (see reference 1):

ASC2FD, ENCHED, FORTIO/Entry WRITIO

c. Calling Sequence:

Call PUTRCD (IUNIT, RUNID, VARID, ISITE, IDL, DATARY, IER)

* VARID='PHDATA' for a photo header record
VARID='1TS ', '2TS ' or '3TS ' for a photo time record (first character represents site ID)
See Figure 4 for possible values of VARID for a photo displacement record

SUBROUTINE PUTRCD (Continued)

where:

ARGUMENT	DESCRIPTION	TYPE
IUNIT	Logical unit number of photo data and directory file	Integer
RUNID	Run number of output data record	6 char
VARID	Variable name of output data record	6 char
ISITE	Site ID of output record	Integer
IDL*	Word length of output data array	Integer
DATARY*	Output data array	Real
IER	Returned error status code (see Appendix 4e for code definitions)	Octal

d. Common Block Usage:

(1) COMMON/PHTHDR/RUNID, VARID, ISITID, F1TIME, JDTDIG, DELTA, IPGVER

(See Figure 2c for format of named common PHTHDR)

(2) COMMON/IOBUFF/IOBUFF(4100)

where:

VARIABLE	DESCRIPTION	TYPE
IOBUFF(4100)	Output array containing FORTRAN (FIELDATA) photo (1) header record, (2) time record or (3) displacement record to be written to core (see Figures 2c-2e for data record formats)	Integer

* This argument has no meaning for header record.

SUBROUTINE EOFPUT

a. Function: This ASCII FORTRAN program is designed to control the writing of an "end-of-file" to the end of a data file. Initially EOFPUT calls subroutine DIRCTY in order to have the output file positioned for the write. Once this is done, the "end-of-file" is written by virtue of a call to entry WRITIO of the subroutine FORTIO. Control is then returned to the calling program. In the event the user wishes to write an "end-of-file" without directory maintenance, (s)he must pass down the negative of the logical unit number (of the photo data file) as the value of the argument IUNIT.

b. Subroutines (see reference 1):

DIRCTY, FORTIO/Entry WRITIO

c. Calling Sequence:

Call EOFPUT (IUNIT, IER)

where:

ARGUMENT	DESCRIPTION	TYPE
IUNIT	Logical unit number of photo data and directory file	Integer
IER	Returned error status code (see Appendix 4e for code definitions)	Octal

ERROR CODE DEFINITIONS

IER (Octal) = error status code of the I/O function

IER (Octal)	Explanation
0	Normal I/O completion
1	"End-of-file" has been identified
5	Attempted to read from unassigned area of mass storage
20	Write attempted on read only file
21	Reference made to unassigned file
22	Write attempted beyond assigned area of mass storage
25	Maximum length of input/output array exceeded
26	FORTRAN sentinels are unequal - read was not executed
30	Run number not found in run directory
31	Variable name not found in variable directory
32	Checksum error
33	Run number not last entry in run directory
34	Update in place aborted due to different data record lengths
35	Invalid FORTRAN sentinel - word length indicated for physical record greater than 249
36	Invalid FORTRAN sentinel - record number indicated for first physical record not equal to 1

END

FILMED

9-84

DTIC