PHOTOGRAPHIC AND PHOTOMETRIC ENHANCEMENT OF LUNAR ORBITER PRODUCTS

PROJECT A

Joining a Series of Framelets

Prepared under contract NAS 9-10477

For MAPPING SCIENCES BRANCH

NASA Manned Spacecraft Center

By Applied Scientific Research, Inc.

PHOTOGRAPHIC AND PHOTOMETRIC ENHANCEMENT OF LUNAR ORBITER PRODUCTS

PROJECT A

Joining a Series of Framelets

PROJECT B

Improvement of Photometric Data

PROJECT C

Statistical Error Analysis of Photometric Data

Prepared under contract NAS 9-10477

For MAPPING SCIENCES BRANCH

NASA Manned Spacecraft Center

September, 1972

By Applied Scientific Research, Inc.

TABLE OF CONTENTS

PROJECT A - JOINING A SERIES OF FRAMELETS

Table of Contents-Cont'

PART III GENERALIZATIONS

LIST OF FIGURES

PART I

List of Figures-Cont'

List of Figures-Cont'

LIST OF TABLES

PART I

PART III

PART III

Part I JOINING A SERIES OF LUNAR ORBITER FRAMELETS

Review of the source of the data

Photographs of the Lunar surface are processed on board the Lunar Orbiter (LO) spacecraft. Subsequently, the photographs are read out using a flying spot scanner which is moved up and down mechanically. About 17,500 video lines are produced for each framelet, which is a strip of 70mm film about one-tenth inch wide. This video signal is transmitted and is then recorded on video tape to be later digitized. The digital tapes, containing an array as large as about 8,500 by 636 6 bit bytes, are edited and rewritten so that each tape will contain some of the 70mm film edge data and a header to make identification easy. The basic concern of this investigation is the joining together, by digital computer, of adjacent framelets. In order to understand the sometimes mysterious gyrations the actual solution takes, we need to study the film handling system, the readout system and the digitization system in much more detail.

1. The film handling system

The film handling system transports and stores the photographic film during some of the photo subsystem operational modes. The system transports the film in two directions, forward and reverse. During the photographic, processing and wind-up modes, the film moves toward the take-up reel (in the forward direction). We are particularly interested in the operation during final read-out. The component which performs this function is called the read-out drive. The following information is taken in part from the document "Photographic Subsystem Reference Handbook for the Lunar Orbiter Program" (#L-018375-RU) prepared by Eastman Kodack Company, Rochester, New York 14650, 15 March 1966.

During both quick-look and final read-out modes, the processed film is moved continuously at a measured rate in the reverse direction by means of a geared-metering roller. Three cam-driven operations take place: (1) a high precision cylindrical cam moves the scanning lens across the film (2) a simple cam operates the film clamp so that focus is obtained (with respect to the scanner lens), and (3) a simple cam step advances film in the read-out gate at the end of each scan. The motor is turned ON when suitable motor signals are supplied from the control to the read-out control electronics power-switch circuitry. During a scan of a framelet, the metering roller slowly draws film out of a loop on the read-out looper side of the read-out gate and continuously feeds the read-out looper. See Fig. 1.1. The roller in this looper is connected by a lever to a similar roller and loop on the take-up side of the read-out gate. As film is drawn out of the output loop the lever transmits the force and the input loop is enlarged and draws film at the same steady rate out of the film take-up assembly. Between these loops the film clamp holds the film section being scanned stationary in the read-out gate. When the end of the scan (or scanning lens turn-around) is reached, the cam operated gate opens and the lever between the loops is cam driven to move film in step fashion through the gate. The film clamp recloses and scan resumes back across the film. The cams are all two cycle to accomplish back and forth scan operation. The metering roller supplies an accurate 0.100 inch of film in each framelet. The sequence continues moving the scanning lens back and forth across the film and advancing the film through the read-out system until the command "read-out drive OFF" or other inhibiting logic occurs.

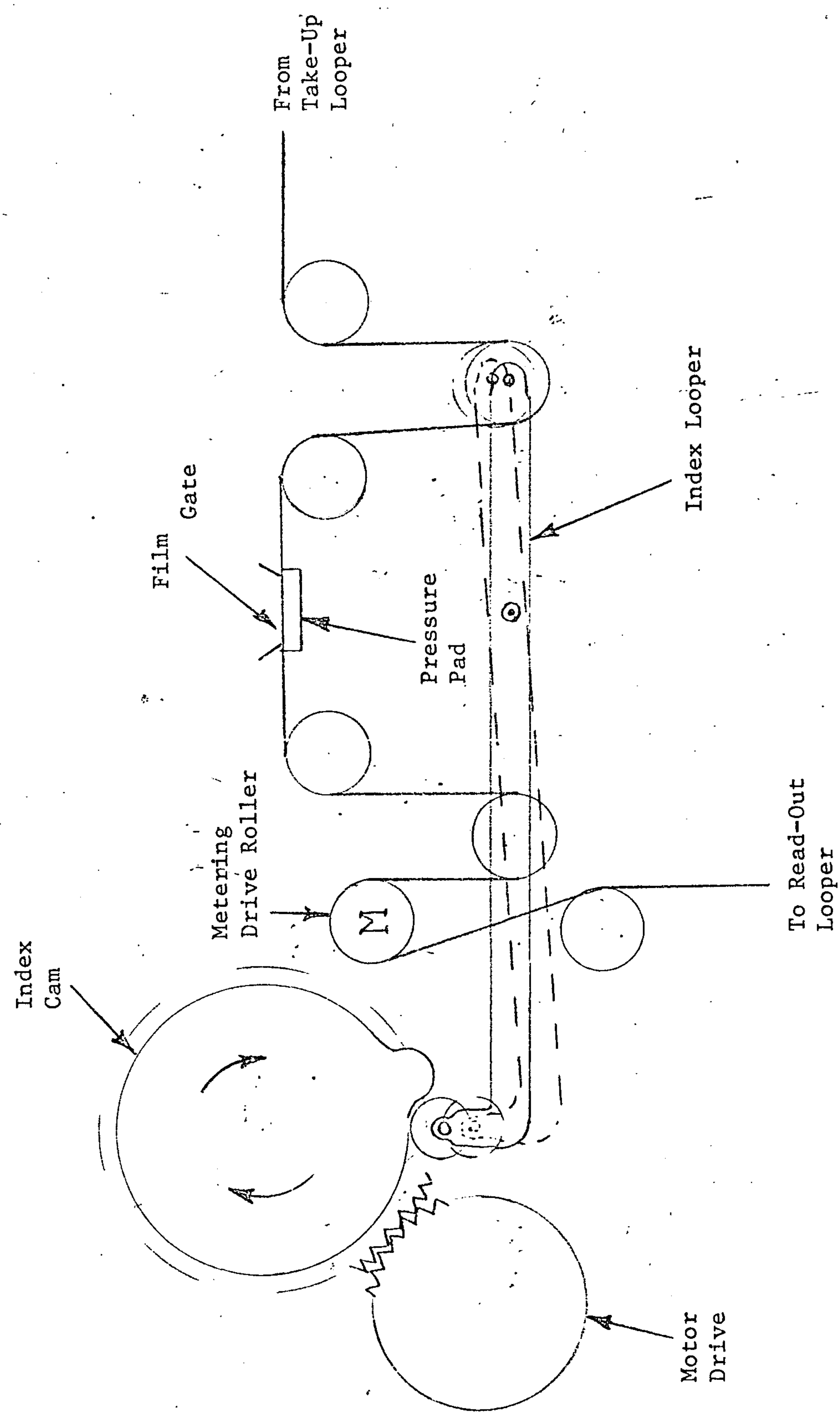


Fig. 1.1 Film Advance Mechanism

Attached to the cam is a two segment encoder switch whose logic indicates two positions of the optical-mechanical scanner. The spot-stop condition indicates that the scanner is positioned at its farthest excursion on the pre-exposed, calibration side of the film where the film clamp is released. When the read-out is commanded OFF, the drive mechanism will continue until it arrives at the spot-stop position. Then the spot-stop condition will turn the drive motor OFF. During modes where film travels in the forward direction through the read-out gate, the read-out drive-motor clutch is disengaged by a certain control signal. With the clutch disengaged and the clamp released, film free-wheels through the scanner.

It should be apparent that there will be little tendancy for the film to rotate during transport. On the other hand, the mechanism which causes the film to advance surely seems to be susceptible to random errors. These errors could effect the output data by making the overlap between adjacent framelets less (or more) than expected. The most obvious sources of the observed errors lie in the metering drive roller and the index looper (See Fig. 1.1)

1.2 If the metering drive allows too much film thru (ie more than .100"), then the overlap between adjacent framelets will be less than expected.

1.3 The actual advance of the film is done by the index looper and index cam. (All the metering drive roller does is furnish a constant average). Since this drive is in effect a friction operated drive, one may expect random errors, causing variable overlap between adjacent framelets. (All experimental investigations reported on here have involved data with between one and two percent overlap, with this amount being highly variable, thus reinforcing the suspicion that the advance mechanism is subject to errors.)

2. The readout mechanism

After the film is positional by the transport mechanism, the film is read out by a flying spot scanner which is moved up and down mechanically. The mechanical movement apparently involves a mirror and a cam which are not syncronized to anything else on the LO system. The flying spot scanner operates by a magnetically focused and deflected electron beam hitting a rotating short persistance phosphor drum. The light from the drum is focused and reflected through the film. A photo multiplier-video subsystem produces an output video signal which is transmitted to Earth. Obvious sources of concern for us here are the characteristics of the rotating drum , the linearity of the electrical scan, the uniformity of the focus, the linearity of the mechanical scan, and the repeatability of any of the possible defects in these properties.

2.3 Rotating drum

Early in the LO program, the rotating drum was recognized as being a source of distortion in LO photographic data since the pattern of the drum will clearly be imprinted on each readout. This pattern is called the "drum signature", and results from nonuniformities of the phosphor coating on the drum. Part of the drum signature is intentionally put on the drum before it is sealed in the line scan tube. Scratches which project about seven microns wide by fifty-six microns high (almost half way around the drum) are placed on the drum near the ends. These marks will be called "drummarks" here to avoid confusion with other callibration marks. The main trouble with the drummarks is that they live in the area of overlap; later, we will display a more exact evaluation of where the drummarks are. For our purposes here, however, it turns out that the drummarks are an almost fatal hinderence, since they combine to leave almost no area of overlap in which matching of adjacent framelets can be accomplished. The problem of the drum signature (leaving aside the drummarks) is also severe, although the signature is undetectable with the eye observing GRE (Ground Reconstruction Equipment) prints of LO imagry. We found that sensitive detection algorithms (namely cross-correlation) picked up the drum signature over the very weak "signal", probably because our algorithm was necessarily strongly biased to be sensitive in the direction perpendicular to the direction of scan (which is exactly the worst possible direction for the drum signature).

2.4 Linearity of electrical scan

Measurements of GRE 35mm film showed that the electrical scan linearity is within specifications for LO data. These measurements were made on the linearity test bars at one edge of each framelet. Nonlinearities of the electrical scan are completely masked by jitter in the digitizer and will not be considered further.

2.5 Uniformity of focus of the scan spot

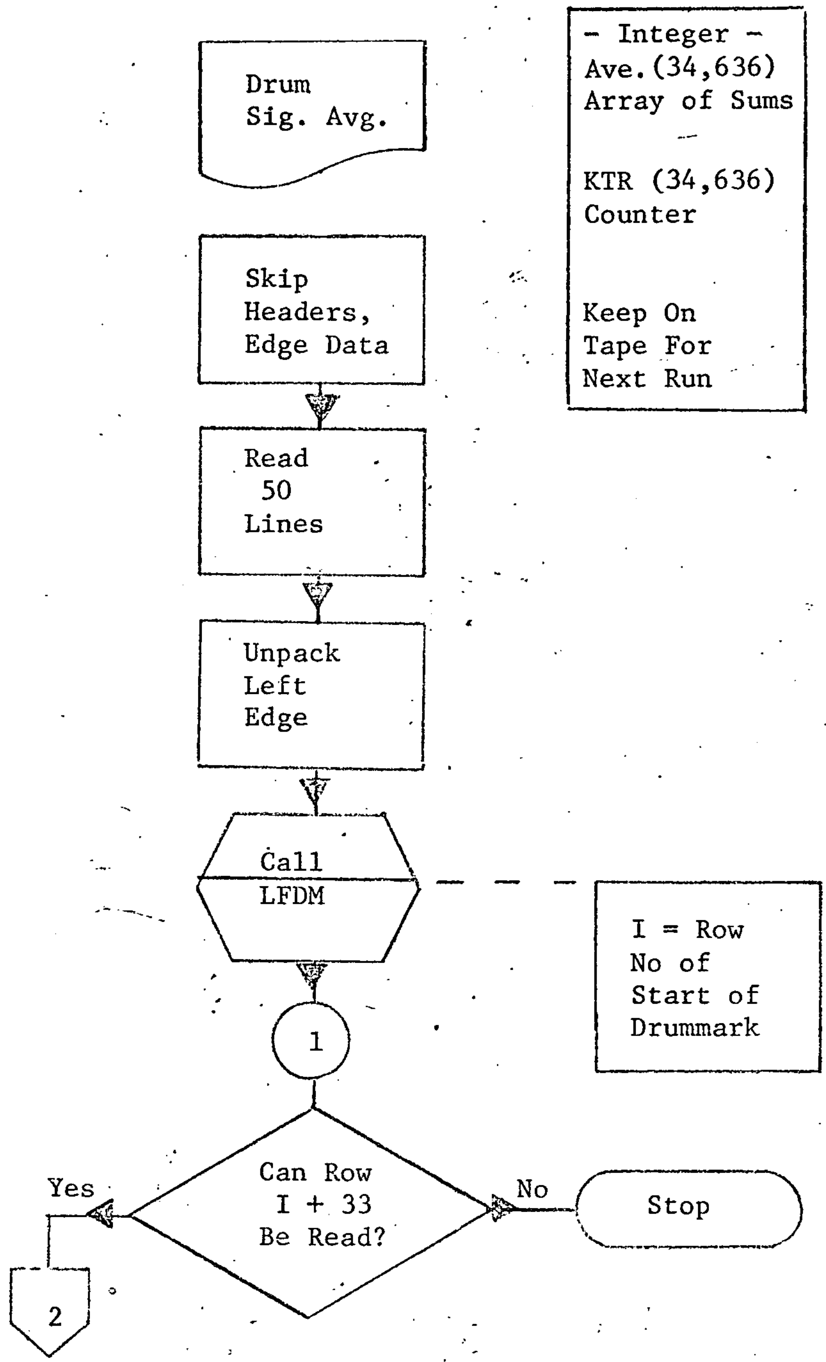
The spot is not uniformly focused onto the film. The most obvious place to blame this nonuniformity on is the focus of the electron beam onto the drum. The characteristic W-shaped variation in intensity of the spot optical focusing onto the film, called the line scan signature, is immediately obvious looking at the early GRE film (before GRE correction was incorporated). In the LO Analysis Program, this effect was noted and corrected for by means of a simple algorithm which, by repeated averaging, obtained corrections and then smoothed the corrections to obtain correction factors which were applied multiplicatively. This is conceptually the correct response to the line scan signature problem. However, the proceedure acceptable for the analysis program was inadequate for our purposes since the smoothing proceedure used there is very poor on the edges. This has the effect of not completely correcting for the line scan signature and leading to unpredictable results in the matching process. Improved correction factors were accordingly developed and applied.

2.6 Linearity of the mechanical scan

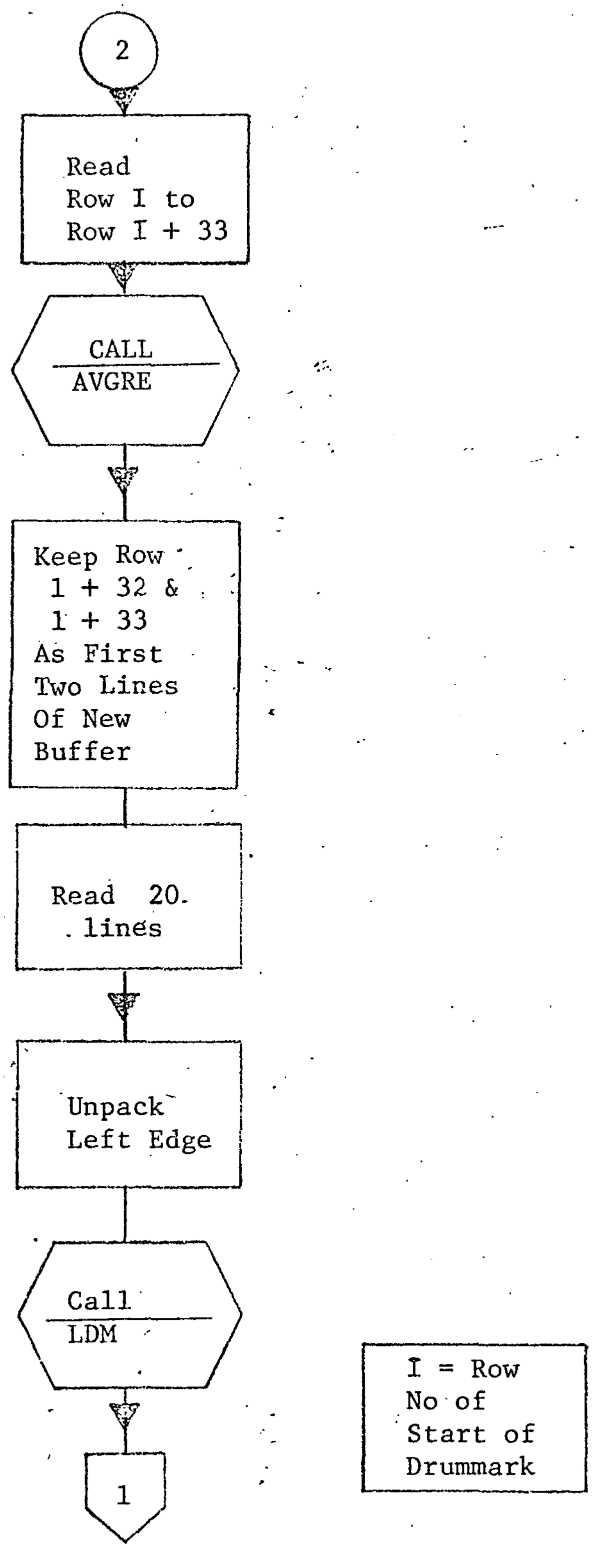
Most suspect of all the possible defects in the LO readout subsystem is the device which moves the scan line up and down. Examination of the GRE film reveals large (five percent) variation in the mechanical scan direction, leading to such unusual moon features as split craters or staggared rills, even when the reconstruction of the 35mm strips is done by hand. Our experimental observation of the digitized information leads to the same conclusion, that is, that a variation of five scan lines (about 18 μ) over a step of one hundred scan lines (about 350 μ) is not unusual. While these numbers seem small, 18μ represents a displacement of nearly two meters (at a nominal height of 46 km). Since the nominal resolution is one meter, it is easy to imagine what this nonlinearity does to stereo-photogrammetric work. The problem of matching adjacent framelets is also gravely effected by this nonlinearity, since it strongly restricts the size of the detection algorithm.

2.7 Possible corrections

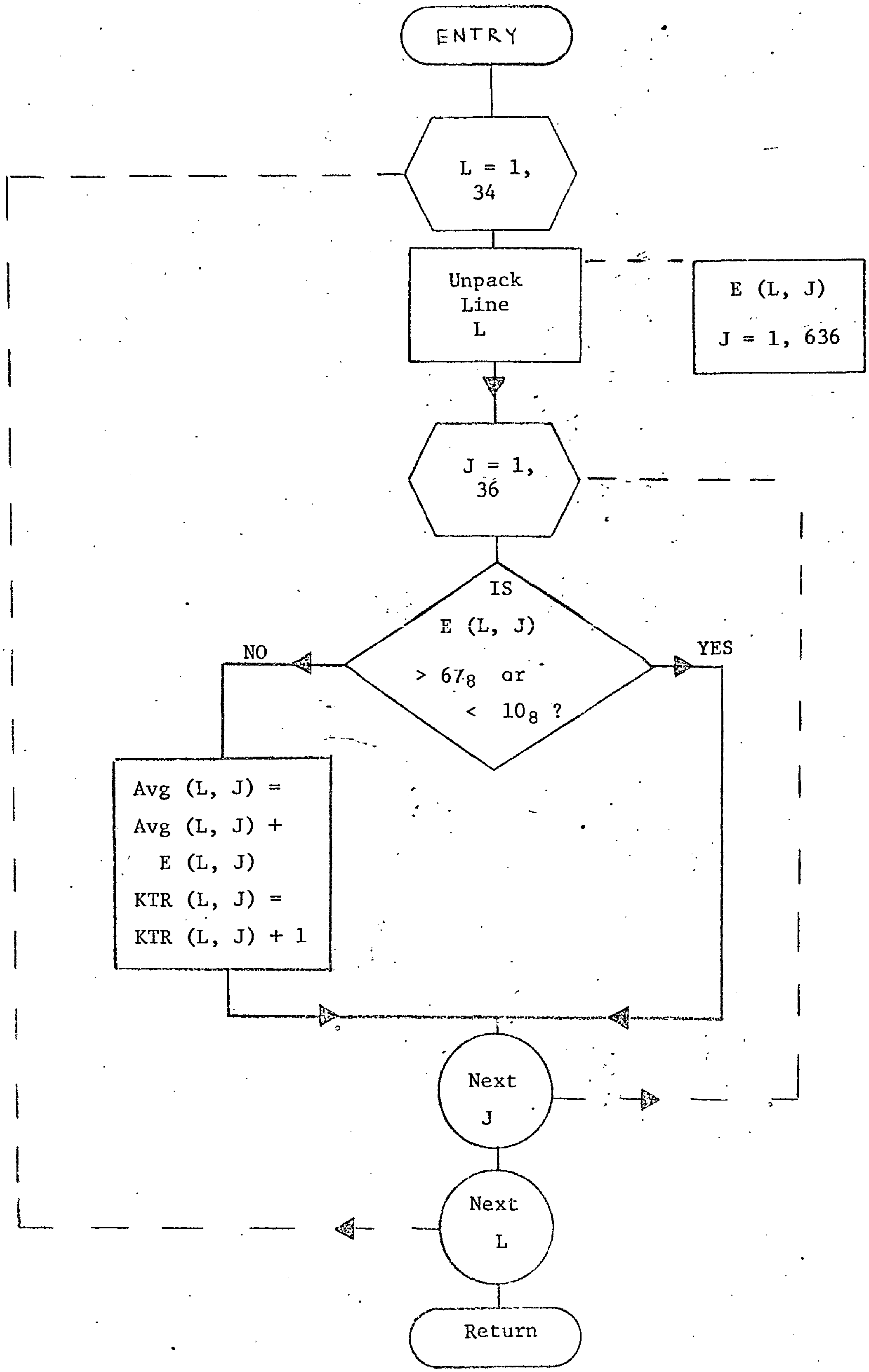
Of the three undesirable problems noted above, the only easily corrected one is the problem of the scan line signature. Neither of the other two problems are very easily corrected for, and, in fact, must merely be lived with. The drum signature can probably be corrected for, but with a large amount of computer time being required to detect just what it is (since as a two dimensional signal it is so weak). Conceptually, this correction would be performed as follows: form a 34 x 636 array of correction factors to be applied over each complete drum signature (syncronized, say, with the start of each (very easily detected) drummark), and apply the correction factors when making other corrections. The method of forming the drum signature correction factors would be like the forming of the line scan signature correction factors except that the averaging would be carried out over successive 34 x 636 areas synchronized with the drummarks. If we accept as adaquate, say, a mere fifty thousand points averaged for each drum signature correction point, (an amount just barely acceptable in view of the weakness of the signature), this would involve reading about two hundred reels of LO digitized data, an insurmountable logistics problem in a reasonable time. (The computations would all be tape read bound). The algorithims for detecting the drum signature in this way were carried as far as the flow chart stage, but were not implemented because another method was found to match adjacent framelets even with the drum signature. See Figs. 2.7.1 thru 2.7.3.



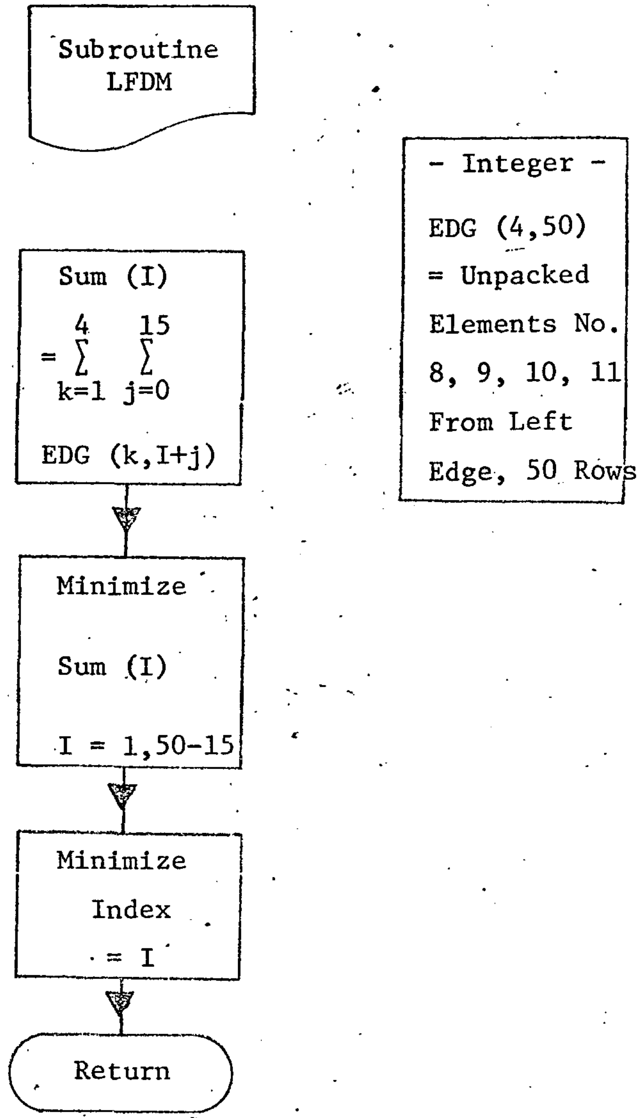
2.7.1 General Flow Chart - Correction For Drum Signature



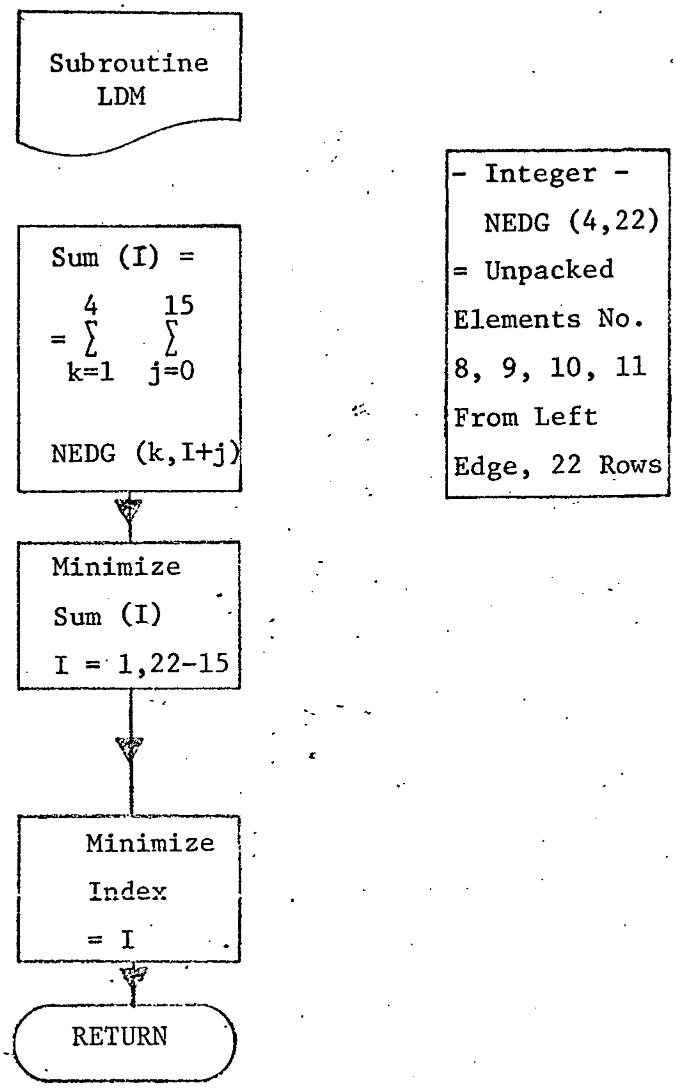
2.7.1 General Flow Chart - Correction For Drum Signature - Continued



2.7.2 Subroutine AVGRE



2.7.3 Subroutine LFDM



2.7.4 Subroutine LDM

Much more difficult, probably impossible, is the problem of the mechanical scan nonlinearity. This nonlinearity seems to be non-periodic. In any event, the period is not short enough to be of any use to us in solving the problem of joining adjacent framelets. (It would have been very useful to have a verticle linearity test pattern (easily machine recognizable and also removable) flashed on the film with the Reseau and edge data pattern, but none was available.) The mechanical scan nonlinearity is severe and nonrepeatable over practical amounts of data.

3. <u>The digitization system</u>

Probably the most unbelievable error of all lies in the design of the digitization system. The main source of this error is the fact that the sampling pulse generator is crystal controlled (and so not phase locked to the sync pulse). This design deficiency is all the more remarkable in view of the fact that the GRE obviously has the necessary basic hardware to obtain phase locked sampling. In order to understand the effect of a free running sample oscillator, we may refer to the block diagram Fig. 3.1. In order to illustrate the effect of a free-running oscillator, consider the following examples: In Fig. 3.2a, the "on" pulse from the sync detector occured at point a on the 555 kHz. sine wave; similarily, in Fig. 3.2b (and c) the "on" pulse from the sync detector occured at point b (respectively c) on the 555kHz sine wave in Fig. 3.2. If we "reconstruct" the three digitized versions of our input video signal by linear interpolation, the three digitized versions (of the same information) appear as in Fig. 3.3. The net effect is that there is a $\pm$ one digital element (peak) jitter in the digitized information.

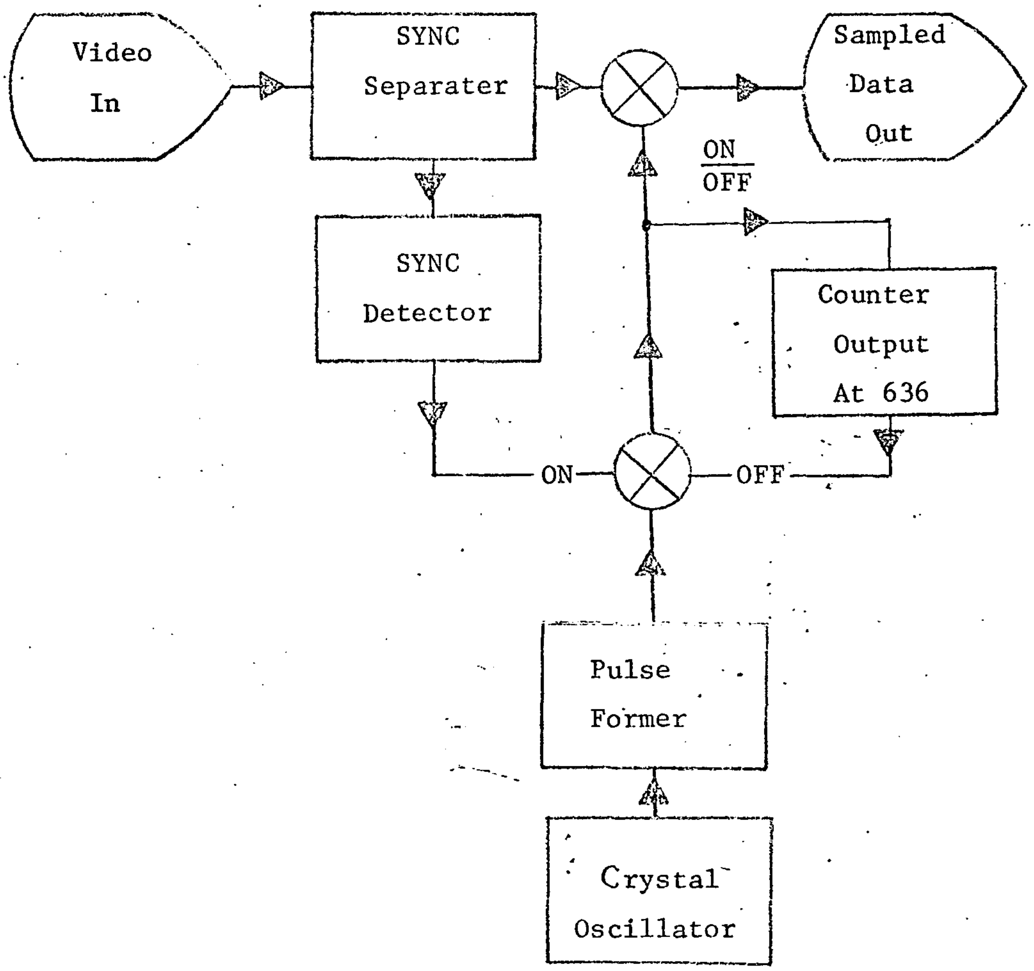


Fig. 3.1 Operation of Digitizer

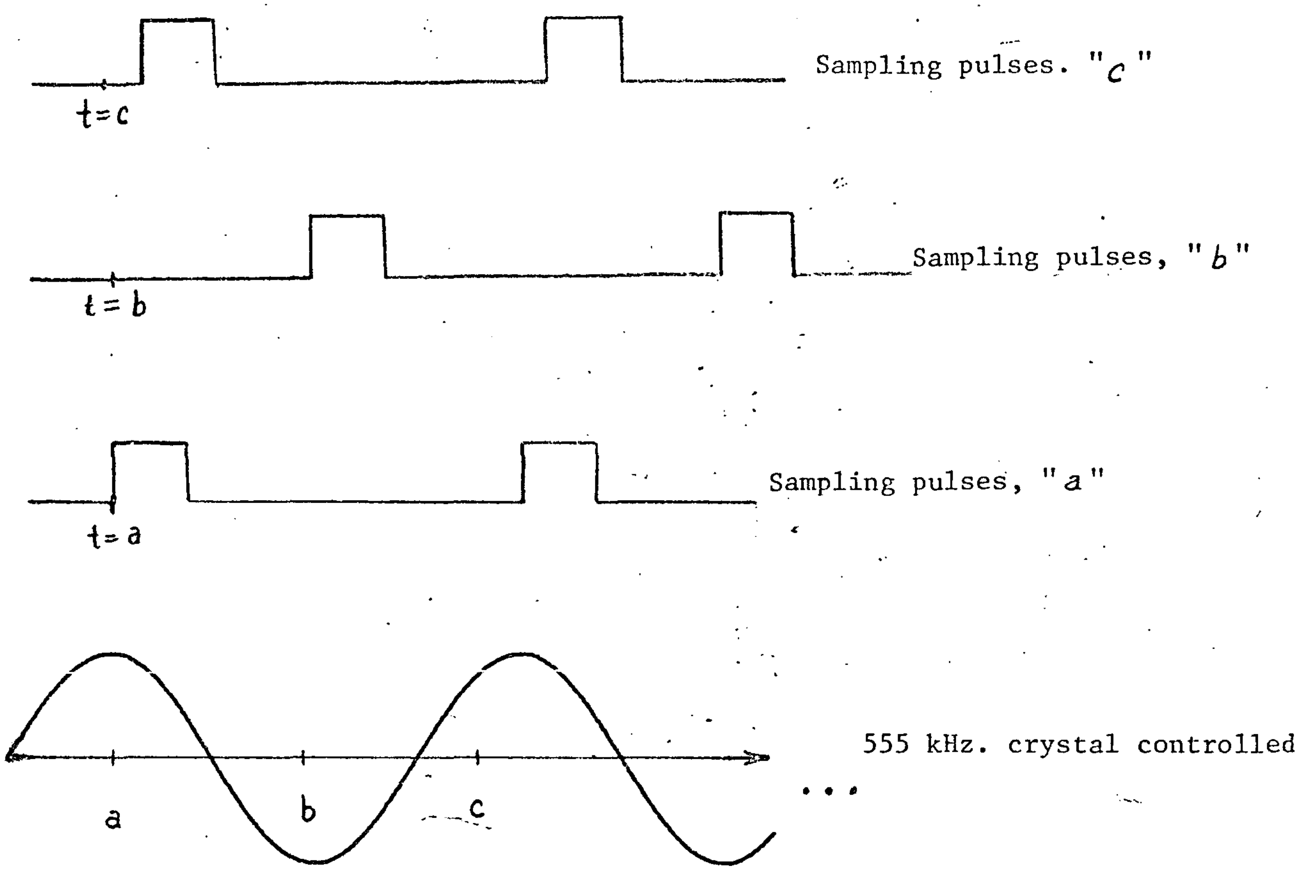

t=c
Sampling pulses. "c"
t = b
Sampling pulses, "b"
t= a
Sampling pulses, "a"
a
b
c
555 kHz. crystal controlled

Fig. 3.2

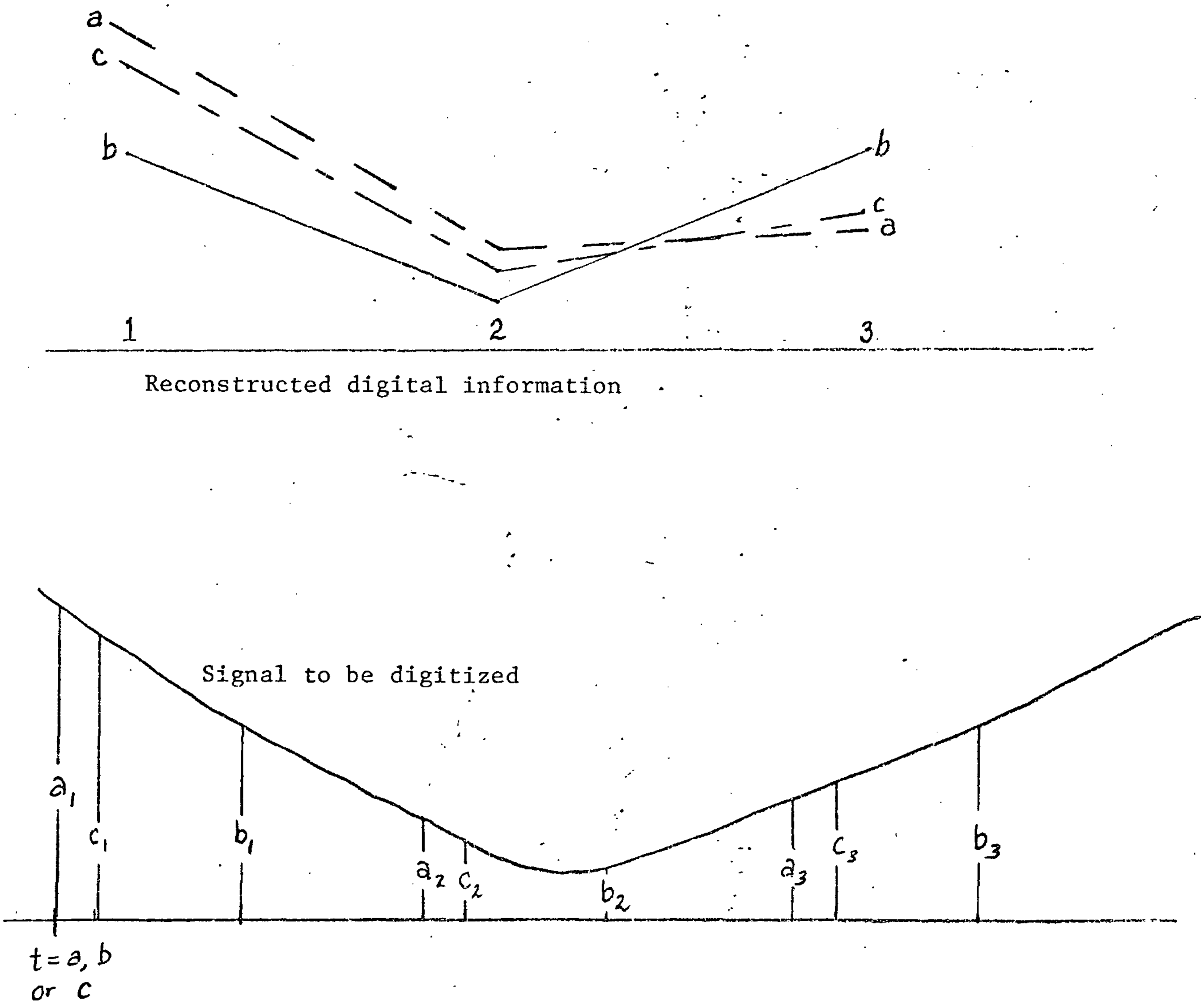

a
c
b
b
c
a
1
2
3
Reconstructed digital information
Signal to be digitized
a1
c1
b1
a2
c2
b2
a3
c3
b3
t = a, b
or c

Fig. 3.3

(It should be pointed out that the sampling rate is adaquate, and that in these examples the input video signal has frequency components well within the limits imposed by the sampling restrictions.) Since, over a short period of time, both the analog tape drive and the crystal oscillator are very stable, one may expect to see a "beat" between the two repetition rates resulting in the jitter not being random at all but moving back and fourth snake-like.

The jitter has three serious effects on the problem of joining adjacent framelets:

(a) The jitter in adjacent framlets is far from being synchronized, and results in a $\pm$ 2 (peak) digital elements relative jitter.

(b) The drummarks are spread by the jitter over an area 4 digital elements wide instead of 2 elements (the width of a cross section of a particular drummark.) This restricts the useable overlap even more.

(c) A feature, such as a crater edge, which we would like to locate and use in the matching process, has had some "wormey noise" added. To illustrate this problem, we refer to Fig. 3.4. In Fig. 3.4a, a Lunar feature (perhaps the edge of a crater) is shown in the left and right edges of adjacent framelets. Any matching algorithm will attempt to match up the jitter in the adjacent framelets, resulting in an

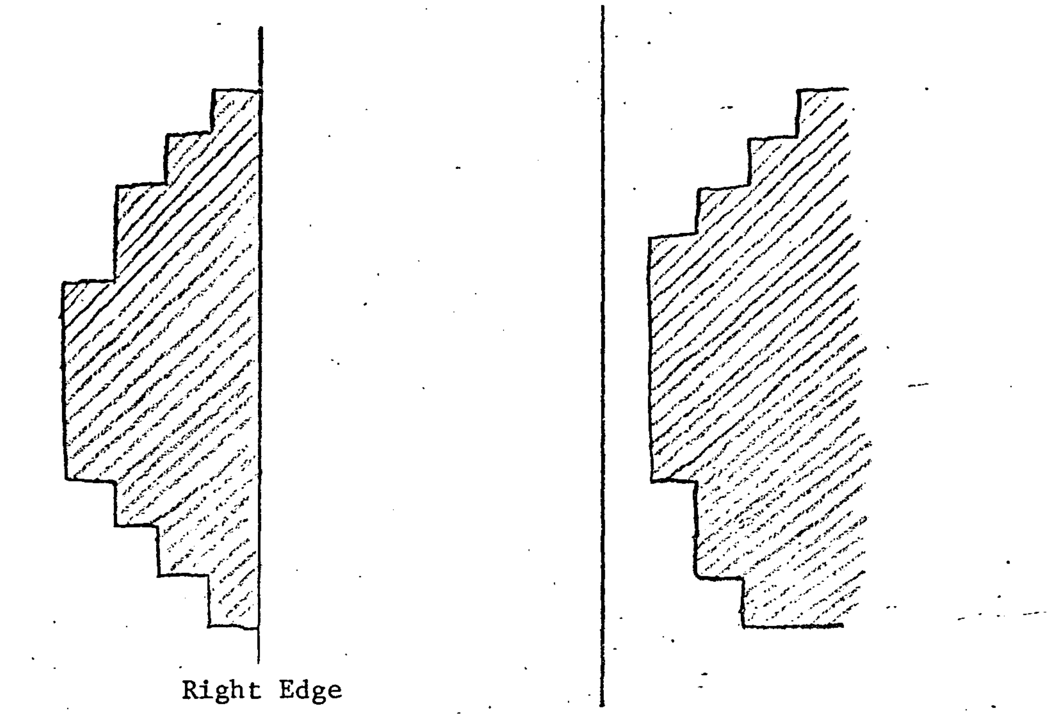


Fig. 3.4a

No Jitter Matching

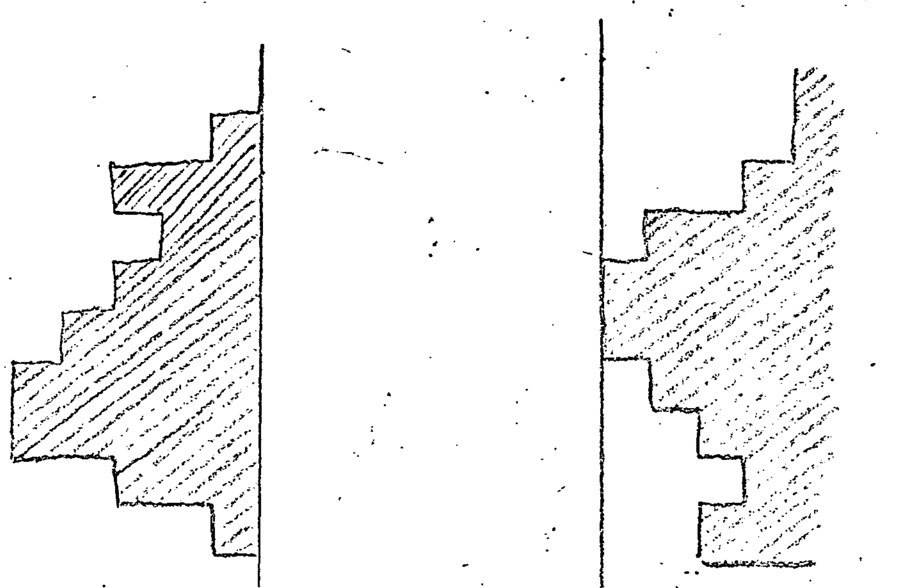


Fig. 3.4b

With Jitter Matching

unreliable match (in the sense that the undistorted data would not show a match at this exact point). Although the jitter is but one element peak, this may move the match point up or down the framelet several elements because of the longer period of the jitter.

There is another problem which the digitizer suffers from. The first element (that is, the first output element from the digitizer in each scan line) has nothing to do with the input data (nor, as far as we can tell, has the first element anything to do with the jitter). It is almost always a very small number. Thus one more element is not available for matching. (It would be interesting but not very profitable to speculate on the cause of the first bad element.)

In order to illustrate the location of the drummarks, we need to see how the video signal is presented to the digitizer. Fig. 3.5 illustrates the form of a single video scan line. The digitization rate is 555.5 kHz. With 1105 $\pm$ 11 μsec between drummarks, this yield 614 $\pm$ 6 characters between drummarks. The actual space between drummarks found by the correction factor program, reported on fully in the report on Project B, is 626.5 - 9.5 + 1 = 618 characters. Since the distance between drummarks is 0.100 $\pm$ 0.001 inches, the horizontal interval between characters can be given as (0.100 $\pm$ 0.001) ÷ 617 inches = 0.0016207 $\pm$ 0.00016 inches = (4.12 $\pm$ .41) μ. (By observation of the RESEAU test pattern and comparison with actual data, we arrive at 4.105 μ as the spacing between elements in the scan line direction; this work will be reported on under Part III of this report on Project A.)

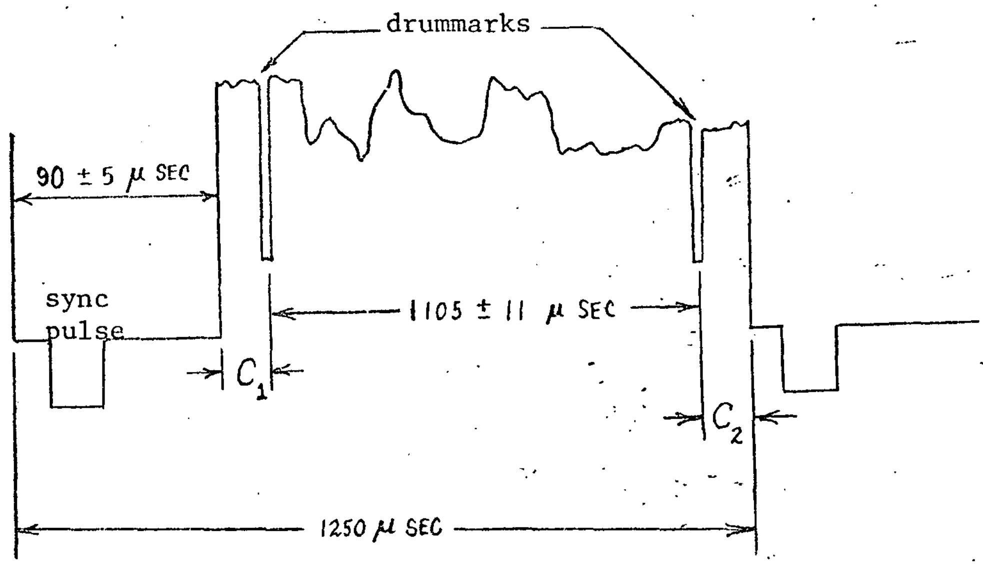


Fig. 3.5 Composite Video Signal

Observation of actual digitized data shows that $C_1 = C_2$ within one digital element. If we take $C_1 = C_2$, we obtain $2C_1 = 1250 - (1105 \pm 11 + 90 \pm 5) = 1250 - (1195 \pm 16)$ so that $C_1 = 27.5 \pm 8$ μs. At the digitization rate of 555.5 kHz, this is between 11 and 20 digital elements. Actual observation gives the drummarks centered at element 9.5 (on the left) and 626.5 (on the right). Thus, either the values given for the video signal are somewhat off or else the digitizer does not digitize the whole line. (It makes no difference in the problem of joining the framelets which; however, by examination of the GRE film, we suspect that the values for the video signal are off, since the drummarks seem to be too near the edge.)

4. Limitations owing to the Univac 1108

The word size of the 1108 is 36 bits, in 6 6 bit bytes. It is very expensive to handle data in any other combination other than a multiple of 6 bits. The byte size choosen for the LO project was 6 bits. The digitized data is packed 6 elements to a word and written in records consisting of 94 lines of 106 words each. This means that, in order to use the LO data as written by the digitizer, one must allocate a buffer area in core of nearly 10,000 words merely for temporary storage of a packed line. In addition, we expect to need lines one at a time on demand. If the tape is only read once every 94 lines, at which time 10,000 word record is read, then we may expect gross inefficiency of computer time in those problems which have a fair balance of reading tape and calculation. Indeed, the join problem is probably not possible unless the information is written one line per record because of the limited core space, and it is certainly not practical. In order to simplify the whole problem, we decided (after several months of struggling with the 94 line records) to write our own special tapes 1 line per record. This process is carried out by two special programs which provide input tapes for the algorithm which finds the match between framelets and which provide input tapes for the join algorithm. The documentation for the first program follows; the second is documented in the report for Project B.

COMPUTER PROGRAM DOCUMENTATION

LEFT-RIGHT EDGE OF FRAMELET

Program LRE

Project A

By

Jack Bryant

and

R. L. Wendt

Prepared by

Applied Scientific Research, Inc.

Houston, Texas

Under Contract NAS 9-10577

For

MAPPING SCIENCES DIVISION

National Aeronautics and Space Administration

Manned Spacecraft Center

Houston, Texas

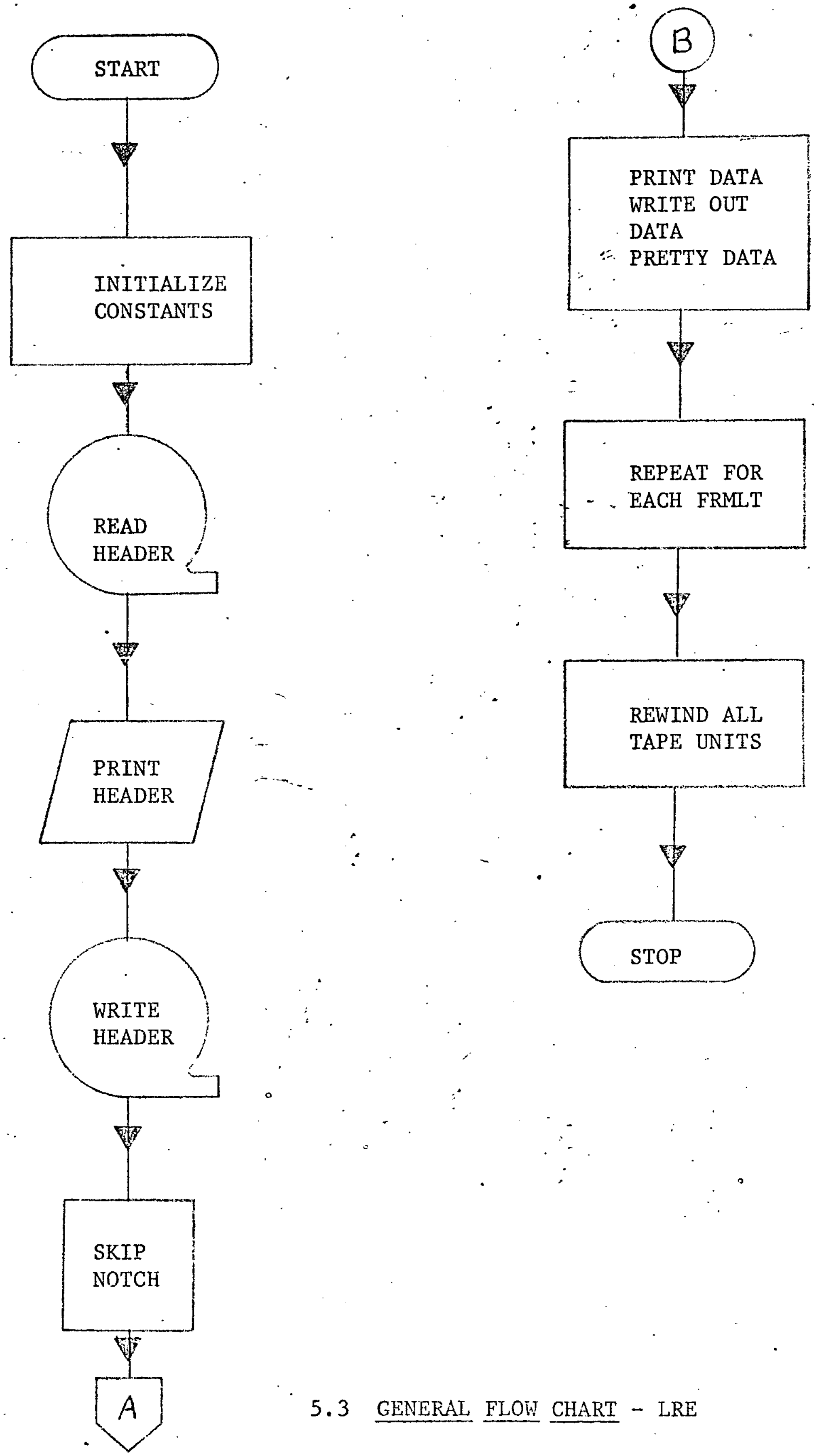
September, 1971

5.1 INTRODUCTION

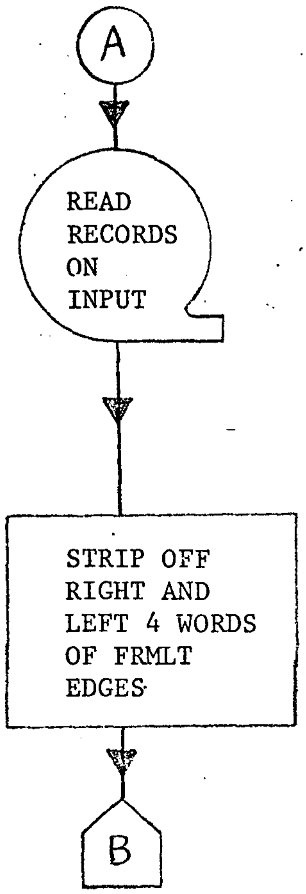
The problem of using LO data as provided by EDITED tapes in the solution of the matching framelets problem lies in the highly inconvenient way the data is spread out in 94 line records. Since all that is involved in the matching problem is the left and right edges of adjacent framelets, the whole problem is made much faster by preparing a preliminary tape with the left and right edges of the three framelets written out by lines on the left and then right edges. This program is dimensioned to handle specific data, viz. LO III, FR 154 H, FRMLTS 028, 027, 025, SURVEYOR-3 area, but can be generalized in the obvious fashion (by changing the number 2068) to fit framelets of arbitrary length.

5.2 PROGRAM DESCRIPTION

The program initializes constants and reads the Header from the input tape. The Header is written on the output tape followed by an end of file mark. The edge data is skipped over and the data framelet is read. The left and right edges of each line in each record is removed and stored in C. When an end of file mark is read on input the data in C is printed, written on output tape, reduced in tonal resolution and printed. The process is repeated for each framelet and when finished all tape units are rewound. The general flow chart is shown in 5.3:



5.3 GENERAL FLOW CHART - LRE



5.3 GENERAL FLOW CHART - LRE

CONCLUDED

5.4 DESCRIPTION OF PROGRAM VARIABLES

JDIM - The column dimension of array C; also twice number of words to be stripped off right and left edges of the framelets.

KDIM - The dimension of D, 2068 x 8.

HDER - Header array.

B - Input array for data tape.

C - Array of data of right and left edges of framelet.

D - Vector of equivalenced to C.

NFRMLT - Number of framelets per reel of tape.

NCHIT - Number of chits per framelet.

NCOL - Number of words to be stripped off each edge of framelet.

NCOLPI - Index on C.

K - Row counter on C

KK - Index used in stripping off words of framelet.

KW - The number of rows in C to be printed, prettied and written on output tape.

5.5 DATA INPUT

Data tape Surveyor-3 LO 1 FR 154H FRMLTS 028, 027, 025; see also 5.7.2 for detailed description.

5.6 DATA OUTPUT

Data tape used as input by SKINNY algorithm.

5.7 USAGE

5.7.1 CARD INPUT DATA

None.

5.7.2 TAPE INPUT DATA

Header

59 words
Integer
NTRAN read - 1 buffer of 59 words
EOF

Notch

150 rows of 106 words each
Integer
NTRAN read - 1 buffer of 9964 words and
1 buffer of 5936 words
EOF

Digitized LO data

2068 or more lines of 106 words each
packed 94 lines per record
Integer
NTRAN read - 22 (1 buffer of 9964 words)
EOF

Edge Data

550 rows of 106 words each
Integer
NTRAN read - 5 (1 buffer of 9964 words) and
1 buffer of 8480 words
EOF
EOF

Repeat until all data exhausted.

5.7.3 DECK SETUP

See fig. 5.7.3

5.7.4 RUN TIME/LINES OUTPUT

Time: Approximately 2 min. 1108 time. Actual time may depend on delay in hanging tape. Output: Approximately 13,000 lines output.

5.7.5 ACCURACY/VALIDITY

Perfect.

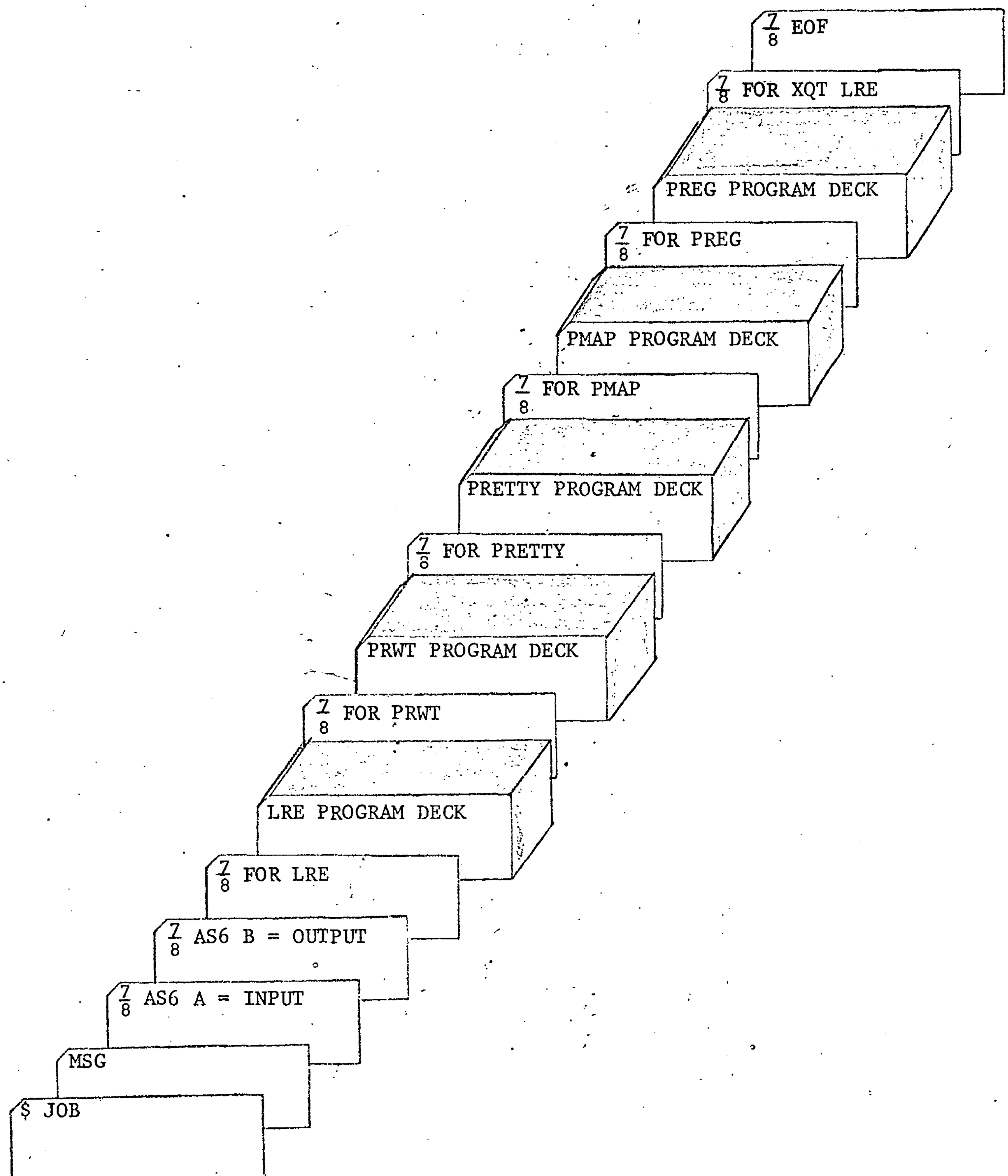
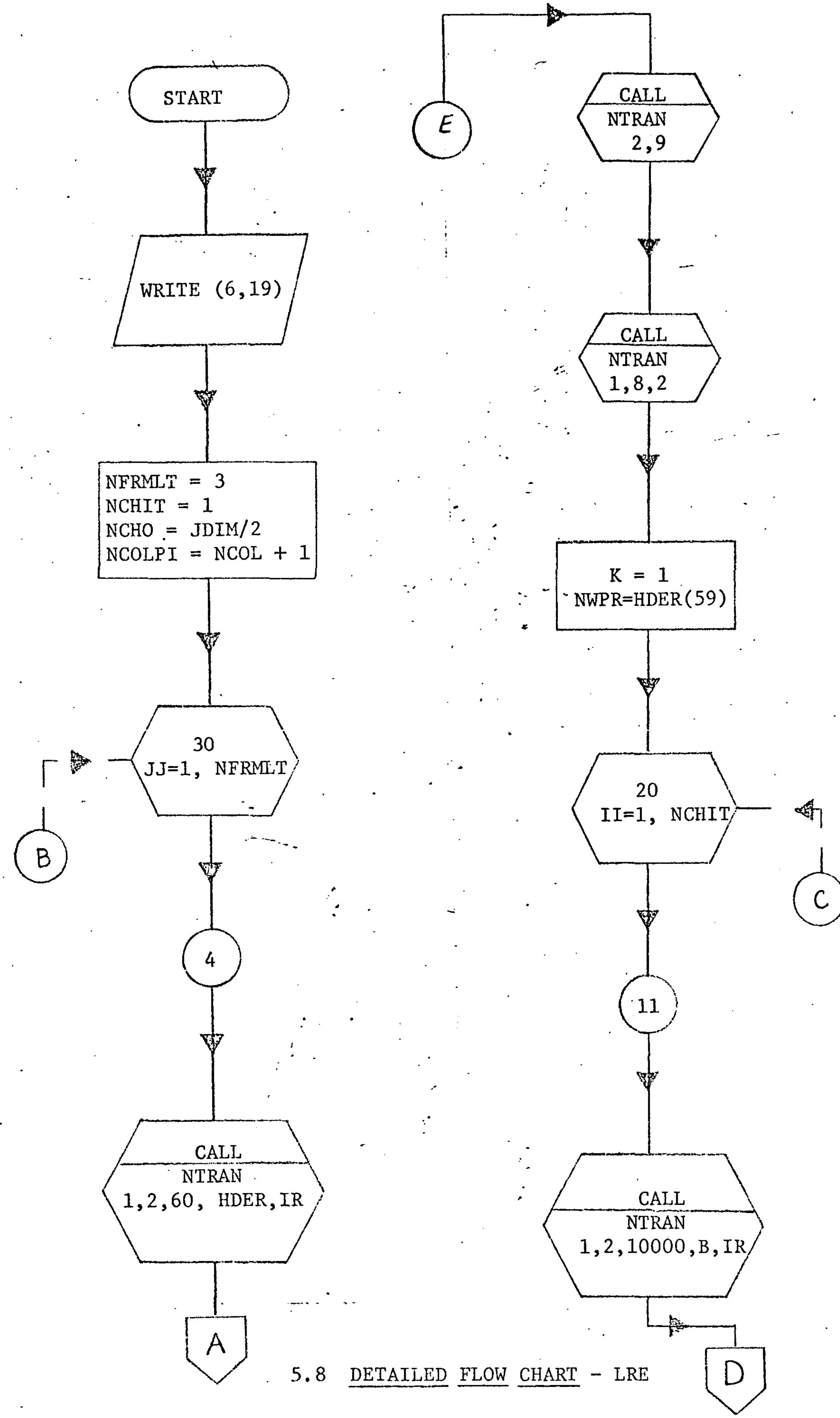


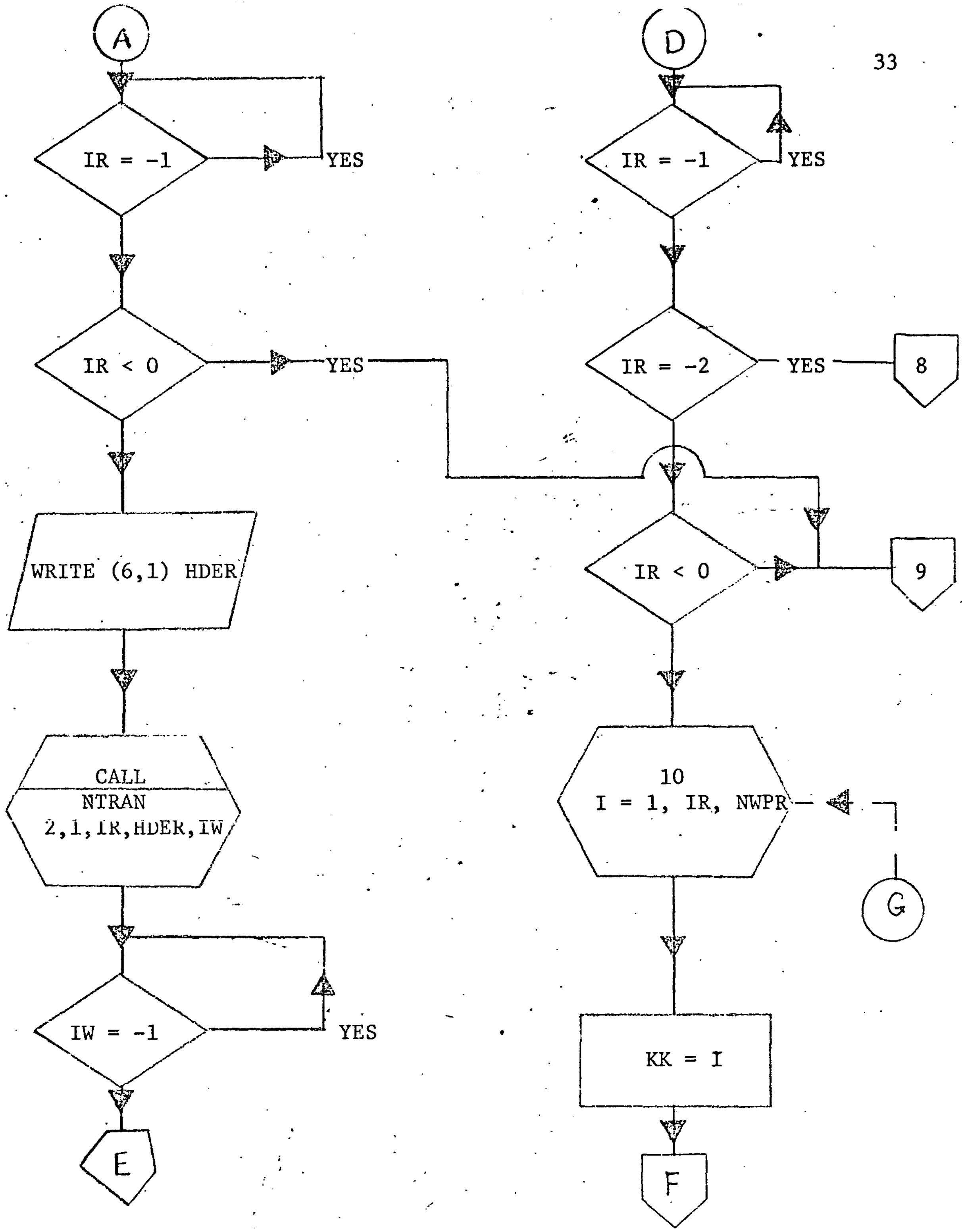
Fig. 5.7.3 DECK SETUP (UNIVAC 1108)

5.7.6 RESTRICTIONS

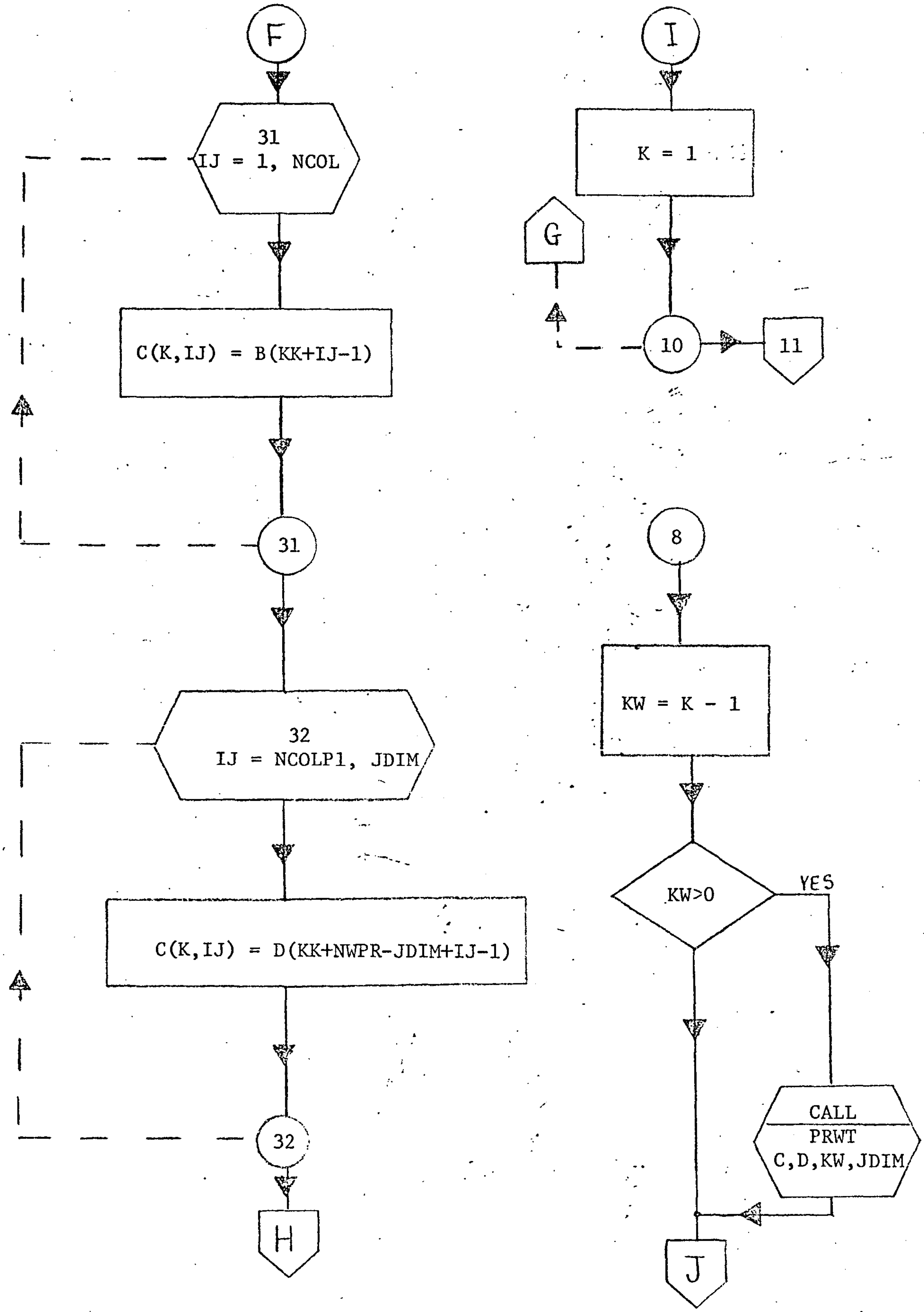
As written, the program wants to have 2068 lines per extended chit as described above. This number can easily be changed to suit other forms of input (as it has been for the complete framelet of 15,690 lines).



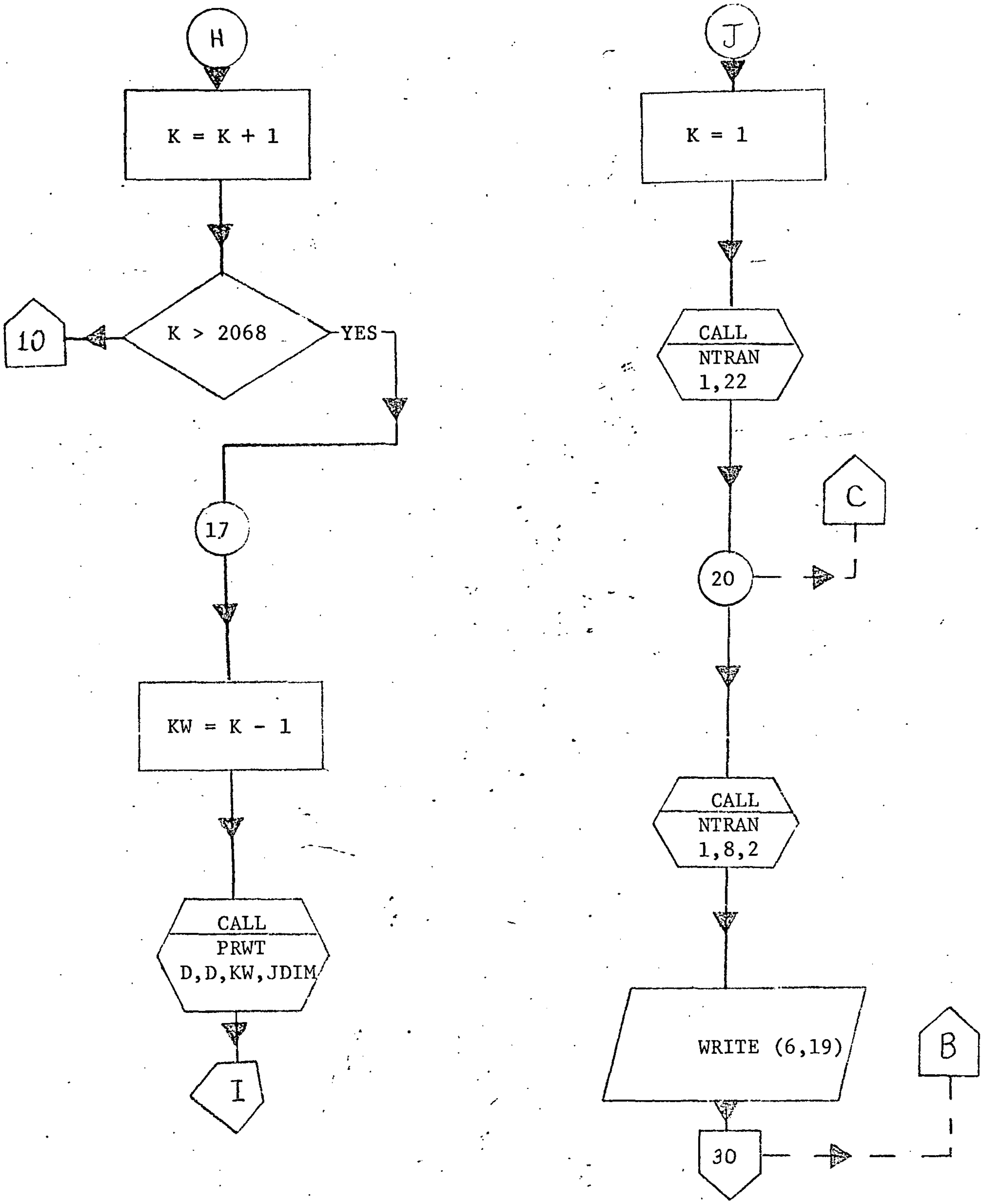
5.8 DETAILED FLOW CHART - LRE



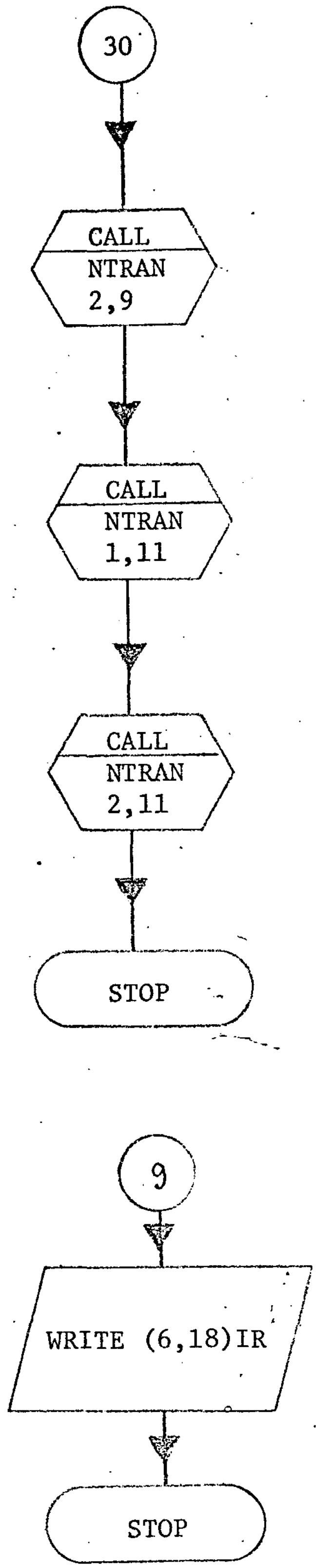
5.8 DETAILED FLOW CHART - LRE



5.8 DETAILED FLOW CHART - LRE (Continued)



5.8 DETAILED FLOW CHART - LRE (Continued)



5.8 DETAILED FLOW CHART - LRE (CONCLUDED)

5.9 DESCRIPTION OF SUBROUTINE

5.9.1 PRWT (PRINT, PRETTY DATA, WRITE OUTPUT TAPE)

5.9.2 CALLING SEQUENCE PRWT (C, D, KW, JDIM)

5.9.3 SUBROUTINE DESCRIPTION

This routine prints the data in C in its original tonal resolution by splitting the data in C into right and left edges of the framelets. The left and right edges of the framelets are written on the output tape. The tonal resolution of the data is reduced by a factor of 8 via PRETTY and the left and right edges of the framelet are printed again.

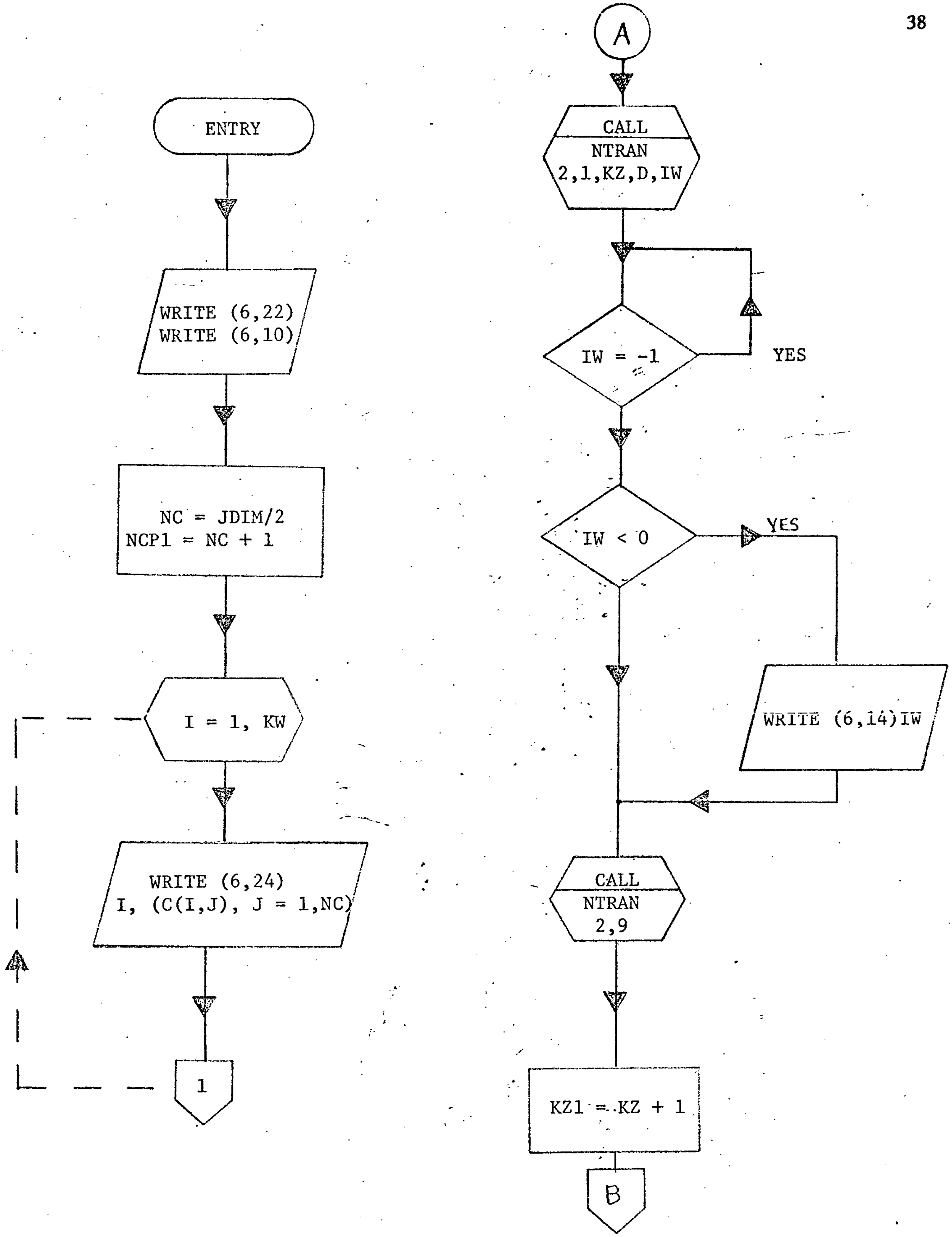
Description of Variables:

C, D, KW, IDIM - Explained in LRE.

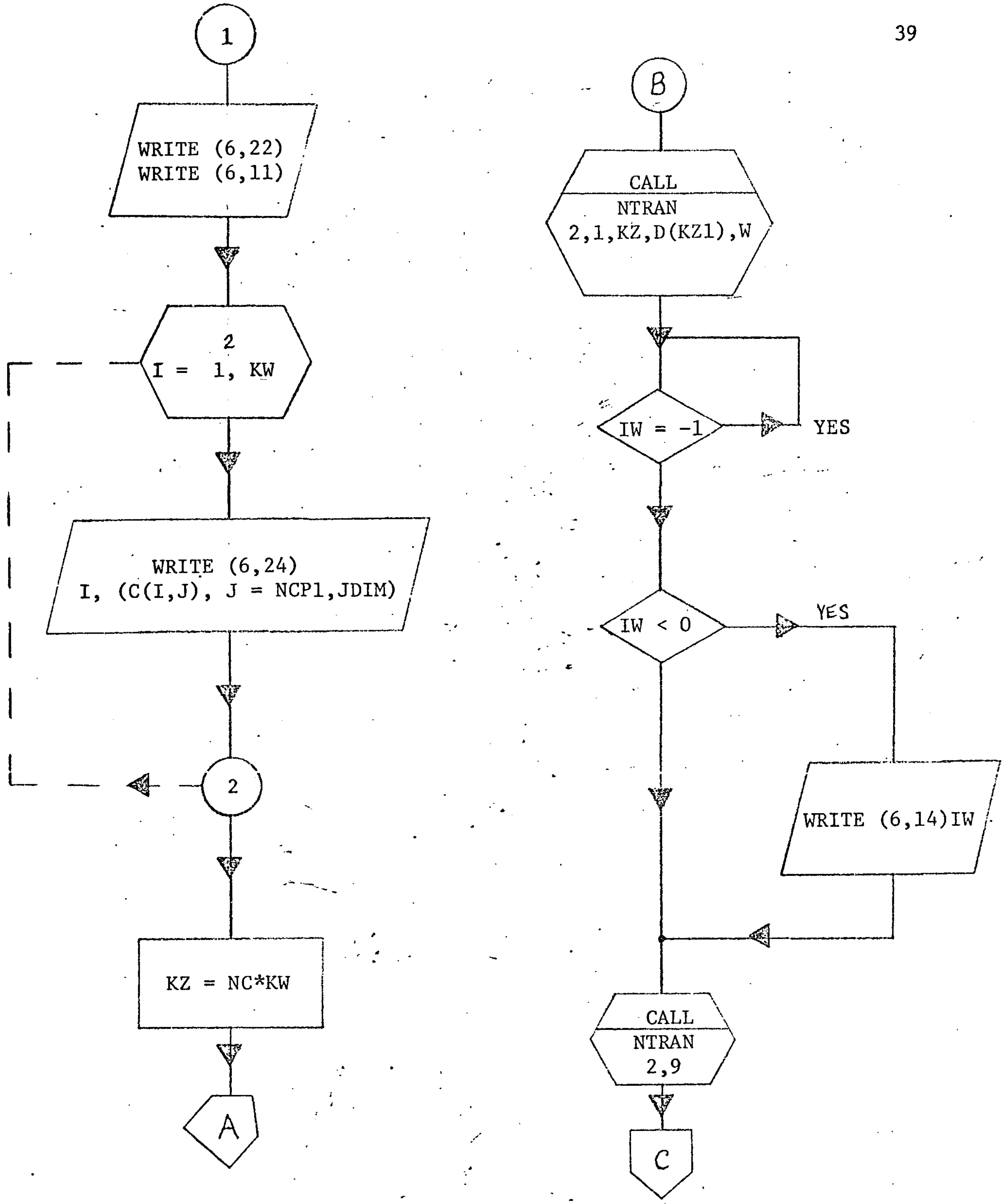
NC, NCP1 - JDIM/2 and JDIM/2 + 1.

KZ - The number of words per record written on output tape.

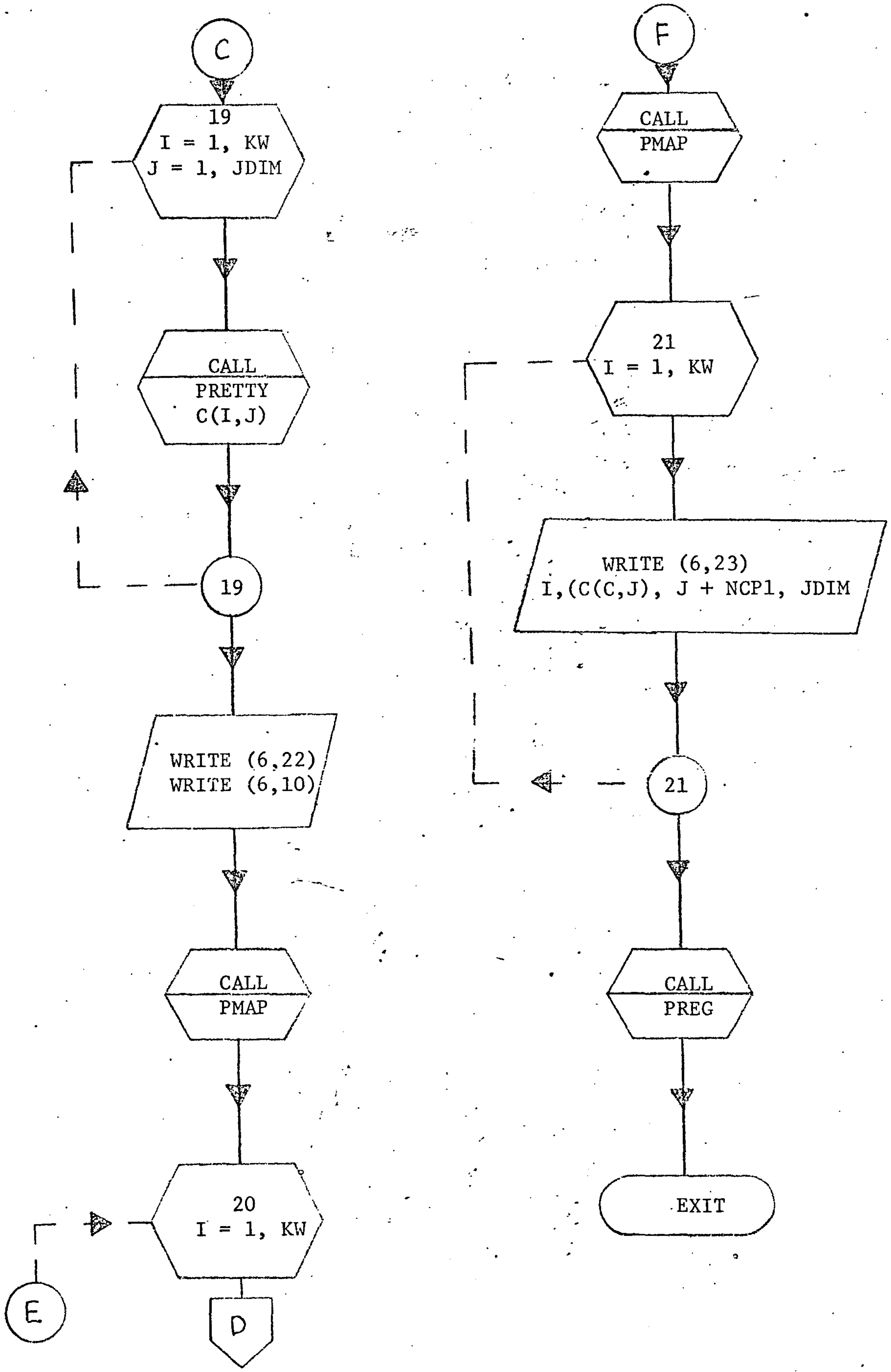
KZ1 - KZ + 1 used in addressing D on output.



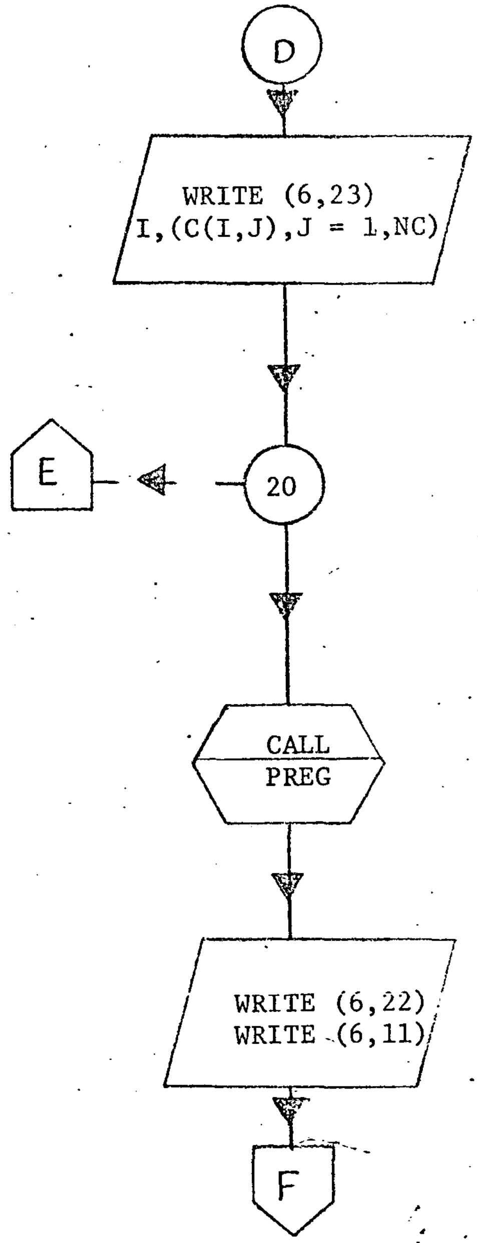
5.9.4 DETAILED FLOW CHART - PRWT



5.9.4 DETAILED FLOW CHART - PRWT (Continued)



5.9.4 DETAILED FLOW CHART - PRWT (Continued)



5.9.4 DETAILED FLOW CHART - PRWT (CONCLUDED)

5.9.5 OTHER SUBROUTINES

NTRAN (UNIVAC I/O ROUTINE)

PRETTY
PMAP } (DOCUMENTED ELSEWHERE IN LO PROGRAM)
PREG

6. Detailed examination of the actual overlap - COLUMNS - in LO III, FR154H, FRMLTS 028, 027, 025, SURVEYOR 3 AREA.

In order to further explain the small extent of the actual overlap, we give an example based on actual test data. The COLUMN numbers are 1 thru 636, with 1 the "left"edge. (Left when the high resolution pictures are held with the number at the bottom, reading correctly; we understand that this system is not consistent with the sun position and that what we have is a mirror image of actual Lunar features. However, thinking of the picture as being a matrix stored a line at a time, our "left" edge does in fact correspond to the column with smallest index.) Historically, the column match between framelets was unknown to us before solution of the match problem, so that this part of the report is out of order; however, it is important to document the fact that the LO data is out of specifications (in that there is less than specified overlap between adjacent framelets), and this will help motivate the match algorithm SKINNY.

6.1 COLUMN overlap found between 028R - 027L

Fig. 6.1.1 following represents the overlap found between 028R (COLUMN number 618 thru 636) and 027L (COLUMN numbers 1 thru 18). The defective elements (not available for matching) are indicated by a * (drummarks) or $ (bad first element). As can be noted, the overlap between 028R and 207L is 11 digital elements.

6.2 COLUMN overlap found between 027R - 025L

Fig. 6.2.1 following represents the overlap found between 027R (COLUMN numbers 618 thru 636) and 027L (COLUMN numbers 1 thru 18). As can be seen, the overlap in this match is 16 digital elements

6.3 Average COLUMN overlap found 028R-027-025L.

The average overlap found was 13.5 digital elements. In order to corrolate this overlap with LO specifications, consider Fig. 6.3. As can be seen, were the data as specified, the overlap would show the drummarks exactly meeting. This overlap (not actually experienced) is illustrated in Fig. 6.3.2. The data actually fails to meet specifications in two ways. First, as mentioned before, the drummarks are too near the edges of the framelet. (This is probably caused by the scan line width being less than specified since independent callibration using the Reseau flash pattern gives 0.100" for the drummark spacing. We may infer that the width of the framelet as digitized is 0.1029 ± .0002" instead of the specified 0.105".) Second, the film advance is more than specified. If we take as 0.1029" the actual width of a framelet, then there should be 0.000162" per digital element, so that 0.0029" ↔ 18 digital elements should be the overlap with a film advance of 0.100". (Since the encountered average overlap is 13.5 digital elements ↔ 0.0022", we may conclude that the film is advanced an average of 0.1007" instead of the specified 0.100", with a variation in our sample of ±.0004".) These two defects combine to make the problem of finding the overlap area nearly impossible, since there is essentially but one COLUMN in the worst cases with which matching can be done.

```
FRAMELET 028 RIGHT EDGE

                                    *   *   *   *
618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636
                                ----------------------------------------------------------------------
                                1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18
                                $                           *   *   *   *

                                            FRAMELET 027 LEFT EDGE
```

$ = BAD LEFT EDGE

* = DRUMMARK AREA

Fig. 6.1.1 Column Overlap Found Between 028L - 027R

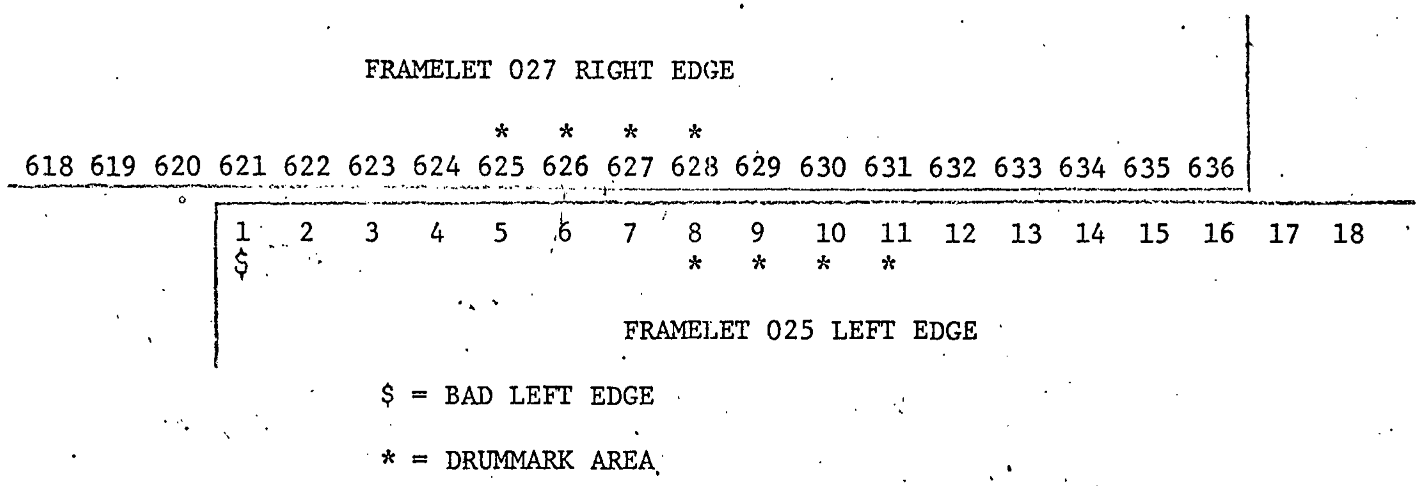
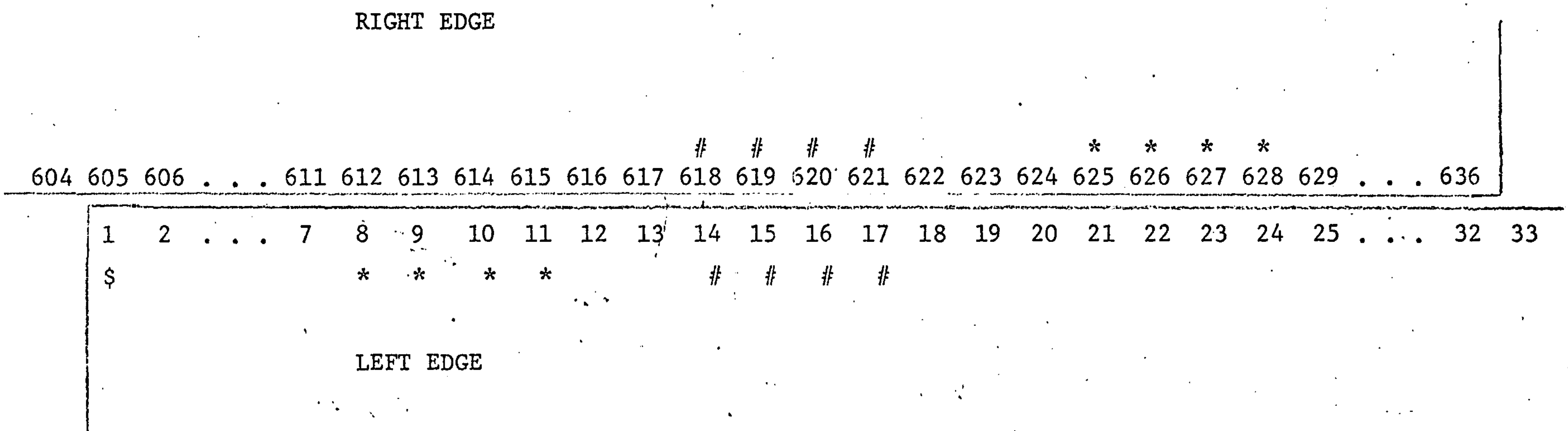


Fig. 6.2.1 Column Overlap Found Between 027L - 025R



$ = BAD FIRST ELEMENT

* = DRUMMARK AREA ACTUALLY FOUND

= DRUMMARK AREA AS SPECIFIED

Fig. 6.3.2 Specified Column Overlap.

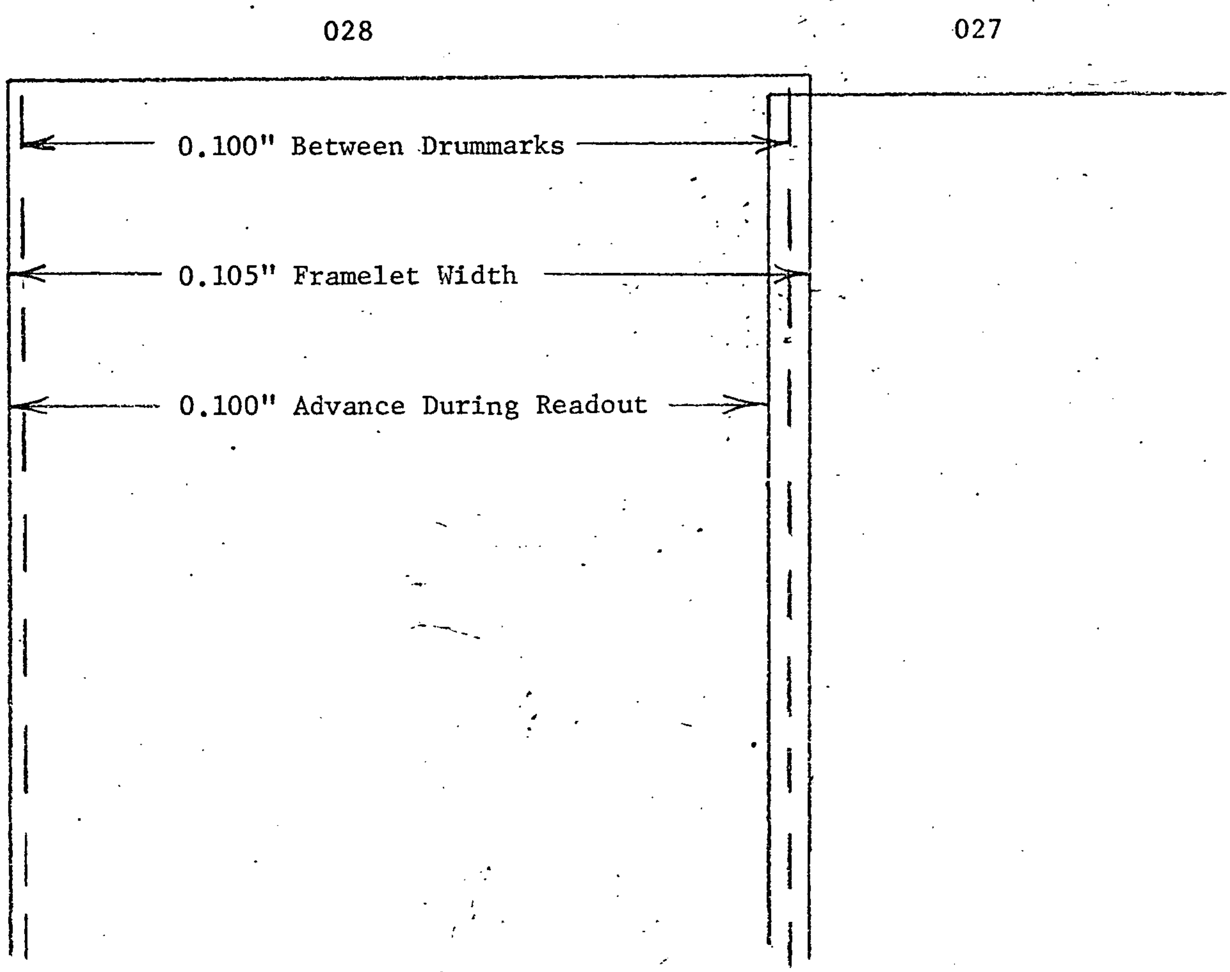


Fig. 6.3.1 LO Photo Read-Out Specifications.

6.4 Analysis of Worst Cases - COLUMNS

Fig. 6.4.1 illustrates possible problems which may arrise owing to the bad left edge and drummark areas killing off possible match areas. For the purpose of illustration, we may think of the right edge as being fixed and the left edge as being positioned in various places to study the interaction. As can be seen, not even one left side framelet column misses the drummark area of the right side framelet. Also, the maximum size of a column area which is consistently(in all cases), (a) not in a drummark and, (b) not itself a drummark or bad left edge, is one. This is the reason for the "jumping around" of the "standard" match area we encounter in SKINNY. Thus, we can use COLUMN 2 of the right (lower numbered) framelet in the match illustrated by A, B, C, D, E, F and K, L, M, and COLUMN 6 of the right framelet in the match illustrated by G, H, I, J. Thus we selected the following COLUMN MATCH SCHEME:

Trial overlap	A-F	COLUMN 2
Trial overlap	G-J	COLUMN 6
Trial overlap	K-M	COLUMN 2

More details on this proceedure will be given in the documentation to SKINNY.

	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636
POSITION A	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
B		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
C			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
027R-025LD				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
E					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
F						1	2	3	4	5	6	7	8	9	10	11	12	13	14
G							1	2	3	4	5	6	7	8	9	10	11	12	13
H								1	2	3	4	5	6	7	8	9	10	11	12
028R-027LI									1	2	3	4	5	6	7	8	9	10	11
J										1	2	3	4	5	6	7	8	9	10
K											1	2	3	4	5	6	7	8	9
L												1	2	3	4	5	6	7	8
M													1	2	3	4	5	6	7

Fig. 6.4.1 A-M Illustration of Likely Column Overlap.

50

7. Detailed examination of the actual overlap - ROWS - in LO III, FR 154 H, FRMLTS 028, 027, 025, SURVEYOR 3 AREA.

Nonlinearity in the ROW direction (up and down) was expected; our tests confirmed this nonlinearity.

7.1 ROW match, 028 R - 027L.

Table 7.1.1 illustrate the ROW match found between 028R and 027L. Because of the scarcity of data (details) found in the overlap, very few match points were found by SKINNY. As can be seen, the deviation is variable with a peak of about 23 to 32, or a difference of 9 rows over 300.

7.2 ROW match, 027R - 025L

Table 7.2.1 illustrates the ROW match found between 027R and 025L. One less match point was found by SKINNY. The nonlinearity here was not found to be as great as in 028R - 027L, but was still substantial with a greater peak (11 rows) variation (but over 600 rows).

7.3 Relative position of framelets

Fig. 7.31 illustrates the relative positions we found of framelets 028, 027 and 025 as we are viewing them. We shall return to this later in the JOIN problem.

028 R ROW NUMBER	027 L ROW NUMBER	DEVIATION (ABSOLUTE VALUE)
312	289	23
683	660	23
985	953	32
1171	1147	24
1344	1315	29
1533	1502	31
1707	1679	28

Table 7.1.1 ROW Match, 028R - 027L

027 R ROW NUMBER	025 L ROW NUMBER	DEVIATION (ABSOLUTE VALUE)
173	224	51
461	516	55
756	818	62
906	965	59
1274	1335	61
1647	1705	58

Table 7.2.1. ROW Match, 027R - 025L

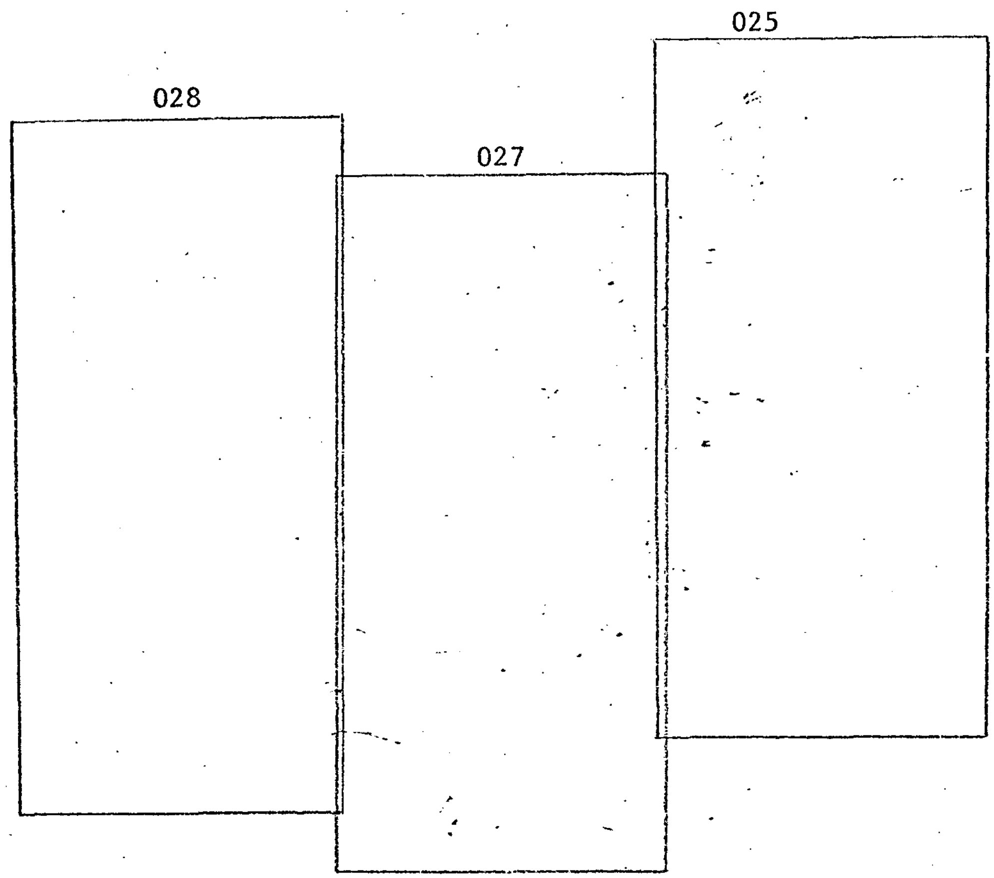


Fig. 7.3 Relative Position of Framelets

COMPUTER PROGRAM DOCUMENTATION

Match Lunar Orbiter Framelets

Program SKINNY

Project A

by

Jack Bryant

and

R. L. Wendt

Prepared by

Applied Scientific Research, Inc.

Houston, Texas

Under Contract NAS 9-10577

For

MAPPING SCIENCES DIVISION

National Aeronautics and Space Administration

Houston, Texas

September, 1971

8. DOCUMENTATION OF SKINNY

8.1 INTRODUCTION

The difficulties inherent in the LO data outlined in §§ 1-7 of the report on Project A have resulted in the algorithm SKINNY which is documented here. (SKINNY was so named because initial algorithms worked over an approximately square area which we felt we could expect to be in the overlap on the basis of LO specfications; when the actual overlap was finally determined, we renamed the algorithm SKINNY because it was). The basic idea is to find the match between adjacent framelets by comparing (using some metric or other criterion) a verticle block of numbers in COLUMN 2 or COLUMN 6 centered at a certain point in the left edge of the right framelet with a "floating" block on the right edge of the left framelet. The main and overwhelming difficulty with this is that, in the data we have to work with, the overlap region is nearly void of interest, so much so, in fact, that a human operator has some difficulty matching the framelets even with the many other clues he may use (for example to ignore the drummarks or to look for large features (e.g. craters) which branch over two or more framelets). Because of the systematic noise (mainly drum signature and jitter), we must first detect an area where the strength of the "signal" (i.e. actual Lunar features) is at least comparable to the noise.

The remainder of this documentation deals with the actual solution of the problem of matching framelets of about three chit size. The problems of matching whole framelets, and discussion of possible alternate solutions to the problem will be deferred to later sections.

8.2 SKINNY ON 2068

The configuration of the three framelets we were given has been noted in §§6 and 7 above. Since a priori one has no knowledge of the relative placement of the rows, the algorithm starts by trying to find a match near the center of the framelets. The success of the algorithm depends critically on the correct starting match since the row nonlinearity is not great and we do not look over as large an area for match points after the start.

One of the basic problems in matching the Surveyor-3 data is that the digitized data constitutes less than a framelet. This means, as has been noted in §7, that the sub-framelets may be offset with respect to the first rows of the sub-framelets. This problem is overcome by attempting to find a match at or near the center of the sub-framelets. Subroutine LKFR is used to determine where an "interesting" area is near the center. Refer to Fig. 8.2.1. The fixed area S receives data from the original array C over IDIM by JDIM rows and columns centered at the point described by the interesting area. The array F over which FM will vary receives data over an array twice as large as S with respect to the rows and 1 1/2 times as large as S with respect to the columns. This allows for a large discrepancy in the rows between two subframelets and helps to insure an initial match without which the algorithm cannot proceed.

The fixed array S has a sub array SM which can vary over columns 2 and 6. The moving array FM moves over all columns of the overlap region except those columns which may contain drummarks. As the arrays vary the norm is computed, all norms, row numbers, column numbers are written on the drum and the strict minimum is selected. The row and column number of the

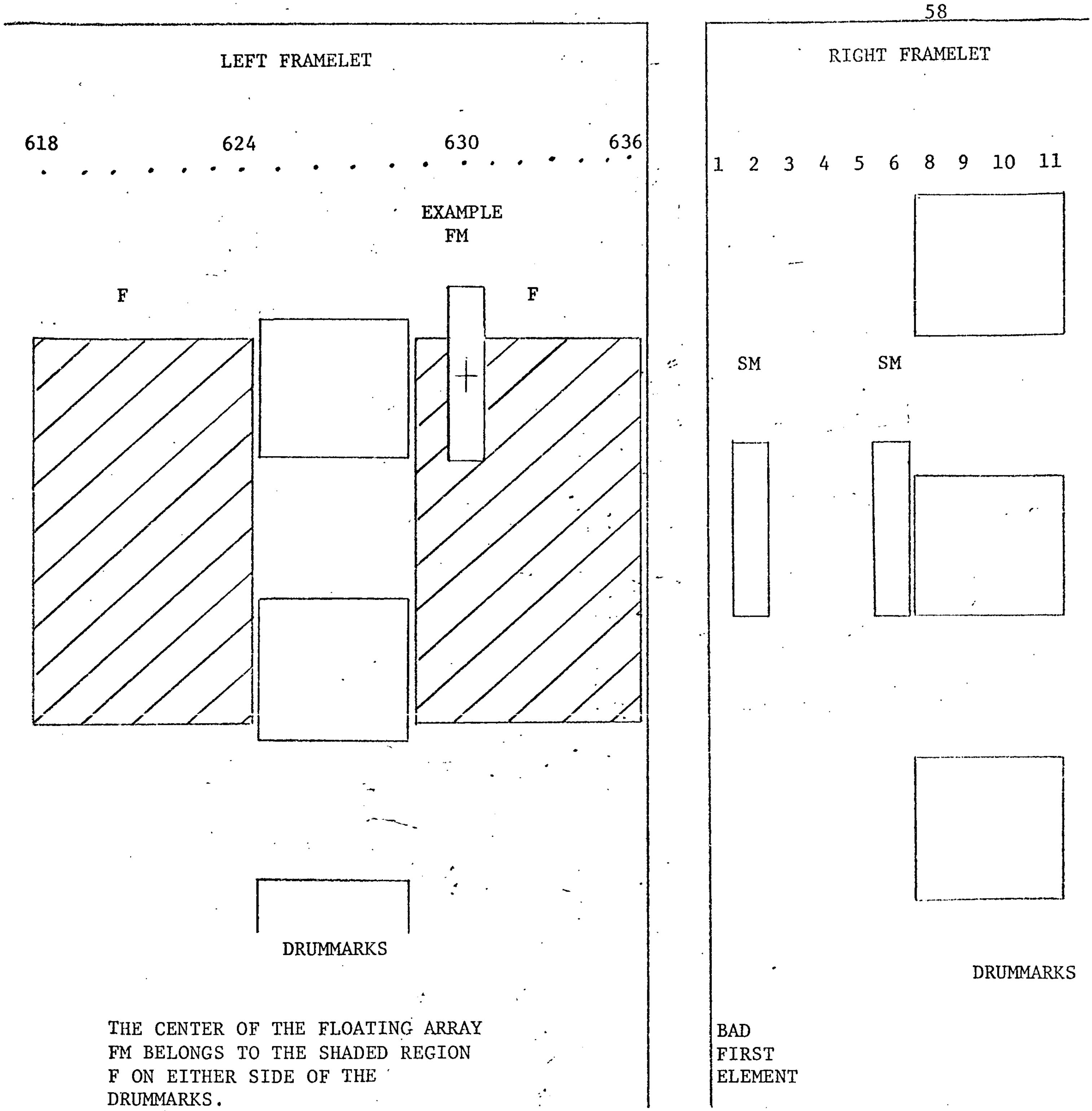


Fig. 8.2.1. Overlap of Two Adjacent Framelets, Illustrating F, SM

strict minimum is selected. The row and column number of the moving array is saved and the column of the match for the array SM. The candidates for the minimum are saved in BUF and printed so the user can observe the order of magnitude of the norm at the minimum.

The drum is rewound, the norms, rows and columns read and those which are within TOL of the strict minimum are written on the drum. In the event that there is more than one point for which the norm is within TOL of the strict minimum then Subroutine BACKUP is called in an attempt to find the best match of those candidates. Otherwise, Subroutine FMPT is called to find match points for the rest of the data. The coordinates of the match points are printed and when this is finished the algorithm switches to the next two edges of adjacent sub-framelets and continues the process.

8.3 GENERAL FLOW CHART

The general flow chart for the main program follows. Documentation of the subroutines LKFR, NORM, BACKUP, FMPT, and PRNT is included in §8.5. The variables and arrays F, S, SM, FM, BUF, I, J, ROW, COL, NORM, I1, I2, J1 and J2 are given detailed descriptions in §8.5.2.

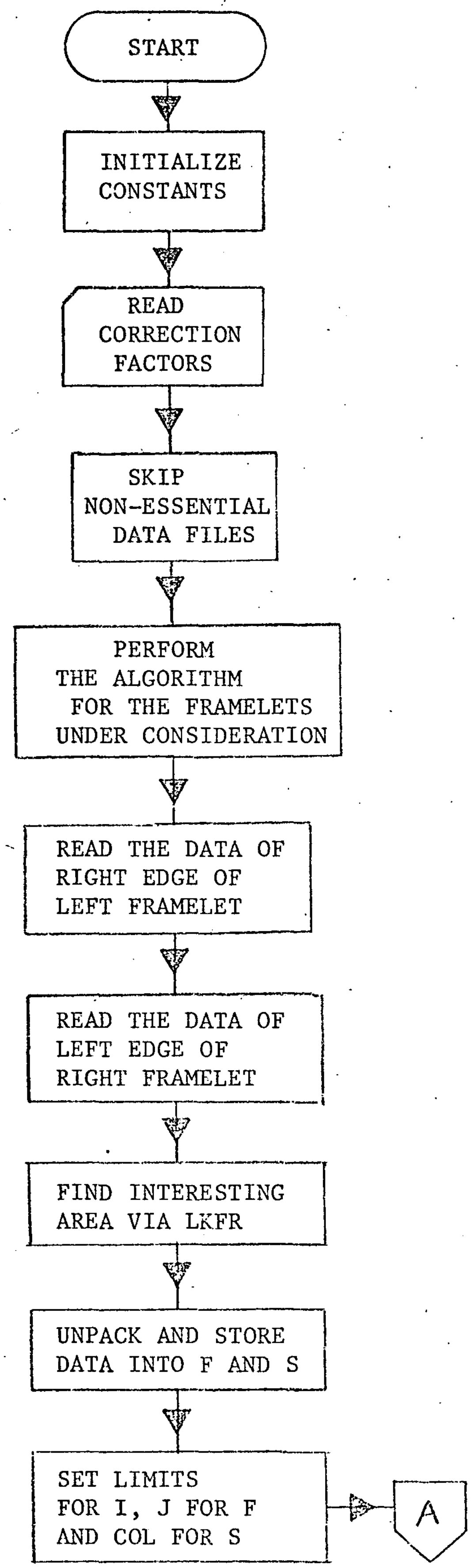


Fig. SKINNY 8.3.1. GENERAL FLOW CHART, SKINNY

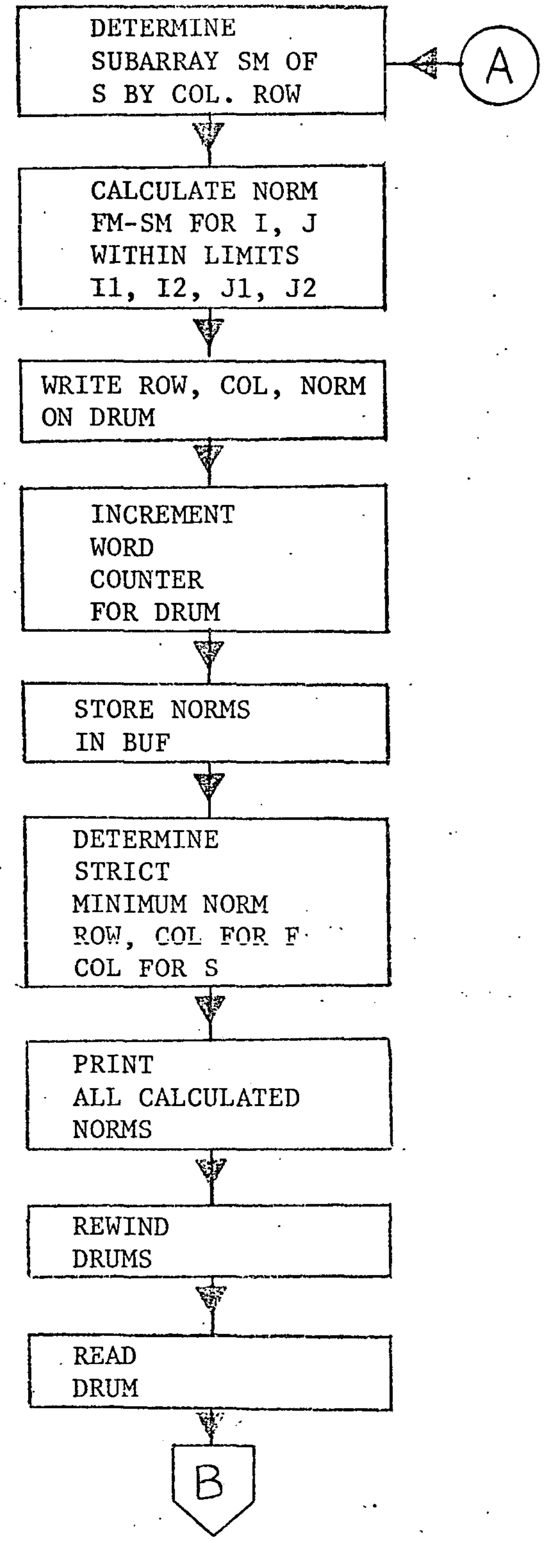


Fig. 8.3.1. GENERAL FLOW CHART, SKINNY, CONTINUED

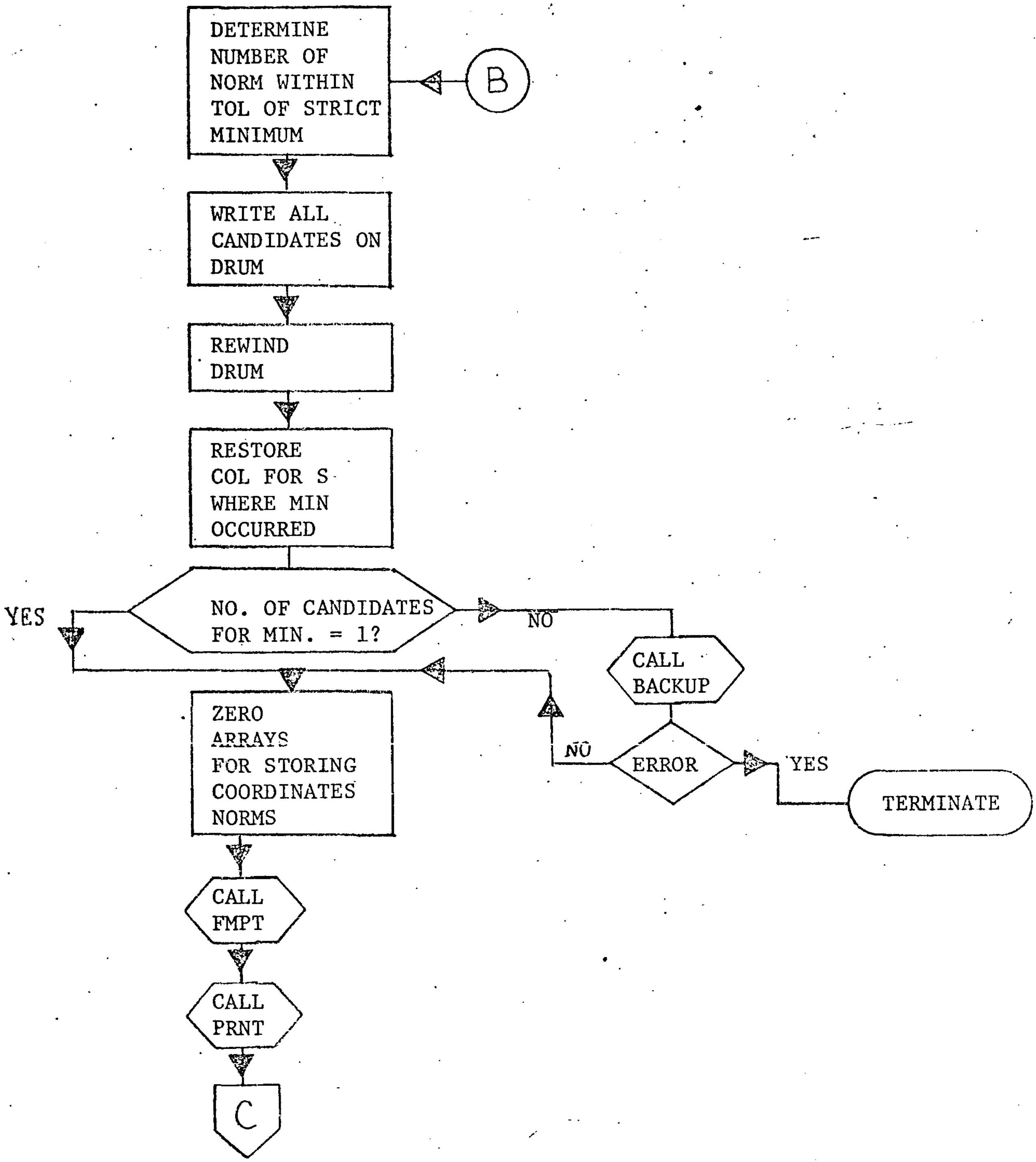


Fig. 8.3.1. GENERAL FLOW CHART, SKINNY, CONTINUED

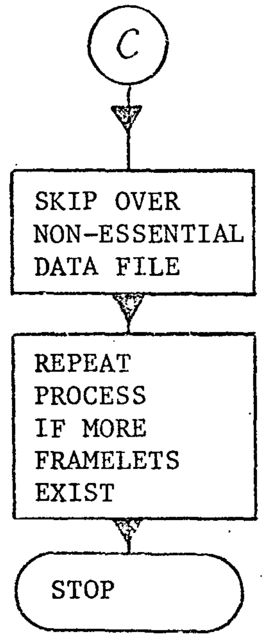


Fig. 8.3.1. GENERAL FLOW CHART, SKINNY, CONCLUDED

8.4 USAGE

8.4.1. INPUT DESCRIPTION

8.4.1.1. Card input

Data cards punched by program CAS 3 (documented with PROJECT B) are read in FORMAT 6E13.8. This data constitutes the smoothed correction factors CS.

8.4.1.2. Tape input

For the exact program under discussion here, the tape format is illustrated in Fig. 8.4.1.2.1. The data records are 4 words x 2068 words in length. There are twice as many data records as framelets to be matched plus a header for each framelet. The pate is made by program LRE (see §5).

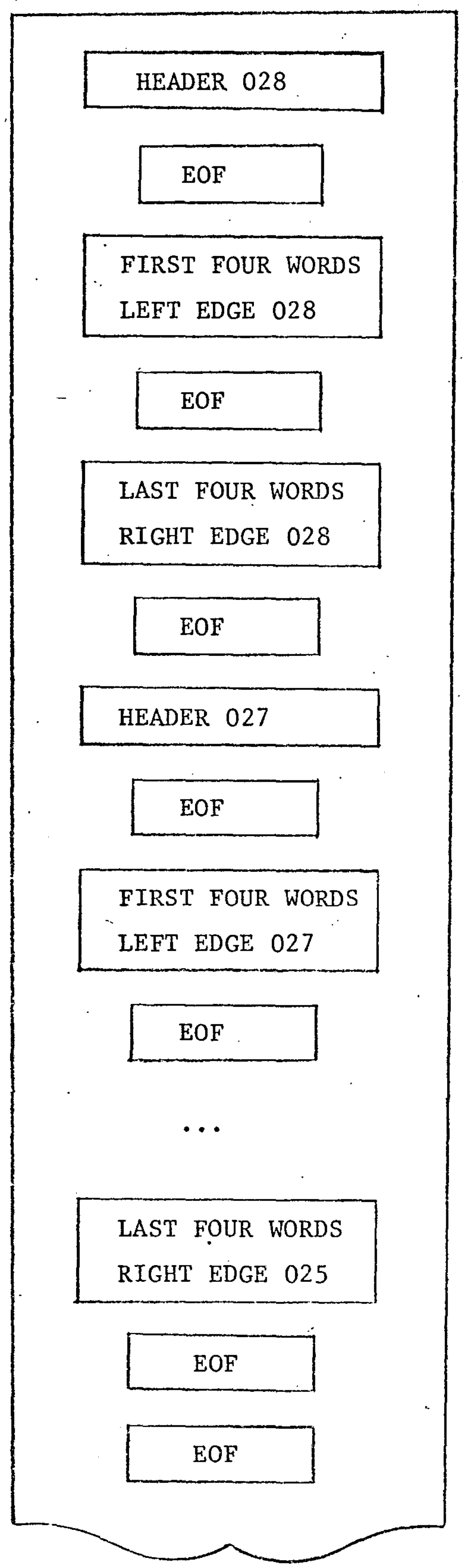


Fig. 8.4.1.2.1. INPUT TAPE FORMAT, SURVEYOR 3, 2068

8.4.2. PROGRAM RUN PREPARTATION

8.4.2.1 Card deck setup.

For the card deck setup, see Fig. 8.4.2.1.1. This card deck is also on PCF tape #8779 currently in the name R. L. Wendt.

8.4.2.2. Required I/O Devices and Special Hardward

The program uses the standard FORTRAN input for data cards. The data tape (Unit A) is read using NTRAN. The program also requires the use of two FH432 drum files. No use is made of FSTRN. NTRAN read/write statements are used to handle the FH432 drum files.

8.4.2.3 Subroutine requirements

The program requires use of the following subroutines:

LKFR

UNPSTR

NORM

BACKUP

FMPT

PRNT

In addition, some of the subroutines of SKINNY have subroutines as follows:

UNPSTR	UNPACK
LKFR	UNPACK, LGSM
FMPT	LKFR, UNPSTR, NORM
BACKUP	UNPSTR, NORM

The documentation of the subroutines is included in §8.5.

8.4.3 OUTPUT DESCRIPTION

8.4.3.1 Printed output

The program first prints out the correction factors. Then it lists the value of the norm over the first attempt to find a match (near the center of the framelet). There follow listings of the values of the norm at other match locations, and then a complete listing of all match points found. The process is repeated until the number of framelets is exhausted. The program produces about 60 pages of output. There is no other output.

8.4.4 EXECUTION CHARACTERISTICS

8.4.4.1 Restrictions

The program documented here is restricted by the size of the 1108 core to sub-framelets with less than about 3000 rows. The program has been generalized to take care of arbitrary framelets with an arbitrary number of rows. The main feature of the generalization is to simply read in batches of (about) 1500 words and repeatedly use the present algorithm.

The program is already set up to handle an arbritray number of sub-framelets.

The program requires about $13,000_8$ core locations for code and $74,000_8$ core locations for storage, leaving $40,000_8$ locations free. (The system uses $14,000_8$ locations).

8.4.4.2 Runtime

The run time is somewhat variable, probably owing to the problems of hanging tapes. However, a good average seems to be about 1 min. 25 seconds per two framelets of 2068 rows matched, so that an approximate 9 chit area will require about 2 min 50 seconds to locate the match points.

8.4.4.4 Accuracy

The program locates match points with sufficient accuracy to enable joining of large areas by the program JOIN. Sometimes, slight adjustments (the magnitude of which can be found by laying the printed, prettyed output of LRE down on the floor (as down a hallway) and looking at the matchpoints found) will result in a slight (3-4 element) improvement. This problem arises when there is simply no detail at all in the overlap which is accessable by SKINNY.. However, the use of SKINNY is indicated even in this situation (where there is very little data in the overlap and the match found may not be perfect) since it finds a very close candidate which is easily adjusted.

8.5 REFERENCE INFORMATION

This section contains the following information.

BLANK PAGE

[redacted]

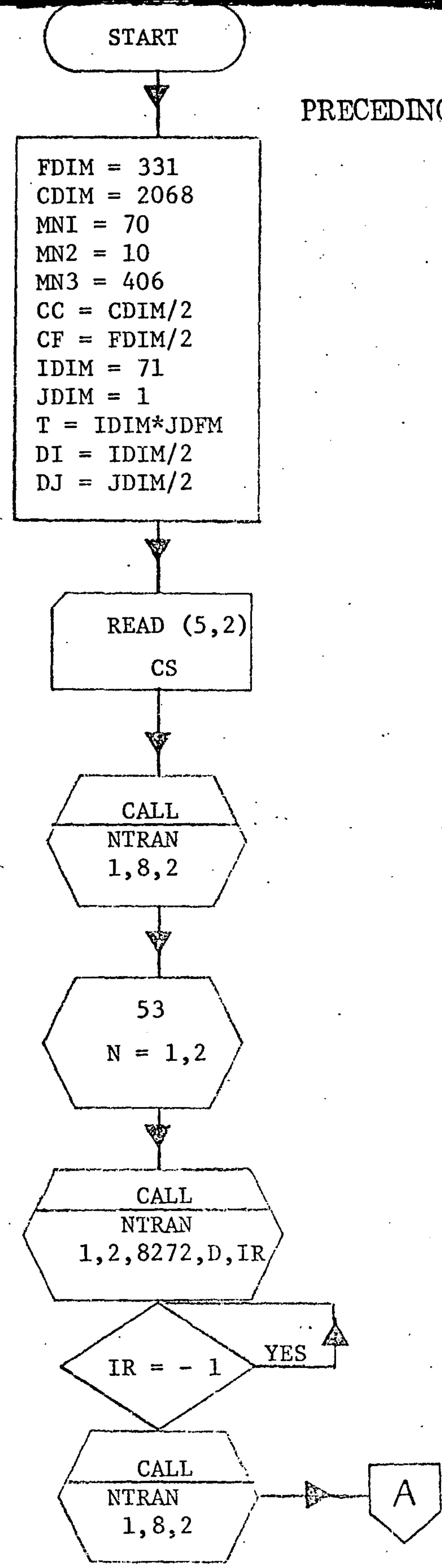


Fig. 8.5.1 DETAILED FLOW CHART - SKINNY

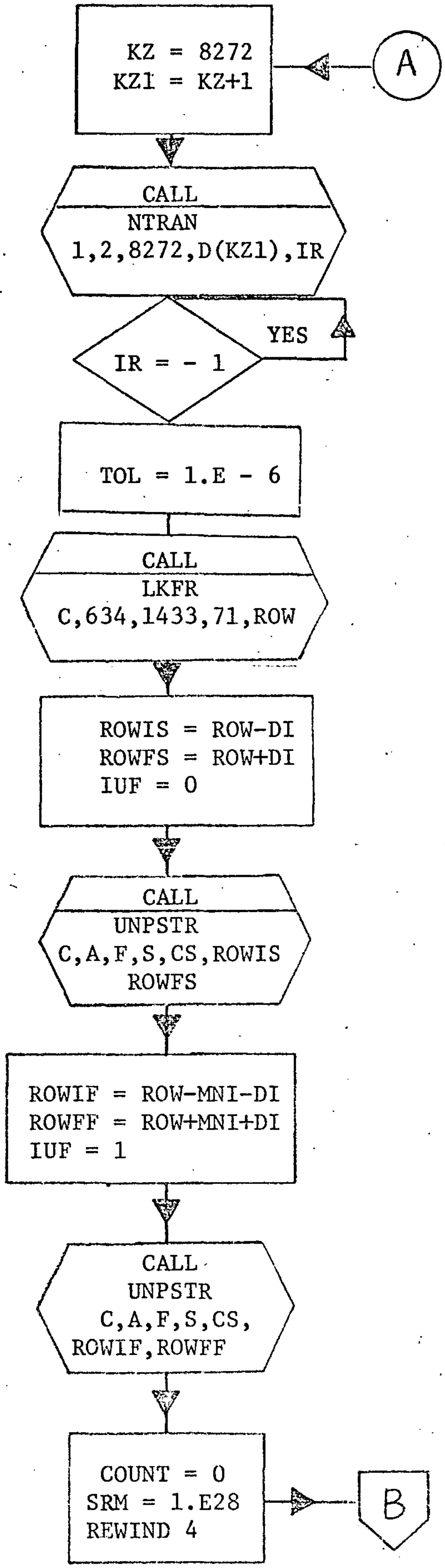


Fig. 8.5.1 Continued

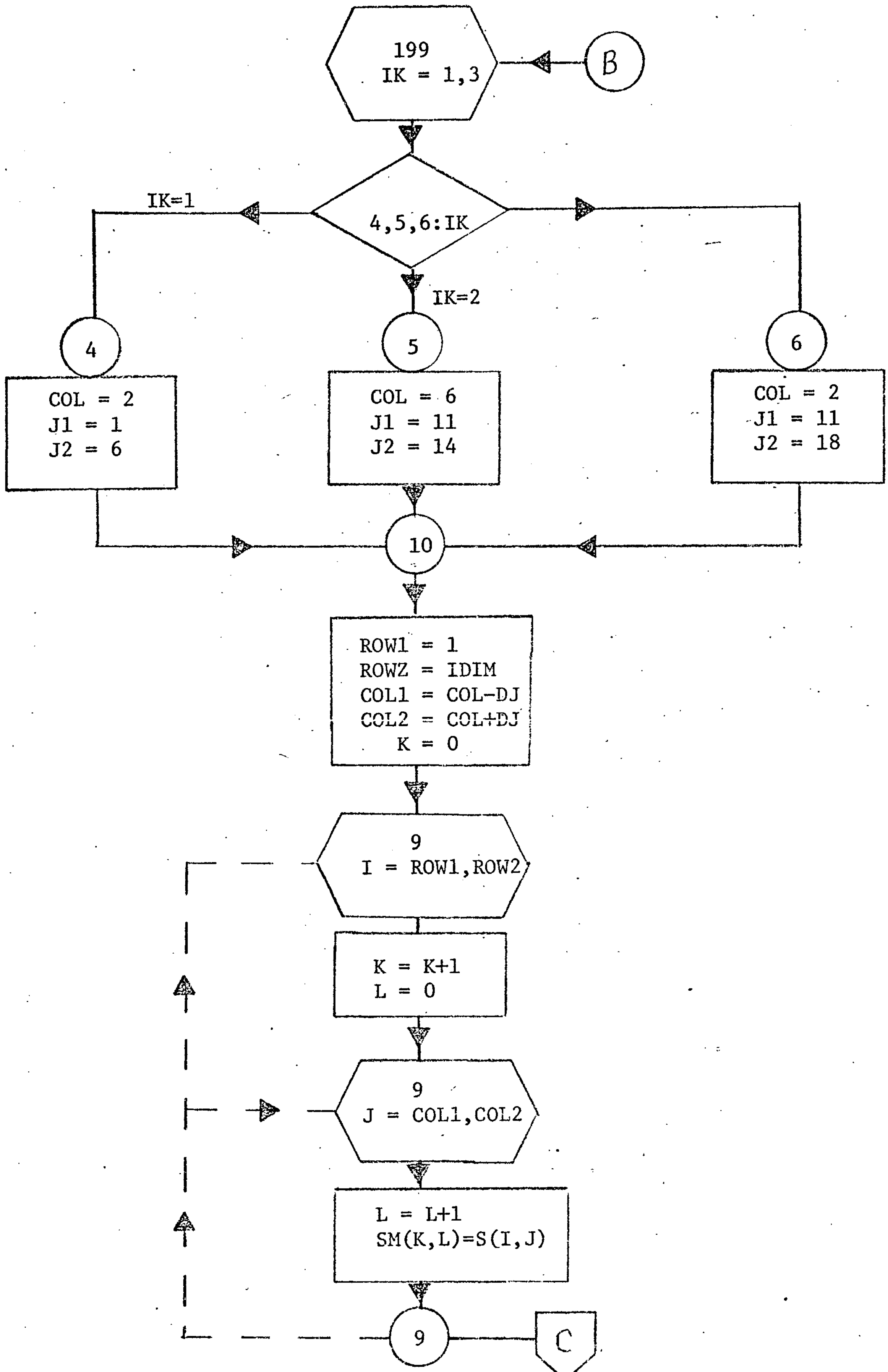


Fig. 8.5.1 Continued

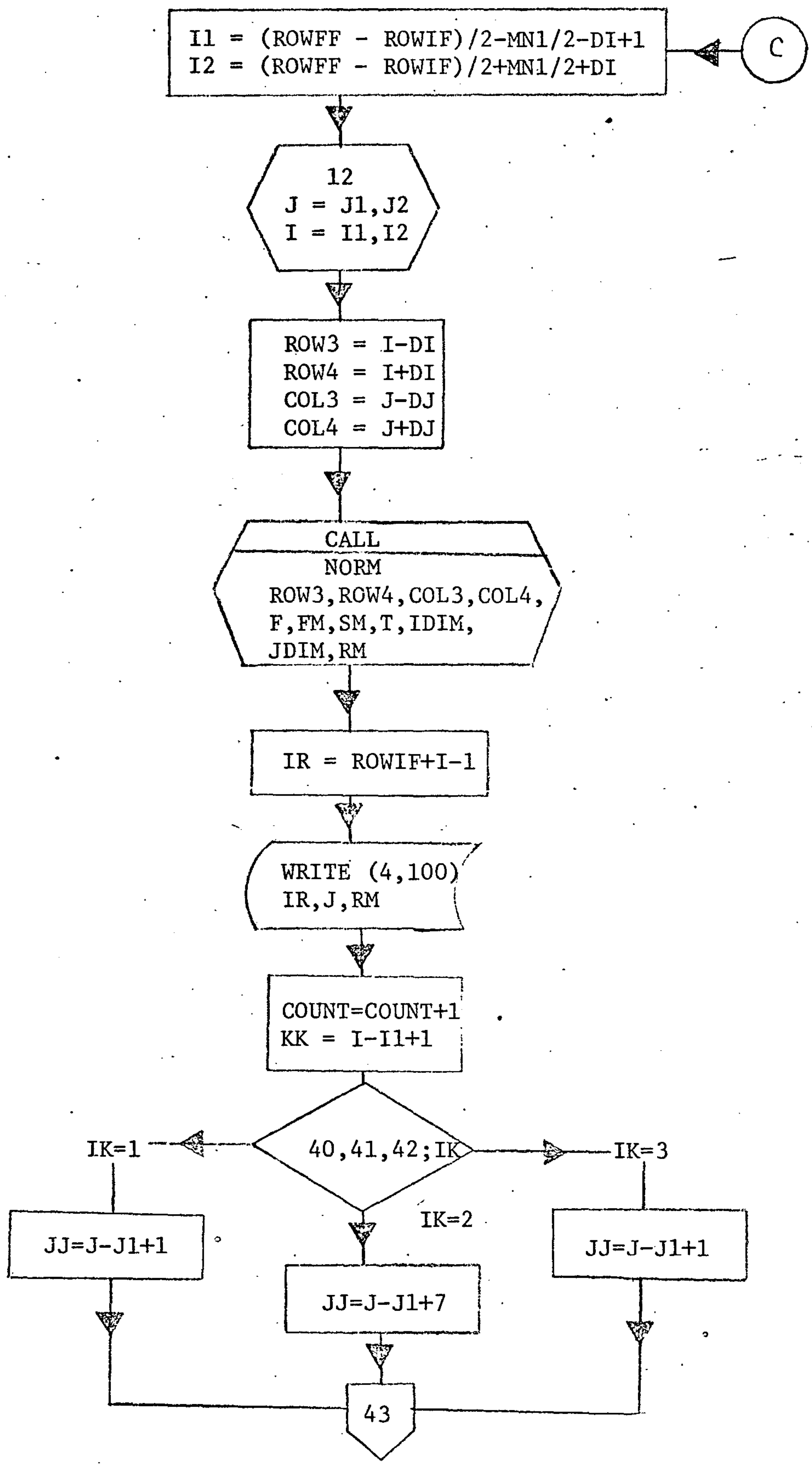


Fig. 8.5.1 Continued

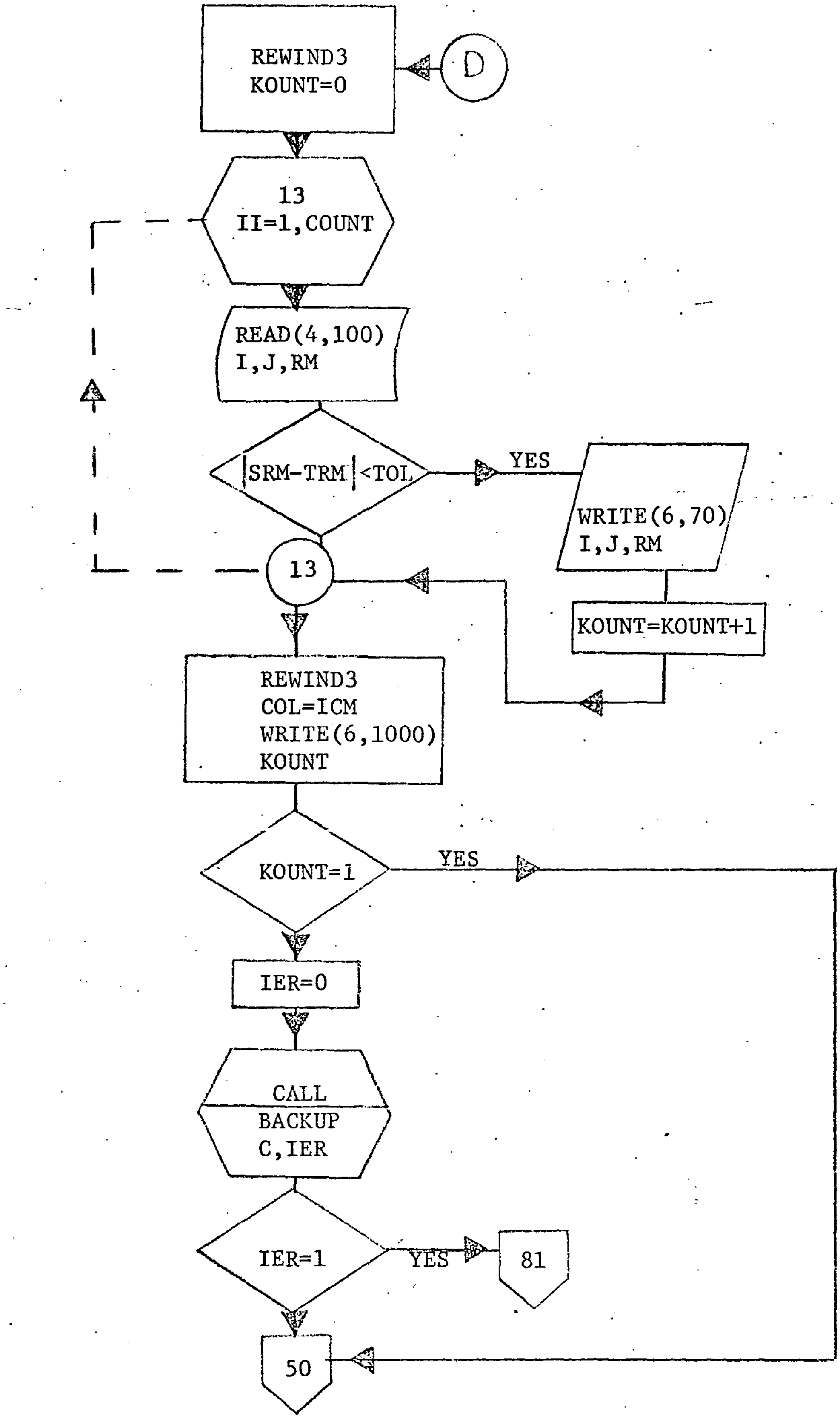


Fig. 8.5.1 Continued

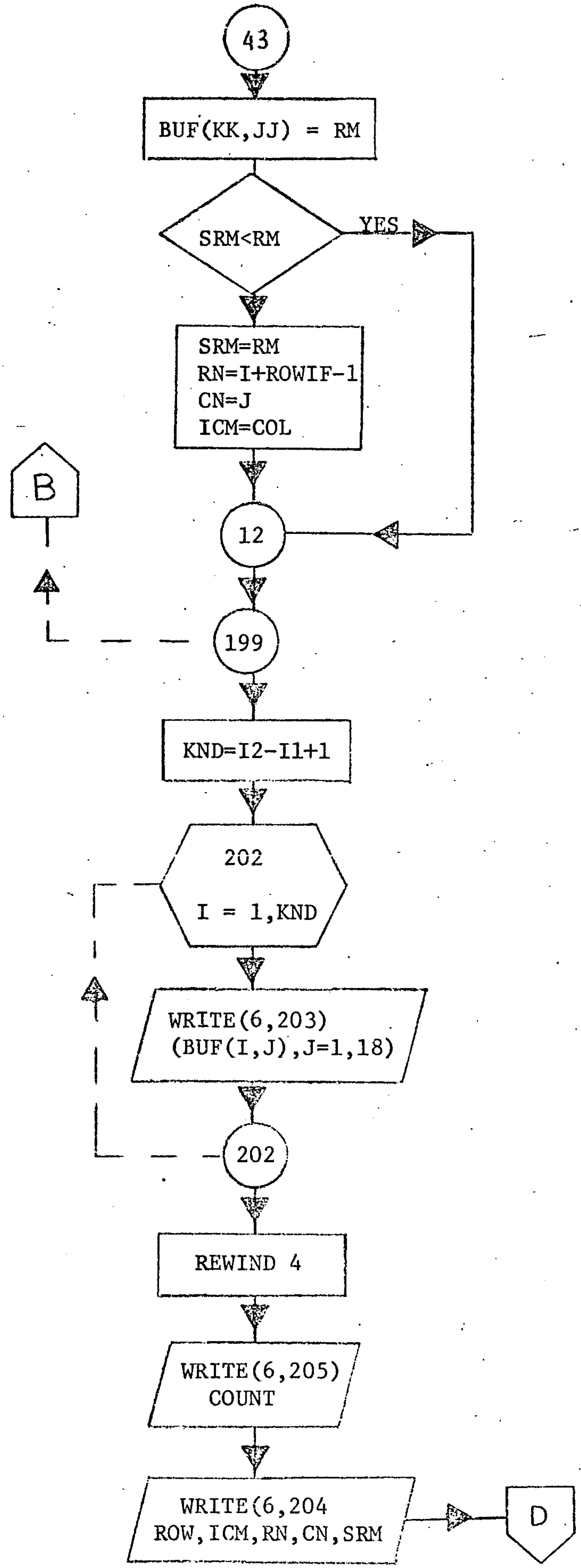


Fig. 8.5.1 Continued

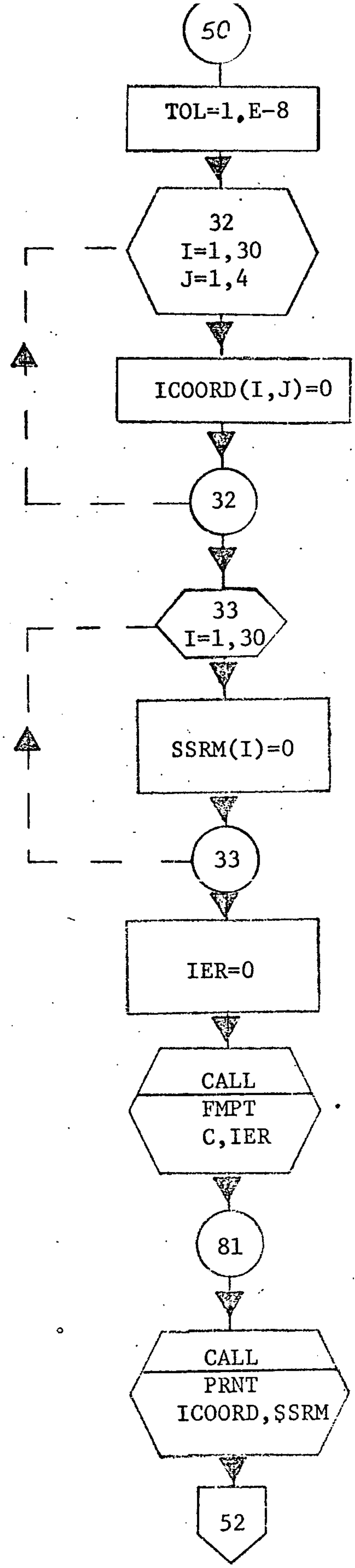


Fig. 8.5.1 Continued

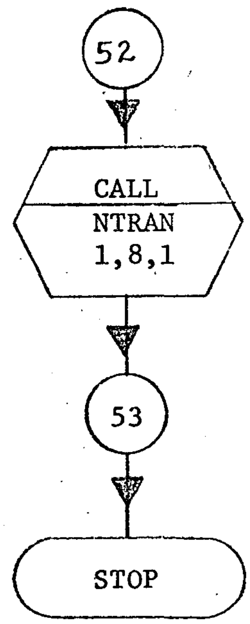


Fig. 8.5.1 CONCLUDED

A - An array used in unpacking data.

BUF - An array used to store values of the norm of FM - SM.

C - An array in which the first 4 and last 4 words of adjacent framelets are stored. The data used by the match filter is contained in C.

CC - An integer which indicates the center row index of C.

CDIM - An integer, the value being the row dimension of C.

CF - An integer, the value being the row dimension of F.

CN - An integer; the column index of the minimum norm at the center of C.

COL - An integer; the column index of the fixed matrix S, which takes on values 2 or 6. Used in attempting to get the minimum norm of FM - SM.

COL1 - An integer, the value describing the initial column of the S matrix.

COL2 - An integer, the value describing the final column of the S matrix.

COL3 - An integer, the value describing the initial column of the F matrix.

COL4 - An integer, the value describing the final column of the F matrix.

COUNT - An integer which determines the number of match points computed at the center of C and the number of norms written on the drum.

CS - An array of smoothed correction factors.

Table 8.5.2 SYMBOL DEFINITION TABLE -- SKINNY

D - A vector equivalenced to the array C.

DI - An integer, the row dimension of the match filter divided by 2.

DJ - An integer, the column dimension of the match filter divided by 2.

DR - An integer (described in routine FMPT).

F - An array of data points which varies over a prescribed area in the right overlap area of the framelet on the left of the two framelets being matched.

FDIM - An integer; the row dimension of F.

FM - A submatrix of F of dimensions of the match filter. Used in calculating the norm of the match.

ICM - The column index of the fixed array where the norm is minimum at the center of C.

ICOORD - An array in which the values of the indices of the fixed and moving arrays at which the minimum norm occurs.

IDIM - The row dimension of the match filter.

IER - A flag which determines if an error has occured in one of the subroutines.

IFB - Flag (described in FMPT)

IND - A difference of indices (described in FMPT)

Table 8.5.2 SYMBOL DEFINITION TABLE - SKINNY -- CONTINUED

INDX - An index counter (described in FMPT)

IR - The row index of the center of the moving array at the match points relative to array C.

IUF - A flag used to determine which array, F or S, will receive unpacked data.

IZ - An index (described in FMPT)

I1,I2 - The initial and final rows of the array F over which the center of FM will vary.

JDIM - The column dimension of the match filter.

J1,J2 - The initial and final columns of the array F over which the center of FM will vary. J1,J2 are dependent upon the columns of the S array.

KK,KND - Two indices which determine the limits of BUF which contains the norms of the match points.

KOUNT - The number of norms that are within TOL of the minimum and that are written on the drum.

KZ - The number of words in a record of tape which contains 4 words for 2068 row. 2068 is the number of rows in the surveyor 3 subframelets.

Table 8.5.2 SYMBOL DEFINITION TABLE - SKINNY -- CONTINUED

KZ1 - KZ+1, the value of the index starting at which the next record from the tape is read into D.

MN1 - The number of rows over which the center of the array FM will vary in attempting to find a match at the center of C.

MN2 - Twice this number is the number of rows over which the center of the array FM will vary in attempting to find other match points after the center of the two sub-framelets have been matched.

MN3 - This number is used by subroutine LKFR in finding a high contrast area for the match algorithm to start near the center of C.

NROW - Used by FMPT to find another high contrast area after a previous match has been determined.

RM - The norm of FM - SM as FM varies over some prescribed portion of F.

RN - The row number of the match point at or near the center of C where the norm is minimum.

ROW - The row number of the fixed array S where LKFR has determined an interesting area exists.

ROWIF ROWFF - The initial and final rows of F which are unpacked to provide data for the match algorithm.

ROWIS ROWFS - The initial and final rows of S which are unpacked to provide data for the match algorithm.

Table 8.5.2 SYMBOL DEFINITION TABLE - SKINNY -- CONTINUED

ROW1 - Describes the row limits of data of S read into SM.
ROW2

ROW3 - Describes the row limits of data of F read into FM.
ROW4

S - The fixed matrix of data points in the left overlap area of the framelet on the right of the two framelets being matched.

SM - A submatrix of S of dimensions of the match filter. Used in calculating the norm of the match.

SRM - The minimum of the norms i.e. the norm of the match point.

SSRM - An array which contains the values of SRM for each match point found.

T - The product of the dimensions of the match filter. Used in calculating the norm.

TOL - Used in determining how many points are candidates for the match.

TRM - Norm plus the BACKUP norm (described in BACKUP).

Table 8.5.2 SYMBOL DEFINITION TABLE - SKINNY -- (CONCLUDED)

8.5.3.1 Description of subroutine LKFR

IDENTIFICATION

Name	--	LKFR
Author/Date	--	R.L. Wendt/June 1970
Organization	--	ASR
Machine Identification	--	UNIVAC 1108
Source Language	--	FORTRAN V

PURPOSE

This program uses a variable threshold pattern recognition scheme to locate areas in which there are some features of relatively more interest than in nearly areas, the purpose being to give SKINNY an area containing the most possible detail to aid the matching process.

USAGE

Calling sequence:

CALL LKFR (C, NS, NF, NL, ISP)

The calling variables are described in Table 8.5.3.1.1 (SYMBOL DEFINITION TABLE).

C	The array of data.
MTX	An array containing moon scene data unpacked from columns 2 and 6 in the overlap region of the fixed array S.
IDUM	A dummy array used to fill out the calling sequence of UNPACK.
IOUT1 IOUT2	Two vectors, one for column 2 and 6 which are set to 1 or 0 depending on whether the data is simultaneously greater than LARGE or less than SMALL in each case.
NS,NF	Two row numbers over which an interesting area is to be found.
NL	IDIM, the number provides the number of rows over which a certain product is to be maximized.
ISP	The row number of an interesting area.
LARGE,SMALL	Two integers which represent values for which 20% of the data is greater than and less than.
FK,SK	The sum over NL rows of IOUT1, IOUT2.
KS	The product of FK and SK.
MAXKS	The maximum value of KS.

Table 8.5.3.1.1. SYMBOL DEFINITION TABLE, LKFR.

METHOD:

This subroutine finds a high contrast area in the data of the overlap region of the framelet on the right of the two being matched. Columns two and six of the data are unpacked and if the elements in columns two and six are simultaneously larger than some number termed LARGE (or simultaneously smaller than some number termed SMALL), then a matrix for LARGE (respectively SMALL) is set to 1, otherwise to 0. The area deemed "interesting" is that area for which the product of the sum over IDIM rows, the dimension of the match filter of the elements in the two arrays is maximal.

The subroutine is provided with two row numbers NS,NF over which the routine is to find the most "interesting" area. The variables NS,NF do not have to be ordered in magnitude in the calling sequence of LKFR. The variables NS,NF only have to represent two row numbers over which the routine is to look for an "interesting" area.

SUBROUTINE REQUIREMENTS:

UNPACK

LGSM

DETAILED FLOW CHART:

See Fig. 8.5.3.1.2

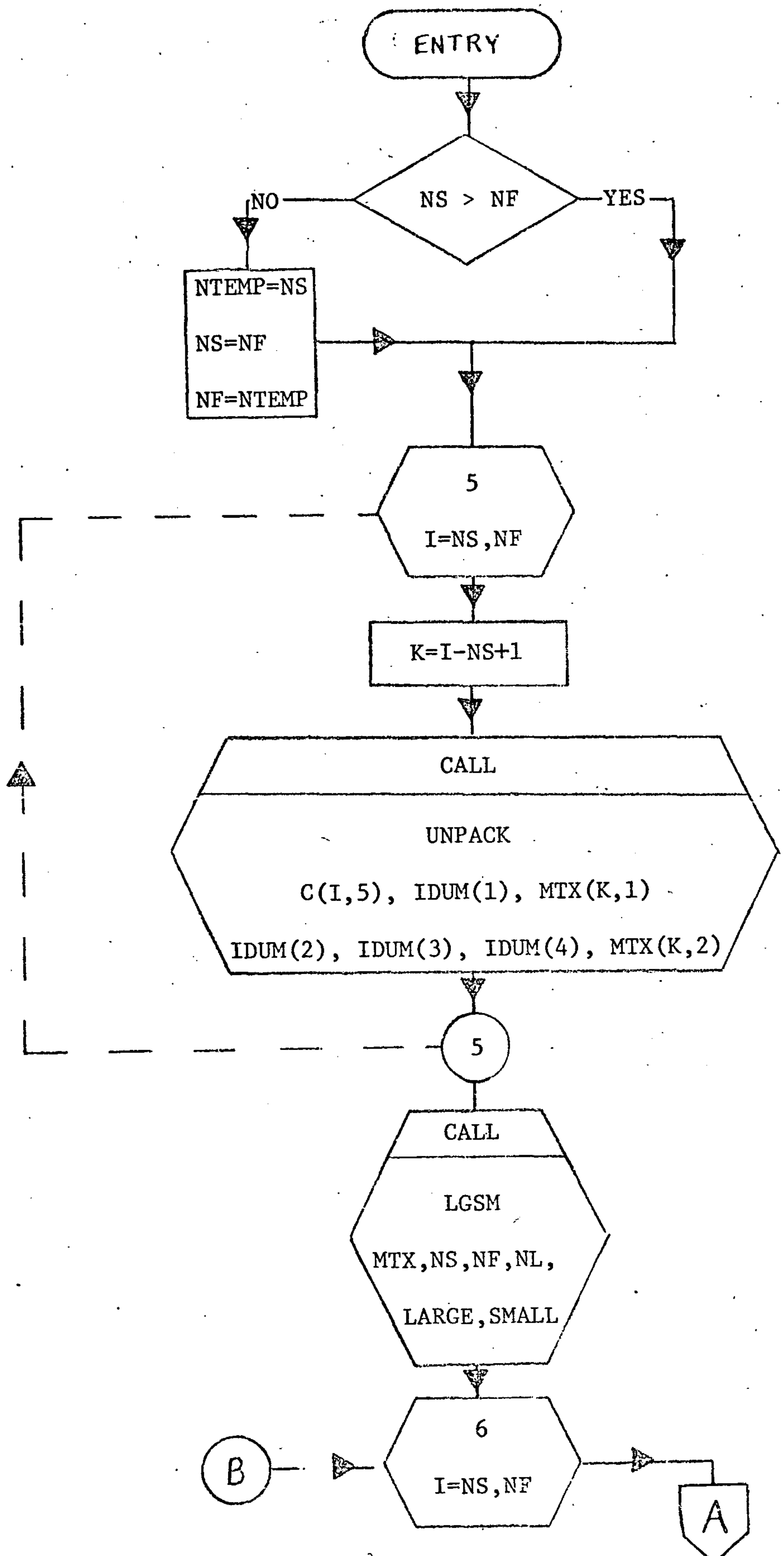


Fig. 8.5.3.1.2 DETAILED FLOW CHART - LKFR

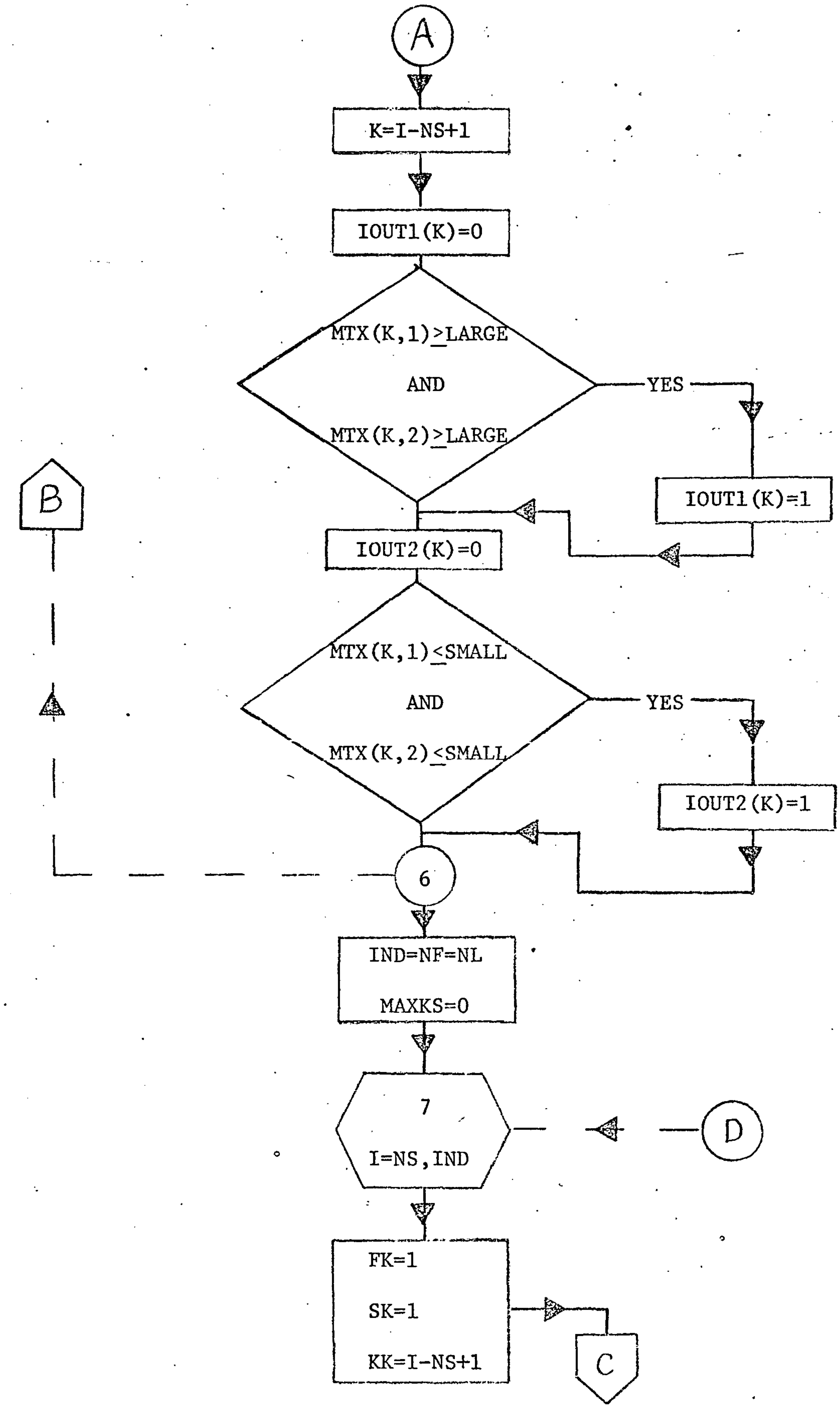


Fig. 8.5.3.1.2 DETAILED FLOW CHART - LKFR -- CONTINUED

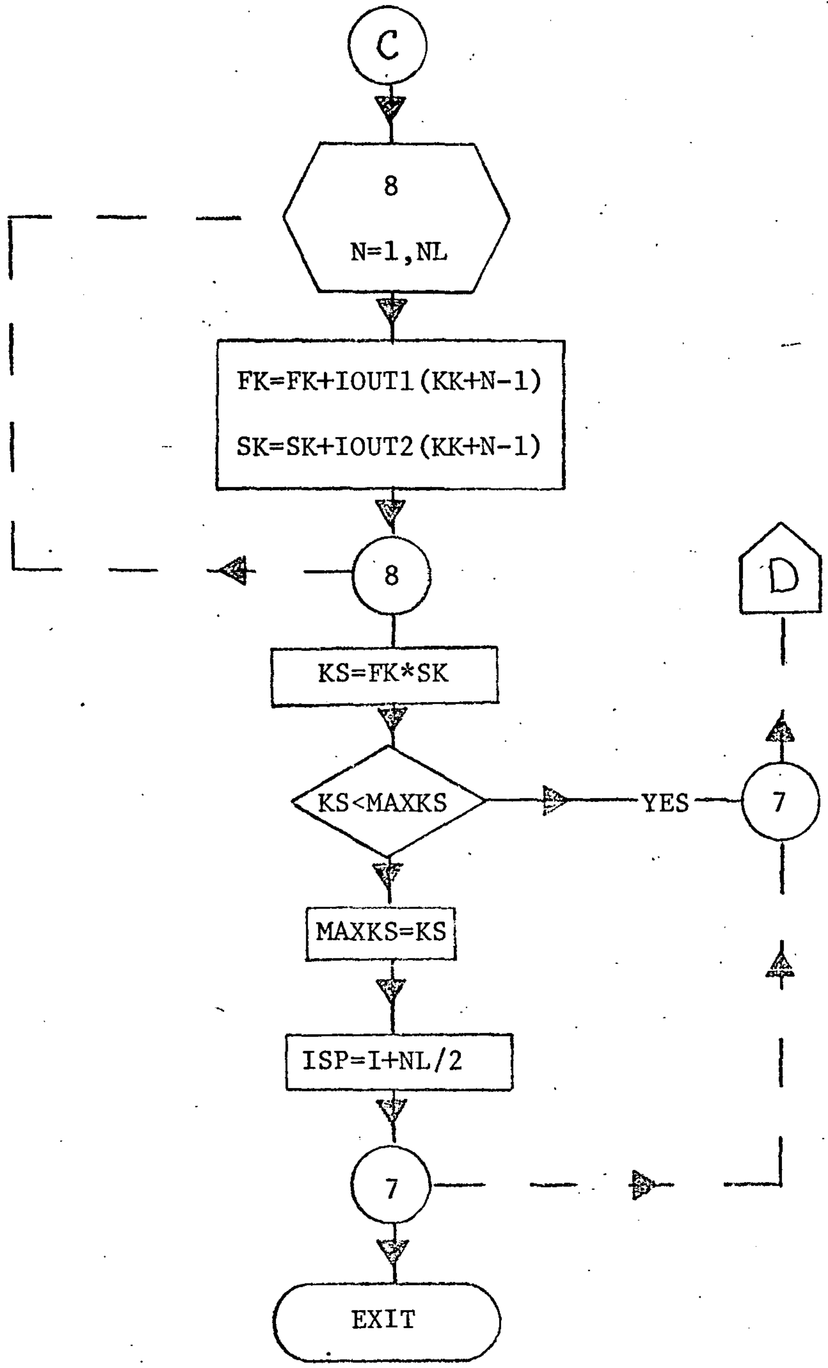


Fig. 8.5.3.1.2 DETAILED FLOW CHART - LKFR -- CONCLUDED

8.5.3.2 Documentation of subroutine LGSM

IDENTIFICATION

Name -- LKFR

Author/Date -- R.L. Wendt/June 1970

Organization -- ASR

Machine Identification -- UNIVAC 1108

Source Language -- FORTRAN V

PURPOSE/METHOD

This routine provides the numbers LARGE and SMALL for LKFR. SMALL is defined to be the number of elements in array MTX which are greater than or equal to 20% of the difference between the numbers NF-NS, which are the rows over which an interesting area is sought. The number large is defined similarly except the counting of the numbers begins with 63 the largest of all possible values an element in the data can have.

This process provides a variable threshold for the pattern recognition process in LKFR.

USAGE

Calling sequence:

CALL LGSM (MTX, NS, NF, LARGE, SMALL)

DESCRIPTION OF VARIABLES

TFP - Two fifths of the difference between NF, NS.

NTEMP - An integer which counts the number of data points which have value equal to an index starting at 1 for SMALL and 63 for LARGE.

C.2

DETAILED FLOW CHART

See Fig. 8.5.3.2.1.

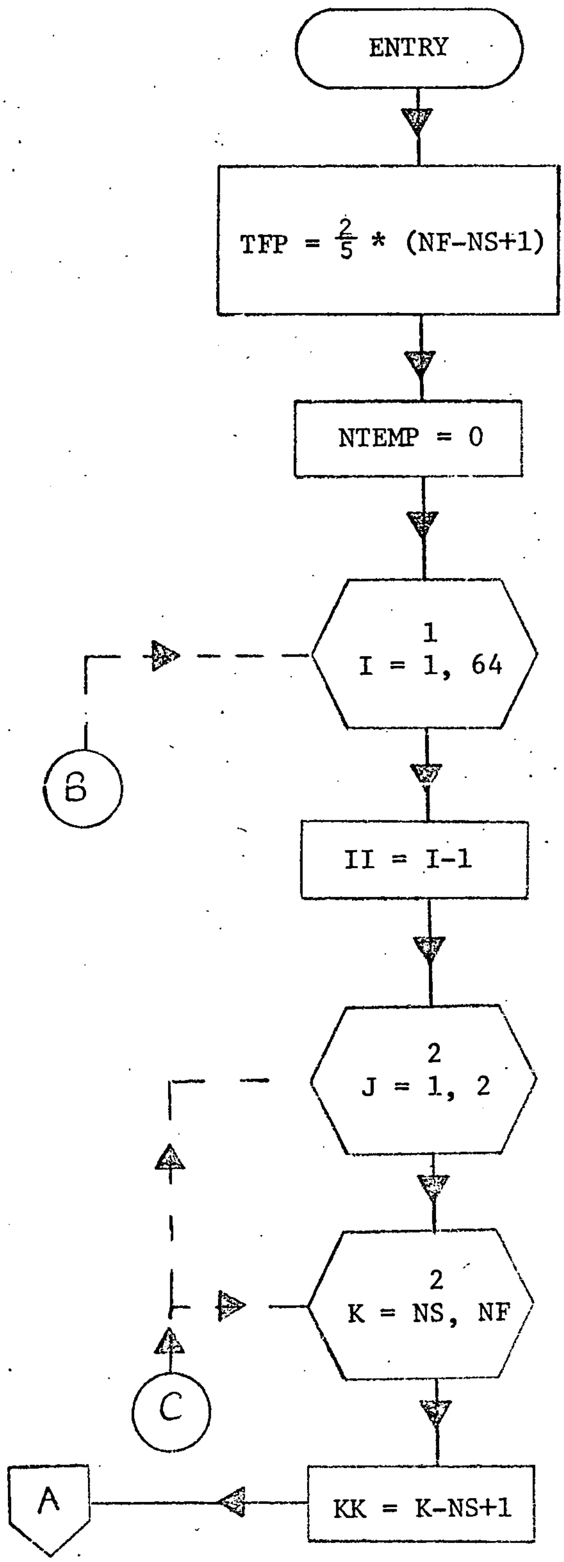


Fig. 8.5.3.2.1 DETAILED FLOW CHART - LGSM

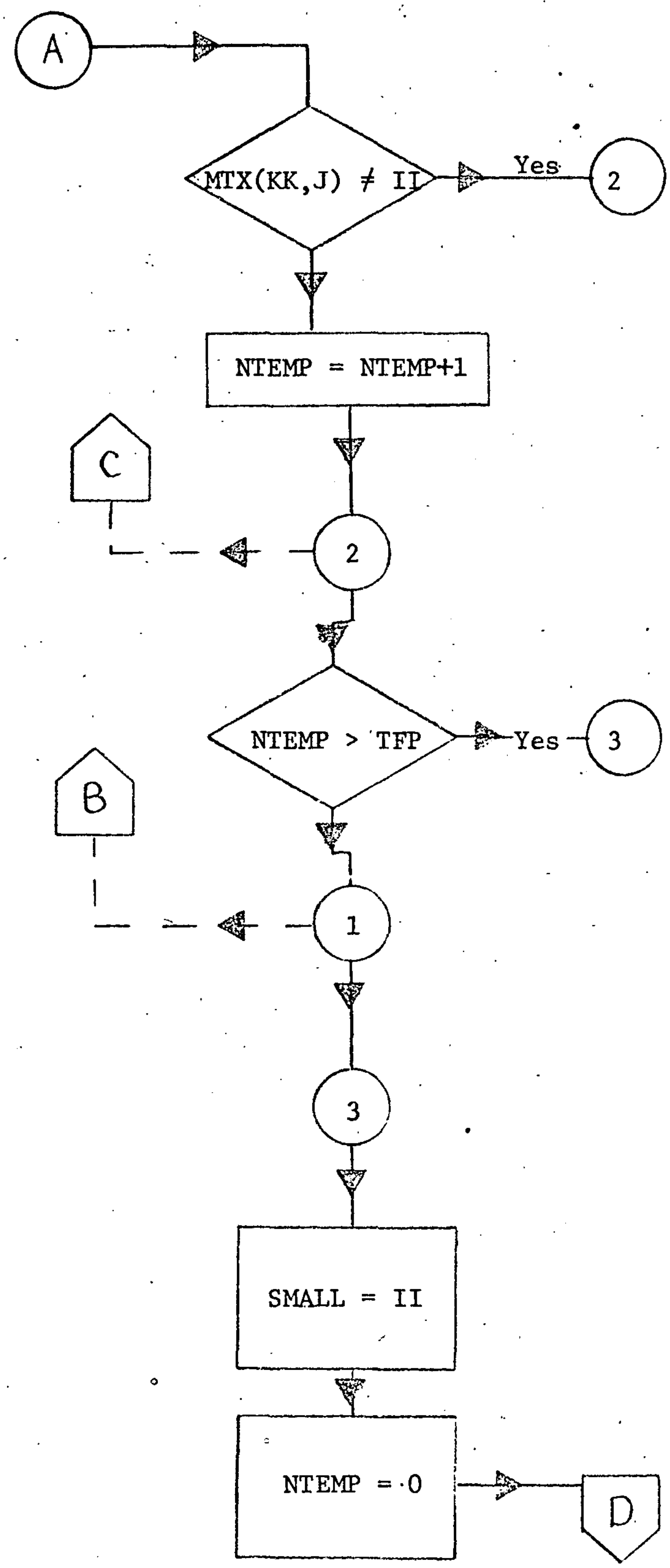


Fig. 8.5.3.2.1 DETAILED FLOW CHART - LGSM (Continued)

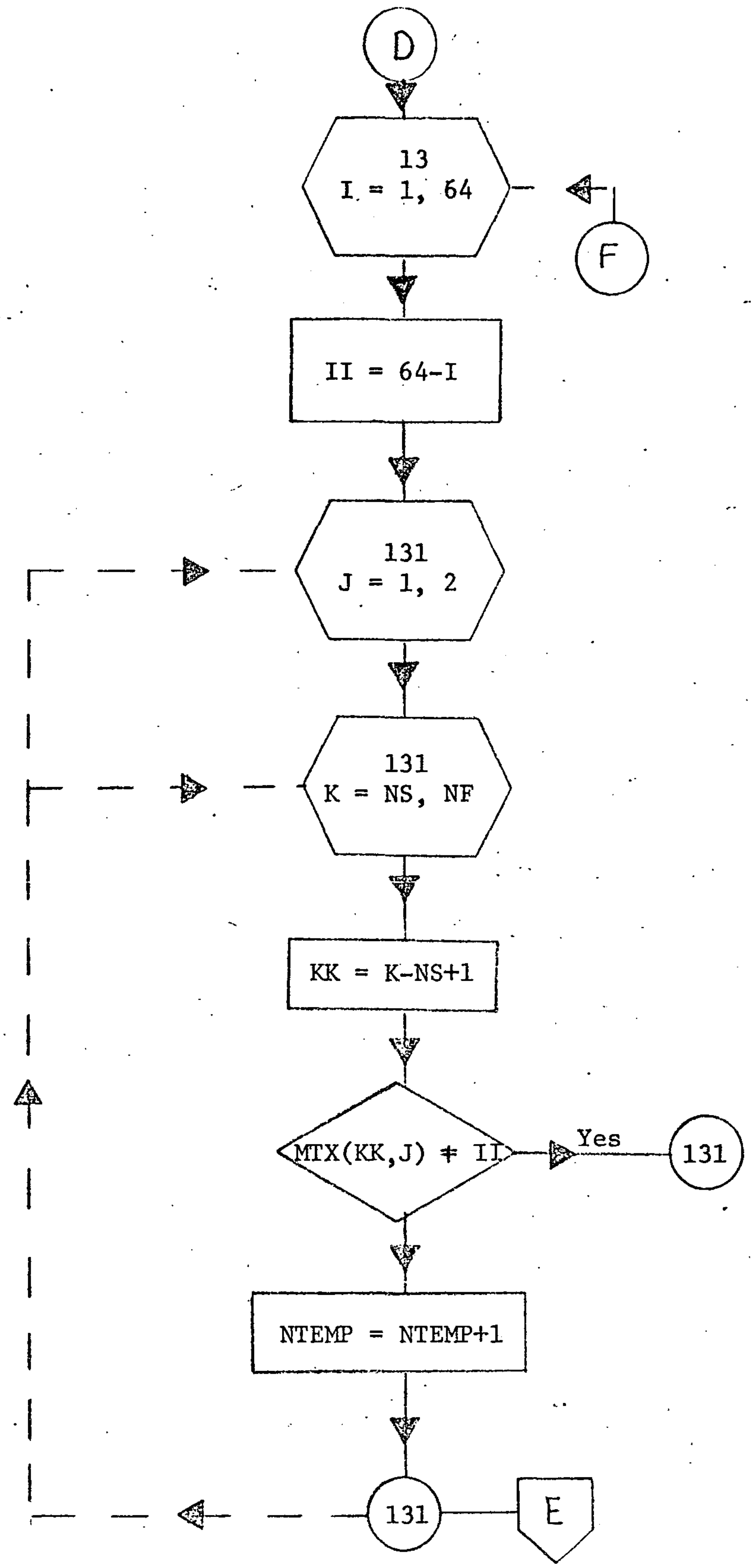


Fig. 8.5.3.2.1 DETAILED FLOW CHART - LGSM (Continued)

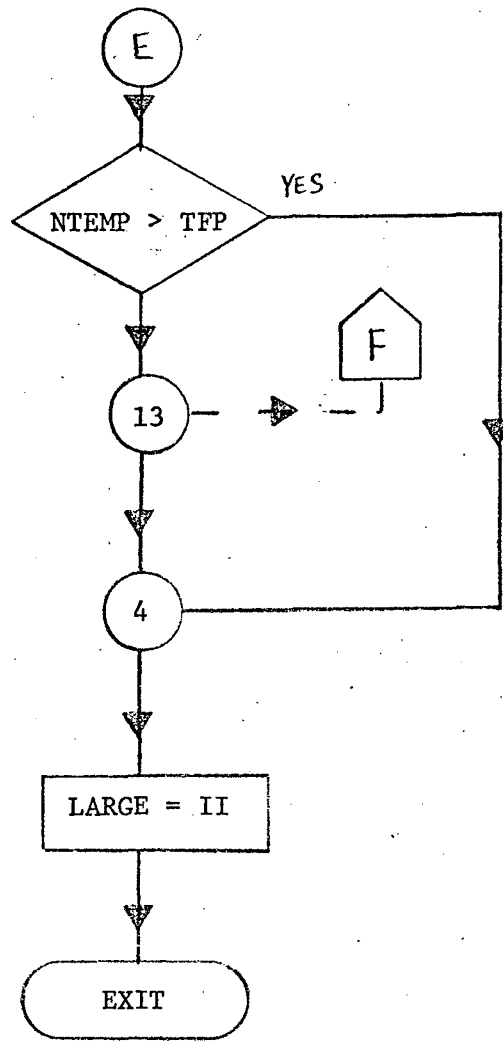


Fig. 8.5.3.2.1 DETAILED FLOW CHART - LGSM (Concluded)

8.5.3.3 Description of subroutine NORM

IDENTIFICATION

Name	--	NORM
Author/Date	--	R.L. Wendt/June 1970
Organization	--	ASR
Machine Identification	--	UNIVAC 1108
Source Language	--	FORTRAN V

PURPOSE/METHOD

This subroutine calculates the norm of FM-SM given some submatrix of F defined by ROW3, ROW4, COL3, COL4. The norm used is a constant times the Euclidean 2 norm. The constant is essentially used to give the numbers an order of magnitude which makes the print of the norms easier to read i.e. values about .10-.50 region.

USAGE

Calling sequence:

CALL NORM (ROW3 , ROW4, COL3, COL4, F, FM, SM, T, IDIM, JDIM, RM)

The calling variables are described in Table 8.5.2.

In addition the following new variable is used by NORM:

SUM 2 The square root of the sum of the squares of the difference between the elements of FM and SM divided by the product of the row and column dimension. i.e. a constant times $\|\ \|_2$.

DETAILED FLOW CHART

See Fig. 8.5.3.3.1

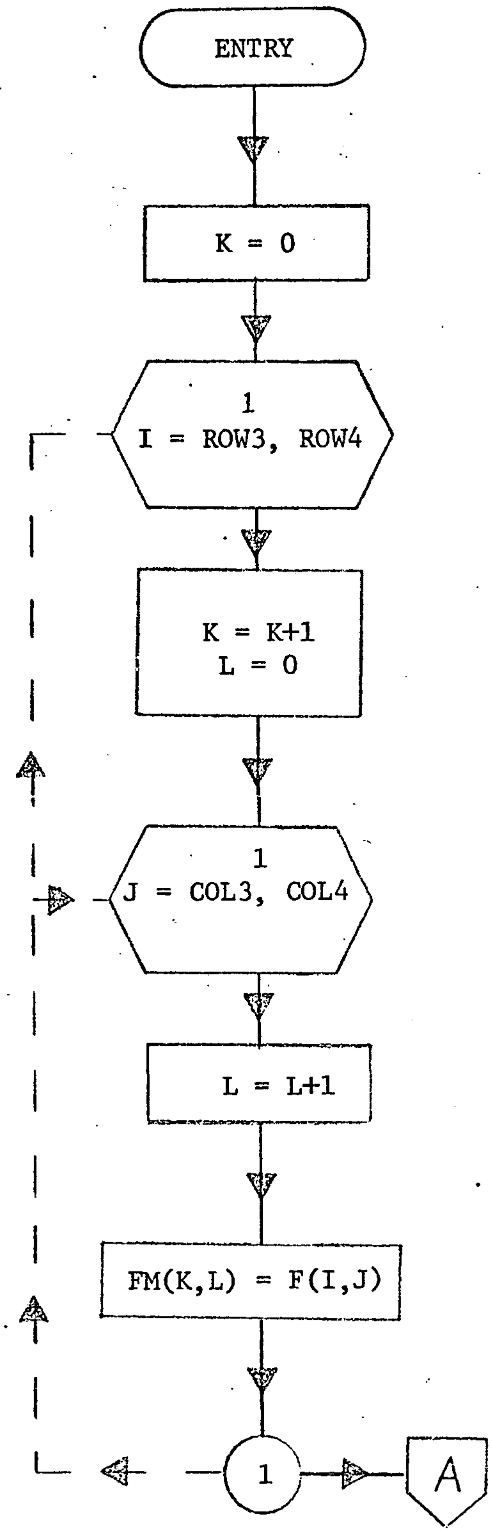


Fig. 8.5.3.3.1 DETAILED FLOW CHART -- NORM

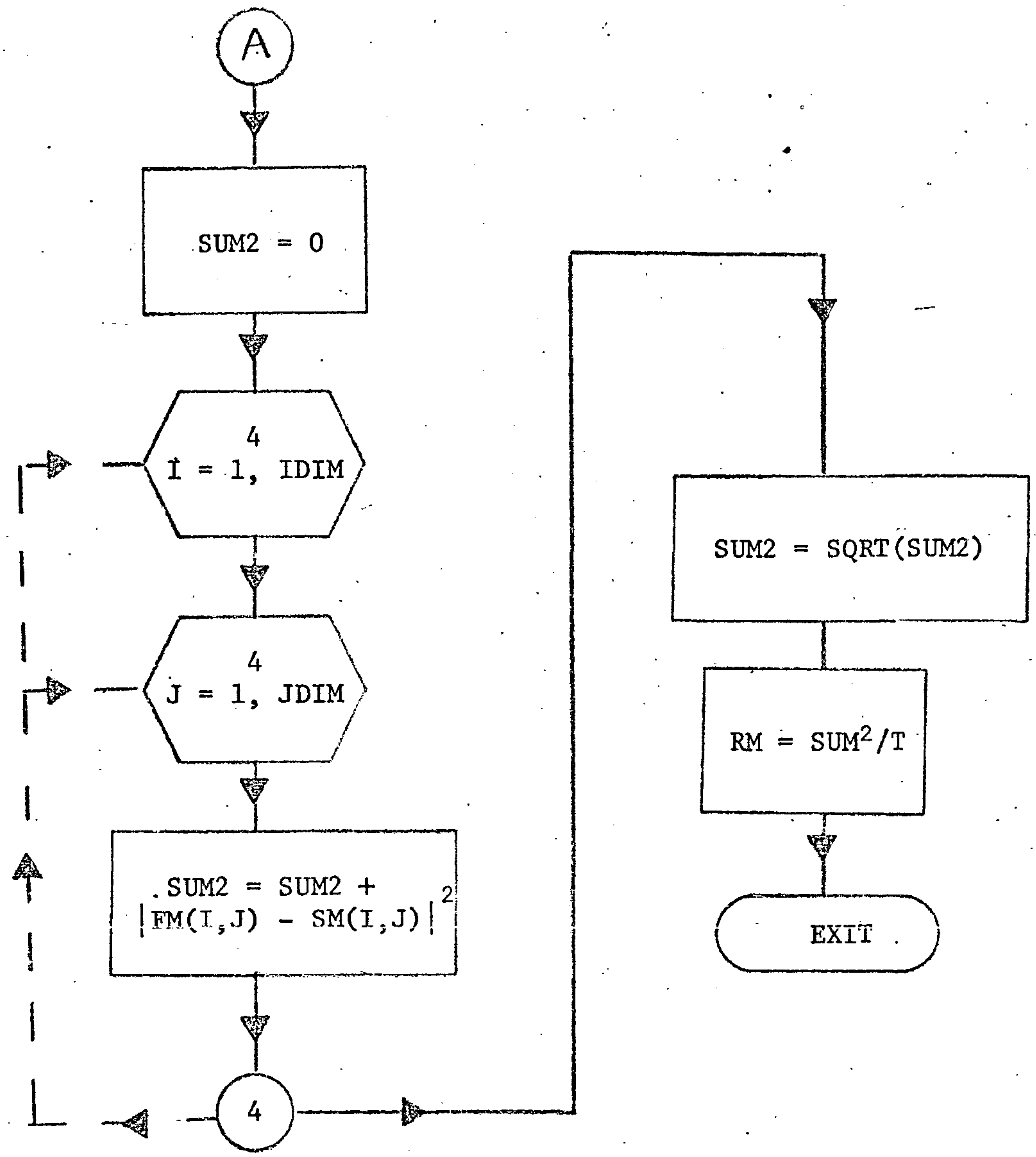


Fig. 8.5.3.3.1 DETAILED FLOW CHART - NORM
(Concluded)

8.5.3.4 Documentation of UNPSTR

IDENTIFICATION:

Name	--UNPSTR
Author/Date	--R.L. Wendt/June 1970
Organization	--ASR
Machine Identification	--UNIVAC 1108
Source Language	--FORTRAN V

PURPOSE/METHOD

This routine unpacks and stores data from the array C into the arrays F or S depending on the value of IUF. The smoothed or unsmoothed corrections factors can be applied at this time. Array C is for Surveyor-3 2068 rows by 8 columns. The data for the framelet on the left is in the 1st four columns and the data for the framelet on the right is in the last four columns. In selecting the correction factors the arrangement of the data in C has to be remembered. The first 2 words of the last 4 words are unpacked for S since the match filter can't be very wide because of the drummarks. The last 3 words of the first 4 words are unpacked for F since the overlap is well within these limits. The data is placed in F and S left justified with respect to the columns.

If unsmoothed correction factors are used, the reciprocal should serve as multiplicative factor. If smoothed and normalized correction factors are used, these should serve as multiplicative factors.

USAGE

Calling sequence

CALL UNPSTR (C, A, F, S, CS, ROW1, ROW2)

DESCRIPTION OF VARIABLES:

C - The array of data of the overlap region of both sub-framelets.

A - An array used to unpack data.

F - The moving array located in the left framelet of the two being matched.

S - The fixed array located in the right framelet of the two being matched.

CS - An array of correction factors.

ROW1 - The first row to be unpacked for either F or S.

ROW2 - The final row to be unpacked for either F or S.

IUF - A flag which determines which array F or S will receive unpacked data.

DETAILED FLOW CHART:

See Fig. 8.5.3.4.1.

SUBROUTINE REQUIREMENTS:

UNPACK

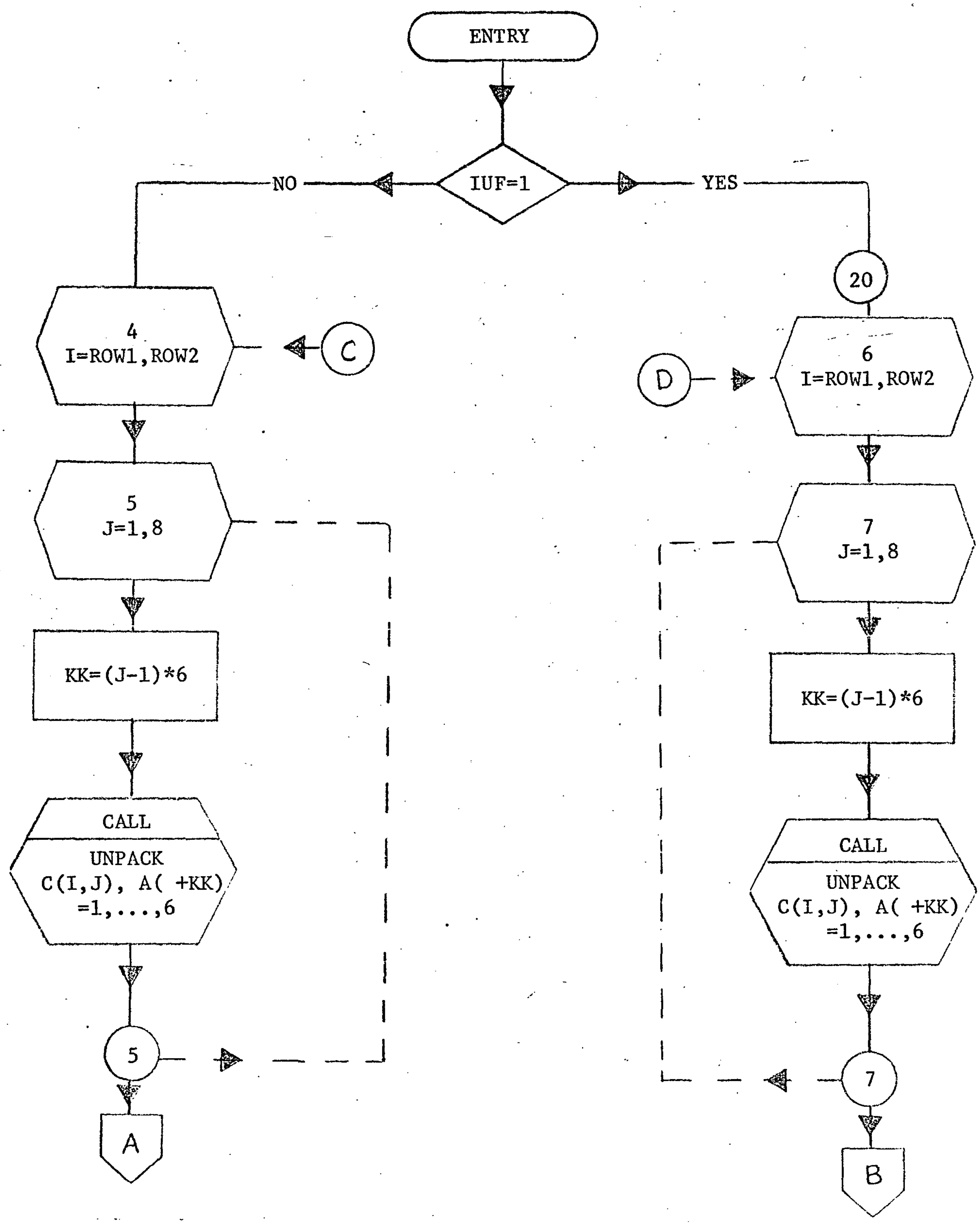


Fig. 8.5.3.4.1. DETAILED FLOW CHART - UNPSTR

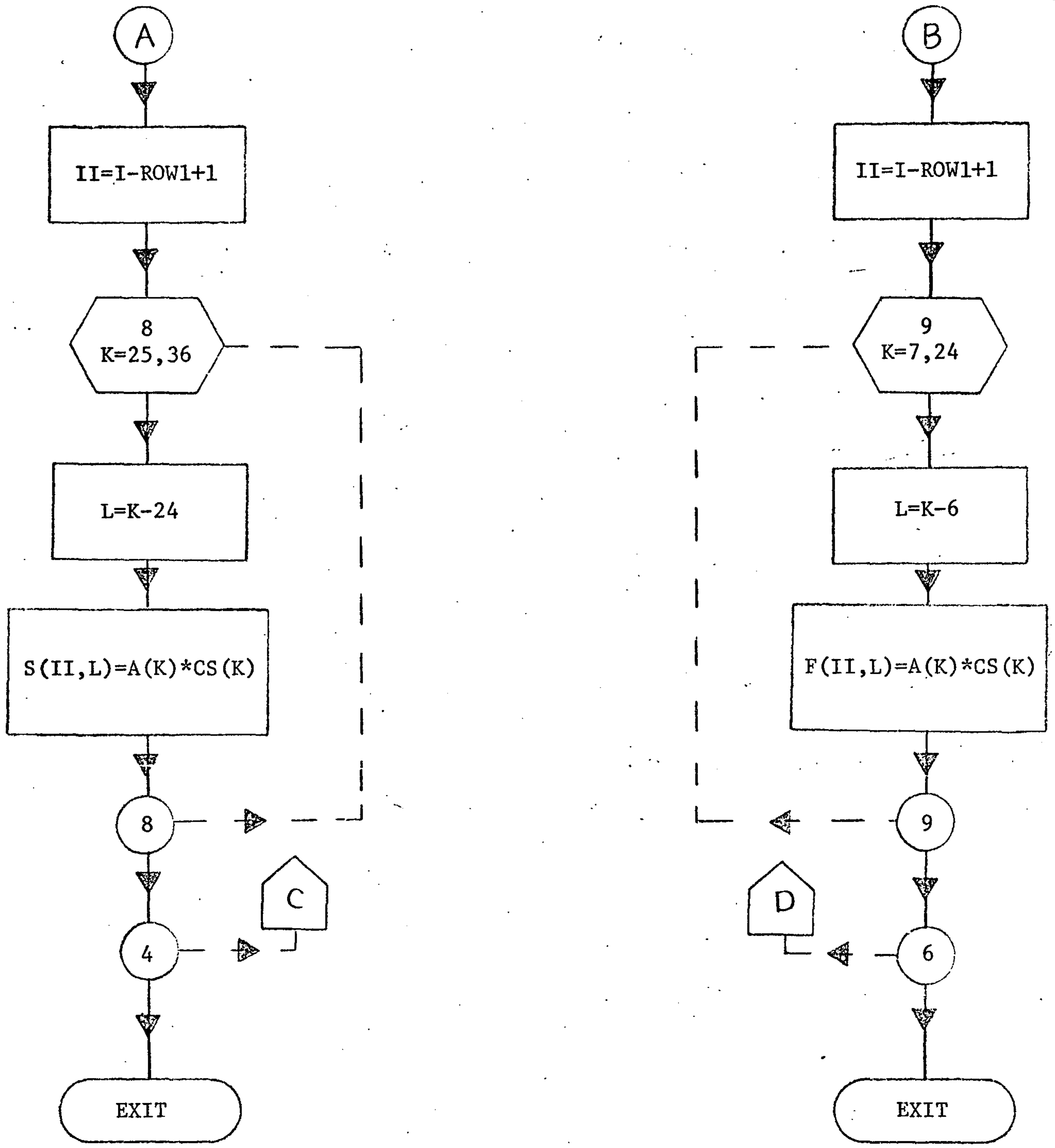


Fig. 8.5.3.4.1. DETAILED FLOW CHART - UNPSTR -- CONCLUDED

8.5.3.5. Description of subroutine BACKUP

IDENTIFICATION:

Name	--BACKUP
Author/Date	--R.L. Wendt/June 1970
Organization	--ASR
Machine Identification	--UNIVAC 1108
Source Language	--FORTRAN V

PURPOSE/METHOD:

This routine provides a means of selecting a minimum of the match points when several points have been found to be within TOL of the strict minimum at or near the center.

The basic idea is to "back up" the fixed array so its last row coincides with its original first row which amounts to moving it IDIM rows, the row dimension of the match filter. The candidates for the minimum i.e. the row, column and norm are read from the drum, a sub array is unpacked and stored from the original data, a new norm is calculated, the sum of the original norm and the norm from "backing up" is written on the drum. When this process is completed the sum of the norms are read from the drum and compared against the new strict minimum. If there are any candidates, the minimum is determined not unique and the process is terminated. If this is not the case a test is made to determine if the new strict minimum occurred on any edge. If so again the process is terminated. A check is made to determine when the strict minimum is read from the drum. This is not ruled a case of when the minimum is not unique.

USAGE:

Calling sequence:

CALL BACKUP (C, IER)

DESCRIPTION OF VARIABLES:

The variables used in BACKUP are the same as those used in SKINNY with the possible exception of TRM. See Table 8.5.2., plus:

TRM - The norm of FM-SM. The sum of the norm achieved by SKINNY and the norm obtained when the routine moved the candidate for the minimum. Minimization is attempted over the sum of the original norm from SKINNY and the new norm from BACKUP.

IER - An error flag set to 1 if certain conditions are met in BACKUP; i.e., the minimum occurred on any edge of the region under consideration.

SUBROUTINE REQUIREMENTS:

NORM, UNPSTR

GENERAL FLOW CHART:

See Fig. 8.5.3.5.1.

DETAILED FLOW CHART:

See Fig. 8.5.3.5.2.

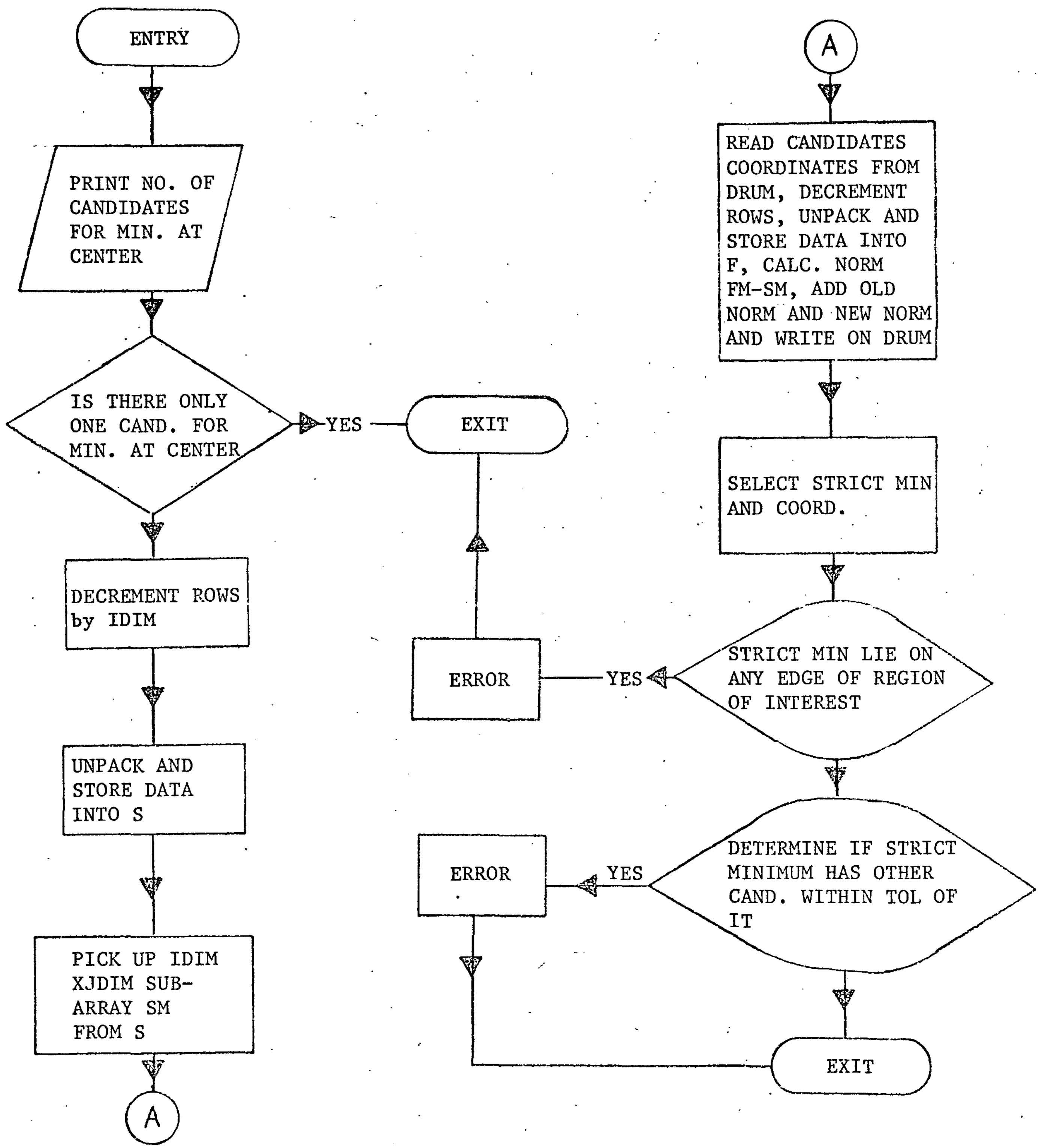


Fig. 8.5.3.5.1. GENERAL FLOW CHART - BACKUP

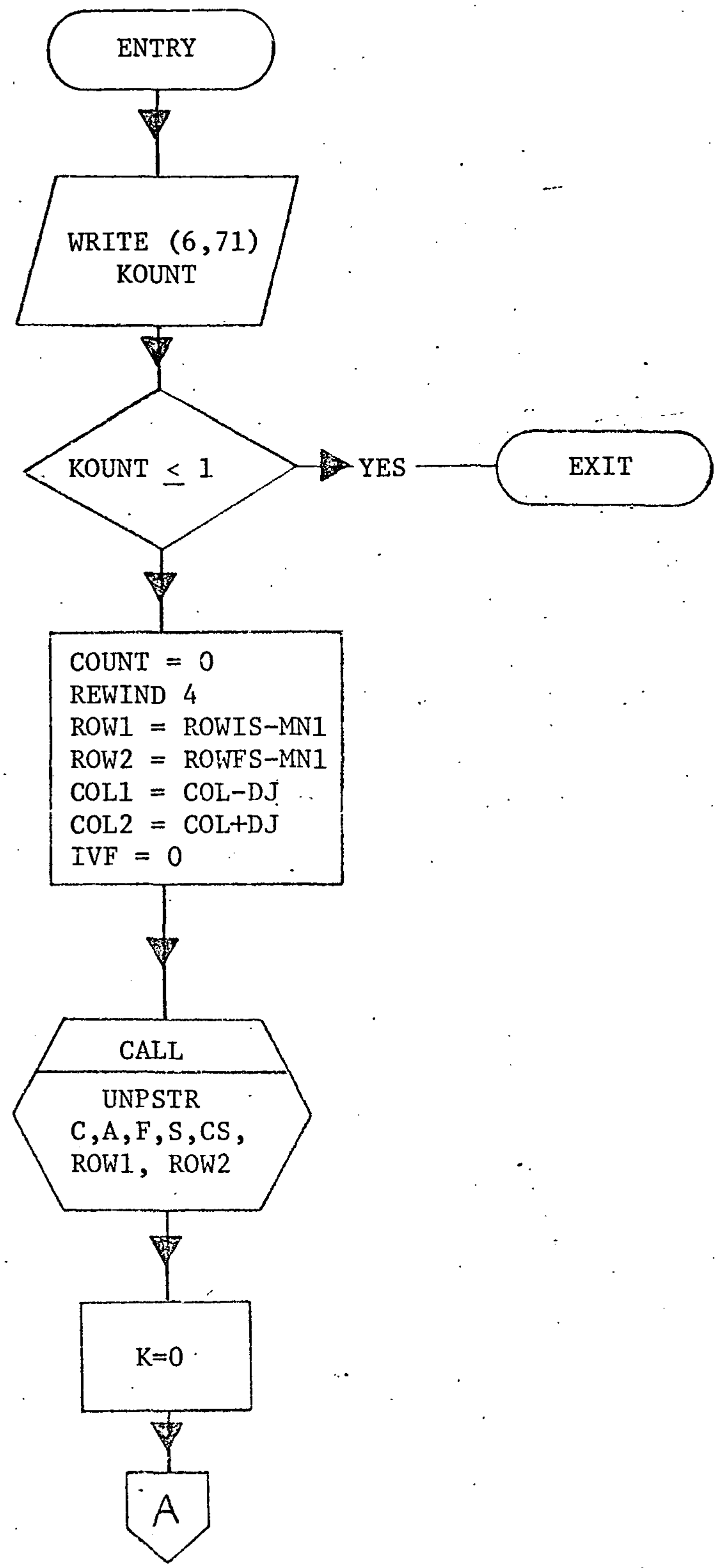


Fig. 8.5.3.5.2 DETAILED FLOW CHART - BACKUP

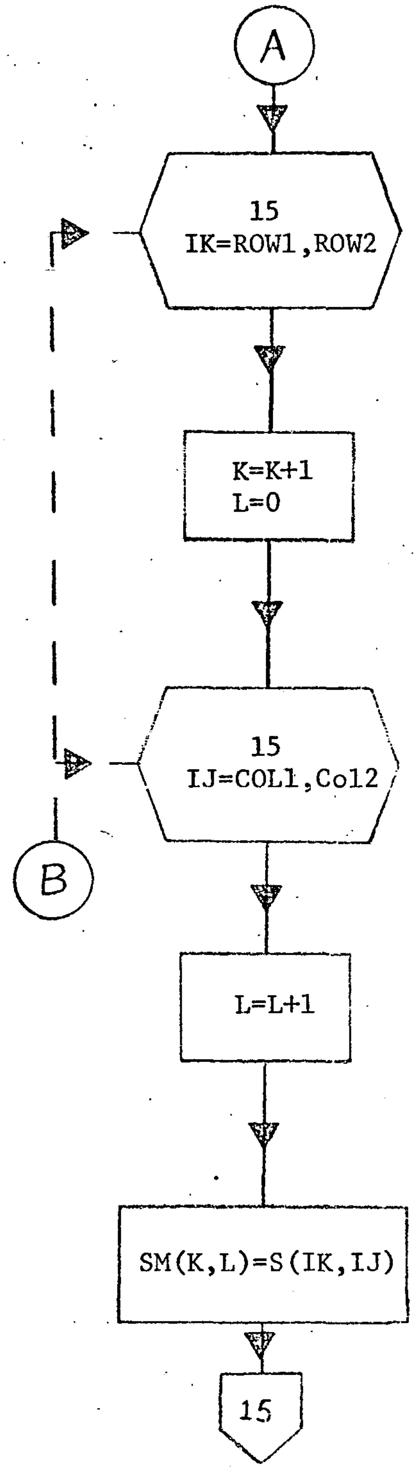


Fig. 8.5.3.5.2 DETAILED FLOW CHART - BACKUP -- CONTINUED

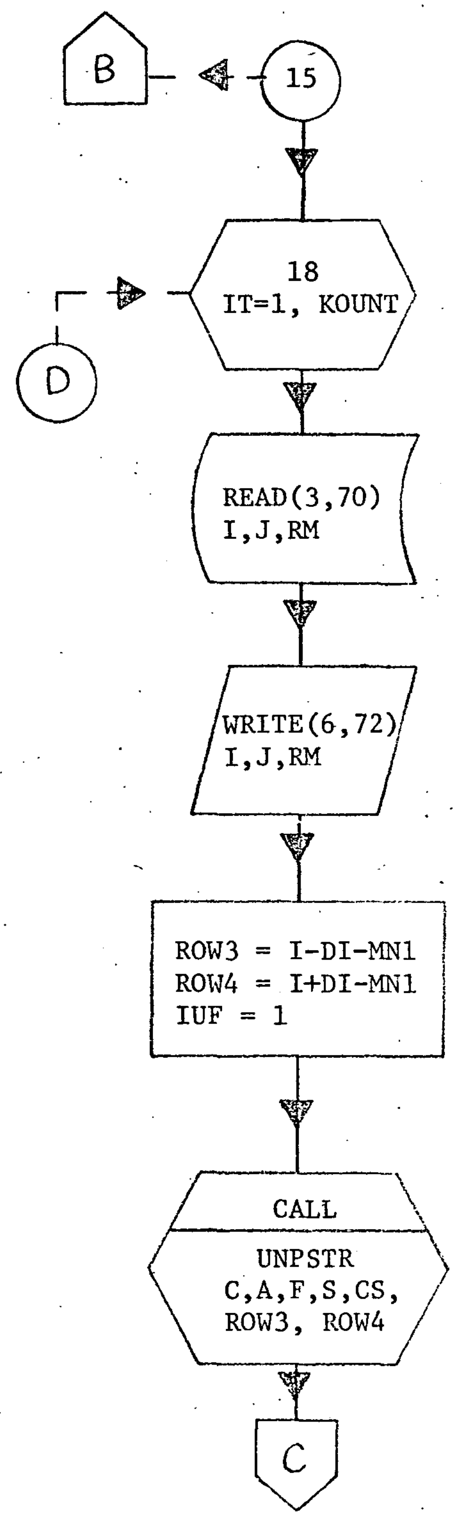


Fig. 8.5.3.5.2 DETAILED FLOW CHART - BACKUP -- CONTINUED

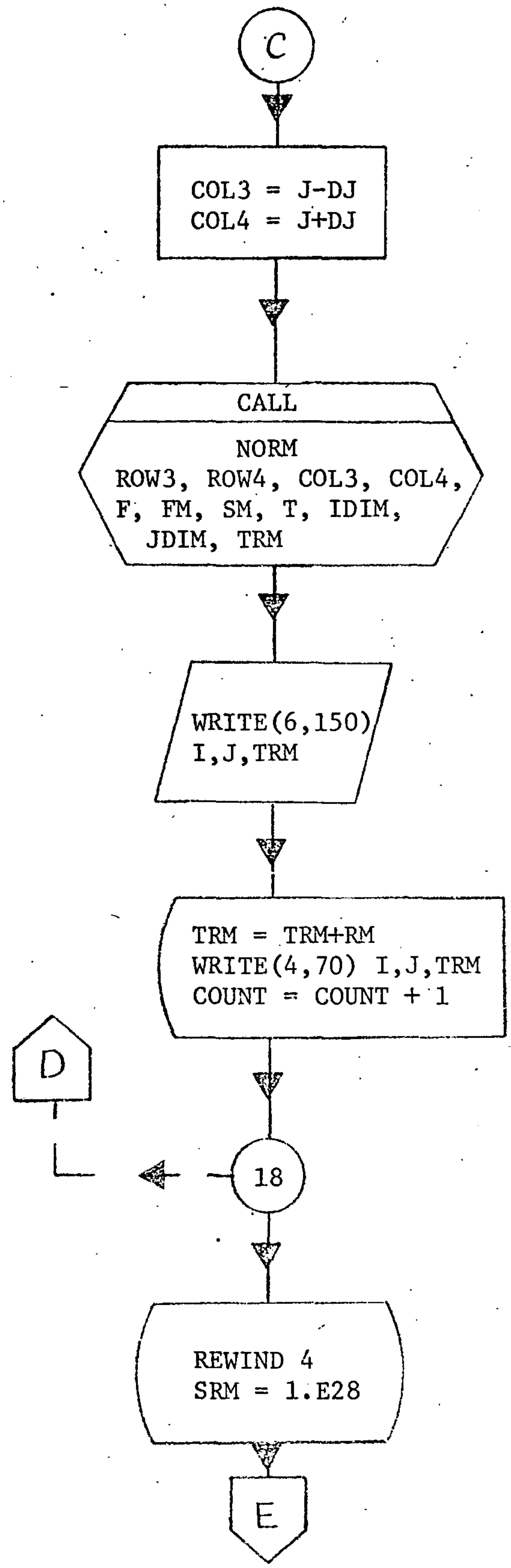


Fig. 8.5.3.5.2. DETAILED FLOW CHART - BACKUP -- CONTINUED

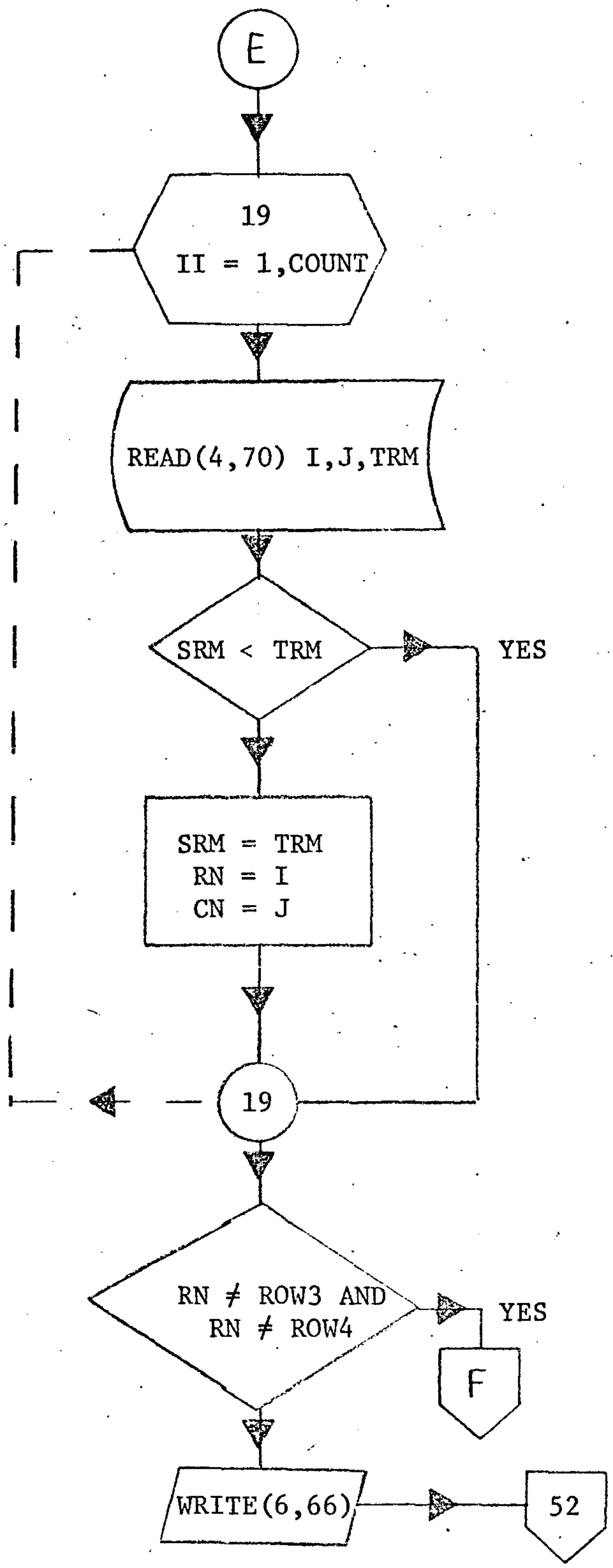


Fig. 8.5.3.5.2 DETAILED FLOW CHART - BACKUP -- CONTINUED

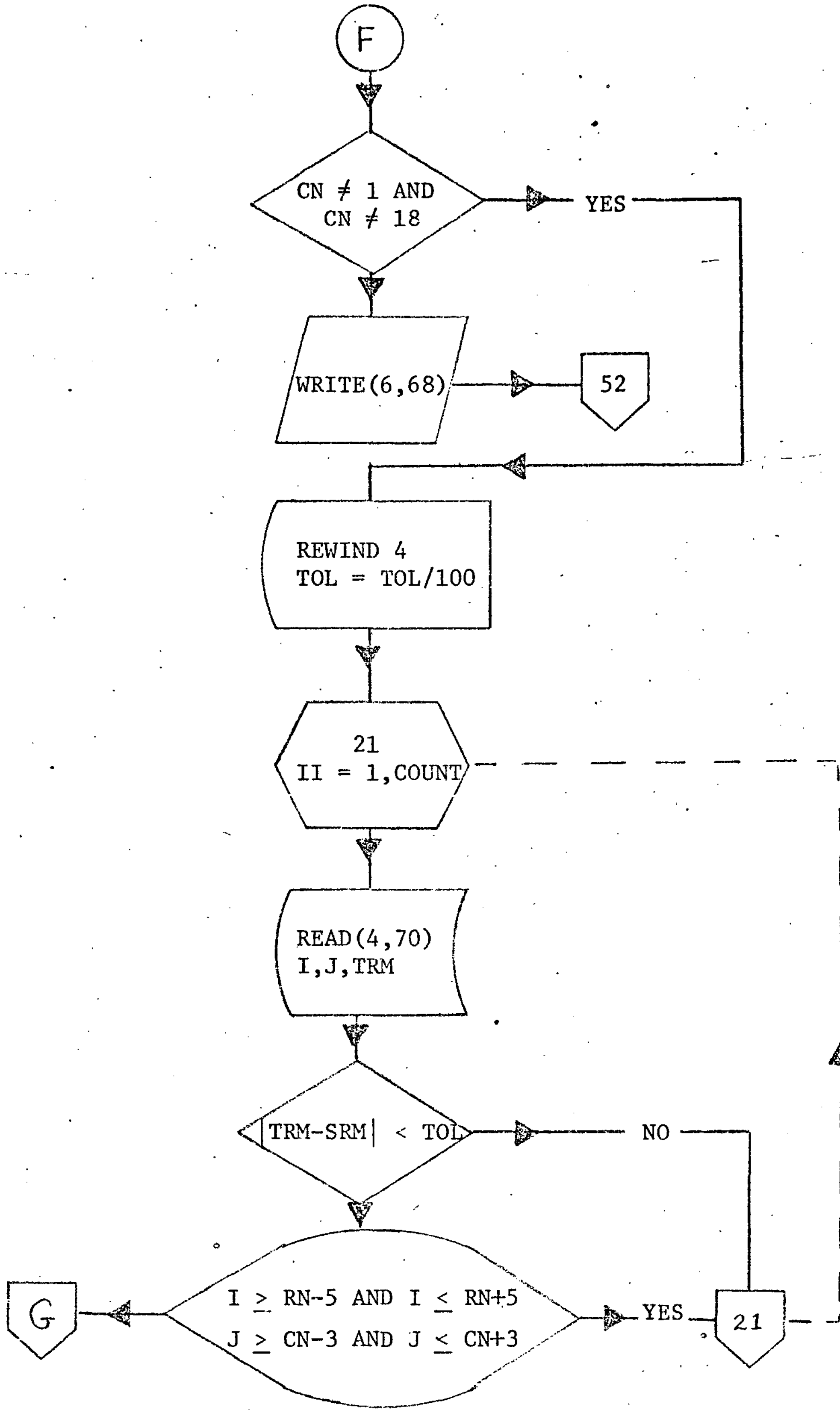


Fig. 8.5.3.5.2 DETAILED FLOW CHART - BACKUP -- CONTINUED

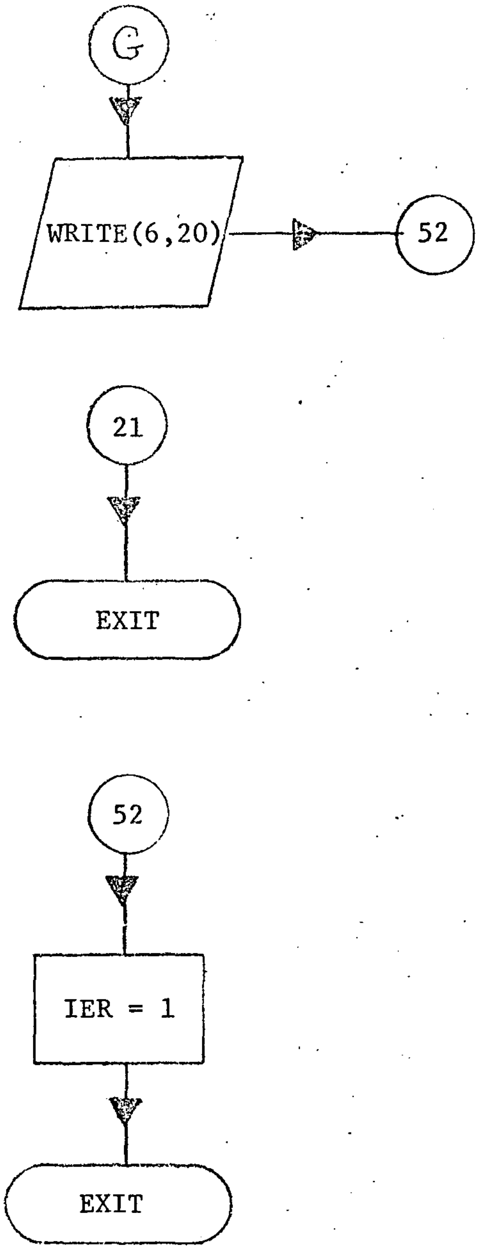


Fig. 8.5.3.5.2 DETAILED FLOW CHART - BACKUP -- CONCLUDED

8.5.3.6 Description of subroutine FMPT

IDENTIFICATION:

Name	--FMPT
Author/Date	--R.L. Wendt/June 1970
Organization	--ASR
Machine Identification	--UNIVAC 1108
Source Language	--FORTRAN V

PURPOSE/METHOD:

This routine finds match points after the initial match has been found at or near the "center" of the subframelets. The basic idea is to save the coordinates of the "center" match, select another interesting area via LKFR, unpack the data for S and F based on the interesting area row number for S and the difference of the row number for S and the differential from the previous match for F. The minimum column number for S at the "center" is saved. The routine begins by decrementing the row numbers, calculating the norms and saving the coordinates of the minimum. The process of decrementing the row numbers is continued until there is insufficient data to continue. The coordinate of the minimum column number at center is restored, the differential in the row numbers is restored and the algorithm increments the row numbers based again on row numbers supplied by LKFR. The usual process of unpacking and storing data, calculating the norm is repeated and the minimum selected. The process continues until there is insufficient data to continue.

The routine in operation is very similar to SKINNY in selecting a minimum norm, but the array FM varies over a smaller area because it is assumed that the non-linearity in the film can be accounted for by varying FM over IDIM±10 rows. As the process is in operation the minimum row, column and norms are saved for the fixed and moving matrices. The usual check is made to determine if the minimum is on any edge or if the minimum is not unique. In the first case the process terminates and in the latter proceeds.

The algorithm also has a feature which allows the moving array F to vary its column center slightly to account for mild non-linearities in the columns of the match of FM and also to keep the minimum off any edge.

The process of incrementing or decrementing the row number is accomplished via LKFR which seeks an "interesting" area to match. Because of this feature the difference between match points is not likely to be linear nor is the number of match points likely to remain fixed for various sub-framelets. The maximum number of match points can be determined by the setting of MN3. The dimensions of the arrays ICOORD and SSRM which contain coordinates of match points and norms has to be increased if more match points are desired by decreasing MN3.

USAGE:

Calling sequence

CALL FMPT (C, IER)

DESCRIPTION OF VARIABLES:

The variables in FMPT are generally the same as used in SKINNY.

DR - The difference between the row number of S and F where the minimum at or near the center of the framelets occurred.

INDX - An index which takes on values that insure that the coordinates and norms of match points are stored in ascending order of the row numbers.

IFB - A flag set to ±1. +1 means that the rows are being incremented toward the final row of the framelets and -1 means that the rows are being decremented toward the first row of the framelets.

IND - A difference of indices. Enables only that portion of BUF to be printed which contains norms calculated over some area of interest.

GENERAL FLOW CHART:

See fig. 8.5.3.6.1

DETAILED FLOW CHART:

See fig. 8.5.3.6.2

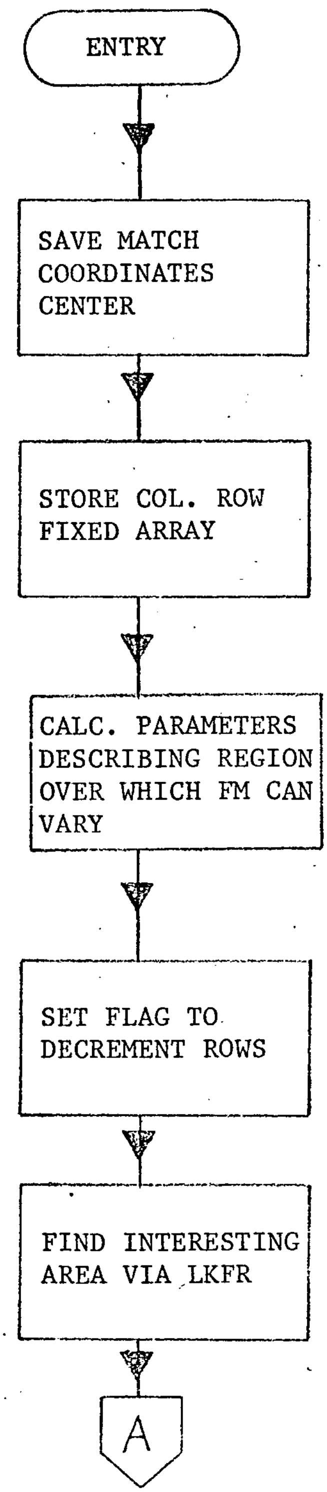


Fig. 8.5.3.6.1 GENERAL FLOW CHART - FMPT

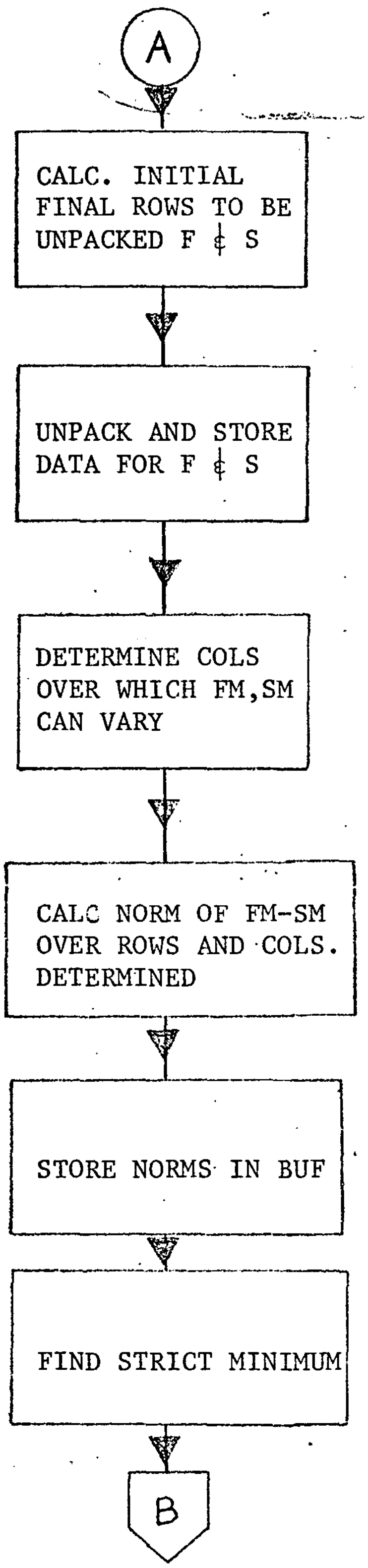


Fig. 8.5.3.6.1 GENERAL FLOW CHART - FMPT -- CONTINUED

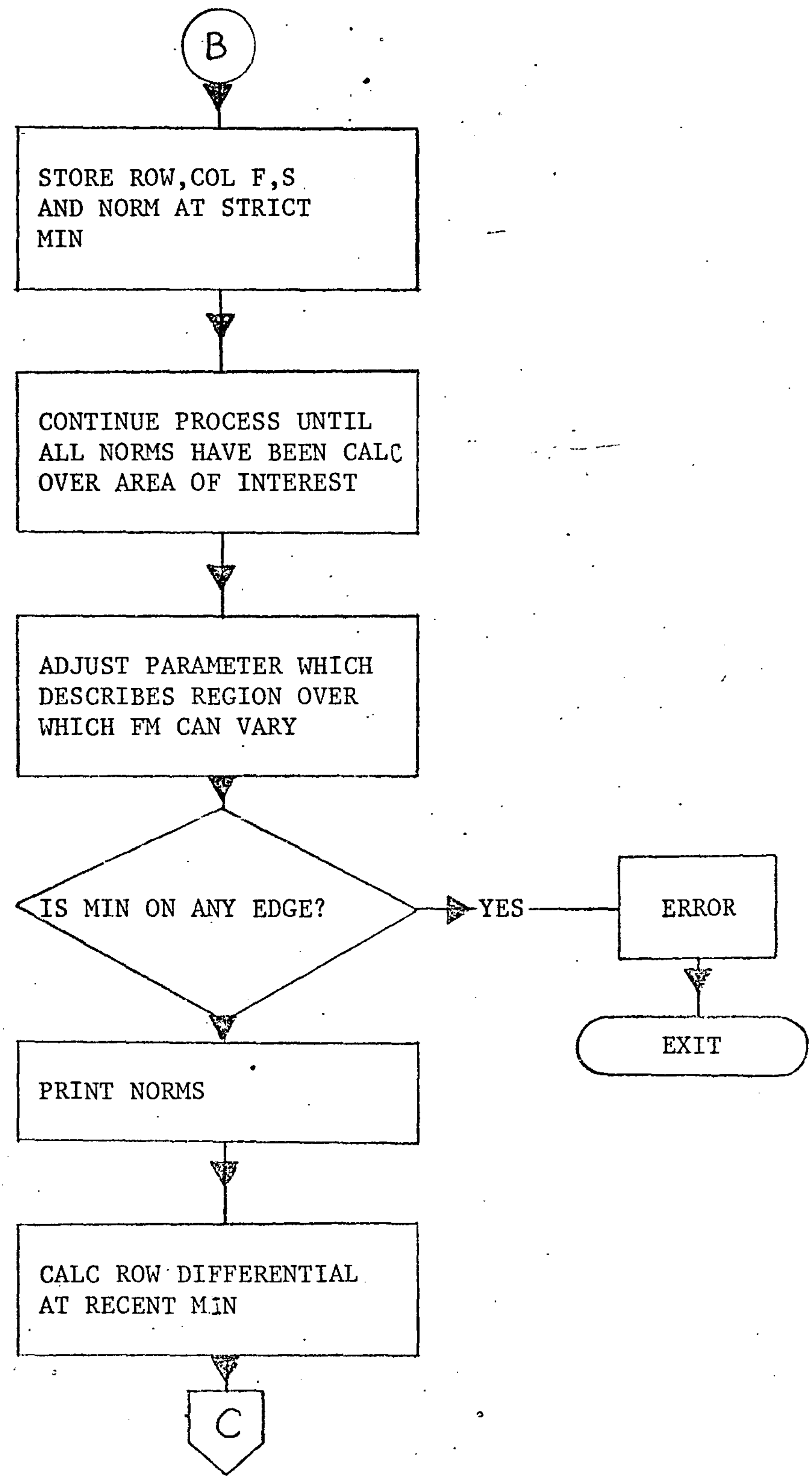


Fig. 8.5.3.6.1 GENERAL FLOW CHART - FMPT -- CONTINUED

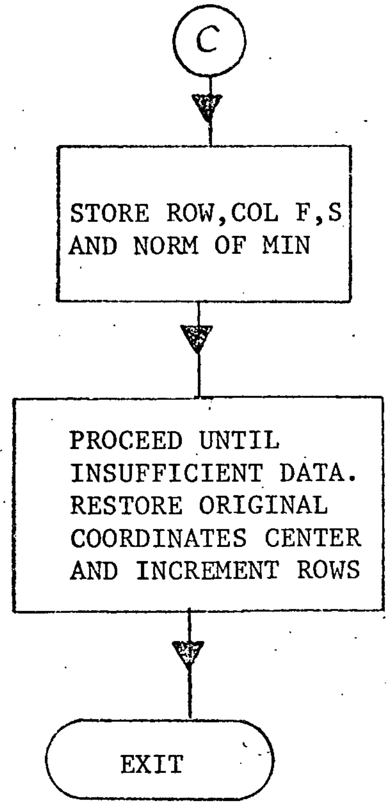


Fig. 8.5.3.6.1 GENERAL FLOW CHART - FMPT -- CONCLUDED

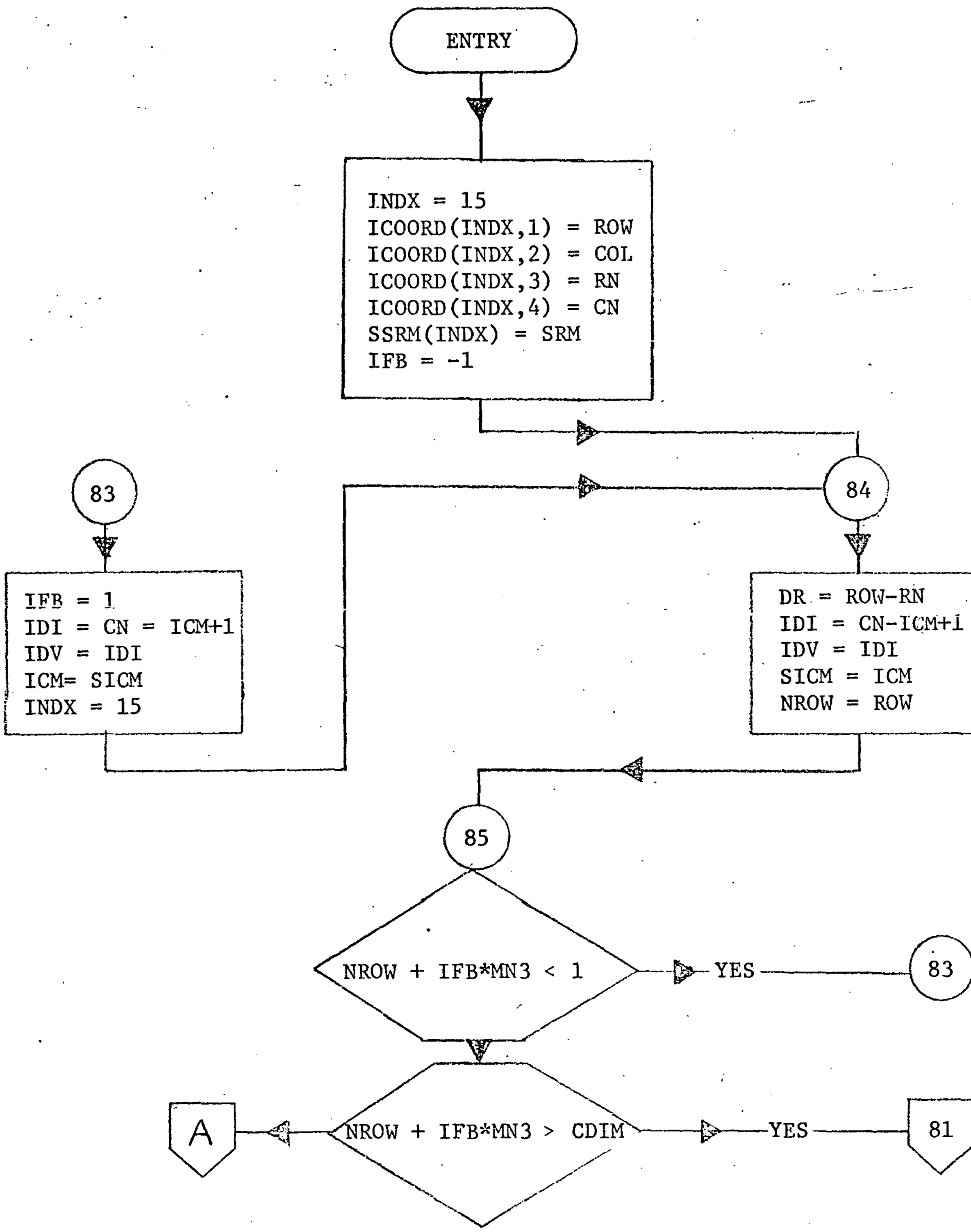


Fig. 8.5.3.6.2 DETAILED FLOW CHART - FMPT

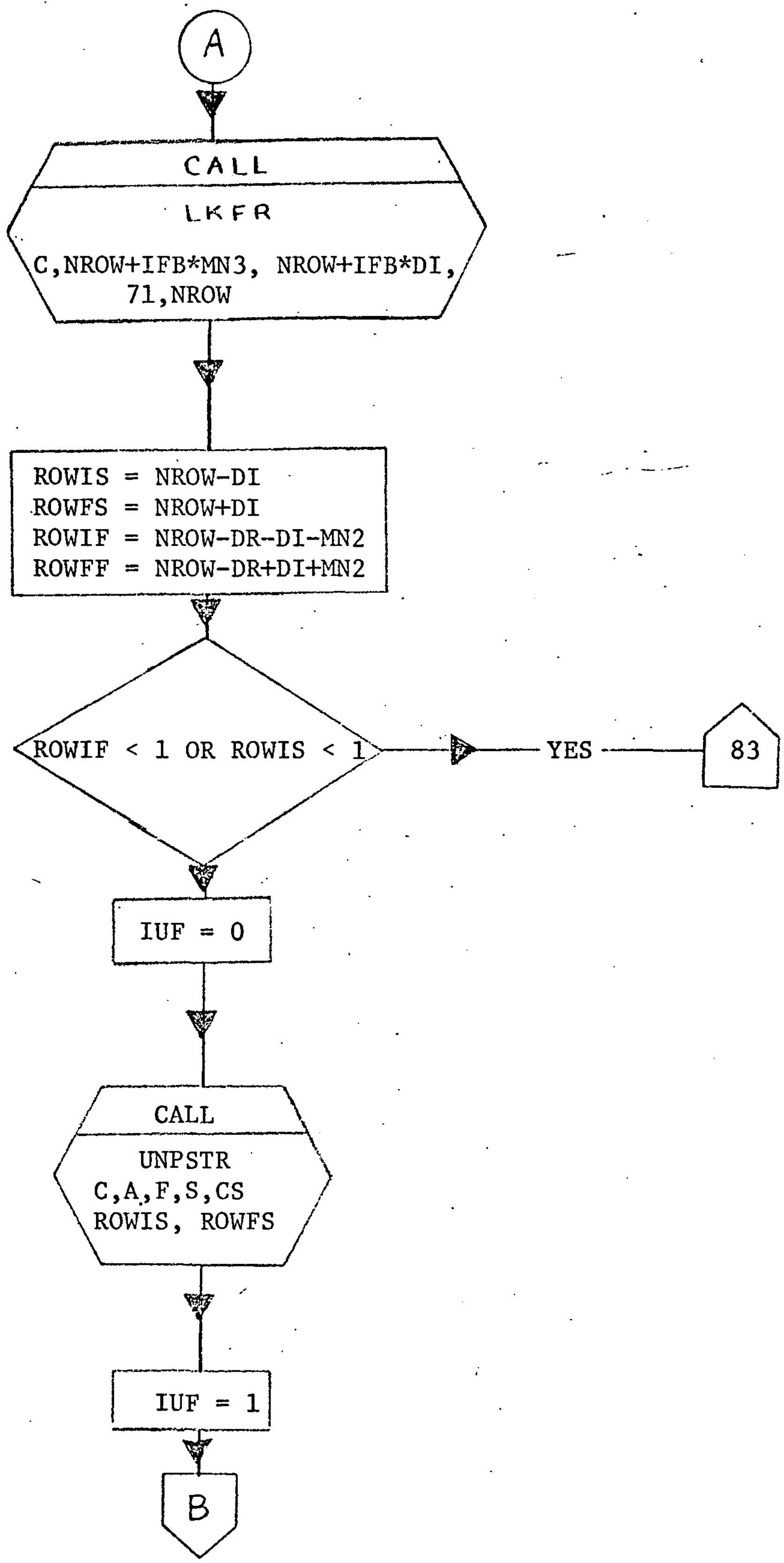


Fig. 8.5.3.6.2 DETAILED FLOW CHART - FMPT -- CONTINUED

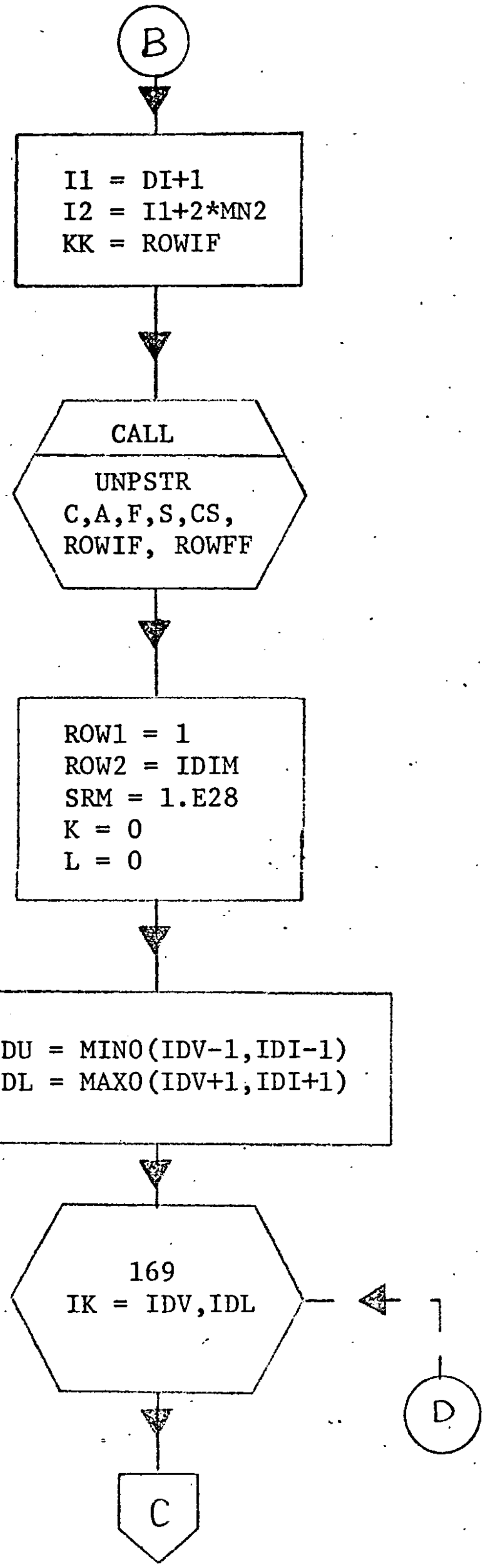


Fig. 8.5.3.6.2 DETAILED FLOW CHART - FMPT -- CONTINUED

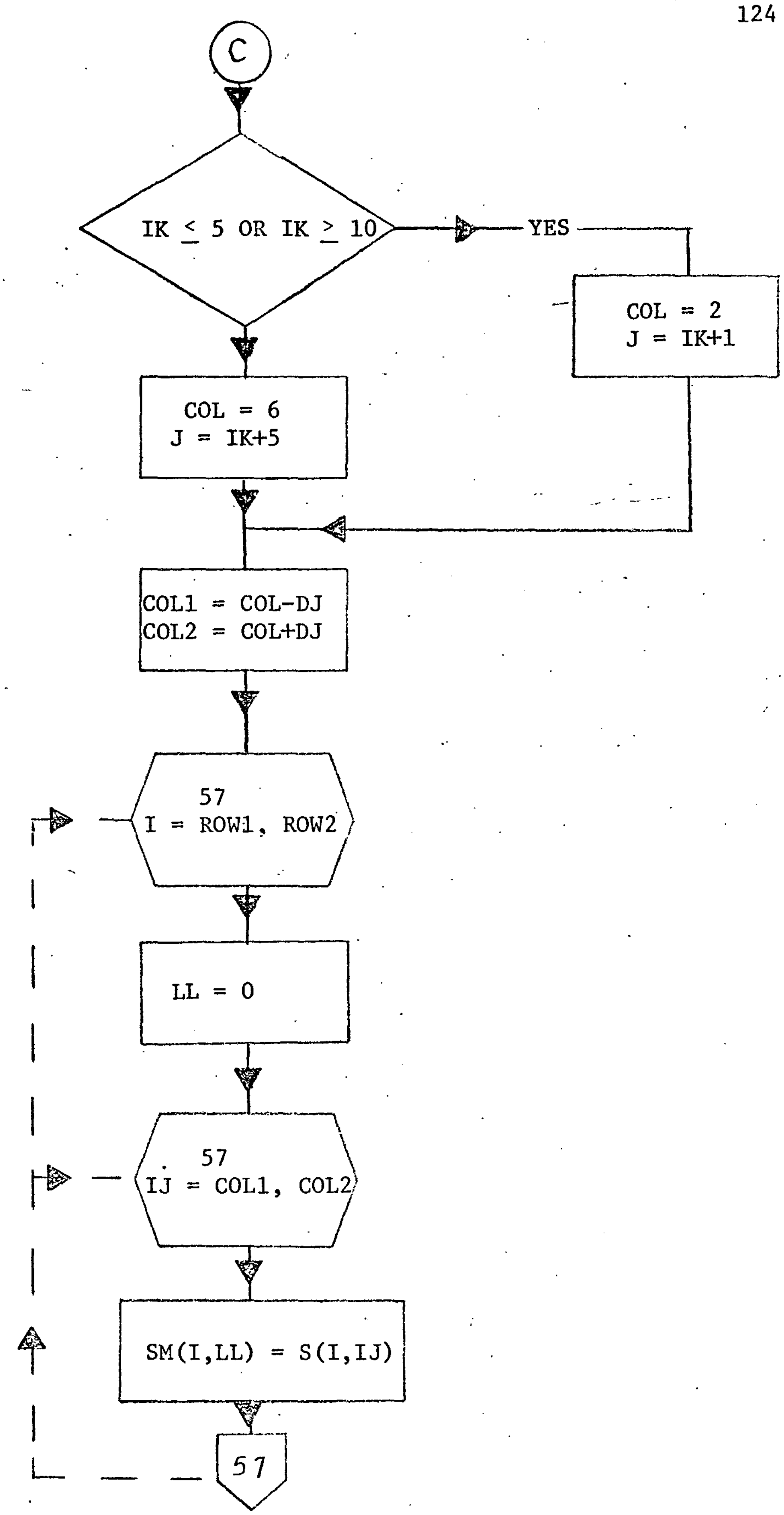


Fig. 8.5.3.6.2 DETAILED FLOW CHART - FMPT -- CONTINUED

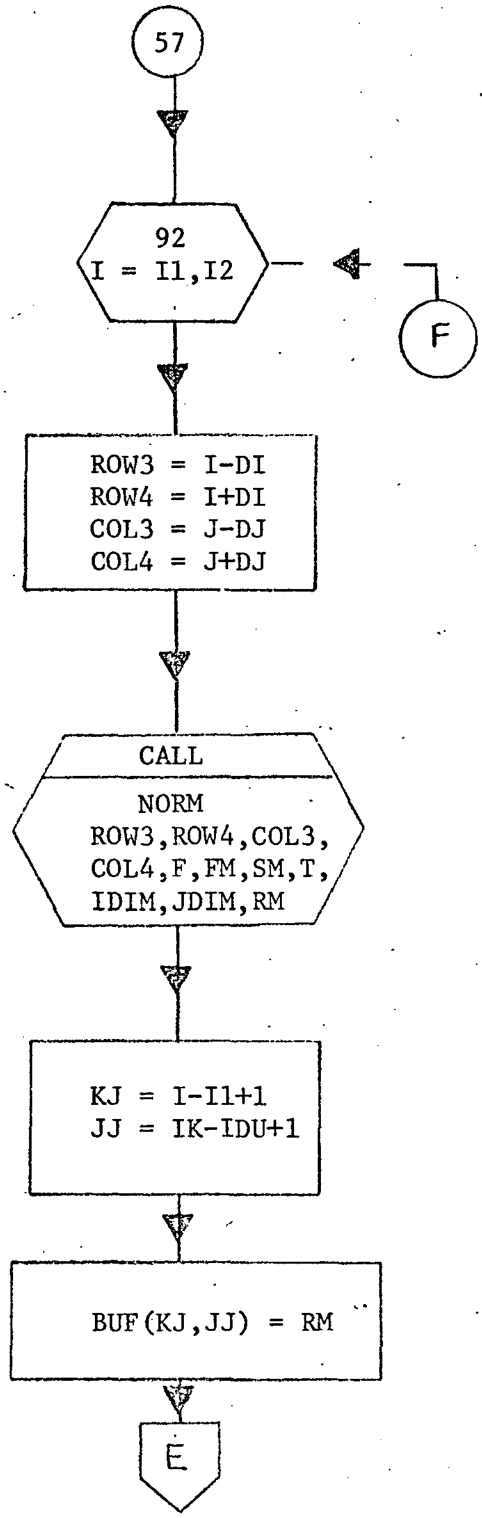


Fig. 8.5.3.6.2 DETAILED FLOW CHART - FMPT -- CONTINUED

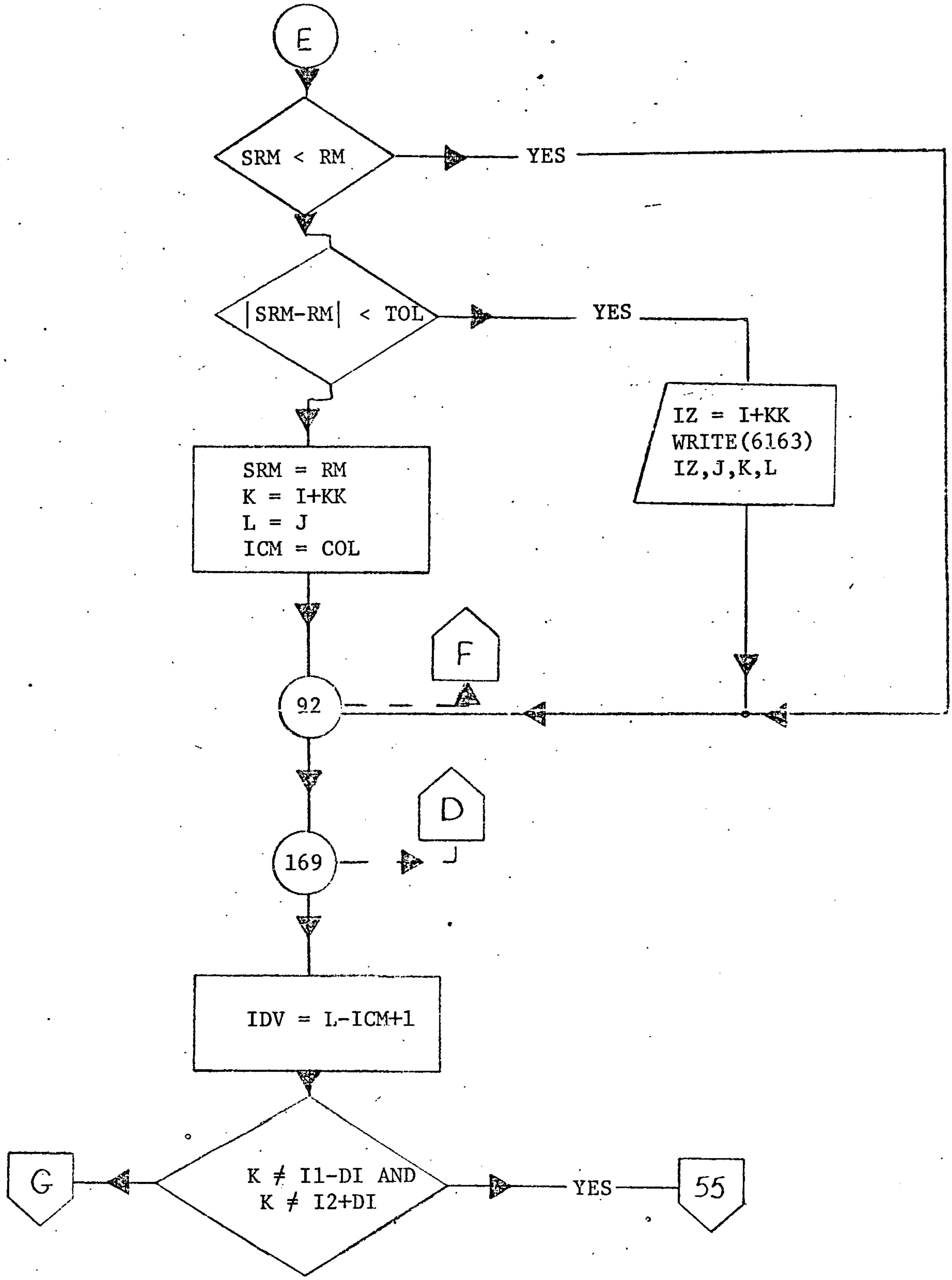


Fig. 8.5.3.6.2 DETAILED FLOW CHART - FMPT -- CONTINUED

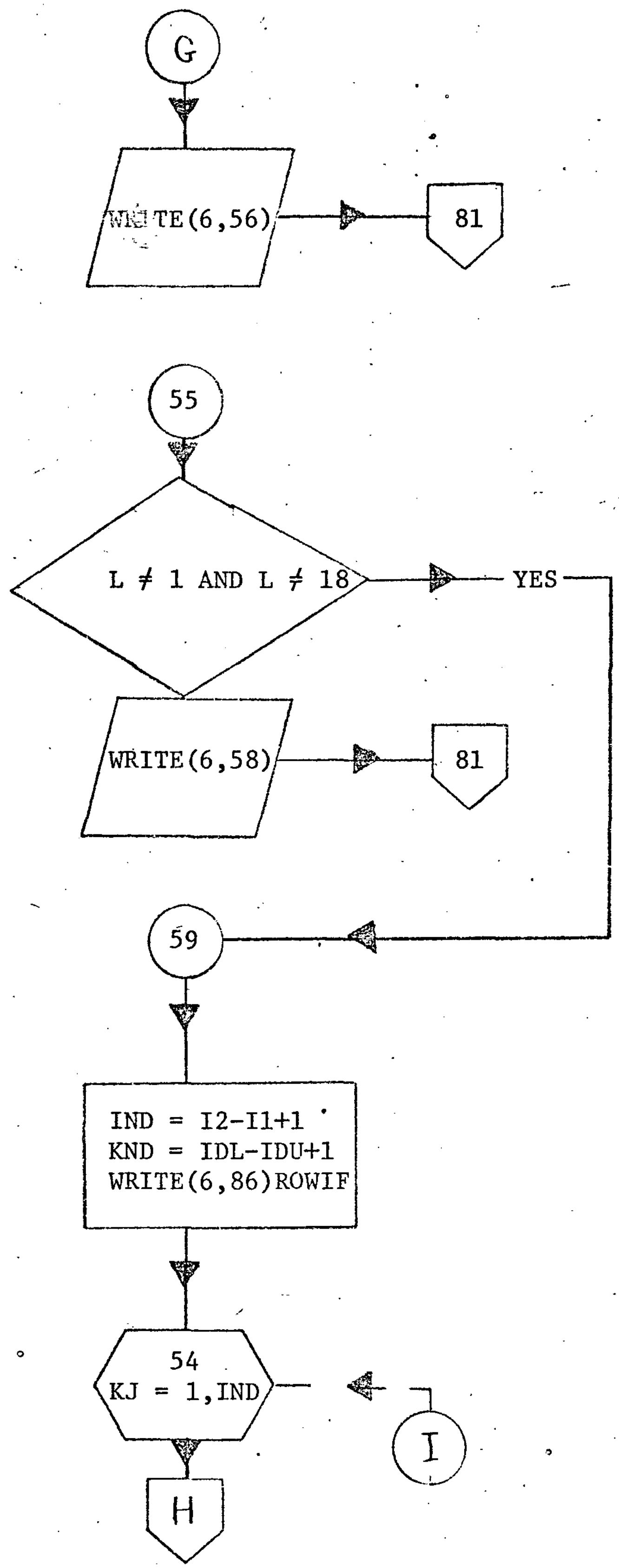


Fig. 8.5.3.6.2 DETAILED FLOW CHART - FMPT -- CONTINUED

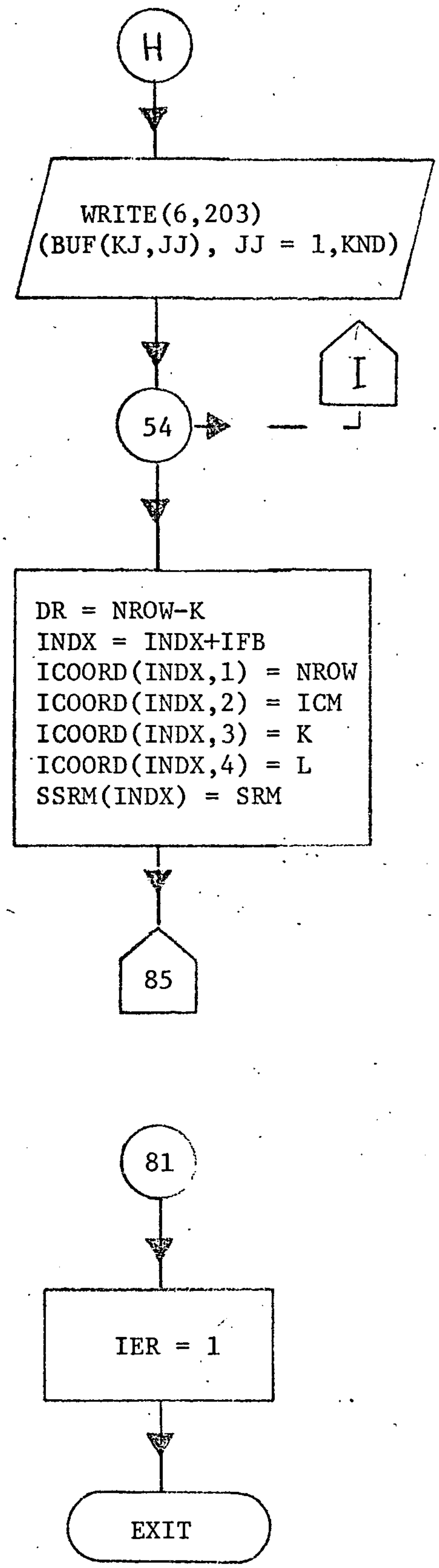


Fig. 8.5.3.6.2 DETAILED FLOW CHART - FMPT -- CONCLUDED

8.5.3.7 Description of subroutine PRNT

IDENTIFICATION

Name -- PRNT

Author/Date -- R.L.Wendt/October 1970

Organization -- ASR

Machine Identification -- UNIVAC 1108

Source Language -- FORTRAN V

PURPOSE/METHOD

This routine prints the coordinates of the match points and norm of the match.

The coordinates of the fixed array S; row column are printed first and the coordinates of the moving array F; row, column are printed and lastly, the norm at the match is printed.

USAGE

Calling sequence:

CALL PRNT (ICOORD, SSRM)

The description of the calling variables are given in Table 8.5.2.

DETAILED FLOW CHART

See Fig. 8.5.3.7.1

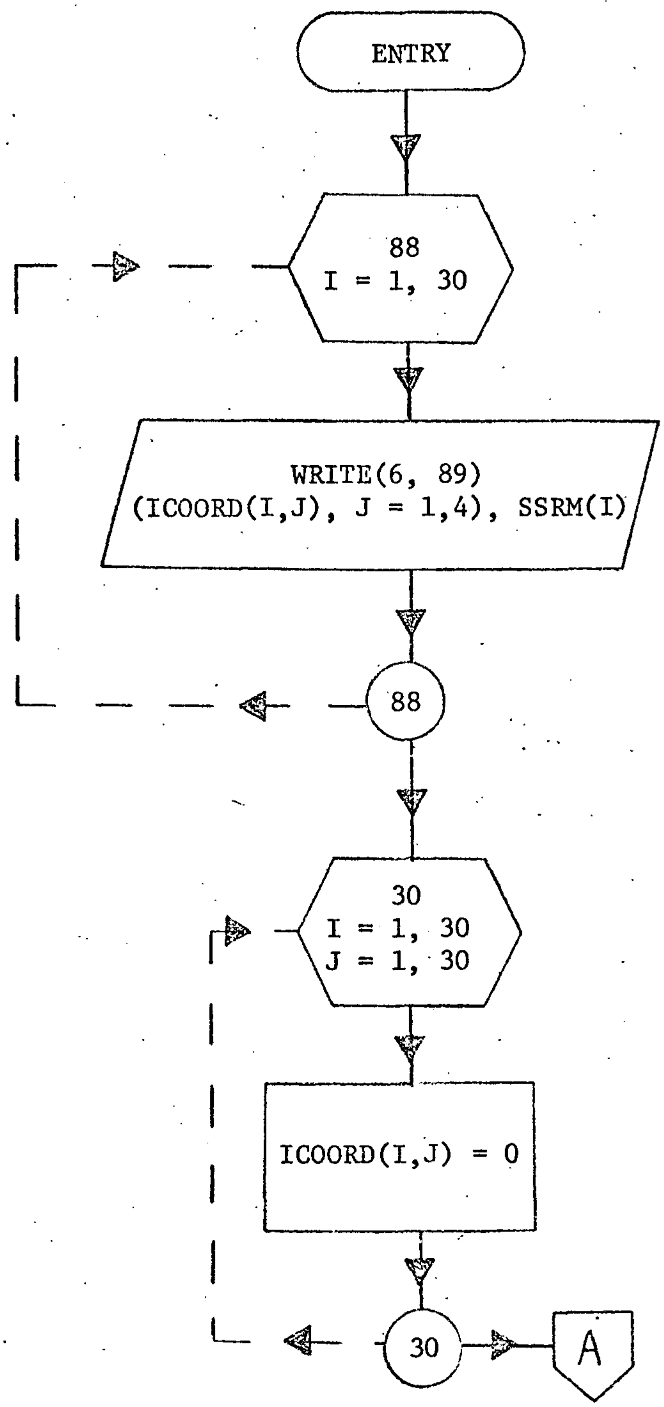


Fig. 8.5.3.7.1 DETAILED FLOW CHART - PRNT

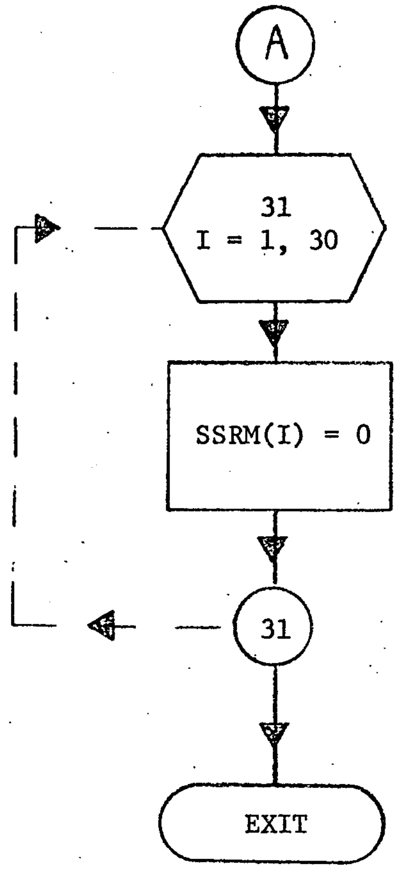


Fig. 8.5.3.7.1 DETAILED FLOW CHART - PRNT
(Concluded)

8.5.3.8 Description of subroutine UNPACK

IDENTIFICATION

Name	--	UNPACK
Author/Date	--	N.W. Naugle - Early 1967
Organization/Installation	--	CAD-MSC
Machine Identification	--	UNIVAC 1108
Source Language	--	Assembly Language

PURPOSE

Subroutine UNPACK converts a packed word into six 6-bit characters and stores each in a separate word.

USAGE

Calling sequence

CALL UNPACK (A, A1, A2, A3, A4, A5, A6)

A Word to be unpacked

A1-A6 Words to receive the six unpacked bytes.

9. The problem of joining framelets

The main output of SKINNY is a list of matchpoints which indicate exactly where the row and column match between adjacent framelets lies. For example, one such match is reproduced here:

COORDINATES OF MATCH POINTS

FIXED ARRAY S		MOVING ARRAY F	
ROW	COL	ROW	COL
297	6	318	630
599	2	626	624
736	6	771	630
890	2	917	625
961	2	988	625
1147	6	1169	630
1315	2	1345	625
1502	6	1531	630
1679	2	1707	625

In this match, we note that, for example, ROW 297 COL 6 of the right hand framelet (in fact, framelet 027) matches ROW 318 COL 630 of the left hand framelet. If we make a small table of COLUMN differences, we confirm the fact that the COLUMN match is fairly steady:

MATCH POINT	Δ COL
1	624
2	622
3	624
4	623
5	623

6	624
7	623
8	624
9	623

The COLUMN jitter of $\pm$ 1 element is, in fact, less than one might accept. We can conclude that COL 636 of the right edge of the left framelet is matched with COL 11 on the left edge of the right framelet.

In joining two or more framelets as we are here, we are attempting to construct the joined framelets on a mesh (called the MASTER MESH) of points on which we define the value of the picture using the match points and (perhaps) other control information. In our case, since we have nothing better to go on, we used a MASTER MESH consisting of the coordinate system of the first sub-framelet translated down the framelet (so that it begins with ROW DR (1) (an integer which is read in and which indicates the first ROW to be joined)) and extended horizontally. Each COLUMN number of the MASTER MESH has associated with an integer 1, 2 or 3 (in a vector IFLT (1878)) which indicates which framelet the point for that COLUMN number is coming from and another integer 1, 2, ..., 636 (in a vector IFLTCL (1878)) which indicate which COLUMN of that framelet the point is coming from. Relatively, the ROW problem is simpler (since there is no problem with switching back and fourth to avoid drummarks), and we perform a simple linear interpolation to arrive at the ROW numbers for sub-framelets 2 and 3.

As hinted at above, the main reason the COLUMN data is read in rather than calculated in a subroutine is that it was simpler to punch the cards than it was to do the logic to jump around between framelets to avoid the drummarks. Were this program to be run on a production basis, it would be worth the effort to read in the COLUMN match points (given by SKINNY) and, using these, call a subroutine which calculates IFLT and IFLTCL.

COMPUTER PROGRAM DOCUMENTATION

Joining Three Surveyor - III Area Framelets

Program JOIN

Project A

By

Jack Bryant

and

R. L. Wendt

Prepared by

Applied Scientific Research, Inc.

Houston, Texas

Under Contract NAS 9-10577

for

MAPPING SCIENCES DIVISION

National Aeronautics and Space Administration

Manned Spacecraft Center

Houston, Texas

September, 1971

10. DOCUMENTATION OF JOIN

10.1 INTRODUCTION

The problem of practically joining several framelets of LO data is mainly one of efficient tape handling and computation. This document is explicitly concerned with three framelets, mainly to fix ideas, but the program can handle as many framelets as there is provision for tape drives in the system. (At least one drive is required for the output.) The program makes extensive use of multiple buffering to protect against read errors and to allow parallel computation and reading. However, the heart of the program is a circular buffer RMTX, dimensioned by (3, 15, 636). (RMTX may be thought of as three circular buffers.) After the initial read phase of JOIN (which fills RMTX), the first block contains 15 unpacked lines from UNIT(1), the second contains 15 unpacked lines from UNIT(2), and the third block of RMTX contains 15 unpacked lines from UNIT(3). As the lines are "used" (that is, as the framelets are joined preceeding down the framelet), the three circular buffers are kept full by appropriate (double buffered) reading.

The joined output of even a nine chit area involves a substantial amount of tape. The possibility that there may not be enough output tape room is considered; in the event that full framelets were being joined, this test would result in output tape units being switched.

10.2 GENERAL DESCRIPTION OF JOIN PROGRAM:

The program initially sets all the necessary constants to the prescribed value. The row mesh, i.e. the interpolated row numbers, are determined by RWMSH. The values for the arrays IFLT, IFLTCL, DR, SCF are read in from data cards. The header array incorporated on each input tape is spaced over by NTRAN since the header essentially provides no necessary information for this program. Next the notch data and the number of rows determined by DR are spaced over. Each input tape should be ready to read in the first line of each framelet for the joined picture.

The process of double buffering NLINES into the array RMTX is performed by unpacking, removing drum marks, storing and correcting data while the next buffer is being read.

At this juncture the program has RMTX filled with corrected data and the process of joining the framelets is begun. It is of interest to note that the arithmetic on the rows of RMTX is treated modulo NLINE so that RMTX is essentially a circular buffer and is depleted of data when the framelets are joined. Because of the circular buffer and modular arithmetic, the question of whether more data is needed to accomplish the join process is continually tested. The process of reading in more lines for RMTX is accomplished via a modified double buffer technique on each framelet individually. The data undergoes the usual process of unpacking, removing drum marks and correction before storing.

After the program has determined that there is sufficient data in RMTX, a row of the output joined picture is generated by linear interpolation. The final joined row is packed into OBUF and a provision is incorporated so that a DLOCE correction may be applied to the output data. Once again the process of output uses double buffering so that the join algorithm never has to wait while a tape is being read or written.

Once the entire framelets have been joined, an end of file is placed on the output tape, all units are rewound and the joined output is printed out at the matched edges of the framelets.

It should be noticed that the join program incorporates, within the process of reading tape while joining rows together, the feature that a read error may occur. The process of circular buffering may allow the algorithm to join rows of the picture repetitiously while a read error is occurring. This eventuality has been accounted for and in the event that recovery from a read error is achieved the process continues. Not accounting for this eventuality could lead to disastrous results in the outputted picture. If no recovery from a read error is possible, then nothing can be done at any rate and the process is terminated.

One further observation is that the joining process is slightly biased to reading tape since it is clear that the process is computation bound. This feature allows for a better balance of computing while reading tape, thereby minimizing overall execution time.

10.3 GENERAL FLOW CHART

Fig. 10.3.1 gives the general flow chart for JOIN. Two of the blocks are not completely self-explanatory, and are expanded in detail in Fig. 10.3.2 and Fig. 10.3.3. All of the symbols and subroutines mentioned are given detailed descriptions in the remainder of §10.

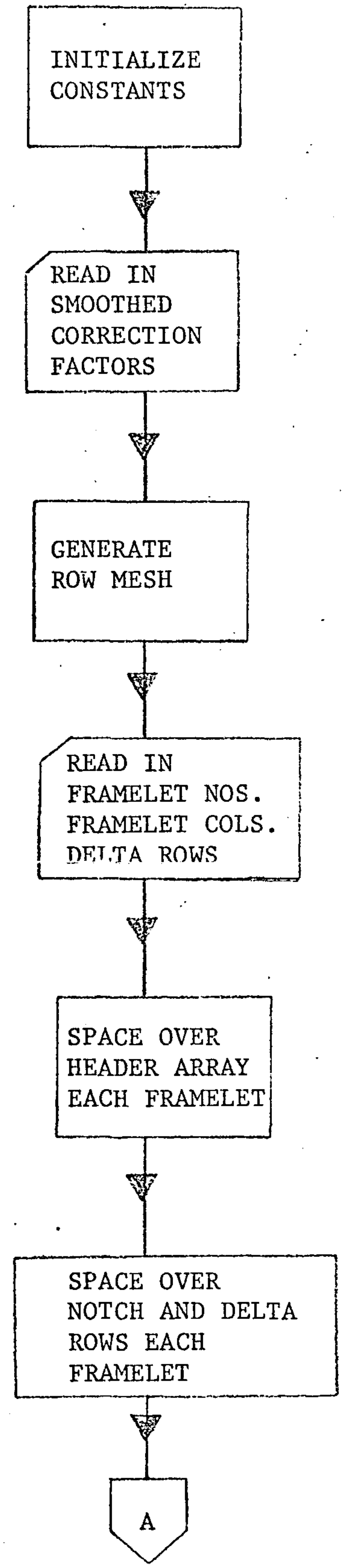
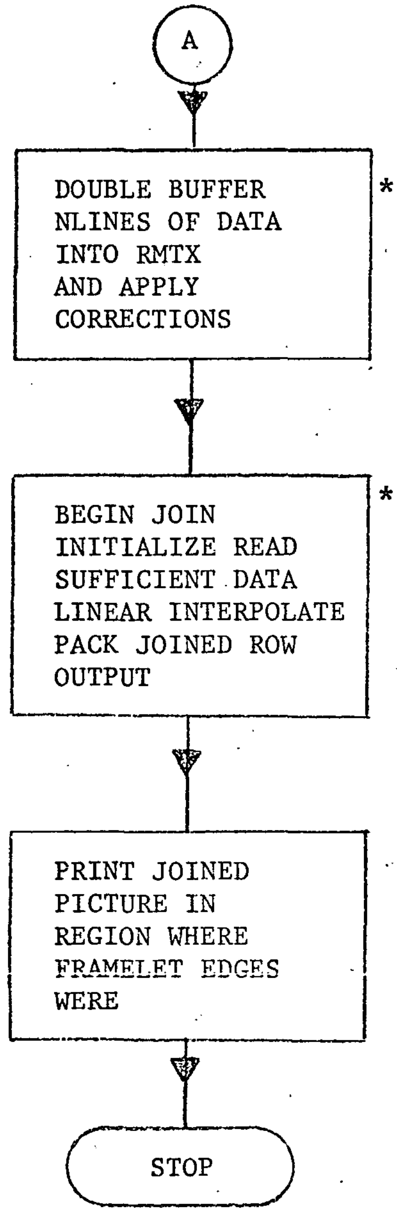


FIG. 10.3.1 GENERAL FLOW CHART JOIN



*THESE BLOCKS ARE EXPANDED FURTHER IN DETAIL IN FIG. 10.3.2 AND FIG. 10.3.3 (TOP, BOTTOM RESPECTIVELY).

Fig. 10.3.1 GENERAL FLOW CHART JOIN (concluded)

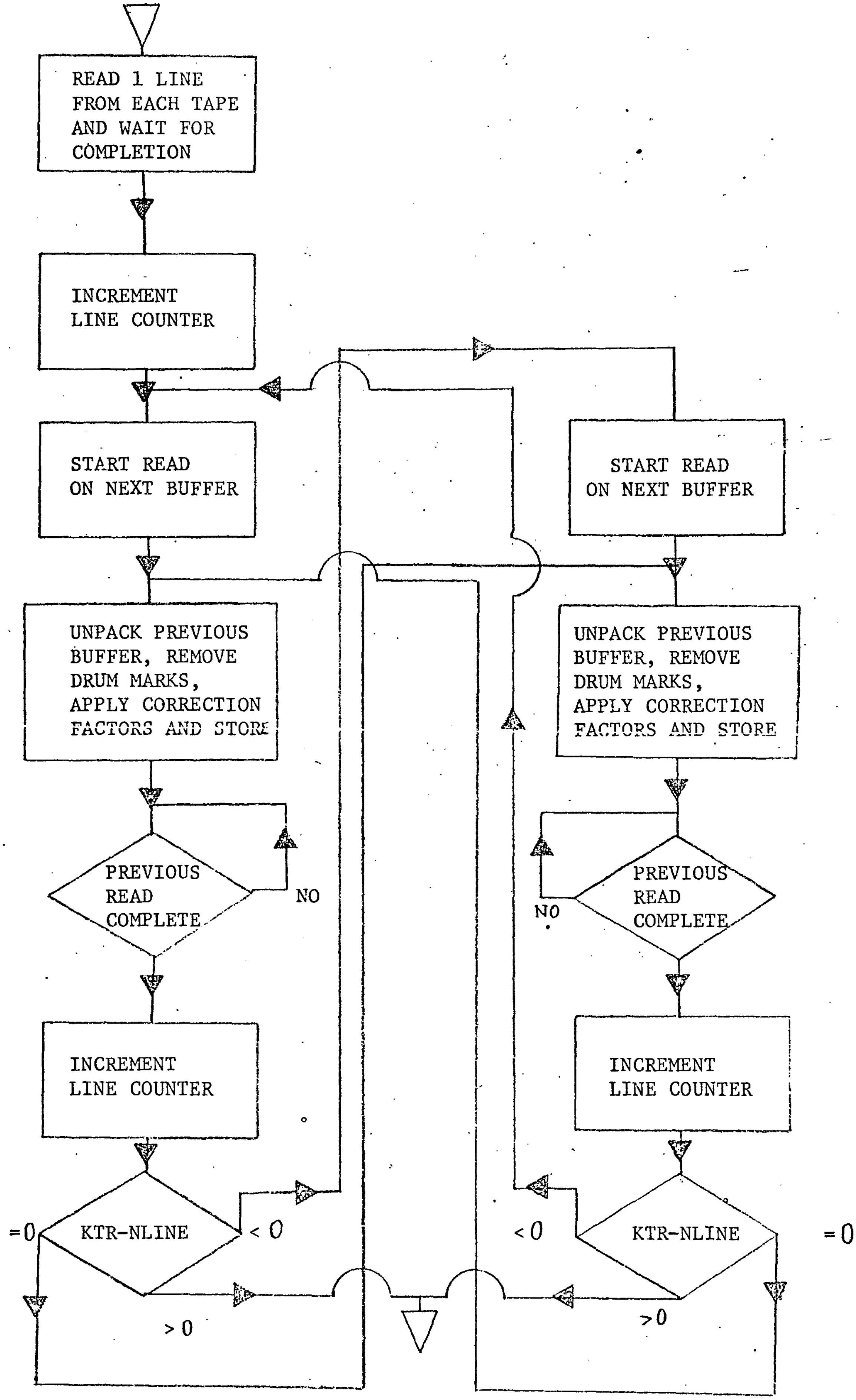


FIG. 10.3.2 DETAIL

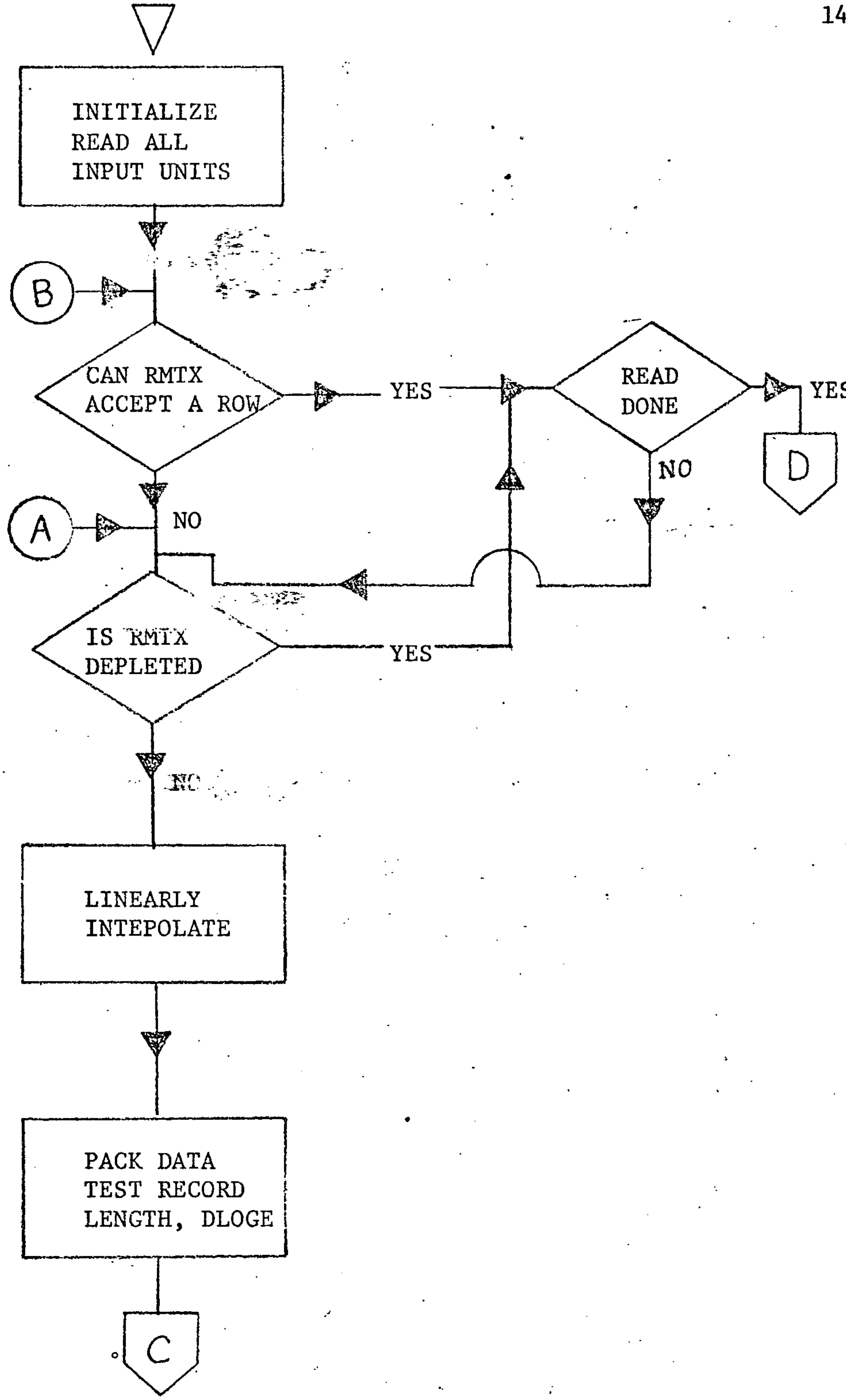


Fig. 10.3.3 DETAIL

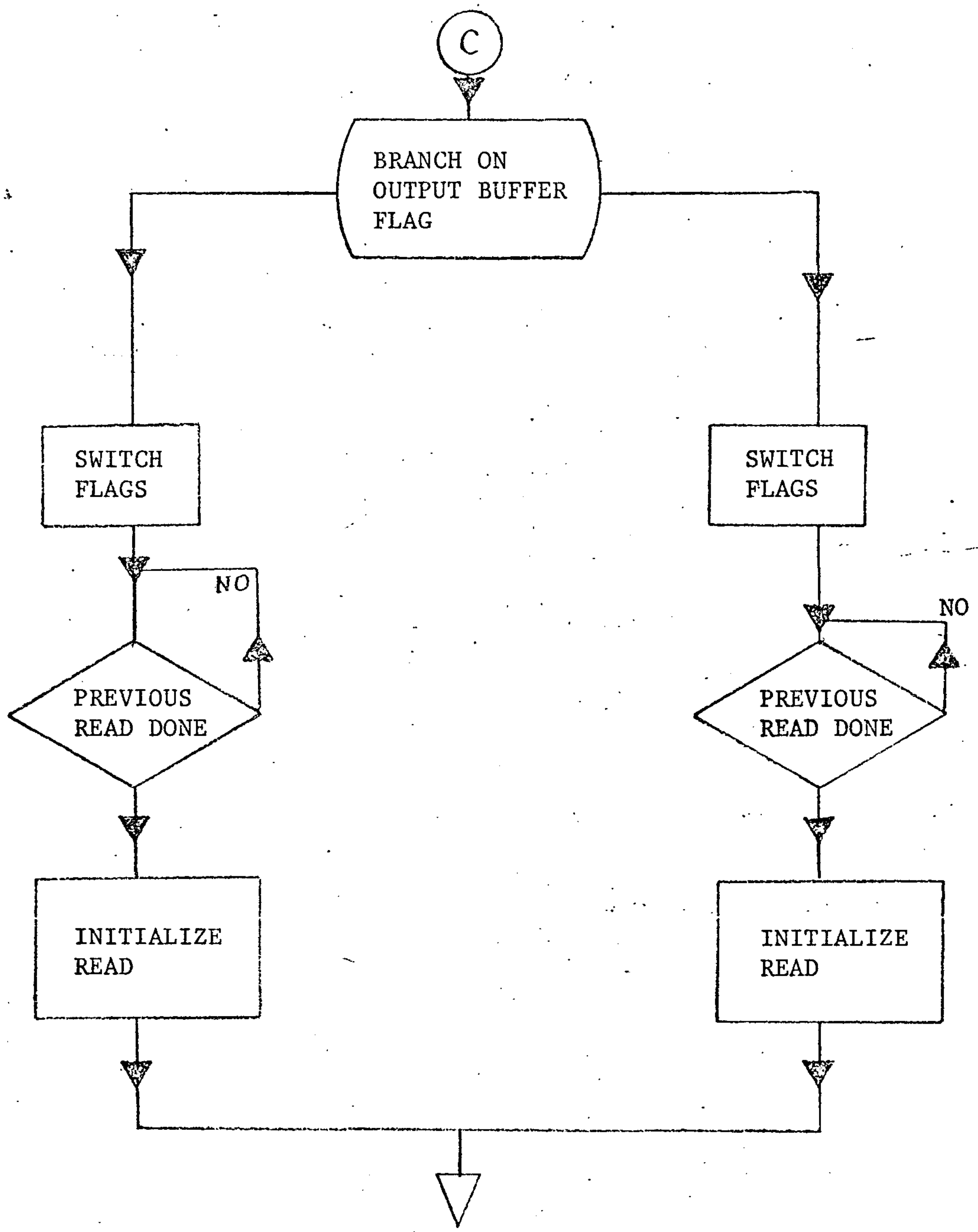


Fig. 10.3.3 DETAIL (Continued)

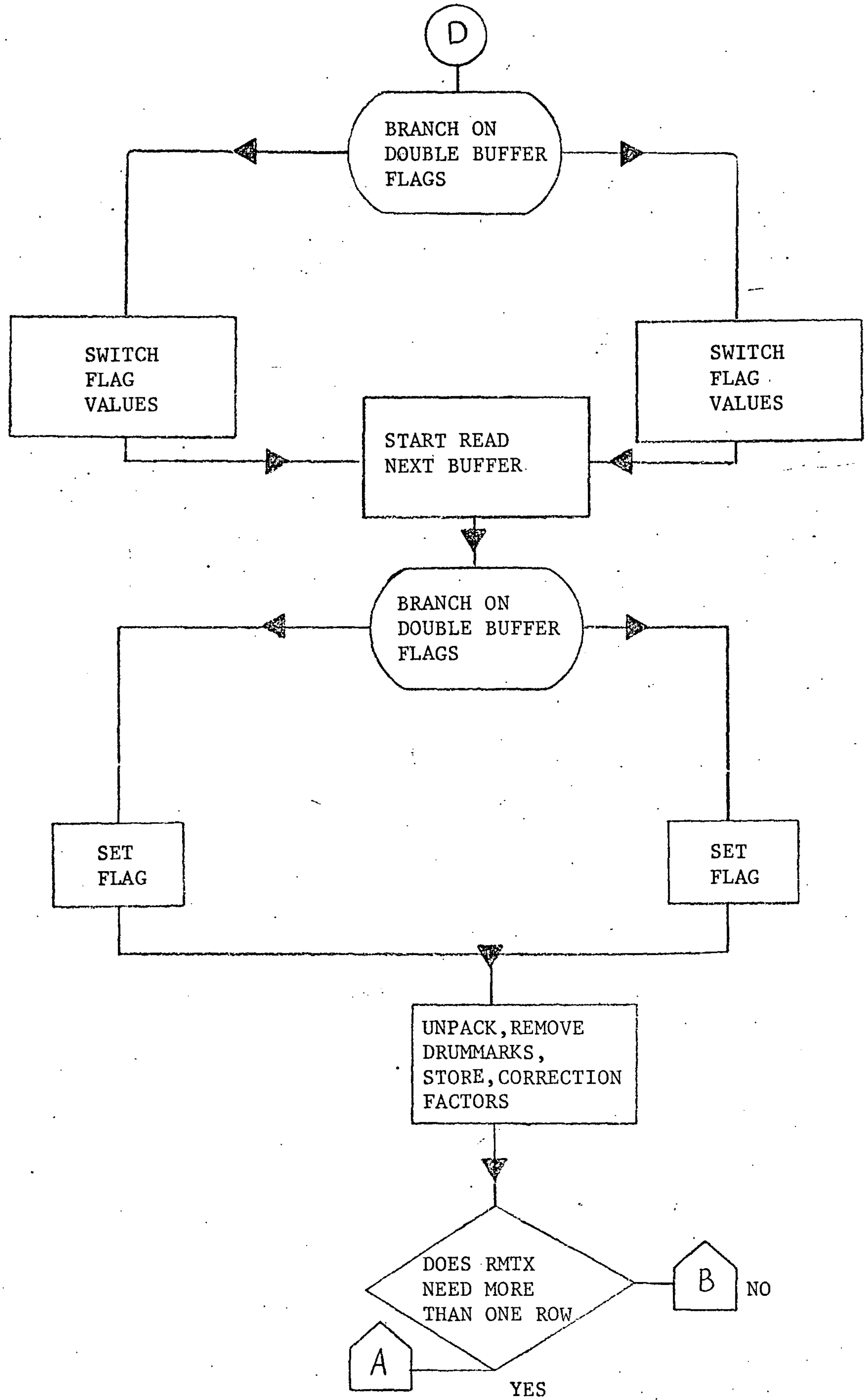


FIG. 10.3.3 Detail (Concluded)

10.4 USAGE

10.4.1 INPUT DESCRIPTION

10.4.1.1 Card input

SCF - Smoothed correction factors made by program CFAC (the documentation of which is included in the report on Project B). These cards constitute 636 numbers punched in FORMAT (6E13.8).

IFLT - An array of 1878 integers indicating from which framelet the k th ($1 \leqq k \leqq 1878$) column of a row of the MASTER MESH is built using. These numbers are read in FORMAT (80 I 1).

IFLTCL - An array of 1878 integers indicating from which column of the IFLT (K) th framelet the k th column of a row of the MASTER MESH is built using. These numbers are read in FORMAT (26 I 3).

DR - An array of three integers which contains the number of the first line from each framelet to be joined. These integers are read in FORMAT (3 I 3).

In addition to the parameters read by the main program, subroutine RWMSH reads

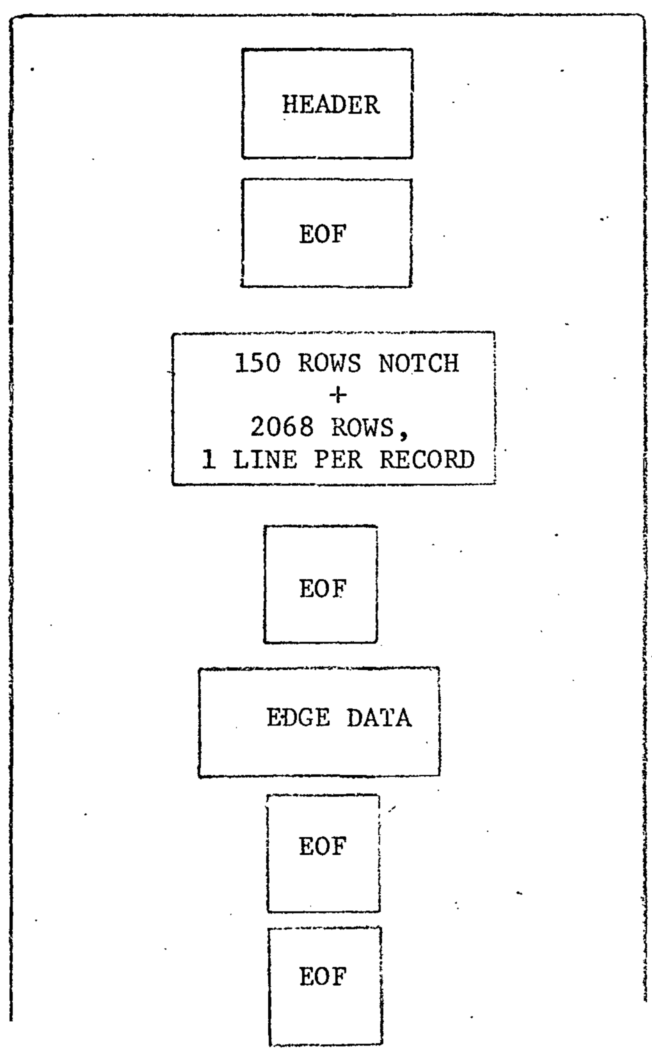
R(I,J) - The right row match numbers given by SKINNY. In our specific problem, there are two sets of them (because there are three framelets). The numbers are read in FORMAT (16 I 5).

L(I,J) - The left row match numbers given by SKINNY. In our problem, there are two sets of them. The numbers are read in FORMAT (16 I 5).

10.4.1.2 Tape input.

The tapes read here are designed to look like full framelet tapes in order to test JOIN without using much computer time. The program will also use the tapes made for the program FILTER. The format of the tapes is discussed fully in the report for Project B. A brief discussion of each input tape follows:

Each input tape has the following format



Each row has 106 words of 6 6bit bytes.

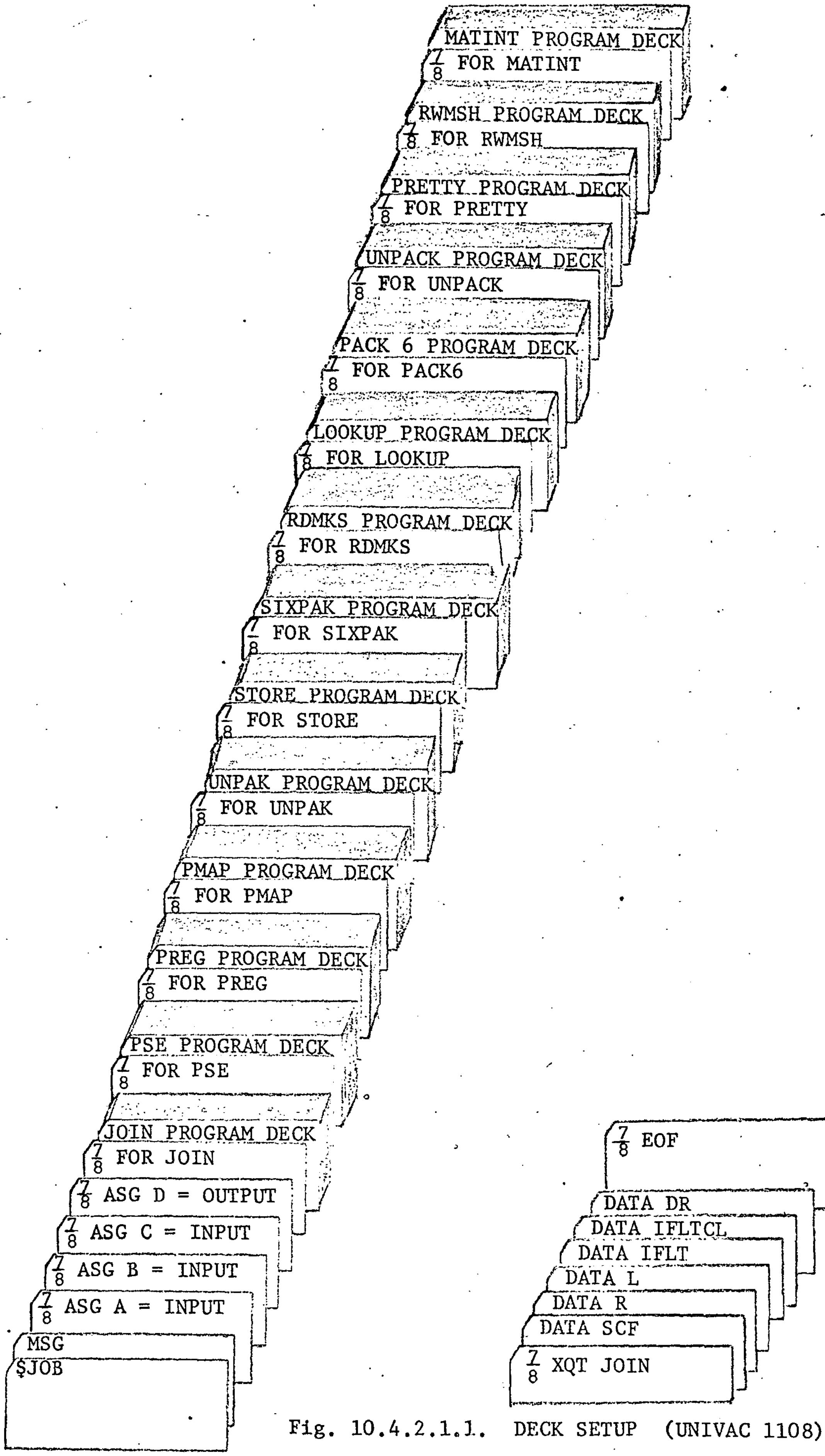


Fig. 10.4.2.1.1. DECK SETUP (UNIVAC 1108)

10.4.2 PROGRAM RUN PREPARATION

10.4.2.1 Card deck setup.

For the card deck setup, see fig. 10.4.2.1.1. The program is also on PCF tape #62769, held in the name R. L. Wendt.

10.4.2.2 Required I/O devices and special hardware.

The program uses the standard FORTRAN V read statements for input card data. For a nine chit, three framelet area with PCF tape, five tape drives are required. One FH432 drum file is required. No use is made of FSTRN. All I/O on tapes and handling of the drum is by NTRAN read/write statements.

(The program may require some education of the operator before it runs smoothly because of the output at the very end. The joined area is printed using symbols to get a look at the joined area; since the printer makes such a strange noise (with page edject suppressed and printing full pages), the operator may think something wrong with the program and abort the run. However, the output tape is already written before the printing subroutine is called so that the main function of JOIN has been accomplished.)

10.4.2.3 Subroutine requirements

The program uses the following subroutines:

PSE	LOOKUP*
PREG*	PACK6*
PMAP	UNPACK*
UNPAK	PRETTY*
STORE	RWMSH
SIXPAK	MATINT
RDMKS	

Subroutines marked with * are standard LO subroutines documented elsewhere. The others are described in §10.5.

10.4.3 OUTPUT DESCRIPTION

10.4.3.1 Printed output

The only normal output will be the joined area printed using subroutine PSE. In case of an error reading tape which cannot be recovered from, appropriate diagnostic messages are printed. The program produces about 5,000 lines of output including program listing.

10.4.3.2 Tape output

The output tape starts with the first row of the MASTER MESH and writes one row per record, each row containing ICMAX (= 1878 in our case) bytes packed 6 6bit bytes to a word. At the end of the data, an end of file mark is written on the output tape.

10.4.4 EXECUTION CHARACTERISTICS

10.4.4.1 Restrictions.

The most severe restriction to the program is the core size of the 1108. This, coupled with restrictions on the number of tape drives, restricts the number of framelets which can be joined at once. Another restriction is the length of output tape available.

The program requires about $11,300_8$ core locations for code and $152,000_8$ core locations for storage, leaving only about $2,500_8$ core locations free. (The system uses $14,000_8$ locations.)

10.4.4.2 Run time

The run time is somewhat variable, probably owing to the problems of hanging tapes. However, approximately 18 minutes should be required for a 2000 x 1878 (approximate 9 chit) area. It is probable that on a very long run (such as joining three framelets together), the initial delay while tapes are being hung and positioned would be vastly overcome by computation time.

10.4.4.3 Accuracy.

Depending, of course, on the accuracy of the input data, the joined picture is perfect. (The mechanical scan distortion of the first framelet is carried over to the next two framelets. Nevertheless, the resulting distortion should be much less than the GRE prints. The Reseau control points are too far appart to allow correction on a 9 chit area.)

10.5 REFERENCE INFORMATION

This section contains the following information.

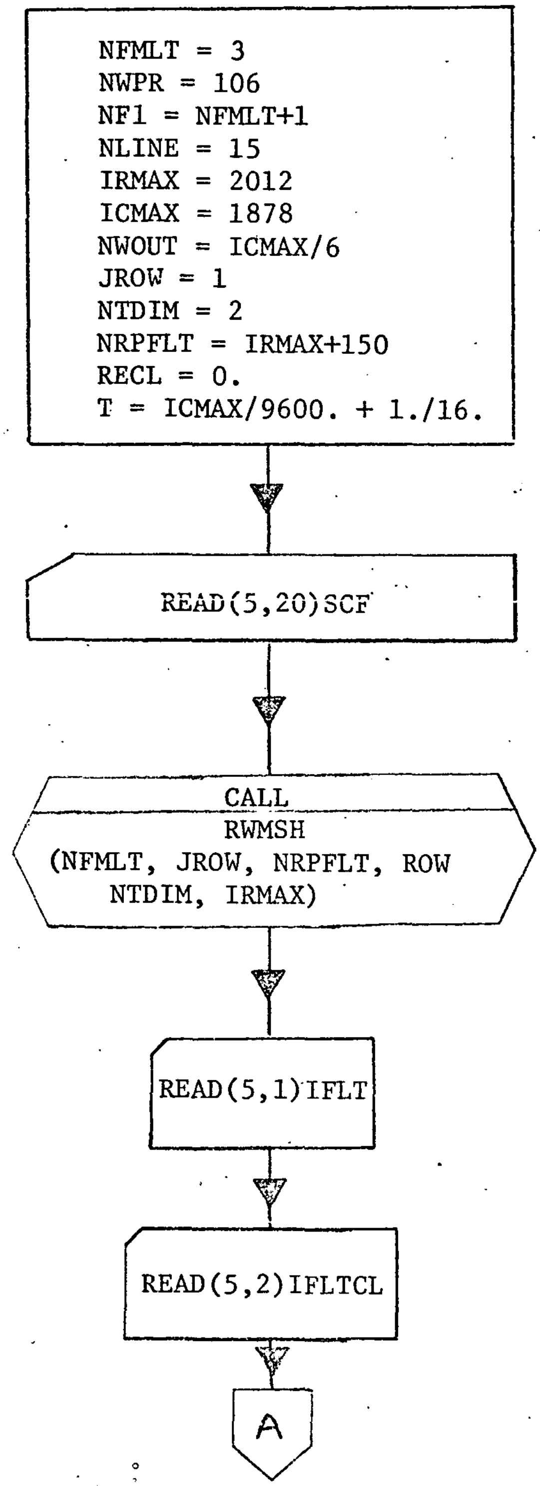


Fig. 10.5.1 DETAILED FLOW CHART - JOIN

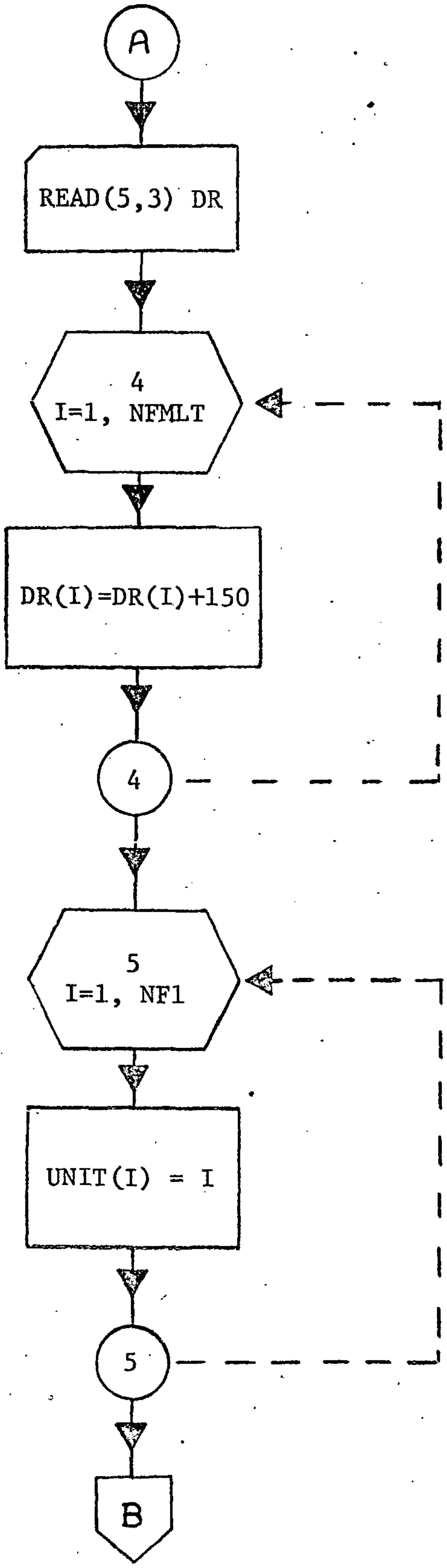


Fig. 10.5.1 continued

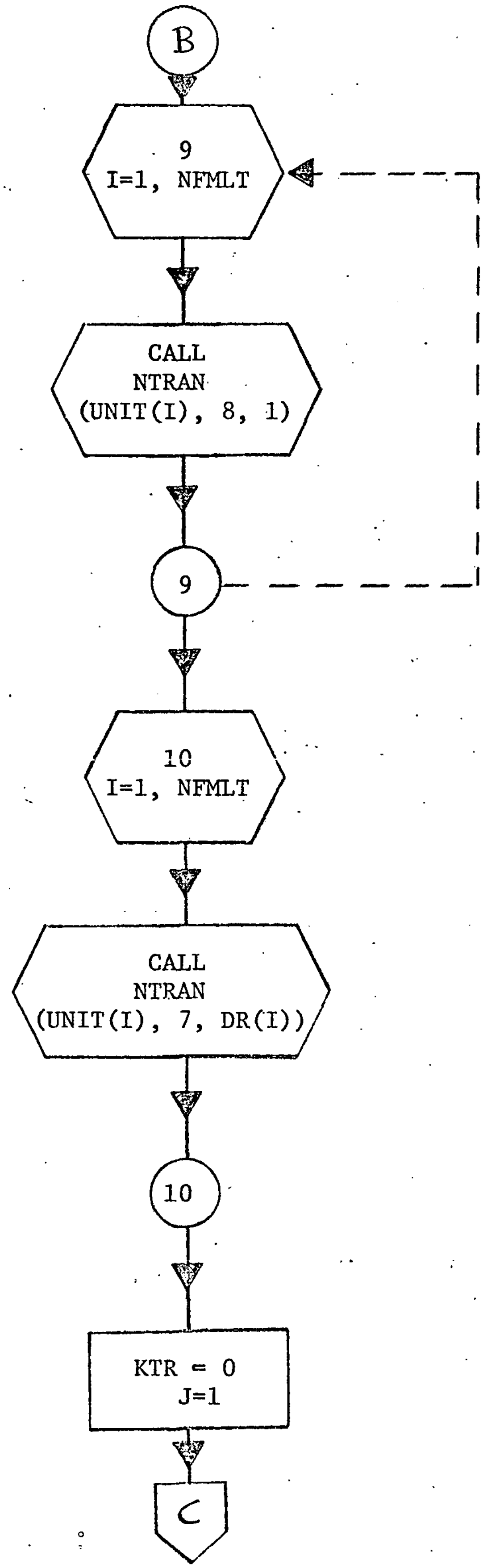


Fig. 10.5.1 continued

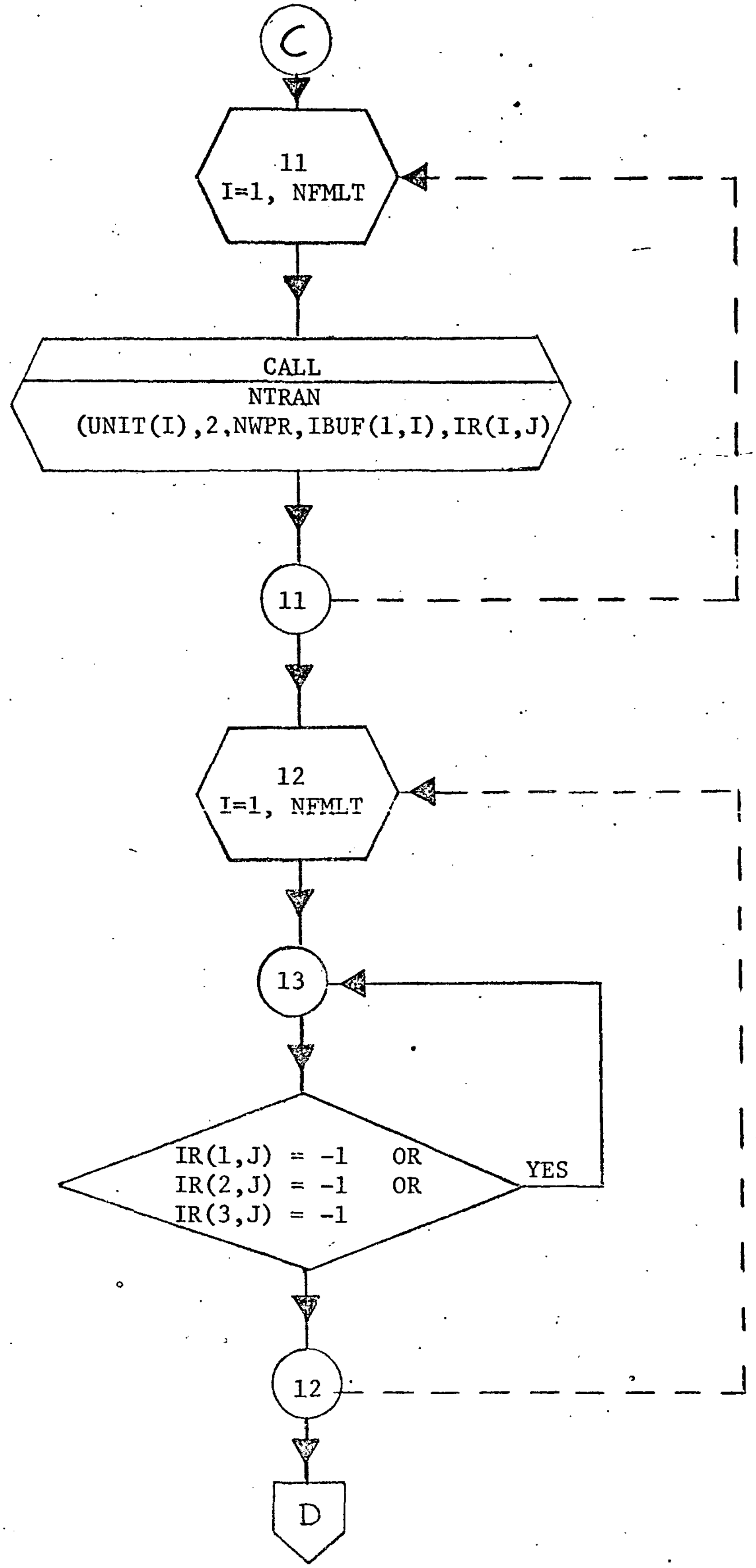


Fig. 10.5.1 continued

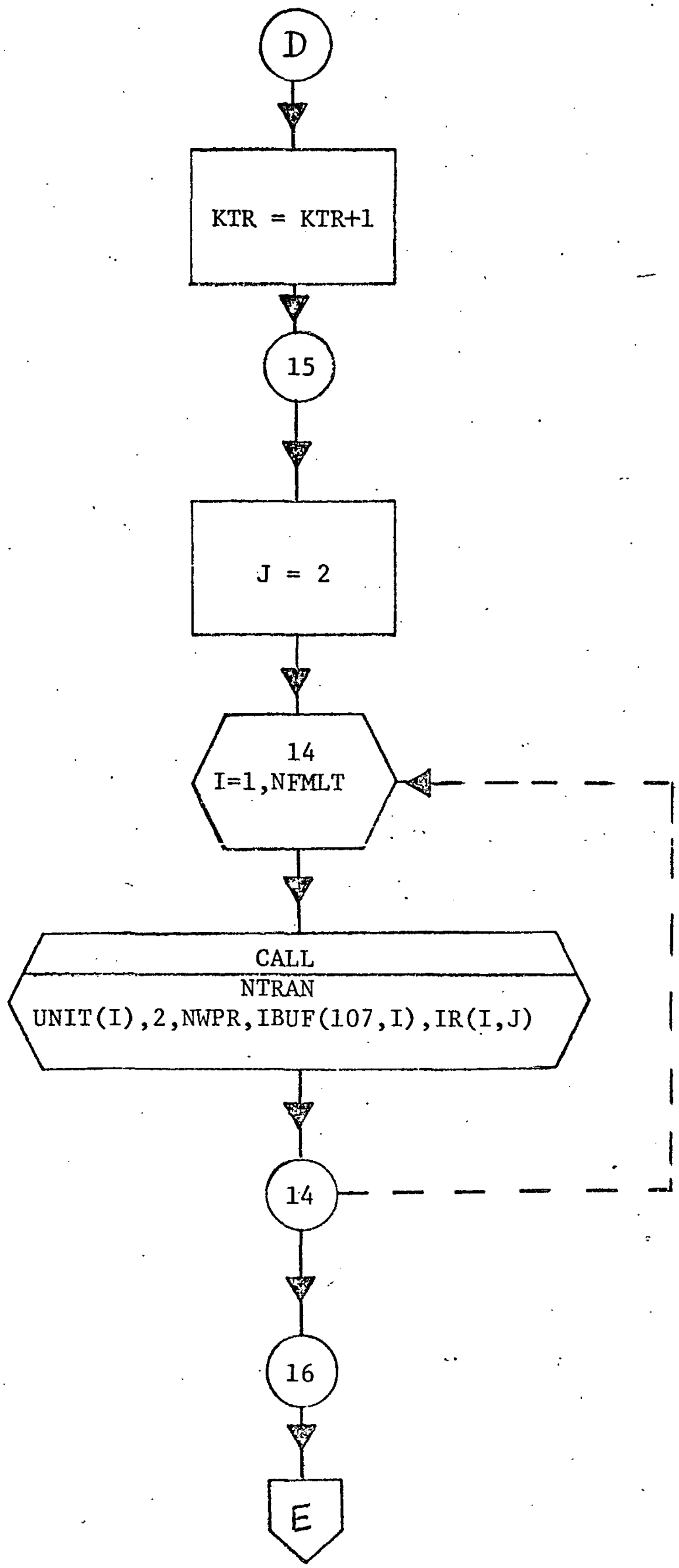


Fig. 10.5.1 continued

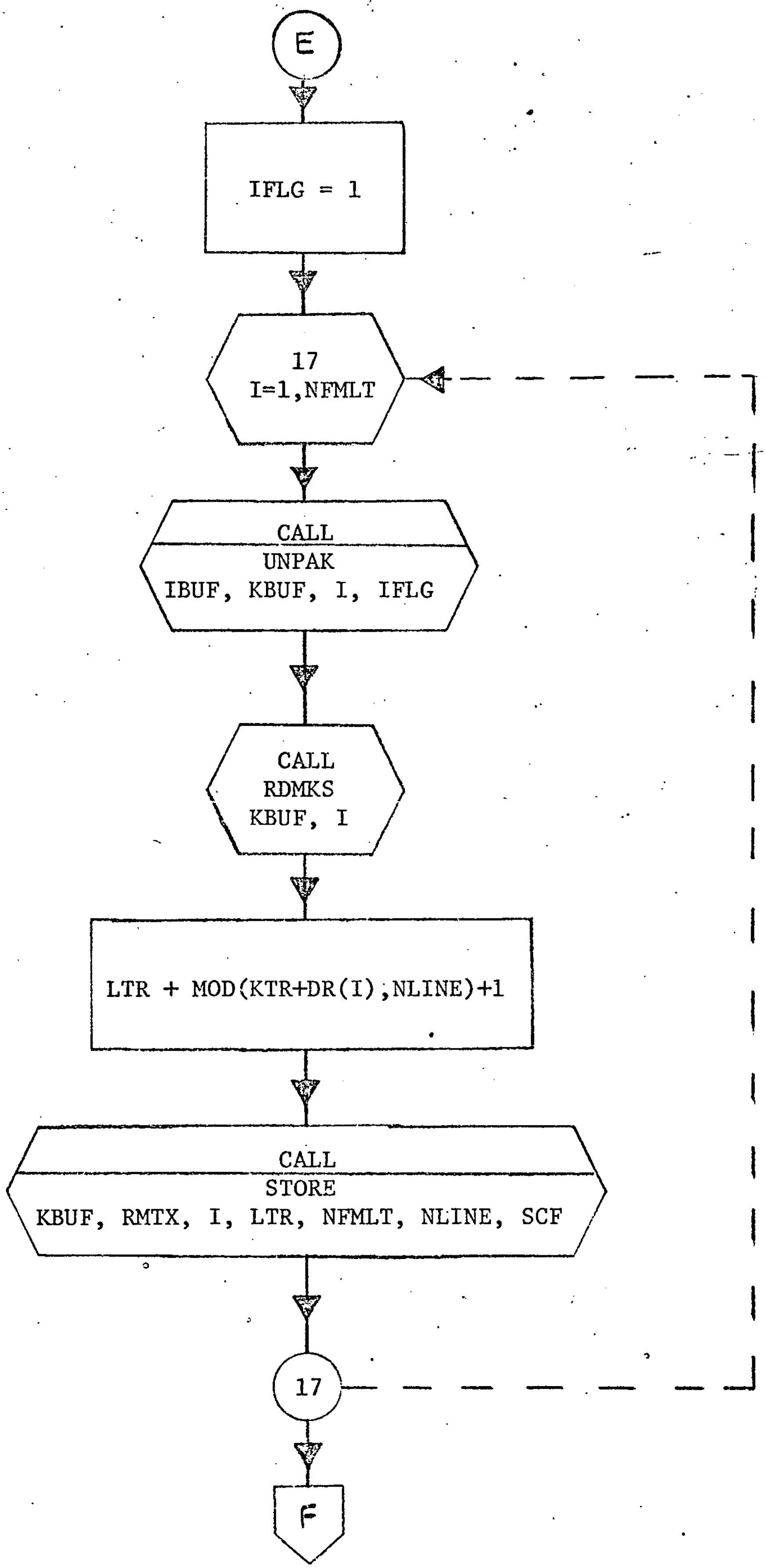


Fig. 10.5.1 continued

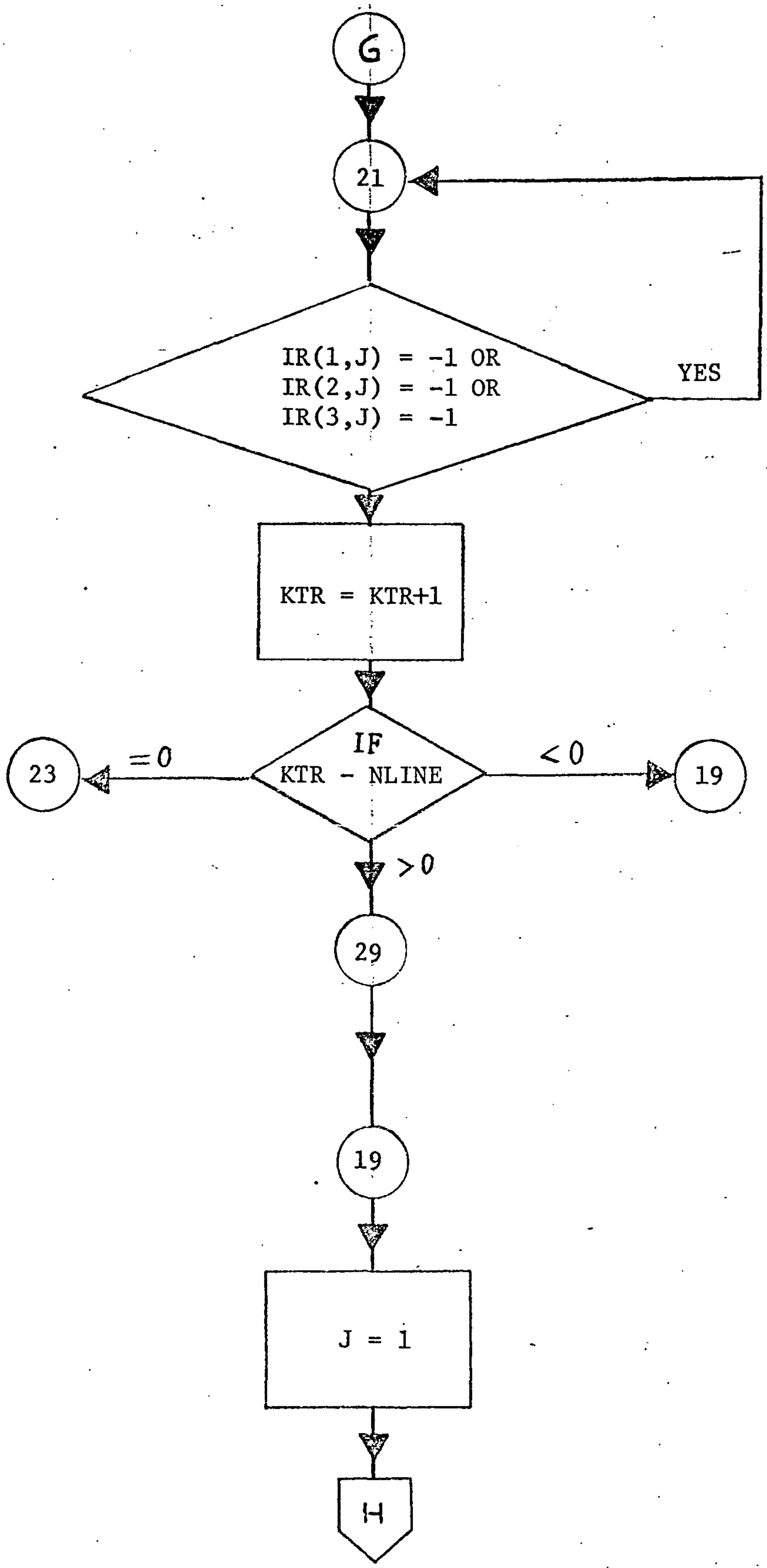


Fig. 10.5.1 continued

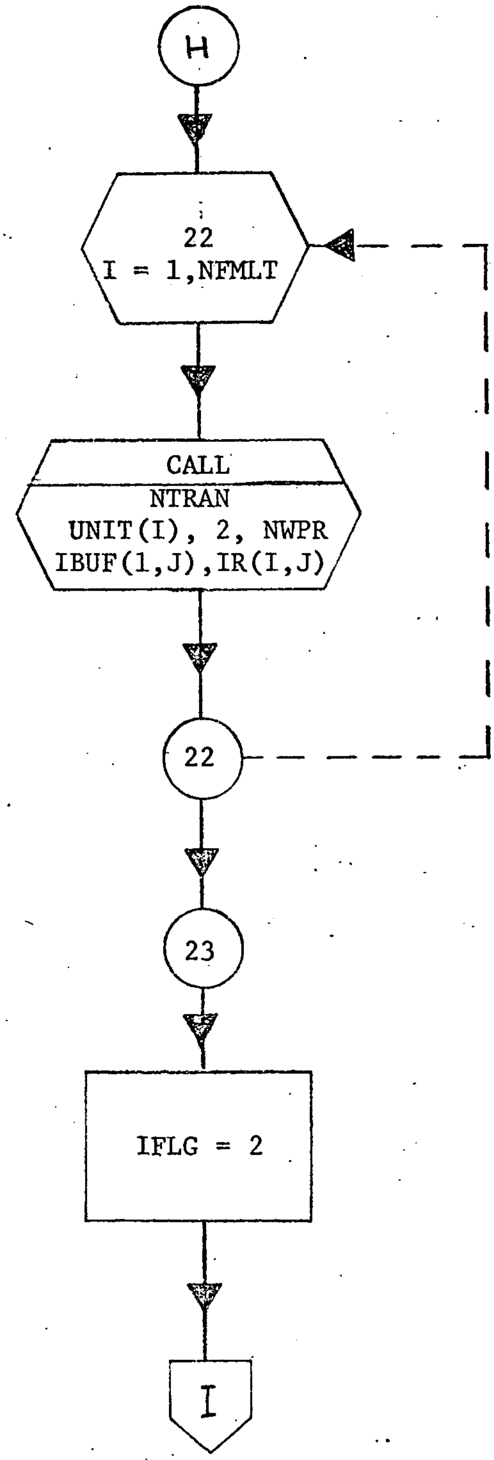


Fig. 10.5.1 continued

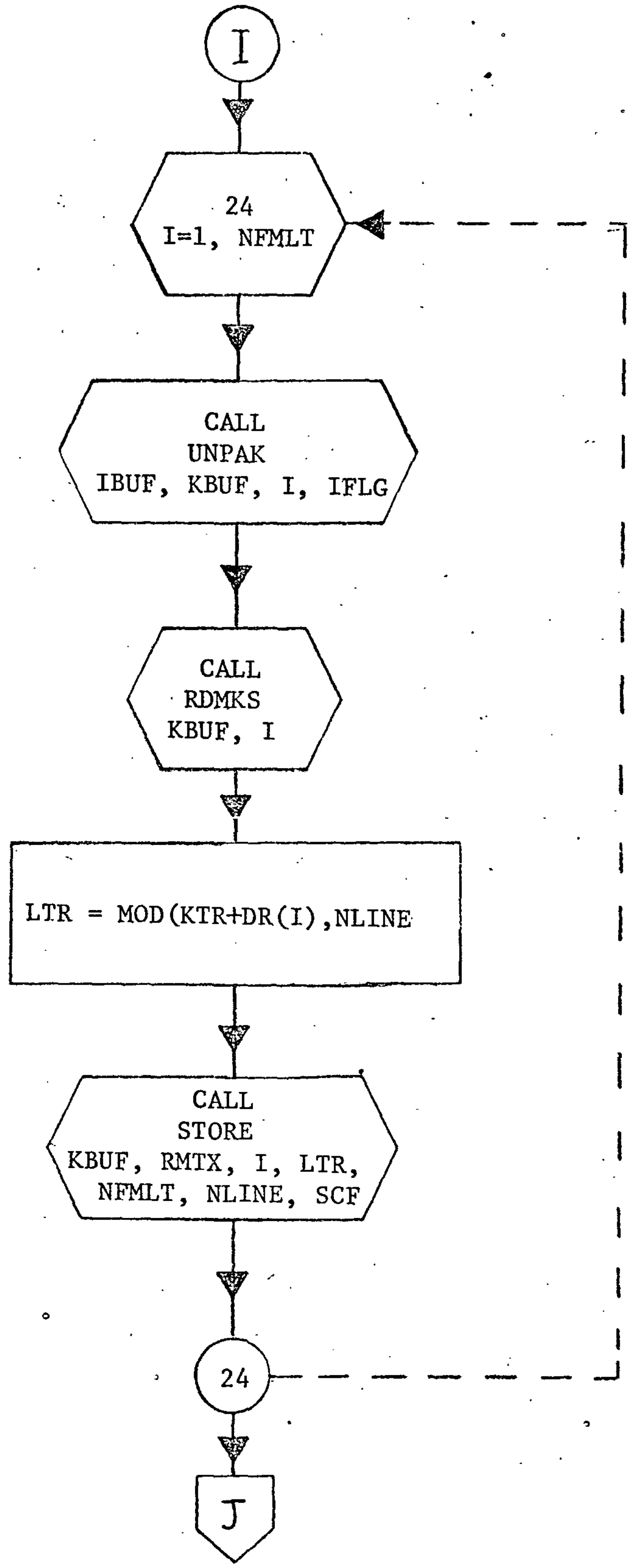


Fig. 10.5.1 continued

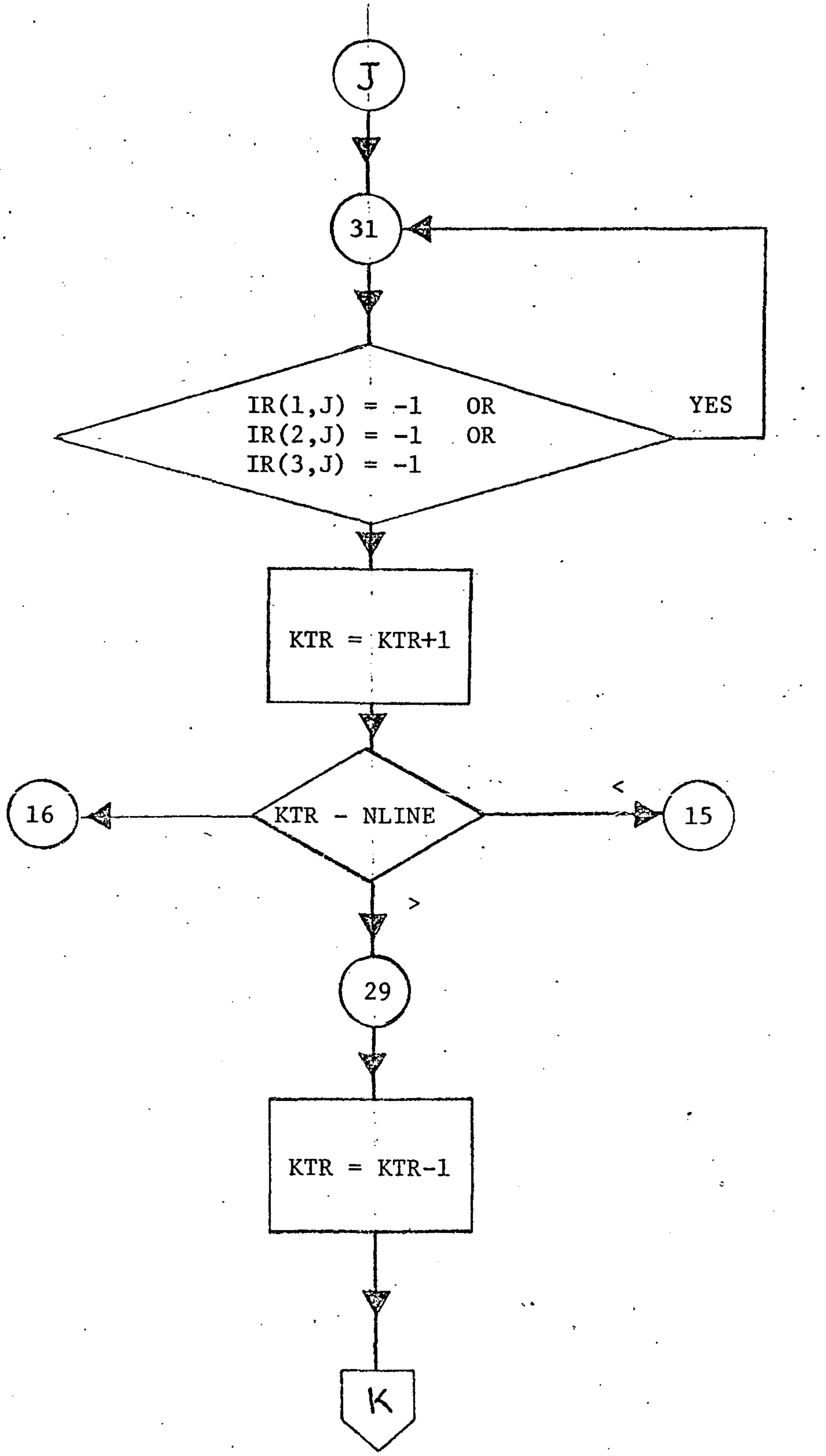


Fig. 10.5.1 continued

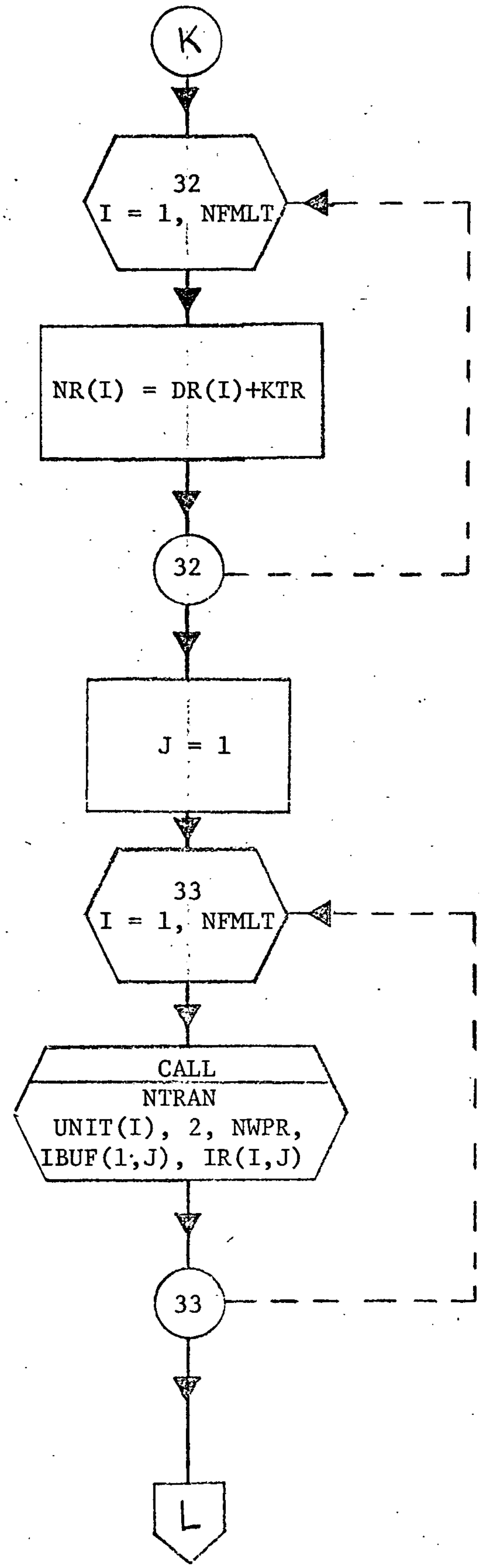


Fig. 10.5.1 continued

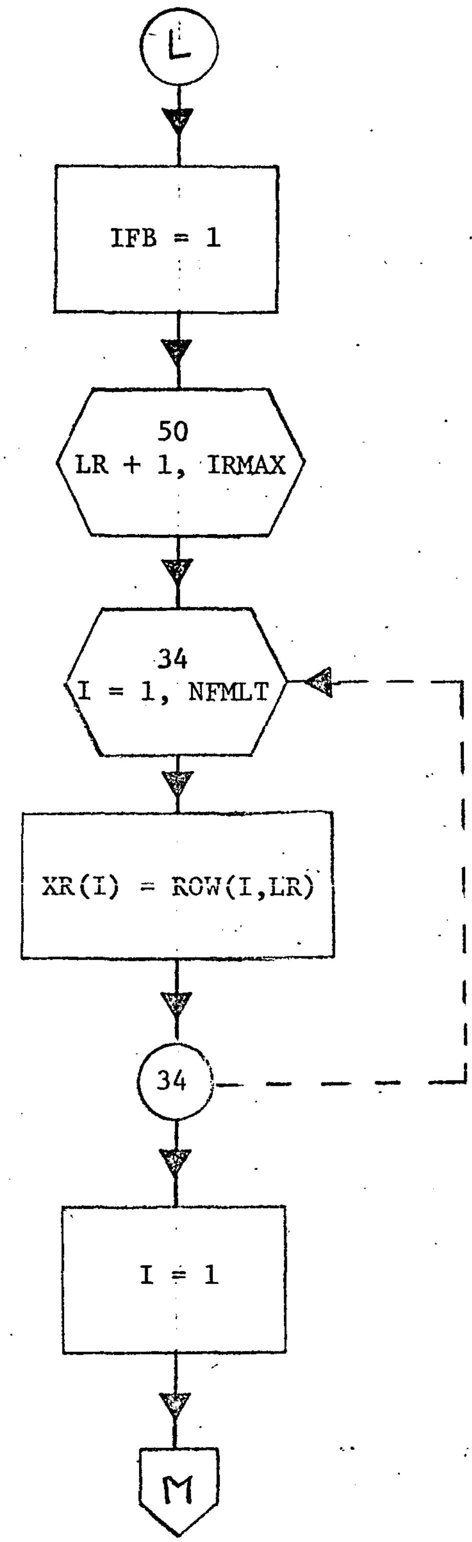


Fig. 10.5.1 continued

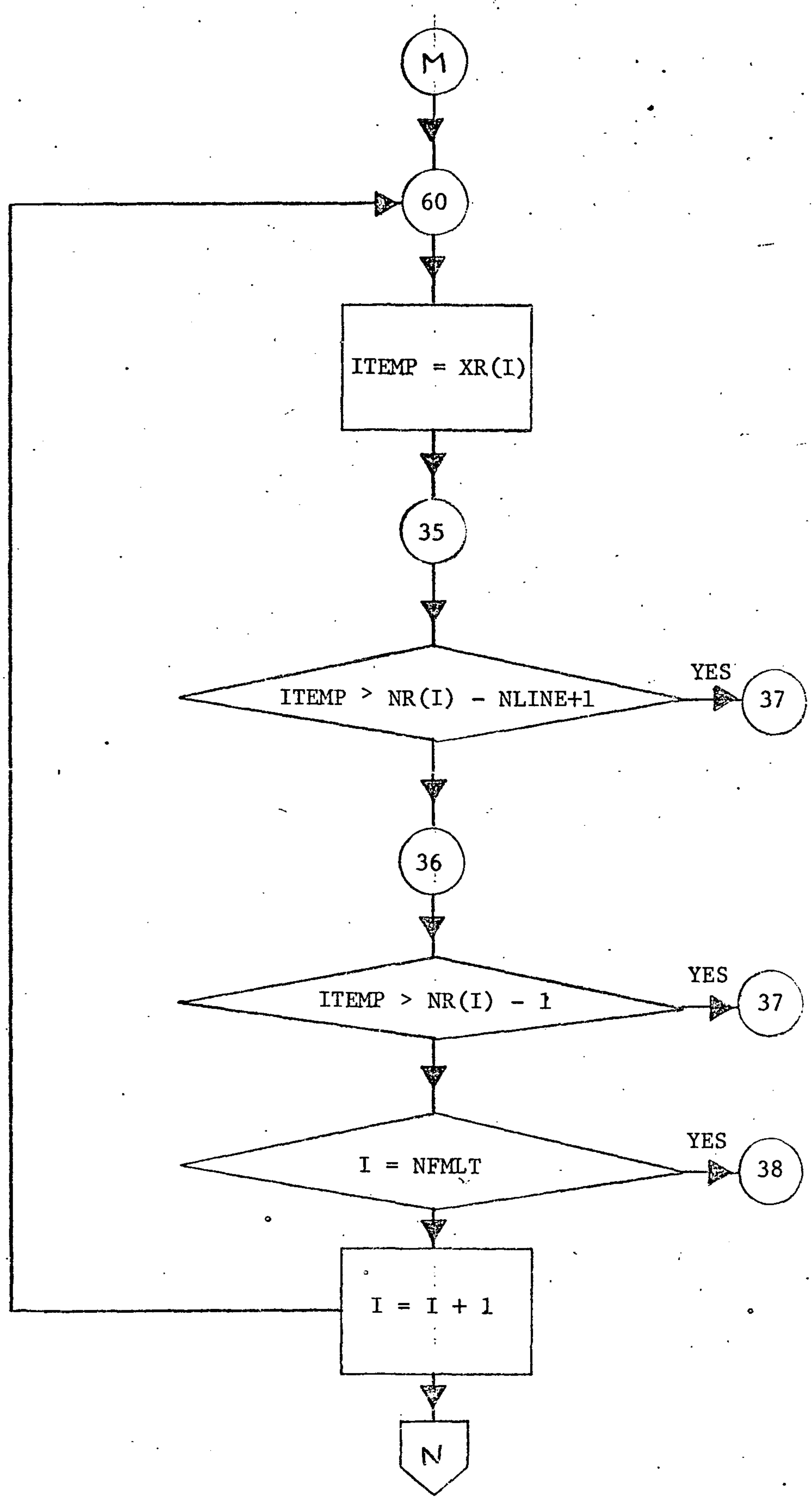


Fig. 10.5.1 continued

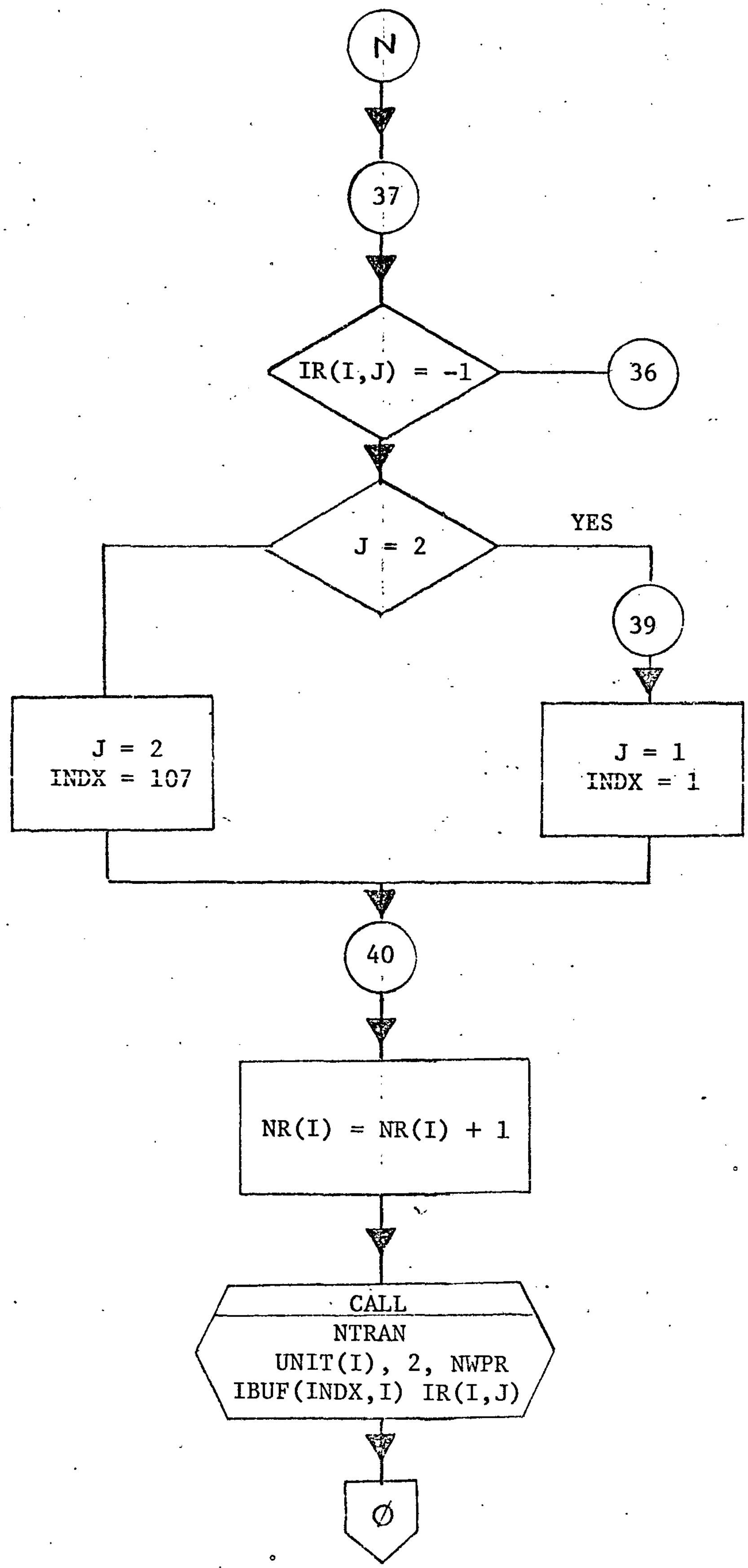


Fig. 10.5.1 continued

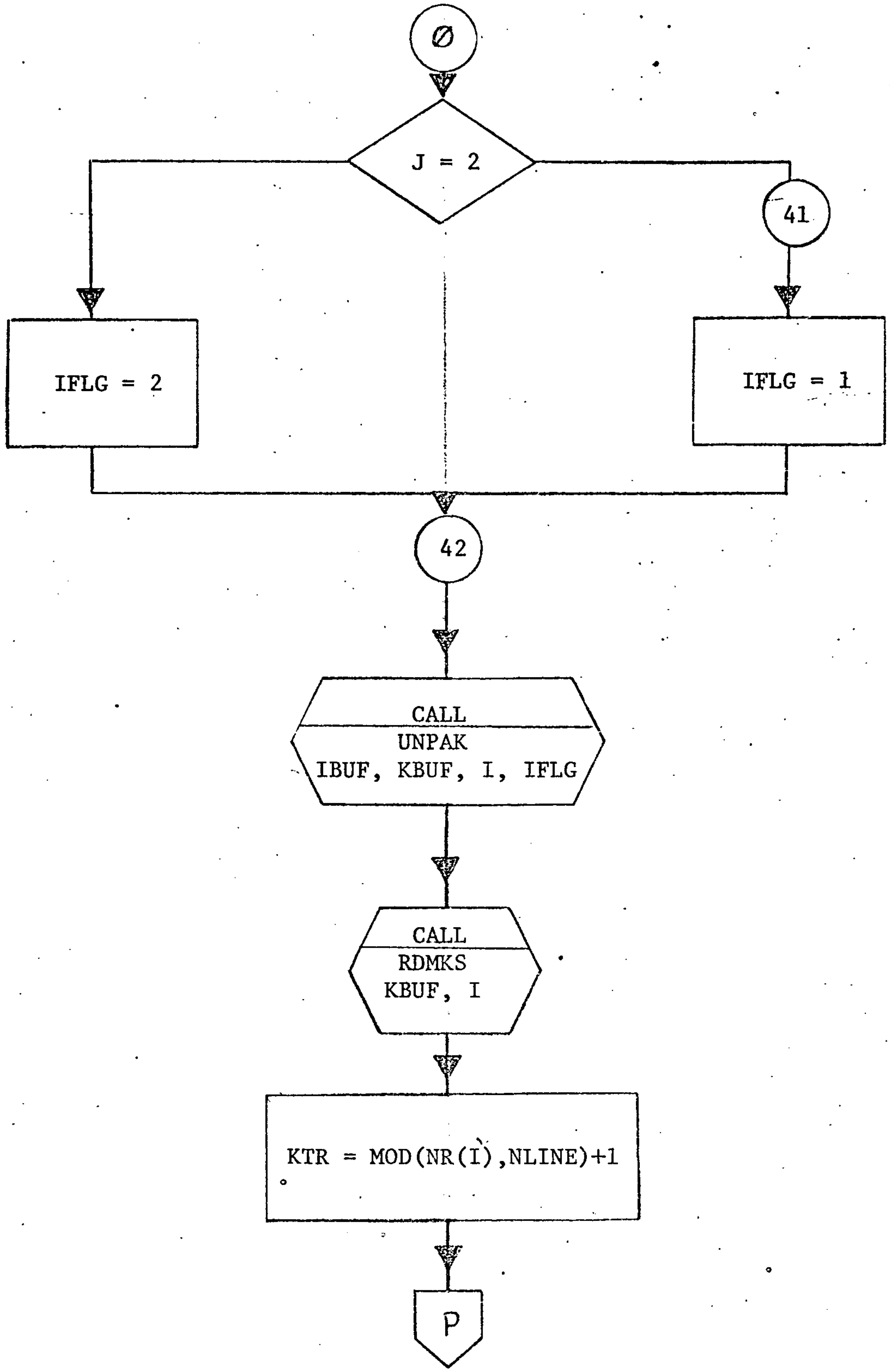


Fig. 10.5.1 continued

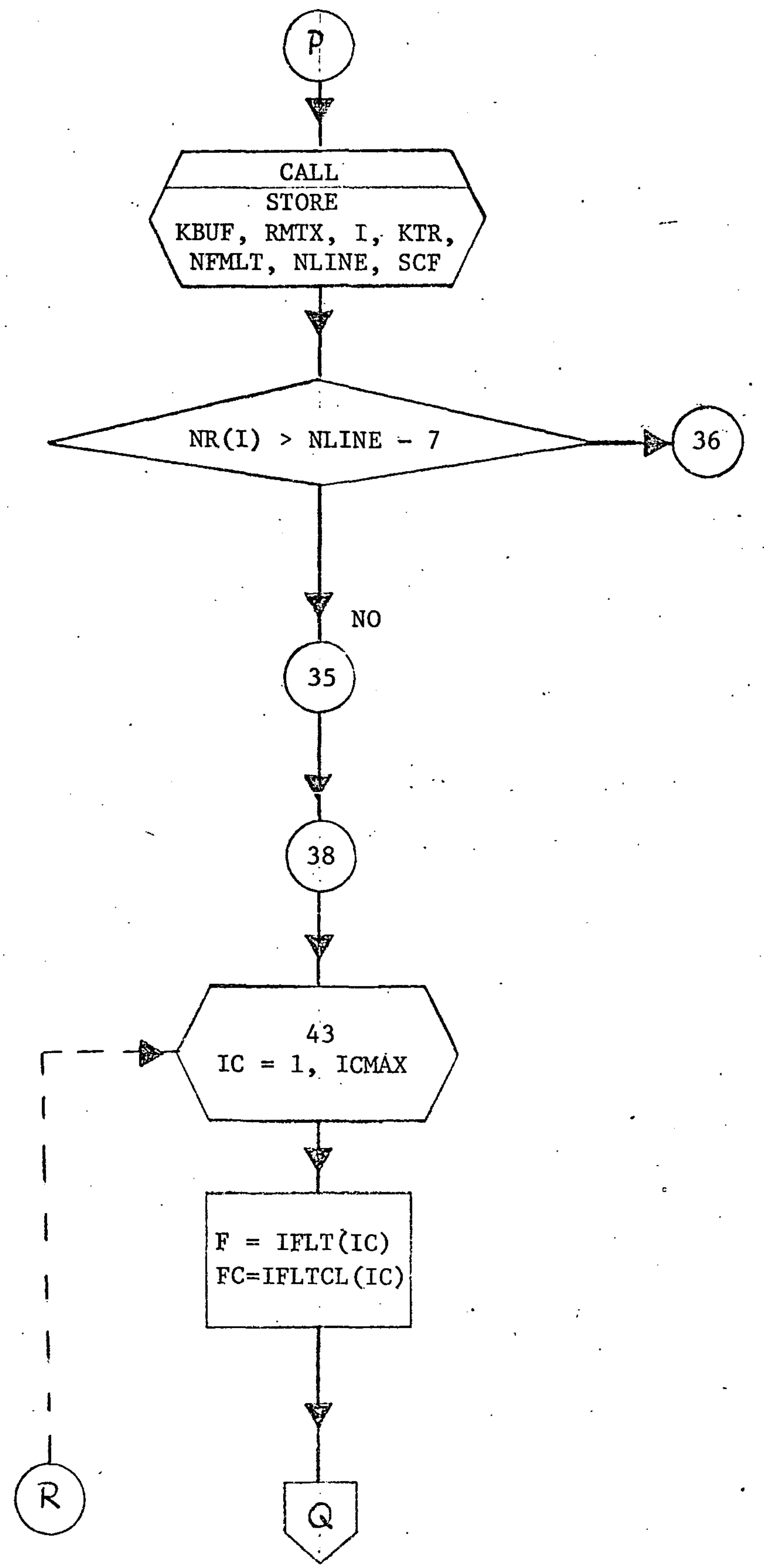


Fig. 10.5.1 continued

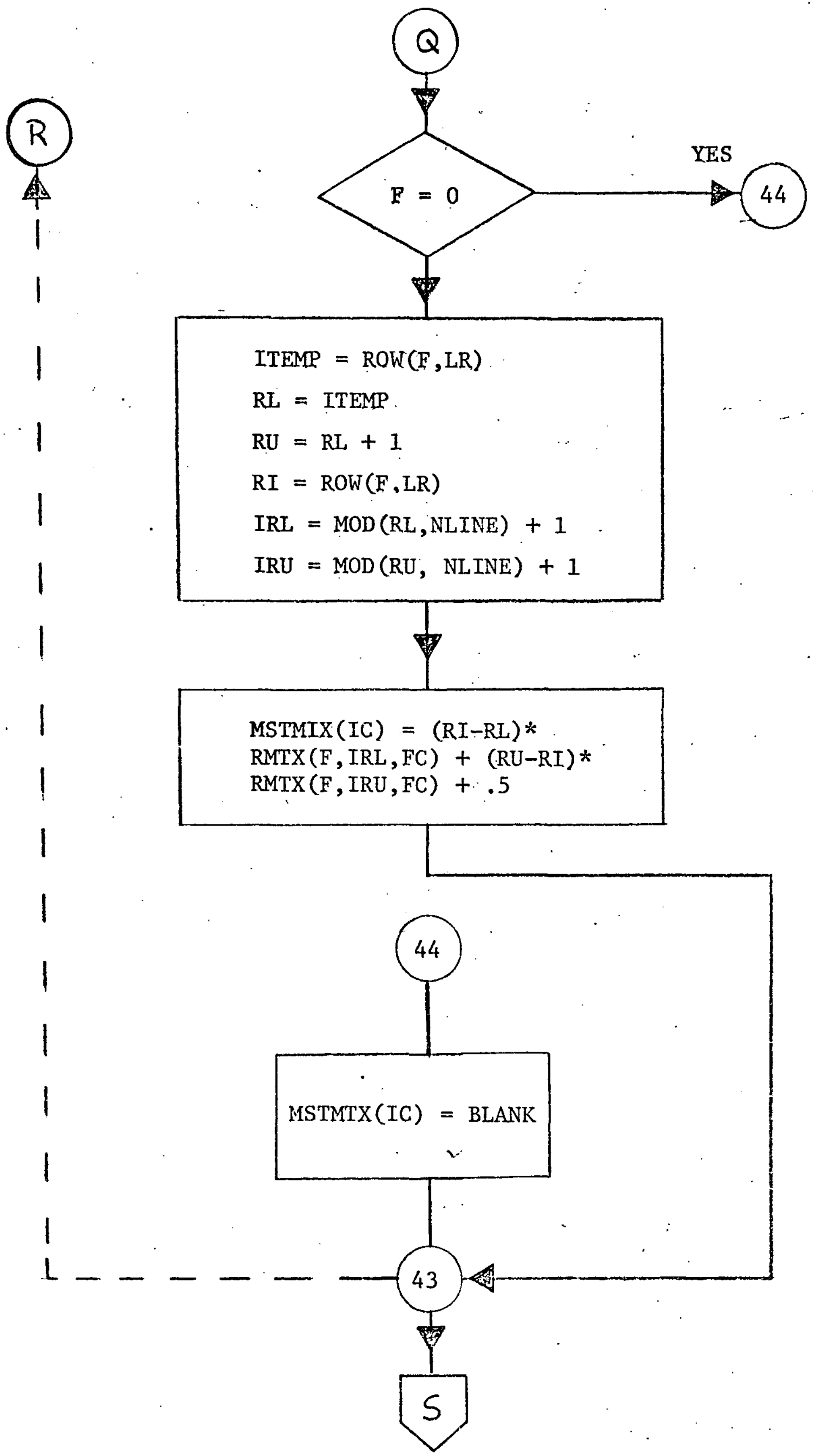


Fig. 10.5.1 continued

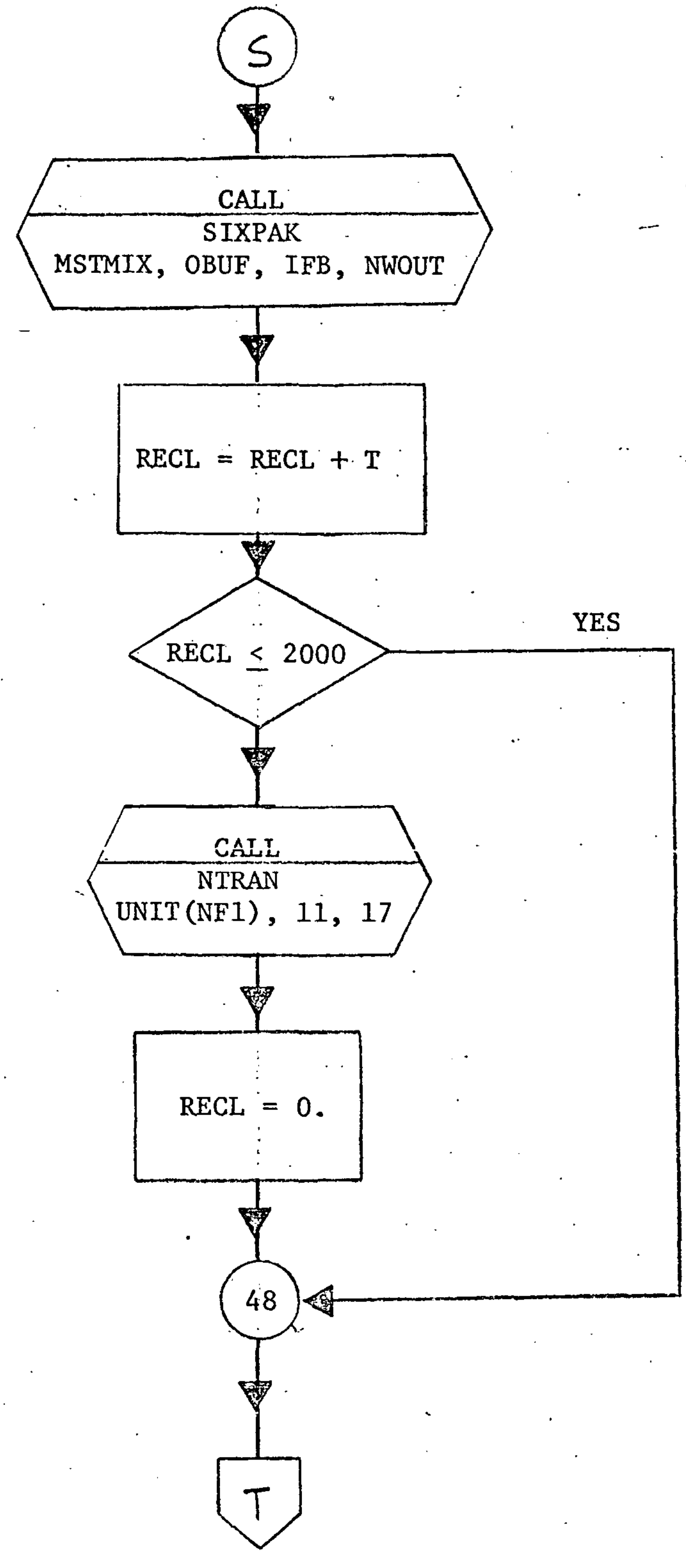


Fig. 10.5.1 continued

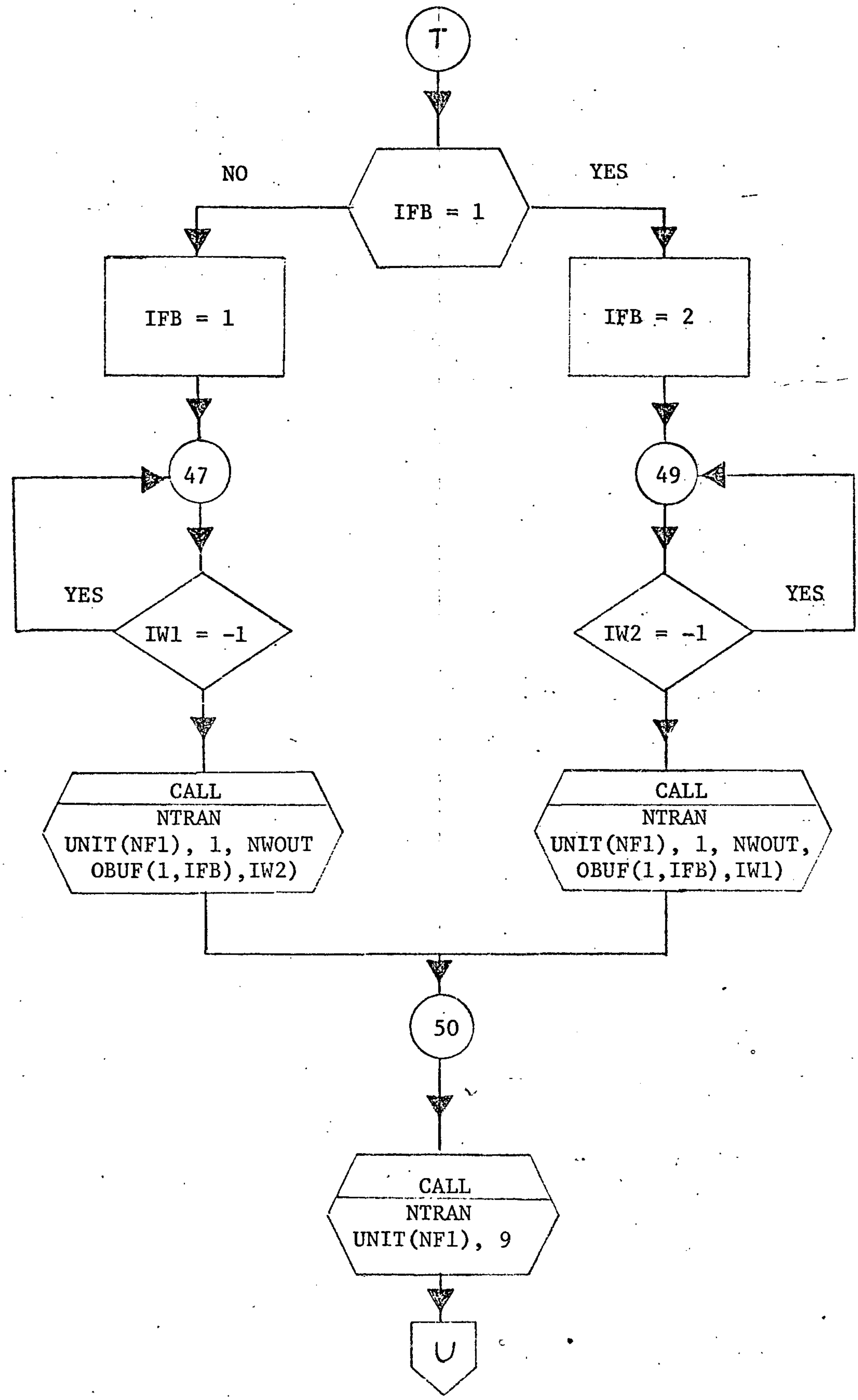


Fig. 10.5.1 continued

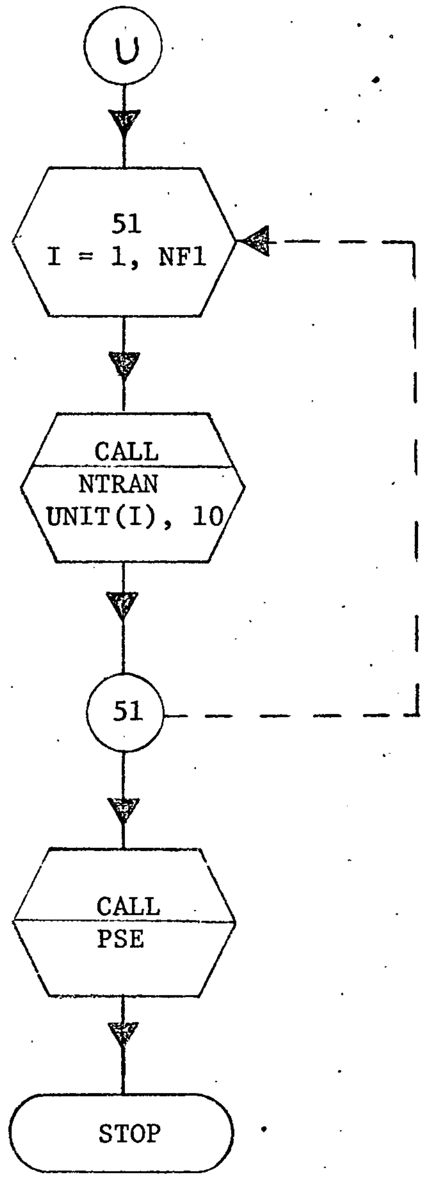


Fig. 10.5.1 (concluded)

DR - An array which contains the number of the first line to be joined for each framelet.

UNIT - An array whose values agree with the tape assignment e.g. UNIT(I), $1 \leq I \leq 3$ is the unit assignment for each tape containing framelet I and the 4th unit is used for the joined output.

IBUF - An array used for double buffering input from the tapes containing data.

IR - An array which receives the status word affecting the transmission of a read from tape.

RMTX - An array into which the data is stored after the data has been unpacked, drummarks removed and correction factors applied.

NR - An array which contains the line number of each line being joined.

LOW - An array which contains the row numbers to be joined for each framelet.

IFLT - An array whose values determines which framelet the row being joined belongs to.

IFLTCL - An array whose values determines which column the interpolation is to be done on in constructing a joined row.

XR - An array which contains the real part of the interpolated row number.

Table 10.5.2 SYMBOL DEFINITION TABLE - JOIN

KBUF - An array which is used in the process of unpacking the data, removing drum marks and applying correction factors.

MSTMTX - An array which contains the joined row from each of the framelets being considered.

OBUF - An array which contains the output data (joined row) in packed form.

NFMLT - The number of framelets being joined.

NWPR - The number of words per row.

NF1 - NFMLT+1, used for determining the number of tape units needed.

NLINE - An arbitrary value, determines the number of lines read into RMTX for each framelet to be joined.

IRMAX - Number of rows in output joined picture.

ICMAX - Number of columns in output joined picture.

NWOUT - Number of columns in output joined picture in packed form.

JROW - The difference between any two rows to be interpolated on; used in constructing ROW.

NTDIM - The number of framelet edges to be joined together, NFMLT-1.

NRPFLT - The number of rows per framelet to be joined biased by 150 because of the notch data included in a framelet.

RECL - The accumulated length of the output tape; used to determine if the output picture is too big for one reel; presently tested against 2000 while actual tape length is 2450.

T - The length of one record of a joined row, data plus inter-record gap in feet.

Table 10.5.2 SYMBOL DEFINITION TABLE - JOIN (Continued)

KTR -	A counter used to determine if NLINES have been inputted into RMTX initially.
IFLG -	A flag used to determine which buffer of a double buffer contains the data to be unpacked, drummarks removed from and correction factors applied to.
LTR -	A counter used to determine which line of input goes into which line of RMTX via modular arithmetic on NLINE.
ITEMP -	The integral portion of the interpolated row number.
F -	Framelet number of framelet being joined.
FC -	Framelet column of framelet being joined.
RL -	Integral portion of the interpolated row number.
RU -	RL+1, next row after interpolated row.
RI -	Real part of the interpolated row.
IRL -	RL (modulo NLINE)+1, used to determine the correct row in RMTX.
IRU -	RU (modulo NLINE)+1, used to determine the correct row in RMTX.
IFB -	Flag used in double buffering output.
IW1,IW2 -	Status words used to determine transmission status of data on out output.
SCF -	Smoothed correction factors.

Table 10.5.2 SYMBOL DEFINITION TABLE - JOIN (Concluded)

10.5.3.1 Description of subroutine RWMSH (row mesh)

IDENTIFICATION

Name	-	RWMSH
Author/Date	-	R.L. Wendt/June 1970
Organization	-	ASR
Machine Identification	-	UNIVAC 1108
Source Language	-	FORTRAN V

PURPOSE/METHOD

This subroutine generates an array containing the row numbers of each framelet to be used in joining the framelets together. It linearly interpolates on the match points which are read in from data cards. Its output is stored in ROW whose dimensions are (NFMLT, IRMAX).

USAGE

Calling sequence:

CALL RWMSH (NFMLT, JROW, NRPFLT, ROW, NTDIM)

The calling variables are described in Table 10.5.2.

SUBROUTINE REQUIREMENTS

The actual interpolation is done by subroutine MATINT. A detailed flow chart of MATINT is shown in Fig. 10.5.3.1.2.

Calling sequence:

CALL MATINT (I,J,MATCH,R,L,N)

where

I -- The number of the left framelet being joined.

J -- A real number, equal to the row being transfered to MATINT.

MATCH -- A real number, the interpolated result from MATINT.

R -- An integer array consisting of right and left match points,
L -- generated by SKINNY

N -- An integer, indicating which framelet pair is under consideration.

DETAILED FLOW CHART

See Fig. 10.5.3.1.1.

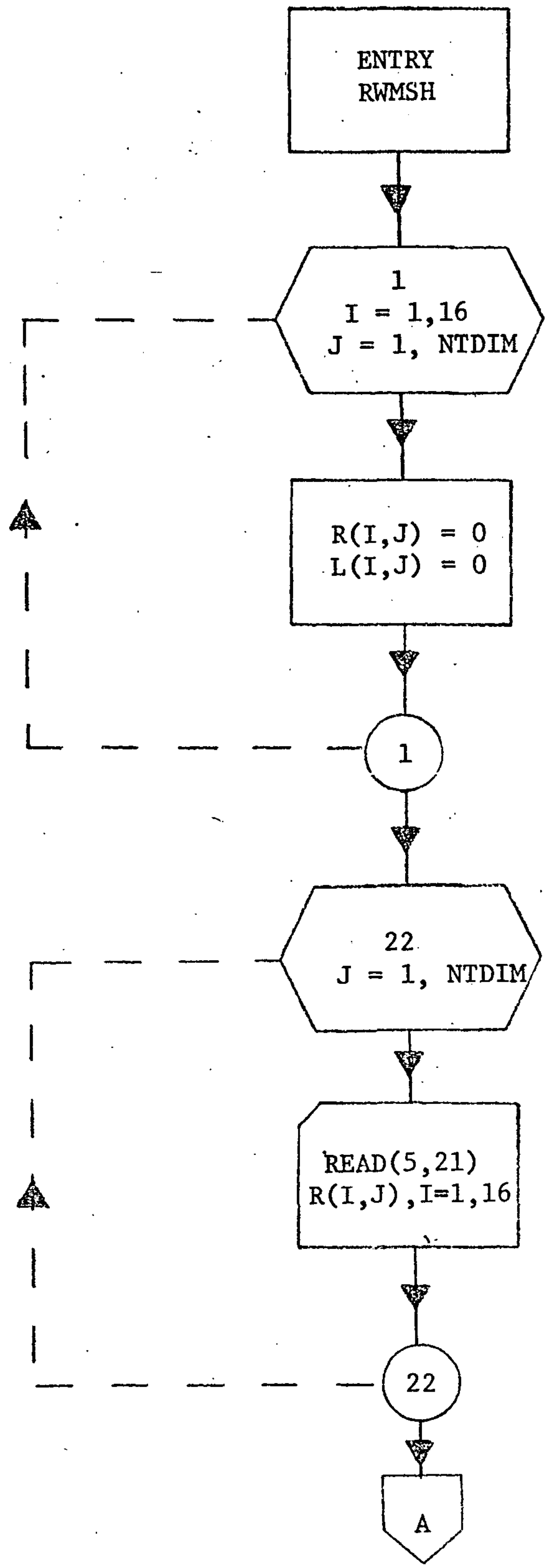


FIG. 10.5.3.1.1 DETAILED FLOW CHART - RWMSH

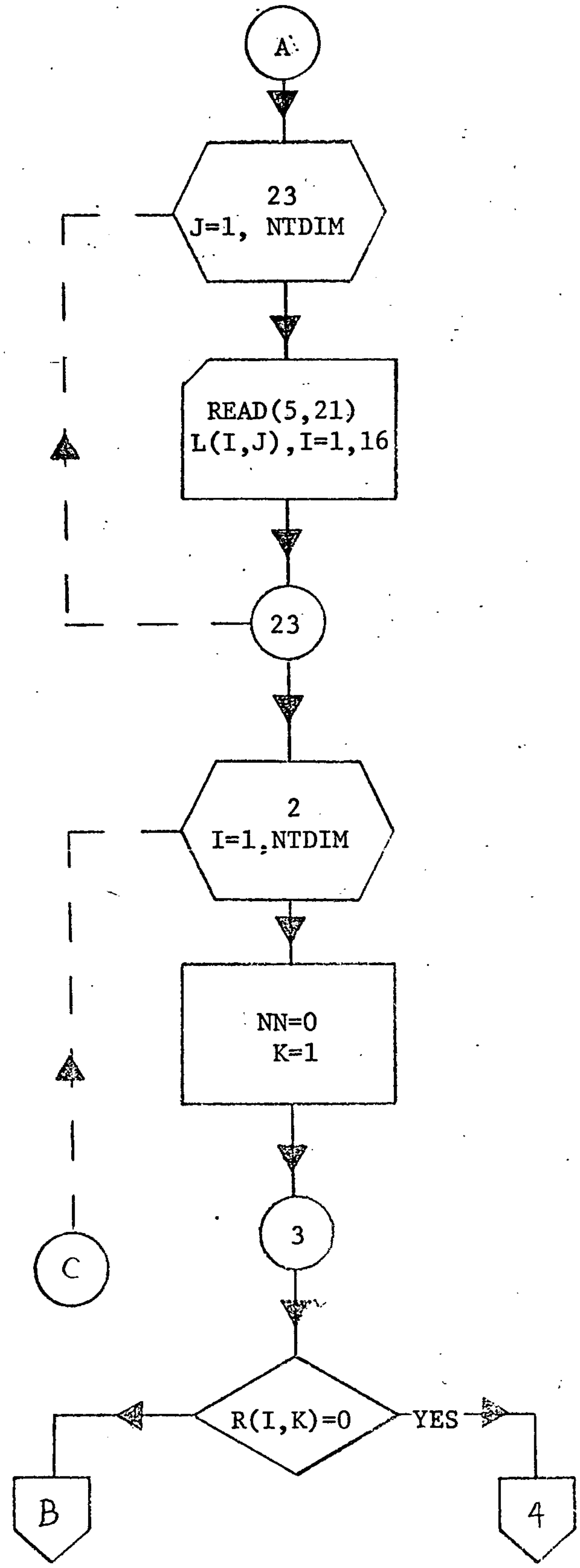


FIG. 10.5.3.1.1 DETAILED FLOW CHART - RWMSH (Continued)

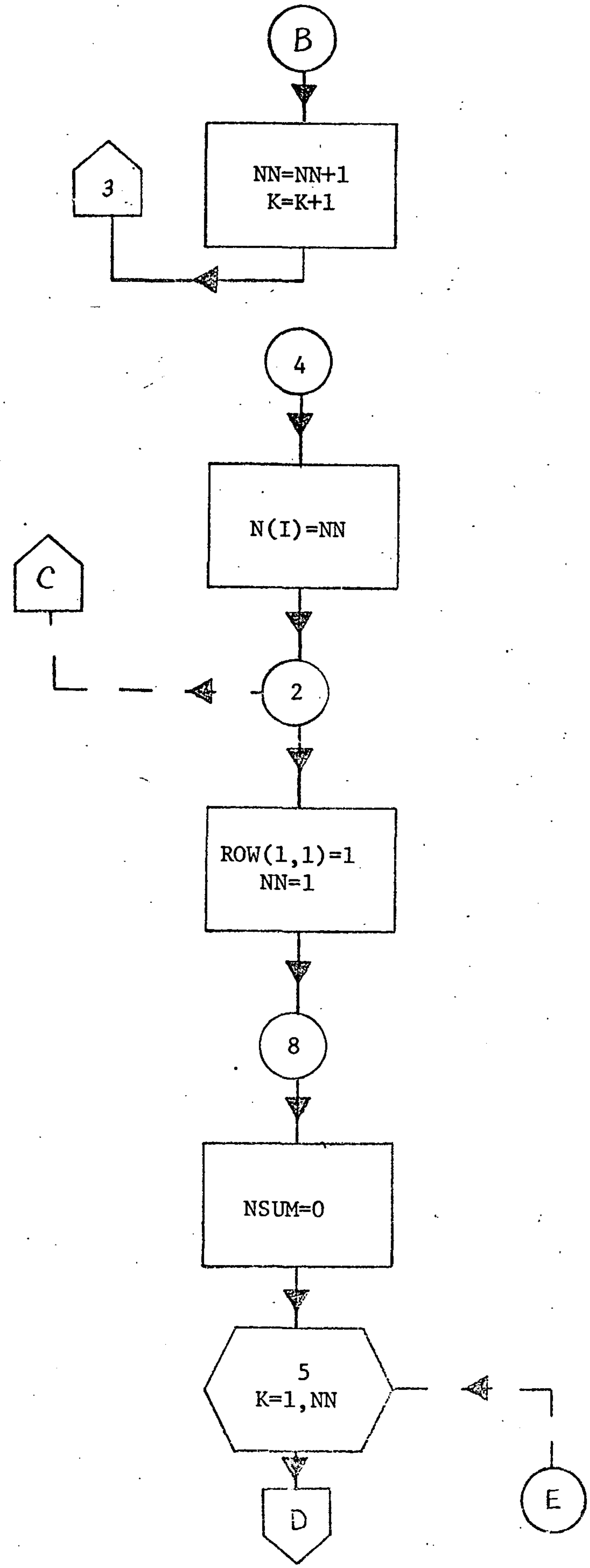


FIG. 10.5.3.1.1 DETAILED FLOW CHART - RWMSH (Continued)

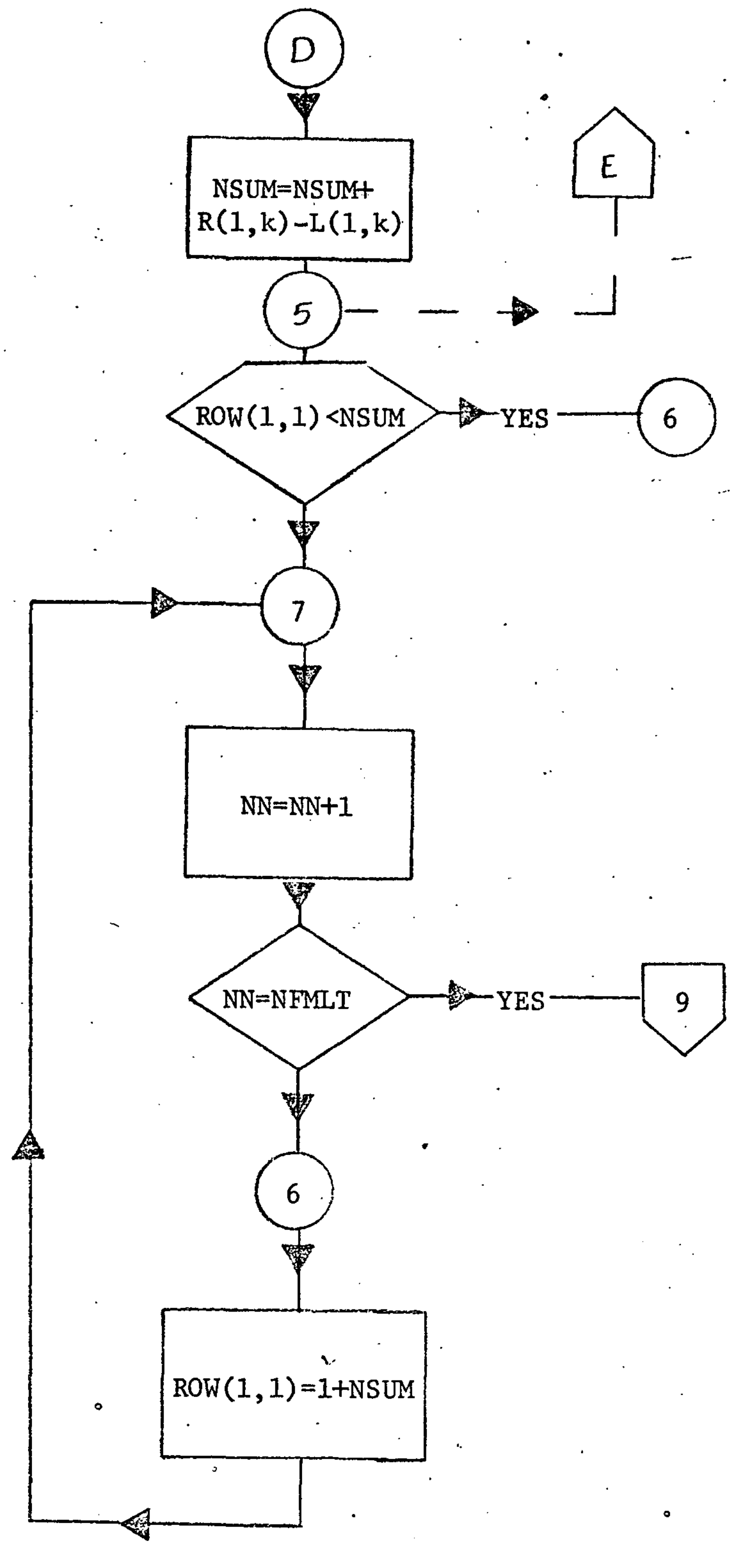


FIG. 10.5.3.1.1 DETAILED FLOW CHART - RWMSH (Continued)

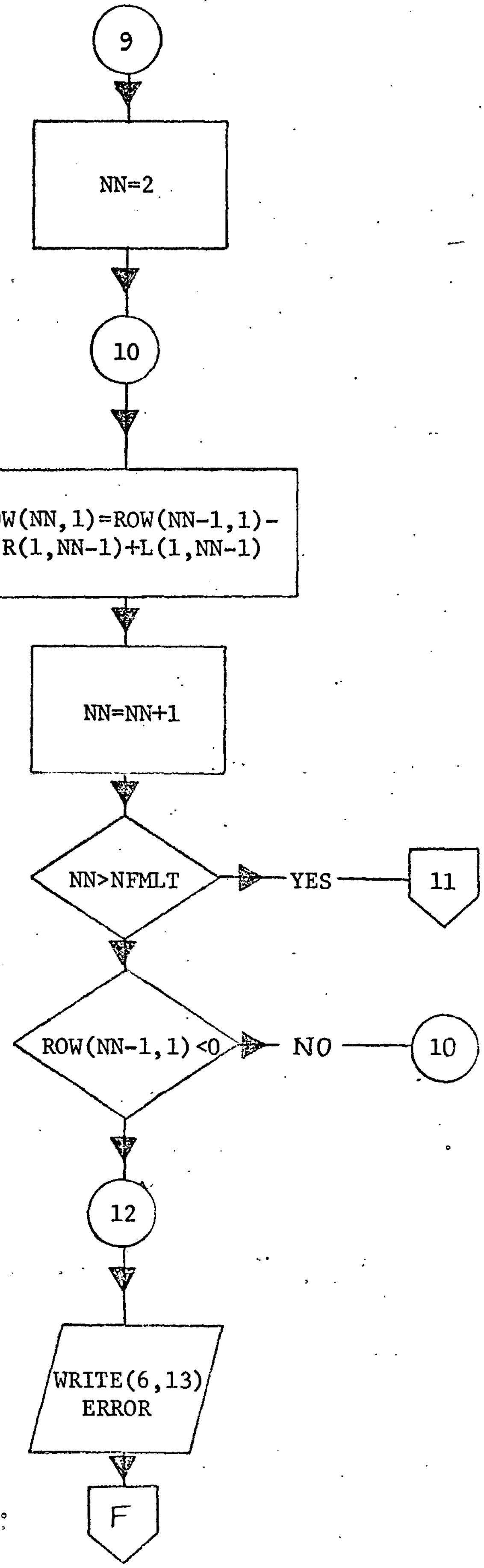


FIG. 10.5.3.1.1 DETAILED FLOW CHART - RWMSH (Continued)

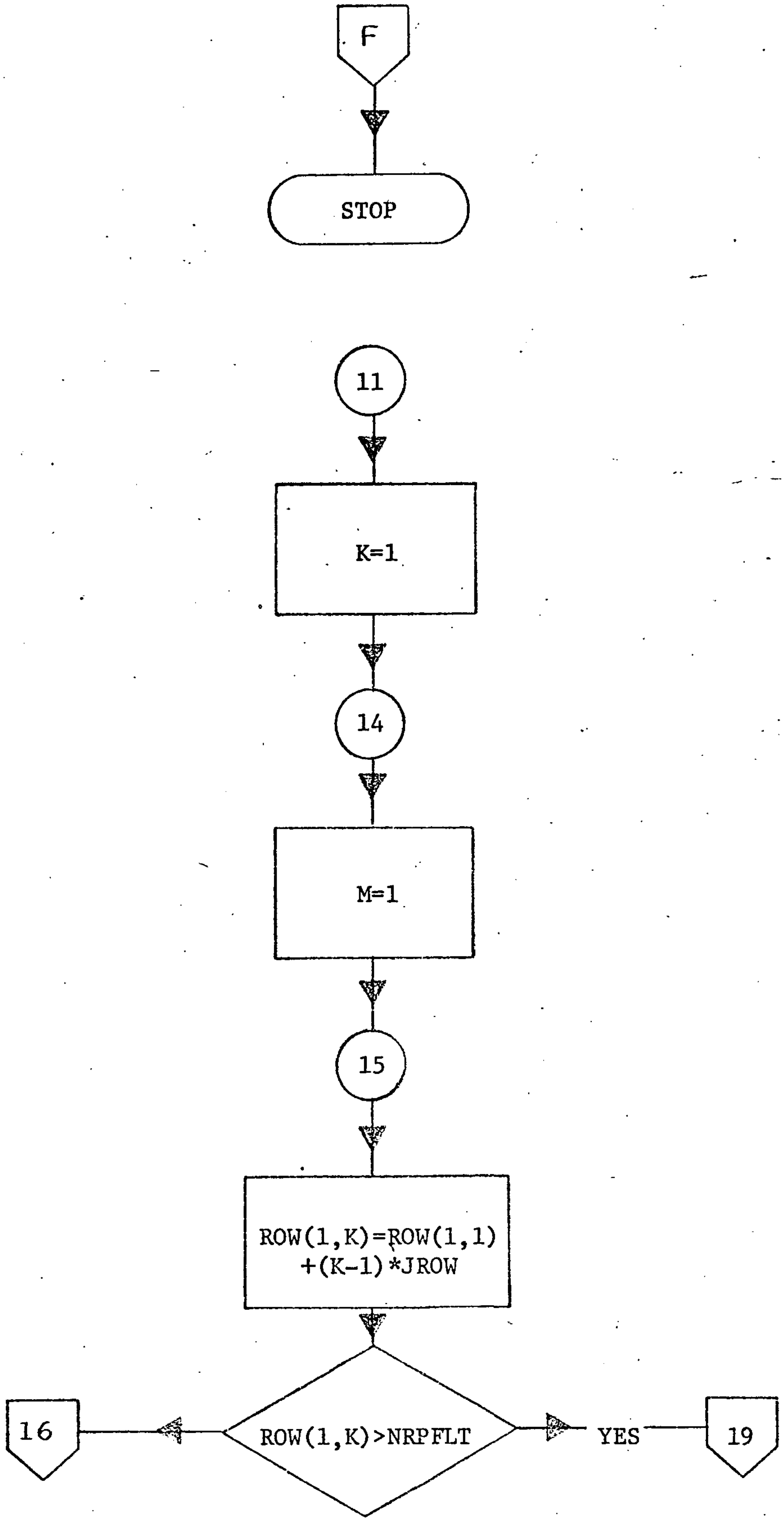


FIG. 10.5.3.1.1 DETAILED FLOW CHART - RWMSH (Continued)

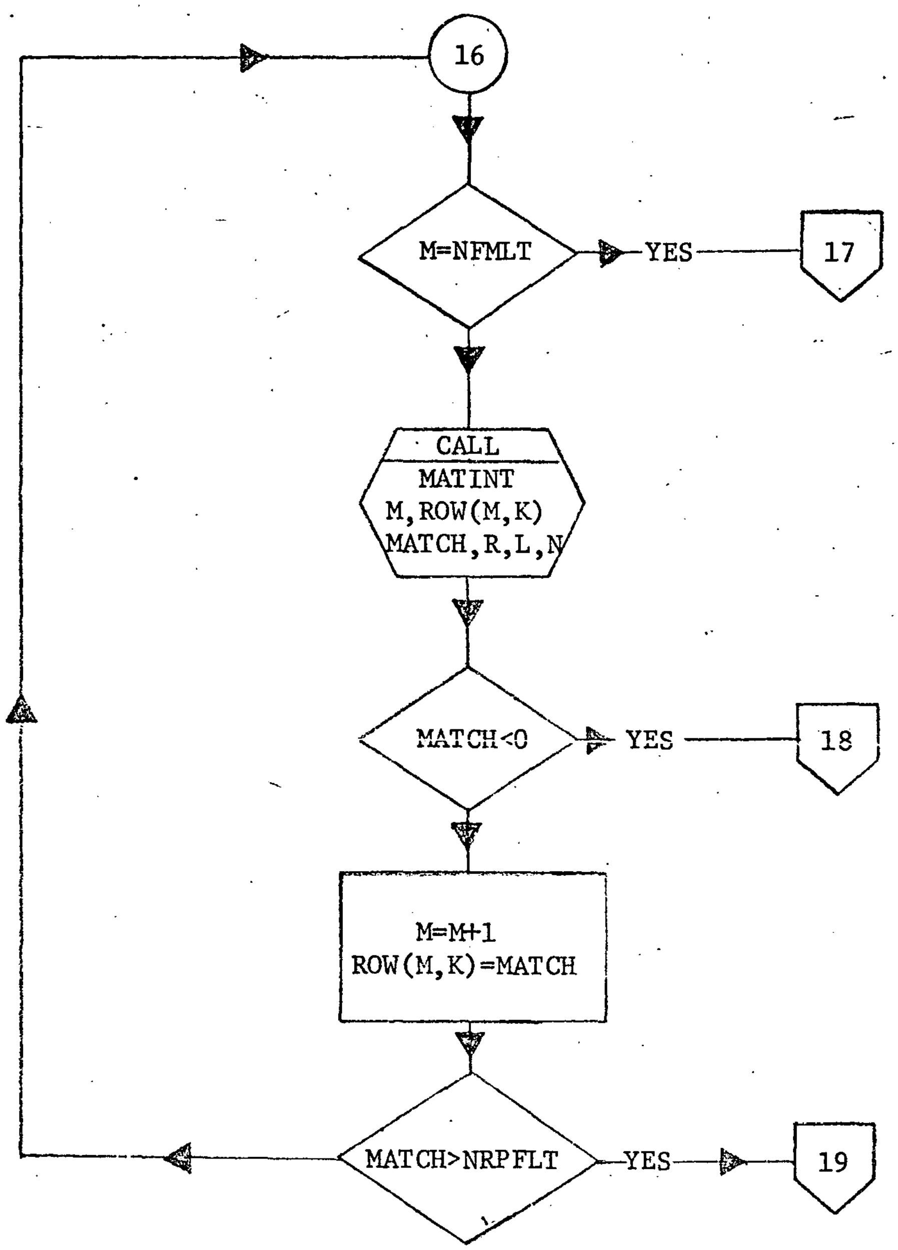


FIG. 10.5.3.1.1 DETAILED FLOW CHART - RWMSH (Continued)

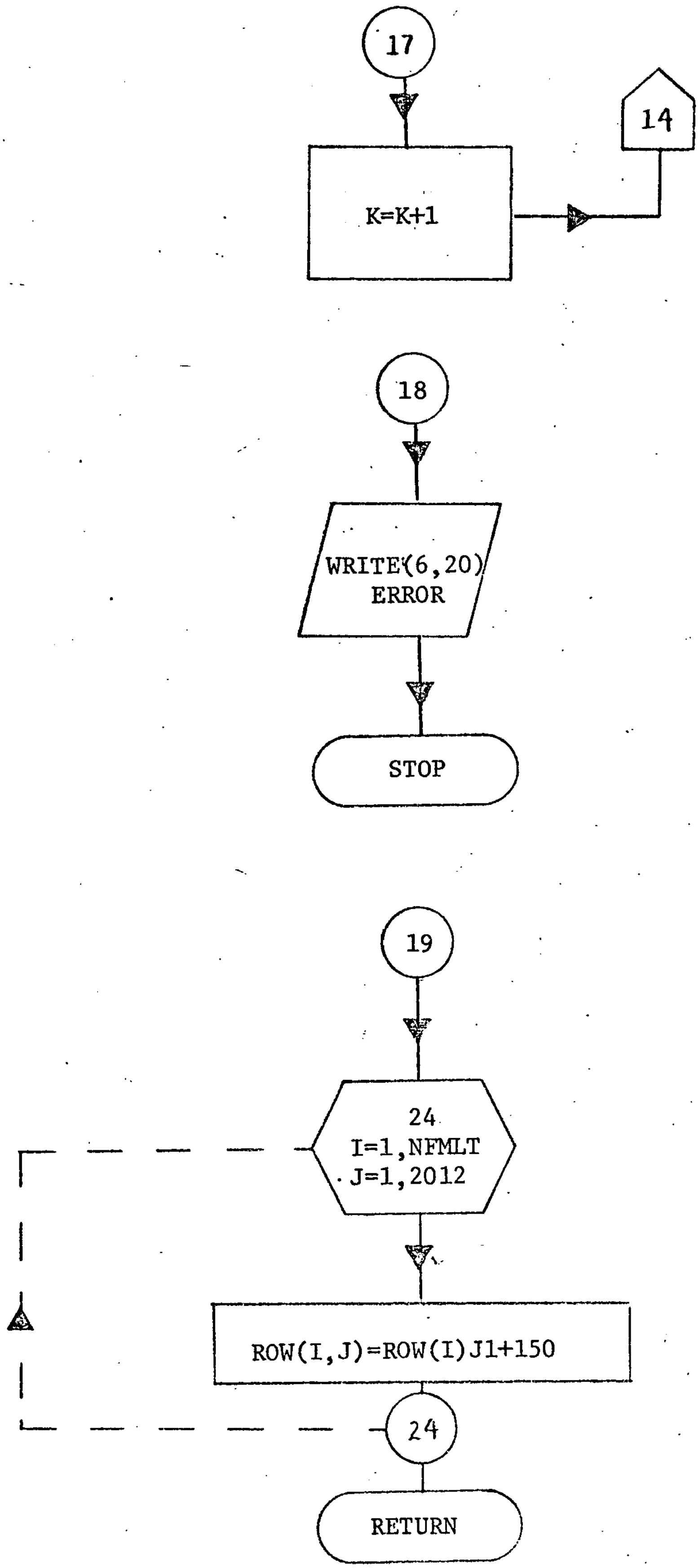


FIG. 10.5.3.1.1 DETAILED FLOW CHART - RWMSH (Concluded)

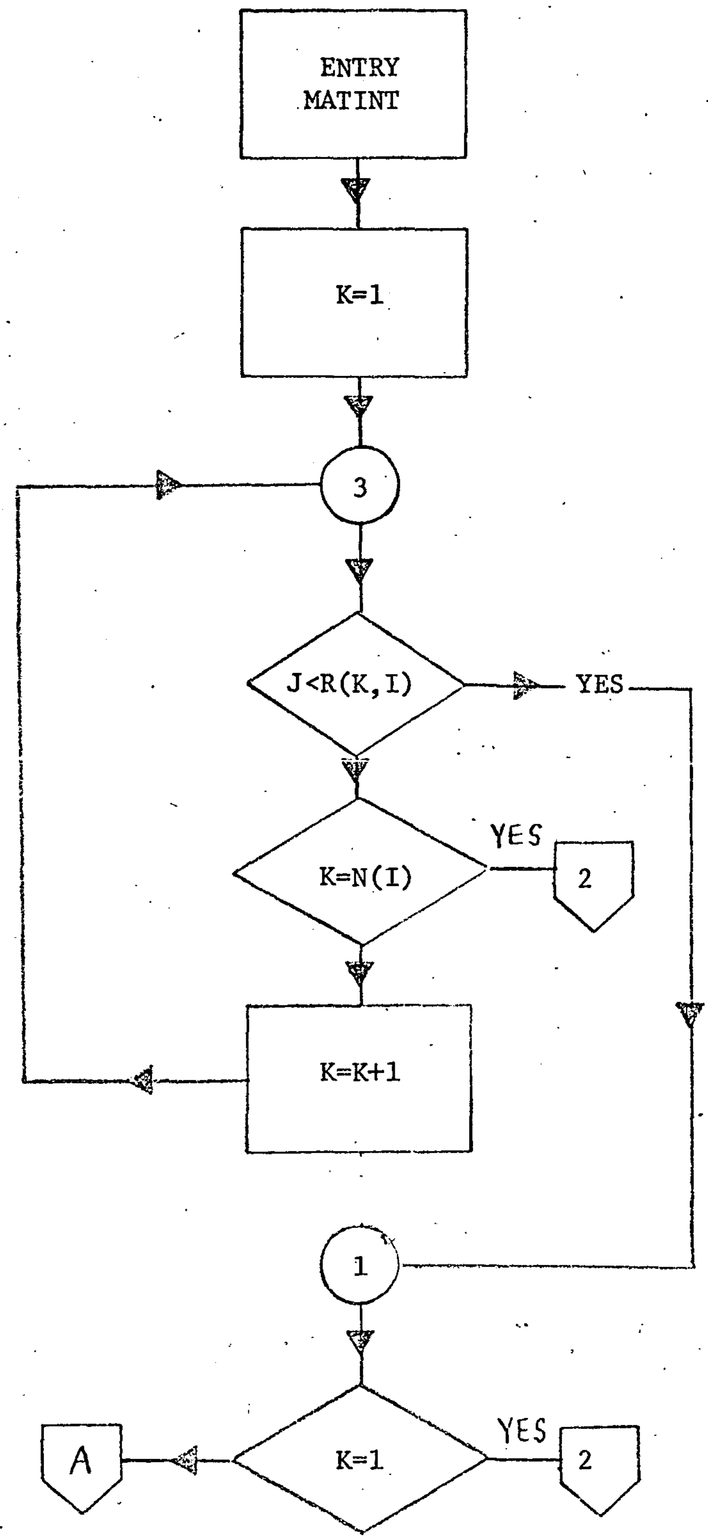


Fig. 10.5.3.1.2 DETAILED FLOW CHART - MATINT

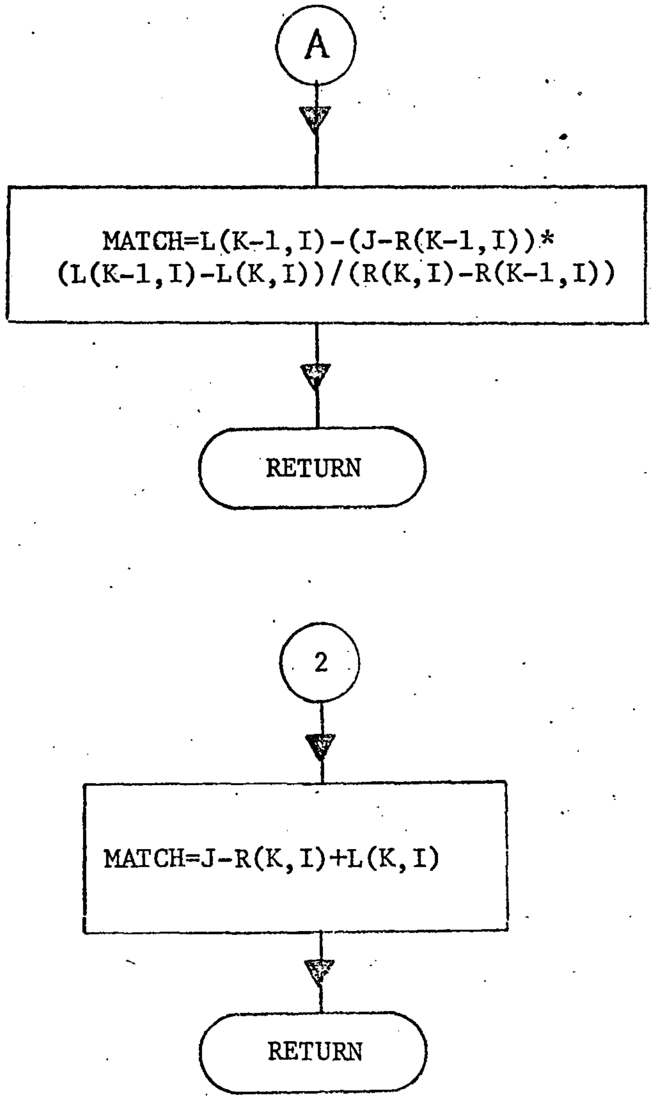


Fig. 10.5.3.1.2 DETAILED FLOW CHART - MATINT (Concluded)

10.5.3.2 Description of subroutine RDMKS (remove drummarks)

IDENTIFICATION

Name	- RDMKS
Author/Date	- R.L. Wendt/June 1970
Organization	- ASR
Machine Identification	- UNIVAC 1108
Source Language	- FORTRAN V

PURPOSE/METHOD

This subroutine removes the drummarks from the data by maximizing the drummark region data and the nearest adjacent data. This process doesn't eliminate craters when craters and drummarks occur together, and is far superior to linear interpolation both in execution time and in resulting correction.

USAGE

Calling sequence

CALL RDMKS (KBUF, I)

The calling arguments are described in Table 10.5.2.

The subroutine uses no subroutines of its own.

DETAILED FLOW CHART

See Fig. 10.5.3.2.1.

C.3

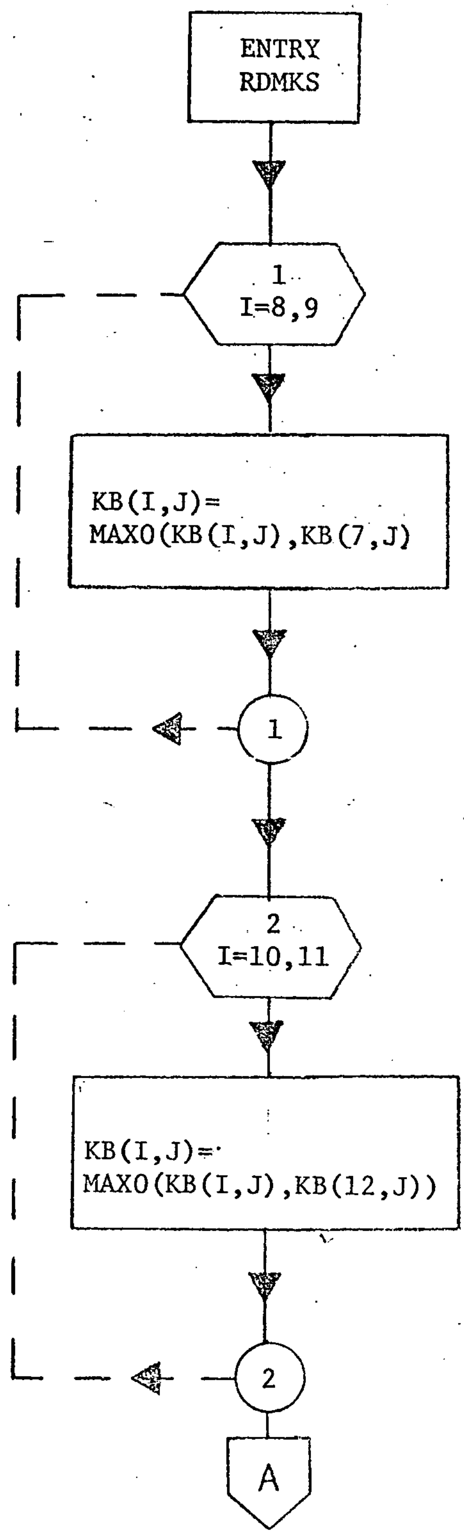


Fig. 10.5.3.2.1 DETAILED FLOW CHART - RDMKS

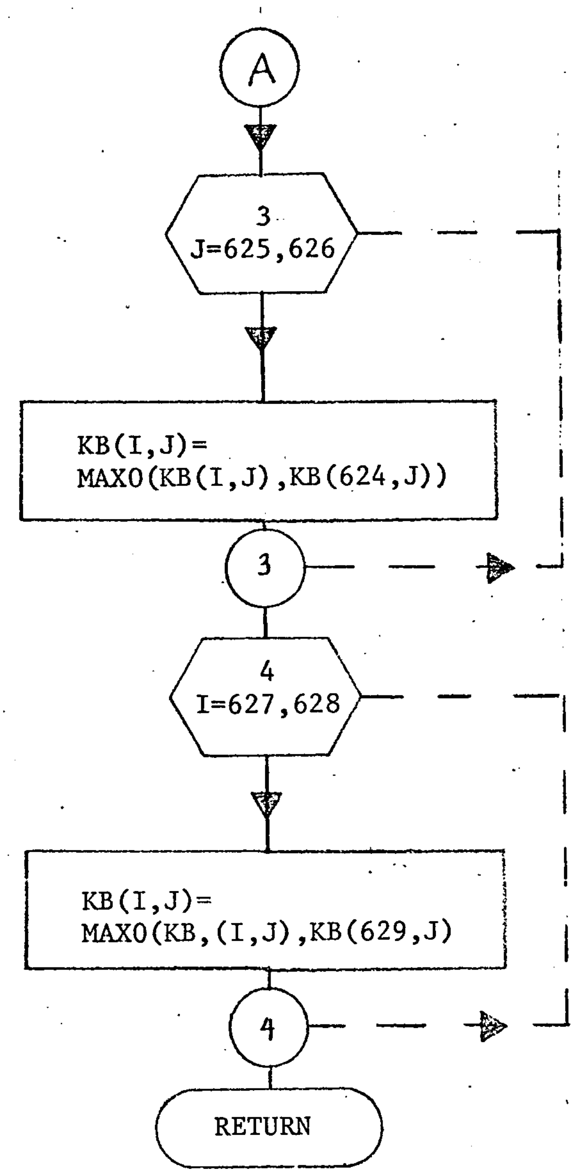


Fig. 10.5.3.2.1 DETAILED FLOW CHART - RDMKS (Concluded)

10.5.3.3. Description of subroutine STORE (store corrected data into RMTX)

IDENTIFICATION

Name	-	STORE
Author/Date	-	R.L. Wendt/June 1970
Organization	-	ASR
Machine Identification	-	UNIVAC 1108
Source Language	-	FORTRAN V

PURPOSE/METHOD

This subroutine stores data and applies smoothed correction factors from KBUF into RMTX based on the framelet number I and the row of the framelet I treated modulo NLINE which is KTR or LTR.

USAGE

Calling sequence (2)

STORE (KBUF, RMTX, I, LTR, NFMLT, NLINE, SCF)

or

STORE (KBUF, RMTX, I, KTR, NFMLT, NLINE, SCF)

The program variables are described in Table 10.5.2.

DETAILED FLOW CHART

See Fig. 10.5.3.3.1.

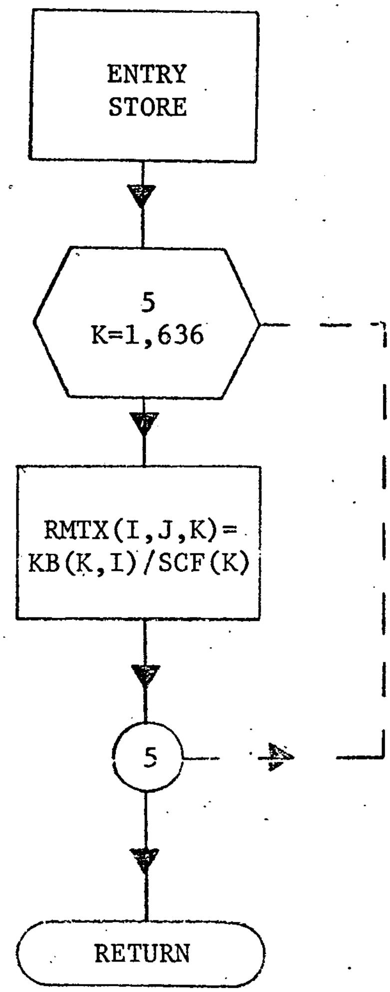


Fig. 10.5.3.3.1 DETAILED FLOW CHART - STORE

10.5.3.4 Description of subroutine SIXPAK

IDENTIFICATION

Name	--SIXPAK
Author/Date	--R.L. Wendt/June 1970
Organization	--ASR
Machine Identification	--UNIVAC 1108
Source Language	--FORTRAN V

PURPOSE/METHOD

This subroutine packs the data six characters per word from the joined master picture matrix MSTMTX into OBUF depending on the value of IFB which is used in double buffering the output.

USAGE

Calling sequence

CALL SIXPAK (MSTMTX, OBUF, IFB, NWOUT)

The calling arguments are described in Table 10.5.2.

SIXPAK uses subroutine PACK6 from the LO package.

DETAILED FLOW CHART

See Fig. 10.5.3.4.1.

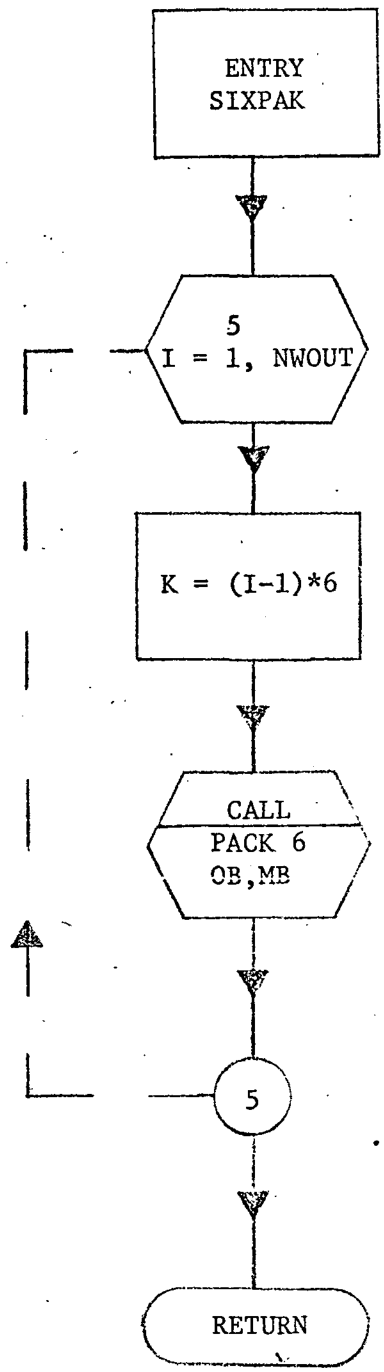


Fig. 10.5.3.4.1 DETAILED FLOW CHART - SIXPAK

10.5.3.5 Description of subroutine PSE

IDENTIFICATION

Name	-	PSE
Author/Date	-	R.L. Wendt/June 1970
Organization	-	ASR
Machine Identification	-	UNIVAC 1108
Source Language	-	FORTRAN V

PURPOSE

This subroutine allows the user a look at the joined picture especially in the region where the framelets were joined by the match points. The program was designed as a temporary means of determining whether the join algorithm is performing correctly.

GENERAL DESCRIPTION

Under the assumption that three framelets have been joined and that the joined edge of framelets 1 and 2 occurs between columns 570 and 690 and that the joined edge of framelets 2 and 3 occurs between columns 1188 and 1308, then words between 95 and 115 and between 198 and 218 of the output tape are reduced in tonal resolution via the transformation PRETTY and printed. Words between 198 and 218 are stored in the drum and after the first joined edge has been printed, the data is read from the drum and the second joined edge is printed. As was mentioned earlier, this routine provides a quick look at the final product. The routine can be removed easily, if necessary. The routine can be generalized in an obvious manner to provide for more framelets or to provide for a greater portion of the output picture to be printed.

USAGE

Calling sequence

PSE

Subroutines used

PMAP

PRETTY

PREG

DESCRIPTION OF VARIABLES

NL - Row index on output joined picture

IBUF - Input buffer used in reading output tape.

JBUF - Joined edges of picture for framelets 1 and 2.

KBUF - Joined edges of picture for framelets 2 and 3.

DETAILED FLOW CHART

See Fig. 10.5.3.5.1

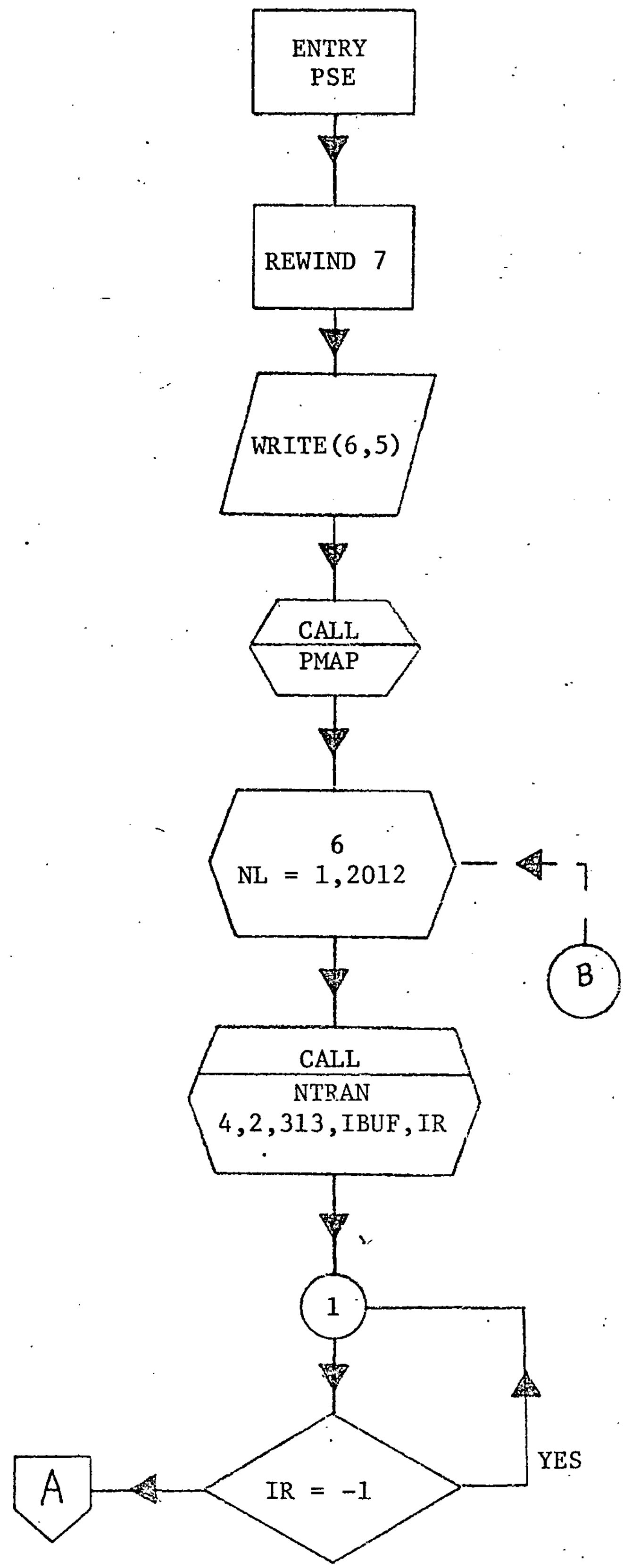


FIG. 10.5.3.5.1 DETAILED FLOW CHART - PSE

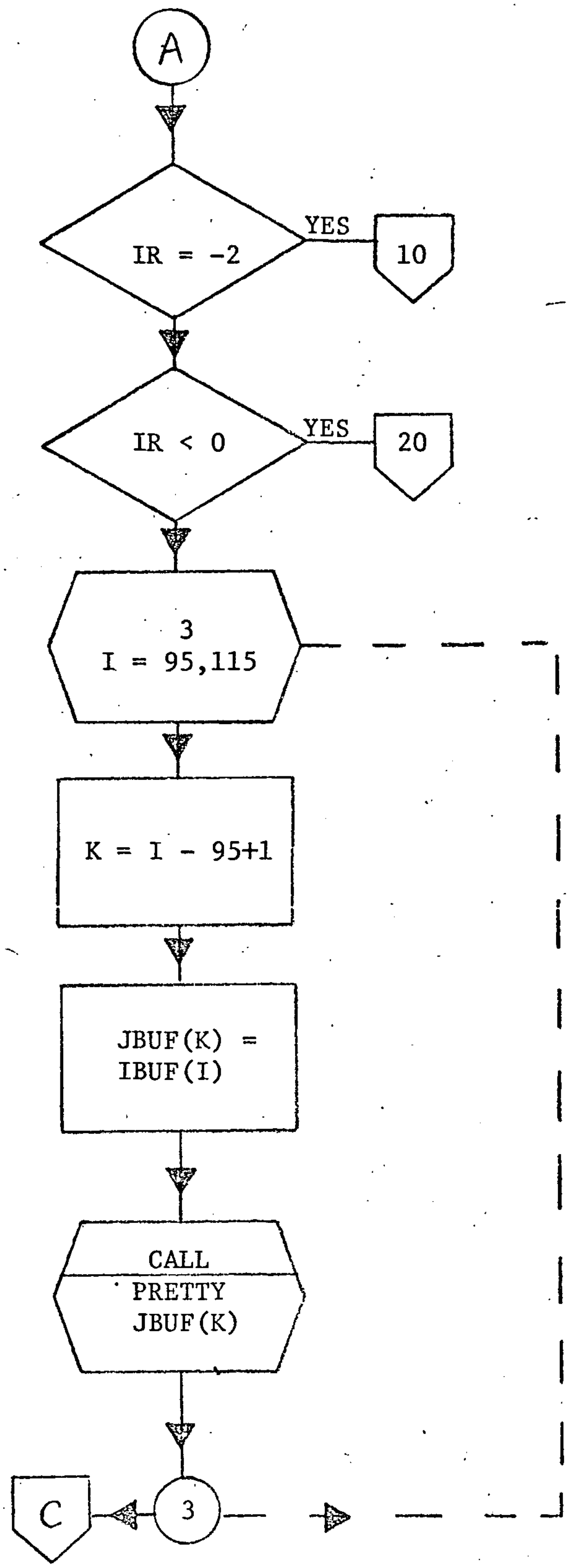


FIG. 10.5.3.5.1 DETAILED FLOW CHART - PSE (Continued)

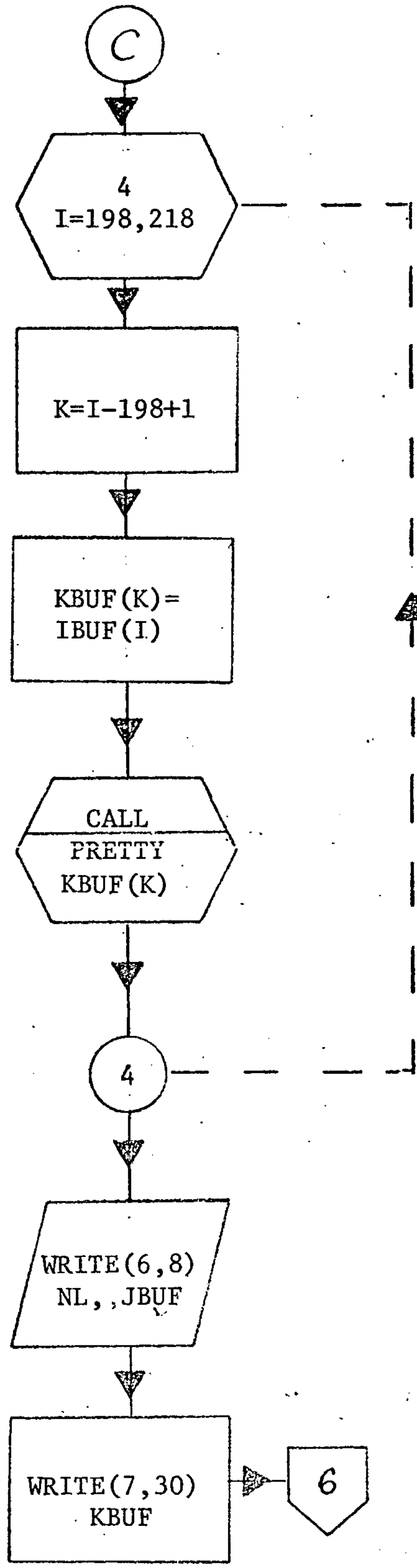


FIG. 10.5.3.5.1 DETAILED FLOW CHART - PSE (Continued)

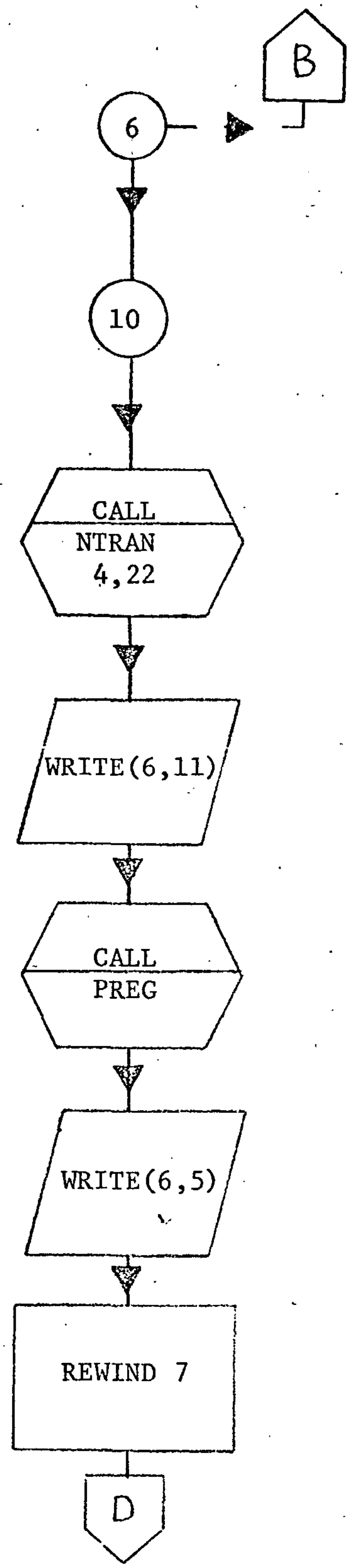


FIG. 10.5.3.5.1 DETAILED FLOW CHART - PSE (Continued)

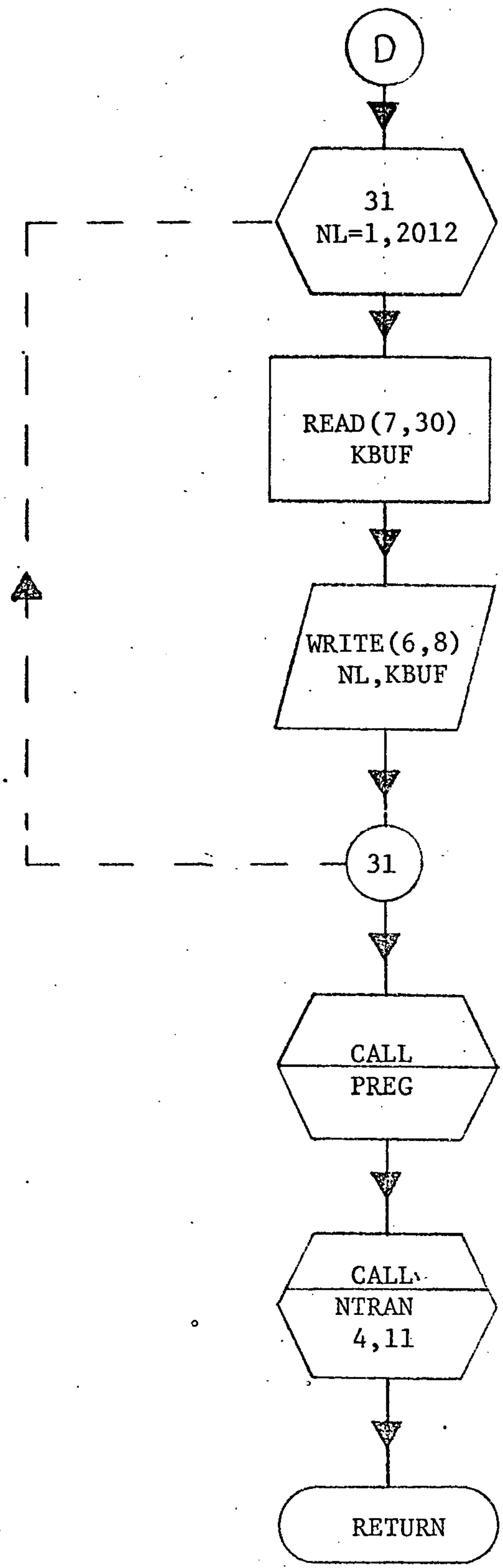


FIG. 10.5.3.5.1 DETAILED FLOW CHART - PSE (Continued)

BLANK PAGE

PRECEDING PAGE BLANK NOT FILMED

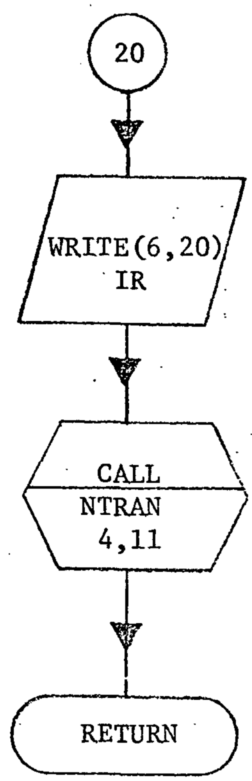


FIG. 10.5.3.5.1 DETAILED FLOW CHART - PSE (Concluded)

10.5.3.6 Description of subroutine UNPAK

IDENTIFICATION

Name	-	UNPAK
Author/Date	-	R.L. Wendt/June 1970
Organization	-	ASR
Machine Identification	-	UNIVAC 1108
Source Language	-	FORTRAN V

PURPOSE/METHOD

This subroutine unpacks NWPR words from IBUF dimensions (2*NWPR, NFMLT) into KBUF dimensions (6*NWPR, NFMLT). IBUF is a double buffer array while KBUF is single buffer array. IBUF is used on input (double buffered) and KBUF is a working array. The argument I determines which framelet is being considered and IFLG determines which buffer of IBUF is to be unpacked into KBUF.

USAGE

Calling sequence

CALL UNPAK (IBUF, KBUF, I, IFLG)

Subroutine used: UNPACK

The description of the calling variables is found in Table 10.5.2.

DETAILED FLOW CHART

See Fig. 10.5.3.6.1.

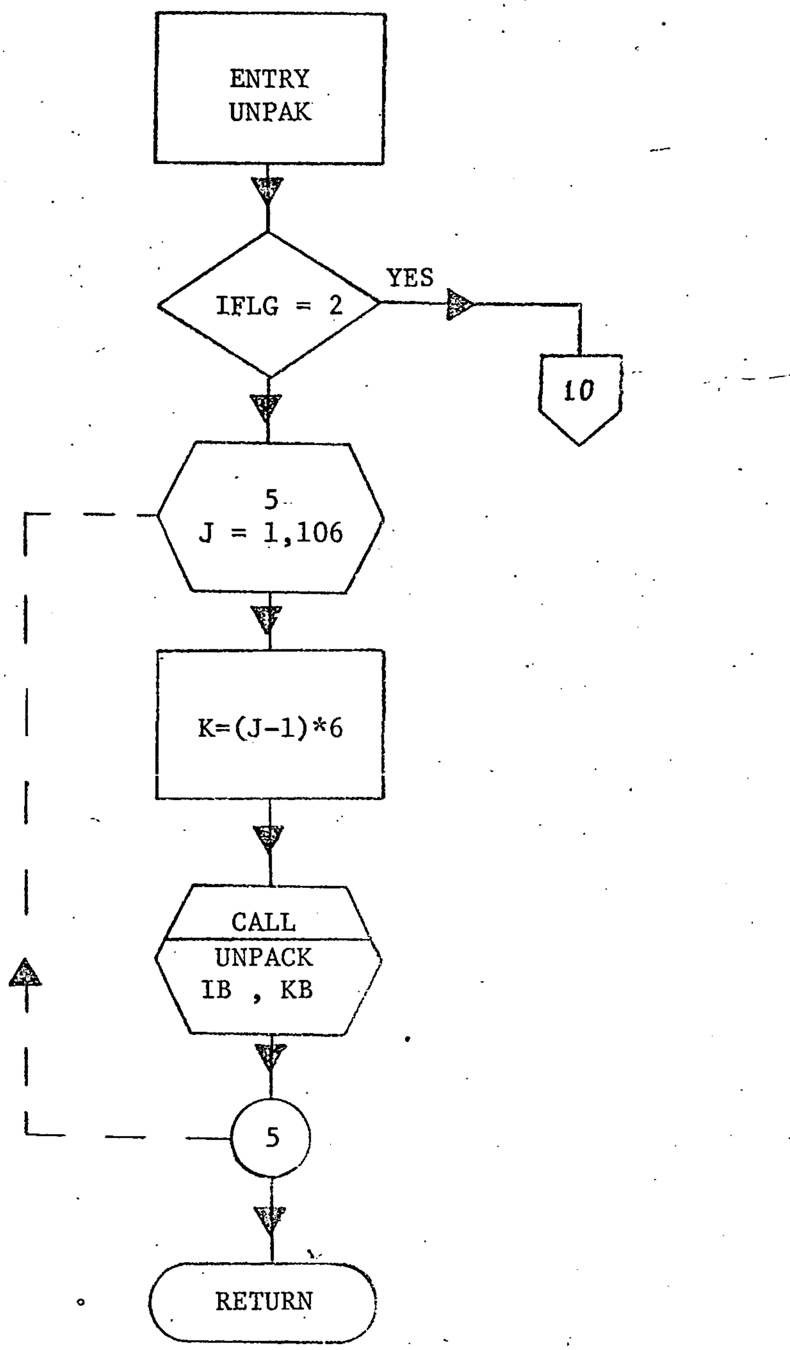


FIG. 10.5.3.6.1 DETAILED FLOW CHART - UNPAK

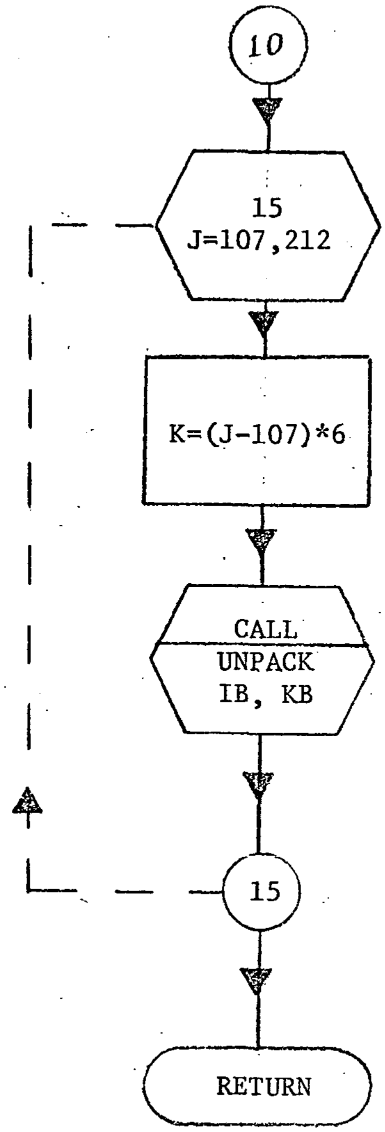


FIG. 10.5.3.6.1 DETAILED FLOW CHART - UNPAK
(Concluded)

11. Overall Operational Procedures

The programs reported on here do not, of course, run themselves. At the minimum, an experienced programmer must supervise the preparation and running of each of the programs reported on here. We have prepared a flow chart (Fig. 11.1) of steps to be performed. (One comment: program LRE and program COPYT (documented in the report on Project B, part 2 (filters)) should be combined; this assumption has been made in Part II of this report. The combination LRE with COPYT will run at the same speed (namely tape read speed) as either separately.) At the minimum, each box represents a turn-around time of one day. Two weeks would be a better estimate for the average time required to join a 9-chit area including turn-around time and apparently unavoidable delays in locating tapes and other information. We must emphasise the importance of an experienced programmer familiar with the generally confused situation LO data seems to be in.

JPL/IPL needs the following information (for example)

Reel #

Play: 7 track, odd parity, 800 bpi, one line/record, one file/picture, stretch to give film T linear with DN, 0 = .135 MCS, 77_8 = .61 MCS.
The geometric spacing: 36.05 μ line spacing, 40.5 μ pixel spacing. Playback normal, gives positive exposure on VFC film.
Give the dimension of each file (e.g. 1878 x 2012).

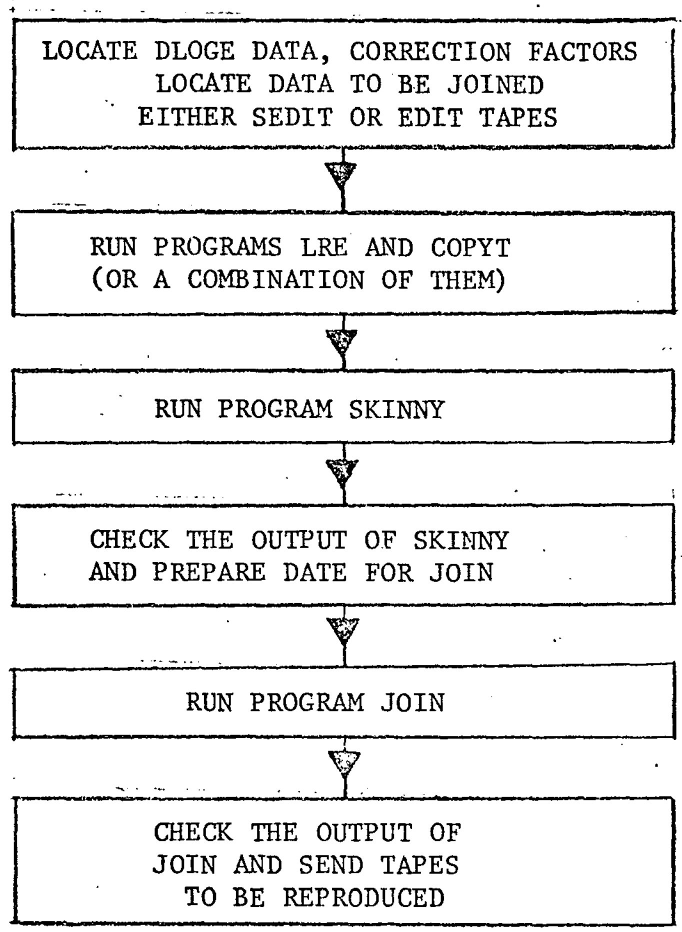


Figure 11.1 Operational Flow Chart, Joining Three Framelets

12. Illustrative Example.

The steps outlined in § 11 were carried out for the 2068-line (approximately 9-chit) area from LO III, FR 154 H, FRMLTS 028, 027, 025 (SURVEYOR-3 area). At the time, the output picture (1878 x 2012) could not be displayed at one pass owing to limitations in the display mechanism. A simple ad hoc program was written to break the joined picture, producing tape # 35419 consisting of two 936 x 2012 pictures. Fig. 12.1 and 12.2 are prints of the pictures produced by JPL/IPL of tape # 35419. (The negatives of all pictures produced in this and other reports are included with the first copy of the report.) Later, after the filter problem had been solved, we prepared another picture of the same area, tape # 15985, displayed in Fig. 12.3, incorporating many of the improvements reported on in Project B. This picture, it should be noted, is not produced by the JOIN program, but instead represents a slight reduction in resolution from the pictures of Fig. 12.1 and 12.2. The process of breaking the picture is no longer necessary since the improved display devices at JPL can handle pictures of the larger 1878 x 2012 size. We expect that an 1878 x 17,000 picture can now be reproduced at JPL/IPL, thus making it possible to reproduce a three framelet wide join area.

In case the interested reader desires a comparison with GRE reproduction, the NASA-LRC number containing this area is III-154H_2. Our picture is actually a mirror image of the GRE picture. We have photographed and reproduced this area for the convenience of the reader in Fig. 12.4.

FIGURE 12.1

FIGURE 12.2

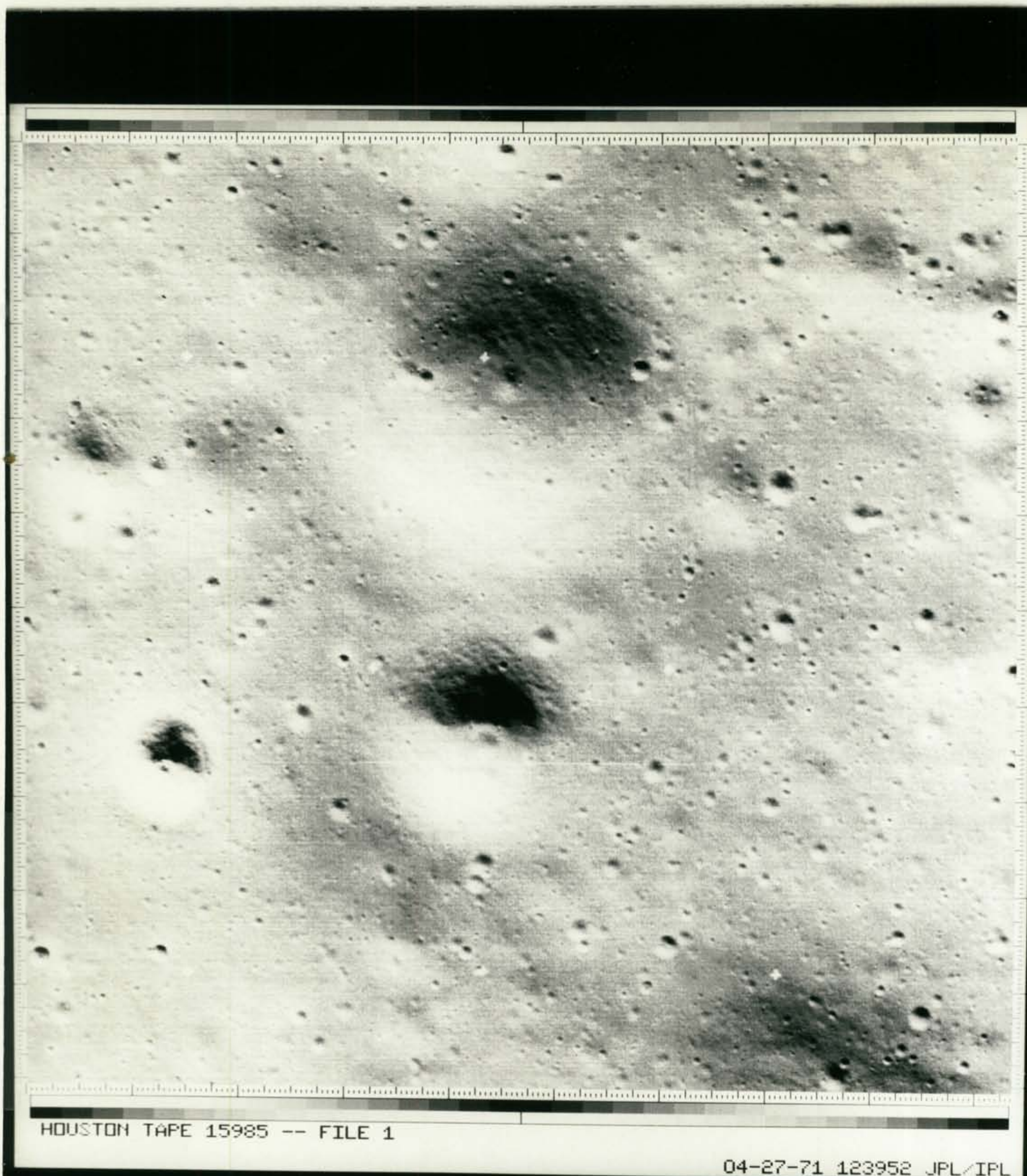


FIGURE 12.3

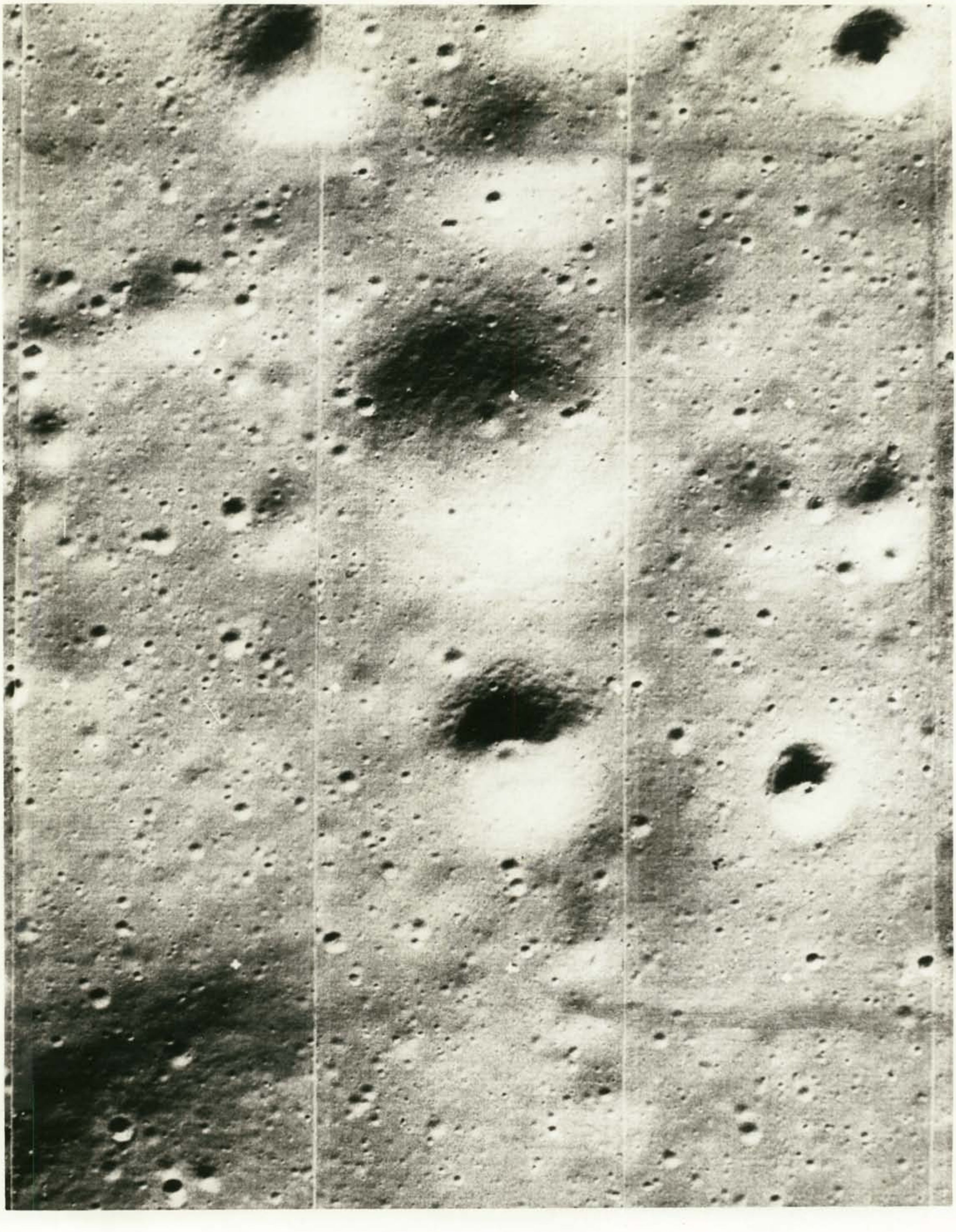


FIGURE 12.4

Part II TIME AND COST ESTIMATES FOR JOINING A SERIES OF LUNAR ORBITER FRAMELETS

1. General Discussion

In forming the following estimates of time and cost in joining a series of LO framelets we make the following assumptions:

1.1 The D log E data and correction factors have been obtained for this particular series of LO framelets in the format required by SKINNY and JOIN.

1.2 All framelets to be joined were digitized under approximately the same conditions of clipping and zero level adjustments.

1.3 Facilities for producing the digital picture made by JOIN are available.

1.4 All input tapes are written one line per record.

These assumptions are reasonable for a production-type job of joining LO framelets with the possible exception of 1.3. In addition, we assume the operator understands the operation and limitations of SKINNY; part of the cost is in fact budgeted to checking (by looking at LRE output) the match points furnished by SKINNY, and to key-punch type operation for making input data for JOIN.

In our problem (a test problem), we had to find new D log E data and correction factors (as discussed in Project BO, aso that assumption 1.1 was false in our case. However, this work, once completed, need not be repeated for larger quantities of LO data. Assumption 1.2 was valid for our data. Assumption 1.3 was not, and our joined pictures were necessarily broken in half before they could be displayed. Assumption 1.4 was also false; if it is not made, add the time taken to run a program which takes

2. Time Estimate Budget, 9-Chit Area

OPERATION	COMPUTER TIME, 1108 min.	sec.	MAN-HOURS
Prepare input tape numbers and locate all software for run of job.			1
Run program LRE	2	0	0.2
Run program SKINNY	2	30	0.1
Interpret results of SKINNY, prepare data for JOIN			3
Run program JOIN	10	50	0.1
Check results of JOIN, prepare description of tape and letter for sending to JPL for display			1.6
Total estimate, 9 chit area, no filterning.	15 min	20 sec	5 hours

TABLE 2.1

3. Time Estimate Budget, Three Framelet Area

OPERATION	COMPUTER TIME, 1108 min.	sec.	MAN-HOURS
Prepare input tape numbers and locate all software for run			1
Run program LRE	6	40	0.2
Run program SKINNY	9	50	0.1
Interpret results of SKINNY, prepare data for JOIN			6
Run program JOIN	70	30	0.1
Check results of JOIN, prepare description of tapes and letter for sending to JPL for display			2.6
Total estimate, three framelet area, no filterning.	1 hr.	17 min.	10 hours

TABLE 3.1 TIME ESTIMATE, 3 FRAMELET (17,000 lines) AREA

the same time as LRE to rewrite the SEDIT tapes one line per record.

4. Conclusions, Time and Cost Estimates

It seems reasonable to postulate a formula of the form $C = an + b$ for the cost of doing n lines, three framelets wide. In the nine chit area, n = 2112. In the example of three framelets, 17,000 lines, n = 17,000. Using the estimates computer time at $130/hour and man time at $15/hour (including overhead), we obtain the following cost estimates:

2012 lines	$108.22
17,000 lines	$316.83

In preparing longer runs of more data, it would be worth the cost to assign a specialist to run the jon (thereby lowering the contribution in man-hour cost). After training, we estimate that a large number of framelets could be joined for a smaller cost as follows:

2012 lines	$58.00
17,000 lines	$216.00

We feel these numbers represent an irreducible minimum in cost of joining three LO framelets, leading to a cost vs number of lines graph as indicated in Figure 4.1. We wish to emphise, however, that the lower estimate can be attained only on a production basis. Our general estimate gives the upper estimate as $75 plus $14 per thousand lines and the lower estimate $35 plus $10.50 per thousand lines for joining three framelets.

Fig. 4.1 Dollar cost vs. number of lines, three framelets.

Part III Generalizations

The main problem in Project A turned out to be the lack of undistorted overlap between adjacent framelets. Therefore, almost our entire effort was spent trying to find match points for adjacent framelets, via program SKINNY, and to join three framelet wide areas using JOIN. Neither of these programs were designed to make use of the preprinted pattern of reseau marks. This is fortunate, for the nonlinearity in the verticle direction turned out to be of such a high frequency that the reseau pattern has too wide a spacing to be of any use in the JOIN program; neither can it be relied on to locate match points. However, once the high frequency deviation has been removed in the relative sense (i.e., between adjacent framelets), the possibility exists that the reseau pattern can be used to correct at least the main remaining sources of distortion. For this reason, an algorithm was developed which locates the exact center of a reseau mark in a given area (or, alternatly, the "exact center" is defined to be the point located by this (template matching) algorithm). Section 1 of Part III is the documentation of test program RESEAU. Section 2 outlines how the reseau marks can be used to build a picture of large areas (say 3 framelets of 15,000 lines) on a master mesh from the output of JOIN.

COMPUTER PROGRAM DOCUMENTATION

Locate Reseau Marks

PROGRAM RESEAU

Project A

by

Jack Bryant

and

R. L. Wendt

Prepared by

Applied Scientific Research, Inc.

Houston, Texas

Under Contract NAS 9-10577

for

MAPPING SCIENCES DIVISION

National Aeronautics and Space Administration

Manned Spacecraft Center

Houston, Texas

September, 1971

1. DOCUMENTATION OF RESEAU

1.1 INTRODUCTION

The algorithm reported on here is a combination of an elementary template match and clipping in which the template was determined by observation of actual LO data. We averaged (on the HP 9100B) actual reseau marks found in the output of a generalization of program LRE when applied to framelet 027. It turns out that the Reseau pattern is present in both the edges of framelet 027. We then, to save computation time, converted the algorithm into a three value one (involving only 1, 0 and -1) and wrote the programs to perform the convolution by DO-loop sums. This resulted in a very fast algorithm for a template match of thin size. Fig. 1.1.1 gives values of the template used. Experimentation indicated a large improvement in performance could be obtained by clipping the data away from the test site by taking the minimum of the data point and 10. This gives a detection threshold of approximately 300; that is, if the value given by the algorithm is less than 300, the hypothesis that there is a reseau mark between ROW1, ROW2-COL1, COL2 should be rejected. Without clipping, the template match failed when tested in an area with a small but distinct crater of about 10 meters diameter, owing to the fact that the black part of the crater made a greater contribution to the sum than it should have.

1.2 PROGRAM DESCRIPTION

1.2.1 GENERAL DESCRIPTION OF PROGRAM

The program skips over the header array on the input tape. The number of lines (at one line per record) of data to be skipped over to get to the approximate area of the reseau mark is computed and the input

COL / ROW	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-8	1	1	1	1	0	-1	-1	-1	0	1	1	1	1
-7	1	1	1	1	0	-1	-1	-1	0	1	1	1	1
-6	1	1	1	1	0	-1	-1	-1	0	1	1	1	1
-5	1	1	1	1	0	-1	-1	-1	0	1	1	1	1
-4	1	1	1	1	0	-1	-1	-1	0	1	1	1	1
-3	0	0	0	0	0	-1	-1	-1	0	0	0	0	0
-2	0	0	0	0	0	-1	-1	-1	0	0	0	0	0
-1	-1	-1	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	-1
0	-1	-1	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	-1
1	-1	-1	-1	-1	-1	-2	-2	-2	-1	-1	-1	-1	-1
2	0	0	0	0	0	-1	-1	-1	0	0	0	0	0
3	0	0	0	0	0	-1	-1	-1	0	0	0	0	0
4	1	1	1	1	0	-1	-1	-1	0	1	1	1	1
5	1	1	1	1	0	-1	-1	-1	0	1	1	1	1
6	1	1	1	1	0	-1	-1	-1	0	1	1	1	1
7	1	1	1	1	0	-1	-1	-1	0	1	1	1	1
8	1	1	1	1	0	-1	-1	-1	0	1	1	1	1

NOTE: All values 1 were applied with clipping by 10 (giving a nonlinear template match algorithm). Values -1 and -2 were applied without clipping.

Figure 1.1.1. Template Values, RESEAU

tape is advanced. The number of lines of data to be read is determined, the tape is read, the data unpacked and stored in A. The convolution (with clipping) A*M (where M is the template match) is performed over the area described by ROW1, ROW2, COL1, COL2 as M varies over A. (The actual filter M is only implicitly defined.) The maximum of the convolutions is recorded as well as the row and column coordinates. At the end of the process the coordinates of the reseau mark are printed and the time the algorithm used in finding the reseau mark is printed also, along with the value of the maximum.

1.2.2 GENERAL FLOW CHART

See Fig. 1.2.2.1

1.2.3 TECHNICAL DESCRIPTION

The algorithm maximizes the following function C(I, J), where I and J vary over ROW1, ROW2 and COL1, COL2

$$
\begin{aligned}
C(I, J) = & \sum_{i=-8}^{-4} \sum_{j=-6}^{-3} \mathrm{MAX0}(A(I + i, J = j), 10) \\
& + \sum_{i=-8}^{-4} \sum_{j=3}^{6} \mathrm{MAX0}(A(I + i, J + j), 10) \\
& + \sum_{i=4}^{8} \sum_{j=-6}^{-3} \mathrm{MAX0}(A(I + i, J + j), 10) \\
& + \sum_{i=4}^{8} \sum_{j=3}^{6} \mathrm{MAX0}(A(I + i, J + j), 10) \\
& - \sum_{i=-1}^{1} \sum_{j=-6}^{6} A(I + i, J + j) \\
& - \sum_{i=-8}^{8} \sum_{j=-1}^{1} A(I + i, J + j).
\end{aligned}
$$

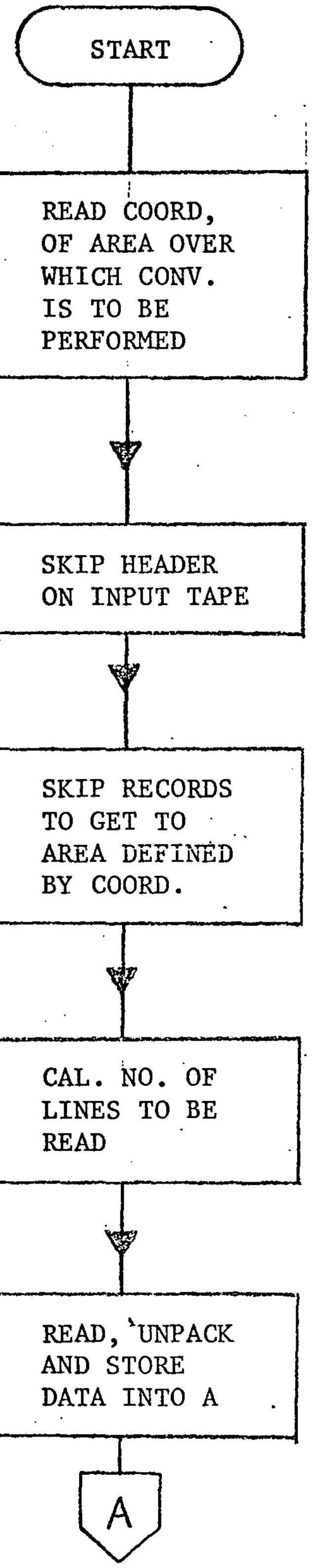


FIGURE 1.2.2.1 GENERAL FLOW CHART, RESEAU

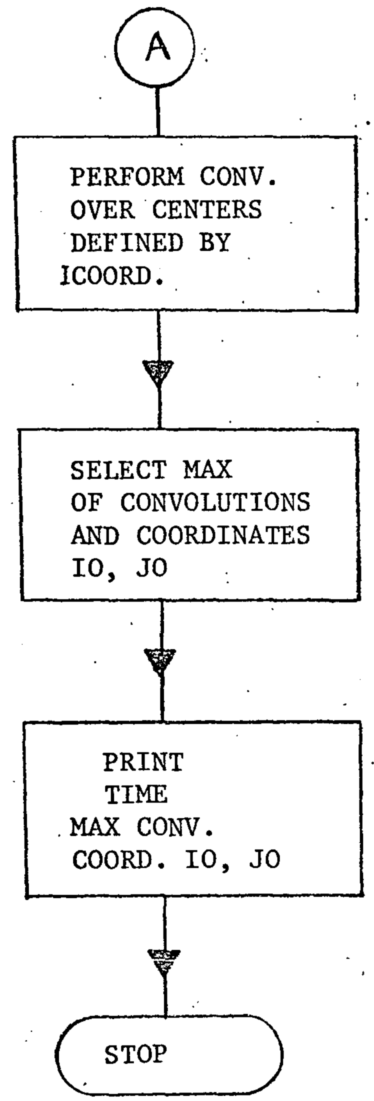


FIGURE 1.2.2.1 GENERAL FLOW CHART, RESEAU

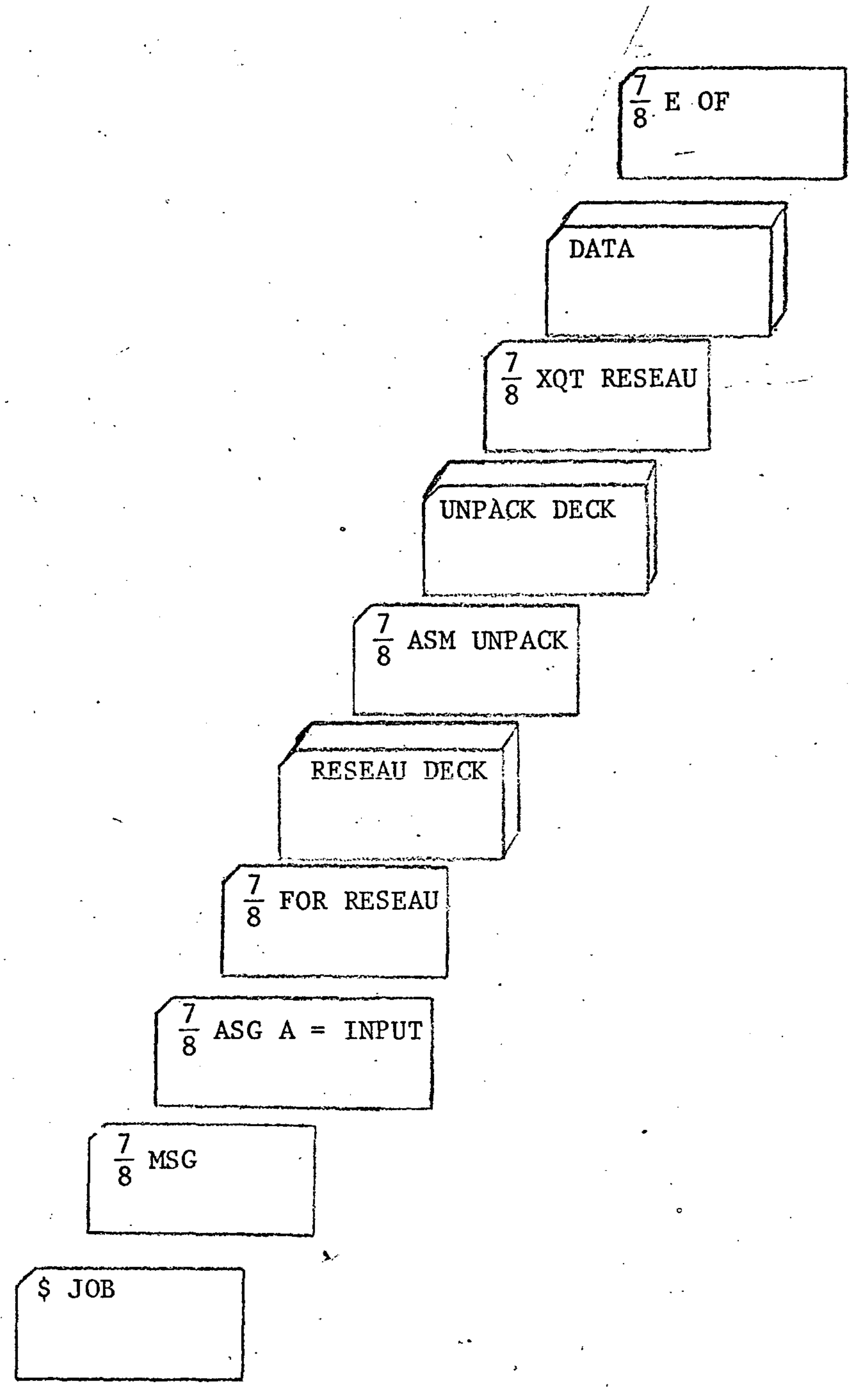


FIGURE 1.3.2.1 DECK SETUP - RESEAU (UNIVAC 1108)

1.3 USAGE

1.3.1 Input Description

1.3.1.1 Card input: When used with only one input reel, the program requires ROW 1, ROW 2, COL 1, COL 2 to be read in FORMAT (4I5). Various versions of the program exist which also use multiple reel assignments, simply putting the entire program in a big loop with the number of reels and the number of files per reel needed read in as data cards. The program documented here is the basic one.

1.3.1.2. Tape input: data tapes as made for JOIN by MRGNDE.

1.3.2 PROGRAM RUN PREPARATION

See Figure 1.3.2.1 for card deck setup. The only subroutine required in UNPACK.

1.3.3. OUTPUT DESCRIPTION

The time (in msec.), max, and row and col index of the max will be printed out. No other output is furnished.

1.4 EXECUTION CHARACTERISTICS

1.4.1 Restrictors: The algorithm will not locate a reseau mark too near the edge; in particular, tested center values must have row index $\geq$ 9 (with similar restrictions on the upper limit) and col index between 7 and 630.

1.4.2. The running time depends on the size to be tested. In most cases, the actual execution will be mostly reading tape, so that about two minutes 1108 (or 1106) time will be required.

1.4.3 Accuracy/Validity: tests (using program SNIF) perfomed confirm that the algorithm (with clipping) can be used as a detector to check if any reseau marks are near, as well as to locate the center of one present. Also, all reseau marks on 2068 were located (taking 2 min 59 sec with three reels of tape being partially read) and printed out in a generalized version of this program, this work being done before the clipping was introduced. It is not hard to see how all reseau marks for any framelet could be located and printed at tape read time using data in the header, although this work has not been done. So far as we are able to determine, the algorithm accurately locates all reseau marks.

1.5 SYMBOL DEFINITION TABLE

See Table 1.5.1.

1.6 DETAILED FLOW CHART

See Figure 1.6.1.

IBUF	An array which is used in reading in data.
B	An array used in unpacking data.
A	A double subscripted array over which the match filter varies in attempting to maximize a convalution and thus finding a RESEAU mark.
ROW 1, ROW 2	Approximate coordinates of where a Reseau mark may be.
COL 1, COL 2	Reseau mark may be.
WREC	The number of lines (records) of the input tape which must be skipped to get to the data of the area described by ROW 1, ROW 2, COL 1, COL 2.
INDX	The number of lines (records) which must be read ie described by ROW 1, ROW 2, COL 1, COL 2.
IL, IU	The limits of the array A which accepts unpacked data with respect to the columns
MAX	The value of the convalution at coordinates IO, JO of the Reseau mark.
I1, I2, J1, J2	The limits on sums used in performing the convalution.
K, L	Indices for A so that the convalution being performed over limits I1, I2, J1, J2 remains within A.
ISUM	The value of the convalution at each point of interest.
KTR	A countve which is used in defining I1, I2, J1, J2 as each of the sums of the convalution are obtained.
IO, JO	The row and column of the Reseau mark i.e. where the convalution is maximal. IO, JO defines the center of the Reseau mark.
ITIME	The time in milliseconds for the algorithm to find the Reseau mark.

TABLE 1.5.1. DESCRIPTION OF PROGRAM VARIABLES - RESEAU

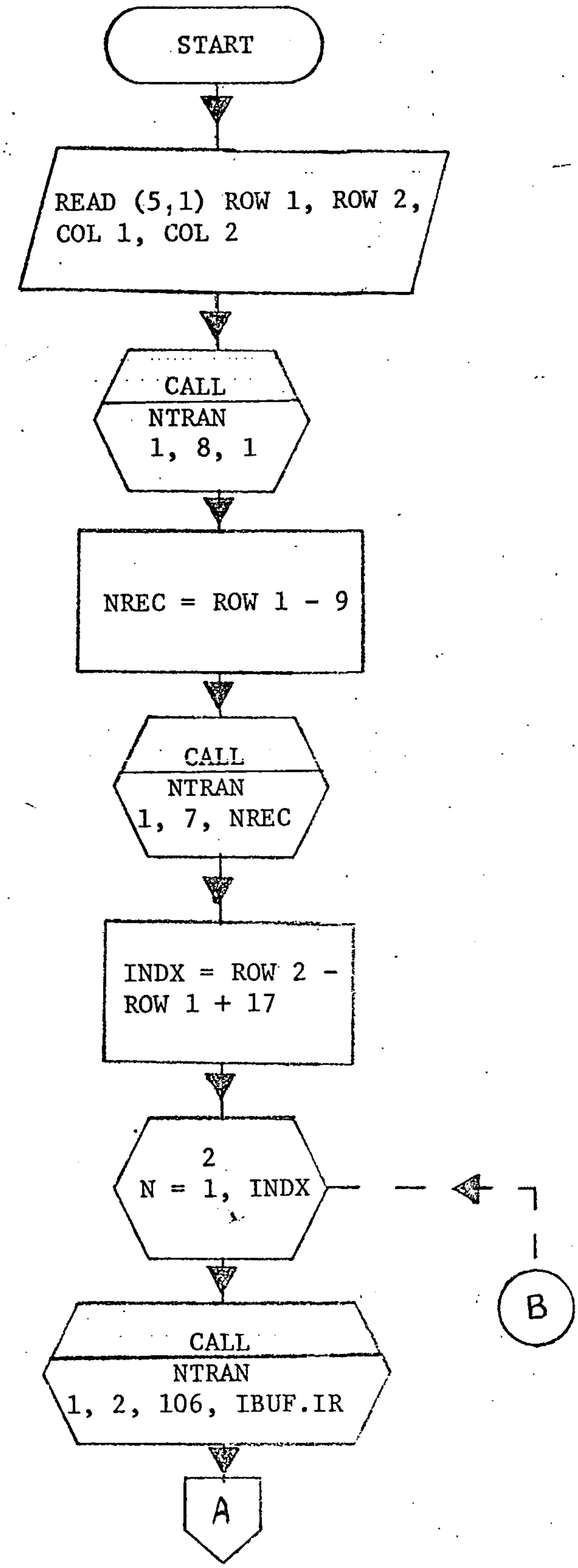


FIGURE 1.6.1 DETAILED FLOW CHART - RESEAU

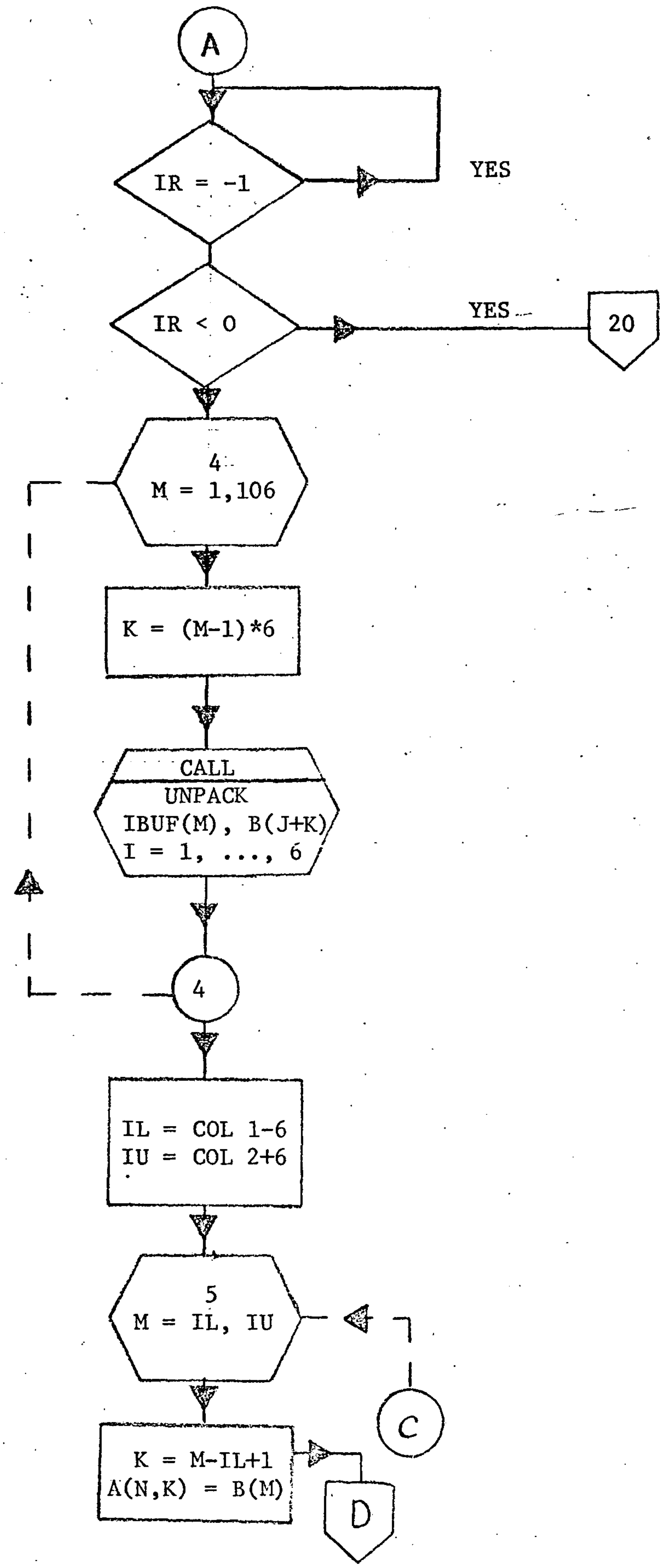


FIGURE 1.6.1 DETAILED FLOW CHART - RESEAU

(CONTINUED)

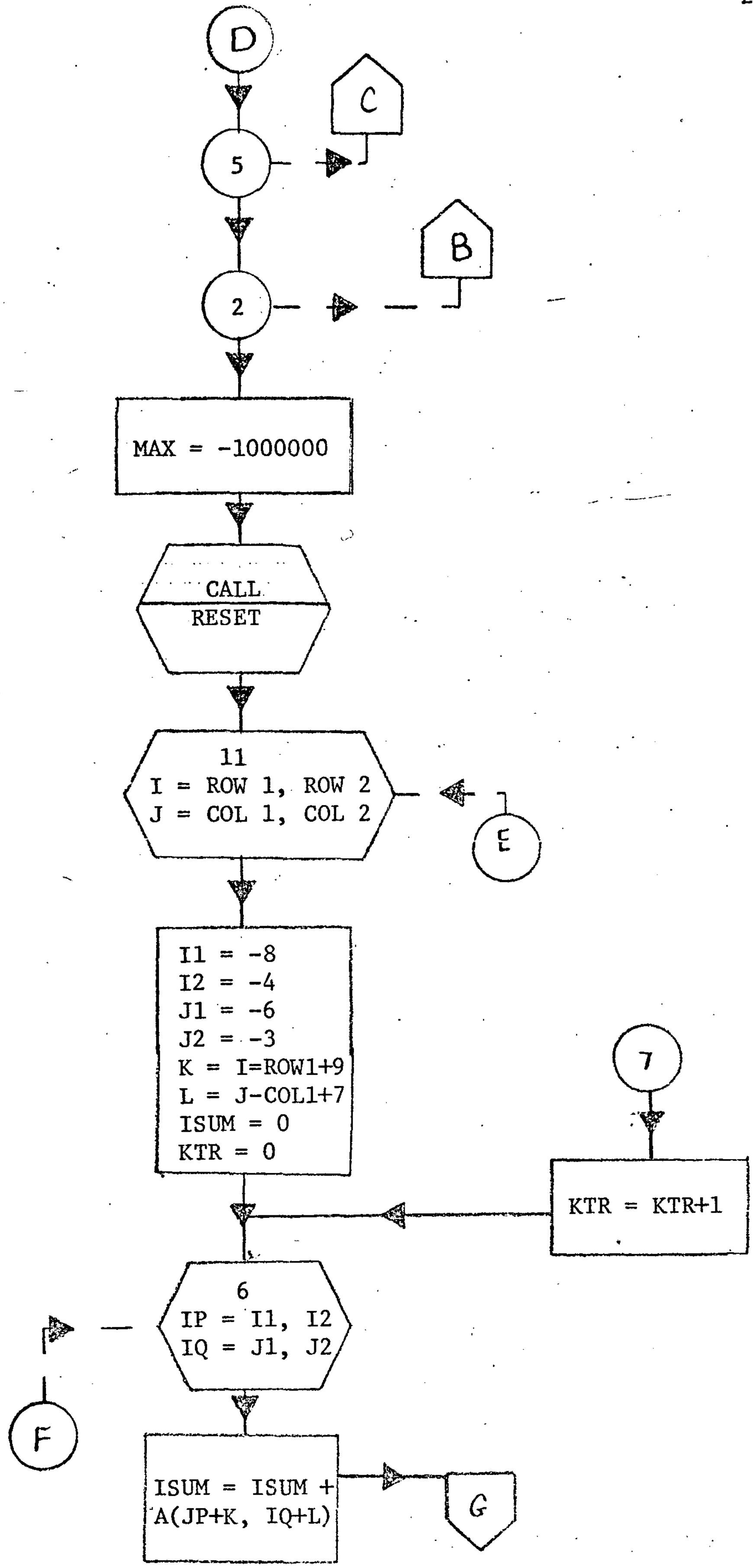


FIGURE 1.6.1 DETAILED FLOW CHART - RESEAU (CONT')

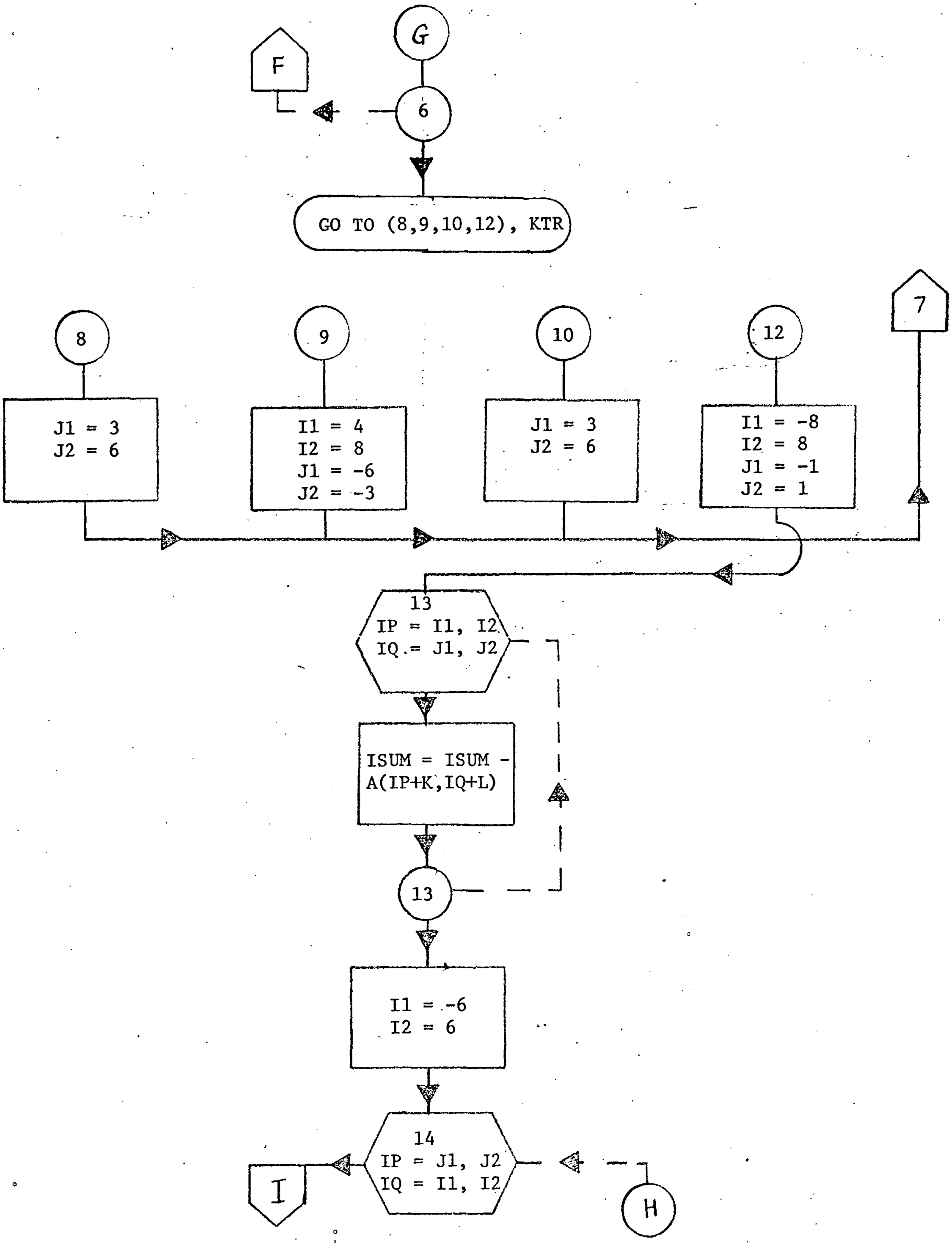


FIGURE 1.6.1 DETAILED FLOW CHART - RESEAU (CONT')

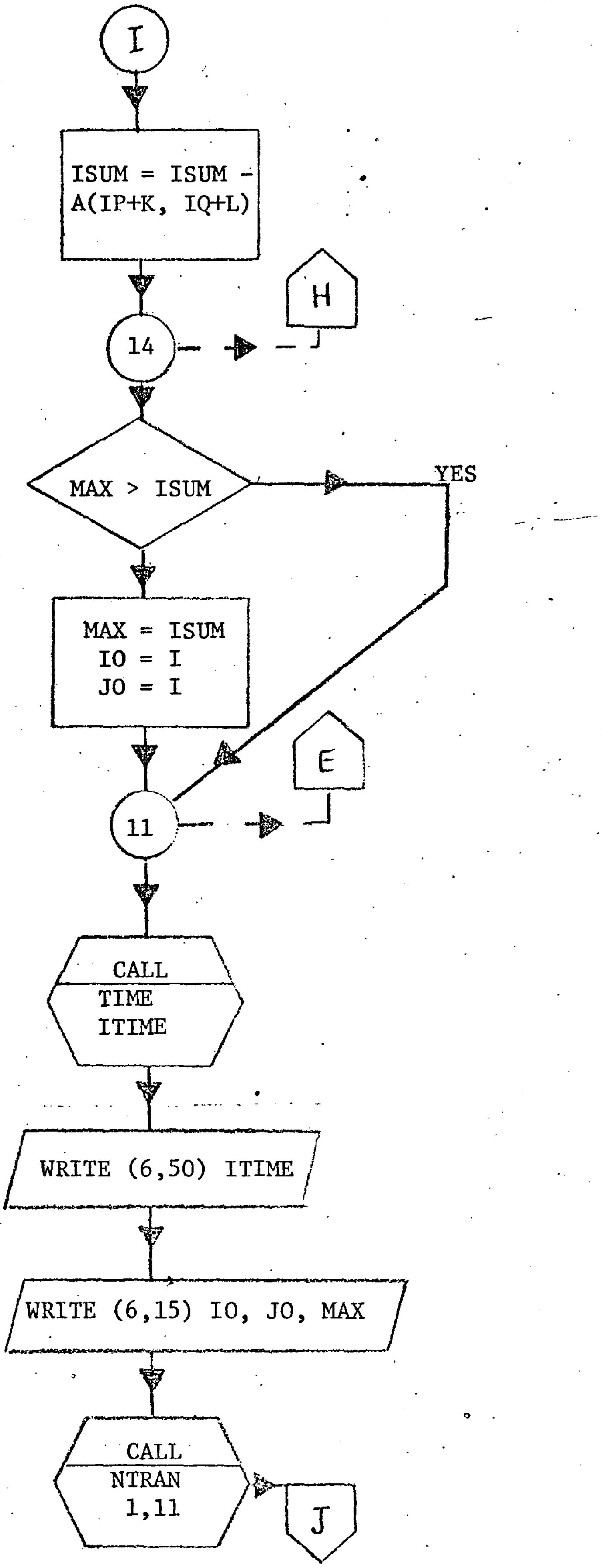


FIGURE 1.6.1 DETAILED FLOW CHART - RESEAU (CONT')

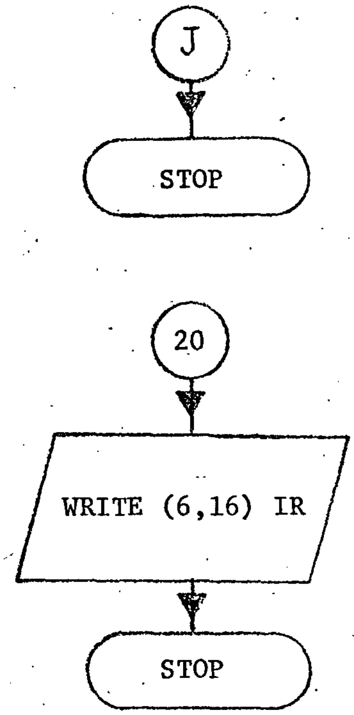


FIGURE 1.6.1 DETAILED FLOW CHART - RESEAU

2. <u>Use of preprinted reseau pattern in large-scale corrections</u>

The information immediately following is taken in part from the callibration data available to us on LO III.

The spacecraft film used in Lunar Orbiter III was preprinted with a pattern of reseau marks. These marks were pre-exposed on the film at the time the edge data printing was accomplished. The marks were formed by a double row of cross-shaped apertures in a bar extending across the width of the 70mm film. As the film was moved beneath at a constant velocity, a flashtube was fired at constant intervals to produce the pattern at equal spacing.

A specimen of the spacecraft film was processed, and the reseau pattern was measured in a comparator. In performing the measurements, the film was aligned with the edge data parallel to the x axis of the measuring stage. All reseau marks were measured for each of a large number of consecutive cycles. The y coordinates were averaged for a given mark as it appeared in all cycles. The x coordinates of all reseau marks within a single cycle were averaged to establish a point of reference for the flash exposure. The individual x coordinates were compared to the mean values for each cycle to establish a point of reference for the flash exposure. The individual x coordinates were compared to the mean values for each cycle to establish deviations from a straight-line pattern. These x increments as well as the measured y coordinates were combined to establish Table 2.1.

Reseau points are numbered increasing away from the edge data. The x axis is directed (positive) toward increasing edge data numbers. By comparing the mean values of the x coordinates of reseau marks in a given cycle with the mean values for succeeding cycles, it was determined that the pattern was prepeated every 2,269mm.

The reseau marks appear in the edge data of 027 (FR 154 H, LO III) and we (for another purpose) printed the edges (using ad hoc program PMEF). By observation of this data, we arrive at the information contained in Table 2.2. The point numbers correspond to the odd point numbers in Table 2.2. Using our best estimate of distance between rows of 0.00360487mm (which we round to 0.003605 for use in other programs), we are able to get the deviation from the ideal values of the row numbers of the reseau marks. (The estimate 0.00360487mm between rows minimizes the sum of the squares of the deviations.) We find an average absolute deviation of 0.009182mm, or about 2 or 3 rows, which does not contradict our previous findings, with a peak deviation of over 5 rows.

Using also the even-numbered reseau marks, it would be relatively easy to reconstruct a corrected version of three framelets (15,000 lines) joined, since our measurements disclosed the fact that no rotation was present in the data, and since the horizontal (x) direction location was exactly ($\pm$ 2 digital elements at most) as specified. We blame the x deviation on the jitter in the digitizer, and assert that nothing can be done about it because of its very high frequency. The basic program to perform this correction is already present in the JOIN program, which consists mathematically of merely linearly stretching or compressing the rows between control points (easily machine recognizable) as specified. In the JOIN program itself, we took a "standard" the left-most framelet

Point No.	x	y	x Std Devatn	y Std Devatn
	(MILLIMETERS ON FILM)			
1	0.000	0.000	0.003	0.002
2	1.404	2.355	0.002	0.002
3	0.009	4.658	0.002	0.003
4	1.404	6.997	0.001	0.003
5	0.011	9.297	0.003	0.003
6	1.411	11.656	0.002	0.002
7	0.016	13.959	0.002	0.002
8	1.414	16.363	0.002	0.003
9	0.021	18.667	0.002	0.002
10	1.418	21.044	0.002	0.002
11	0.025	23.343	0.002	0.002
12	1.422	25.702	0.002	0.003
13	0.029	28.005	0.002	0.002
14	1.429	30.279	0.002	0.003
15	0.033	32.581	0.002	0.002
16	1.429	34.926	0.002	0.002
17	0.036	37.227	0.001	0.003
18	1.437	39.607	0.002	0.002
19	0.040	41.909	0.003	0.002
20	1.442	44.262	0.003	0.002
21	0.048	46.567	0.002	0.003
22	1.444	49.030	0.002	0.003
23	0.050	51.331	0.001	0.003

TABLE 2.1 RESEAU PATTERN, LO III

POINT NO.	ROW NO.	ACTUAL Y VALUE	IDEAL Y VALUE	DEVATN.
1	14850	0.000	0.000	0.000
3	13555	4.668	4.658	0.010
5	12268	9.308	9.297	0.011
7	10979	13.954	13.959	0.005
9	9675	18.655	18.667	-0.012
11	8377	23.334	23.343	-0.009
13	7079	28.013	28.005	0.008
15	5810	32.588	32.581	0.007
17	5810	32.588	32.581	0.000
19	3225	41.907	41.909	-0.002
21	1937	46.550	46.567	-0.017
23	605	51.531	51.331	0.020

TABLE 2.2 ROW NUMBERS AND DEVIATION, 027

and joined the other two it it, making a picture with no relative deviations. A second pass, using a relatively simple _ad hoc_ program, could correct the joined picture of 15,000 lines so that the reseau marks were as specified in framelet 027. The main reason this program was not written is that there was no display device available to display the joined picture even if we could correct the picture to adjust the reseau pattern to be as specified.

Even if this work were carried out, it is probably that the basic nonlinearity in the y-direction is too high frequency to be completely removable. This is particularily suggested by looking at the last two points in Table 2.2 (with deviation of -4 and +5 rows from standard).

3. SKINNY on framelets

In order to check the operation of the match locating algorithm SKINNY, we devised a very severe test in which the algorithm was applied to 17,000 line framelets. Recall that SKINNY on 2068 lines read all the data from the tapes into a buffer C dimensional 2068 x 8. If we were to try to use the same approach on 17,000 line framelets, we would exceed the core limits of the 1108 (by a large amount); in addition, we would waste some time which might possibly be saved by double buffering on input. Since we were more interested in testing SKINNY than in matching on a production basis, we devised the following test: Run SKINNY on 1500 line batches (i.e., change the number 2068 to 1500) and, after initial match is found, advance to the center of the next 1500 line batch, keeping the same column match as before and using LKFR to find new candidate areas. The obvious flaw here is that interesting areas broken by the 1500 line batch limits are not available for matching at all, because LKFR cannot "see" them. The remedy is to treat the 1500 line array as a circular buffer and always "rotate" the buffer by the amount needed to move the most recently found match-point just "off" the buffer, using LKFR over the next 1500 line buffer and SKINNY to move the match algorithm up and down as usual. (It is uncanny how many interesting match points (which would have been found by LKFR with spectacular matches) lie exactly at 1500 line intervals.) Nevertheless, SKINNY worked on full framelets, finding the same column match consistently as on 2068, and finding in all 5, 5, 4, 4, 4, 5, 4, 5, 6, 5 and 1 match points (over the 1500 line batches) for a total of 48 match points. The first match point actually found (with the most interesting area) was suprisingly not on the edge at all, but in the center of the first 1500 line batch. (We expected just the opposite, that LKFR would like the high contrast edge; however, apparently it likes a spectacular crater around line 820. A look at the printout of this area confirms that the area first found by LKFR is very rough and full of detail.) With the minor modifications suggested, we expect SKINNY could be used on full framelets with approximately twice the speed the present SKINNY operates. This work should take an expert programmer about 40 man hours to complete, fully generalizing SKINNY to operate on full framelets.

For reference and study of the nonlinearity of the mechanical scan, we summarize the match points found by SKINNY on full framelets in Table 3.1. We aligned by hand the edges so that 1-1 was a perfect row match.

028 ROW	027 ROW	028 ROW	027 ROW
183	181	8208	8221
490	483	8316	8324
822	820	8617	8622
1082	1084	9332	9341
1312	1319	9579	9587
1667	1676	9773	9779
1895	1903	10140	10149
2000	2010	10825	10830
2364	2373	11084	11085
2590	2601	11164	11167
3168	3164	11524	11525
3454	3475	11898	11894
3827	3845	12075	12076
4171	4186	12434	12434
4669	4679	12705	12705
5013	5028	12883	12885
5388	5398	13049	13053
5740	5755	13286	13289
6317	6336	13779	13783
6671	6690	14031	14039
6899	6921	14396	14409
7125	7152	14488	14499
7678	7700	14854	14869
7892	7909	15291	15311

Table 3.1 Match points found by SKINNY, 028-027

TABLE OF CONTENTS

PROJECT B — IMPROVEMENT OF PHOTOMETRIC DATA
PROJECT C — STATISTICAL ERROR ANALYSIS OF PHOTOMETRIC DATA

Table of Contents -- Continued

LIST OF FIGURES

List of Figures -- Continued

List of Figures -- Continued

List of Figures -- Continued

LIST OF TABLES

PART I D LOG E CORRECTIONS

INTRODUCTION The problem of using photographic data in photogrammetric problems is complicated by the fact that neither the density D nor transmittance $T = \frac{1}{D}$ is linearly related to the exposure values (which are, of course, linearly related to the brightness of the scene, ignoring lens distortion effects). Since photometry utilizes the brightness of the scene, one must convert the data, given as transmittances, to (e. g.) exposure values, from which the scene brightness can be found knowing the properties of the camera and lens. The LO analysis program utilizes a table look-up method to form the corrections. That is, after digitization the numbers are converted from transmittances to exposure values by looking in a table for the converted number. The purpose of this part of Project B is to attempt improvement of the existing "D log E curve", i.e., the values in the table. Our key findings:

1. Spectacular improvements are possible.
2. Operationally, the making of a new D log E curve is not usually necessary with different digitizations, even with clipping.
3. The making of a new D log E curve should be replaced by changes in the analysis program of the exposure values in meter-candle-seconds the analysis is to take for 0 and 63. This does not affect the D log E curve, since in the range of LO data we are working with the D log E curve is approximately linear.

There is another problem which complicates our particular problem. That is that the gray shade calibration data (which one should use to set the point values of the D log E table) is (a) not trivial to address perfectly, (b) rendered practically useless at times by clipping, and, (c) effected by the line scan signature.

Since the problem of the line scan signature must be solved to do both Projects A and B (the algorithm in Project A to locate match points was partly photometric), we begin by discussion of the improved correction factors.

1. CORRECTION FACTORS

In order to obtain correction factors to partly remove the line scan signature, we perform (operationally) the following steps:

1.1 Accumulate LO data from the mission and (if possible) digitization, and use in a program like CAS3 to obtain averages of long columns of actual LO data.

1.2 Remove the drummark areas and any "scratches" noted and apply a smoothing filter like CFAC to smooth the column averages.

1.3 Replace any "scratches" noted and use these correction factors (after normalization and a division) to remove the line scan signature.

Our data was obtained from either

(1) About three chits each from three framelets 028, 027, and 025 of LO III, frame 154H; or

(2) The three complete framelets 028, 027 and 025 from the same area.

Very little difference was found in the smoothed correction factors (although substantial differences were noted in the unsmoothed ones) between 2068 rows and a full framelet (about 15,500 rows). The routine reported on next takes the column averages for use in the smoothing program. We then furnish documentation (in Section 3) of the smoothing routine we found to be a substantial improvement on the edges of the framelets, necessary for use in the algorithm SKINNY reported on in Project A.

COMPUTER PROGRAM DOCUMENTATION

PROGRAM CAS 3

Project B

By

Jack Bryant

and

R. L. Wendt

Prepared by
Applied Scientific Research, Inc.
Houston, Texas
Under Contract NAS 9-10477

for

MAPPING SCIENCE BRANCH
National Aeronautics and Space Administration
Manned Spacecraft and Center
Houston, Texas

September, 1971

2. COMPUTER PROGRAM DOCUMENTATION - CAS 3 (Column Averages Surveyor 3)

2.1 PURPOSE/METHOD

The program supplies column averages for use as input data by program CFAC.

The program initializes the data tape by spacing over the header and notch data. The framelet data is read and the elements of each column are summed by COLSUM a line at a time. A count of the number of lines is recorded and the final sum for each column is divided by the number of lines. The column sums are plotted, the sums are written on the drum and the process is repeated for the remaining framelets.

The drum is set to its initial address and the average of the column sums is obtained. The correction factors are printed, plotted and punched and the input tape is rewound.

2.2 USAGE

22.1 INPUT DESCRIPTION:

The edited tape from LO III Surveyor - 3 frame 154 H framelets 028, 027, 025, read from Unit 1.

2.2.2 OUTPUT DATA FURNISHED

Punched cards to be used by CFAC. Film output from 4020.

2.2.3 PROGRAM RUN PREPARATION

2.2.3.1 Card Deck Setup

See Fig. 2.2.3.1.1 for the deck setup, which includes the required control cards.

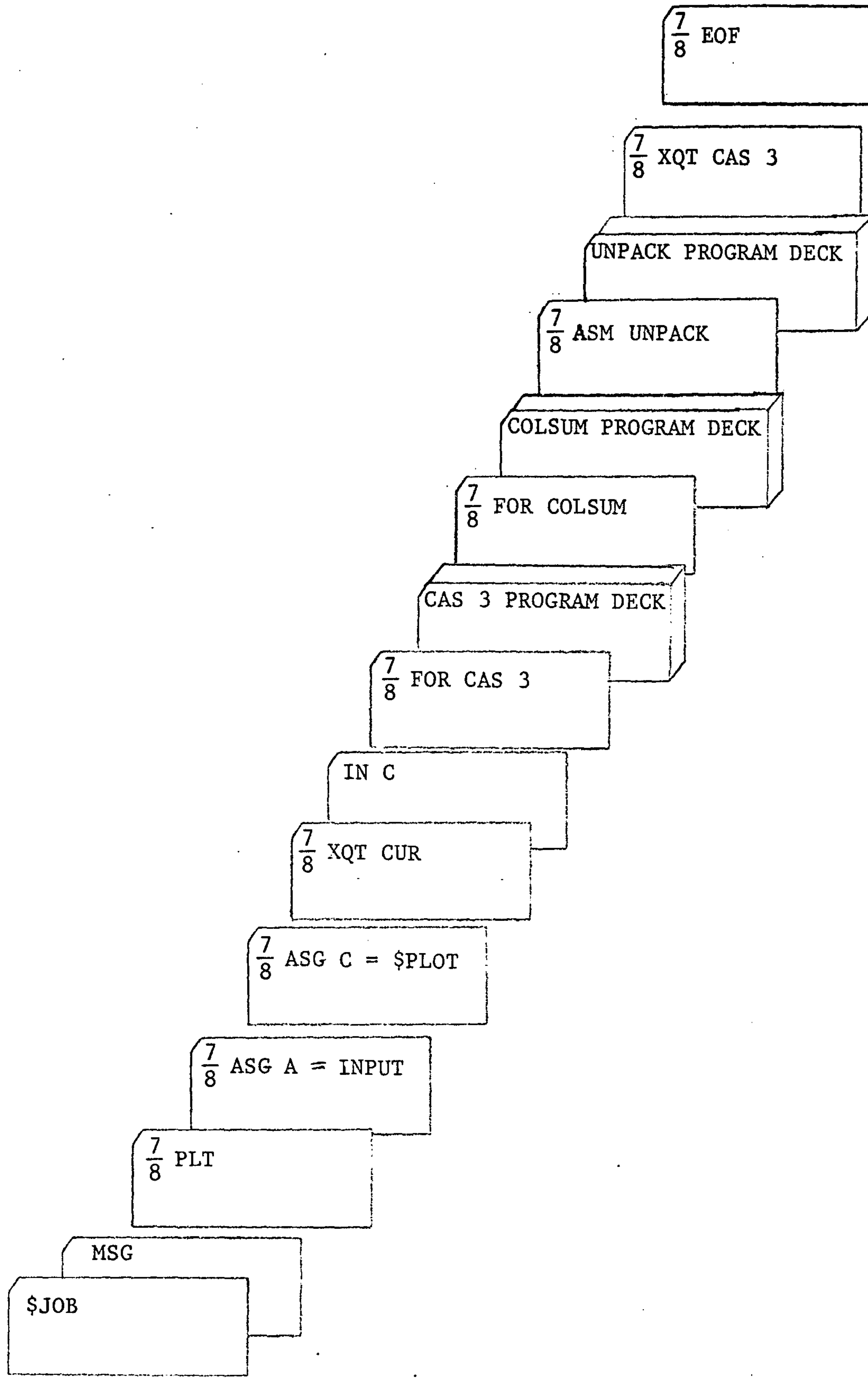


Fig. 2.2.3.1.1. DECK SETUP (UNIVAC 1108) CAS 3.

2.2.3.2 Required I/O Devices

The program uses the standard FORTRAN input on Unit 5 and output from Unit 6. In addition, the program uses NTRAN read statements to read the chit data from tape on Unit 1 and NTRAN to handle one FH432 drum file. The plot routines occupy one FSTRN track.

2.2.3.3 Subroutine Requirements

The subroutine COLSUM is required. (It is in addition to the furnished system subroutines NTRAN, GRIDIV and APLOTV.) COLSUM requires the standard LO subroutine UNPACK. The subroutine COLSUM is described in Section 2.4.3.

2.2.4 EXECUTION CHARACTERISTICS

2.2.4.1 Restrictions

The program as it is written now requires a special data input format (special in the sense that only one tape is read, on which there are to be three main blocks of specially edited LO data). Obvious modifications in the NTRAN read statements allow use on any number of framelets.

2.2.4.2 Storage Requirements

The storage requirement for the code is 526_8. The data takes $3,477_8$ locations. System plot and service routines take about $15,000_8$ locations.

2.2.4.3 Run Time / Lines of Output / Other Output

The usual difficulties with tape dependent jobs affects the run time here. In effect, the job runs at tape read speed. In addition to the program listing, there are approximately three pages of output for each block of data (i.e. subframelet) used. A film strip giving a plot of the unsmoothed column averages for each framelet and overall should accompany the output.

2.2.4.4 Accuracy / Validity

Since exact arithmetic is done at each step preceeding the divide, the numbers furnished are as good as the FORTRAN divide. It may, of course, be noted that certain systematic features (such as, for example, the dark edge of a large crater in one side of one framelet) would affect the validity of the COLUMN averages, as will excessive clipping. It seems unlikely that these errors are significant for the data under consideration here.

2.3 GENERAL FLOW CHART

See Fig. 2.3.1

2.4 REFERENCE INFORMATION

2.4.1 DETAILED FLOW CHART

See Fig. 2.4.1.1 for a detailed flow chart for CAS 3.

2.4.2 DESCRIPTION OF VARIABLES

See Table 2.4.2.1 for a detailed description of the variable in CAS 3.

2.4.3 DESCRIPTION OF SUBROUTINE COLSUM

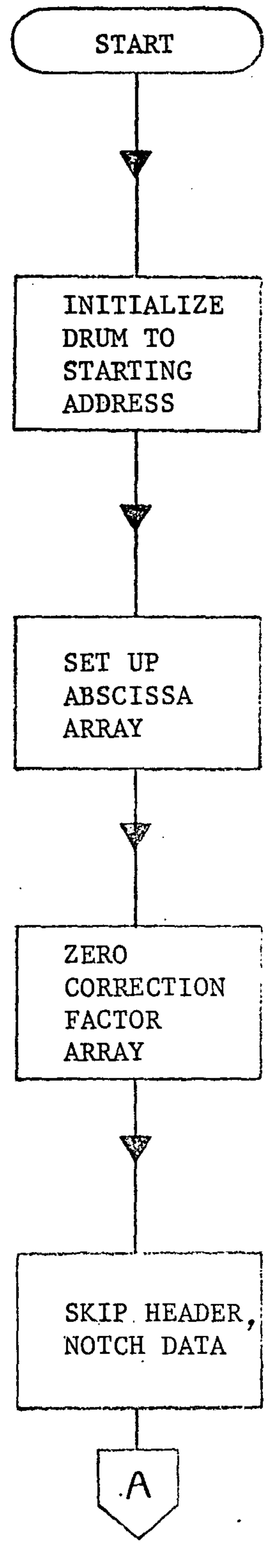


Fig. 2.3.1 GENERAL FLOW CHART - CAS 3

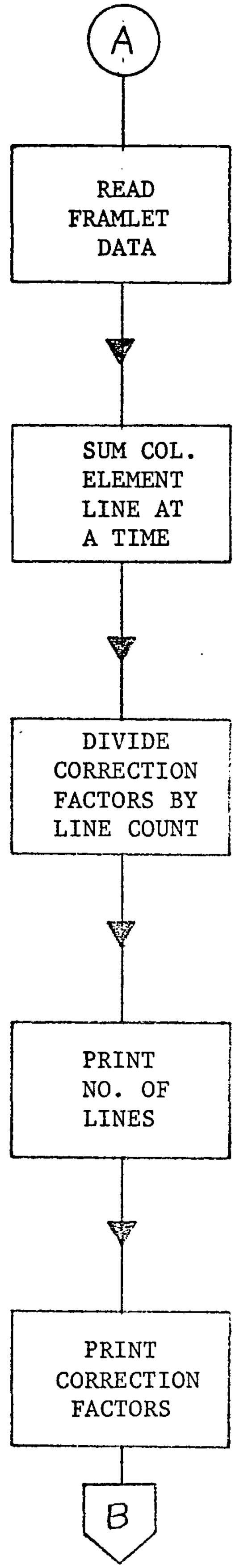


Fig. 2.3.1 CONTINUED

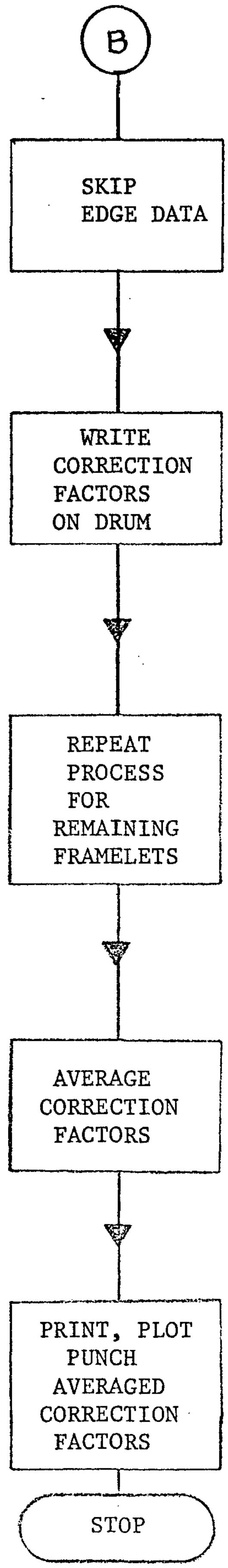


Fig. 2.3.1 CONTINUED

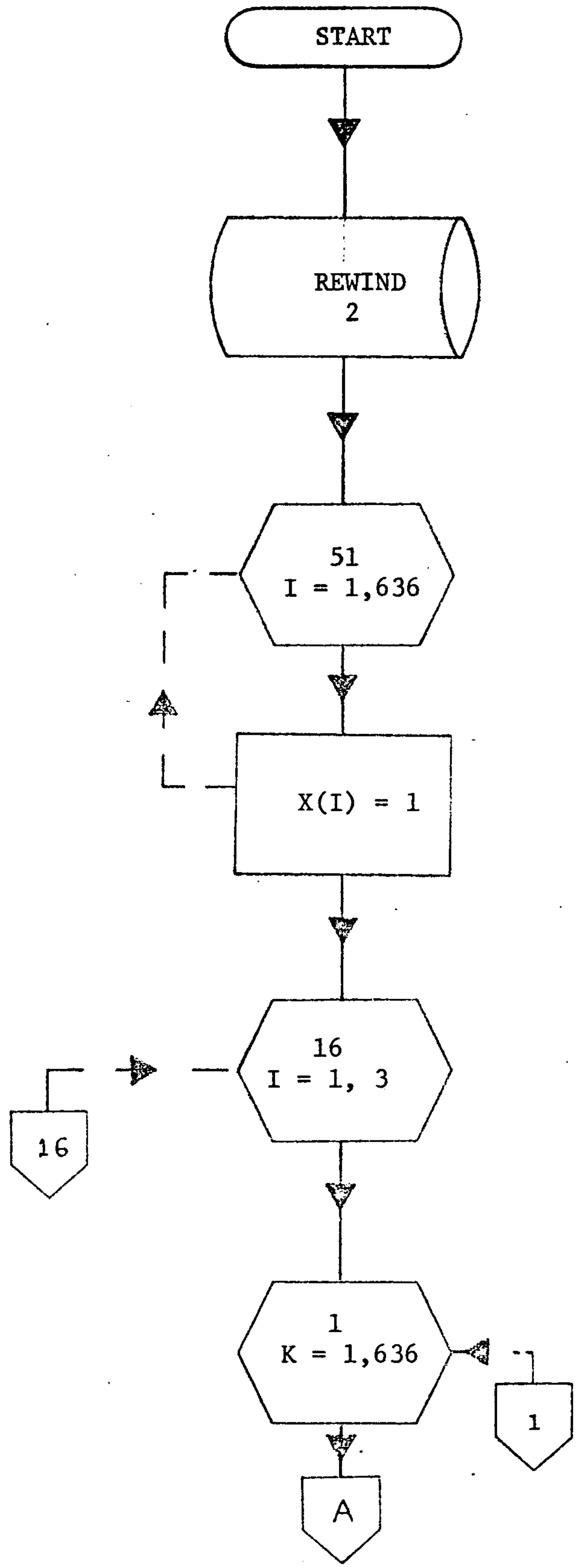


Fig. 2.4.1.1 DETAILED FLOW CHART - CAS 3

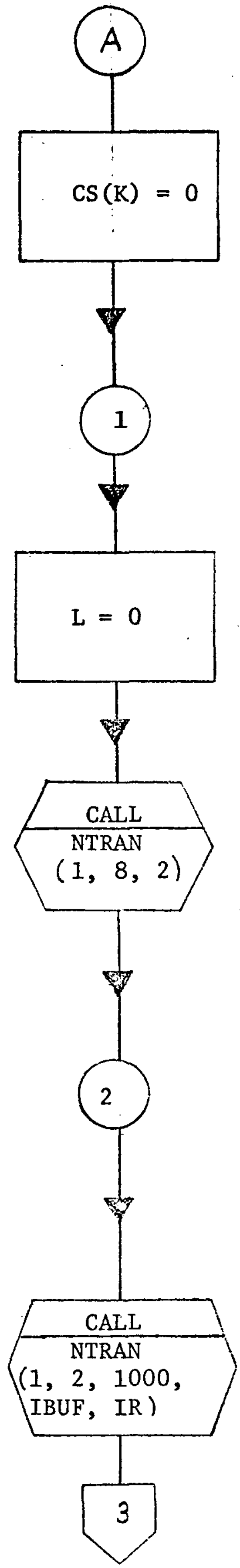


Fig. 2.4.1.1 CONTINUED

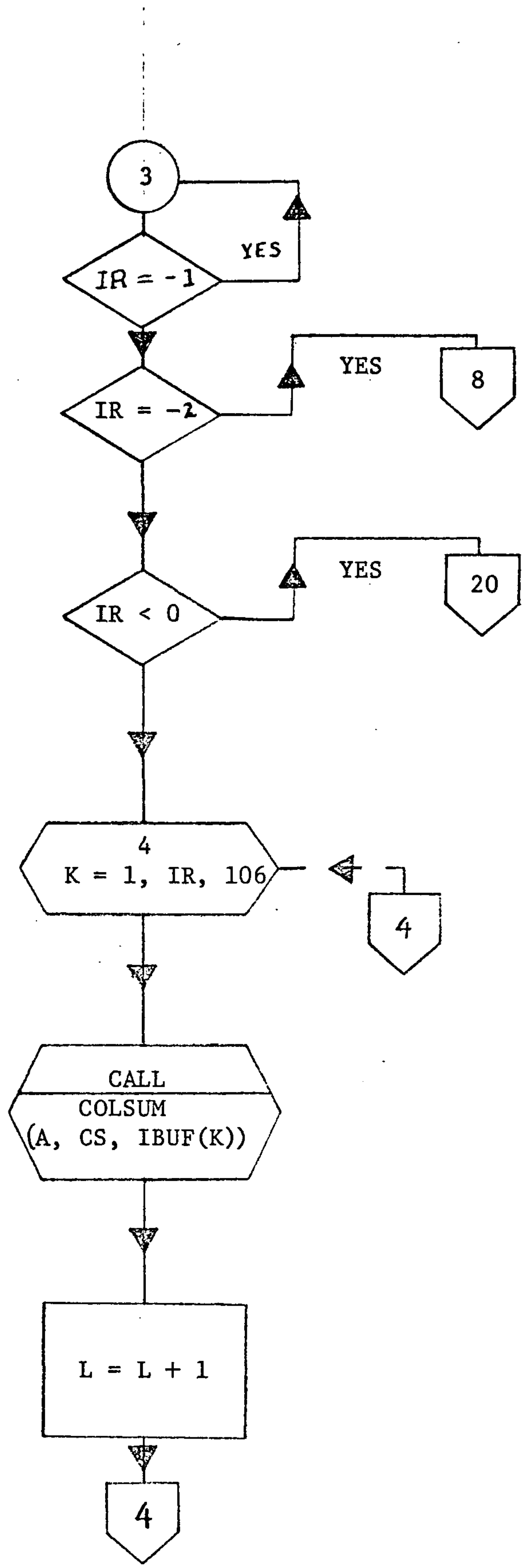


Fig. 2.4.1.1 CONTINUED

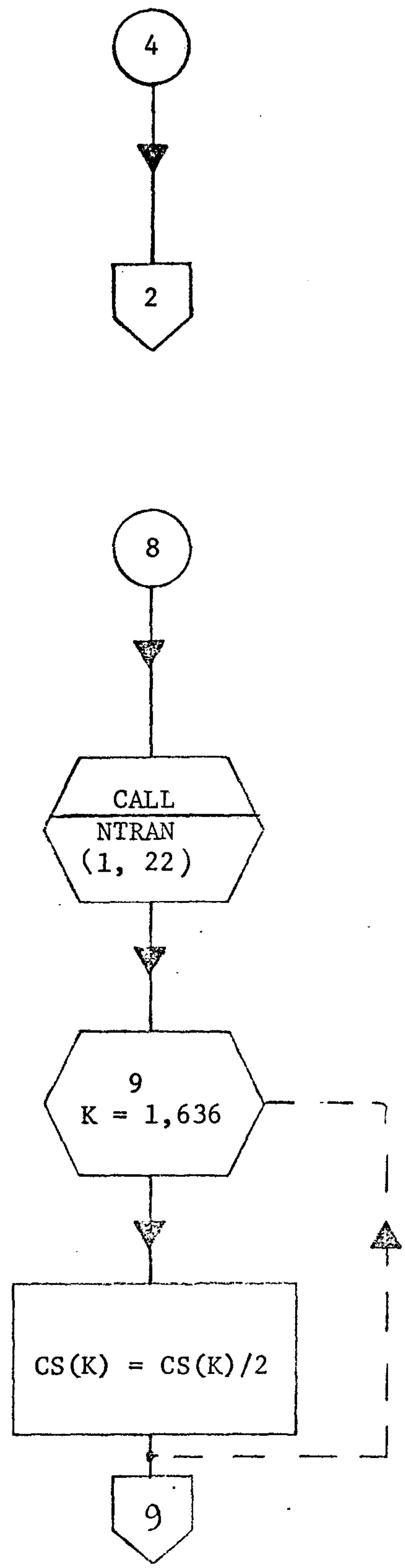


Fig. 2.4.1.1 CONTINUED

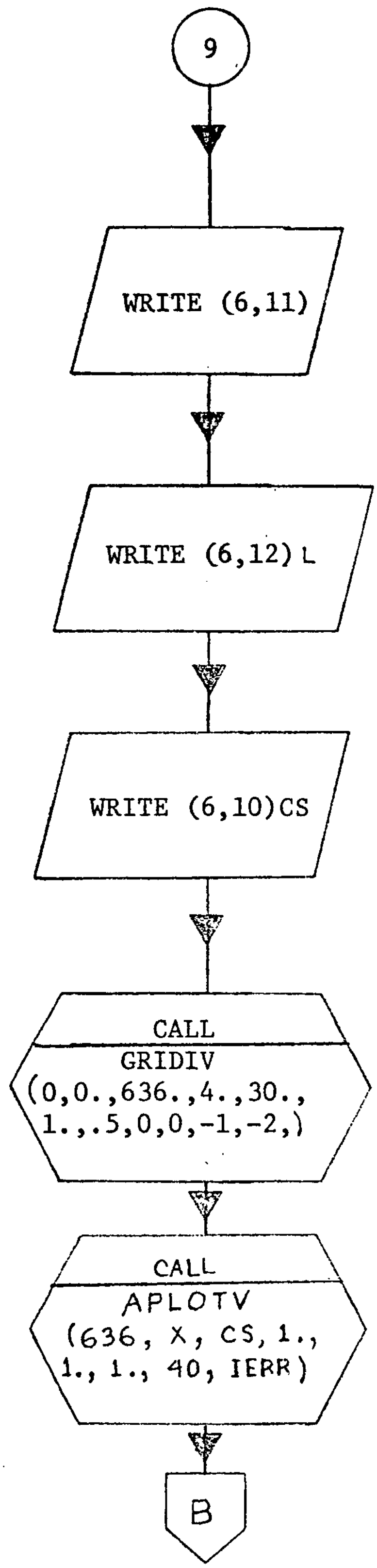


Fig. 2.4.1.1 CONTINUED

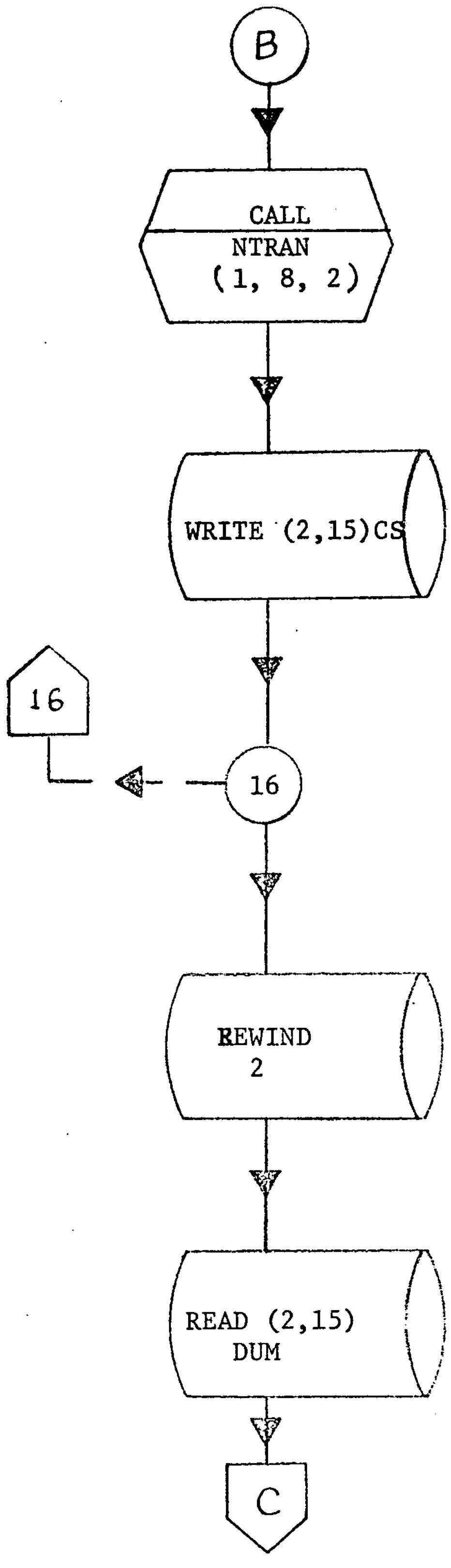


Fig. 2.4.1.1 DETAILED FLOW CHART - CAS 3

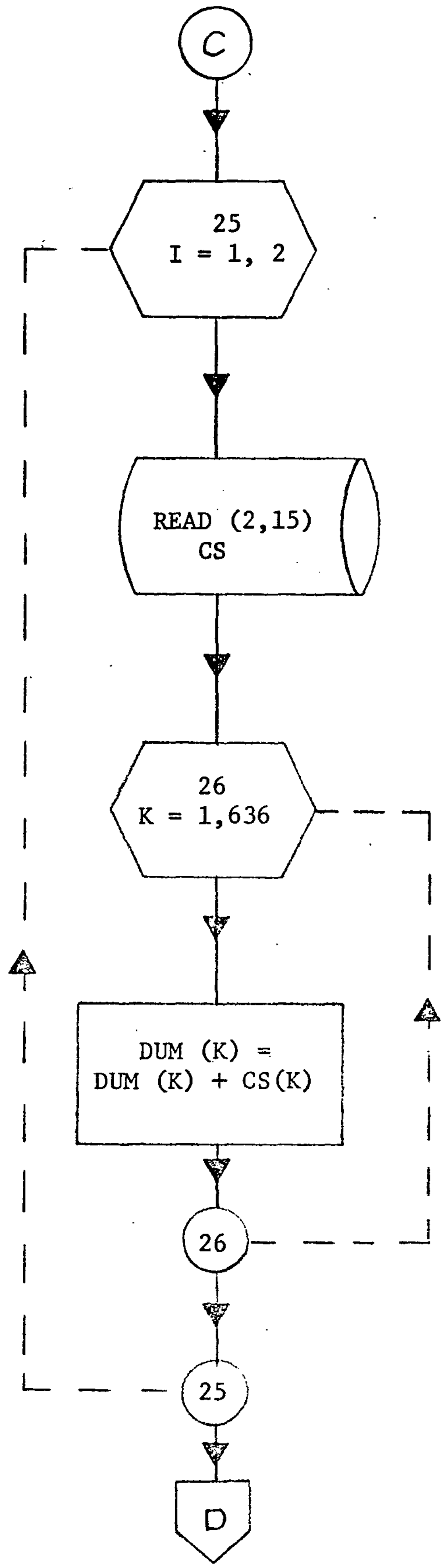


Fig. 2.4.1.1 CONTINUED

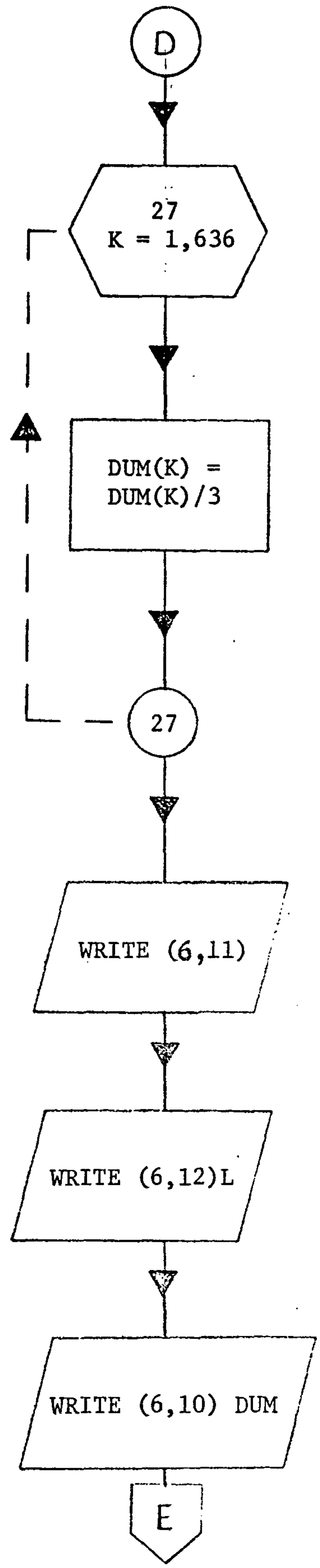


Fig. 2.4.1.1 CONTINUED

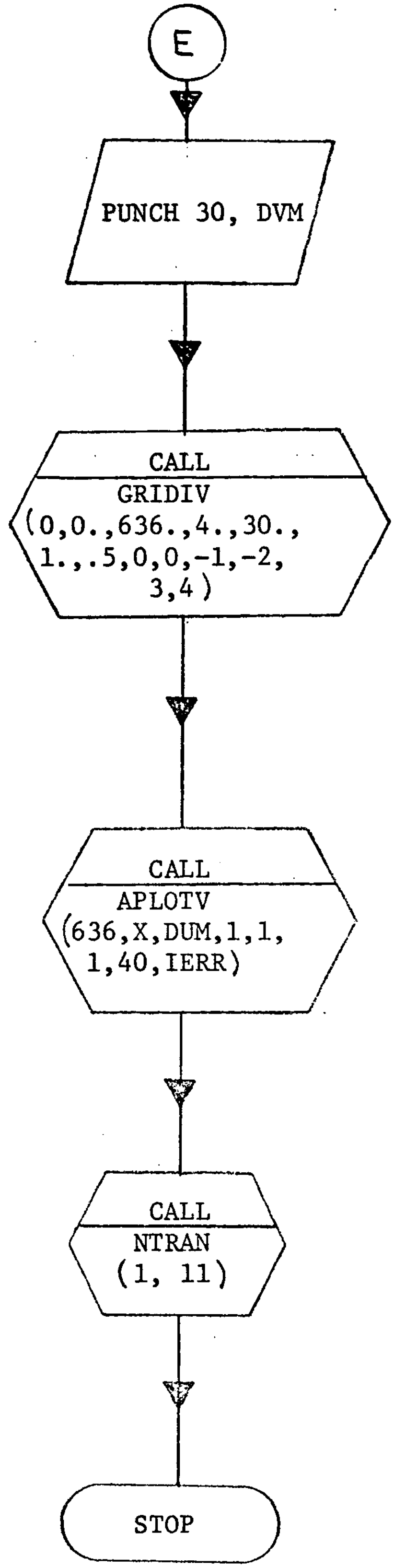


Fig. 2.4.1.1 CONTINUED

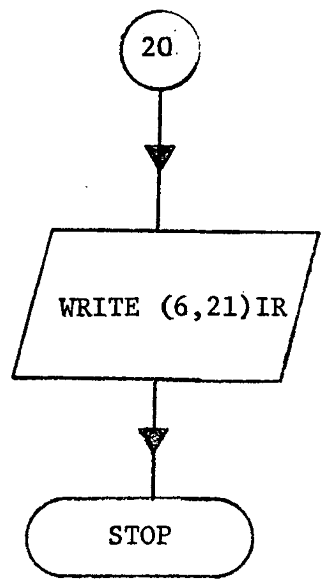


Fig. 2.4.1.1 DETAILED FLOW CHART - CAS 3

IBUF - Array of dimension 10000; used to input edited data.

A - Integer array of dimension 636; used to unpack data.

CS - Array of dimension 636; the column averages or unsmoothed correction factors.

DUM - Array of dimension 636; a dummy array.

X - Array of dimension 636; used to generate abscissas.

Table 2.4.2.1 DESCRIPTION OF VARIABLES - CAS 3.

SUBROUTINE COLSUM

IDENTIFICATION

Name/Title	- COLSUM (Column Sums)
Author/Date	- R. L. Wendt, September 1970
Organization	- ASR
Machine Identification	- UNIVAC 1108
Source Language	- FORTRAN V

PURPOSE

This routine sums the elements of the framelet down the framelet in the direction of increasing row numbers.

USAGE

Calling Sequence

CALL COLSUM (A, CS, IBUF(K))

where A, CS, IBUF are as in the main program.

In COLSUM, IBUF is called C.

DETAILED FLOW CHART

See Fig. 2.4.3.1 for a detailed flow chart of COLSUM.

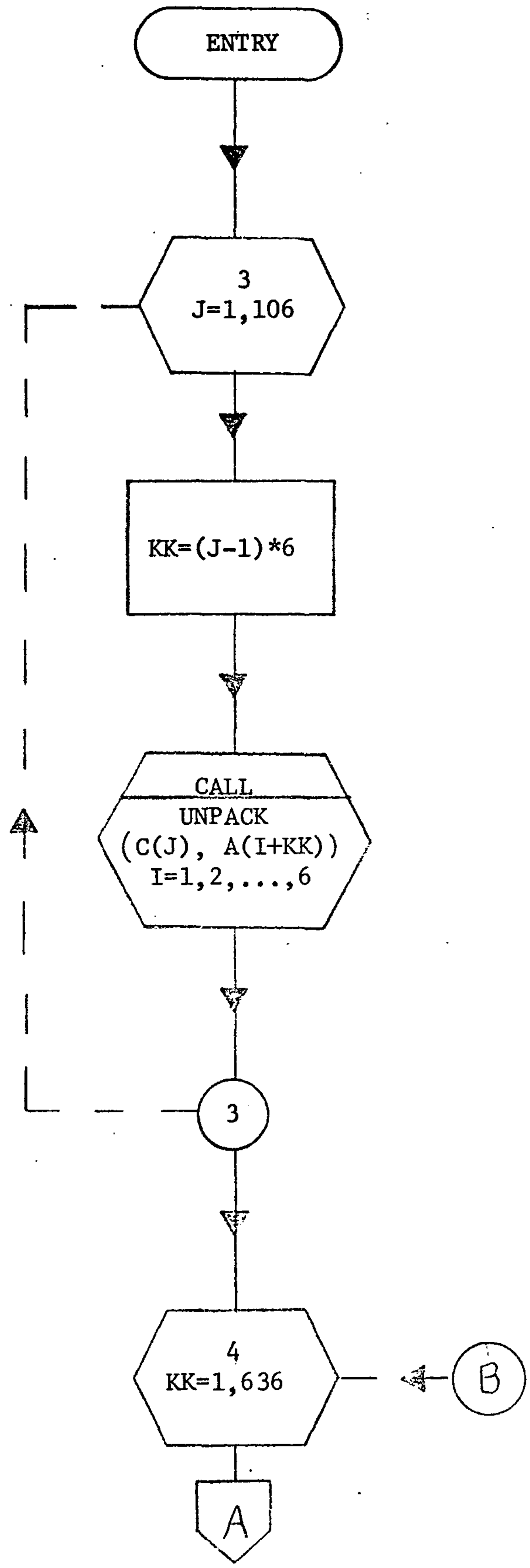


Fig. 2.4.3.1 Detailed Flow Chart, COLSUM.

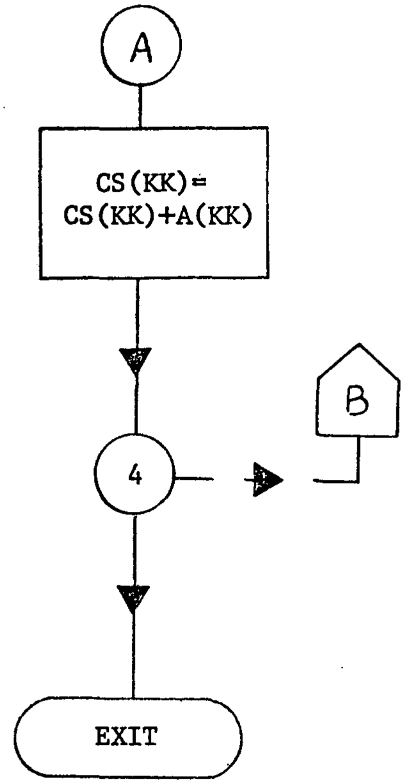


Fig. 2.4.3.1 Detailed Flow Chart, COLSUM

COMPUTER PROGRAM DOCUMENTATION

Program CFAC

Project B

by
Jack Bryant
and
R. L. Wendt

Prepared by
Applied Scientific Research, Inc.
Houston, Texas
Under contract NAS 9-10477

for
MAPPING SCIENCE BRANCH
National Aeronautics and Space Administration
Manned Spacecraft Center
Houston, Texas
September, 1971

3. COMPUTER PROGRAM DOCUMENTATION - CFAC (Correction Factors)

3.1 PROGRAM DESCRIPTION

3.1.1 GENERAL DESCRIPTION

The program initializes all parameters and sets up the abscissa vector for APLOTV. The unsmoothed correction factors are read in and CF(1) is set equal to CF(2). The first element of the digitized data is not reliable because of a digitizer problem so setting CF(1) = CF(2) helps to alleviate this problem. The unsmoothed correction factors are plotted. The drummarks are removed by linear interpolation. The array of cosines is initialized and the smoothing process begins by performing the normalized convolution first on the center i.e. elements from I1 to I2, then on the left edge of the framelets i.e. elements from 1 to J2 and lastly on right edge i.e. elements from I3 to 636. The correction factors and smoothed correction factors are printed and the smoothed correction factors plotted. The smoothed correction factors are normalized, printed, plotted and punched for later use.

3.1.2 MATHEMATICAL DESCRIPTION OF THE FILTER

Mathematically, the main purpose of CFAC is to apply a normalized filter to the unsmoothed correction factors. Initially in the development of the routine CFAC, a rectangular filter (in the real domain) was used. This filter turned out to ring in the presence of of the kind of noise present in the column averages, and,accordingly, was replaced by the cosine squared filter used in the present CFAC. The new filter shows remarkably little tendency to ring, with the resultant correction factors being, as far as our tests have revealed, much better.

The cosine squared filter here is described as follows: (for the particular program run used to smooth the CAS3 correction factors).

Let the unsmoothed correction factors be u(1), ..., u(636). The first step is to replace u(1) by u(2). The drummark areas 7, 8, 9, 10, 11, 12 (and 624, 625, 626, 627, 628, 629) are replaced by a linear interpolation of the adjacent number u(6) and u(13) (respectively u(623) and u(630)). The program then proceeds to apply the following formula giving the smoothed correction factors: If s(1), ..., s(636) is the array of smoothed correction factors, then

$$s(i) = \frac{1}{A(i)} \sum_{j=\ell(i)}^{n(i)} \left(\cos^2 \frac{\pi}{100} j\right) \cdot u(j+i)$$

where

$$A(i) = \sum_{j=\ell(i)}^{n(i)} \cos^2 \frac{\pi}{100} j$$

and

$$n(i) = \min\{i+49,\ 636\}$$

$$\ell(i) = \max\{i-49,\ 0\}.$$

This has the effect of taking care of the edges correctly (at least well enough so that SKINNY and operate and so that good D log E curves can be made).

3.2 USAGE

3.2.1 DESCRIPTION OF INPUT

Data cards (unsmoothed correction factors) from program CAS3 or equivalent.

3.2.2 DESCRIPTION OF OUTPUT

Card output - Punched cards containing the smoothed correction factors.

Printed output - The unsmoothed and smoothed correction factors are printed.

Other output - Film output containing plots of the smoothed and unsmoothed correction factors is also furnished.

3.2.3 RUN PREPARATION

For a card deck setup see Figure 3.2.3.1. Other than 4020 output, no special Input/Output devices are required. No subroutines are used.

3.2.4 EXECUTION CHARACTERISTICS

Restrictions --J1 and J2 < 318.

Storage requirements - code, 614_8; data, 4033_8; systems plot routines, $15,000_8$.

Run time - About 30 seconds (1108)

Lines output - 15 pages

Other output - 4020 Film strip

Accuracy/validity - While the choice of parameters J1, J2 largely determines the action of the filter, the choice J1 = J2 = 49 results in the best correction factors in our work.

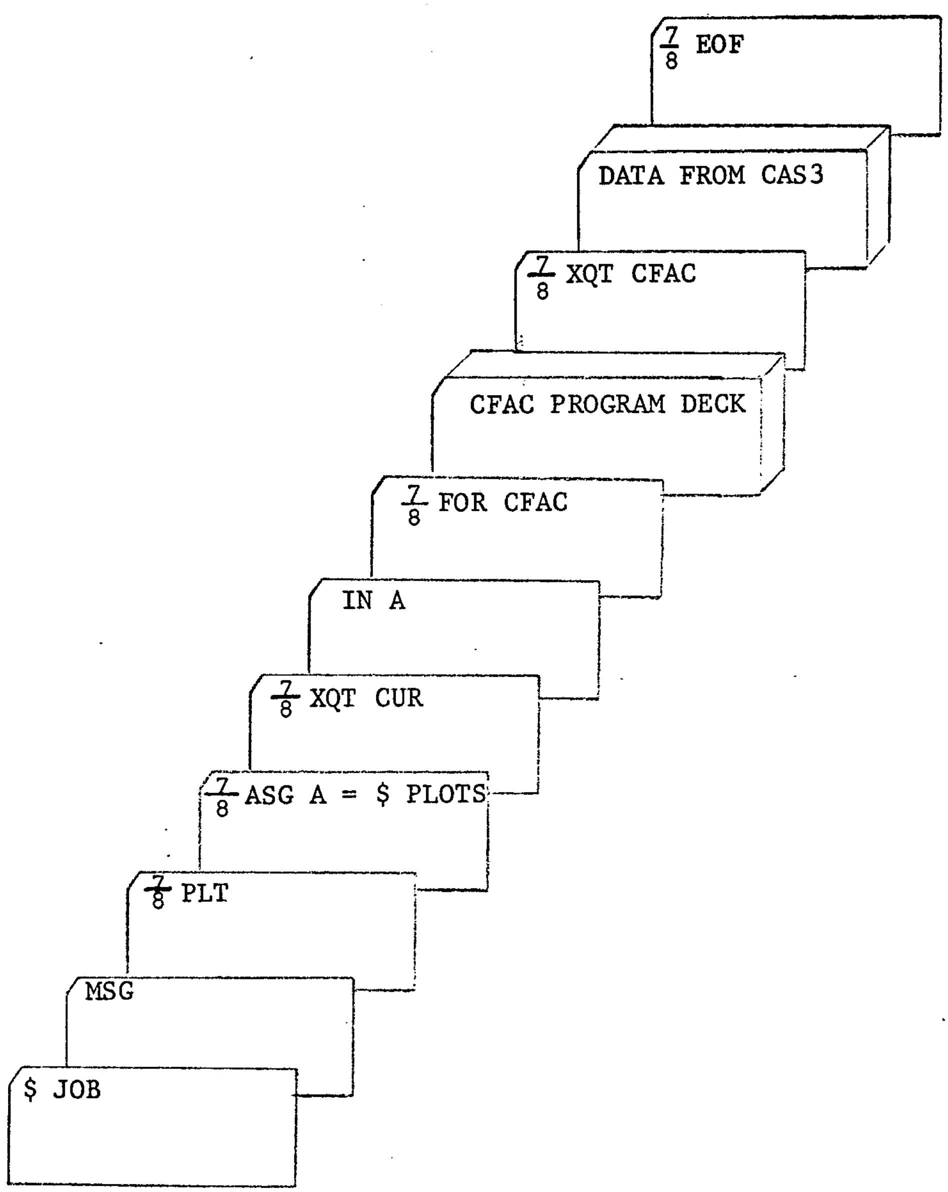


Figure 3.2.3.1 DECK SETUP (UNIVAC 1108) CFAC

3.3 GENERAL FLOW CHART

See Figure 3.3.1 for general flow chart.

3.4 REFERENCE INFORMATION

See Figure 3.4.1 for a detailed flow chart.

See Table 3.4.2 for a description of variables.

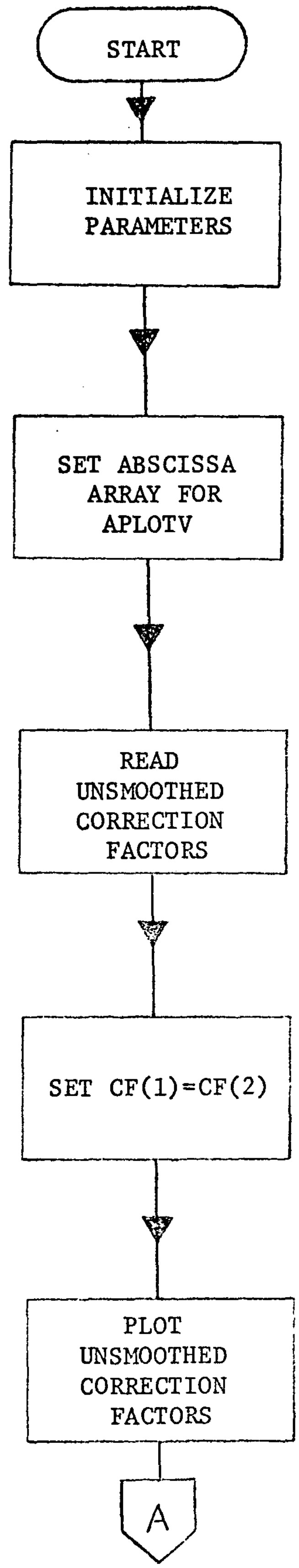


Figure 3.3.1 GENERAL FLOW CHART - CFAC

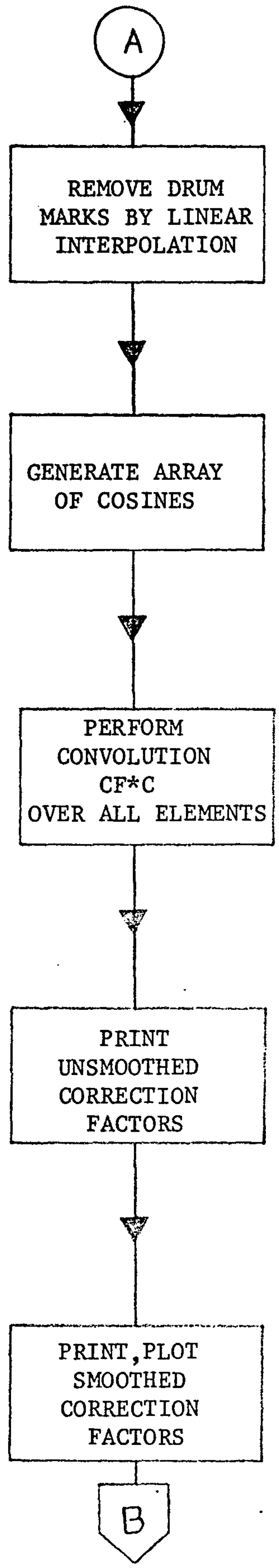


Figure 3.3.1 GENERAL FLOW CHART - CFAC (Continued)

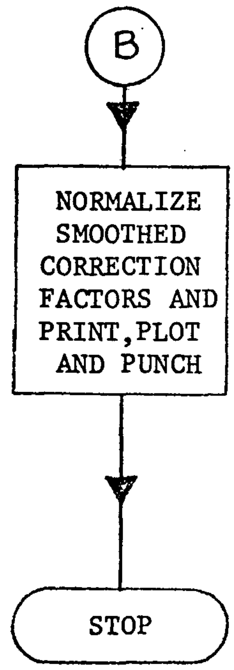


Figure 3.3.1 GENERAL FLOW CHART - CFAC (Concluded)

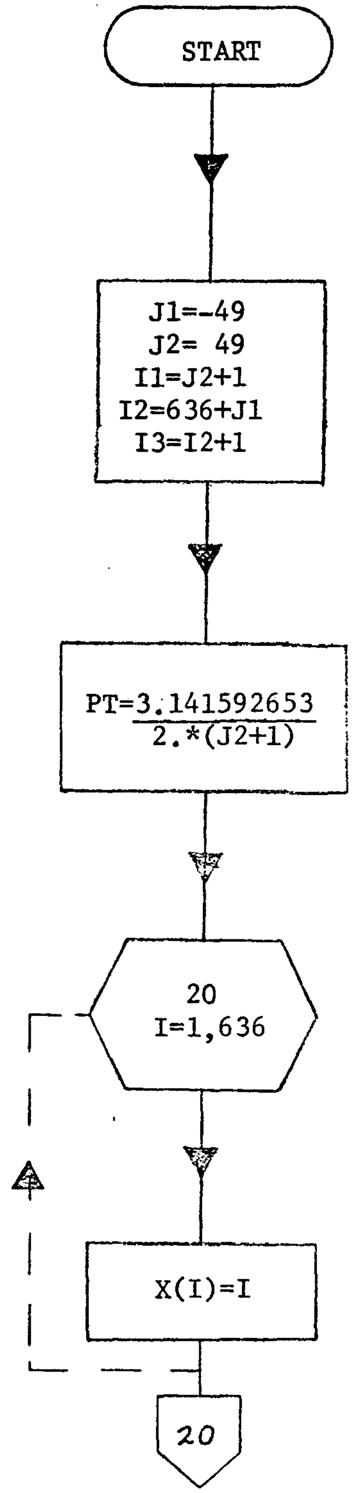


Figure 3.4.1 DETAILED FLOW CHART - CFAC

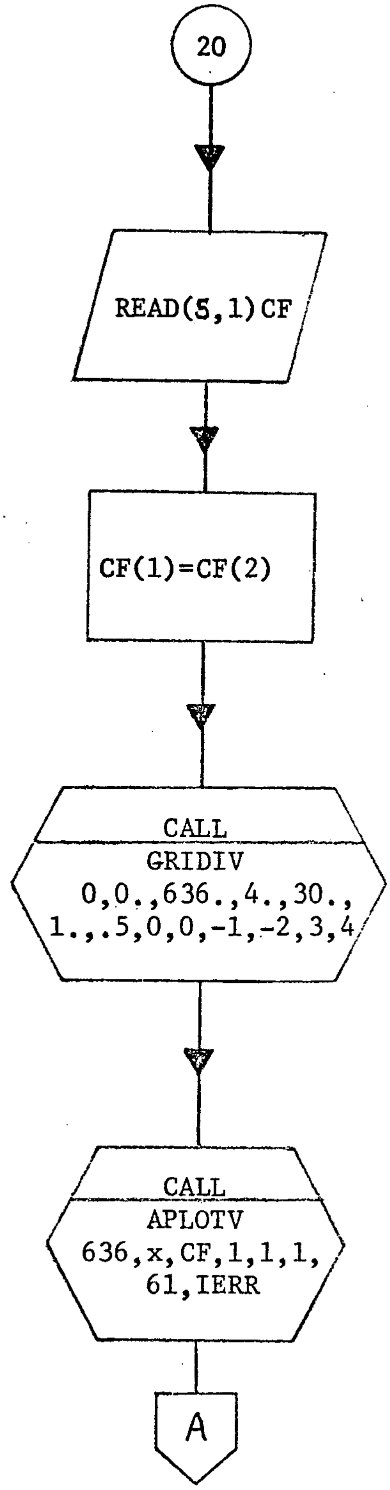


Figure 3.4.1 DETAILED FLOW CHART - CFAC (Continued)

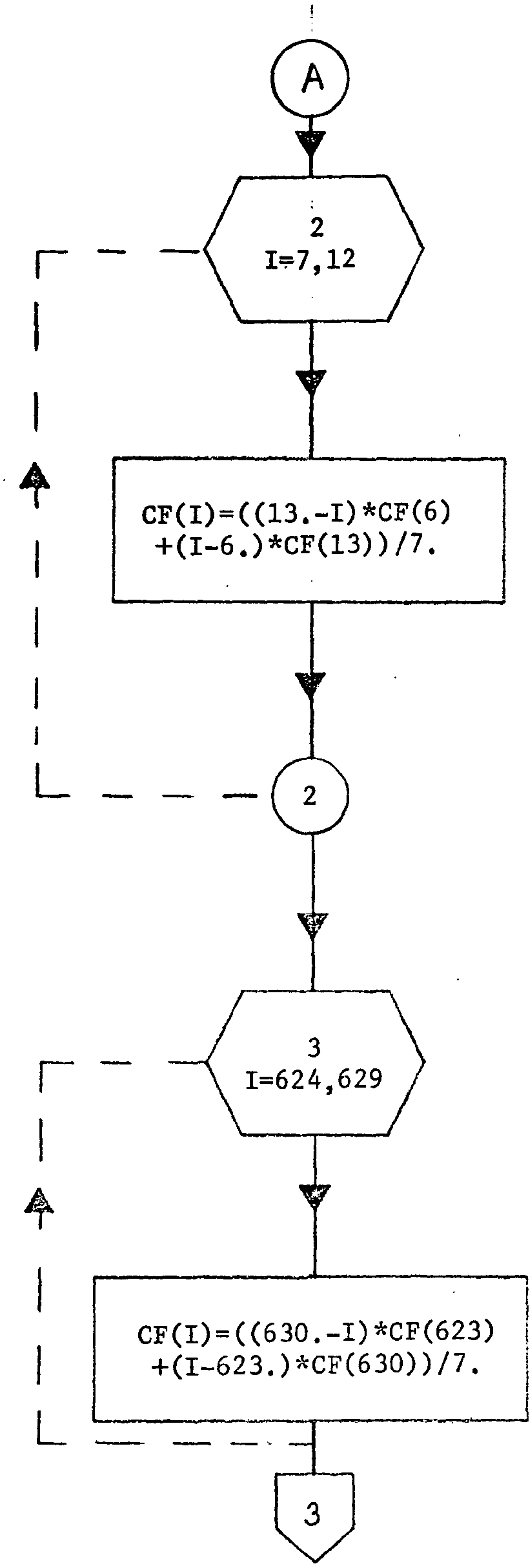


Figure 3.4.1 DETAILED FLOW CHART - CFAC (Continued)

C4

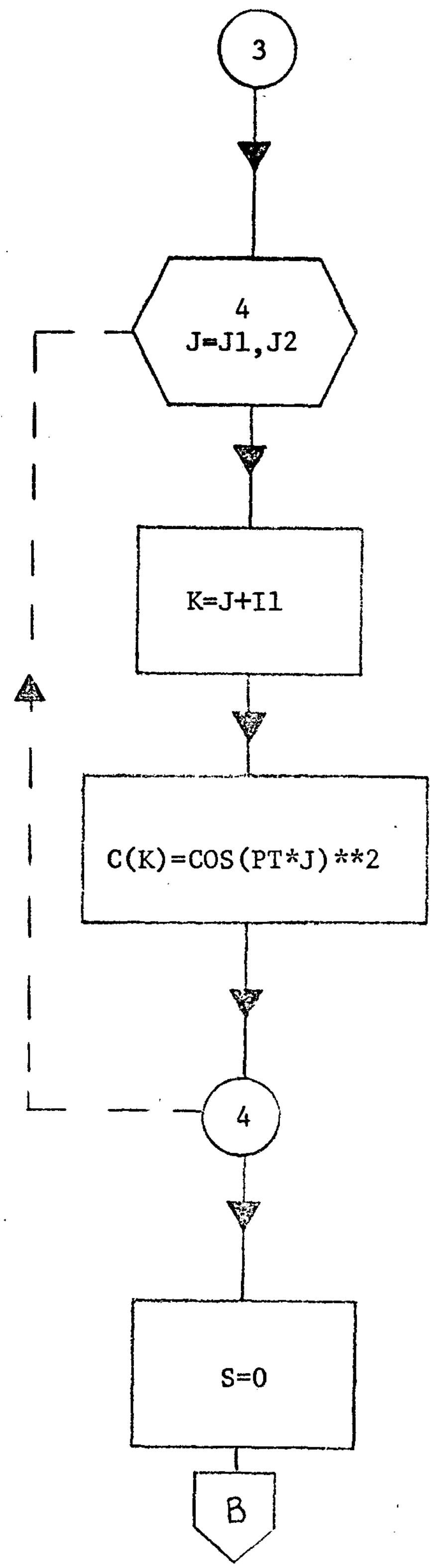


Figure 3.4.1 DETAILED FLOW CHART - CFAC (Continued)

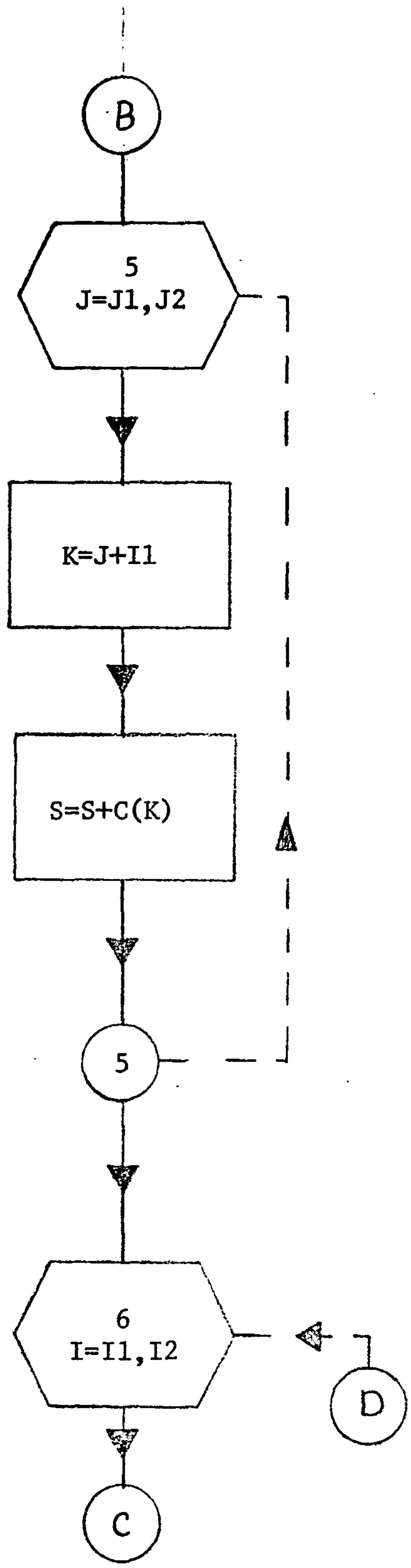


Figure 3.4.1 DETAILED FLOW CHART - CFAC (Continued)

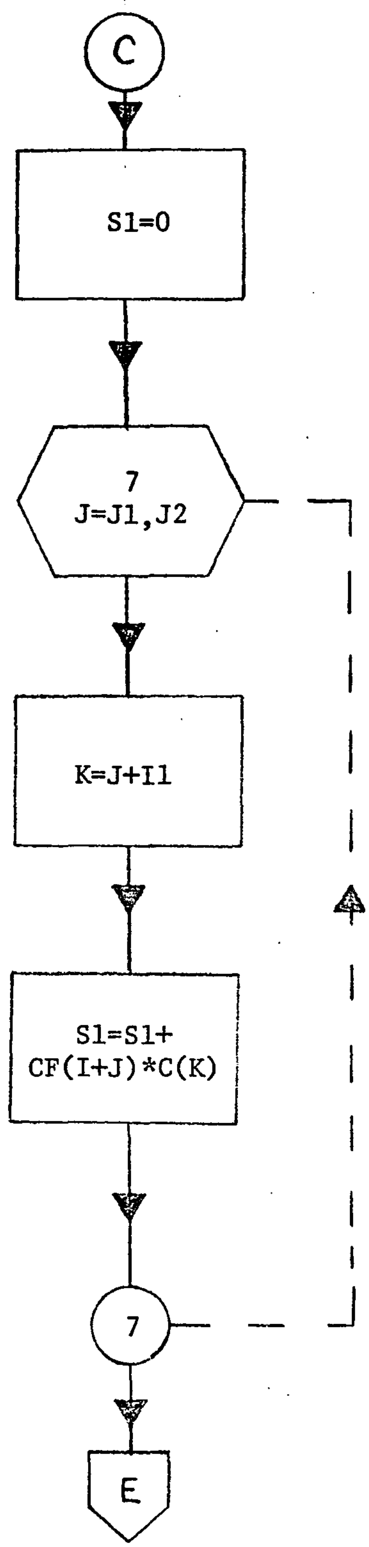


Figure 3.4.1 DETAILED FLOW CHART - CFAC (Continued)

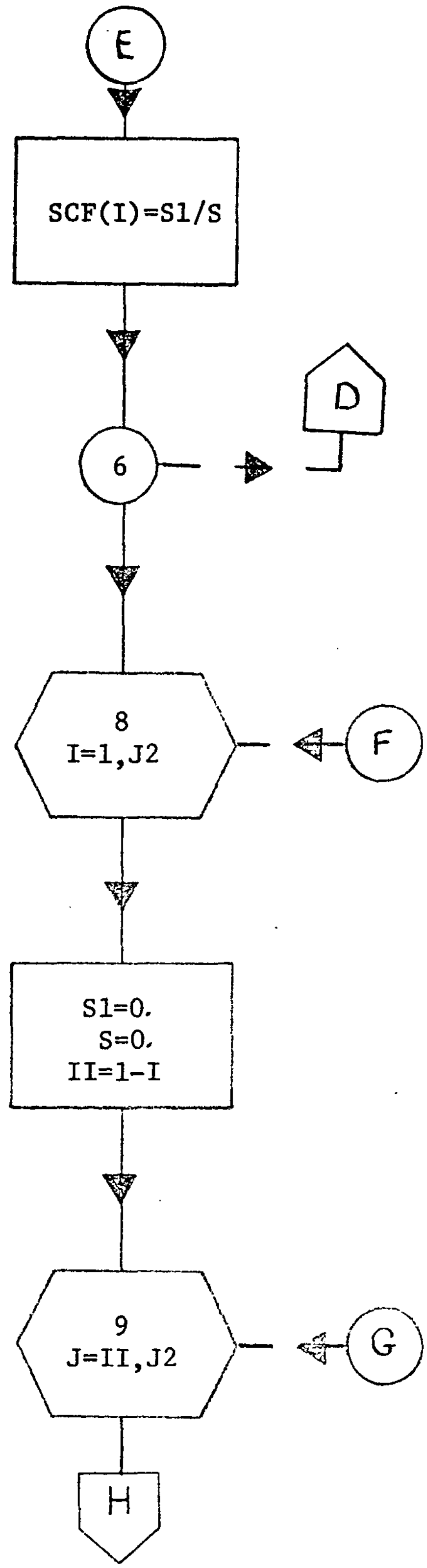


Figure 3.4.1 DETAILED FLOW CHART - CFAC (Continued)

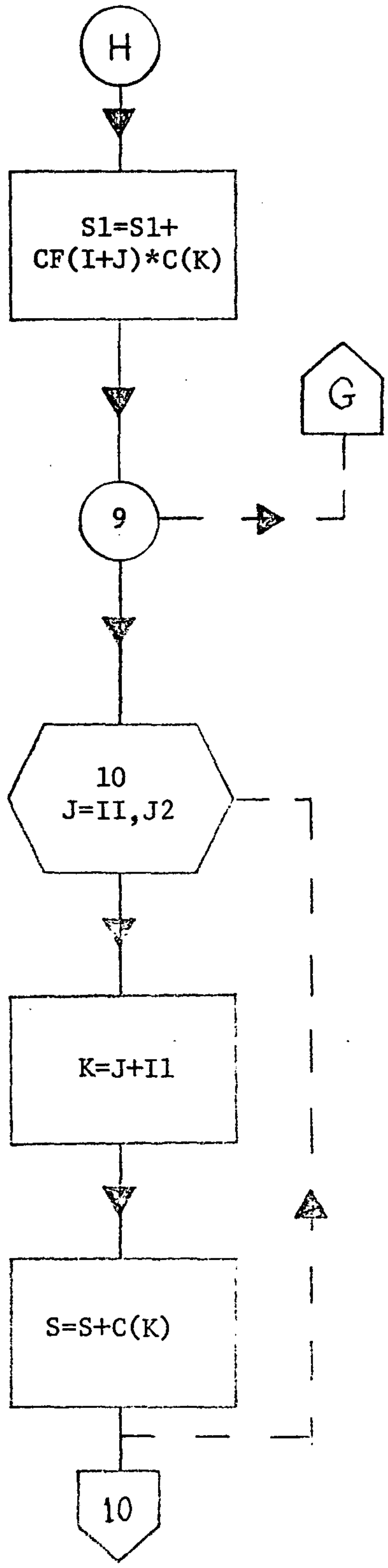


Figure 3.4.1 DETAILED FLOW CHART - CFAC (Continued)

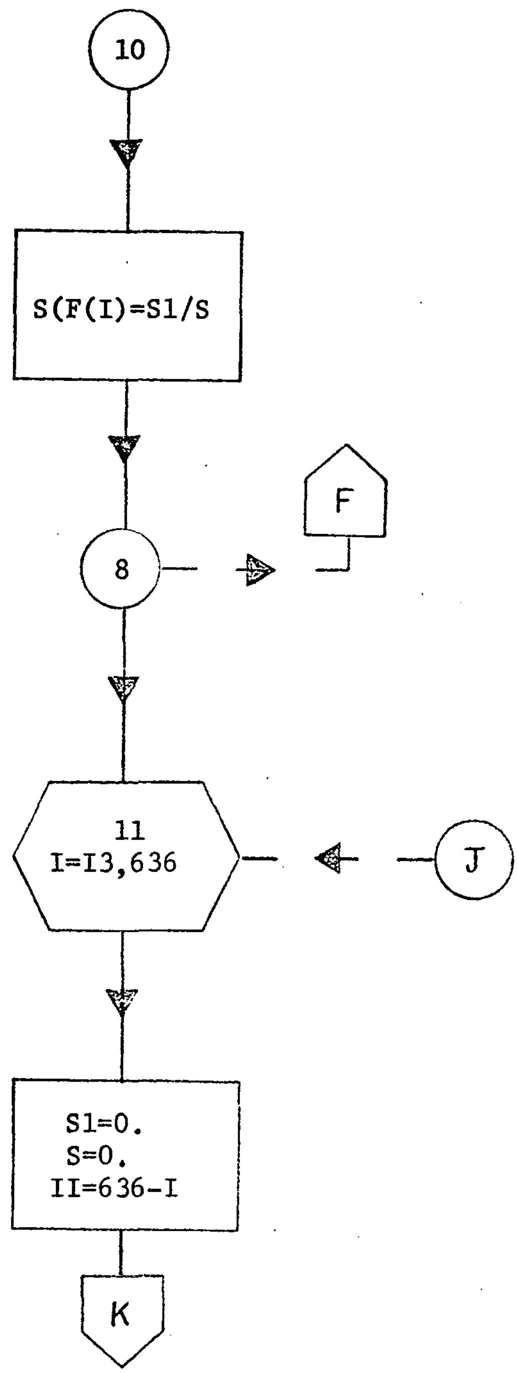


Figure 3.4.1 DETAILED FLOW CHART - CFAC (Continued)

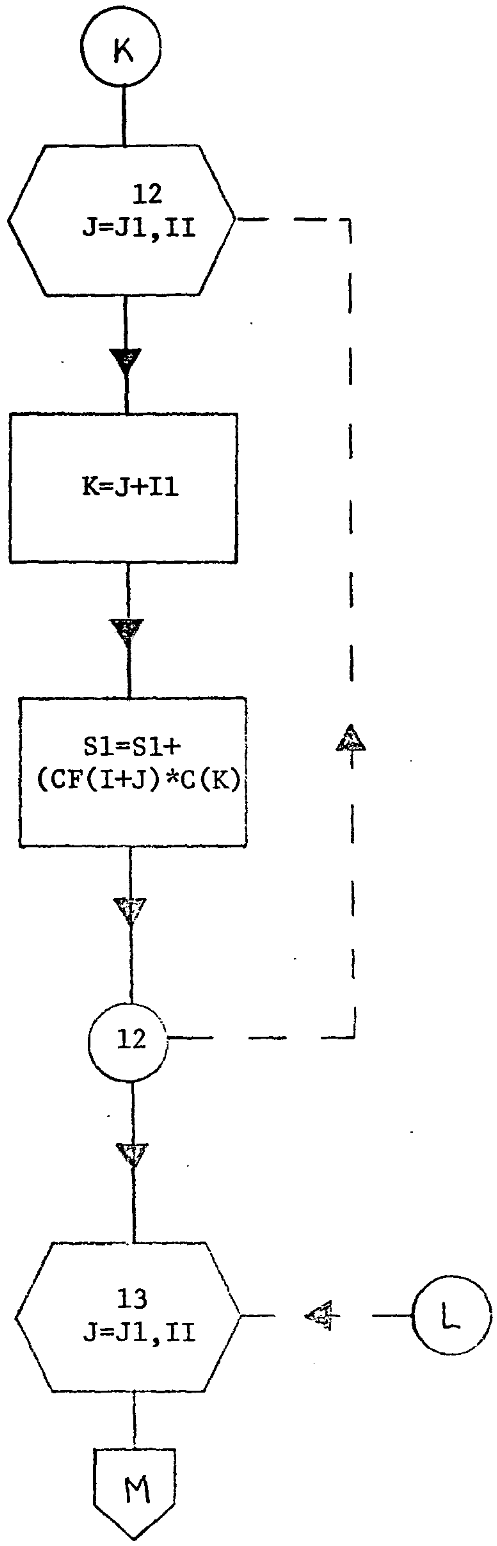


Figure 3.4.1 DETAILED FLOW CHART - CFAC (Continued)

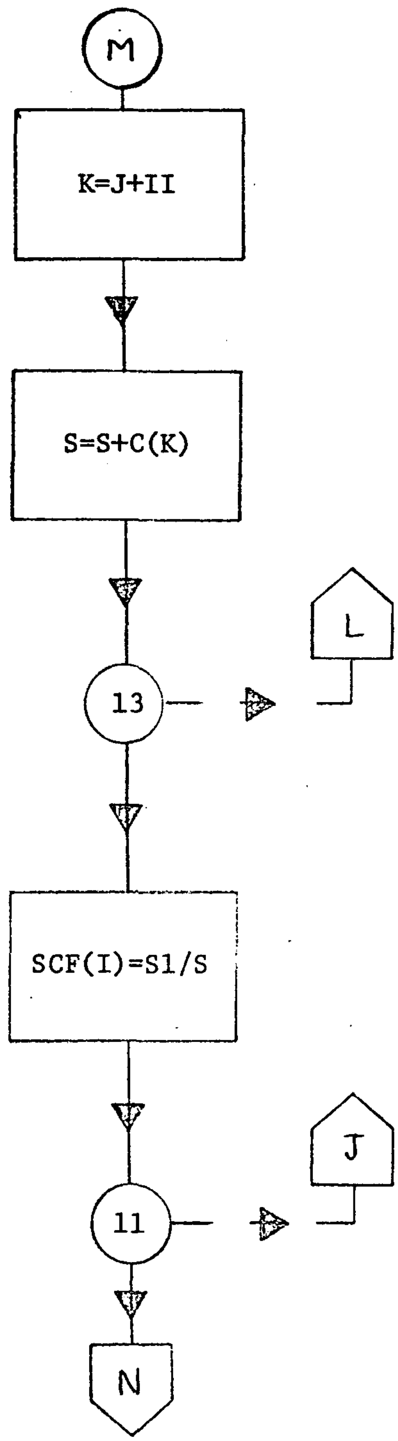


Figure 3.4.1 DETAILED FLOW CHART - CFAC (Continued)

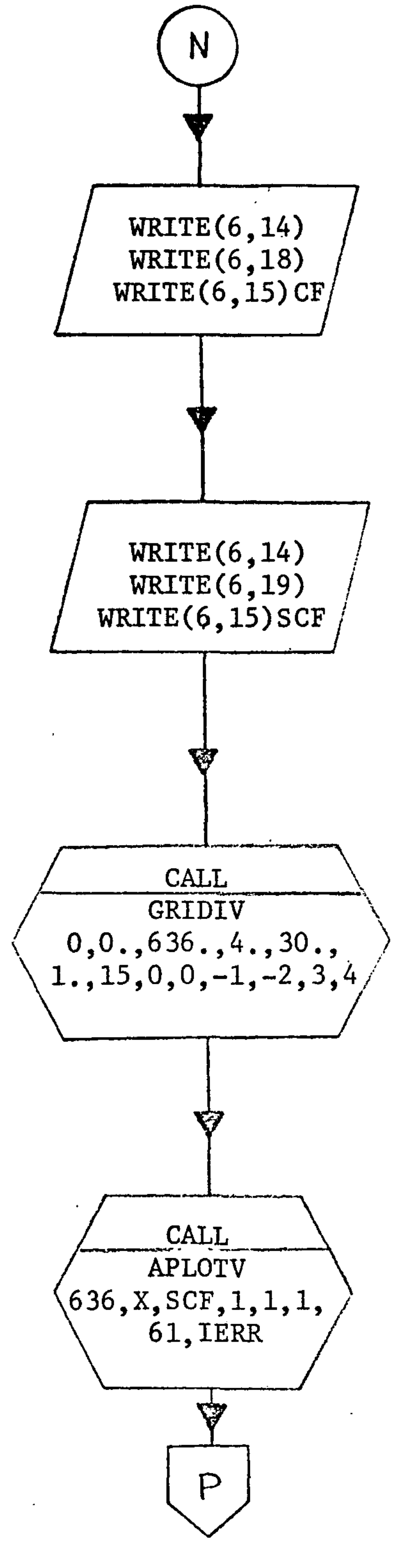


Figure 3.4.1 DETAILED FLOW CHART - CFAC (Continued)

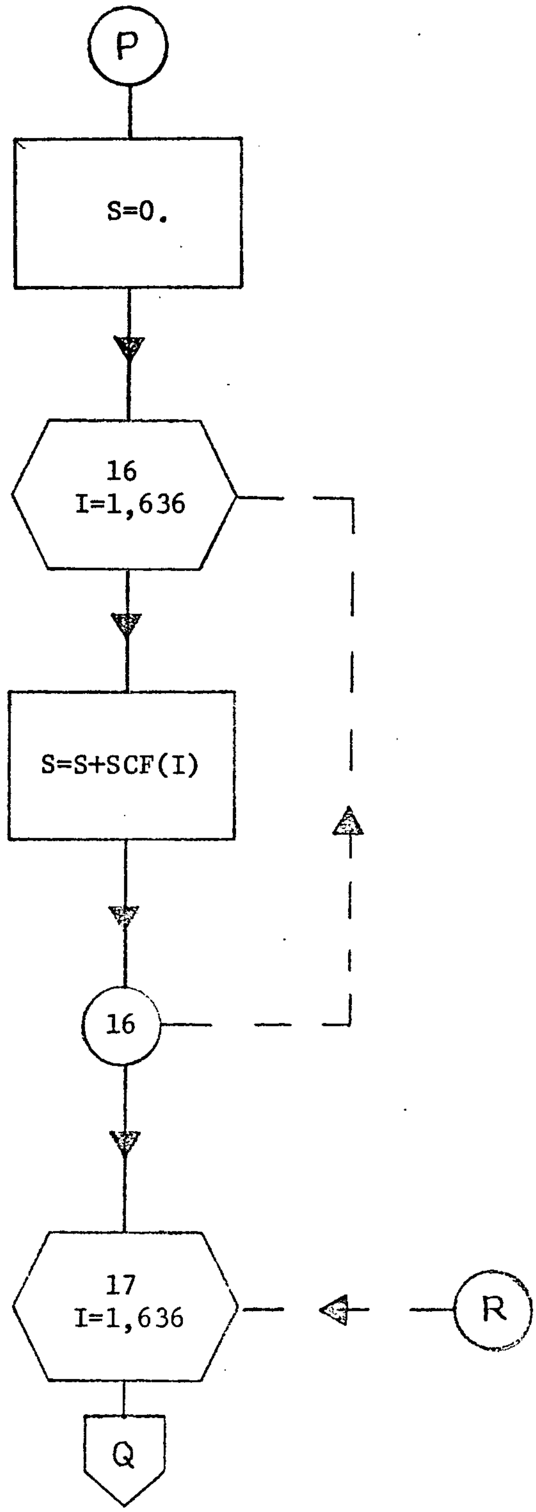


Figure 3.4.1 DETAILED FLOW CHART - CFAC (Continued)

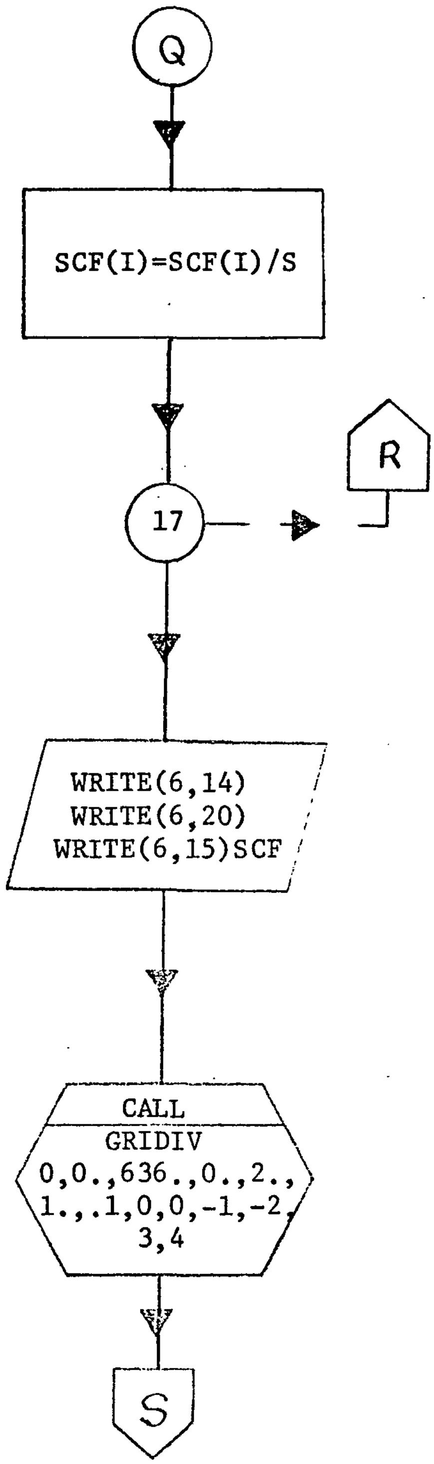


Figure 3.4.1 DETAILED FLOW CHART - CFAC (Continued)

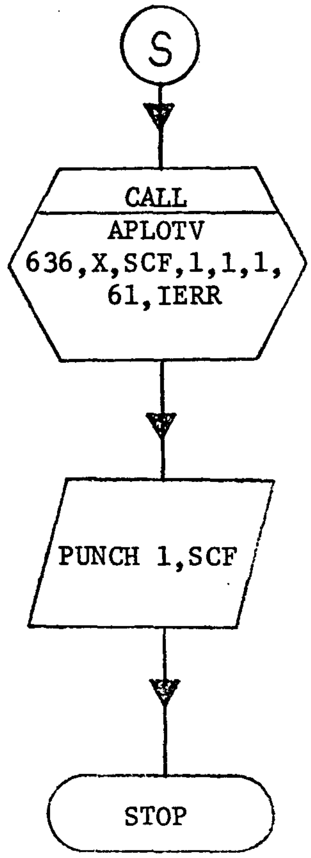


Figure 3.4.1 DETAILED FLOW CHART - CFAC (Continued)

CF	-	Array of dimension 636; the elements of which are unsmoothed correction factors.
SCF	-	Array of dimension 636; the smoothed correction factors.
C	-	Array of dimension 99; an array of cosines.
X	-	Array of dimension 636; used to generate abscissas for APLOTV.
J1,J2	-	Parameters set by the user; variation of these parameters determines the number of cosines used in the evolution CF*C and hence the "smoothness" of SCF.
I1,I2,I3	-	Parameters internally set by the program once J1, J2 are determined. Used in determining the ranges of DO loops.
PT	-	Constant equal to $\pi/(2*(J2+1))$.
S,S1	-	Variables used to obtain the convolution CF*C.
K	-	Index which determines which elements of C are used in the convolution.

Table 3.4.2 DESCRIPTION OF PROGRAM VARIABLES - CFAC

4 OBTAINING GRAY SHADE CALIBRATION

4.1 The current analysis program is used as if it were developed with the philosophy that changes in the D log E curve will be used to account for variations in the mission, digitization and lunar area. That is, instead of redefining the limiting values the analysis procedure is to take for 1 and 63 (the range of six-bit numbers available, zero having a special meaning), the D log E curve (really table) is adjusted. Over a very wide range of exposure values, the D log E curve of SO 243 film is approximately linear. Thus there is a good argument for having one D log E curve (which is made using data furnished by the manufacturer of the film), with differences in the conditions of any particular digitization being taken into account by redefinition of the extreme values 1 and 63 (with resulting automatic linear adjustment between) for use in the photometric function.

Whether this seemingly better program is carried out or the current operational procedure is followed, accurate calibration of the meaning of each value in the data in (say) meter-candle-seconds (MCS) must be obtained. The only calibration available for use at the time the analog tapes are digitized is the edge data.

In the edge data, there is an area approximately in line with the framelet numbers containing a nine level gray scale and resolving power charts. Each digitization includes enough of this edge data to obtain the gray scales. By manual examination of the printout of these areas, one is able to locate the limits of the gray areas. An *ad hoc* program was written to apply correction factors (obtained from the same digitization) and then calculate averages of areas whose coordinates were read in as data. While it would have been theoretically trivial to locate the gray shade areas on the computer and automate the whole procedure (using in effect the same elementary principles of pattern recognition used to locate the Reseau marks as outlined in the report on Project B, Part II), the practical problem of developing the software to accomplish this, coupled with the existence of more interesting and pressing problems extant at the time, precluded the development of working algorithms which would have automated the calculation of gray shade area averages.

4.2 In order to illustrate the program outlined above, we shall take a particular pair of digitizations, not done with any special care, and show how the gray shades are related to both the values in the LO data and to exposure values, visual densities and readout transmissions. (As usual, the data is taken from LO III, HR frame 154, complete Framlets 028, 027, 025 or 2068 lines of Surveyer 3 area.) The data in Table 4.2.1 was collected from various calibration furnished on the LO III mission. On referring to Table 4.2.1, we may conjecture that at most three of the gray scales are likely to be of any use in forming a realistic D log E curve; indeed, the gray scales with numbers 5 or less (corresponding to darker areas on a positive) are useless owing to clipping, while gray scale number 9 is also unlikely to be reliable for a similar reason (noise added in transmission). Between,

STEP NUMBER	READOUT DENSITY	READOUT TRANSMISSION	EXPOSURE MCS	GRAY SCALE FRAMELET	AVERAGES 2068
1	-	-	0.0257	**	**
2	0.301	0.500	0.0398	**	**
3	0.321	0.477	0.0562	**	**
4	0.360	0.437	0.0708	**	**
5	0.446	0.358	0.1122	61.5	**
6	0.633	0.233	0.1778	45.9	45.1
7	0.879	0.132	0.2884	23.3	22.2
8	1.215	0.061	0.4571	8.5	7.6
9	1.347	0.045	0.6761	3.1	1.9

Table 4.2.1 D LOG E DATA; ** = CLIPPED

the relationship between gray scale average and readout transmission should be linear; this is verified by Fig. 4.2.2.

To account for clipping, the correct solution of the new D log E curve (table) problem seems to be to use only the three reliable data points and extrapolate a linear regression of those three data points. This procedure is probably best carried out with the aid of a small programable calculator, using standard well-known programs, and will not be reported on in detail here. The result of this is summarized in Table 4.2.3.

One feature of Table 4.2.3 is the LO numbers in the range 64-80, which in fact do not exist owing to clipping. The computations necessary to obtain the other numbers furnished these additional numbers free of additional cost, and there is some value in having an idea of what exposure values would be clipped (are clipped) by the digitization process. Based on an examination of Table 4.2.3, the D log E curve sketched in Fig. 4.2.4 was developed. For comparison, a D log E curve in use at the time this work was being done is indicated.

Note that the assignment of 0 and 63 (or 1 and 63 if to be used in the analysis program) is essentially arbitrary, and, in effect, can result in further clipping of the data. A further advantage of the method outlined here is that the value (in MCS) of 0 and 63 (here 0.61 and 0.135 respectively) can be determined easily for possible use in photogrammetric studies of the linearized data.

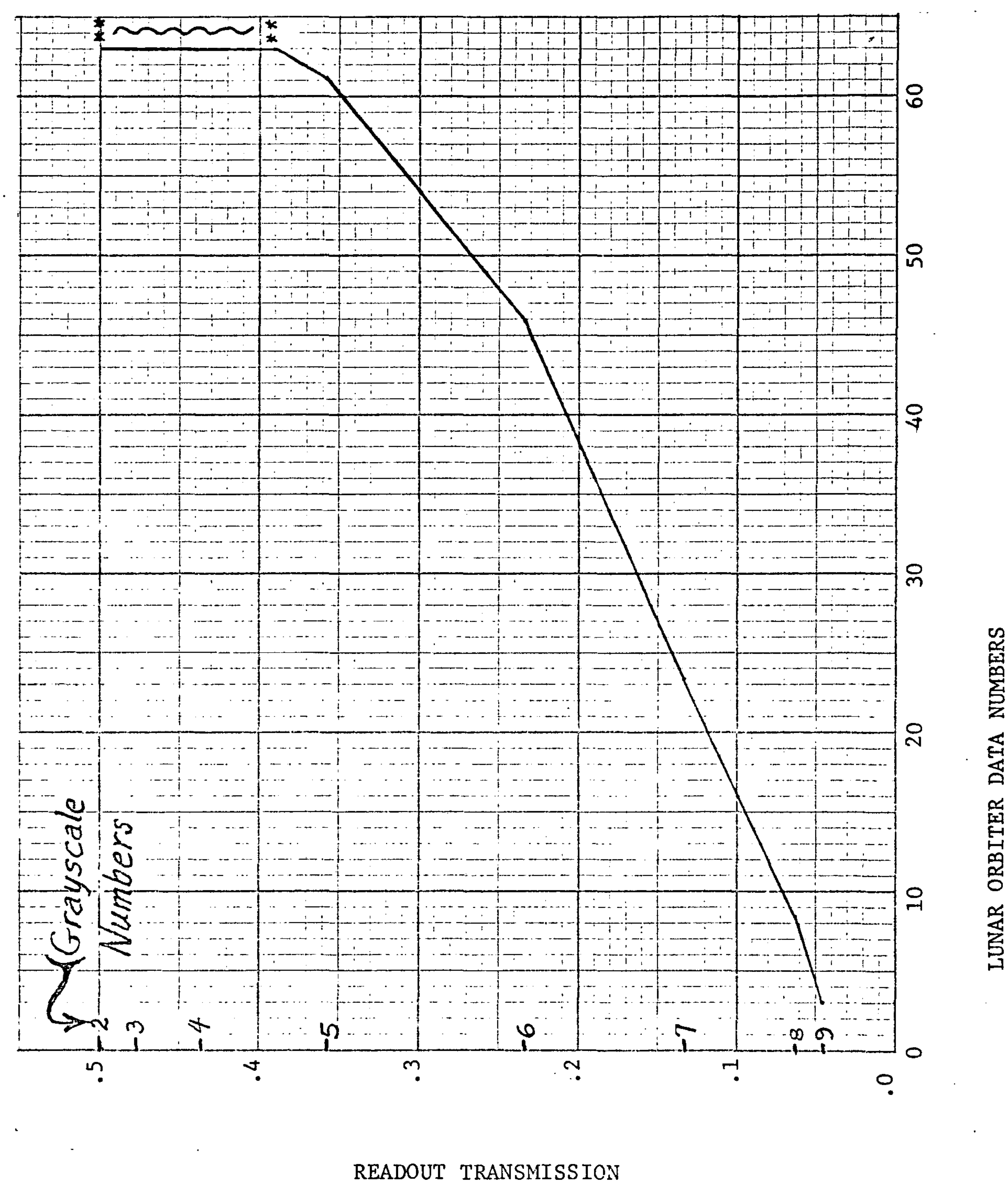


Fig. 4.2.2 Transmission vs. L.O. Numbers.

LO DATA NUMBER	V	LOG E	E	SMOOTHED E
0	0.25	**	0.800	0.800
8	0.61	$\bar{1}.700$	0.501	0.500
16	0.97	$\bar{1}.512$	0.325	0.330
24	1.32	$\bar{1}.422$	0.264	0.264
32	1.68	$\bar{1}.348$	0.223	0.233
40	2.03	$\bar{1}.281$	0.191	0.192
48	2.39	$\bar{1}.221$	0.166	0.167
56	2.74	$\bar{1}.164$	0.146	0.145
64	3.10	$\bar{1}.108$	0.128	0.128
72	3.45	$\bar{1}.042$	0.110	0.112
80	3.81	$\bar{2}.972$	0.094	0.100

Table 4.2.3 V - LOG E CURVE INTERPOLATION

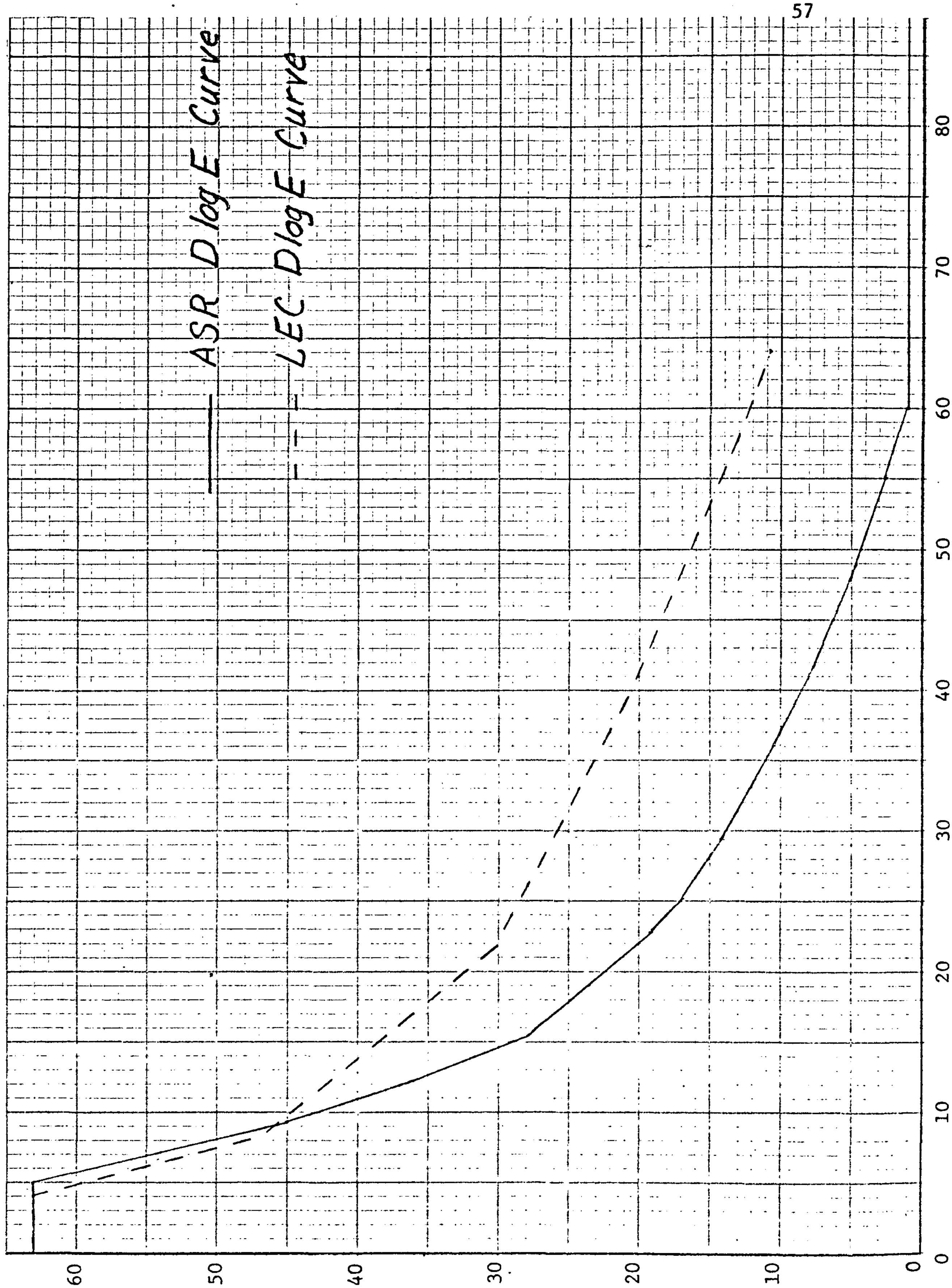


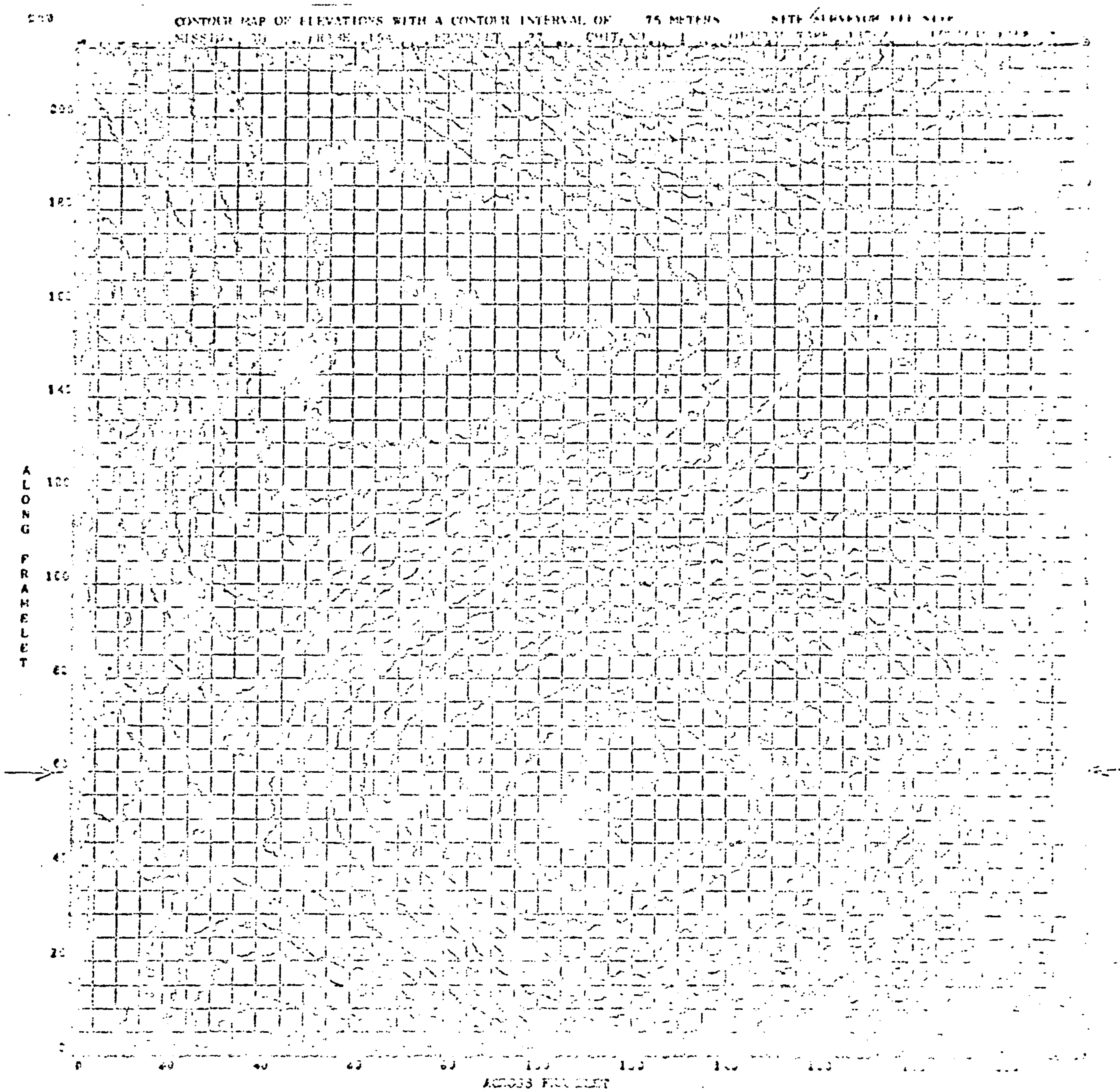
Fig. 4.2.4 Comparison of DlogE Curves.

4.3 In addition to making pictures which look "better", our D log E curve has been compared with the one used in the analysis program furnished, and contour maps (which have been reported on in the monthly reports) indicate a much more accurate representation of the Surveyor III area. For reference, these reports are summarized at this time:

For the comparison of D log E curves, see Fig. 4.2.4; contour maps are shown in Figs. 4.3.1, 4.3.2, 4.3.3 and 4.3.4. Fig 4.3.1 shows a contour map made using the LEC correction factors and D log E curve. Figs. 4.3.2, 4.3.3 and 4.3.4 show contour maps resulting from the change in correction factors, D log E curve, and both, respectively. Fig. 4.3.4 shows our best contour map without filtering. In Fig, 4.3.5, we show elevation profiles thru the contour maps; here

a is from Fig. 4.3.1
b is from Fig. 4.3.3 (new correction factors)
c is from Fig. 4.3.4 (new correction factors and new D log E)

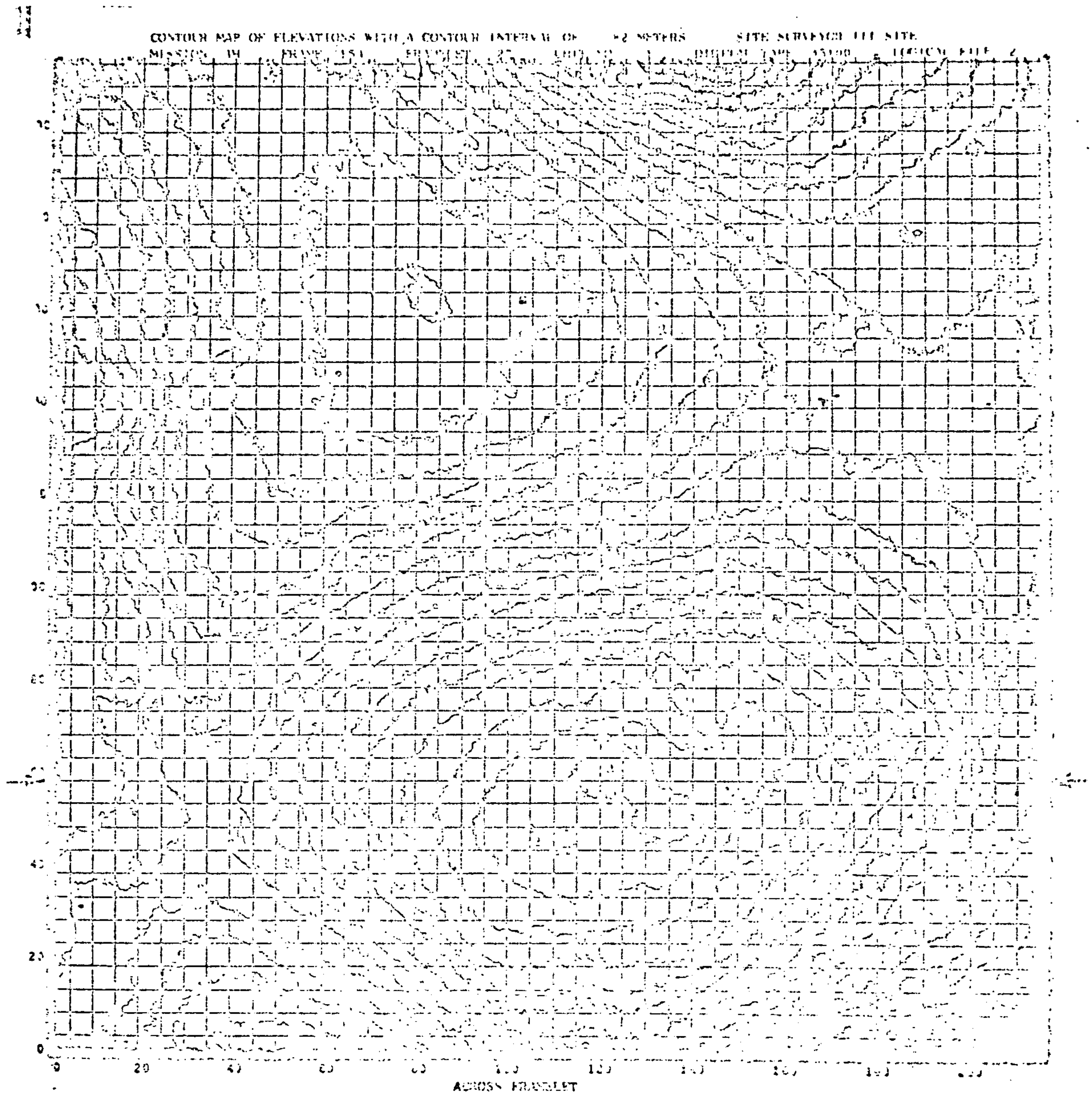
In our discussion of the Project C (Part III of this report), we return to further study these results.



CONTOUR IDENT.

1	-7.26
2	-6.50
3	-5.75
4	-5.00
5	-4.24
6	-3.49
7	-2.73
8	-1.98
9	-1.22
A	-.47
B	.28
C	1.03
D	1.78
E	2.54
F	3.29
G	4.05
H	4.80
I	5.56
J	6.31
K	7.06

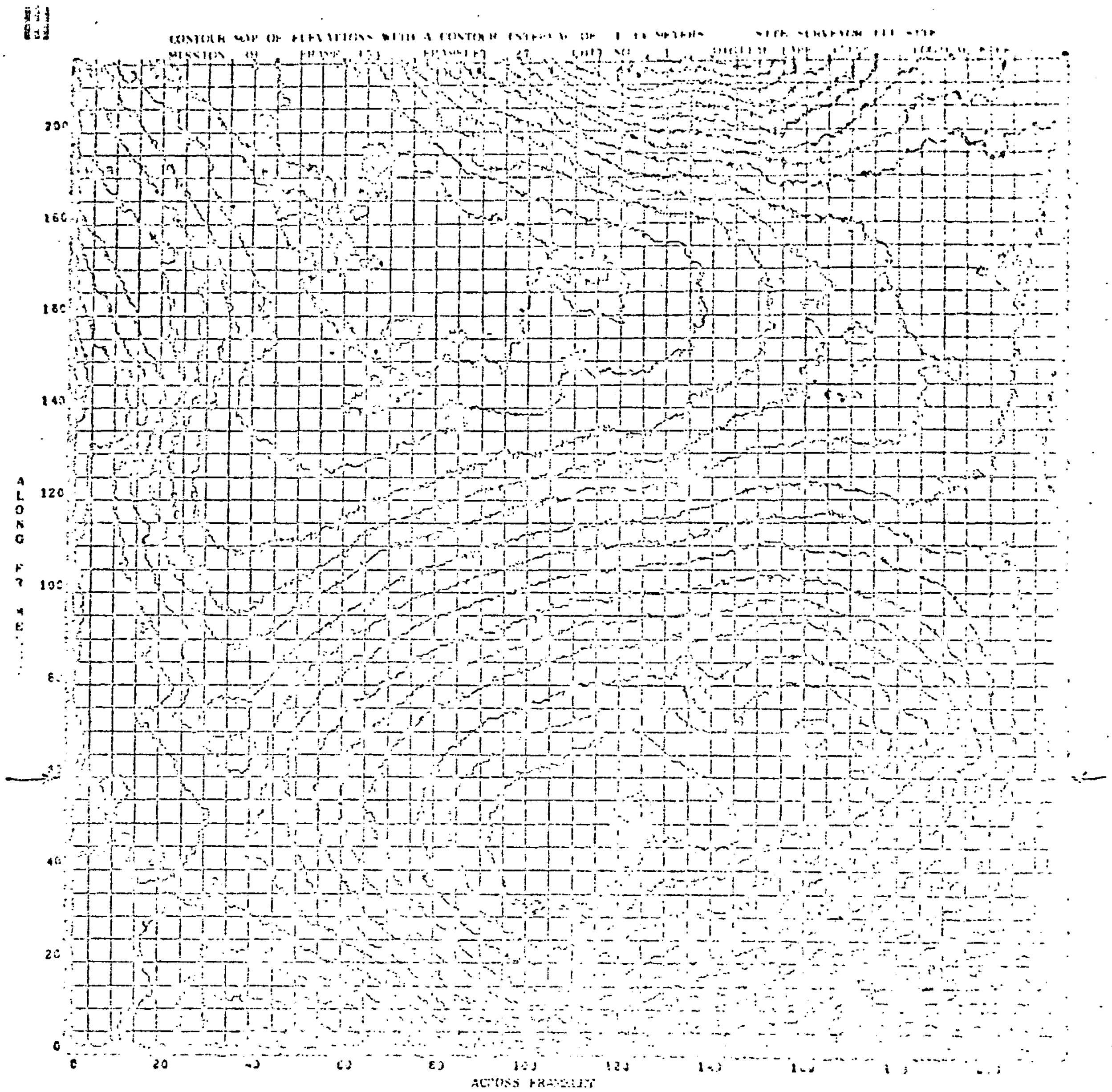
Fig. 4.3.1. LEC Contour Map.



CONTOUR IDENT.

1	-7.71
2	-6.90
3	-6.08
4	-5.26
5	-4.44
6	-3.63
7	-2.81
8	-1.99
9	-1.19
A	-.36
B	.45
C	1.27
D	2.08
E	2.90
F	3.72
G	4.53
H	5.35
I	6.17
J	6.98
K	7.80

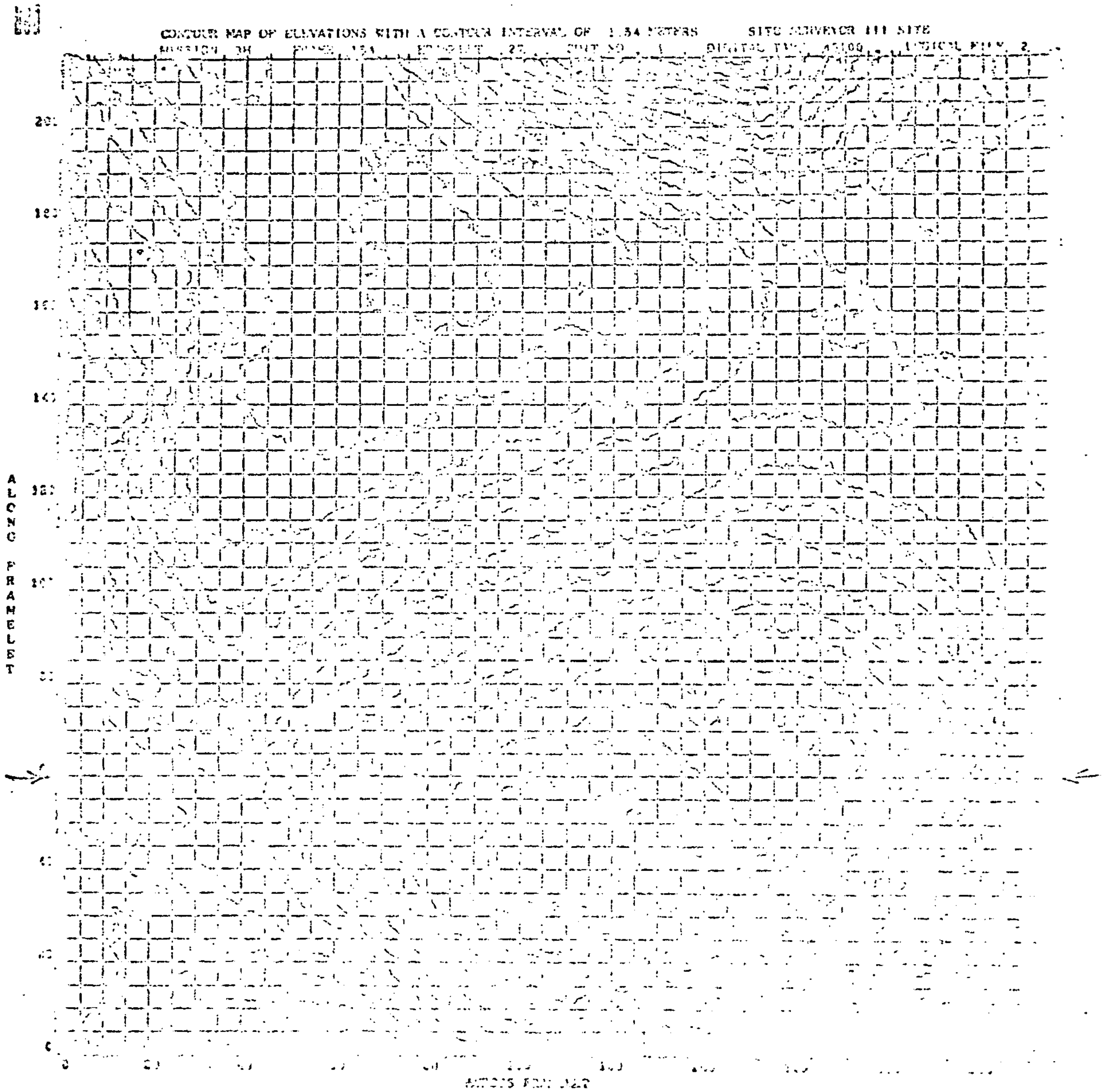
Fig. 4.3.2. New Correction Factors.



CONTOUR IDENT.

1	-11.71
2	-10.36
3	-9.02
4	-7.68
5	-6.34
6	-4.99
7	-3.65
8	-2.31
9	-.97
A	.37
B	1.71
C	3.05
D	4.39
E	5.74
F	7.08
G	8.42
H	9.76
I	11.11
J	12.45
K	13.79

Fig. 4.3.3. New DlogE.



CONTOUR IDENT.

1	-13.75
2	-12.21
3	-10.67
4	-9.13
5	-7.59
6	-6.04
7	-4.[illegible]
8	-2.9[illegible]
9	-1.42
A	.11
B	1.6[illegible]
C	3.2[illegible]
D	4.74
E	6.2[illegible]
F	7.82
G	9.36
H	10.90
I	12.45
J	13.9[illegible]
K	15.53

Fig. 4.3.4. New Correction Factors and DlogE.

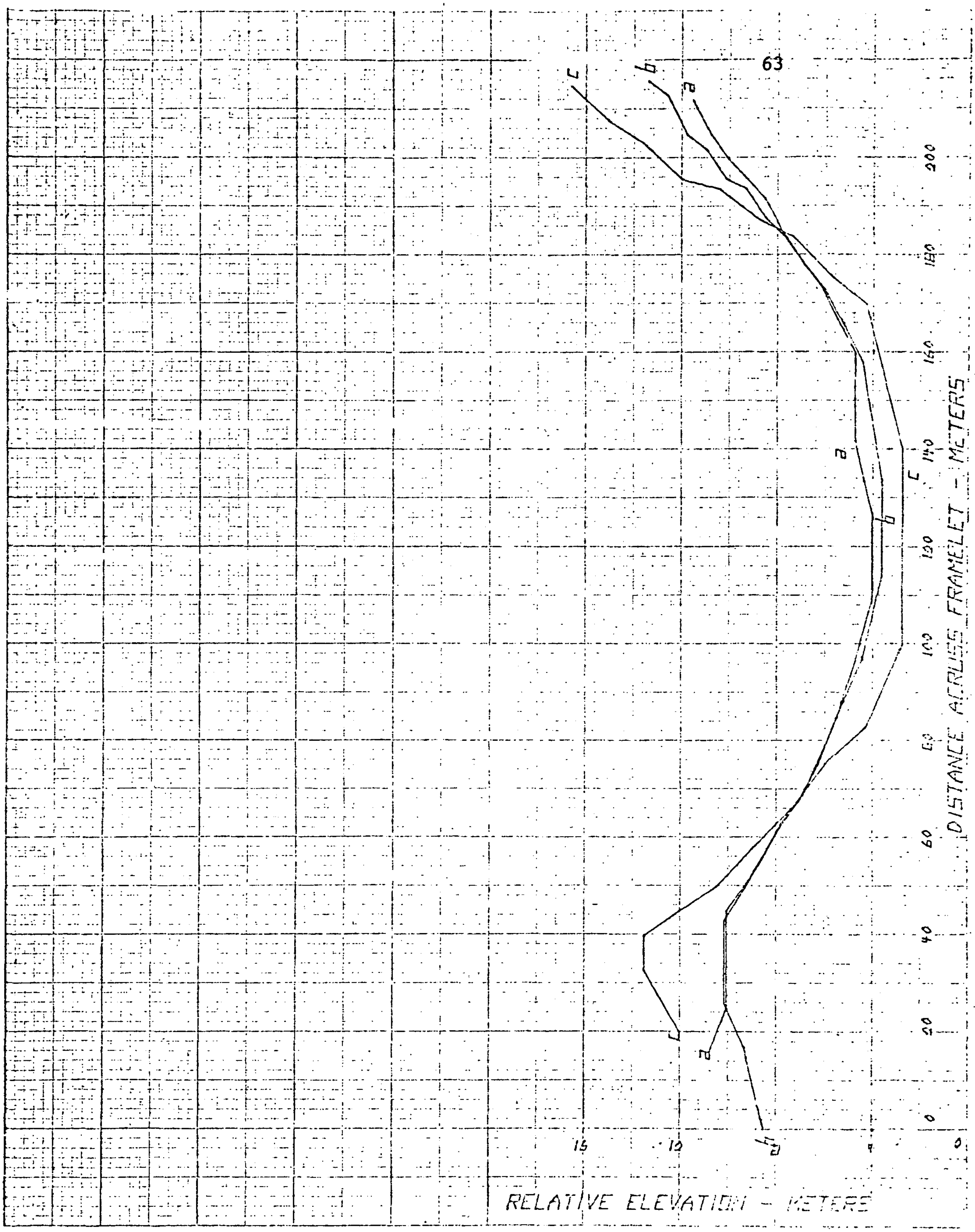


Fig. 4.3.5. Elevation Profiles Through "60".

a LEC Contour Map.
b New Correction Factors.
c New Correction Factors and D log E.

PART II ENHANCEMENT OF DATA

INTRODUCTION The LO digitized data suffers from six different degradations. They are: (i) geometric distortion, (ii) spatial frequency dependent gain variations, (iii) spatial frequency independent gain variations, (iv) analog noise (added in transmission), (v) digitization (jitter, quanitization and clipping), and (vi) film granularity noise. Most of these vary with position in the scene being reproduced. Just how many of the factors need be considered depends on the use of the data. For example, in Project A (joining framelets) we found it necessary to consider problems introduced by degradation (i) (in the readout device on LO), (iii) (particularly the line scan signature) and (v) (jitter, and, to fix (iii), clipping). In order to use the data for Apollo landing site selection, it has been possible to consider only (i) and (iii); even with imperfect corrections, passable contour maps are possible.

Part II is particularly concerned with (ii), (iv), (v) and (vi). As a review of the problems and a sketch of some of the mathematical techniques we used in this part of Project B, we include first part of the Master of Science Thesis "Position Invariant Linear Operations in Image Processing" by Terrence Lee Dillon (Texas A&M, May, 1972; directed by Jack Bryant). This thesis is mainly expository, and includes many examples in one dimension of the kind of analysis we are attempting in two dimensions. (The numbering in this section departs from that adopted elsewhere in the report.)

1. POSITION INVARIANT LINEAR OPERATIONS IN IMAGE PROCESSING

INTRODUCTION TO PICTURE FUNCTIONS

Any photograph can be considered as a collection of data points whose value changes from point to point. If we confine our attention to black and white photographs and define the collection of image points as a two-dimensional matrix, we can refer to the photograph as a picture function, f, whose value at any point, f(x, y), is the gray level of the picture at that point. Since a physical picture is of finite size, we can suppose that the picture function is non-zero only in a bounded region and zero elsewhere. When we process pictures via digital computers, we usually want to regard them as discrete arrays of numbers, i.e. matrices. Such a picture function will be called a digital picture function. [1]

One problem of picture processing is that of filtering, i.e. separating the picture as cleanly as possible from the contaminated combination of the picture plus degradation (noise). Let K be the filtering system which performs some mathematical operation, Ø, on the digital picture function f(x, y),

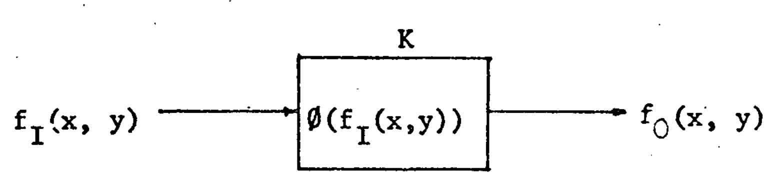


Figure 1 IMAGE FILTER SYSTEM

where $f_I(x, y)$ is degraded input, and $f_0(x, y)$ is filtered output. The first problem is to find the operator $\emptyset$ of the filter system. [2]

POSITION INVARIANT LINEAR OPERATIONS

Many physically meaningful operations performed on pictures have the following property: if the input picture is moved, then the output picture is also moved by the same vector. (Such operations are analogous to time invariant operators of electrical network theory.) In this section, we outline some of the characteristics of operators which exhibit this and other properties and give some examples.

Position Invariant Operations.

Let $\mathcal{L}$ be the set of picture functions of two real variables defined on the entire plane, P. Let $\emptyset: \mathcal{L} \to \mathcal{L}$ where $\emptyset$ describes an operation on $\mathcal{L}$. Let T_{uv} be the translation of f(x, y) by (u, v) defined by,

$$[T_{uv}(f)](x, y) = f(x - u, y - v)$$

$$\text{for all } f \in \mathcal{L} \text{ and } (x, y) \in P . \qquad (1\text{-}1)$$

The operation $\emptyset$ is said to be a position invariant operation on $\mathcal{L}$ if $\emptyset$ commutes with every T_{uv}, [3]

$$\emptyset[T_{uv}(f)] = T_{uv}[\emptyset(f)] . \qquad (1\text{-}2)$$

It is easy to verify that the translation operation itself is position invariant.

In general geometrical correction operations other than translations are not position invariant. For example, let

$\emptyset = S_{\alpha\beta}$ be defined as an operation which changes the scale of f via.

$$[S_{\alpha\beta}(f)](x, y) = f(\alpha(x, y), \beta(x, y)) \cdot \tag{1-3}$$

We have

$$S_{\alpha\beta}[T_{uv}(f)](x, y) = [T_{uv}(f)](\alpha(x, y), \beta(x, y))$$

$$= f(\alpha(x, y) -u, \beta(x, y) -v) , \tag{1-4}$$

whereas,

$$T_{uv}[S_{\alpha\beta}(f)](x, y) = [S_{\alpha\beta}(f)](x - u, y - v)$$

$$= f(\alpha(x - u, y - v), \beta(x - u, y - v)) \tag{1-5}$$

and clearly $S_{\alpha\beta}[T_{uv}(f)] \neq T_{uv}[S_{\alpha\beta}(f)]$ for all f (unless

$\alpha(x) = x$ and $\beta(y) = y$), (1-6)

which indicates that the scaling operation is not position invariant.

Point Operations and Local Operations.

An operation is called a point operation if the gray level of the output picture at a point depends only on the gray level of the input picture at that point. Point operations involve modifying the gray level of each point to improve the message content of the picture. Important examples include [4]:

1.) Attenuation. $\emptyset(f(x, y)) = kf(x, y)$ where k is a real number. This is pointwise multiplication of the gray level by a constant to uniformly "brighten" or "dim" a picture.

2.) Intensification. $\emptyset(f(x, y)) = (f(x, y))^2$. Pointwise squaring to suppress some features and enhance others.

3.) Clipping. $\emptyset(f(x, y)) = G_o$ if $f(x, y) > G_o$, $\emptyset(f(x, y)) = f(x, y)$ if $f(x, y) \leq G_o$. Pointwise removal of some portion of the picture which contains useless or undesirable information.

An operation is called a local operation if the gray level at a point in the output picture is the result of averaging the gray level over some neighborhood of that point in the input picture. A local operation is a smoothing process generally used to suppress random noise. Some examples are [5]:

1.) Simple Averaging. Replace the value of each point by the average of the values over a neighborhood of the point.

2.) Thresholding. Replace the value of each point by the average in some neighborhood if the average exceeds the point gray level by more than some threshold.

Linear Operations.

An operation is called linear if the principle of superposition holds - i.e.

$$\emptyset(af + bg) = a\emptyset(f) + b\emptyset(g) . \qquad (1-7)$$

Clearly translation operators are linear. As we shall see, linear operations which commute with the translation operators are especially easy to analyze. [6]

Physical Processing in the Lens-Film-Digital System.

Since we are concerned with digital pictures, let us define our image sensing system as being a lens-film-digital system. The lens-film system is, very simply, the interaction of light and photosensitive paper; however, the digital aspect requires some explanation. To form the digital picture function, a flying spot scanner or densitometer is used to assign values to each matrix element according to the gray level of that element. This process is called quantizing or digitizing.

Basically the flying spot scans the photograph line by line and illuminates parts of it at regular intervals during successive sweeps across the viewing area. When the spot illuminates a white (or transparent) area a bright light is reflected into a photomultiplier which transduces it into a high voltage signal. Dark areas produce low voltage signals. The pattern of high and low voltages then correspond to the shading of the picture, i.e. black, white, or gray. These voltages are represented as integer values to form the digitized (quantized) picture, and the range of integer values represents the grey level scale. [7]

Point Spread Functions.

Because the flying spot does not produce an exact point source but rather a spot perhaps 5/10,000 of an inch in size, the light from the scanner is spread over a small but finite area of the picture. This spread of illumination is called the point spread function, S, of the flying spot scanner. In fact, each element of our lens-film-digital system has a point spread function. In film the image point is spread by scattering, reflections, and the granularity of the film itself.

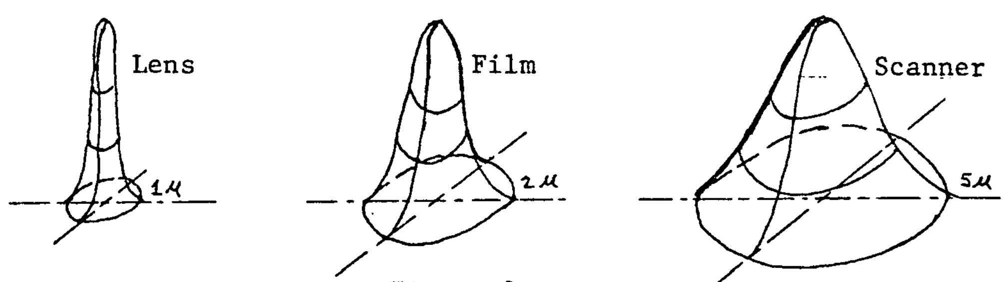


Figure 2
TYPICAL SPREAD FUNCTIONS (Adapted from [8])

Suppose, for the moment, that it is possible to obtain all the individual point spread functions which characterize our lens-film-digital system. It is known from optics that image illumination is obtained from the object luminance by a two-dimensional convolution of the object with the point spread function. [9] Hence, at any point in our system, there exists a transient image which is the convolution of the object and the initial point spread function, or an intermediate image (which represents a previous convolution) convolved with another point spread function. In particular, we have a linear system such that

$$f_1(x, y) = f * S_1(x, y) = \iint_{-\infty}^{\infty} f\,(x^1,y^1)S_1(x - x^1,y - y^1)dx^1dy^1 \quad (1\text{-}8)$$

$$f_2(x, y) = f_1(x, y) * S_2(x, y) \quad (1\text{-}9)$$

$$f_o(x, y) = f_{n-1}(x, y) * S_n \quad (1\text{-}10)$$

for $i = 1, \ldots, n$ elements in the system,

where f is the original (ideal) object image,

$f_i(x, y)$ is the transient image,

$S_i(x, y)$ is an intermediate point spread function,

$f_o(x, y)$ is the output image, and

$*$ denotes the convolution operation.

If we let $\emptyset$ be the convolution defined as,

$$\emptyset[f(x, y)] = \iint_{-\infty}^{\infty} f(x^1,y^1)S(x - x^1,y - y^1)dx^1dy^1 \quad (1\text{-}11)$$

then,

$$\emptyset[T_{hg}f](x, y) = \iint_{-\infty}^{\infty} f(x^1,y^1)S(x - x^1 - h, y - y^1 - g)dx^1dy^1 \qquad (1\text{-}12)$$

$$= \iint_{-\infty}^{\infty} f(x^1,y^1)S((x - h) -x^1,(y - g)-y^1)dx^1dy^1 \qquad (1\text{-}13)$$

$$\emptyset[T_{\substack{h\\g}} f](x, y) = \emptyset f(x - h,y - g) = T_{\substack{h\\g}} [\emptyset f(x, y)] . \qquad (1\text{-}14)$$

Hence, the convolution operation (*) is position invariant and clearly linear. In view of this physical motivation, convolution becomes the position invariant linear operation necessary to process picture functions. Fortunately, convolutions are easy to analyze using Fourier transform theory.

PRINCIPLES OF IMAGE PROCESSING

The realization of convolution leads us to Fourier transforms via the convolution theorem which states that the convolution of two functions is equal to the inverse Fourier transform of the product of their Fourier transforms. [10]
(For a proof see Appendix A)
That is,

$$f^1(x,y) = f*S(x,y) = F^{-1}[F[f(x,y)]F[S(x,y)]] \qquad (2\text{-}1)$$

where the two-dimensional Fourier Transform of $f(x,y)$ is defined as,

$$F(u,v) = F[f(x,y)] = \int_{-\infty}^{\infty}\int f(x,y)e^{-2\pi i(xu+yv)}dxdy\,; \qquad (2\text{-}2)$$

and by the inverse Fourier transform of $F(u, v)$ is meant the reflection through (0, 0) of its transform,

$$f(x,y) = F^{-1}[F[f(x,y)]] = \int_{-\infty}^{\infty}\int F(u,v)e^{i2\pi(xu+yv)}dudv\,. \qquad (2\text{-}3)$$

In its simplest form, linear image enhancement can be considered as an attempt to solve the convolution integral for the original image. Suppose linear degradations, i.e. those in which the point spread function characterizes the image degradation and is invariant over the region of the image. The degraded image can then be described by the convolution integral,

$$f_I(x,y) = f{*}S_I(x,y) = \int_{-\infty}^{\infty}\!\!\int f(x-x^1,\ y-y^1)S_I(x^1,y^1)dx^1dy^1 \qquad (2\text{-}4)$$

where $f(x,y)$ is the ideal image without degradation,

$S_I(x,y)$ is the input spread function of the digital system,

$f_I(x,y)$ is the input image (a result of previous convolutions in the lens-film system),

with the implication that the resolution of the input image is characterized by the input spread function. [11]

Application of the convolution theorem allows partial solution of the convolution integral of equation (2-4) by using Fourier transforms. Hence,

$$F[f_I(x,y)] = F[f{*}S_I(x,y)] = F[f(x,y)]F[S_I(x,y)]. \qquad (2\text{-}5)$$

Solving for the ideal image,

$$F[f(x,y)] = F[f_I(x,y)]/F[S_I(x,y)] \qquad (2\text{-}6)$$

and taking the inverse transform gives

$$f(x,y) = F^{-1}\{F[f_I(x,y)]/F[S_I(x,y)]\}. \qquad (2\text{-}7)$$

This equation describes a process by which the ideal can be determined from the degraded image and knowledge of the point spread function associated with the degradation. The Fourier transform of the input point spread function is by definition the input transfer function. It is immediately apparent that the input transfer function may have regions of zero value; in which case, equation (2-7) becomes meaningless because of an indeterminate ratio. To ease this problem, image restoration requires a slightly different formulation. This is obtained by multiplying both sides of equation (2-6) by the term $F[S_o(x, y)]$ which gives,

$$F[f(x, y)]F[S_o(x, y)] = F[f_I(x, y)]F[S_o(x, y)]/F[S_I(x, y)]. \tag{2-8}$$

The left-hand side of equation (2-8) is the Fourier domain representation of a convolution integral of the form

$$f_o(x, y) = \int_{-\infty}^{\infty}\int f(x-x^1, y-y^1)S_o(x^1, y^1)dx^1dy^1 = F^{-1}[F[f(x,y)]F[S_o(x,y)]. \tag{2-9}$$

The term $f_o(x, y)$ is defined as the output image; $S_o(x, y)$ is the output point spread function. Use of only the output transfer function, $F[S_o(x, y)]$, has the same disadvantages described for the input transfer function. By defining

$$F[S_p(x, y)] = F[S_o(x, y)]/F[S_I(x, y)] \tag{2-10}$$

where $S_p(x, y)$ is the processing point spread function, the right side of equation (2-8) becomes the Fourier domain statement of the convolution integral,

$$f_o(x, y) = \int\int_{-\infty}^{\infty} f_I(x-x^1, y-y^1) S_p(x^1, y^1) dx^1 dy^1 = f_I(x,y) * S_p(x,y), \tag{2-11}$$

and

$$f_o(x,y) = F^{-1}[F[f_I(x,y)]F[S_p(x,y)]] \tag{2-12}$$

where $F[S_p(x,y)]$ is defined as the cummulative digital transfer function which characterizes degradation due to processing. Equation (2-12) is preferable to equations (2-6) or (2-9) because it indicates that, in the presence of noise, it is not possible to obtain a perfect restoration. However, proper selection of $S_p(x,y)$ avoids the problems of a zero-valued transfer function and can serve to suppress the noise level in the output image to a tolerable level. [12-16]

Hence, we have a linear filter system, the heart of which is a two-dimensional Fourier transforming capability. A two-dimensional spatial filter, $S_p(x,y)$, can be formed and the filter operation defined by a convolution in the spatial domain or a multiplication in the Fourier domain. If $f_o(x,y)$ is the output of the filter with $f_I(x,y)$ as input, then [17]

$$f_o(x,y) = \int_{-\infty}^{\infty}\!\!\int f_I(x,y)\,S_p(x-u,y-v)\,dudv = f_I{*}S_p(x,y) \qquad (2\text{-}13)$$

where u, v are spatial frequencies in lines per millimeter. (See Appendix B for an explanation of spatial frequency.)

The problem is now to choose the filter in such a way as to reproduce $f_o(x,y)$ as a good approximation or replica of $f(x,y)$, the ideal image.

FILTERING DEGRADED IMAGES

In general there are two types of noise: random noise and system noise. Random noise creates random points of bad data which appear as "snow" or "salt and pepper" in the degraded image. To reduce this noise, the basic concept is to replace each value by a weighted average of the neighboring points. Systematic noise, however, is usually clustered very heavily around a single frequency. A filter peaked near this frequency is all that is needed to clean out the noise. Subjective judgement and numerous trials are usually made to determine the optimum filter for each type noise. [18]

<u>System End-to-End Analysis</u>.

Since no element of the lens-film-digital system can attain absolute fidelity of output to input signal, a progressive degradation of the signal occurs as it passes through the system. The elements in the data chain may alter or distort the signal and they may introduce spurious values, i.e. noise. Because the system operations are sequential, the signal changes and noise contributions are usually additive.

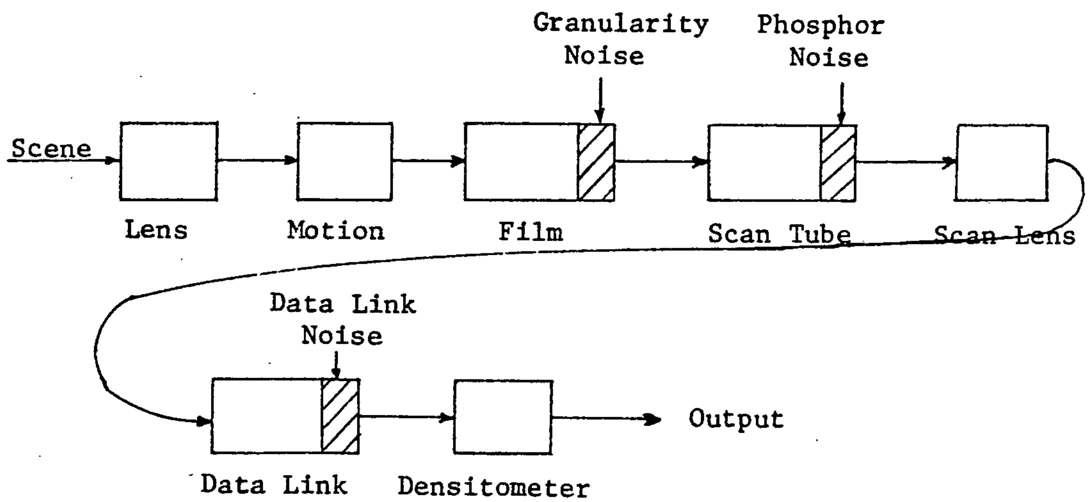


Figure 3

LENS-FILM-DIGITAL SYSTEM SIGNAL AND NOISE BLOCK DIAGRAM (Adapted from [19])

Random Noise and Scan Line Filtering.

Random noise is considered as additive noise, i.e. noise which is added during transmission but cannot be attributed to any specific process in the lens-film-digital system. In this section we consider position invariant operations (not necessarily linear) which can be used to smooth a picture in order to suppress additive random noise.

One can smooth a picture by simply replacing the value at each point by the average of the values over a neighborhood of the point; however, better results can be obtained by taking a weighted average over the neighborhood. The points of the neighborhood are given weights that decrease as their distances from the center increase. [20-21] This concept is best described by an illustrative example.

Suppose we have a framelet of a lunar crater with a one-dimensional scan extracted and quantized.

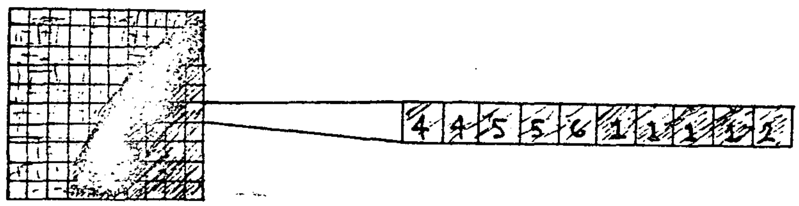


Figure 4

QUANTIZED IMAGE AND ONE DIMENSIONAL SCAN

Let f_I of equation (2-13) be the original quantized one-dimensional image plus additive noise,

$$f_I(x_o) = f(x) + n(x)$$

$$= [4 \vdots 4\ 5\ 5\ 6\ 1\ 1\ 1\ 1 \vdots 2] + [0 \vdots -1\ 0\ -1\ 0\ 0\ 0\ 1\ 0 \vdots 0\ ]$$

$$= [4 \vdots 3\ 5\ 4\ 6\ 1\ 1\ 2\ 1 \vdots 2]$$

let $S_p(x)$ be a three-element weighted averaging filter,

$$S_p(x) = \begin{array}{|ccc} -1 & 0 & 1 \\ \hline \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{array} \;** \;.$$

**Note that the filter is normalized, $\frac{1}{4}$ [1 2 1]. This preserves the gray level range of the picture function. Adapted from [22].

Perform the convolution of the image f_I and the filter S_p to give the recovered output.

$$f_o(x) = f_I * S_p(x) = \sum_{x=-1}^{1} f_I(x_o + x) S_p(x) \quad \text{for} \quad x_o = 2 \text{ to } 9$$

$$= [4 \,\vdots\, 4\ 4\ 5\ 4\ 2\ 1\ 1\ 1 \,\vdots\, 2]$$

Note that the filter is effective only within the dashed lines of the image matrix, i.e. noise which occurs near the edge will not be filtered

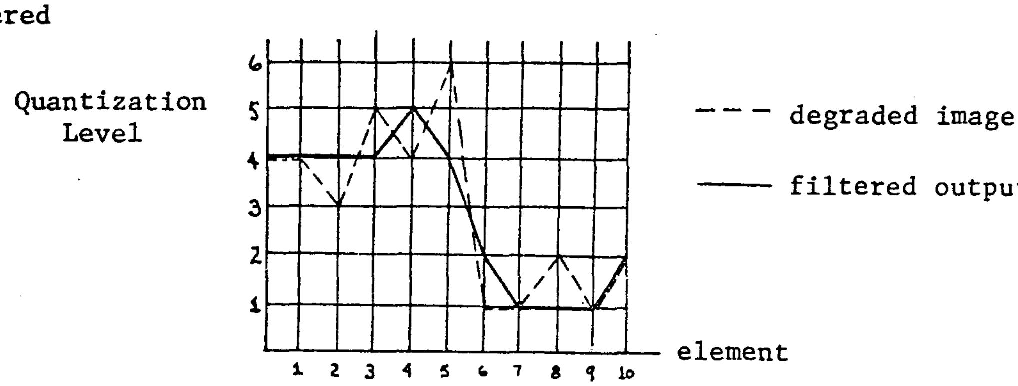


Figure 5

SIMULATED SCANNER READOUT FOR A SINGLE LINE

AND SMOOTHING BY WEIGHTED AVERAGE FILTER

Noise removal by smoothing has several drawbacks. If a picture is smoothed too extensively blurring (loss of resolution or loss of proper gray level intensities) can occur. This can be partially overcome by combining a threshold technique with averaging. For example, let the neighbors of an element E be denoted by,

A	B	C
D	E	F
G	H	I

If $\left|E - (A + B + C + D + F + G + H + I)/8\right| > \delta$, a threshold value, replace E by $(A + B + C + D + F + G + H + I)/8$; otherwise, leave E unchanged. [23] (For an excellent review and examples of this technique see [24].)

Additional difficulties arise from features which are very sharp in contrast. When the filter comes into such an area, it resonates and gives rise to false echoes of the feature. This is known as "ringing" and is illustrated in Figure 6.

Experimental Results on One-Dimensional Images.

Various linear and weighted average filters were examined on a 100 element one-dimensional image. This filtering was done by convolving a noisy image of the form $f_I(x) = f(x) + n(x)$ with filters of varying size, e.g. 13 to 47 elements. In general simple (unweighted) averaging filters are the least effective for restoring the original image and create blurring especially in areas of rapidly changing contrast. If the noise is truly random, Gaussian filters (weighted average filters using Gaussian distribution) give generally pleasing results and blurring is not as severe as with the simple filter. The best overall filter seems to be the $\frac{\sin \pi k}{\pi k}$

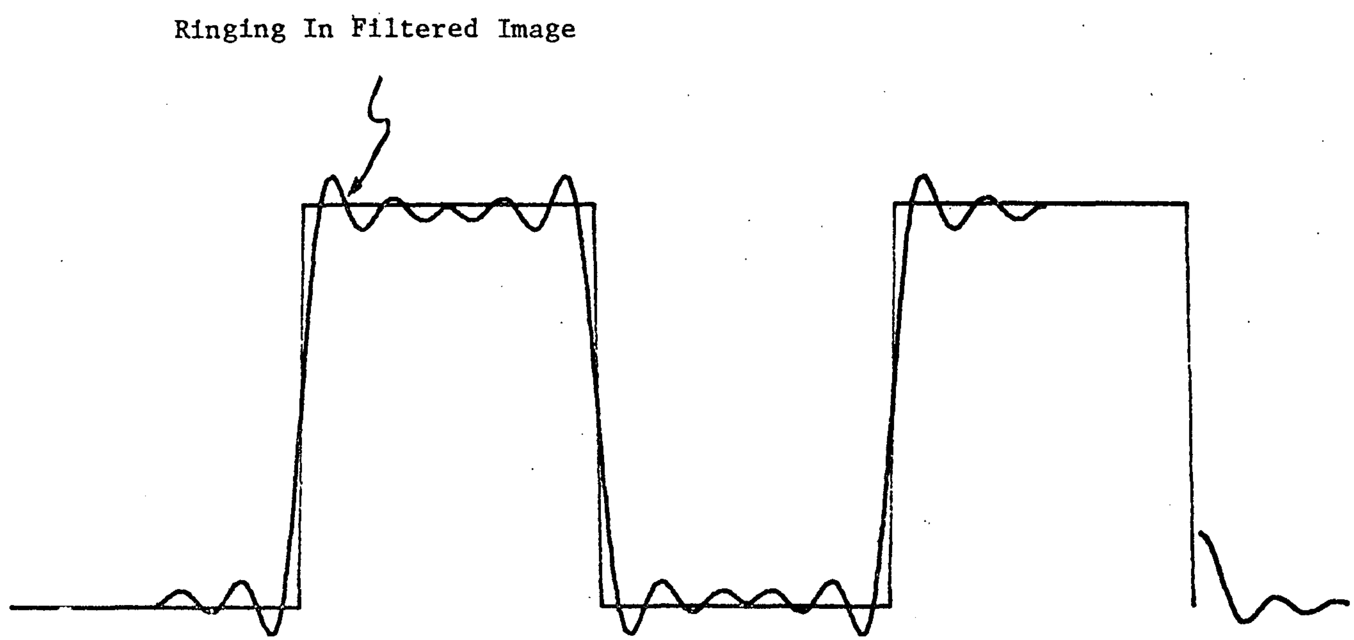


Figure 6 RINGING NEAR FEATURES OF HIGH CONTRAST

Ringing would appear as a blur or loss of resolution in two-dimensional images. Simple (unweighted) averaging filters cause severe blurring.

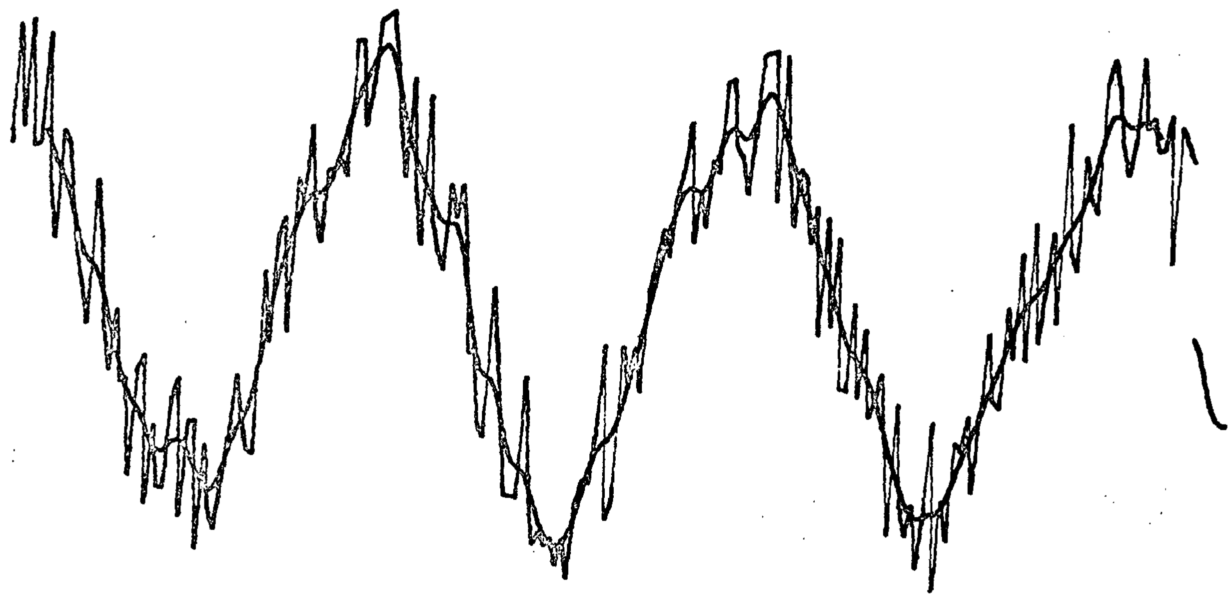


Figure 7 GAUSSIAN FILTER APPLIED TO RANDOM NOISE

Essential image is detected with slight ringing.

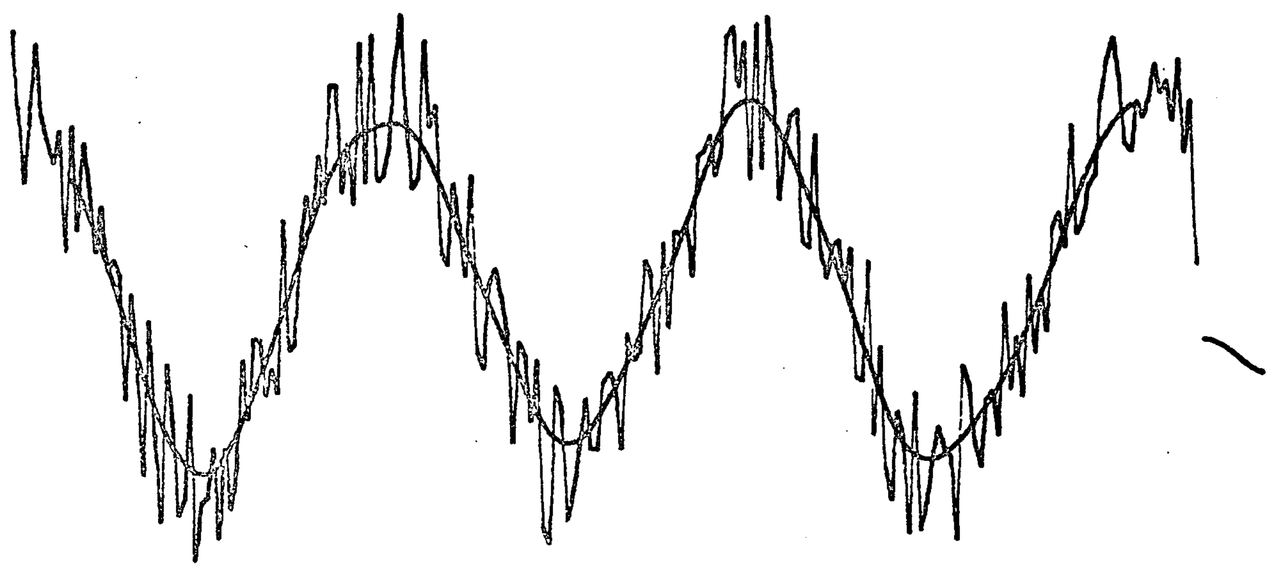


Figure 8 COSINE FILTER APPLIED TO RANDOM NOISE

Detection of essential image by the Gaussian filter suggested use of a cosine filter to improve the quality of the output image.

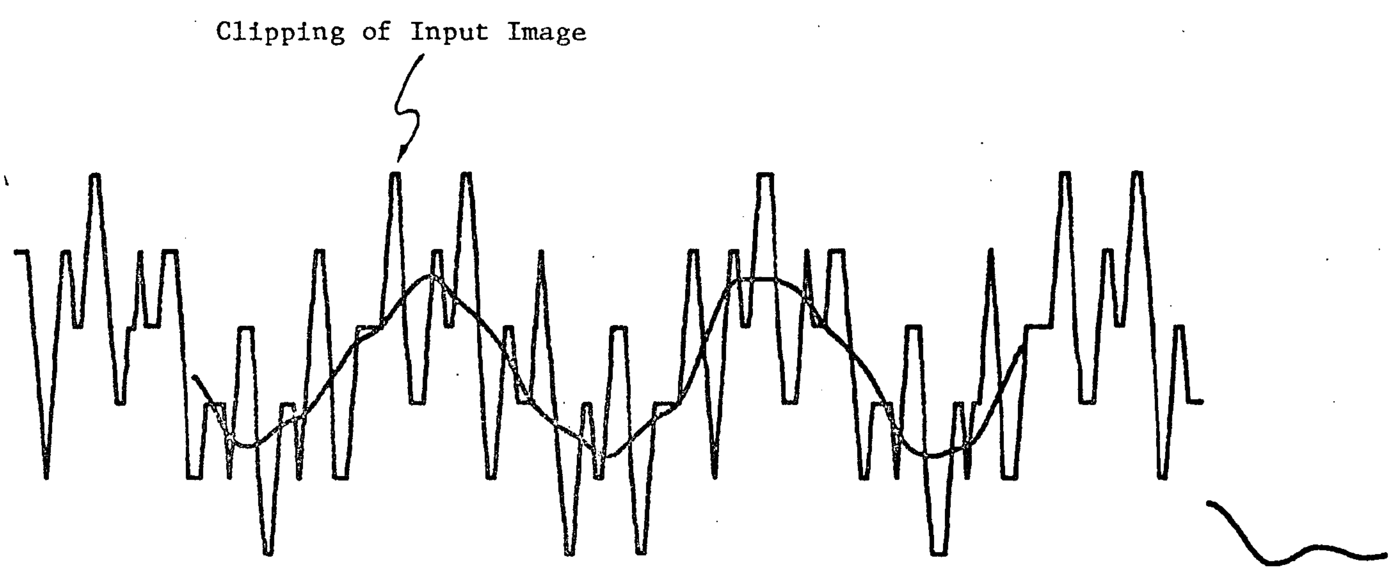


Figure 9 FILTERING BY WEIGHTED AVERAGE

The sine filter detects the essential image with slight ringing even though the input image is partially clipped. Clipping may indicate a partial loss of the input signal or it may be done intentionally to remove useless information prior to filtering.

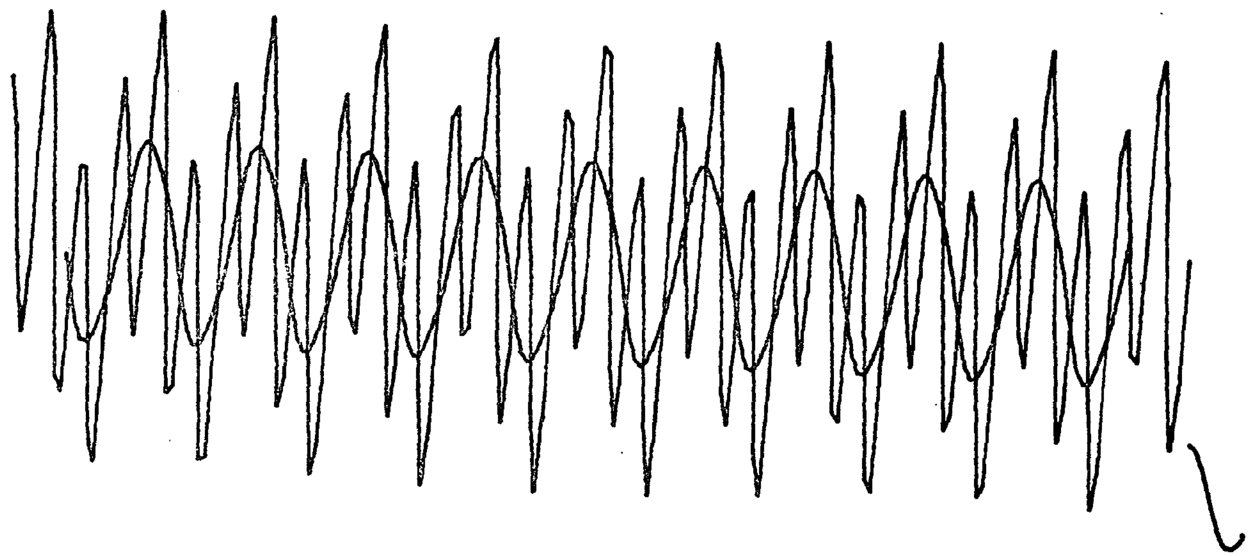


Figure 10 ATTENUATION OF DISTORTING HIGH FREQUENCIES

The sine filter preserves essential high frequencies but attenuates high frequencies causing image degradation.

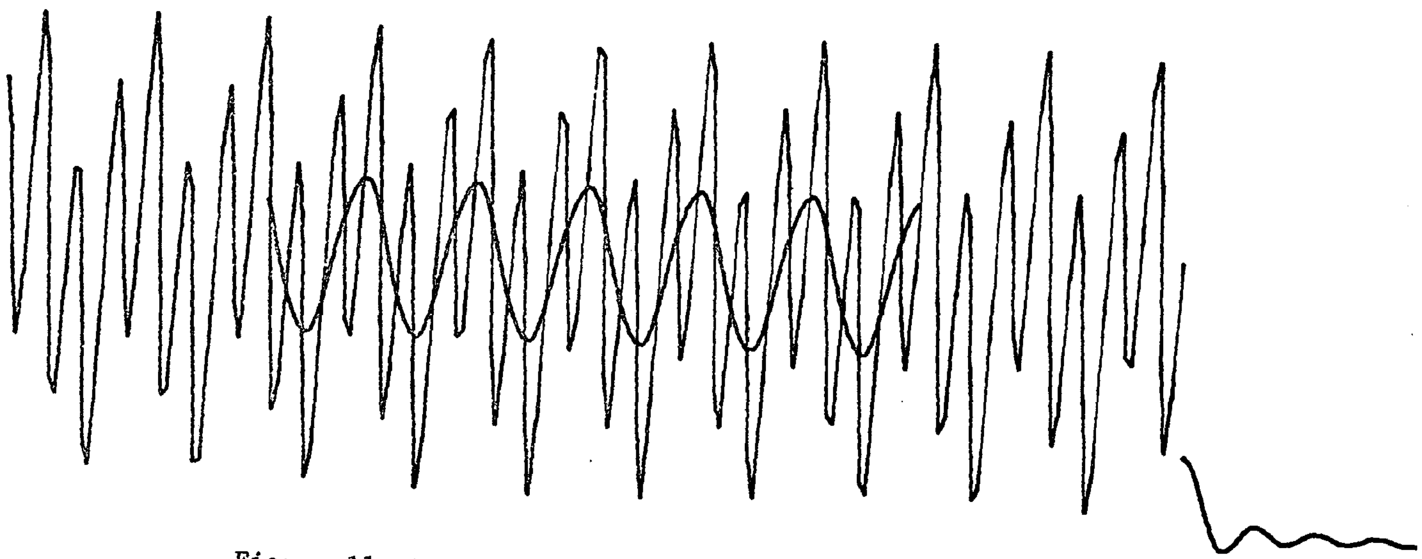


Figure 11 IMPROVING IMAGE BY INCREASING FILTER SIZE

Increasing filter size from 10 to 45 elements further attenuates distorting high frequencies and improves resolution of filtered image.

Table 1 COMPARISON OF FILTERED IMAGES

Figure	Input Image	Filter Size (d)	Filter
6	$\frac{1}{2}\,[(-1)^{(\frac{x}{50})} + 1]$	25	$\frac{\sin .6x}{x}$
7	$\frac{1}{2}\,(\cos \frac{x}{10}$ + Random number)	6	$e^{-(\frac{x}{2.5})^2}$
8	$\frac{1}{2}\,(\cos \frac{x}{10}$ + Random number)	10	$\cos^2 \frac{\pi x}{2(d+1)}$
9	$\frac{1}{5}\,[3 + \sin x + \cos \frac{x}{2} + \sin \frac{x}{9}]$	30	$\frac{\sin .4x}{x}$
10	$\frac{1}{5}\,[2 \cos x + \sin \frac{x}{3} + 3]$	10	$\frac{\sin .6x}{x}$
11	$\frac{1}{5}\,[2 \cos x + \sin \frac{x}{3} + 3]$	45	$\frac{\sin .6x}{x}$

function. If the noise is truly random, it performs as well as the Gaussian filter. If the noise is periodic, the $\frac{\sin \pi k}{\pi k}$ function acts as a low-pass filter. It seems to preserve essential high frequencies, but attenuates those high frequencies which cause distortion. It was also noted that this filter gave better results when the filter size was increased from 13 to 47 elements.

To test the efficiency of the filter, the filter is convolved with a noiseless image. A good filter should preserve the original image with little attenuation. See figures 6 through 11.

Systematic Noise and Modulation Transfer Functions.

Let us consider the filtering of systematic noise, i.e. noise generated by certain system elements. Suppose the input picture of equation (2-13) has been degraded by position invariant linear operations, i.e. by the successive convolutions of equation (1-10) such that

$$f_I = f*S_1*S_2*....*S_n \quad . \tag{3-1}$$

Then equation (2-13) becomes,

$$f_o = f*S_1*S_2*....S_n*S_p \quad . \tag{3-2}$$

Taking the Fourier transform of both sides,

$$F_o = F\hat{S}_1\hat{S}_2....\hat{S}_n\hat{S}_p \quad . \tag{3-3}$$

Let $G = (\hat{S}_1\hat{S}_2....\hat{S}_n)$ be the cummulative transfer function for the entire lens-film-digital system (the system transfer function), then

$$F_o = (FG)\hat{S}_p . \tag{3-4}$$

The most obvious means of constructing a compensating filter is to let $\hat{S}_p = 1/G$, multiply by the Fourier transform of the degraded image, and take the inverse Fourier transform to restore the ideal image. Letting $\hat{S}_p = 1/G$ equation (3-4) becomes

$$f_o = F^{-1}[FG/G] = f . \tag{3-5}$$

When this approach is used, the filtering is carried out in the spatial frequency domain (Fourier domain) rather than in the space domain; hence, the name spatial filtering. [25] (Some authors refer to this technique as inverse filtering [26].)

Modulation Transfer Functions.

It is obvious that knowledge of the system transfer function G is essential for construction of the filter S_p which will give the best restoration. When normalized the Fourier transform of the spread function is called the optical transfer function. Using one dimension for simplicity we denote the normalized transform by

$$N(k) = \frac{\int_{-\infty}^{\infty} S(x) e^{-2\pi ikx} dx}{\int_{-\infty}^{\infty} S(x) dx} \tag{3-6}$$

where k denotes spatial frequency. The above expression for the optical transfer function contains both an amplitude term and a phase term. Usually the phase term is important only when considering coherent illumination, as in a laser. However, for most TV and photographic analysis the illumination is incoherent so only the amplitude term is important. [27] The transfer function is then called the modulation transfer function (MTF), and it is designated

$$T(k) = \left|N(k)\right| = \frac{\left|\int_{-\infty}^{\infty} S(x)\cos 2\pi kx dx - i\int_{-\infty}^{\infty} S(x)\sin 2\pi kx dx\right|}{\left|\int_{-\infty}^{\infty} S(x)dx\right|} . \tag{3-7}$$

In actual practice, $T(k)$ is used instead of S_p in equation (3-4). The normalized filter is used to preserve the range of gray levels in the picture function.

Modulation transfer functions are used for evaluating degradation of information by the system components individually or in combination. The lens will be used as the primary example for discussion, although the utility of the transfer function lies in the fact that it can be used to describe the net effect on the image as well as the individual contributions of lens, film, scanner elements, etc.

The MTF or frequency response is commonly measured with a sine wave test target. This is a pattern of bars and spaces with a sinusoidal variation in intensity which can be described by the equation: (see also Figure 12)

$$I_o(X_o) = A_o + B_o \cos W_o X_o$$

where,

$I_o(X_o)$ = target brightness as a function on X_o

A_o = average brightness = $\frac{I\ max + I\ min}{2}$

B_o = $\frac{I\ max - I\ min}{2}$

w_o = $2\ \pi\ N_o$, where N_o indicates lines/mm

x_o = distance in object plane perpendicular to target bars.

The modulation, $M_o = \frac{B_o}{A_o}$ and

the contrast, $C_o = \frac{1 + M_o}{1 - M_o}$.

In the relationships above, the subscript o denotes the object plane. The same formulas describe the relationships of these parameters as the object scene is reproduced in the image plane with the subscript i used to denote the image plane. Thus, the modulation in the image plane becomes

$$M_i = \frac{B_i}{A_i}\text{, and the contrast } C_i = \frac{1 + M_i}{1 - M_i} .$$

The plot, T(k), of the ratio $\frac{M_i}{M_o}$ as a function of frequency (lines/mm) is the modulation transfer function (MTF) of the lens

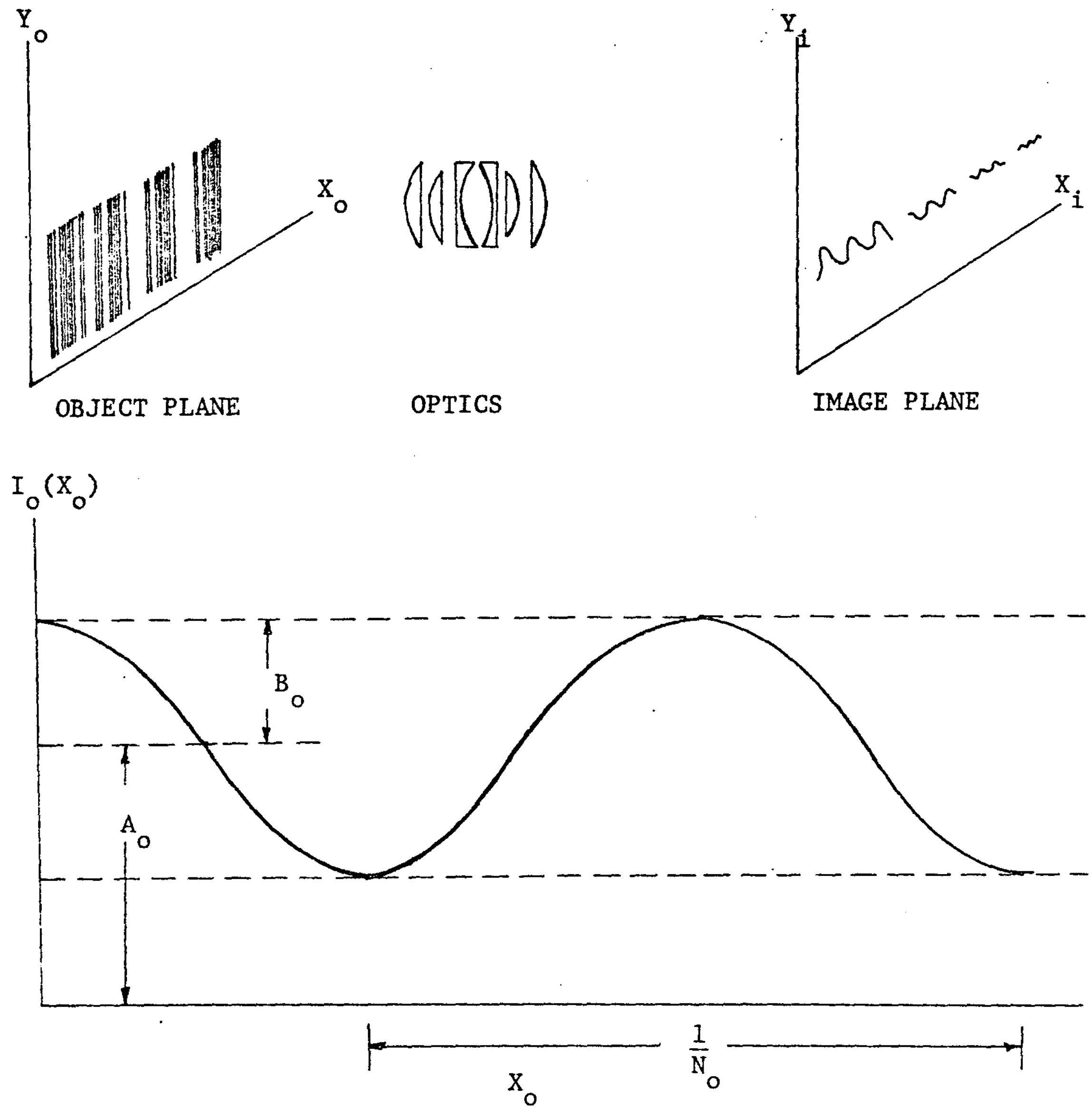


Figure 12 REPRESENTATION OF SINE WAVE RESPONSE MEASUREMENT (Adapted from [28])

(or particular system element),

$$\text{MTF} = \frac{M_i}{M_o} \text{ as a function of frequency}\ .$$

Experimentally, the MTF of a lens (or lens + film, etc.) can be determined by measuring the object modulation (contrast) and resultant image modulation for a series of sine wave test targets of varying spatial frequencies or line spacings. [29].

The abberation-free lens system reproduces a point according to the diffraction limit as represented by the spread function of the common Airy Disc. Taking the Fourier transform of the Airy diffraction image gives the frequency response curve or MTF as shown in Figure 13. An actual lens system is not abberration free and thus the response is lower than that of the theoretical diffraction limited function. The MTF of the 24 inch focal length, f/5.6 LOS lens is shown for comparison by the dashed lines in Figure 13. [30].

Cascading the MTF's.

The film, photomultiplier tube, and other elements of the system have finite minimum image spread functions and thus, MTF's. The MTF for the total system is simply the product of the MTF's for each individual element. An illustrative example is shown in Figure 14.

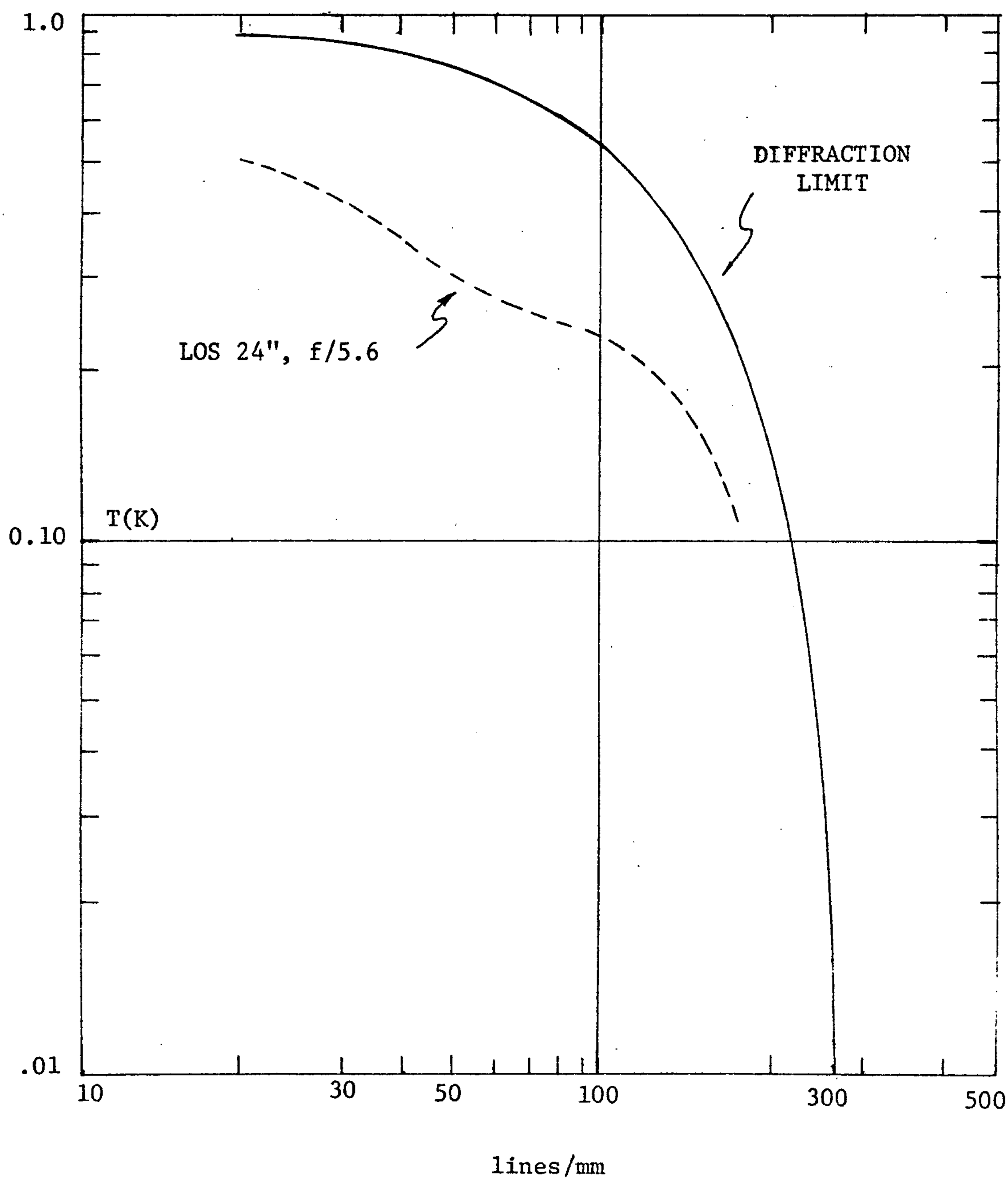


Figure 13 MTF (Fourier transform) OF DIFFRACTION LIMITED (Airy Spread Function) f/5.6 OPTICS AT 5500Å (Adapted form [28])

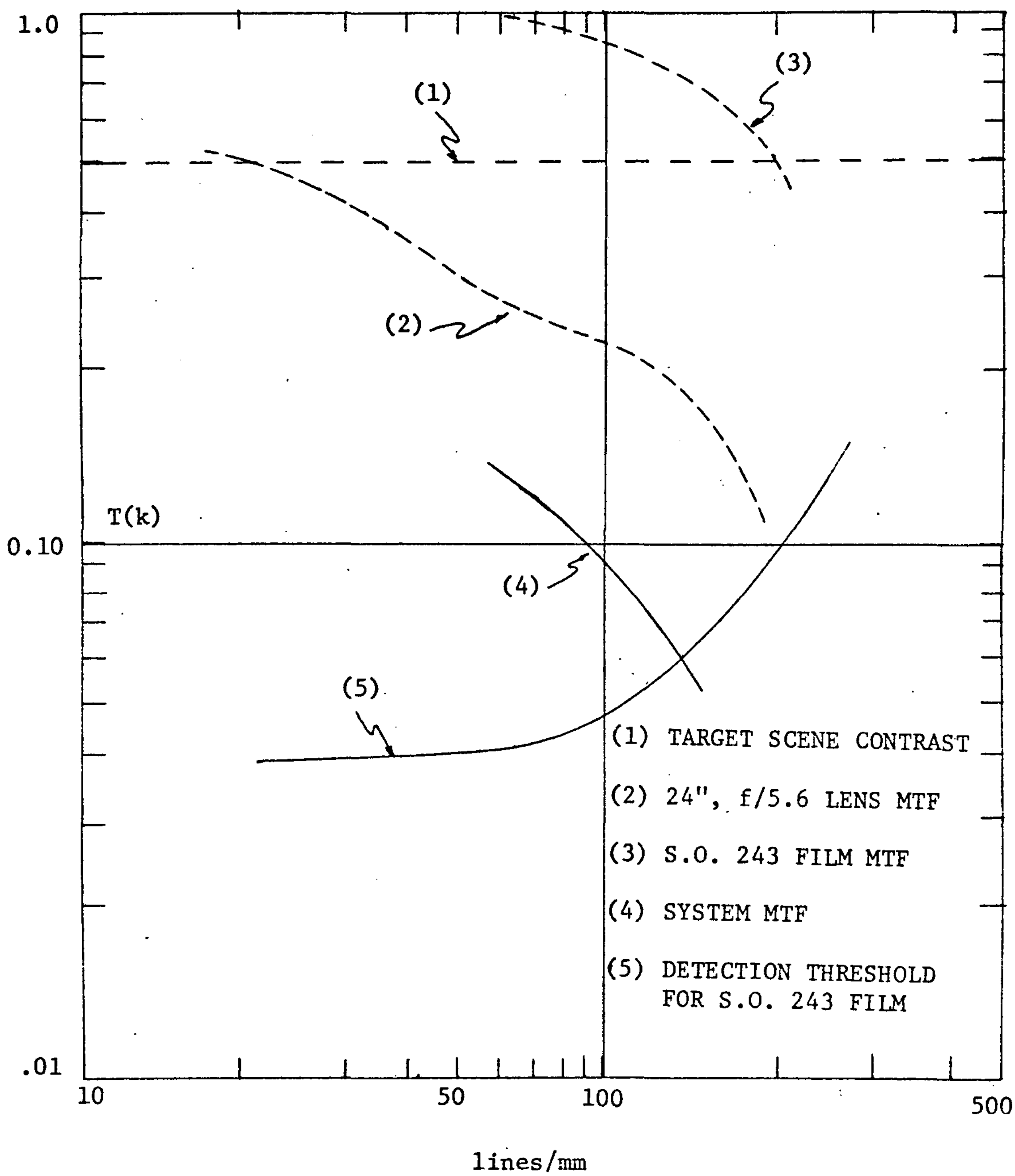


Figure 14 MTF ANALYSIS OF SIGNAL AND MINIMUM MODULATION REQUIRED FOR DETECTION (Adapted from [28])

In many instances the modulation or contrast of an object may be reduced by atmospheric haze. With reference to the notations on Figure 14, a 50% modulation of the target scene is plotted as line (1); the MTF of the 24 inch lens is curve (2); the MTF, of S.0.243 film with NIMAT processing is curve (3). Thus, by cascading these functions, the resultant system MTF is shown by curve (4). [31]

Image Resolution and MTF's

Whether the combined system response, with any certainty, can resolve the target scene depends on the location of the detection threshold. If the detection threshold intersects the system response curve at a spatial frequency greater than the spatial frequency of the target, detection is unlikely. [32] The threshold detection curve for S.0.243 film is shown as curve (5) in Figure 14. As indicated, the resolution of this system is approximately 140 lines/mm.

Problems Associated with the MTF's.

Image degradation is seldom the result of only position invariant linear operations (convolutions). Zeros occur in some MTF's making it impossible to obtain finite reciprocal functions for compensating filters. [33] Nevertheless, good results can be obtained by approximating the reciprocal functions with finite values.

When little is known about the system MTF, assume a Gaussian transfer function. The compensating filter then becomes an inverse Gaussian curve which can be implemented on a digital computer. [34] (See [8, 19] for an excellent discussion on the calculation and application of MTF's.)

SUMMARY

A major question for investigation still remains: what to do when noise and systematic noise occur simultaneously? There is no standard answer. There is no single filter which will restore all degraded images; each image must be uniquely filtered and analyzed to restore its information content. Because the researchers of image processing come from many diverse fields (e. g. Fourier analysis, optics, spectral analysis, communication and information theory, photographic reconnaisance, micro-biology, etc.), each person, depending upon his research information, developed heuristic techniques to solve his particular problem. Only a general algorithm has evolved.

It is generally safe to assume the presence of both systematic and random noise in a degraded image. Operate first by the method of modulation transfer functions to obtain a useful image. (If one does not know the system MTF, one may try to filter with an inverse Gaussian curve.) Secondly, continue image enhancement by smoothing the remaining noise with some type of averaging filter.

Until the field of image processing becomes more standardized, each researcher must develop the filters to suit his particular needs. Any attempt at image processing must understand the following [35]:

1. Because of limitations on resolution and because of the presence of noise, any image contains only a finite

amount of information.

2. The fundamental limit of processing is achieved when the human visual system is able to extract from the processed image all information contained in the original degraded image.
3. No amount of linear processing can increase the information content of an image.

APPENDIX A

Proof of the convolution theorem. One dimension is shown for simplicity.

The Fourier transform $G(\omega)$ of a function $g(x)$ is defined by

$$G(\omega) = \int_{-\infty}^{\infty} g(x)e^{-i\omega x}dx \qquad \text{where} \quad \omega = 2\pi u \quad .$$

Given the convolution

$$f_o(x) = \int_{-\infty}^{\infty} f(x')S(x-x')dx'$$

taking the Fourier transform of both sides,

$$F_o(\omega) = \int_{-\infty}^{\infty}\int f(x')S(x-x')e^{-i\omega x}dxdx' \quad ;$$

letting

$$t = (x-x') \quad ,$$

$$F_o(\omega) = \int_{-\infty}^{\infty}\int f(x')S(t)e^{-i\omega t}e^{-i\omega x'}dx'dt$$

$$F_o(\omega) = \int_{-\infty}^{\infty} f(x')e^{-i\omega x'}dx' \int_{-\infty}^{\infty} S(t)e^{-i\omega t}dt$$

$$F_o(\omega) = F(\omega)S(\omega).$$

APPENDIX B

Spatial Frequency and Resolving Power.

The Figure 15 shows an Air Force resolution test chart made up of a series of progressively smaller patterns. Each pattern consists of lines and spaces of equal width. The usual procedure employed to find the resolving power of an imaging system is to photograph such a test chart and then pick out the finest pattern that can be discerned. The reciprocal of the width of a line plus a space is called the limiting resolving power of the system. The units are stated as spatial frequency in optical line pairs per millimeter or, more simply, lines per millimeter (lpm) or cycles per millimeter (cpm). If we denote the limiting spatial frequency by K_o, then

$$K_o = \frac{1}{a} = \frac{D}{\lambda F} = \frac{1}{\lambda(\text{f-number})}$$

where "a" now represents the combined width of a line and space. The ratio F/D is called the F-number or relative aperture of the lens, and λ is light wavelength. [36]

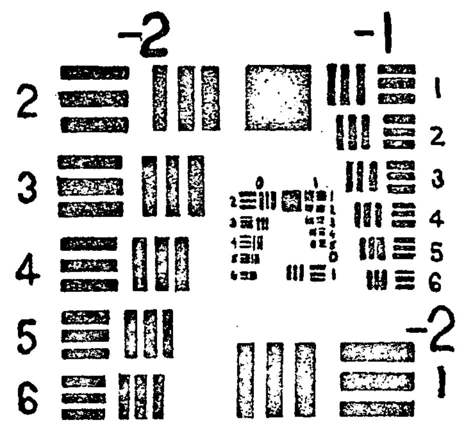


Figure 15 HIGH-CONTRAST RESOLUTION TEST CHART
(Adapted from [36])

REFERENCES

1. A. Rosenfeld, Picture Processing by Computer, 1-2 (Academic Press, New York, 1960).

2. L. A. Wainstein and V. D. Zubakov, translated from the Russian by R. A. Silverman, Extraction of Signals from Noise, 1-2 (Dover Publications, Inc., New York, 1970).

3. A. Rosenfeld, op. cit., 35.

4. J. D. Bryant, Position-invariant operations, (Lecture notes, Texas A&M University, 1970).

5. A. Rosenfeld, op. cit., 88-93.

6. H. C. Andrews, Computer Techniques in Image Processing, 6-8 (Academic Press, New York, 1970).

7. R. A. Wilson, Optical Page Reading Devices, 27-46 (Reinhold Publishing Corp., New York, 1966).

8. N. Jensen, Optical and Photographic Reconnaisance Systems, 21-24 (John Wiley and Sons, Inc., New York, 1968).

9. L. Levi, Applied Optics; A Guide to Optical System Design/ Vol I, 130-132 (John Wiley & Sons, Inc., New York, 1968).

10. A. Rosenfeld, op. cit., 40-42.

11. J. L. Harris, Image evaluation and restoration, J. Opt. Soc. Amer., 56, 570 (May, 1966).

12. J. L. Harris, Ibid., 569-574.

13. B. L. McGlamery, Restoration of turbulence-degraded images, J. Opt. Soc. Amer., 57, 293-297 (March, 1967).

14. G. B. Anderson and T. S. Huang, Errors in frequency-domain processing of images, AFPIS Conference Proceedings, 34, 173-175 (Spring, 1969).

15. P. F. Mueller and G. D. Reynolds, Image restoration by removal of random media degradation, J. Opt. Soc. Amer., 57, 1338-1343 (November 1969).

16. H. C. Andrews, op. cit., 19-22.

17. H. C. Andrews in Automatic Interpretation and Classification of Images, Edited by A. Grasselli, 192 (Academic Press, New York, 1969).

18. R. Nathan, Digital Video-Data Handling, JPL Technical Report No. 32-877, 6-8 (January 5, 1966).

19. The BOEING Company, Picture Data Systems Analysis, Document No. D2-100293-1, 15 (December 9, 1964).

20. A. Rosenfeld, op. cit., 88.

21. S. Wantanabe, Methodologies of Pattern Recognition, 220-221 (Academic Press, New York, 1970).

22. R. Nathan, op. cit., 15-16.

23. A. Rosenfeld, op. cit., 88-93.

24. R. E. Graham, Snow removal - a noise stripping process for picture signals, IRE Trans. Information Theory, IT-8, 129-144 (February 1962).

25. A. Rosenfeld, op. cit., 79-83.

26. H. C. Andrews, Computer Techniques in Image Processing, 20, (Academic Press, New York, 1970).

27. N. Jensen, op. cit., 26.

28. The BOEING Company, op. cit., 89-97.

29. The BOEING Company, op. cit., 92.

30. The BOEING Company, op. cit., 90.

31. The BOEING Company, op. cit., 95.

32. N. Jensen, op. cit., 204.

33. H. C. Andrews, op. cit., 36.

34. H. C. Andrews, op. cit., 36-40.

35. J. L. Harris, op. cit., 574.

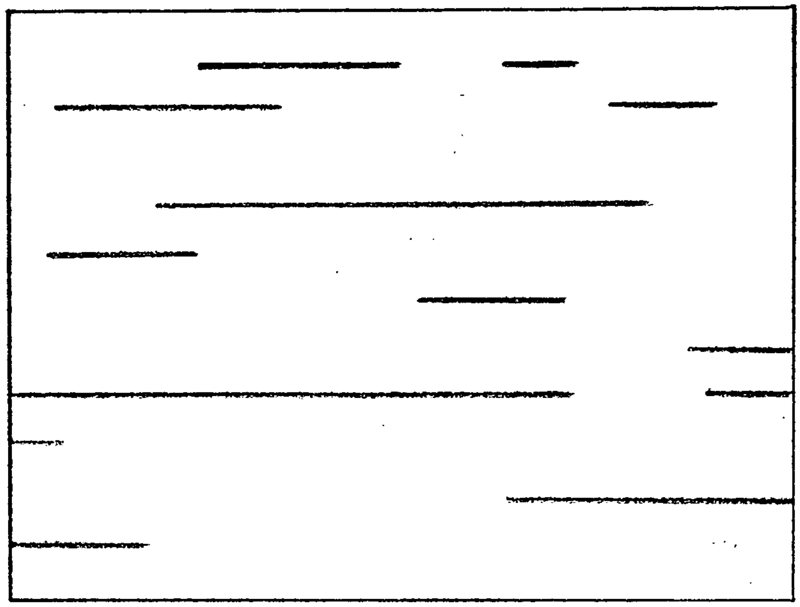
36. N. Jensen, op. cit., 13-14.

2. EARLY ATTEMPTS TO ENHANCE DATA

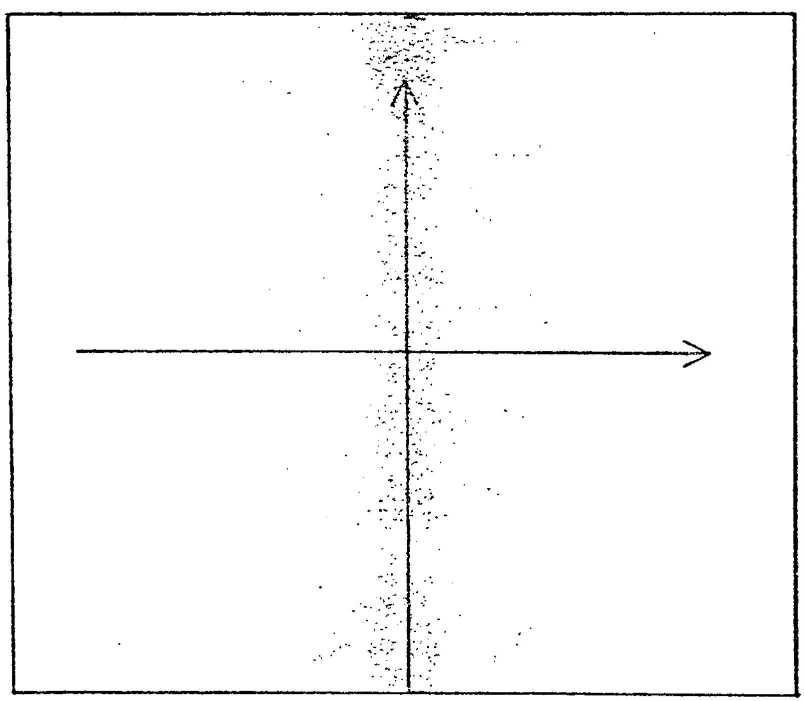
Early attempts to construct filters were modeled after an approach suggested by Robert Nathan. His data was not photographic, but some of the noise sources (viz. (iv), analog signal noise added in transmission) he had to cope with were also present in our data. It is not necessary to completely describe Nathan's method; however, we will outline the steps we took to attempt to utilize it.

The treatment of the kinds of noise in LO data requires bringing the discussion to two dimensions in both the real and frequency domain. Our data is different from ordinary photography in that it has been scanned and dititized in some particular direction. Because not every scan line is perfectly reproduced, a very high frequency noise is added to the signal in the spatial frequency domain in the direction perpendicular to the scan line direction (Fig. 2.1). To supress noise in his (video) system, Nathan developed a "scan-line filter" (which is used in the Analysis Program to remove noise in the (real) direction perpendicular to the radial lines) which was to remove scan line noise. We found this filter to be particularly ineffective in our (digitized LO) data. A closer examination of the filter reveals high sensitivity to jitter. Also, film granularity noise appeared after application of the scan line filter, which we had not anticipated, and which took on an "oblong" appearance. Finally, the filter had only a negligible effect on the analysis program produced elevation profiles.

Since our noise was not purely scan line noise, we decided to



Scan Line Noise, Real Domain



Scan Line Noise, Frequency Domain

Fig. 2.1 Scan Line Noise.

construct filters which corrected the frequency response up to spatial frequencies of 120 l/mm (approximately the maximum reproducible at the sampling rate in the LO digitization and scan line) with a maximum enhancement of 5.0 (to keep from enhancing noise too much - the same factor choosen by Nathan in his one-dimensional analysis), with very sharp cutoff above 120 l/mm. A program was written to calculate and normalize the Fourier transform of the frequency response desired. The filters constructed were extremely large; mathematically, we discovered the reason was the rapid cutoff above 120 l/mm. A small (9x9) version of this filter was tried and found to enhance the noise so much as to be virtually useless. We predicted that larger versions would also fail; the alternation of signs in the direction of the scan line, for example, suggested a marked tendency to ring, probably owing to the sharp cutoff above 120 l/mm. Even with a 9x9 filter, ringing was observed in the neighborhood of the drummarks. Accordingly, we abandoned Nathan's approach completely in favor of an approach which took into account the noise present as well as had a more gradual cutoff at high spatial frequencies.

After these early failures, we decided to attempt to improve Nathan's filter construction method. For this purpose, we needed an estimate on both the (spatial) frequency response of the LO system and an estimate on the (frequency) spectrum of the noise to be suppressed. The first we had already obtained from calibration data furnished by Kodak and Boeing with the film and LO photographic subsystem. The second was obtained by (numerical) Fourier analysis of the grayshade areas (in the edge data) for the digitization of interest. To some extent the spectrum found by this method was as expected (in general, bell-shaped with peaks along and perpendicular to the scan line direction), although there were some unexpected (and unexplained) dips. Following the classical methods used for one-dimensional analysis and optimal filtering, we attempted to construct a filter which would be optimal (in the sense of minimal mean square error), assuming we actually knew the noise spectrum to be as we found it numerically, and that the system response was as specified. It should be emphasized that this program is more general than that carried out by Nathan and later by L. D. Nelson (of Bellcomm, 1967) in that we considered the problem of noise as well as the problem of correction of the system spatial frequency response. The Fourier transform of the presumed optimal filter (given in the frequency domain) was computed and sampled at the proper points. This filter is given in Fig. 2.2, in which only the upper left part of the filter is reproduced. (The actual filter is 7x7.) This filter was tried on edge data and on moon scene and was found to enhance the jitter even more than the filters constructed without concern for noise. We decided to start all over.

	COL 1	COL 2	COL 3	COL 4
ROW 1	-.00000482	-.00057181	-.00897643	-.02170715
ROW 2	-.00021514	-.02099293	-.23823154	-.45379540
ROW 3	-.00198360	-.15015079	-.64222568	.74363720
ROW 4	-.00409769	-.26832731	.17202499	4.91794133

Fig. 2.2 Early Filter.

3. DESIRED CORRECTIONS

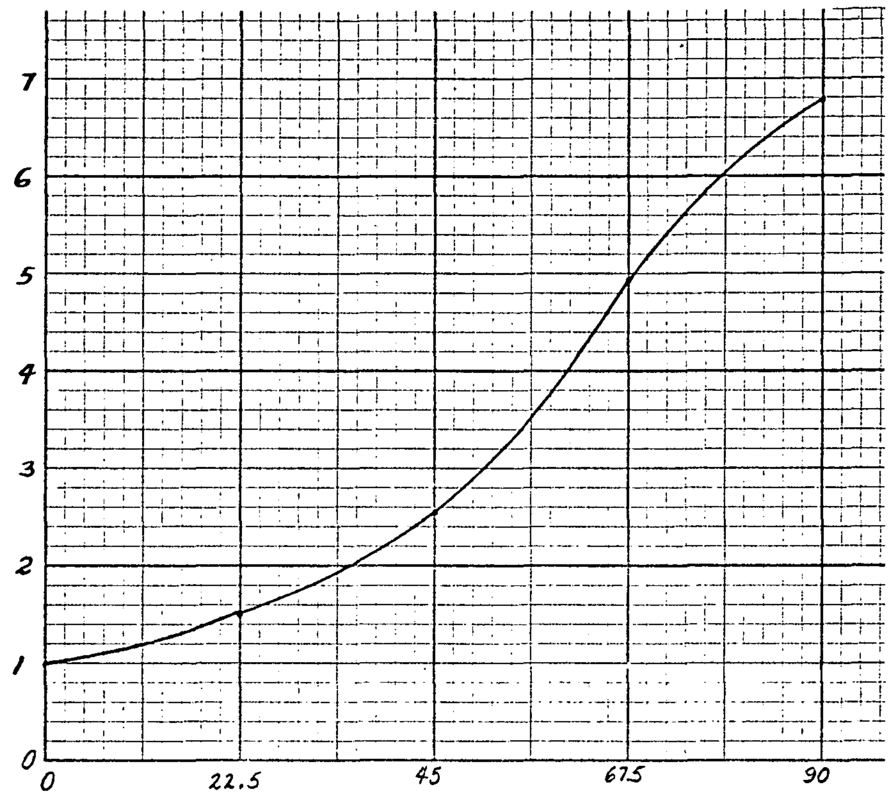
An examination of all available LO III calibration was again undertaken, this time to see if an acceptable compromise response could be obtained to allow construction of a mathematical model with enough simplicity to allow exact analysis. Table 3.1 gives the results of these averages. (Not all points are represented in all calibration data, so that some of the points are derived from an average of more data than are others.)

Frequency, 1/mm	Average Response	Ideal Correction Response
0.0	1.0	1.0
22.5	0.675	1.48
45.0	0.395	2.53
67.5	0.2015	4.96
90.0	0.1475	6.78

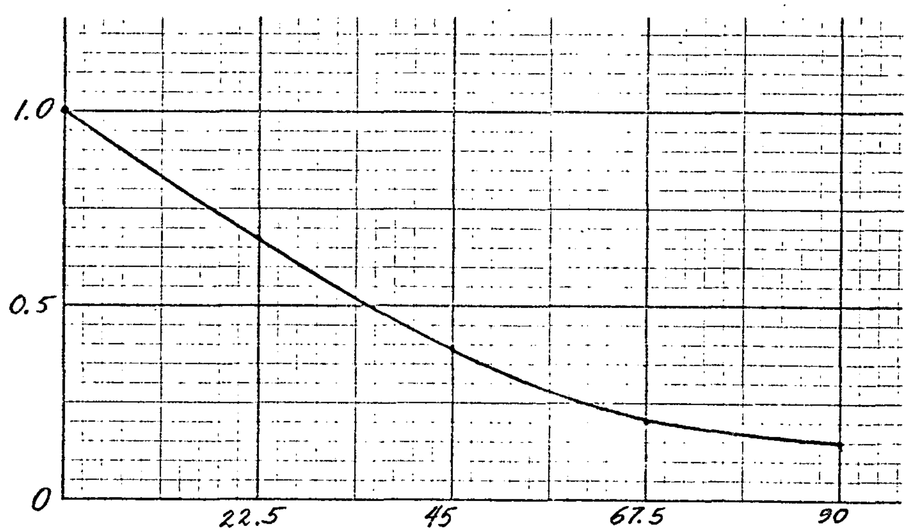
Table 3.1 Average Response, LO III.

These results are also sketched in Fig. 3.2.

An examination of Fig. 3.2 suggests that the desired correction is nearly exactly parabolic for spatial frequency under 67.5 1/mm. Since our data is sampled in two dimensions over a grid of about 4μ x 4μ, the highest frequency cutoff should be about half of 1/.0004 1/mm, that is about 125 1/mm. Accordingly, we looked for functions which were approximately parabolic near 0 and fell off rapidly above 125 1/mm.



Ideal Response vs. Spatial Frequency, Corrections.



Average Response vs. Spatial Frequency.

Fig. 3.2 Average Frequency Response and Ideal Corrections.

4. EXACT MATHEMATICAL ANALYSIS OF CHOSEN FILTERS

The desired correction of §3, combined with the requirement of smooth roll off to avoid the problems of §2, lead to filters with spatial frequency response

$$F(x,y) = a((x/b)^2 + (y/b)^2) \exp[1-((x/b)^2 + (y/b)^2)] + \exp[-((x/b)^2 + (y/b)^2)].$$

The parameter a influences how rapidly the correction function rises near 0 while the parameter b governs the behavior for large frequencies. If $a = 0$, the filter is equivalent to scanning by a Gaussian spot, the size of which is related to b by the formula $\sigma = 22/b\ \mu$. For example, if $b = 50$, $a = 0$, the filter corresponds to scan by a 4.5 μ Gaussian spot.) As a increases, the filter with this frequency response shows a peak in response for moderate frequency, decreasing rapidly (but smoothly) above the peak, and displaying approximately parabolic response near 0.

There are two reasons for this choice of filter. First, the function is easily graphed (on, e.g., the HP 9100 B system) and the parameters may be selected from an examination of families of graphs. Second, the Fourier transform

$$f(u,v) = \int\int F(x,y)e^{-2\pi i(xu + yv)}dxdy$$

of F is easily computed analytically, avoiding numerical integration of Fourier transforms. (The actual calculations are deferred to §5.)

In Figures 4.1 thru 4.5, families of curves are displayed which represent the effect of changing the parameter a with a fixed parameter b. Points at which the desired correction is known are marked with "+". In Fig. 4.6 and Table 4.7, we display ten different responses chosen for analysis. All ten were developed into filters and tests performed. The filters (which are numerical approximations to the Fourier transform of the function F given above) are applied as convolutions in the real domain, with values and results being displayed in §7.

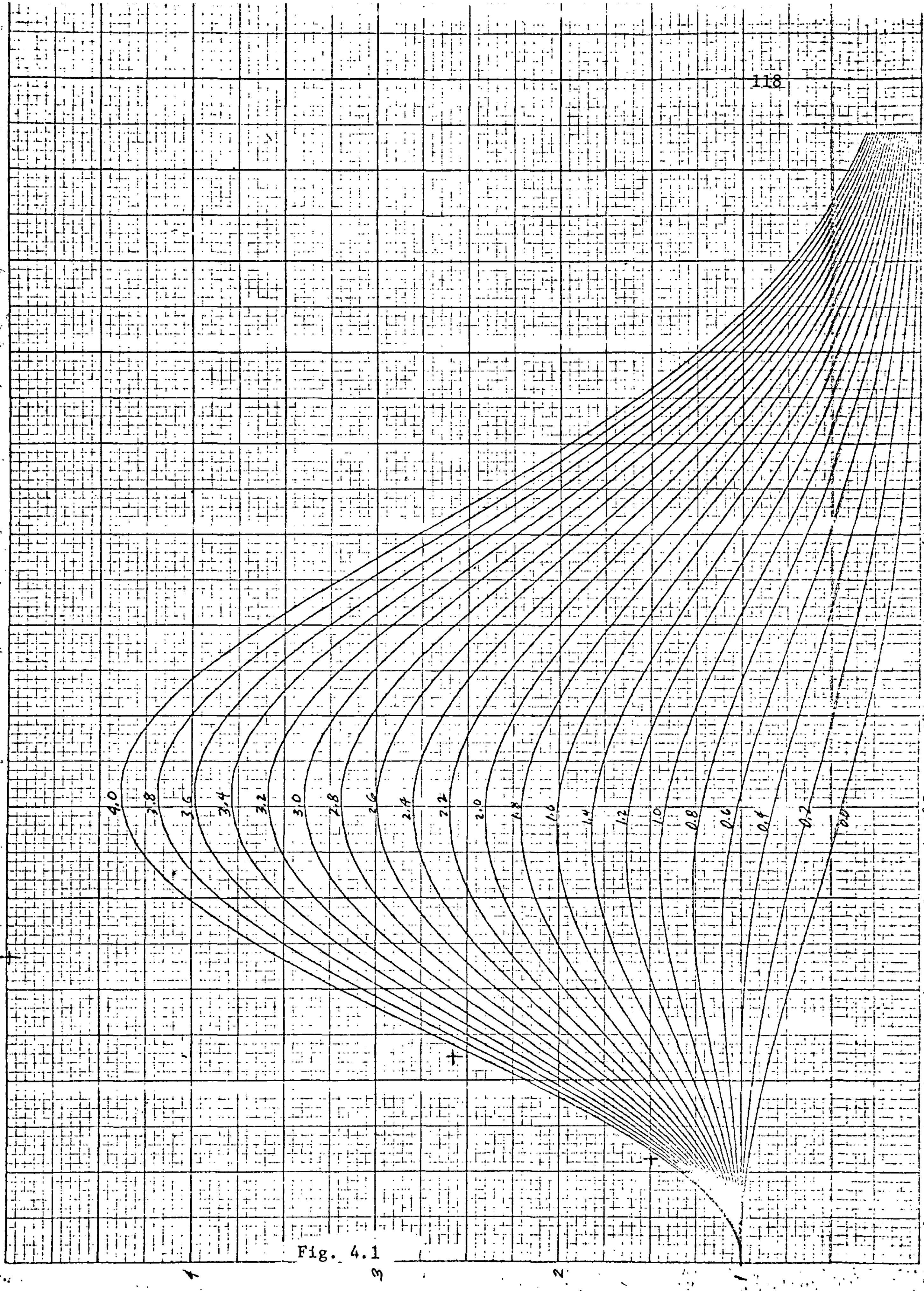

4.0
3.8
3.6
3.4
3.2
3.0
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100
120
140
160
180
200
220
240
1
2
3
4

Fig. 4.1

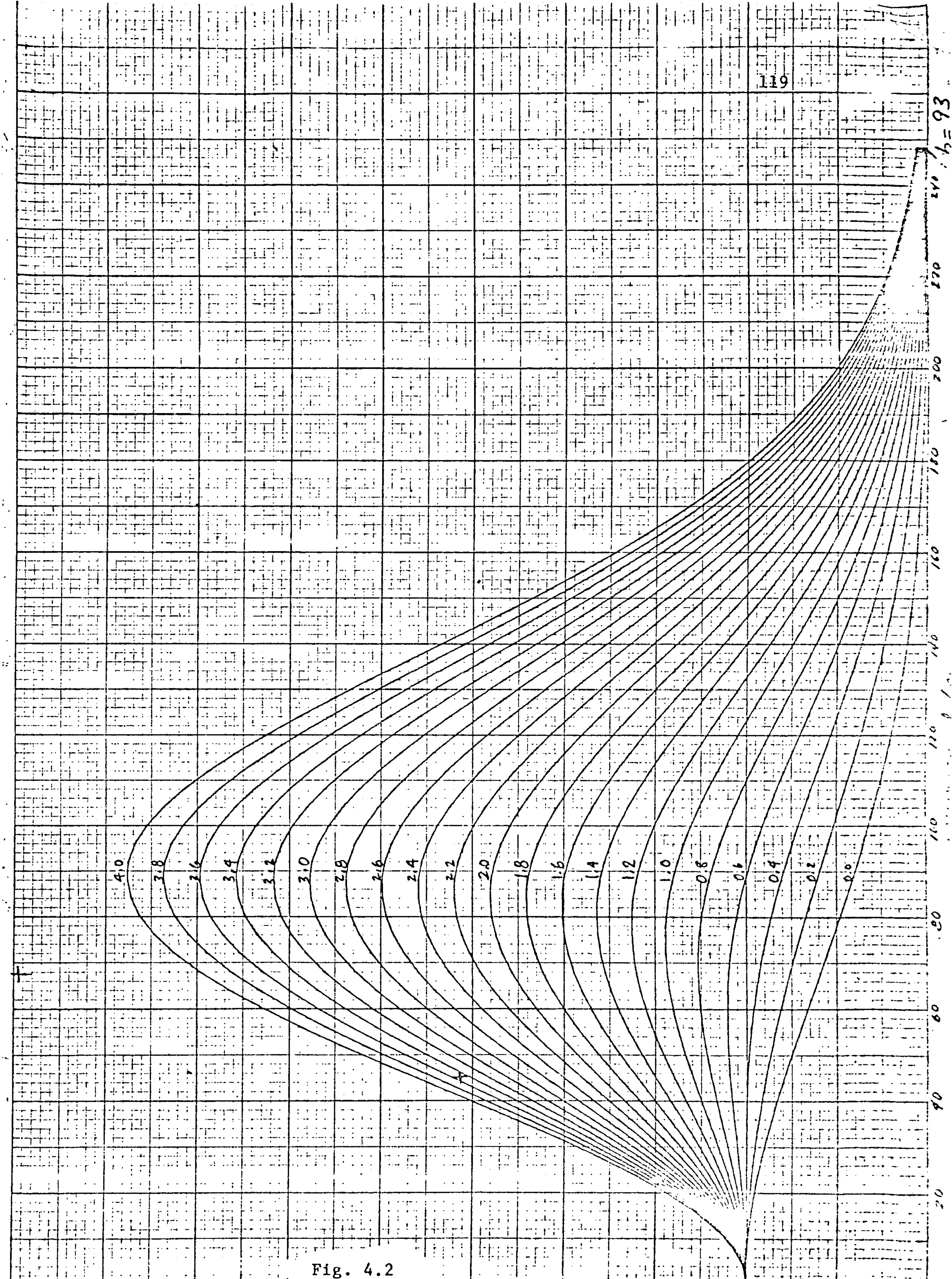


Fig. 4.2

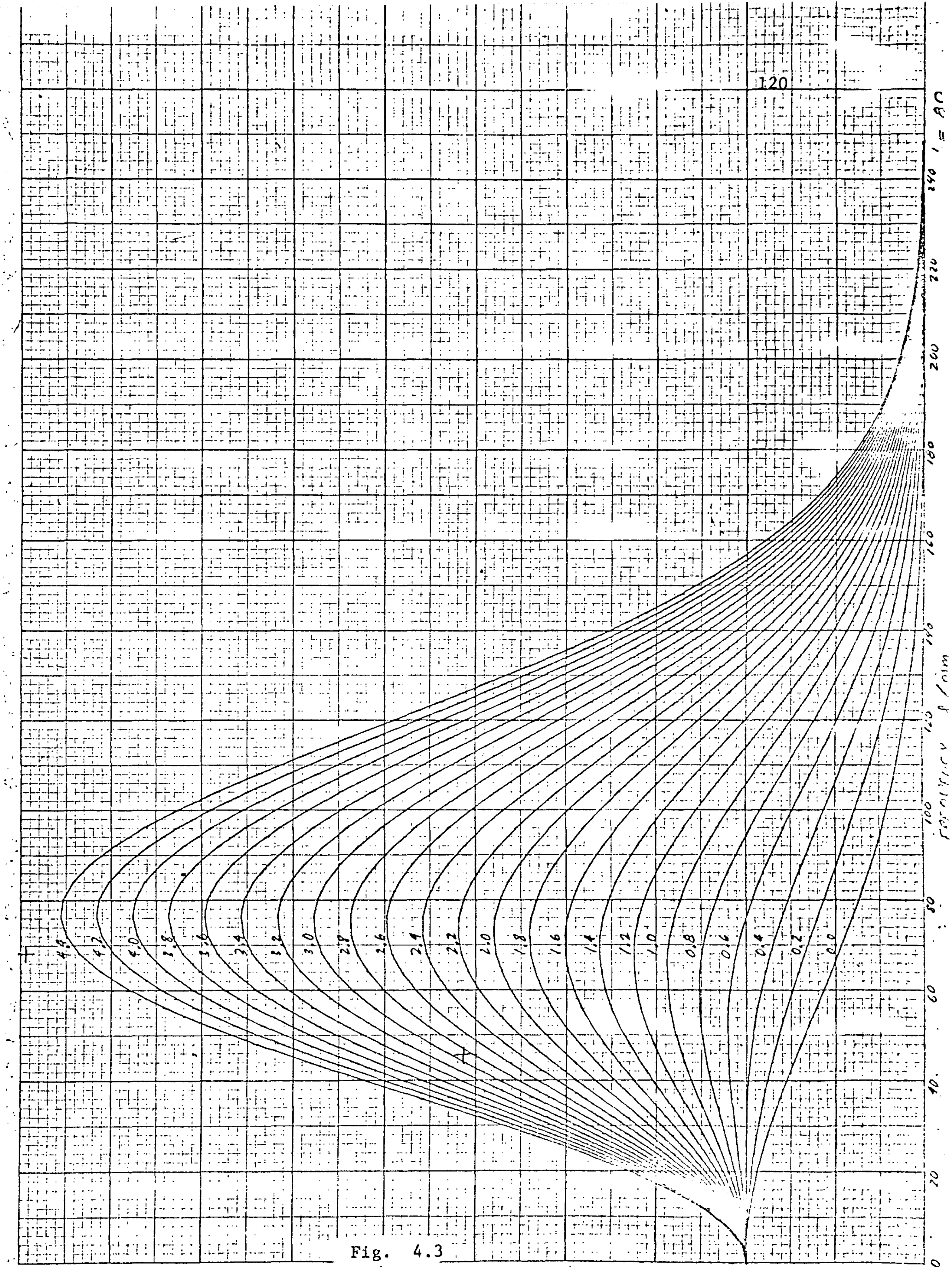

4.4
4.2
4.0
3.8
3.6
3.4
3.2
3.0
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100
140
160
180
200
220
240
1
2
3
4

Fig. 4.3

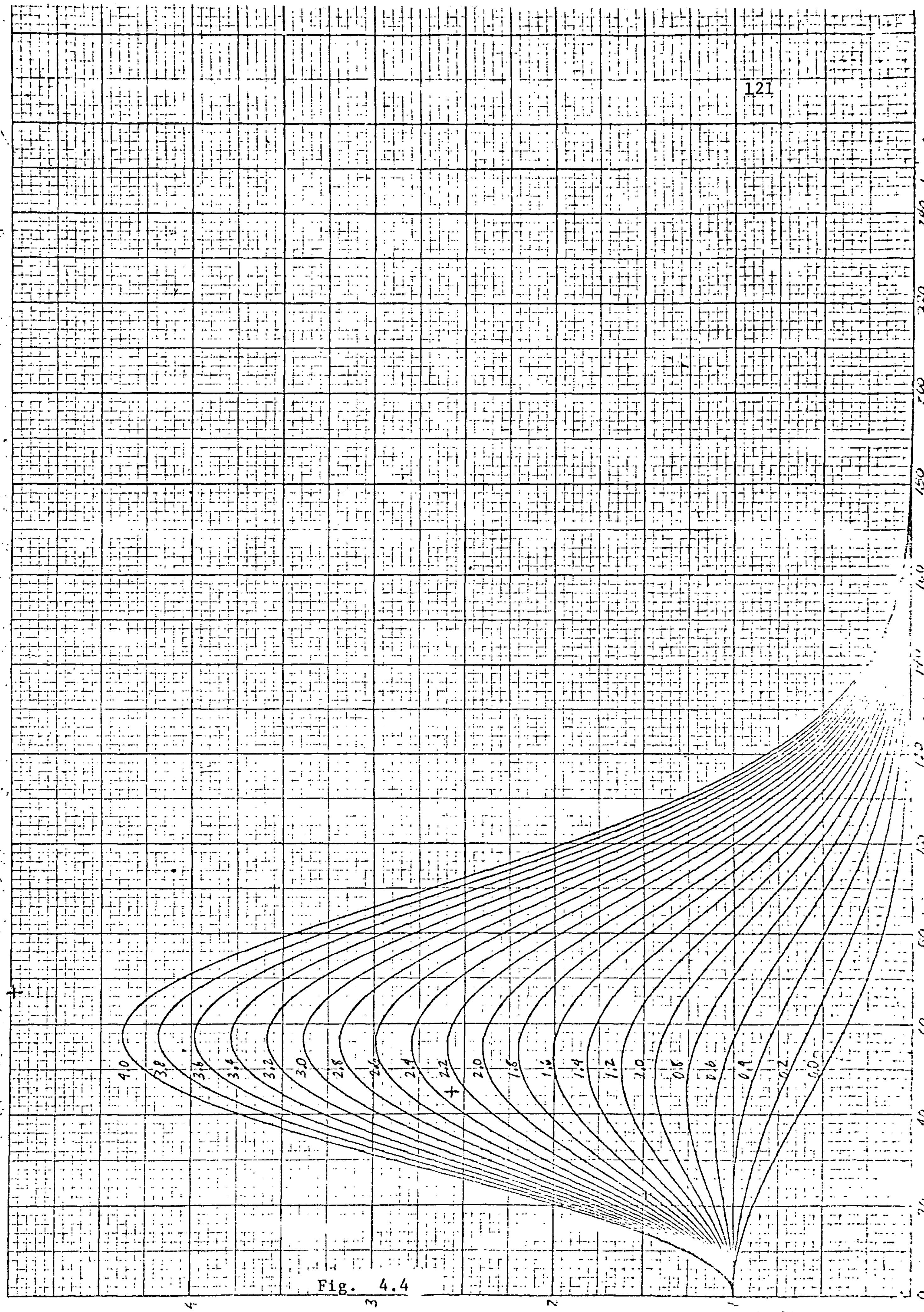

b = 60
4.0
3.8
3.6
3.4
3.2
3.0
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.4
0.2
0.0
0
20
40
60
80
240
2
3
4

Fig. 4.4

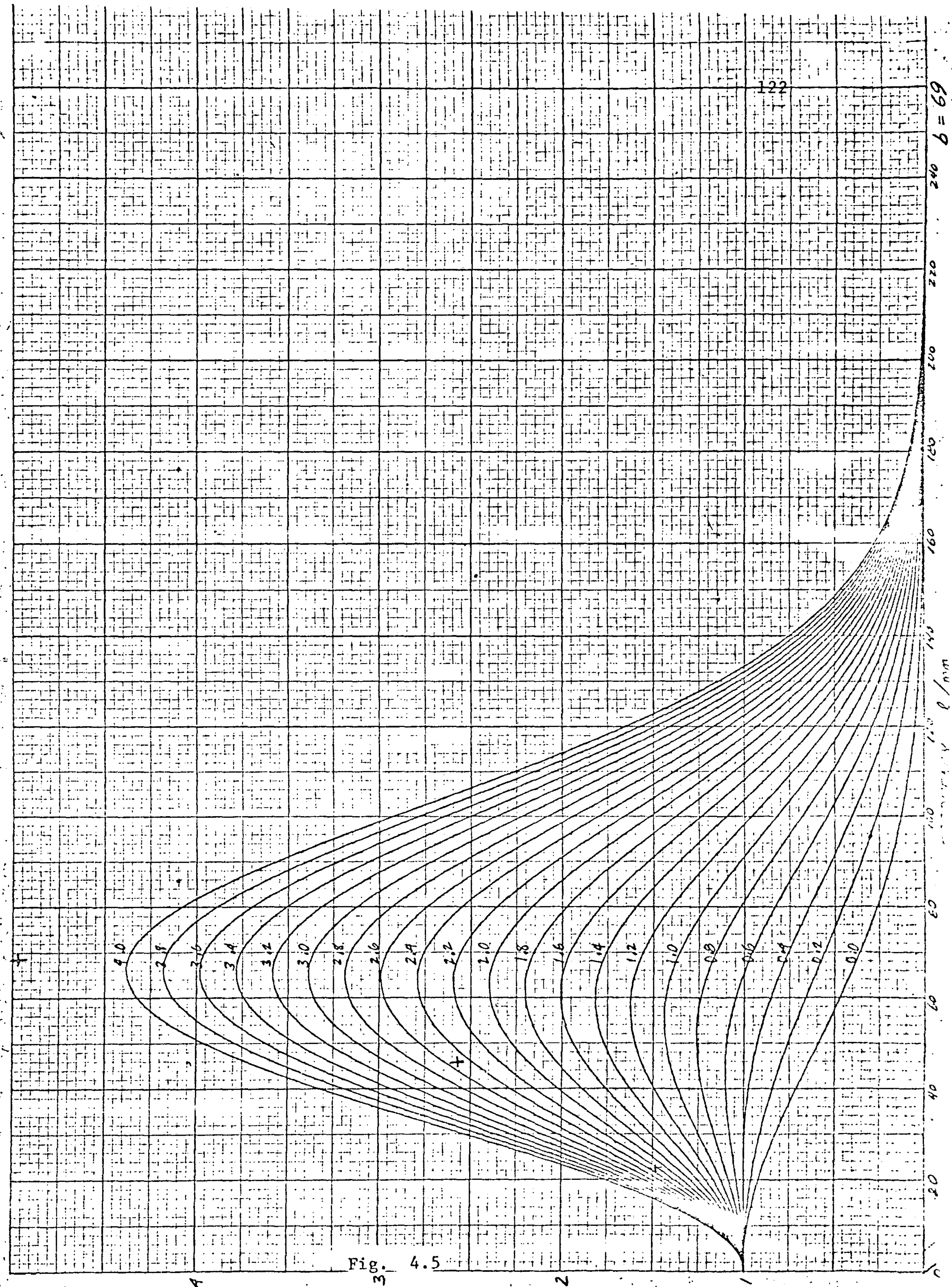

b = 69
l / mm
20
40
60
80
100
120
140
160
180
200
220
240
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0
1
2
3
4

Fig. 4.5

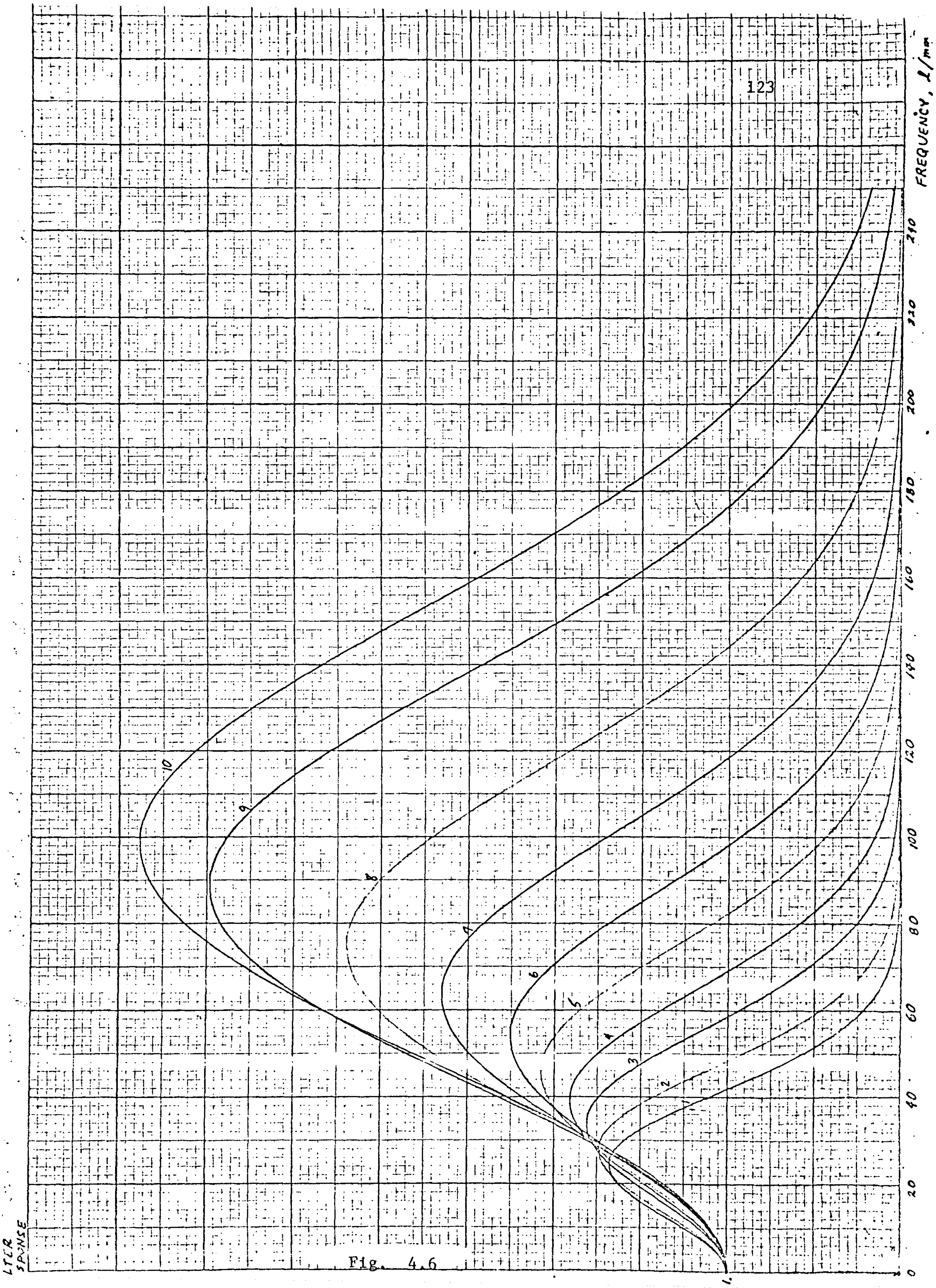

FREQUENCY, l/mm
0
20
40
60
80
100
120
140
160
180
200
220
240
1
2
3
4
5
6
7
8
9
10

Fig. 4.6

Curve	a	b
1	1.25	29
2	1.31	33
3	1.39	39
4	1.49	45
5	1.66	52
6	1.85	60
7	2.25	69
8	2.80	80
9	3.60	93
10	4.00	103

Table 4.7 Parameters in Fig. 4.6.

5. CALCULATION OF FOURIER TRANSFORM

In this section, we present the calculation of the Fourier transform of the function F given above in §4.

$$F_{\alpha\,\beta}(x,y) = \alpha((\beta x)^2 + (\beta y)^2)e^{1-((\beta x)^2 + (\beta y)^2)}$$

The Fourier transform of $F_{\alpha\,\beta}$ is given by

$$f_{\alpha\,\beta}(u,v) = \alpha \int\int ((\beta x)^2 + (\beta y)^2 e^{1-((\beta x)^2 + (\beta y)^2)} e^{-2\pi i(xu + yv)} dxdy$$

$$= \frac{\alpha}{\beta^2} \int\int (w^2 + z^2)e^{1-(w^2 + z^2)} e^{-2\pi i(wu/\beta + zv/\beta)} dwdz$$

$$= \frac{\alpha}{\beta^2} f_1(\frac{u}{\beta}, \frac{v}{\beta}) \text{ , where}$$

$$f_1(s, t) = \int\int (x^2 + y^2)e^{1-(x^2 + y^2)} e^{-2\pi i(xs + yt)} dxdy$$

$$= e \int e^{-y^2} e^{-2\pi iyt} dy \int x^2 e^{-x^2} e^{-2\pi ixs} dx$$

$$+ e \int e^{-x^2} e^{-2\pi ixs} dx \int y^2 e^{-y^2} e^{-2\pi iyt} dy$$

$$= e(f(t)g(s) + g(t)f(s)) \text{ , where}$$

$$f(t) = \int e^{-x^2} e^{-2\pi ixt} dx \text{ and}$$

$$g(t) = \int x^2 e^{-x^2} e^{-2\pi ixt} dx.$$

Now the function g is the Fourier transform of $x(xe^{-x^2})$. We have (with Fourier transforms being denoted by "$\widehat{\ }$").

$$g(t) = \widehat{x(xe^{-x^2})}(t)$$

$$= \frac{1}{-2\pi i}\frac{d}{dt}\widehat{xe^{-x^2}}(t)$$

$$= \left(\frac{1}{-2\pi i}\right)^2 \frac{d^2}{dt^2}\widehat{e^{-x^2}}(t)$$

$$= -\frac{1}{4\pi^2} f''(t).$$

Also, it is well known that (with $\sigma = 1/\sqrt{2}$)

$$f(t) = \widehat{e^{-x^2}}(t) = \widehat{e^{-\frac{1}{2}(\frac{x}{\sigma})^2}}(t)$$

$$= \sigma(2\pi)^{1/2}\, e^{-\frac{1}{2}\left(\frac{t}{1/(2\pi\sigma)}\right)^2}$$

$$= \sqrt{\pi}\, e^{-\pi^2 t^2}.$$

We have

$$f'(t) = -2\pi^{\frac{5}{2}} t e^{-\pi^2 t^2}$$

$$f''(t) = 4\pi^{\frac{9}{2}} t^2 e^{-\pi^2 t^2} - 2\pi^{\frac{5}{2}} e^{-\pi^2 t^2} ,$$

so that

$$g(t) = \sqrt{\pi}(\frac{1}{2} - \pi^2 t^2) e^{-\pi^2 t^2} .$$

Hence (skipping some of the algebra)

$$f_1(s,t) = \pi e^{1-\pi^2(s^2 + t^2)}(1 - \pi^2(s^2 + t^2))$$

$$= \pi h(\pi\sqrt{(s^2 + t^2)})$$

where

$$h(u) = e^{1-u^2}(1 - u^2) = k(1 - u^2) ,$$

$$k(v) = ve^v .$$

Recall (from §4)

$$F(x,y) = a((x/b)^2 + (y/b)^2)e^{1-((x/b)^2 + (y/b)^2)} + e^{-((x/b)^2 + (y/b)^2)}$$

$$= F_{a,\ 1/b}(x,y) + f(x/b)\ f\ (y/b) .$$

Hence (combining our results above)

$$f(x,t) = \int\int F(x,y)e^{-2\pi i(xs + yt)}dxdy$$

$$= F_{a,1/b}(s,t) + b^2 \ f\ (bs)\ f\ (bt)$$

$$= ab^2 \ f_1 \ (bs,\ bt) + b^2 \ \pi \ e^{-\pi^2(s^2 + t^2)}$$

$$= ab^2 \ \pi \ (1 - \pi^2(s^2 + t^2))e^{1-\pi^2(s^2 + t^2)}$$

$$+ b^2 \ \pi \ e^{-\pi^2(s^2 + t^2)} \ .$$

In order to gain an understanding of this function (the filter) we plot $f(u,v)$ for $v = 0$, $0 \leq u \leq 20\mu$ with $a = 3$, $b = 80$, in Fig. 5.1. Since the grid we are sampling over is spaced about $4\mu \times 4\mu$, we anticipate difficulty properly sampling this function; this and related problems are discussed in the next section.

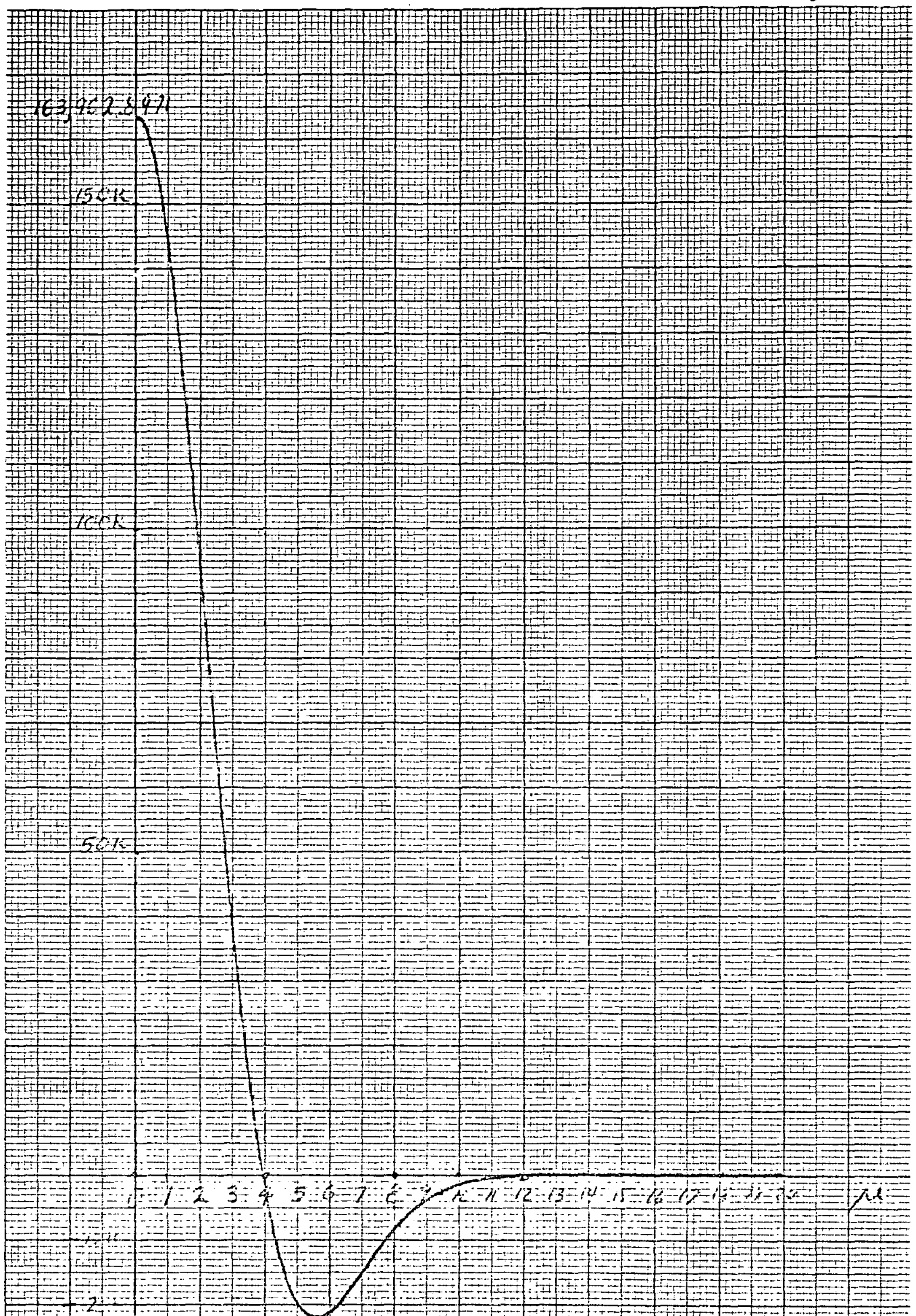


Fig. 5.1. Real Domain Cross-section of Filter, a = 3. b = 80.

6. NUMERICAL CONSTRUCTION OF FILTERS

This section contains a description of the mathematical background which supports the documentation of §7′(MKF3).

Refer to Fig. 5.1, a filter to be applied (mathematically) as a convolution extending over all the plane. Clearly, the filter is negligible (even when integrated) for distances $\geq$ 12μ from the center. Thus a 7x7 filter would seem to be lavish (with edges at least 14μ from the center), and a 3x3 may even be acceptable. The main problem is sampling such a wildly varying function adequately. We resolve this problem in the filter construction program by numerically integrating the function over a mesh centered between the DX, DY lattice points (DX = 0.003605, DY = 0.00405), using a simple unweighted average for the contribution associated with each mesh rectangle. (This method has great stability, and computer time is not an important consideration since the job need be done only once.) More precisely, we fill an array F as follows:

$$F(I,J) = \sum_{k=-10}^{10} \sum_{j=-10}^{10} f(DX(I + k/20),\ DY(J + j/20)),$$

I,J = −10, •••, 10. This gives an array F containing an unnormalized filter. Subarrays, centered at (0,0) and with dimensions varying from 3x3 to 21x21, are sampled,and normalized (to have sum 1), and are printed. For example, in the 3x3 case, we take

$$G_1(I, J) = F(I, J), \quad I, J = -1, \cdots, 1,$$

$$S = \sum_{k=-1}^{1} \sum_{j=-1}^{1} G_1(I, J),$$

$$G(I, J) = G_1(I, J)/S \; .$$

The array G is the normalized filter. (Owing to habit (induced by FORTRAN's reluctance to use negative subscripts), the printed filters are indexed differently.)

COMPUTER PROGRAM DOCUMENTATION

Program MKF3

Project B

by

Jack Bryant

and

R. L. Wendt

Prepared by

Applied Scientific Research, Inc.

Houston, Texas

Under Contract NAS 9-10577

For

MAPPING SCIENCES BRANCH

National Aeronautics and Space Administration

Manned Spacecraft Center

Houston, Texas

November, 1971

7'. COMPUTER PROGRAM DOCUMENTATION - MKF3

7'.1. PROGRAM DESCRIPTION

7'.1.1. GENERAL DESCRIPTION

This program makes filters to be used as input data cards by the filter program. The program is a straight forward application of the theory described in the mathematical documentation. The program is self contained except the constants A,B,C are input by the user and are also described in the mathematical documentation. Given constants A,B,C the program calculates all filters of orders from 3x3 to 11x11.

7'.1.2. MATHEMATICAL DESCRIPTION

A detailed description of the mathematical purpose and method of the program is given in §§5 and 6. The constants A,B and C enter into the calculation as parameters in the formula

$$F(X,Y) = A\ B^2\ \pi(1-B^2\pi^2(X^2+Y^2))e^{1-B^2\pi^2(X^2+Y^2)}+C^2\pi e^{-C^2\pi(X^2+Y^2)}.$$

In §§5 and 6, A=a and B=C=b.

7'.2. USAGE

7'.2.1. DESCRIPTION OF INPUT

The constants A,B,C are read from data cards in FORMAT (3F7.2).

7'.2.2. DESCRIPTION OF OUTPUT

Printed output: Approximately 184 pages of output are furnished. The filters are printed by a modified version of subroutine PRINTM, one filter to a page for the small filters, with the filters having 9 or 11 columns requiring two pages.

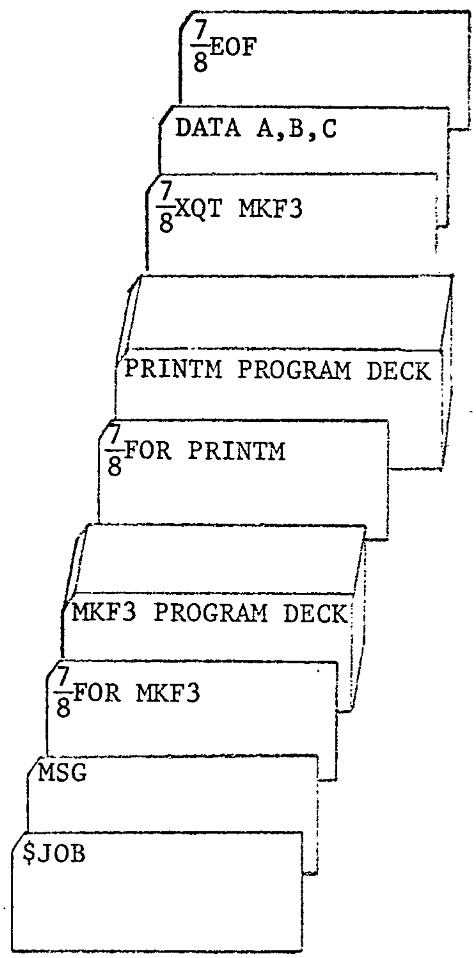


Fig. 7.2.3.1 CARD DECK SETUP (UNIVAC 1108) — MKF3

C5

Card output: Logic is included to punch all 7x7 filters, which is easily changed to meet other requirements. A 7x7 filter will result in 7 cards output for each input set.

7'.2.3. RUN PREPARATION

Card deck setup: See Fig. 7'.2.3.1.

Subroutine requirements: PRINTM

7'.2.4. EXECUTION CHARACTERISTICS

Restrictions: None.

Storage requirements: Code, 5370_8. Arrays, 3036_8.

Run time: 30 seconds per set of input parameters.

Lines output: 2000 lines per set of input parameters.

Accuracy/Validity: Perfect!

7'.3. REFERENCE INFORMATION

7'.3.1. DETAILED FLOW CHART - See Fig. 7'.3.1.1.

7'.3.2. DESCRIPTION OF VARIABLES

F - Array in which values of the filters are stored.

G - Array which has normalized values of F.

DX,DY - The distance in millimeters between elements of the lunar orbiter data.

X,Y - Points at which the filter is calculated.

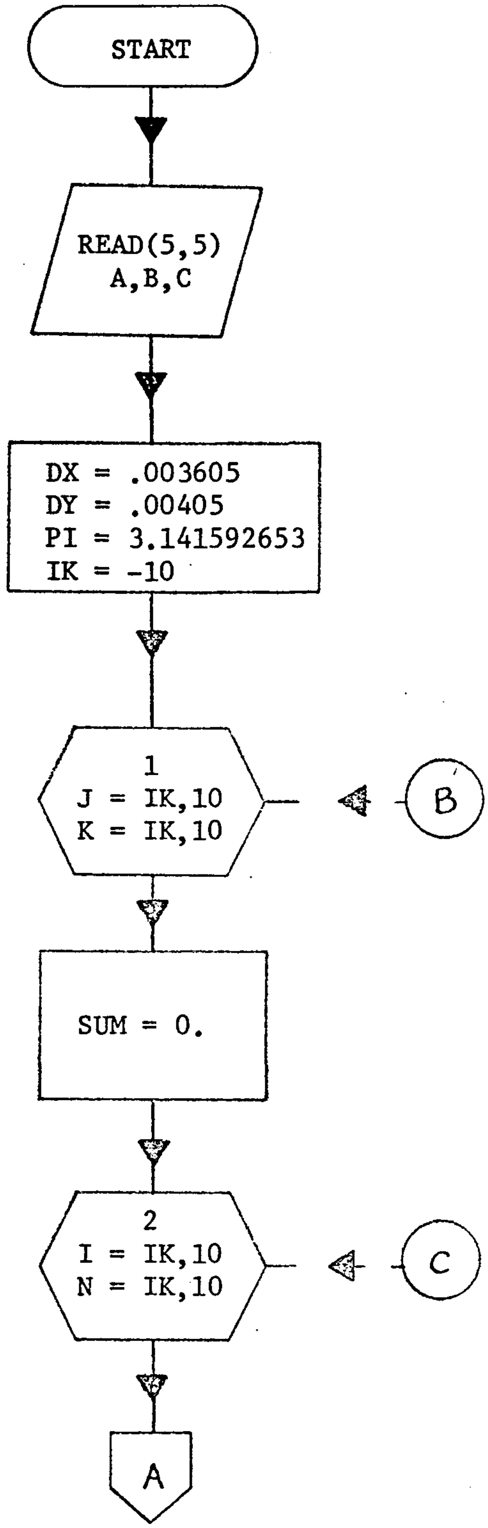


Fig. 7.3.1.1. DETAILED FLOW CHART - MKF3

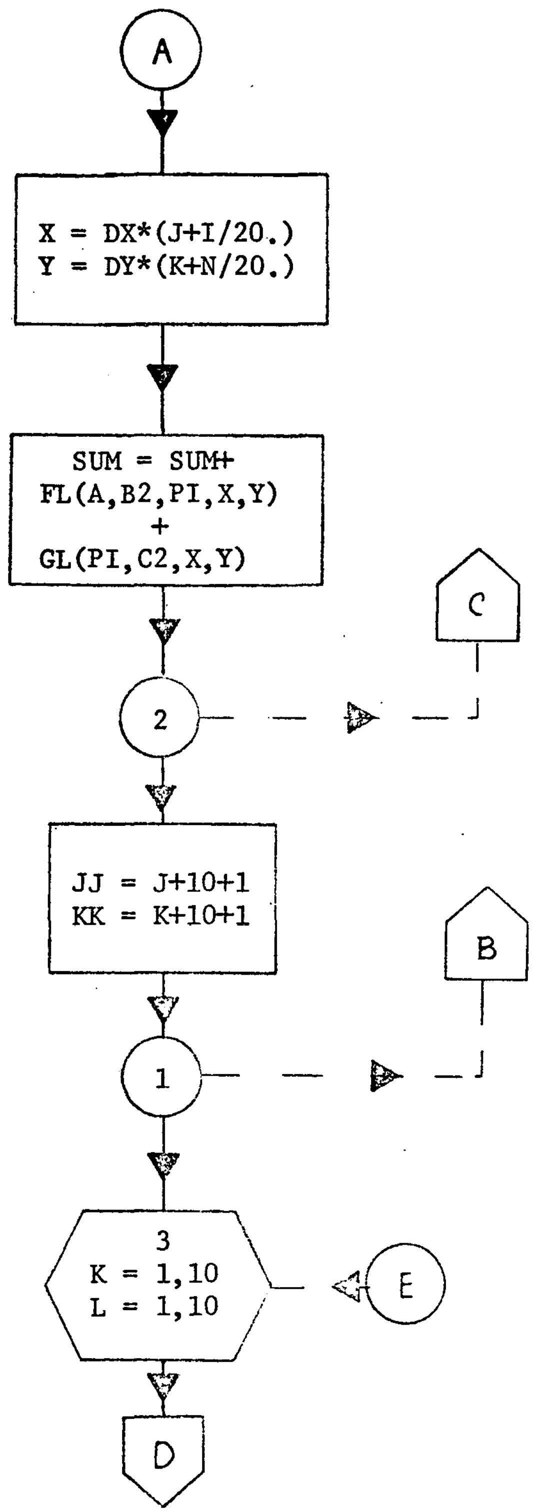


Fig. 7.3.1.1. DETAILED FLOW CHART - MKF3 (Continued)

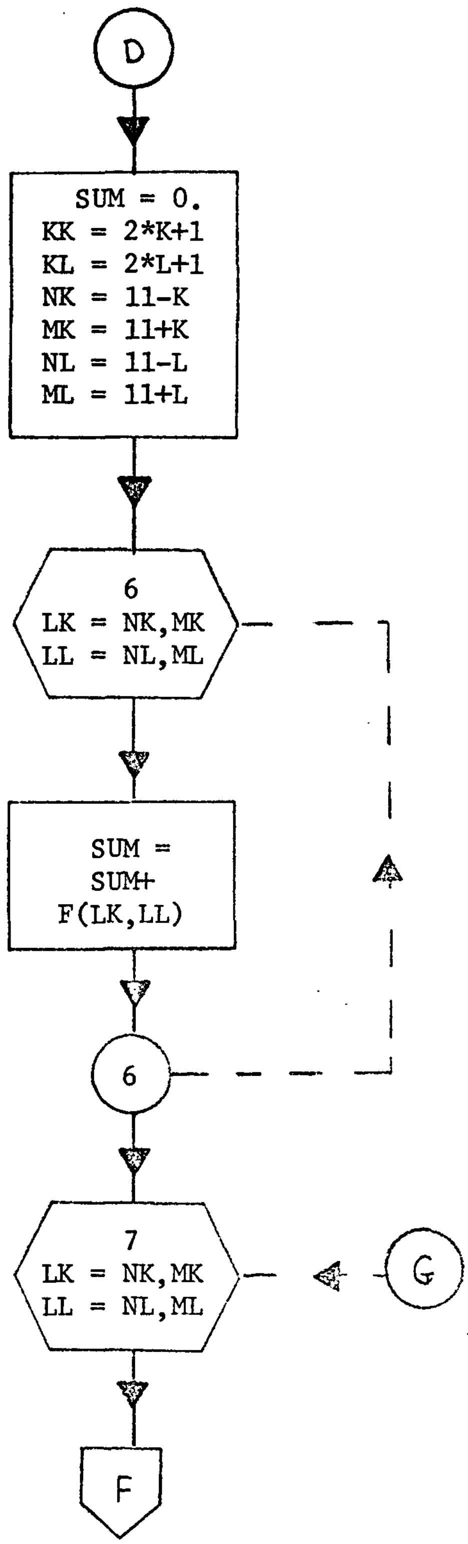


Fig. 7.3.1.1. DETAILED FLOW CHART - MKF3 (Continued)

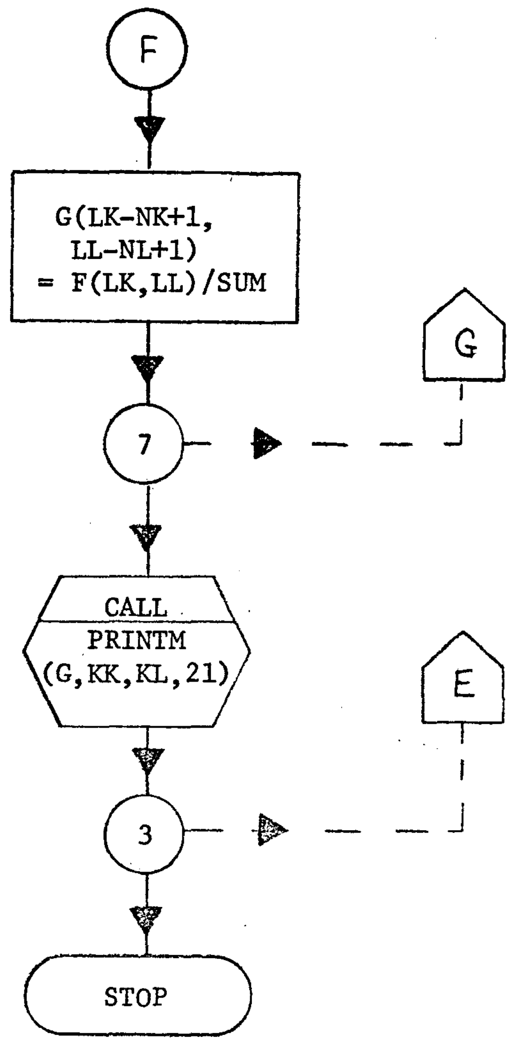


Fig. 7.3.1.1. DETAILED FLOW CHART - MKF3 (Concluded)

7. RESULTS OF APPLYING FILTERS TO MOON SCENE

The filters developed above were applied to part of Framelet 025. This segment, containing 250 lines only, contains a Reseau mark. The photographs we obtained from JPL/IPL were apparently intended for projection; that is, normal printing results in a negative. This is just as well, for the noise (jitter) is more evident in a negative near the Resau mark.

There are 19 photographs in this section. Table 7.1 contains the values of the parameters for all the pictures as well as the size of the filters involved. Then, in Fig. 7.2 and in Fig. 7.37, we display an original of the area of interest. There follow pictures and filters in that order - that is Fig. 7. (2n + 1) will be a picture, Fig. 7. (2n + 2) the filter associated with Fig. 7. (2n + 1), n = 1, ..., 17. All the pictures have been corrected for scan line signature and have had drummarks removed.

Photo	Filter	a	b	Size
7.3	7.4	4.00	103	7x7
7.5	7.6	4.00	103	3x3
7.7	7.8	3.60	93	7x7
7.9	7.10	3.60	93	3x3
7.11	7.12	2.80	80	7x7
7.13	7.14	2.80	80	3x3
7.15	7.16	2.25	69	7x7
7.17	7.18	2.25	69	3x3
7.19	7.20	1.85	60	7x7
7.21	7.22	1.85	60	3x3
7.23	7.24	1.66	52	7x7
7.25	7.26	1.66	52	5x5
7.27	7.28	1.49	45	7x7
7.29	7.30	1.49	45	5x5
7.31	7.32	1.39	39	7x7
7.33	7.34	1.31	33	7x7
7.35	7.36	1.25	29	7x7

Table 7.1. VALUES OF PARAMETERS IN FIGS. 7.3 - 7.36.

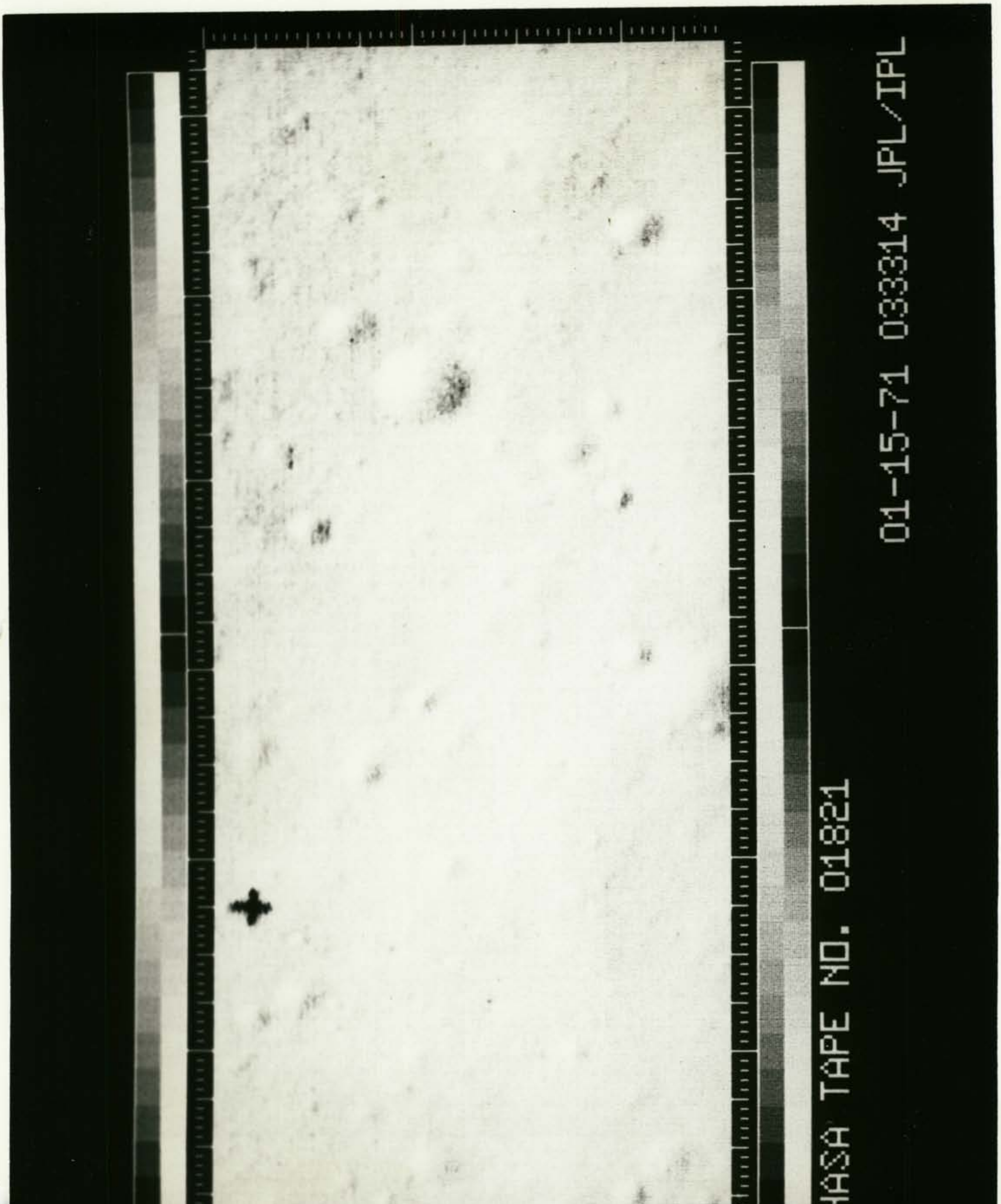


Fig. 7.2 Original (Unfiltered) Area, #01821, 033314.

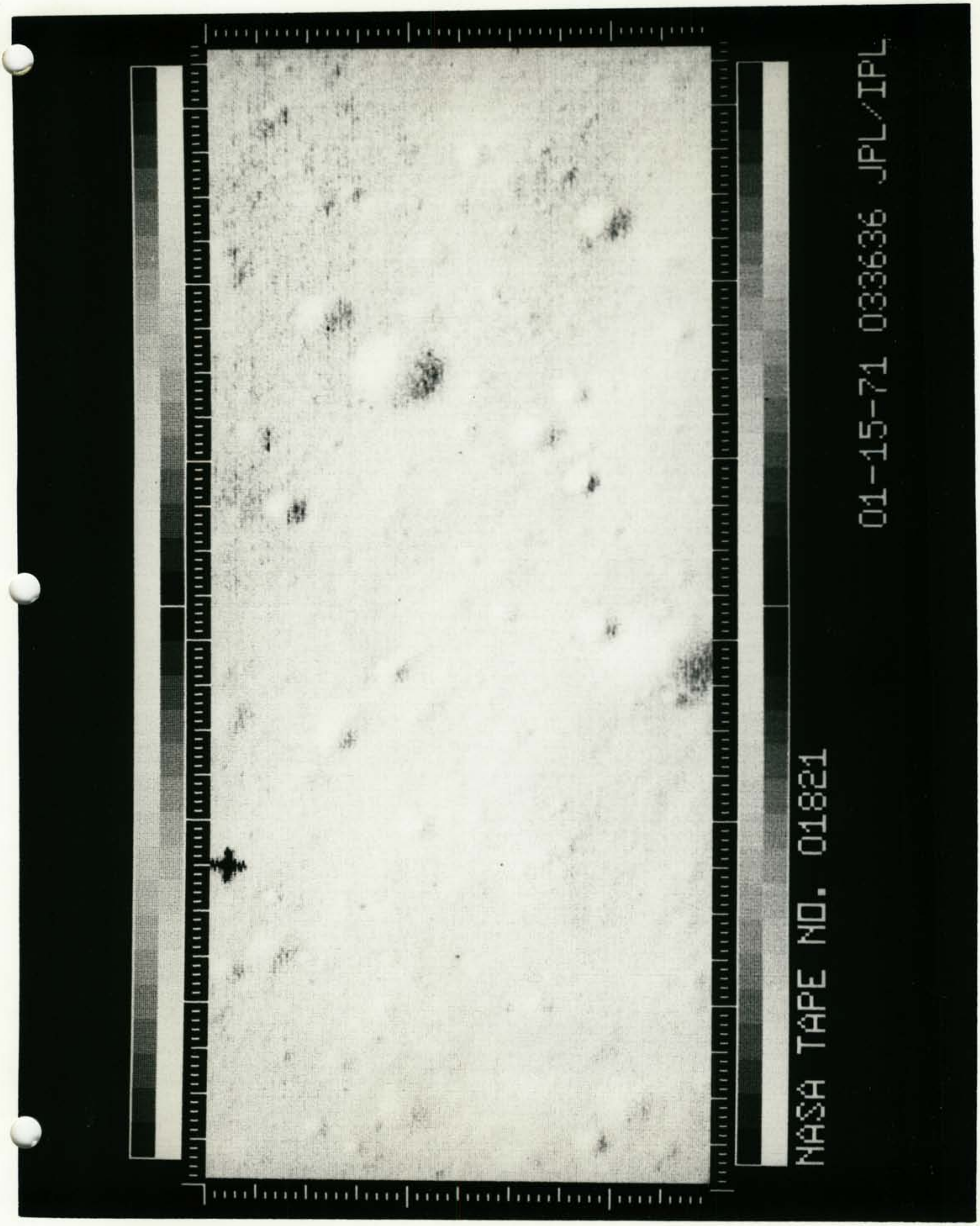


Fig. 7.3 Filtered a = 4.0, b = 103, 7x7, #01821, 033636.

	COL 1	COL 2	COL 3	COL 4	COL 5	COL 6	COL 7
ROW 1	.00004354	.00052729	.00222841	.00345978	.00222841	.00052729	.00004354
ROW 2	.00032474	.00341008	.00021519	-.01845520	.00021519	.00341008	.00032474
ROW 3	.00106644	.00361318	-.06978295	.05822838	-.06978295	.00361318	.00106644
ROW 4	.00157040	-.00317164	.01621468	1.11772366	.01621467	-.00317164	.00157040
ROW 5	.00106644	.00361318	-.06978295	.05822833	-.06978295	.00361318	.00106644
ROW 6	.00032474	.00341008	.00021519	-.01845520	.00021519	.00341008	.00032474
ROW 7	.00004354	.00052729	.00222841	.00345978	.00222841	.00052729	.00004354

Fig. 7.4 Filter for Fig. 7.3.

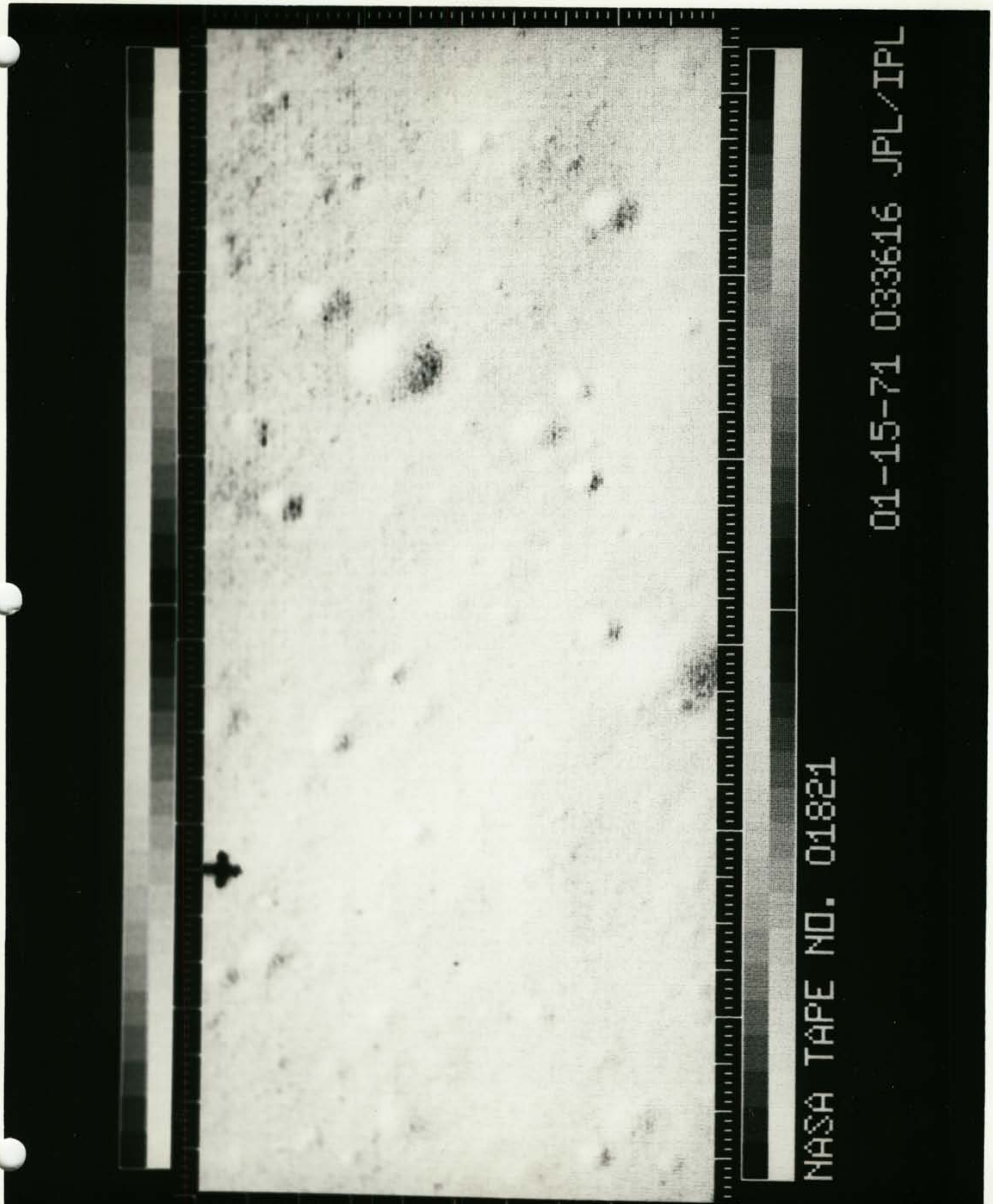


Fig. 7.5 Filtered a = 4.0, b = 103, 3x3, #01821, 033616

	COL 1	COL 2	COL 3
ROW 1	-.07066785	.05896677	-.07066785
ROW 2	.01642029	1.13189735	.01642029
ROW 3	-.07066786	.05896672	-.07066786

Fig. 7.6 Filter for Fig. 7.5.

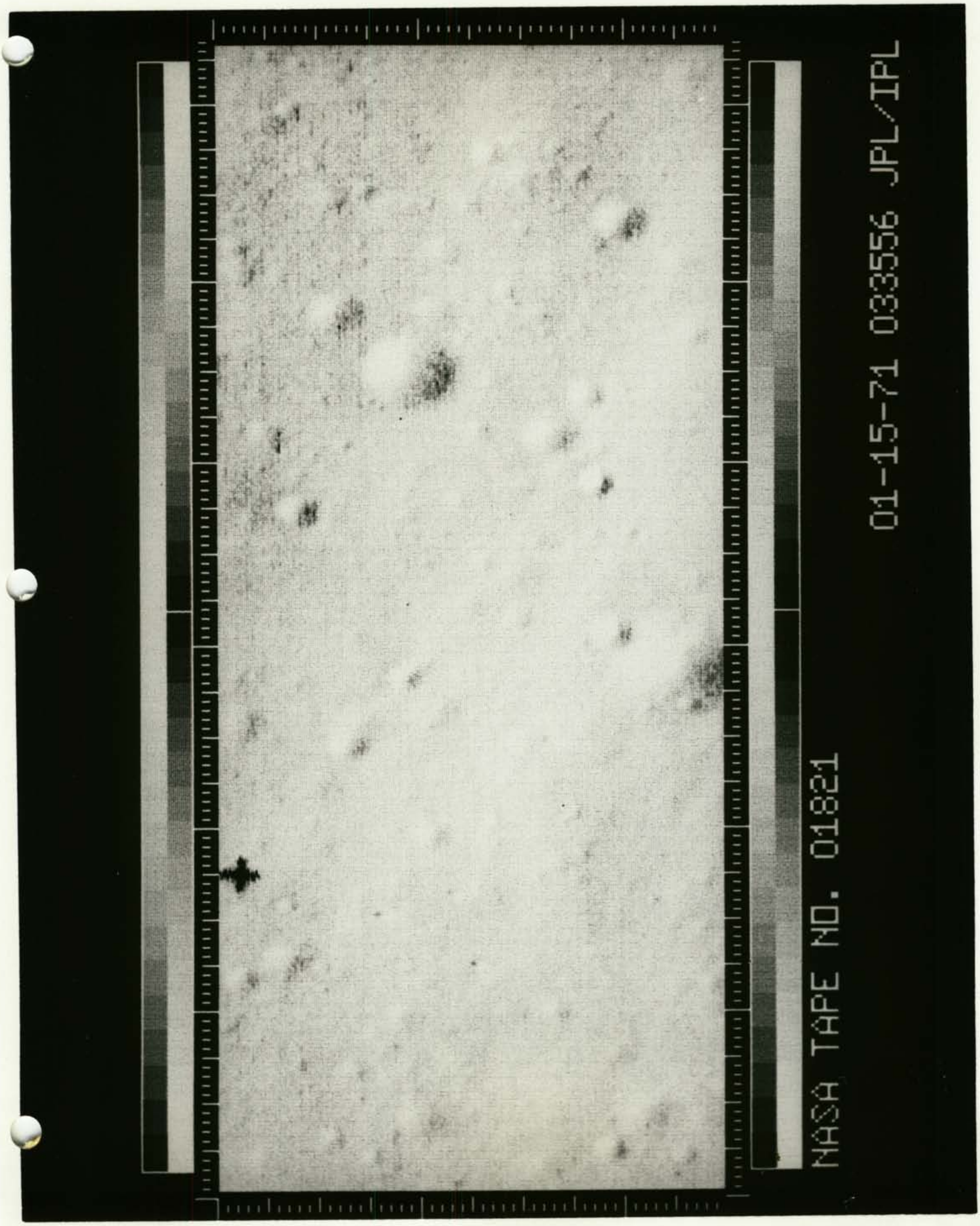


Fig. 7.7 Filtered a = 3.6, b = 93, 7x7, #01821, 033556.

	COL 1	COL 2	COL 3	COL 4
ROW 1	.00014599	.00112987	.00344834	.00460744
ROW 2	.00076031	.00440295	-.00575874	-.02719562
ROW 3	.00197171	-.00015938	-.05070085	.10610035
ROW 4	.00265202	-.01021709	.05974271	.90765967
ROW 5	.00197171	-.00015938	-.05070084	.10610024
ROW 6	.00076031	.00440295	-.00575874	-.02719563
ROW 7	.00014599	.00112987	.00344834	.00460744

	COL 5	COL 6	COL 7
ROW 1	.00344834	.00112987	.00014599
ROW 2	-.00575874	.00440295	.00076031
ROW 3	-.05070085	-.00015938	.00197171
ROW 4	.05974271	-.01021709	.00265202
ROW 5	-.05070084	-.00015938	.00197171
ROW 6	-.00575874	.00440295	.00076031
ROW 7	.00344834	.00112987	.00014599

Fig. 7.8 Filter for Fig. 7.7.

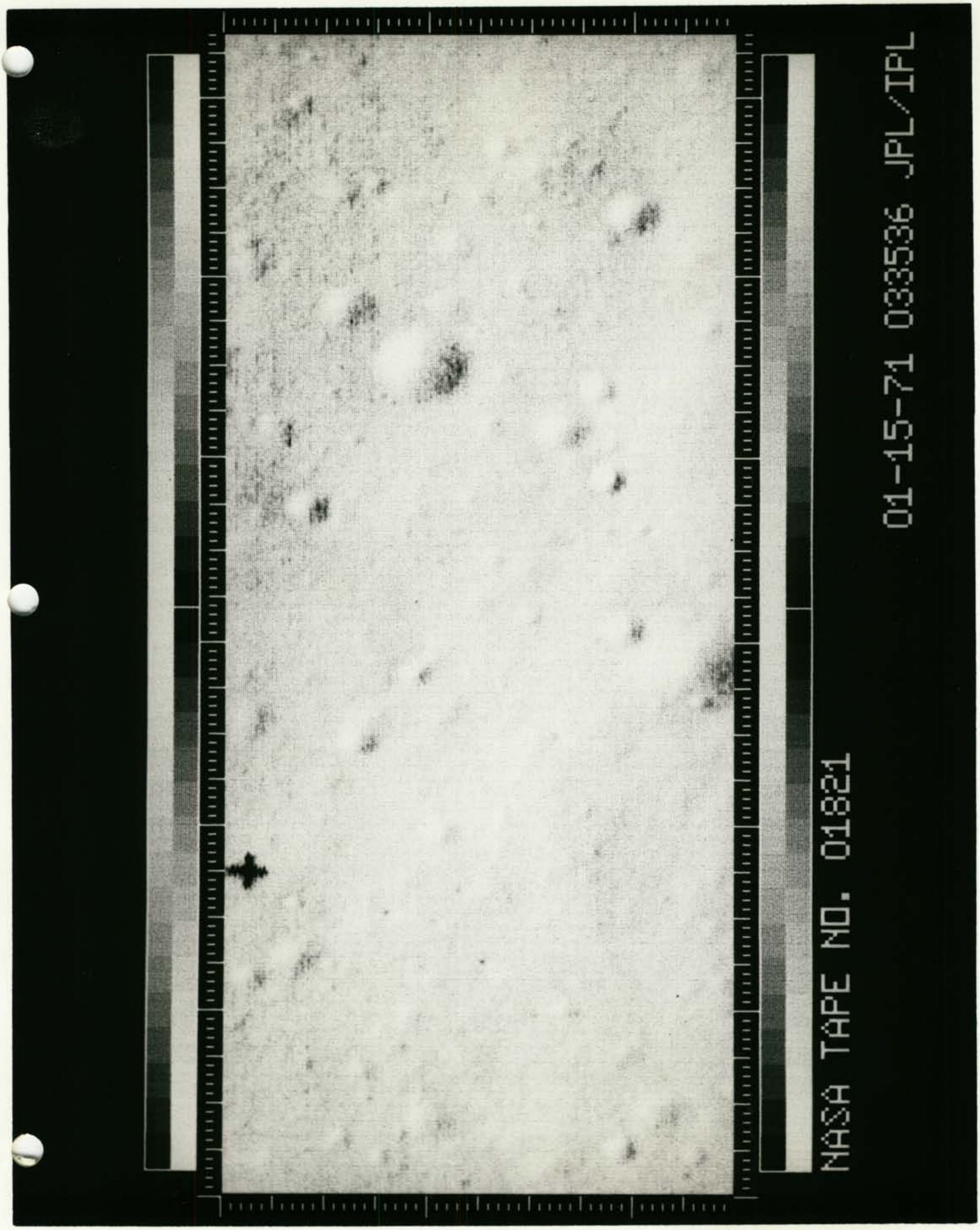


Fig. 7.9 Filtered a = 3.6, b = 93, 3x3, #01821, 033536.

	COL 1	COL 2	COL 3
ROW 1	-.04891344	.10235989	-.04891344
ROW 2	.05763654	.87566099	.05763654
ROW 3	-.04891343	.10235977	-.04891343

Fig. 7.10 Filter for Fig. 7.9.

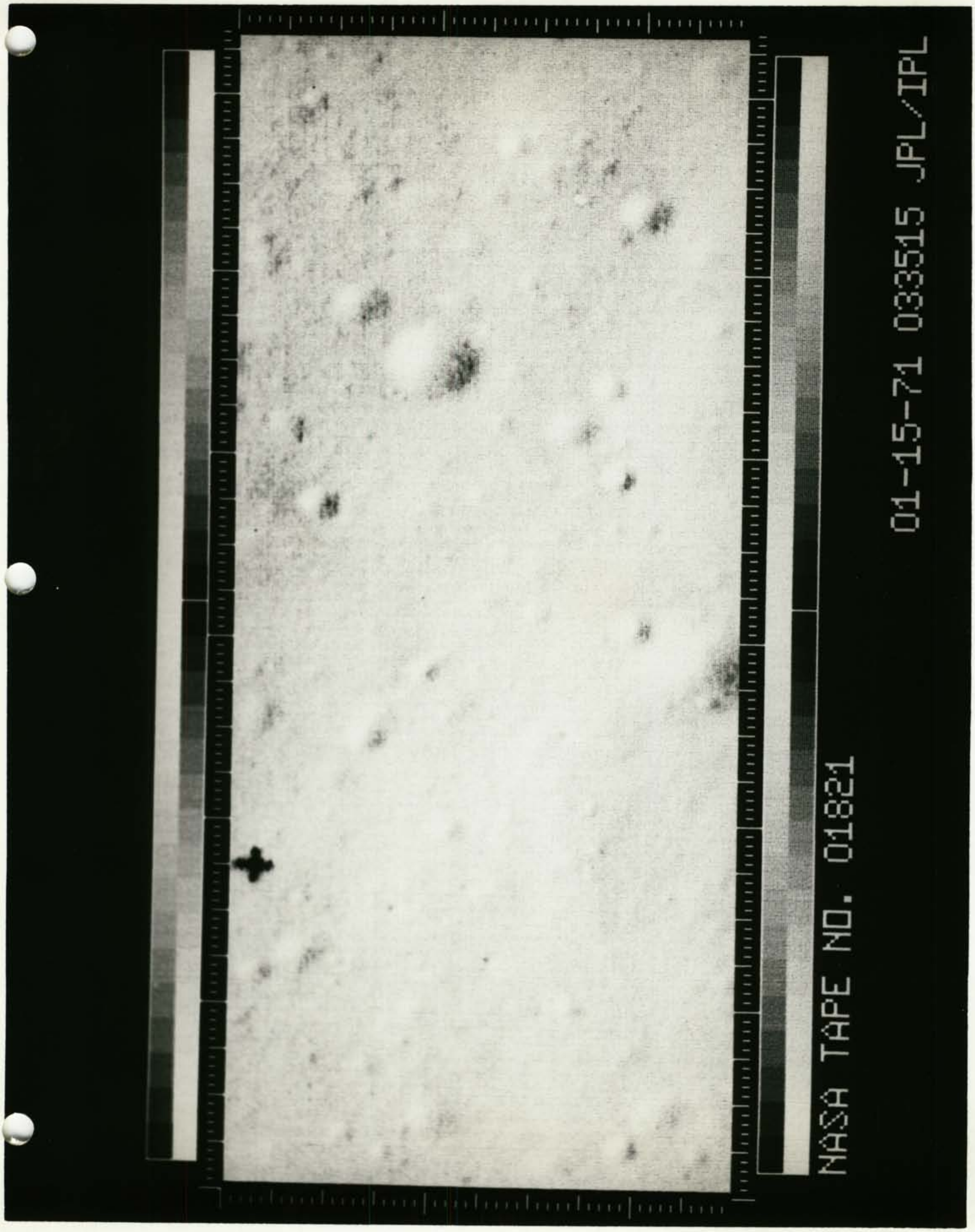


Fig. 7.11 Filtered a = 2.8, b = 80, 7x7, #01821, 033515.

	COL 1	COL 2	COL 3	COL 4	COL 5	COL 6	COL 7
ROW 1	.00057772	.00253016	.00474992	.00488713	.00474992	.00253016	.00057772
ROW 2	.00191934	.00457106	-.01113812	-.02564855	-.01113812	.00457106	.00191934
ROW 3	.00357365	-.00523636	-.00875264	.14169169	-.00875264	-.00523636	.00357365
ROW 4	.00418641	-.01445745	.10085722	.60578831	.10085721	-.01445745	.00418641
ROW 5	.00357365	-.00523636	-.00875263	-.14169160	-.00875263	-.00523636	.00357365
ROW 6	.00191934	.00457106	-.01113812	-.02564856	-.01113812	.00457106	.00191934
ROW 7	.00057772	.00253016	.00474992	.00488712	.00474992	.00253016	.00057772

Fig. 7.12 Filter for Fig. 7.11.

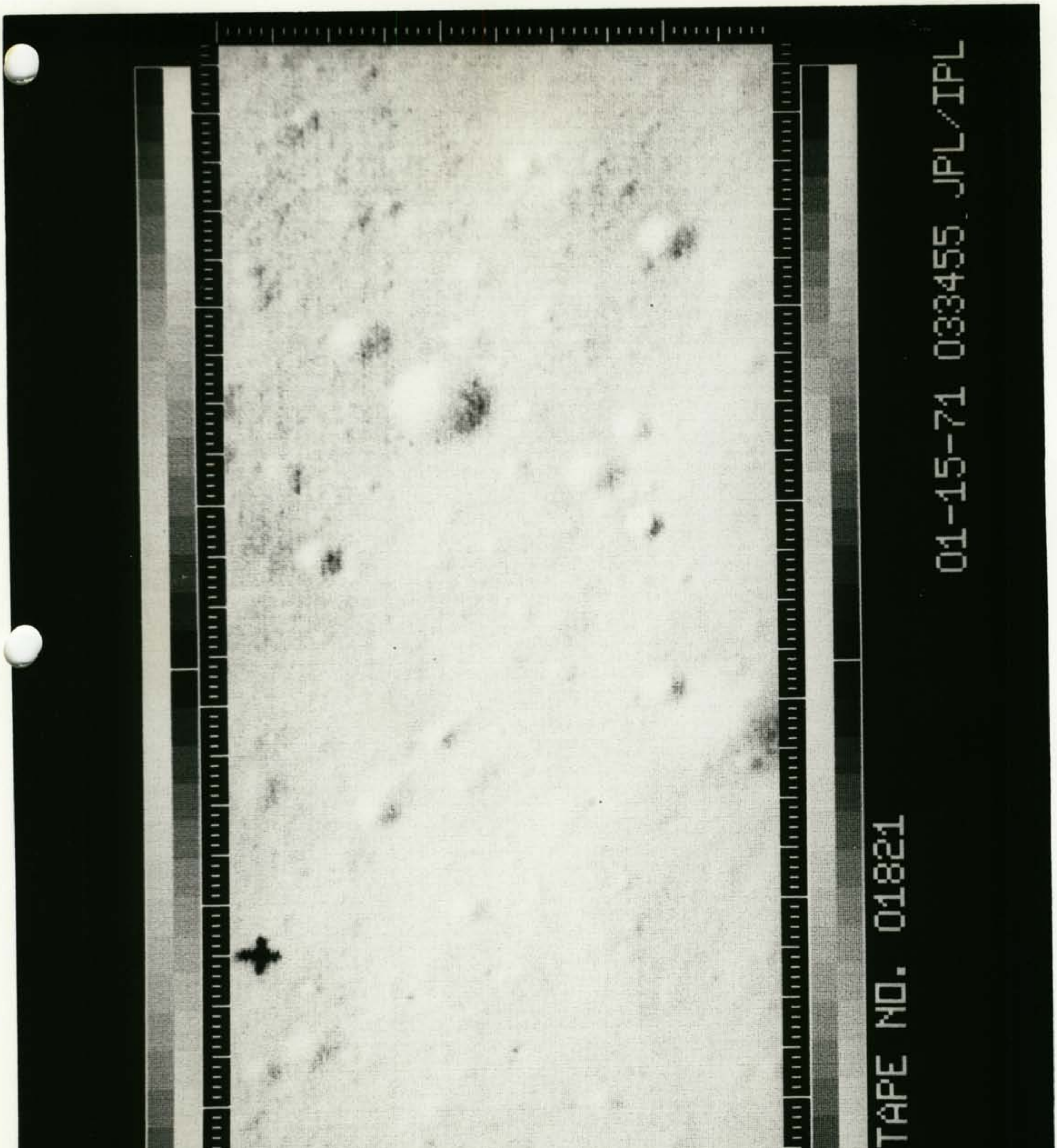


Fig. 7.13 Filtered a = 2.8, b = 80, 3x3, #01821, 033455.

	COL 1	COL 2	COL 3
ROW 1	-.00828946	.13419356	-.00828946
ROW 2	.09551999	.57373082	.09551999
ROW 3	-.00828946	.13419347	-.00828946

Fig. 7.14 Filter for Fig. 7.13.

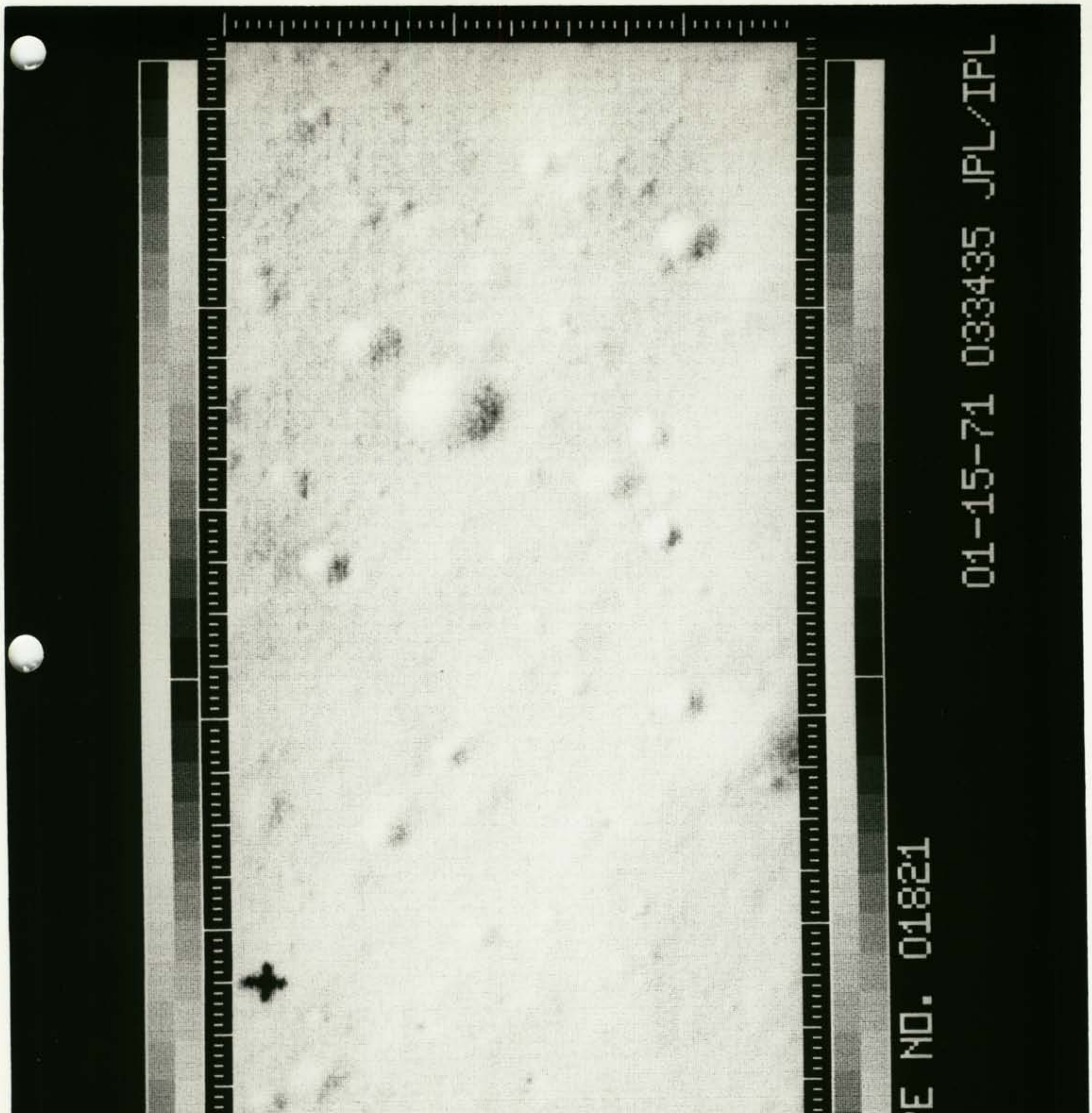


Fig. 7.15 Filtered a = 2.25, b = 69, 7x7, #01821, 033435.

	COL 1	COL 2	COL 3	COL 4	COL 5	COL 6	COL 7
ROW 1	.00150211	.00392169	.00423224	.00283430	.00423224	.00392169	.00150211
ROW 2	.00334852	.00263702	-.01145633	-.01504517	-.01145633	.00263702	.00334852
ROW 3	.00444514	-.00779241	.02425540	.14713940	.02425540	-.00779241	.00444514
ROW 4	.00445352	-.01240137	.11560139	.41446259	.11560138	-.01240137	.00445352
ROW 5	.00444514	-.00779241	.02425539	.14713928	.02425539	-.00779241	.00444514
ROW 6	.00334852	.00263702	-.01145633	-.01504517	-.01145633	.00263702	.00334852
ROW 7	.00150211	.00392169	.00423224	.00283430	.00423224	.00392169	.00150211

Fig. 7.16 Filter for Fig. 7.15.

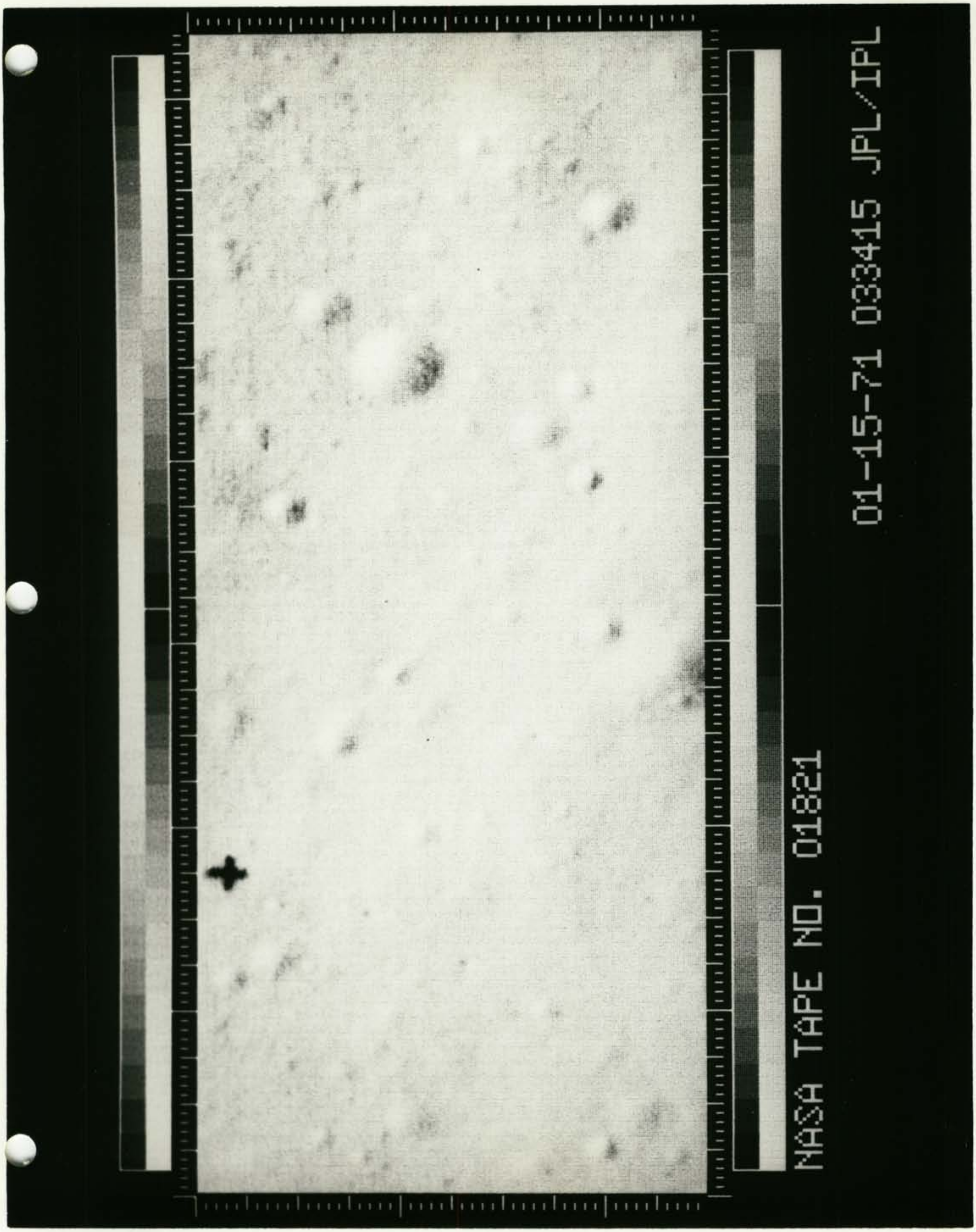


Fig. 7.17 Filtered a = 2.25, b = 69, 3x3, #01821, 033415.

	COL 1	COL 2	COL 3
ROW 1	.02339074	.14189420	.02339074
ROW 2	.11148045	.39968789	.11148044
ROW 3	.02339074	.14189409	.02339073

Fig. 7.18 Filter for Fig. 7.17.

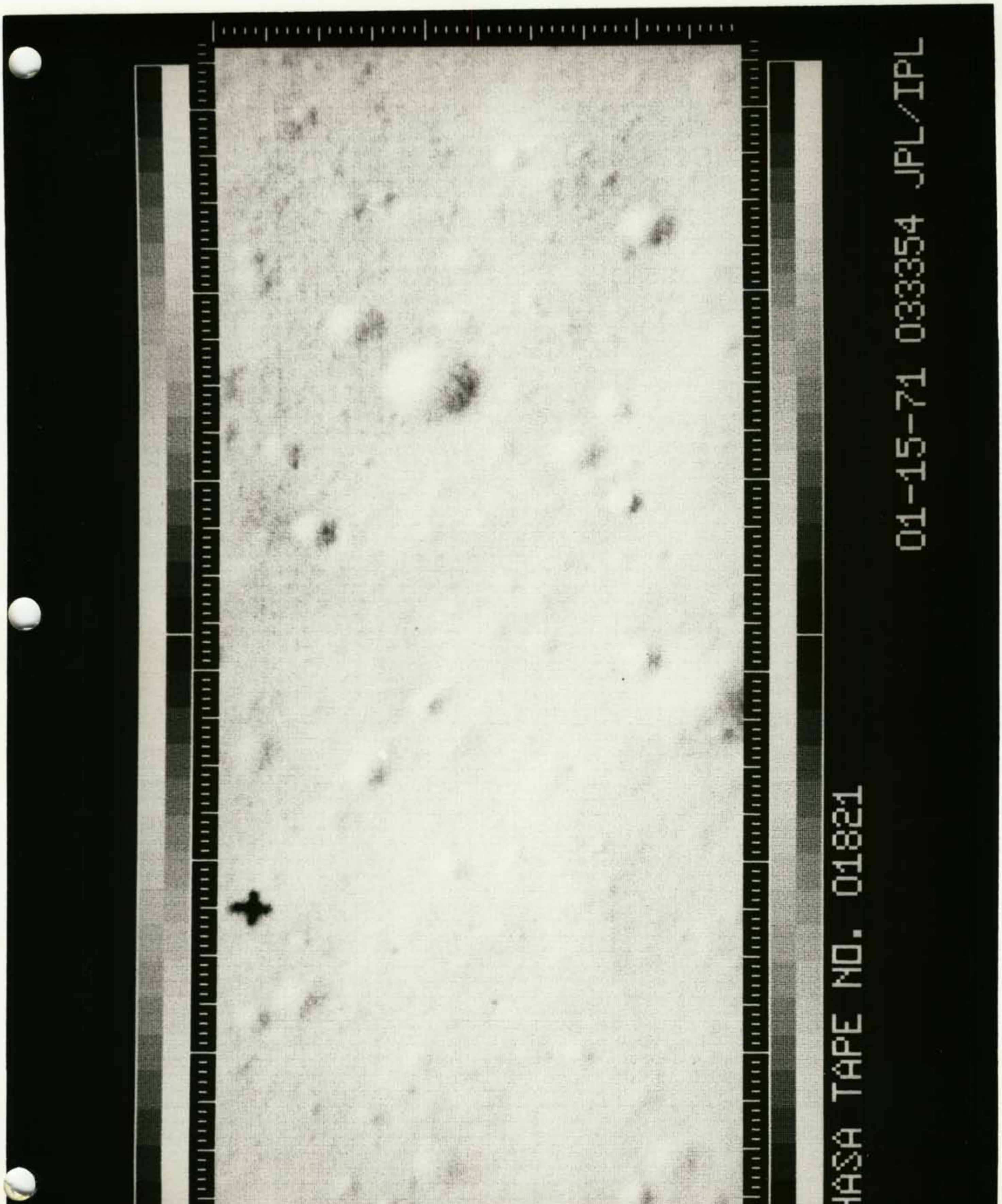


Fig. 7.19 Filtered a = 1.85, b = 60, 7x7, #01821, 033354.

	COL 1	COL 2	COL 3	COL 4
ROW 1	.00270182	.00432089	.00234154	.00031135
ROW 2	.00415906	.00032713	-.00573625	-.00016277
ROW 3	.00390073	-.00544436	.04528152	.13686382
ROW 4	.00325556	-.00469549	.11433506	.29277682
ROW 5	.00390073	-.00544436	.04528152	.13686367
ROW 6	.00415906	.00032713	-.00573625	-.00016277
ROW 7	.00270182	.00432089	.00234154	.00031135

	COL 5	COL 6	COL 7
ROW 1	.00234154	.00432089	.00270182
ROW 2	-.00573625	.00032713	.00415906
ROW 3	.04528152	-.00544436	.00390073
ROW 4	.11433506	-.00469549	.00325556
ROW 5	.04528151	-.00544436	.00390073
ROW 6	-.00573625	.00032713	.00415906
ROW 7	.00234154	.00432089	.00270182

Fig. 7.20 Filter for Fig. 7.19.

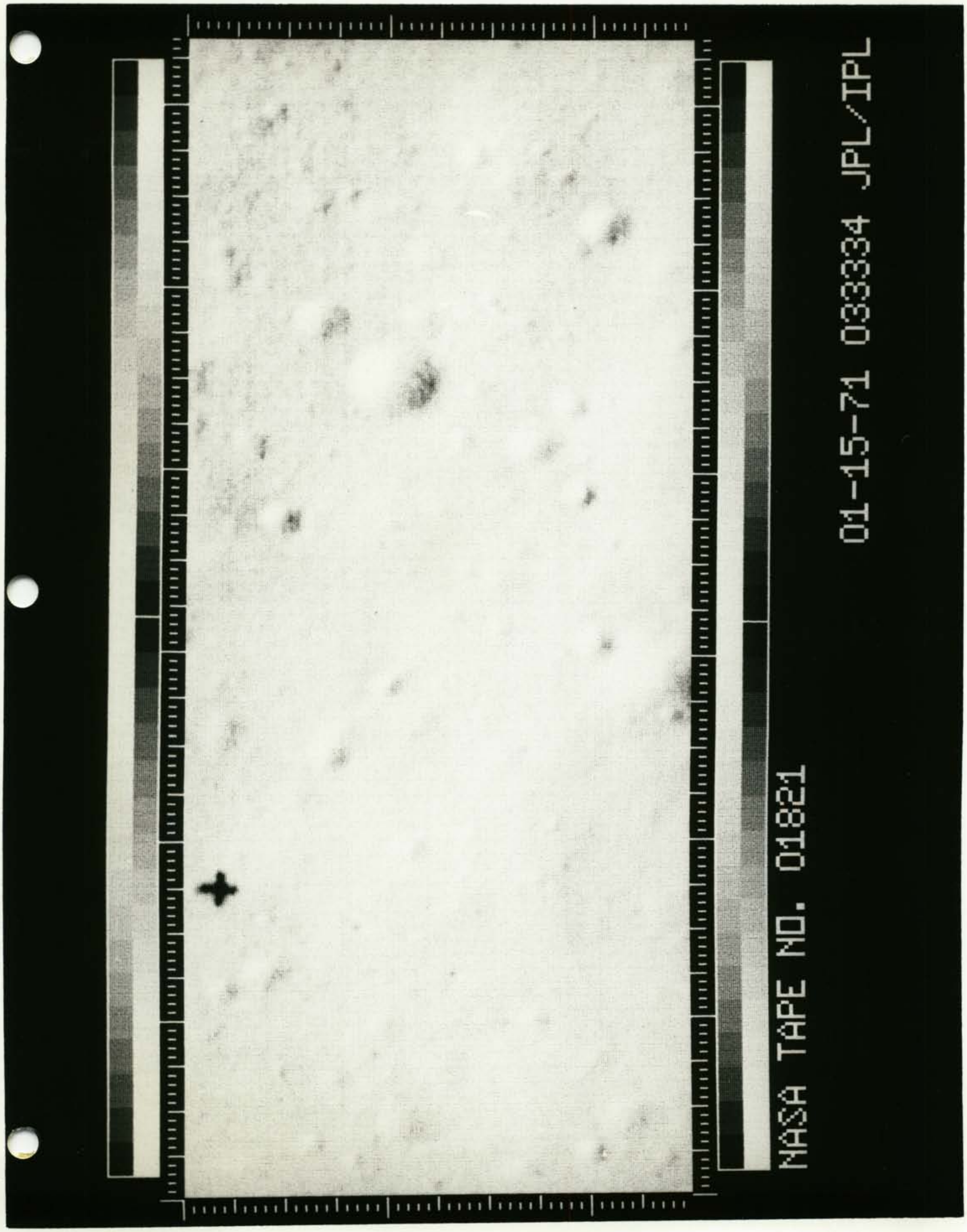


Fig. 7.21 Filtered a = 1.85, b = 60, 3x3, #01821, 033334.

	COL 1	COL 2	COL 3
ROW 1	.04638072	.14018617	.04638072
ROW 2	.11711052	.29988393	.11711052
ROW 3	.04638072	.14018601	.04638072

Fig. 7.22 Filter for Fig. 7.21.

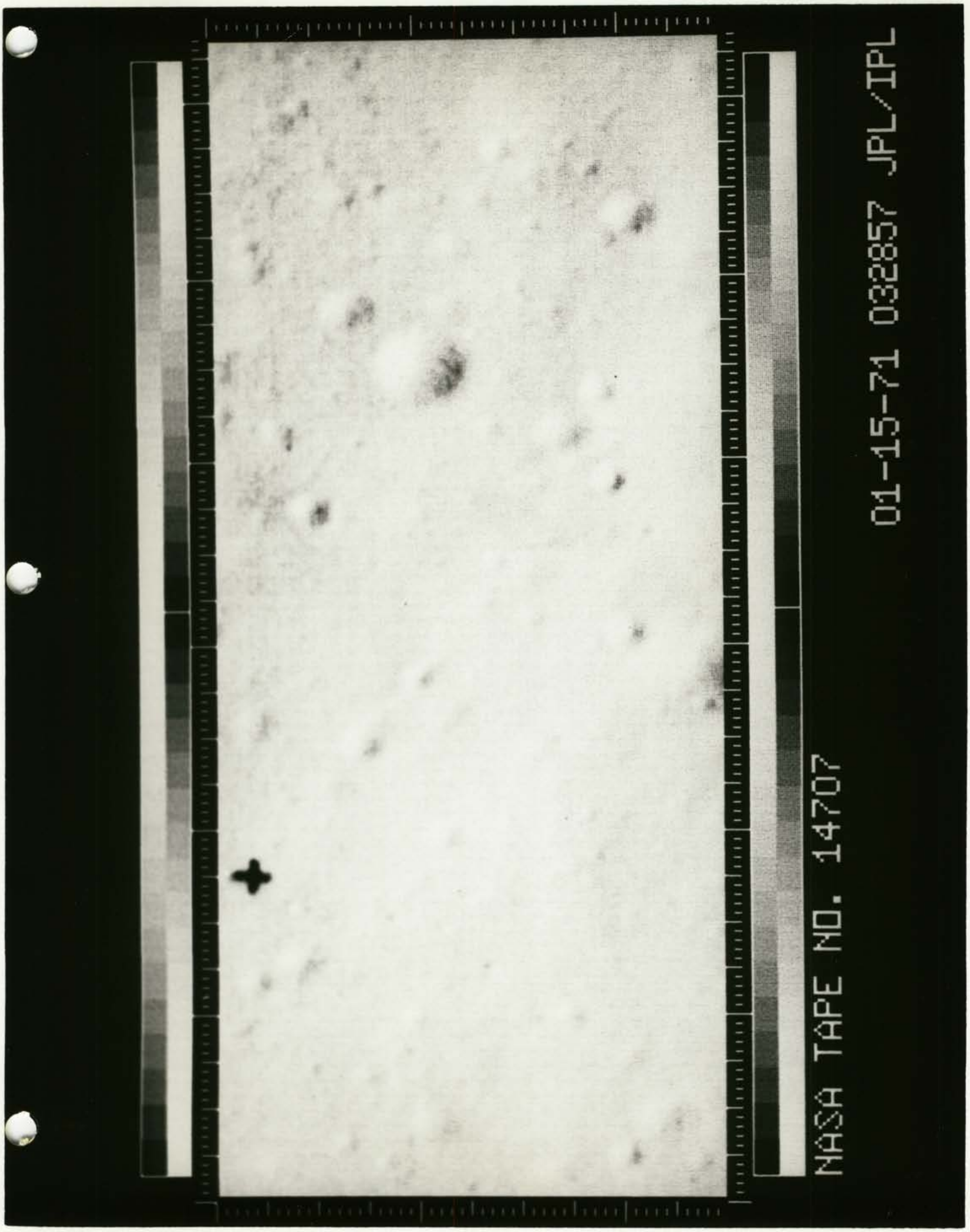


Fig. 7.23 Filtered a = 1.66, b = 52, 7x7, #14707, 032857.

	COL 1	COL 2	COL 3	COL 4
ROW 1	.00353643	.00302040	-.00031043	-.00184211
ROW 2	.00348212	-.00166518	.00210921	.01352619
ROW 3	.00178445	-.00065838	.05780098	.12533298
ROW 4	.00082064	.00433335	.10929832	.22066311
ROW 5	.00178445	-.00065838	.05780097	.12533284
ROW 6	.00348212	-.00166518	.00210921	.01352619
ROW 7	.00353643	.00302040	-.00031043	-.00184211

COL 5	COL 6	COL 7
-.00031043	.00302040	.00353643
.00210921	-.00166518	.00348212
.05780098	-.00065838	.00178445
.10929831	.00433335	.00082064
.05780097	-.00065838	.00178445
.00210921	-.00166518	.00348212
-.00031043	.00302040	.00353643

Fig. 7.24 Filter for Fig. 7.23.

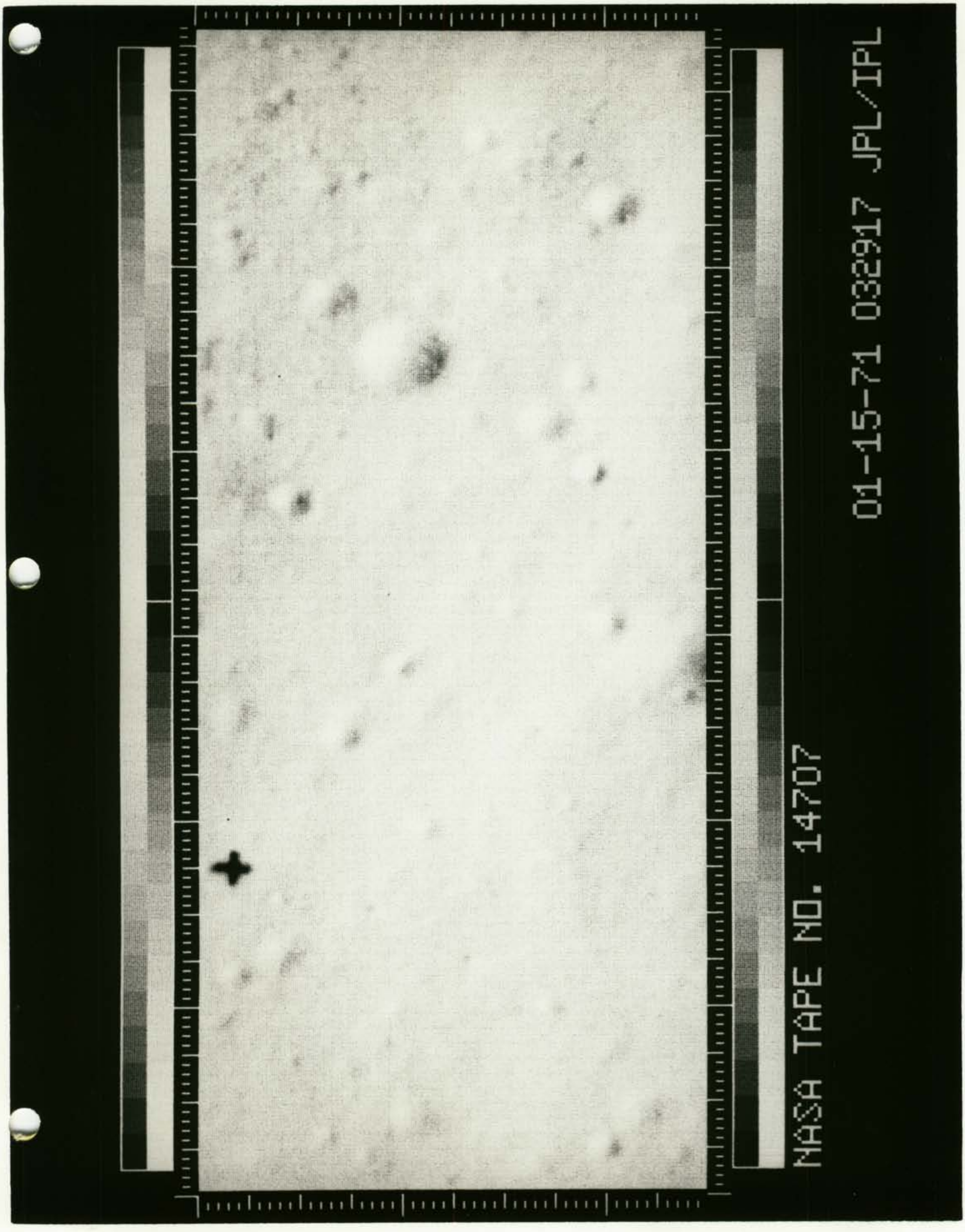


Fig. 7.25 Filtered a = 1.66, b = 52, 5x5, #14707, 032917.

	COL 1	COL 2	COL 3	COL 4	COL 5
ROW 1	-.00174184	.00220630	.01414886	.00220630	-.00174184
ROW 2	-.00068869	.06046184	.13110266	.06046184	-.00068869
ROW 3	.00453283	.11432965	.23082131	.11432984	.00453283
ROW 4	-.00068869	.06046183	.13110252	.06046183	-.00068869
ROW 5	-.00174184	.00220630	.01414887	.00220630	-.00174184

Fig. 7.26 Filter for Fig. 7.25.

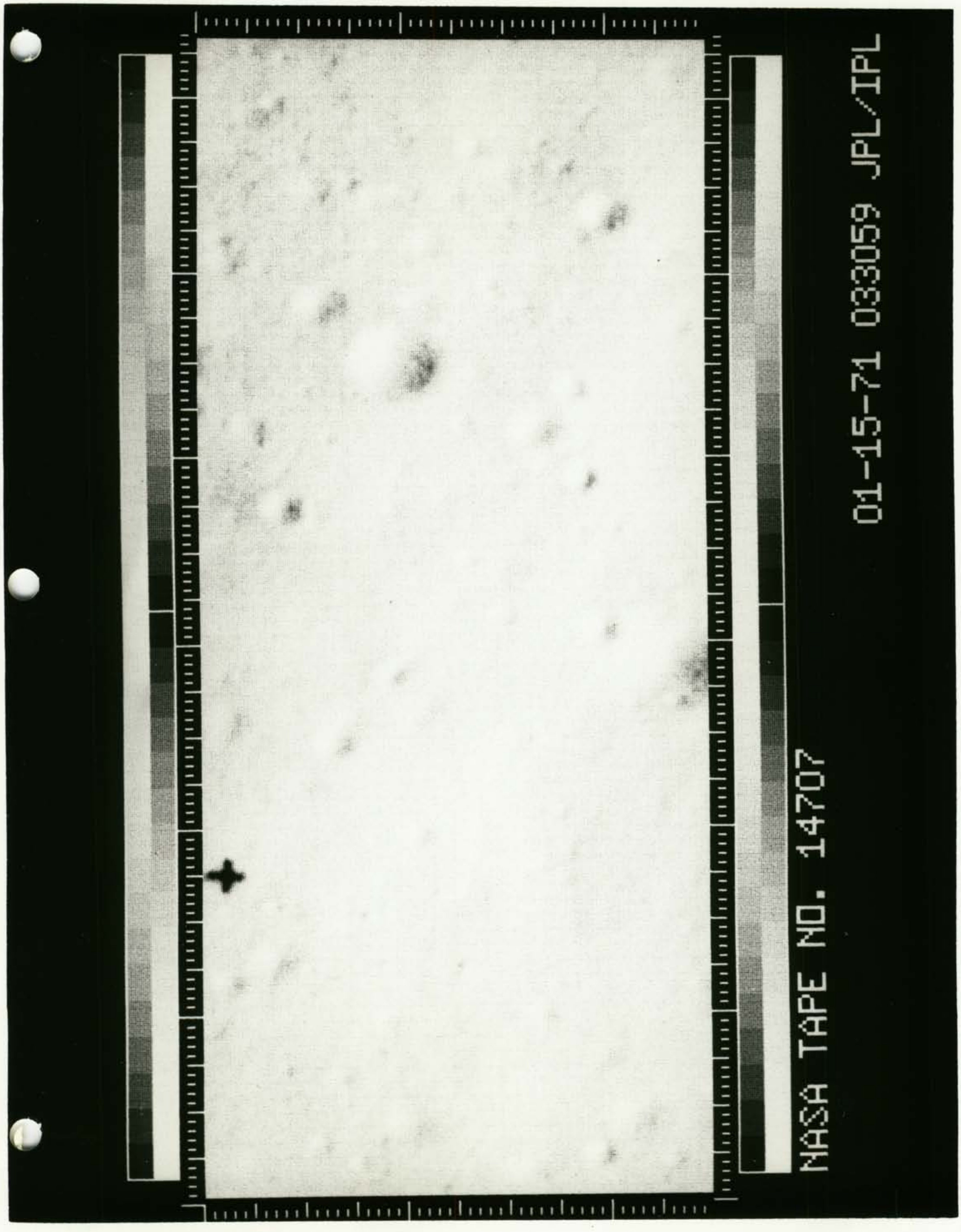


Fig. 7.27 Filtered a = 1.49, b = 45, 7x7, #14707, 033059.

	COL 1	COL 2	COL 3	COL 4	COL 5	COL 6	COL 7
ROW 1	.00322103	.00114372	-.00070837	-.10055669	-.00070837	.00114372	.00322103
ROW 2	.00180451	-.00034954	.01207127	.12533104	.01207127	-.00034954	.00180451
ROW 3	.00017551	.00737744	.06201503	.10795621	.06201503	.00737744	.00017551
ROW 4	-.00029796	.01460892	.09736551	.16414372	.09736551	.01460892	-.00029796
ROW 5	.00017551	.00737744	.06201500	.10795614	.06201499	.00737744	.00017551
ROW 6	.00180451	-.00034954	.01207128	.12535105	.01207128	-.00034954	.00180451
ROW 7	.00322103	.00114372	-.00070837	-.10055669	-.00070837	.00114372	.00322103

Fig. 7.28 Filter for Fig. 7.27.

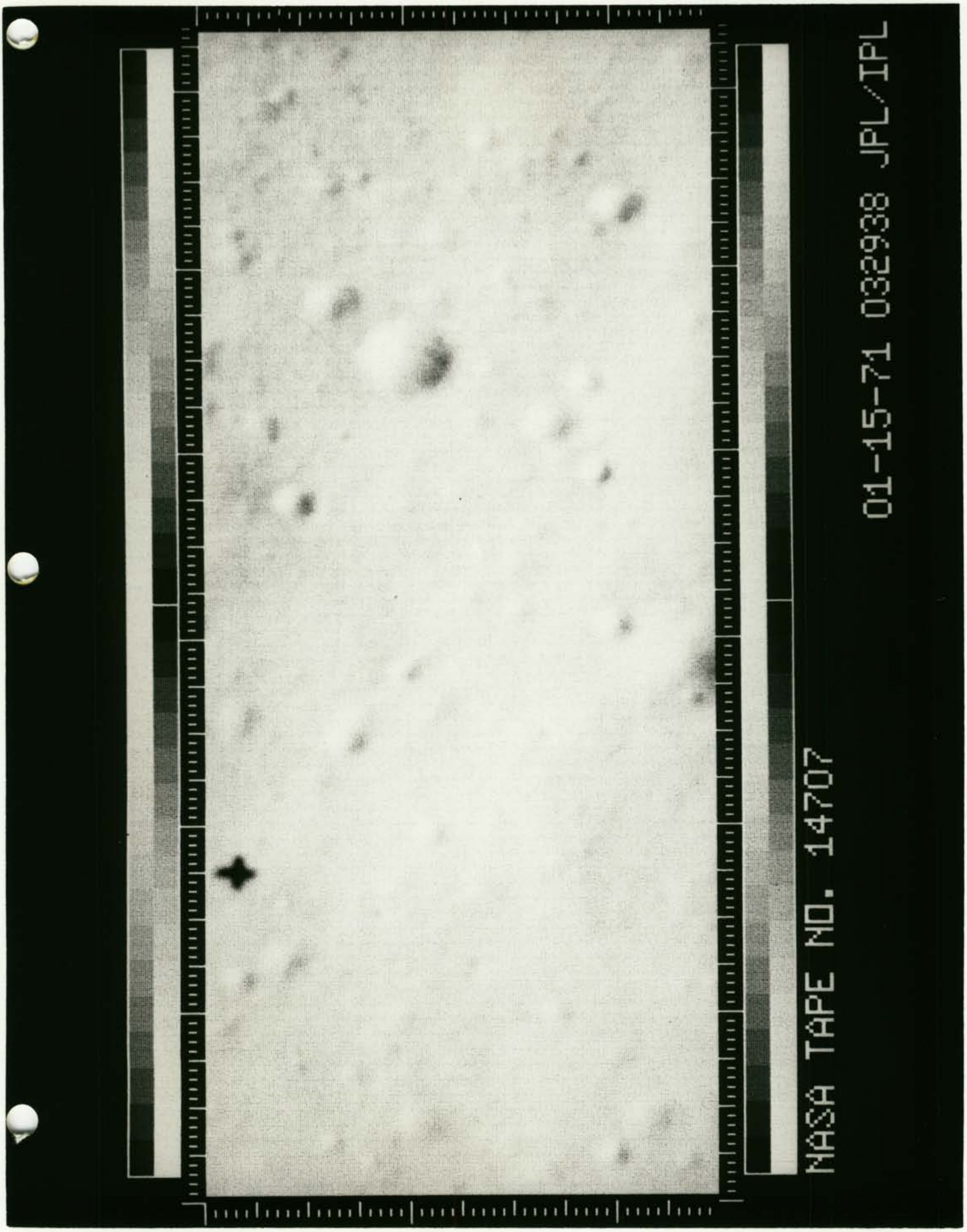


Fig. 7.29 Filtered a = 1.49, b = 45, 5x5, #14707, 032938.

	COL 1	COL 2	COL 3	COL 4	COL 5
ROW 1	-.00035698	.01232815	.02589051	.01232815	-.00035698
ROW 2	.00753443	.06333469	.11025349	.06333469	.00753443
ROW 3	.01491979	.09943742	.16763465	.09943742	.01491979
ROW 4	.00753443	.06333466	.11025342	.06333466	.00753443
ROW 5	-.00035698	.01232815	.02589051	.01232815	-.00035698

Fig. 7.30 Filter for Fig. 7.29.

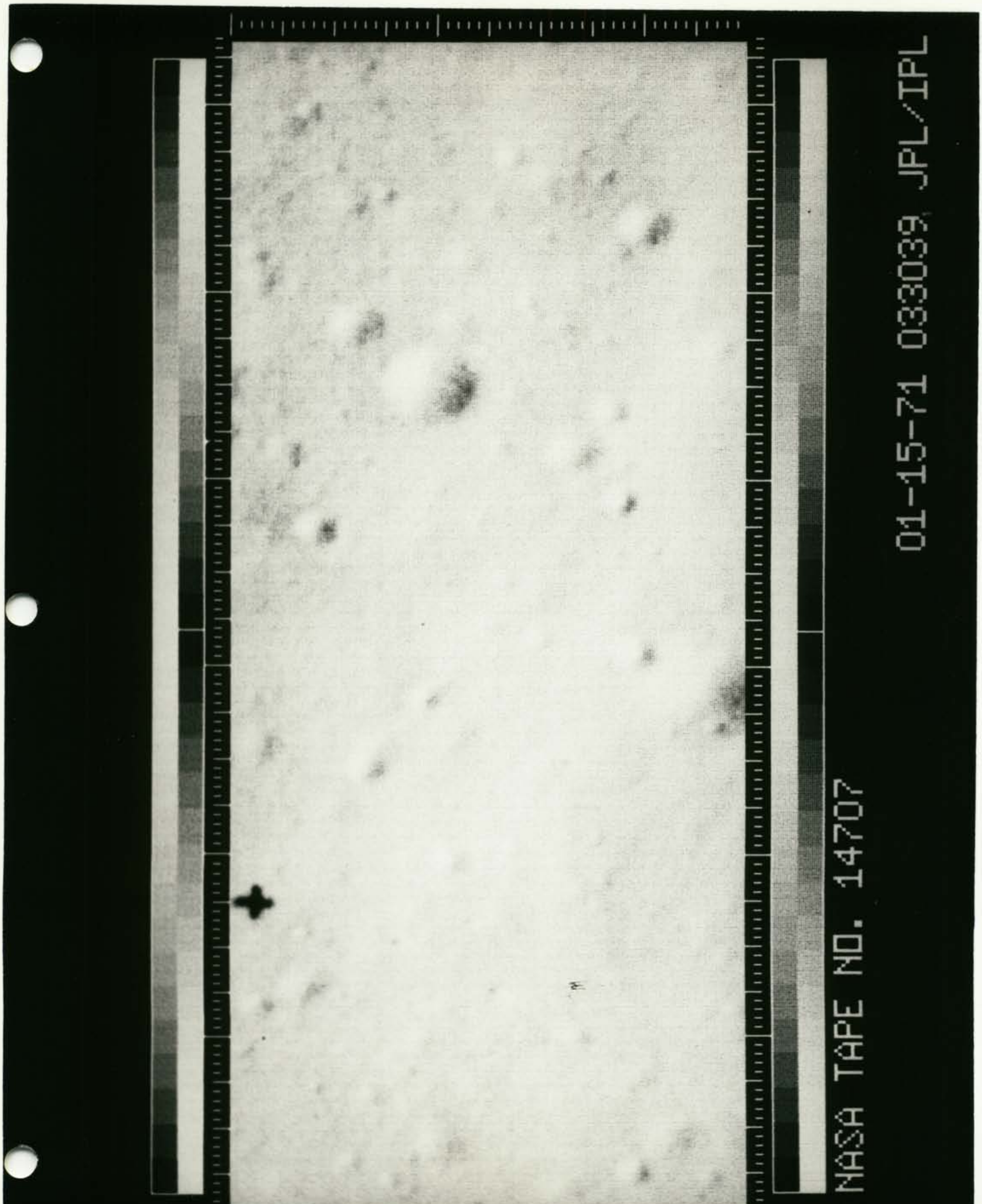


Fig. 7.31 Filtered a = 1.39, b = 39, 7x7, #14707, 033039.

	COL 1	COL 2	COL 3	COL 4
ROW 1	.00185055	.00020706	.00142444	.00325251
ROW 2	.00046770	.00339789	.02009164	.03237963
ROW 3	.00033074	.01490748	.06011628	.09033931
ROW 4	.00074431	.02233410	.08353386	.12355773
ROW 5	.00033074	.01490749	.06011624	.09033925
ROW 6	.00046770	.00339789	.02009165	.03237963
ROW 7	.00185055	.00020706	.00142444	.00325251

COL 5	COL 6	COL 7
.00142444	.00020706	.00185055
.02009164	.00339789	.00046770
.06011627	.01490748	.00033074
.08353386	.02238410	.00074431
.06011624	.01490749	.00033074
.02009165	.00337789	.00046770
.00142444	.00020706	.00185055

Fig. 7.32 Filter for Fig. 7.31.

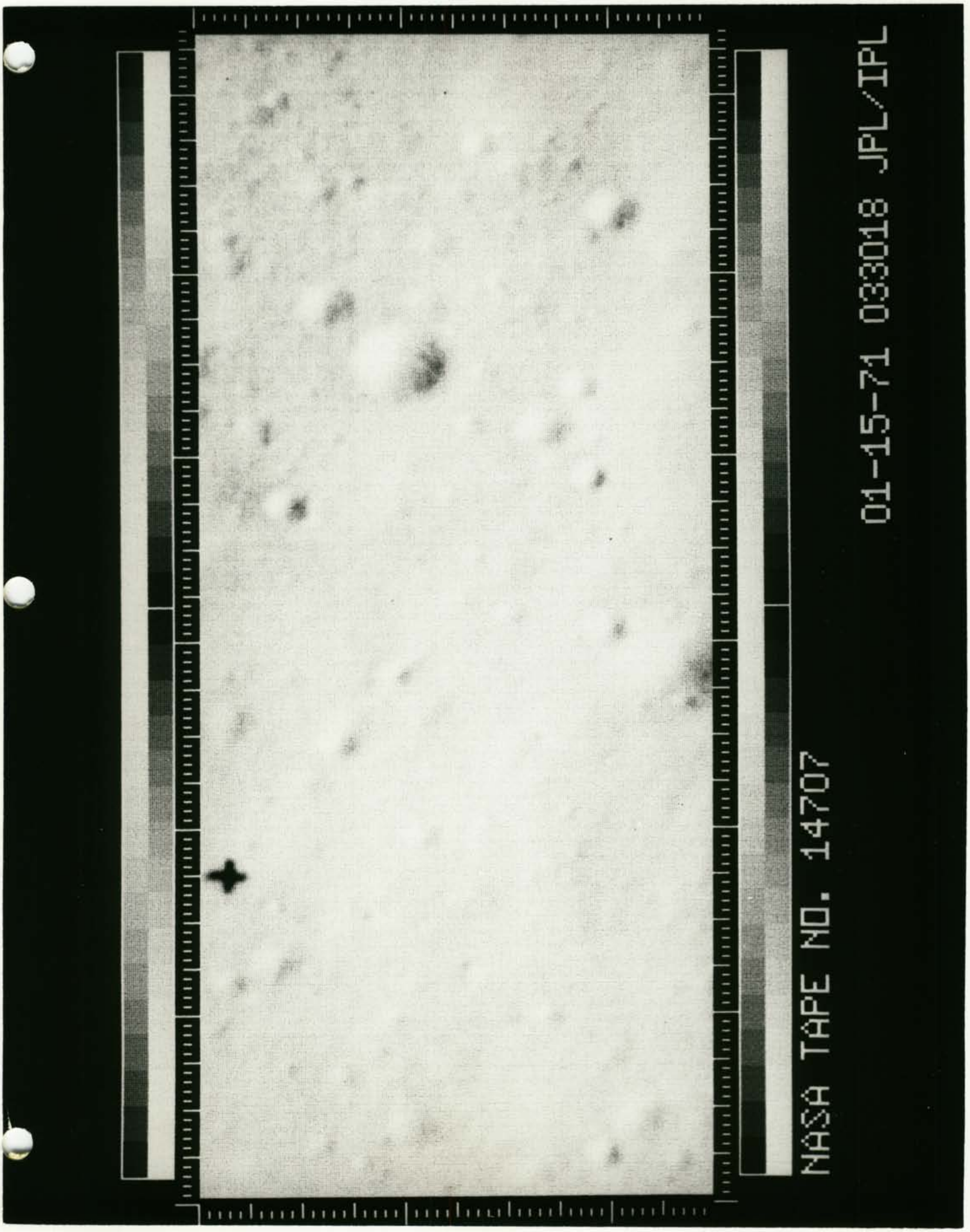


Fig. 7.33 Filtered a = 1.31, b = 33, 7x7, #14707, 033018.

	COL 1	COL 2	COL 3	COL 4	COL 5	COL 6	COL 7
ROW 1	.00061459	.00162536	.00606273	.00895210	.00606273	.00162536	.00061459
ROW 2	.00105196	.00899937	.02549766	.03492886	.02549766	.00899937	.00105196
ROW 3	.00312338	.02098867	.05318965	.07088792	.05318965	.02098867	.00312338
ROW 4	.00440951	.02721399	.06698702	.08862798	.06698702	.02721399	.00440951
ROW 5	.00312338	.02098867	.05318963	.07088789	.05318963	.02098867	.00312338
ROW 6	.00105196	.00899937	.02549766	.03492883	.02549765	.00899937	.00105196
ROW 7	.00061459	.00162536	.00606273	.00895211	.00606273	.00162536	.00061459

Fig. 7.34 Filter for Fig. 7.33.

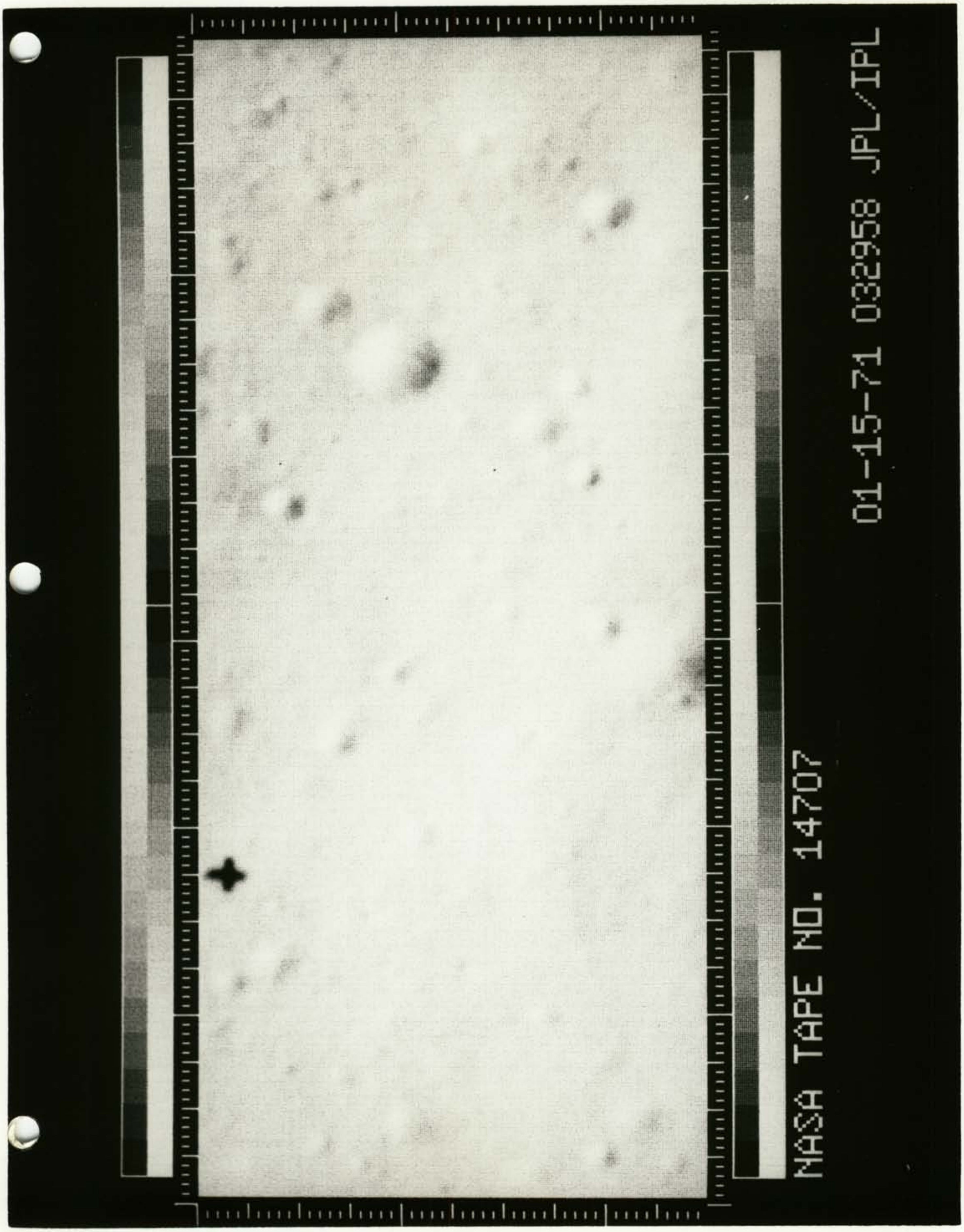


Fig. 7.35 Filtered a = 1.25, b = 29, 7x7, #14707, 032958.

	COL 1	COL 2	COL 3	COL 4	COL 5	COL 6	COL 7
ROW 1	.00098969	.00420105	.00983970	.01282352	.00983970	.00420105	.00098969
ROW 2	.00311456	.01281672	.02696816	.03406077	.02696816	.01281672	.00311456
ROW 3	.00631493	.02334816	.04670498	.05817053	.04670498	.02334816	.00631493
ROW 4	.00790128	.02823893	.05567566	.06906700	.05567566	.02823893	.00790128
ROW 5	.00631493	.02334816	.04670496	.05817052	.04670496	.02334816	.00631493
ROW 6	.00311456	.01281672	.02696815	.03406075	.02696815	.01281672	.00311456
ROW 7	.00098969	.00420105	.00983971	.01282353	.00983971	.00420105	.00098969

Fig. 7.36 Filter for Fig. 7.35.

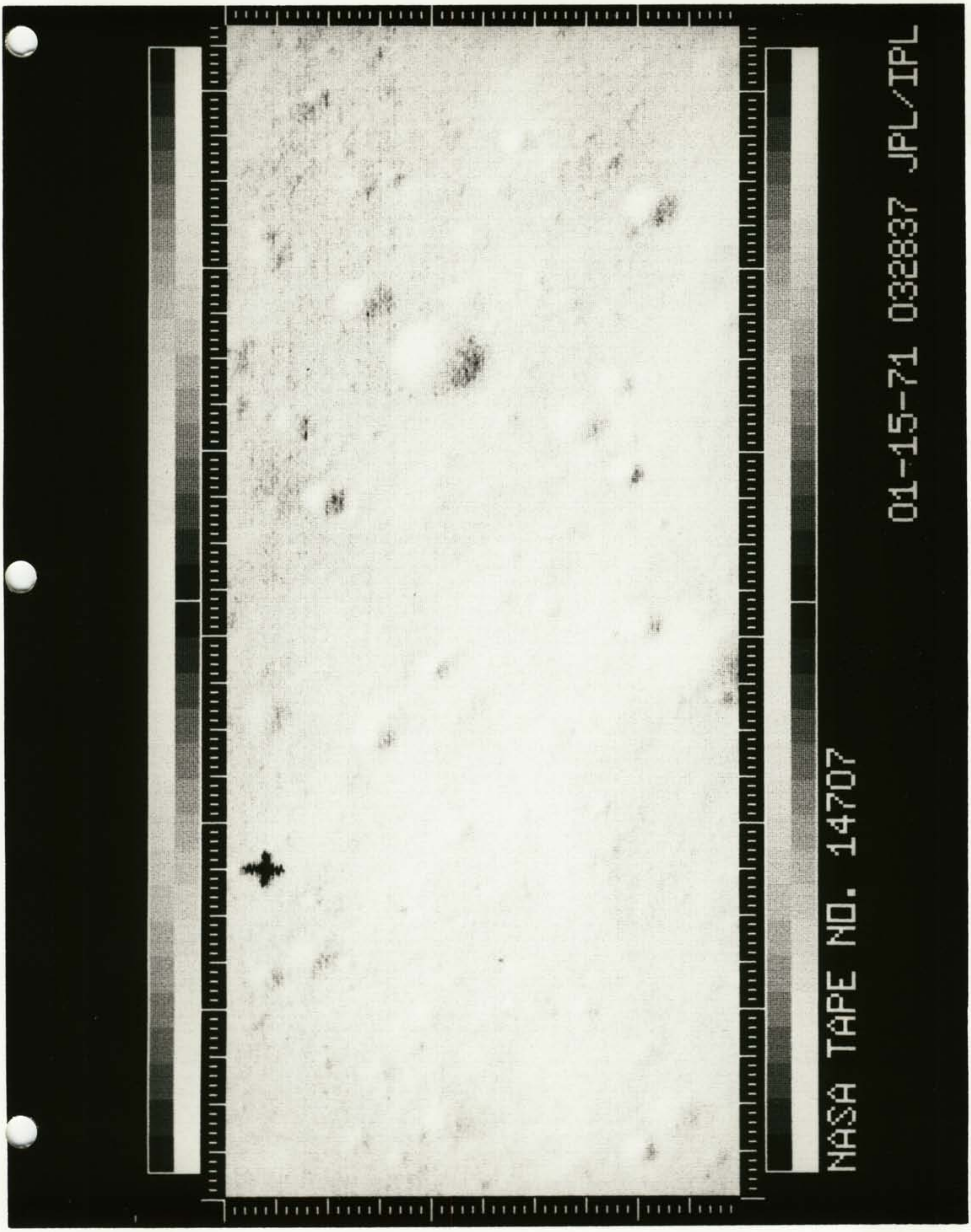


Fig. 7.37 Original (Unfiltered) Area, #14707, 032837.

8. APPLICATION OF FILTERS TO LO DATA.

8.1. THEORY

Mathematically, the filters were conceived in the spatial frequency domain; since, as has been indicated in §1, multiplication in the frequency domain corresponds to convolution in the real domain, our filters are applied as convolution transforms. Thus our filters (denoted by f) change the input data i into output data o by the formula

$$o(u,v) = \int\int f(x,y)i(u-x,v-y)dxdy.$$

Our real data, however, is not given by a function (nor is the filter), but is given as an array, with spacing between data points .003605 mm in the x-direction and .00405 mm in the y-direction. The filters were approximated by numerical integration over rectangles centered at these grid points. Our filters are applied as numerical convolutions

$$o(n,m) = \sum_i \sum_j f(i,j)i(n-i,m-j),$$

it being assumed that the filters have previously been normalized.

(An aside: Refer to Fig. 5.1. The filter value near 0 of about 150,000 may seem surprising; however, when one considers that, numerically, dx = .003605 and dy = .00405, we obtain the weighted value at 0 for the numerical filter of about 2.2(≐ .003605 x .00405 x 150,000); this filter was not actually used. The filter with b=80 had a=2.8, resulting in a much smoother response in the real domain.)

8.2. PRACTICE

Since FORTRAN arrays have positive subscripts, the formula in the program looks different; for example, for a 3x3 filter and a 636x738 picture

$$O(N,M) = \sum_{I=-1}^{1} \sum_{J=-1}^{1} F(I+2,J+2)\,I(N-I,M-J)$$

for $N = 2, \cdots, 635$ and $M = 2, \cdots, 737$. Thus the discrete filter application produces a smaller picture than the original picture. For a (2n+1)x(2m+1) filter, the border of unfiltered data is n from the left and right and m from the top and bottom. This problem can be solved in several different ways:

(i) Leave the border out and just have a smaller output picture than input picture.

(ii) Replace the border areas with black or white (77 or 00) to make the output picture the same size as the input picture.

(iii) Replace the border areas with unfiltered original data.

(iv) Fill out the border with constant data with value equal to the value at the nearest filtered data point.

(v) Design and furnish special border filters based on the center filter. As an example, the 3x3 filter

.1	-.1	.1
-.3	1.4	-.3
.1	-.1	.1

might have associated with it four special border filters:

-.1	.1	.1	-.1	-.3	1.5	-.3	-.3	1.5	-.3
1.3	-.3	-.3	1.3	.1	-.1	.1	.1	-.1	.1
-.1	.1	.1	-.1						
left edge		right edge			top			bottom	

The special filters may be obtained from the large inside filter by (a) changing the center value to make the sum of the filter val es 1 (the method used in the above example) or (b) by renormalizing the filter (leading to more complicated values but keeping what is probably closer to the intended spatial frequency response). In method (b), for example, the left edge filter would be:

-.091	.091
1.272	-.272
-.091	.091

If the filter is larger, then more special filters (and, accordingly, more complex logic in the program) will be required. For a (2n+1)x(2m+1) filter, 2(n+m) filters will be required.

We used method (ii) for our initial tests, methods (iii) and (iv) for the filtered data for use in the analysis program. In the program documented in the next section, method (iv) is used since it seemed to produce better edges on the contour maps.

Another (simple) problem remains. If the filter is one of those which enhance the high frequency response to a marked degree, the possibility exists that a filtered value will be negative or larger than 63. When such numbers are converted to octal numbers, they will "fold over". For example, the 3x3 filter a=4.0, b=103 (see Fig. 7.6) applied to the area

10	30	10
30	60	30
10	30	10

gives the convolved value (truncated) 66, which (when converted to an actual number by PACK6) becomes 02_8. This is, of course, not the intended value, and will cause white areas where there should be dark areas. The solution we employ is to clip the output data before packing with PACK6. A (perhaps) better solution is to predict when foldover will happen by looking at the variation of values and turning the filter off in the neighborhood of an extremely high-contrast area. (This solution would reduce ringing; however, our filters seldom ring.)

The filter program documented next is general enough to apply filters to any size area (with the only limitation being the core size); because of the usual tape/core/drum/tape problems, we employ double buffering and a circular buffer to speed up input/output (using NTRAN) and allow calculations to be performed simultaneously.

COMPUTER PROGRAM DOCUMENTATION

Program FILTER

Project B

by

Jack Bryant

and

R. L. Wendt

Prepared by

Applied Scientific Research, Inc.

Houston, Texas

Under Contract NAS 9-10577

For

MAPPING SCIENCES BRANCH

National Aeronautics and Space Administration

Manned Spacecraft Center

Houston, Texas

December, 1971

9. COMPUTER PROGRAM DOCUMENTATION - FILTER

The filter program described in this document is incorporated (and so documented) as a subroutine. We first describe the parent program - MAIN.

9.1. PROGRAM DESCRIPTION

The Main program is the parent routine of the FILTER subroutine and associated subroutines and provides for initialization of data, parameters and positioning of tapes.

9.2. USAGE

9.2.1. DESCRIPTION OF INPUT

Card input:

R Gray shade data for making D log E; the array R contains the exposure values in MCS which are to be assigned to the LO data numbers. In our problem, we obtained (from study of the gray scale averages, described in Part I above) the following array for R (read in FORMAT (11F7.4)):

LO#N	I	R(I)	LO#N	I	R(I)	LO#N	I	R(I)
0	1	.800	32	5	.223	64	9	.128
8	2	.500	40	6	.192	72	10	.112
16	3	.330	48	7	.167	80	11	.100
24	4	.264	56	8	.145			

For example, in this digitization, the number 24 represents an exposure value of .264 MCS. The exposure values corresponding to $\geq$ 64 are obtained by extrapolation of the D log E curve, and are needed owing to the effects of clipping.

CF An array of 636 correction factors, smoothed, obtained by smoothing the column averages associated with this digitization, read in FORMAT (6E13.1).

K,L The half-dimension of the filter; for example, a 7x7 filter has K = L = 3. Read in FORMAT (2I3).

NFLTR Number of filters to be applied this run, read in FORMAT (I2).

FLTR The normalized filter of dimension (2K + 1) x (2L + 1), read in FORMAT (8F10.6)(one row per card when K $\leq$ 3).

Tape input:

The specific program documented here requires one of our special one line per record tapes, produced by program COPYT documented in §10.

9.2.2. DESCRIPTION OF OUTPUT

Tape output:

The filtered pictures, written one line per record with EOF between pictures for NFLTR pictures, and two EOF after all pictures. Should one desire to use the output in the LO Analysis Program, this tape output is used as input for program MCSF documented in §12.

Printed output:

The correction factors and filter values are printed. In the present version, the filtered picture is printed using subroutines PRETTY and PRINTO to reproduce pictures using symbols and overprinting. This output provides a quick check on the results, and has been used before in program JOIN.

9.2.3. RUN PREPARATION

Card deck setup: see Fig. 9.2.3.1.

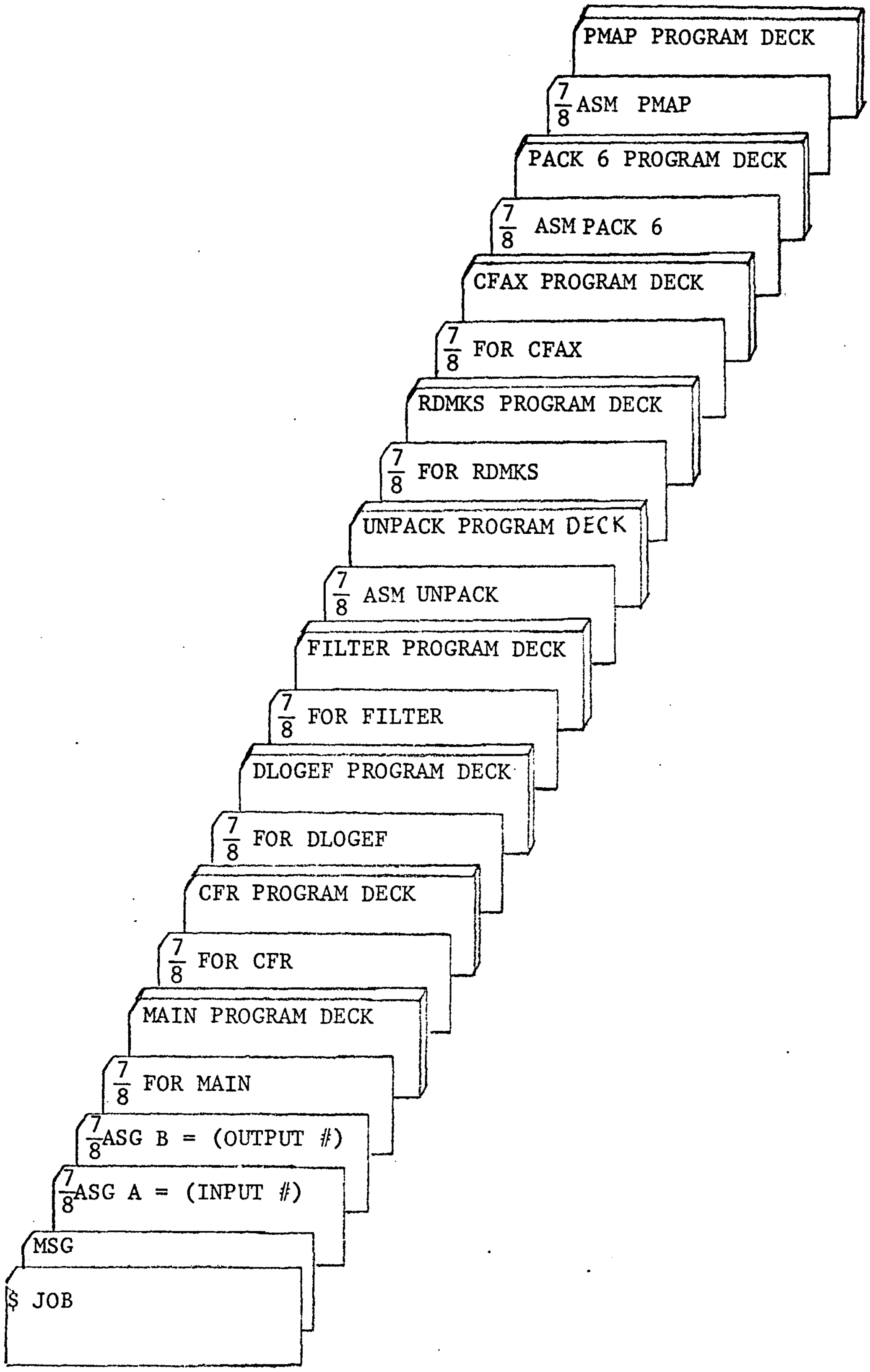


Fig. 9.2.3.1. DECK SETUP (UNIVAC 1108) MAIN (FILTER)

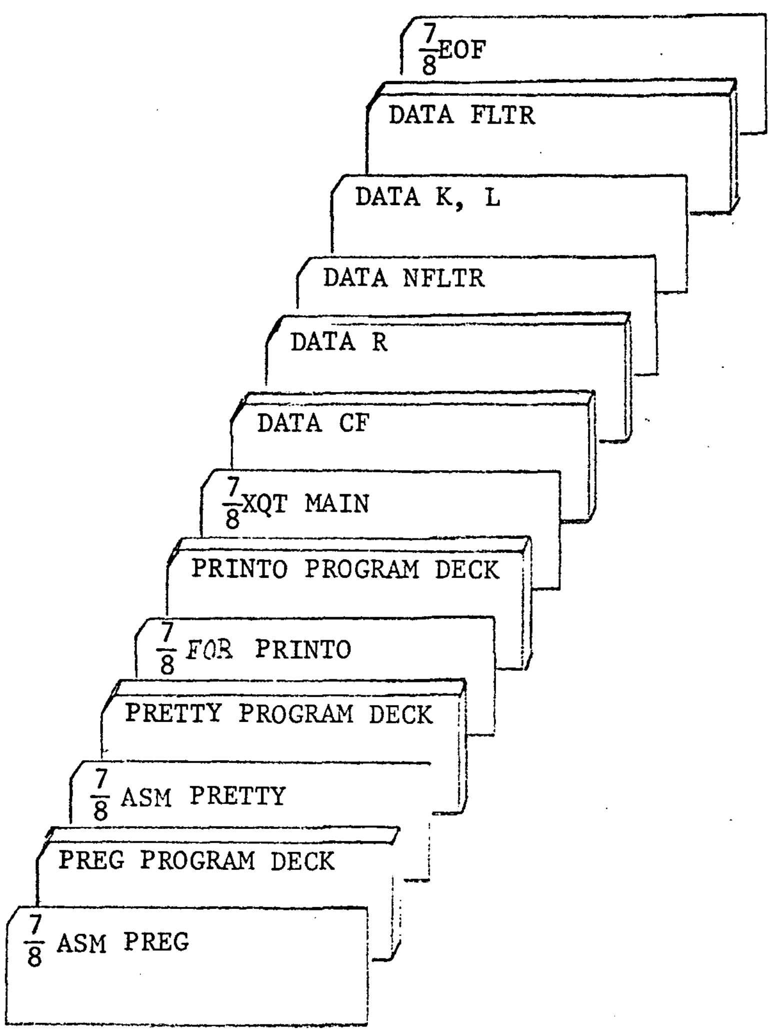


Fig. 9.2.3.1. DECK SETUP (UNIVAC 1108) MAIN (FILTER) (Concluded)

Subroutine requirements:

CFR — A routine which performs CF(I) = 10./CF(I); see 9.3.3.

DLOGEF — A routine which calculates D log E data given gray shade data R; see 9.3.4.

FILTER — The filter routine; see 9.3.5.

RESET TIME — System timing routines.

NTRAN — System tape read, write routines.

9.2.4. EXECUTION CHARACTERISTICS

Restrictions: Owing to the use of PARAMETER variables in the FORTRAN program MAIN, the routine can be easily redimensioned to filter data of any size (except for the limitations of the core). For large filters, some DIMENSION statements must be modified in the obvious manner.

Storage requirements: Code and storage: 132053_8. This leaves free 15724_8 for possible larger areas. Using the same code, with only minor modifications, the nine chit JOIN picture was filtered, with no storage problems.

Run time: For use of all features employed here, we found the following formula for filtering and performing all corrections: $t = 9(2K + 1)(2L + 1) + 206$ seconds to correct one chit (= 738x636 elements).

For use in the analysis program, only one-fourth as many filtered data points need be processed. This fact, plus the speed-up suggested in the filter program, should allow all corrections (including filtering) to be performed on a chit in less than one minute, exclusive of hanging tapes. Even the present algorithm, applied to every element in every row, requires less than five minutes per chit for a highly effective 3x3 filter.

Printed output: Approximately 100 pages per chit.

Accuracy: Well within the limits of the LO data - The procedure takes special pains to improve accuracy in the application of correction factors, removal of drummarks and in the correction of the D log E curve. For example, we expect the accuracy to be within 2% for our correction factors and D log E curve, whereas the corrections in the Analysis Program are probably not better than 20%.

9.3. <u>REFERENCE INFORMATION</u>

DETAILED FLOW CHART - See Fig. 9.3.1.

DESCRIPTION OF VARIABLES - See Table 9.3.2.

DESCRIPTION OF SUBROUTINES: See the appropriate section:

9.3.3. CFR

9.3.4. DLOGEF

9.3.5. FILTER

Other subroutines, called by the FILTER subroutine, are documented in the report on Project A.

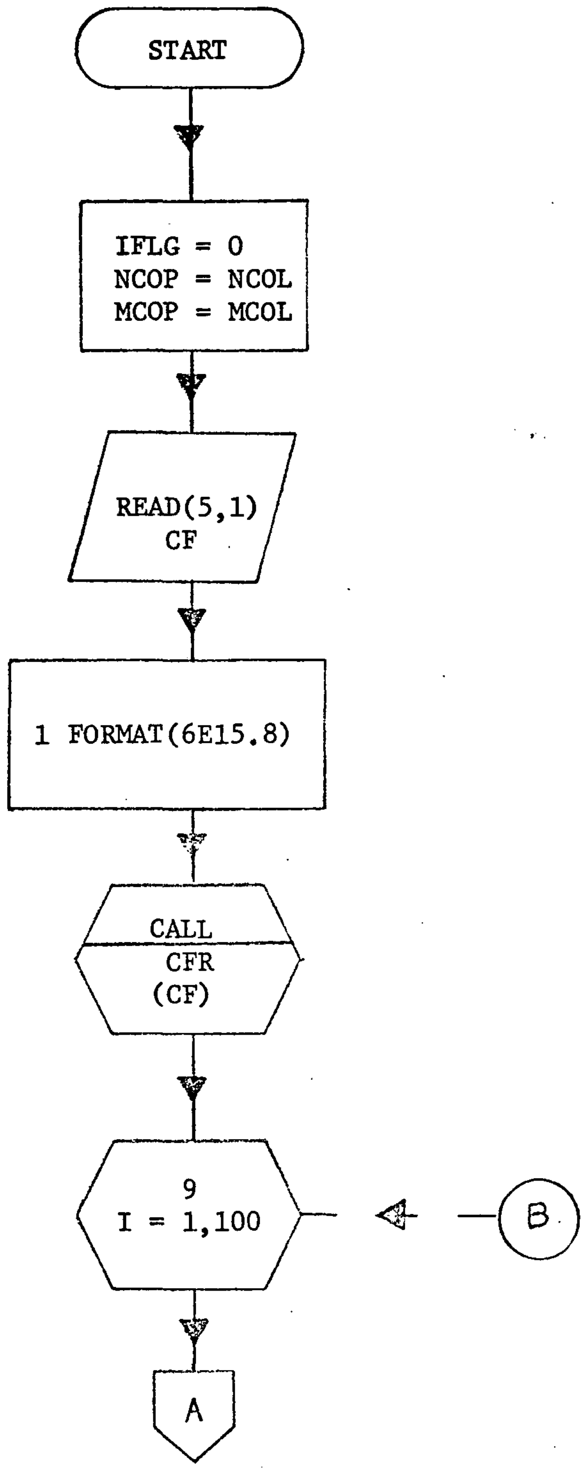


Fig. 9.3.1. DETAILED FLOW CHART - MAIN

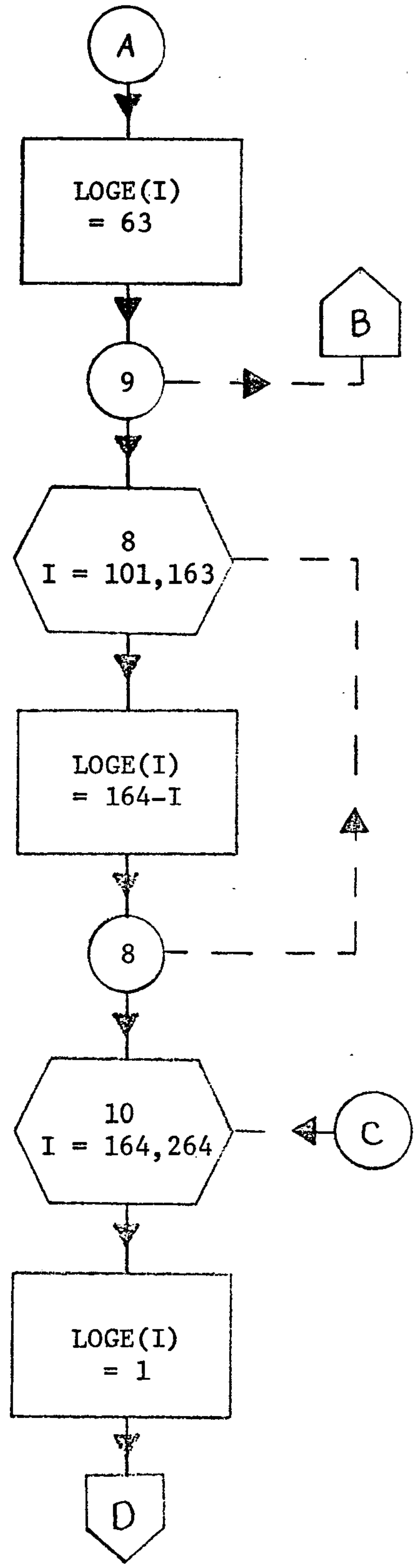


Fig. 9.3.1. DETAILED FLOW CHART - MAIN (Continued)

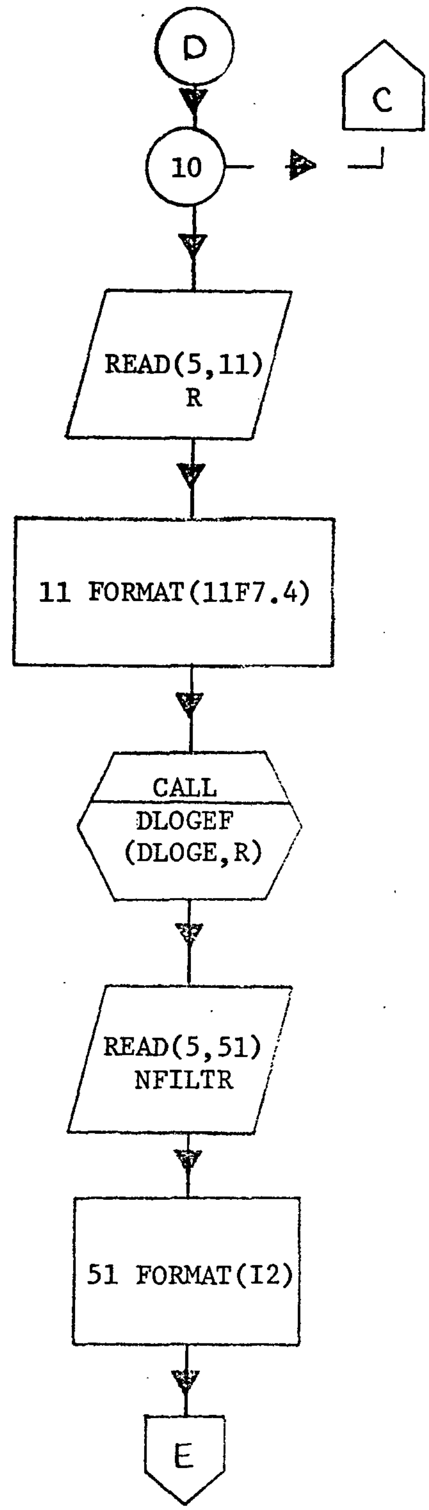


Fig. 9.3.1. DETAILED FLOW CHART - MAIN (Continued)

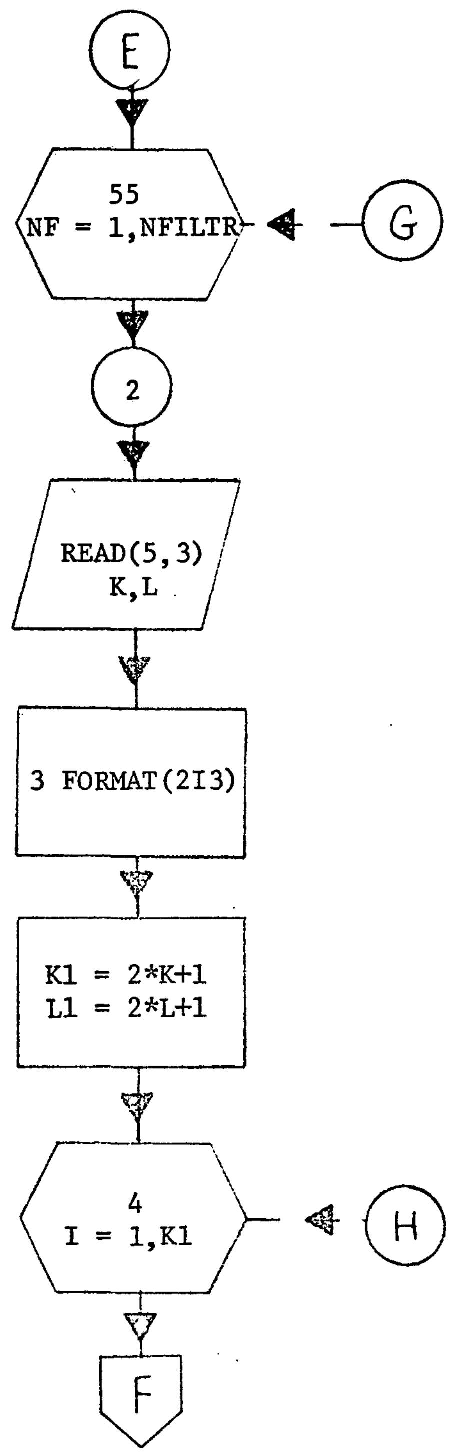


Fig. 9.3.1. DETAILED FLOW CHART - MAIN (Continued)

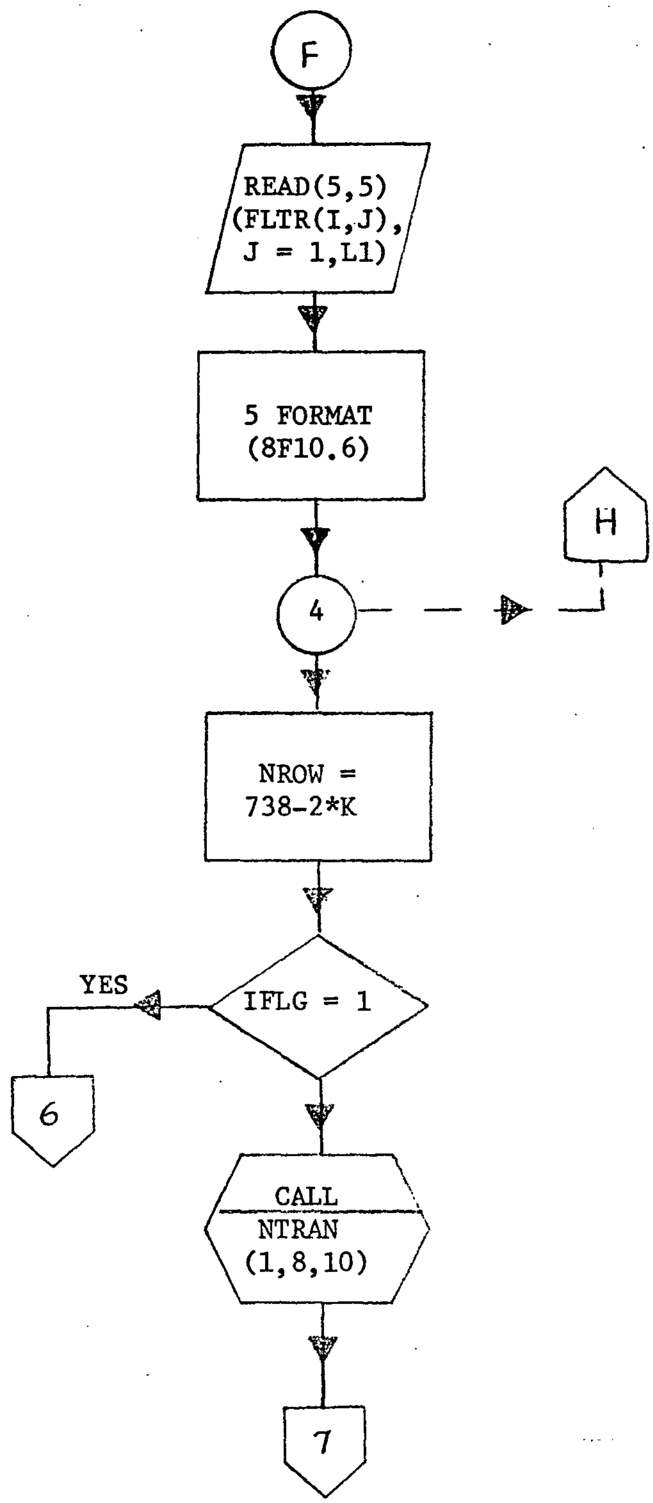


Fig. 9.3.1. DETAILED FLOW CHART - MAIN (Continued)

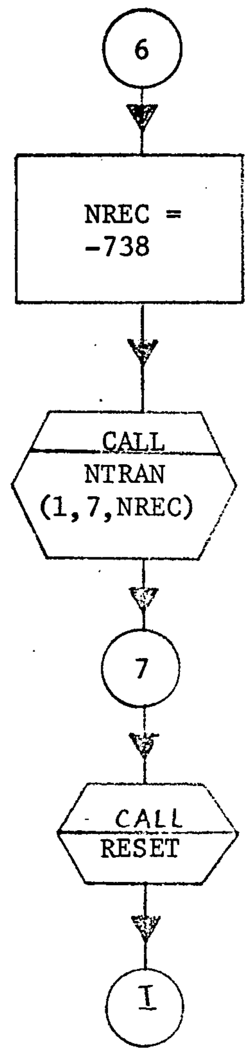


Fig. 9.3.1. DETAILED FLOW CHART - MAIN (Continued)

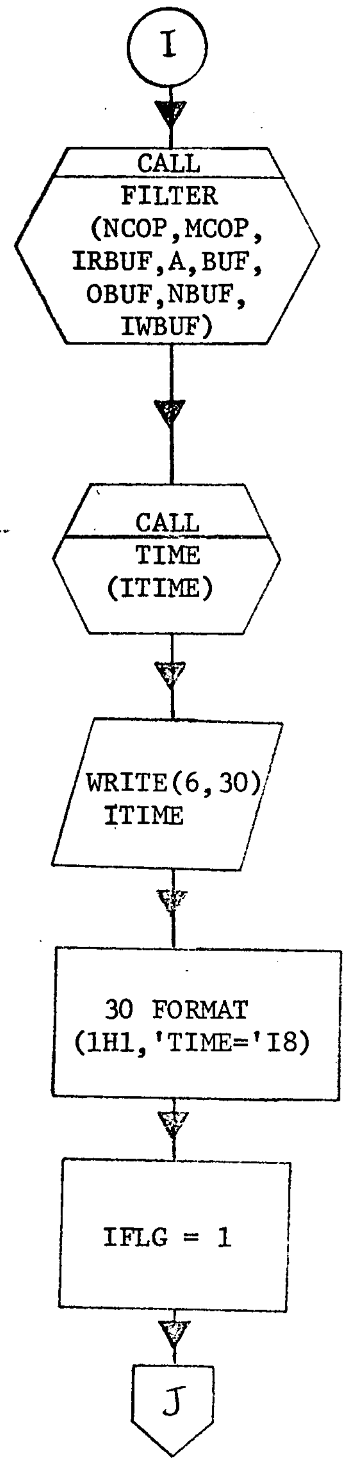


Fig. 9.3.1. DETAILED FLOW CHART - MAIN (Continued)

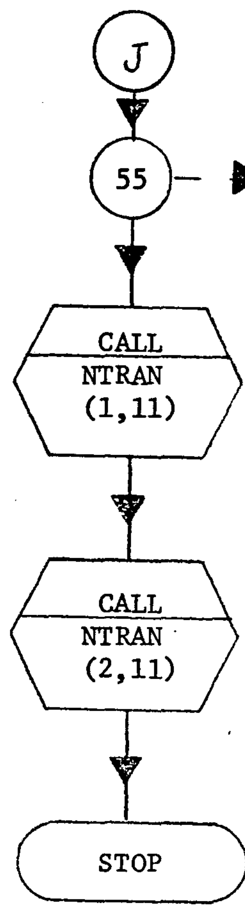


Fig. 9.3.1. DETAILED FLOW CHART - MAIN (Concluded)

NCOL - Number of characters per row of data.

MCOL - Number of words per row of data.

IRBUF - Array used in input by NTRAN.

A - Array used in unpacking data.

BUF - Array which is used is circular buffer by FILTER program.

OBUF - The filtered output (1 line) in floating point.

NBUF - The corrected filtered output in meter candle seconds and is either a positive or negative type data depending on LOGE.

IWBUF - Packed data from NBUF; used in output by NTRAN.

CF - Smoothed correction factors; then the output of CFR.

R - Gray shade data read in by data cards; used to generate D log E curve.

DLOGE - D log E data.

K,L - Dimension of 1 quadrant of filter.

K1,L1 - Dimension of filter 2K+1,2L+1.

FLTR - filter.

LOGE - Data which converts picture to positive or negative.

NCOP,MCOP - Dummy NCOL,MCOL.

NFILTR - Number of filters to be applied to the data.

NROW - Number of rows to be filtered.

NREC - Number of records to back up the input tape to the filter program if additional filters are to be applied.

IFLG - Flag used to bypass certain instructions if more than one filter is to be applied to the input data.

ITIME - Time in milliseconds elapsed in filtering the data.

Table 9.3.2. DESCRIPTION OF VARIABLES - MAIN

9.3.3 SUBROUTINE CFR

IDENTIFICATION

Name/Title	- CFR (correction factor reciprocal)
Author/Date	- R. L. Wendt, January 1971
Organization	- ASR
Machine Identification	- UNIVAC 1108
Source Language	- FORTRAN V

PURPOSE

This routine changes the smoothed correction factors into correction factors suitable for use as multiplicative factors, and at the same time multiplies the factors by 10.0 to allow better tonal resolution in the application of D log E corrections.

USAGE

Calling Sequence

CALL CFR (CF)

where CF go in as the array of smoothed correction factors and return as 10./CF.

DETAILED FLOW CHART

See Fig. 9.3.3.1. for a detailed flow chart of CFR.

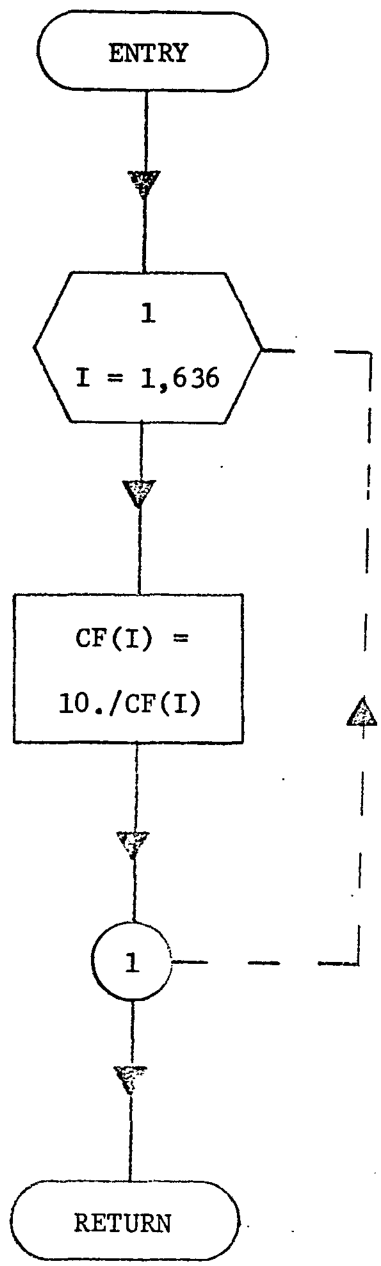


Fig. 9.3.3.1. DETAILED FLOW CHART-CFR.

9.3.4 SUBROUTINE D LOG EF

IDENTIFICATION

Name/Title	- D LOG EF (D log E filler)
Author/Date	- R. L. Wendt, December 1971
Organization	- ASR
Machine Identification	- UNIVAC 1108
Source Language	- FORTRAN V

PURPOSE/METHOD

D LOG EF provides data for the conversion from corrected (and scaled by a factor of 10.0) LO data to exposure values in meter-candle-seconds. The values are obtained by linear interpolation between the data points at abscissa valves 1 thru 800, the interpolation being performed on data given at multiples of 80 with eleven data points being required.

USAGE

Calling Sequence

CALL D LOG EF (D LOG E, R)

where D LOG E and R are as described in MAIN

(see Table 9.3.2).

DESCRIPTION OF VARIABLES

N, N1, N2 - Values of the abscissa used in linear interpolation.

K - Index used in storing D log E data.

Y1, Y2 - Values of the ordinate used in the linear interpolation.

D LOG E, R - Described in MAIN.

DETAILED FLOW CHART

See Fig. 9.3.4.1. The function F defined (using FORTRAN V DEFINE statement) is given by

```
F(N,N1,N2,Y1,Y2) = (Y2-Y1)*(N-N1)/(N2-N1)+Y1.
```

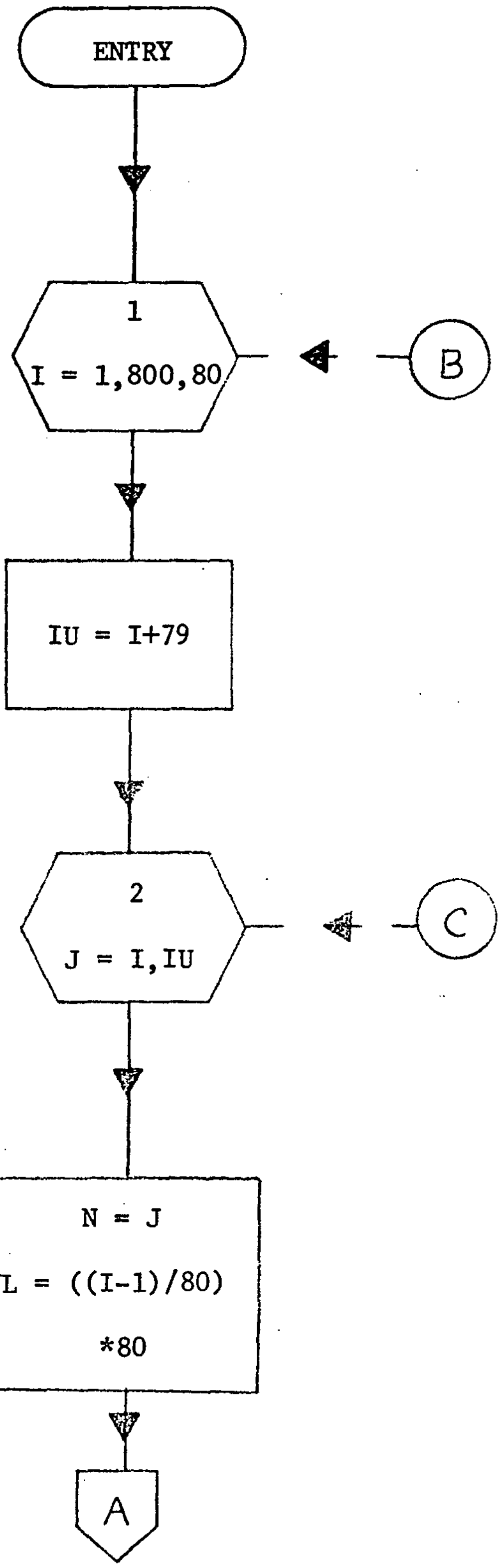


Fig. 9.3.4.1. DETAILED FLOW CHART - D LOG EF

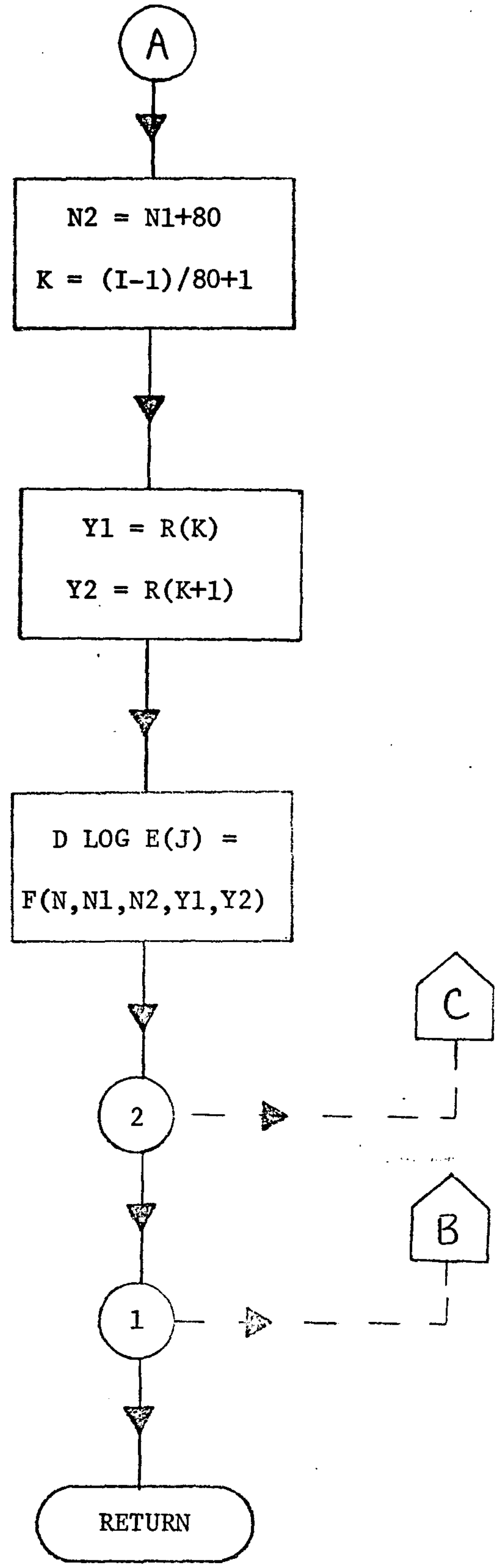


Fig. 9.3.4.1. DETAILED FLOW CHART - D LOG EF (Concluded)

9.3.5 SUBROUTINE FILTER

IDENTIFICATION

Name/Title	- FILTER (Filter program)
Author/Date	- R. L. Wendt, November 1970
Organization	- ASR
Machine Identification	- UNIVAC 1108
Source Language	- FORTRAN V

GENERAL DESCRIPTION

The Filter program that will be described in this document is a symmetric type filter and is incorporated as a subroutine.

The routine can be used to perform filtering of Lunar Orbiter data in general, but what will be described in this document is a specific application so that the output may be input to the Analysis program.

Data correction is provided by the filter program. Drum mark removal, correction factors and D LOG E corrections are applied during the filtering process. There is evidence that all of the data correction done presently by the Analysis program or utility programs is deficient in theory or programming or both.

The filter program, being a subroutine, requires the driver routine MAIN to provide data input, both cards and tape, data correction arrays D LOG E and CFAC, and the geometric data size, i.e., number of rows to be filtered and number of data elements per row.

The standard Lunar Orbiter chit is 738 lines by 636 characters per line, but the program could be (and has been) modified easily to filter such data as a mosaic of Lunar Orbiter chits.

In the documentation, one of the DO-Loops could be modified in such a manner that the algorithm would execute faster. This modification will be indicated

DESCRIPTION OF THE FILTER ALGORITHM:

The filter program reads 2K+1 lines of data, unpacks the data, applied correction factors and D LOG E corrections, and stores this real array as the first 2K+1 rows of BUF. The routine initializes the read of another line except when an index equals NROW. The filter ranges across the framelet applying the filter FLTR as a convolution with BUF. The filtered lines are stored in OBUF and the data (now in MCS) is scaled linearly, in this program so that 100 ↔ .135MCS . The resulting integer array NBUF is reconverted to the same type as the input data was and simultaneously clipped, so that the bias by a factor of 100 is removed. The data is then packed into IWBUF. The filtered line is outputted by NTRAN and the line which was being read while the previous line was being filtered is processed, corrected and stored in BUF, using modular arithmetic for the index, which treats BUF as a circular buffer.

The edges are treated differently, being held constant and equal to the nearest filtered value. This provided better contour maps than other attempted solutions of the edge problem.

A provision is included so that the filtered picture can be reduced in tonal resolution and printed via. PRETTY and PRINTO.

<u>USAGE</u>

Calling sequence

CALL FILTER (NCOL, MCOL, IRBUF, A, BUF, OBUF, NBUF, IWBUF)

Common block:

COMMON/FILTR/K, L, K1, L1, NROW, FLTR, CF, D LOG E, LOG E

All variables are as described in Table 9.3.2.

DESCRIPTION OF PROGRAM VARIABLES - See Table 9.3.5.1.

DESCRIPTION OF SUBROUTINES:

UNPACK - Standard Lunar Orbiter subroutine.

RDMKS - Removes drum marks. Documented in the report for Project A, §10.5.3.2.1.

CFAX - Applies smoothed correction factors (the output of CFR, §9.3.3) as multiplicative correction factors - see §9.3.6.

PACK6
PMAP - Standard Lunar Orbiter subroutines.
PREG

PRETTY
PRINTO - Modified versions of PRETTY and PRINTM as presently exist in Lunar Orbiter documentation, providing for better printed picture via. overprinting and the use of symbols; see also 9.3.7. and 9.3.8.

DETAILED FLOW CHART - See Fig. 9.3.5.2.

NCOL, MCOL, IRBUF, A, BUF, OBUF, NBUF, IWBUF, K, L, K1, L1, NROW, FLTR, CF, D LOG E, LOG E - Described in MAIN, Table 9.3.2.

MTX - Array which is used to reduce the tonal resolution and in printing the output picture.

NCO1 - The next picture element following the last picture element which is filtered in any row.

NWOUT - The number of words outputted per row.

C, D - Conversion factors used to convert data to meter candle seconds.

INDX - An index used in a FORTRAN type table look-up.

JL, JN - Upper and lower limits over which an index in a DO-Loop can vary. The first and last elements in any row to be filtered.

IKI - Index which serves as row counter for MTX.

KI, NI - Index determined by modular arithmetic used in making BUF circular.

Table 9.3.5.1. DESCRIPTION OF PROGRAM VARIABLES - FILTER.

FLOW CHART: FILTER

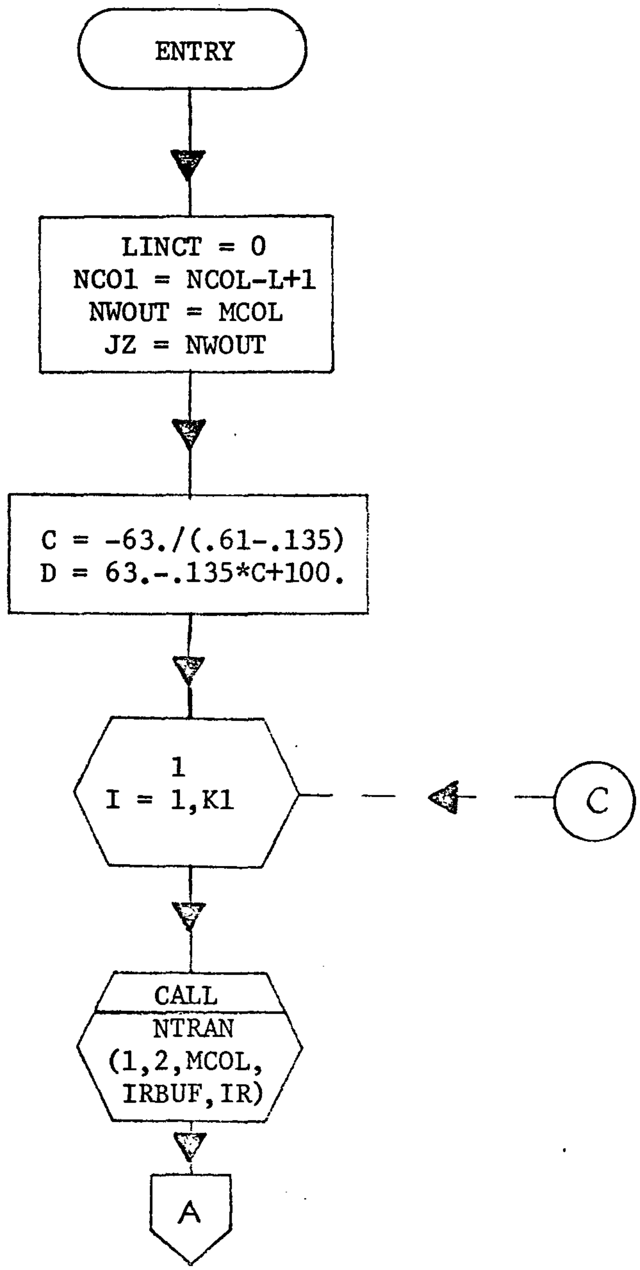


Fig. 9.3.5.2. DETAILED FLOW CHART - FILTER

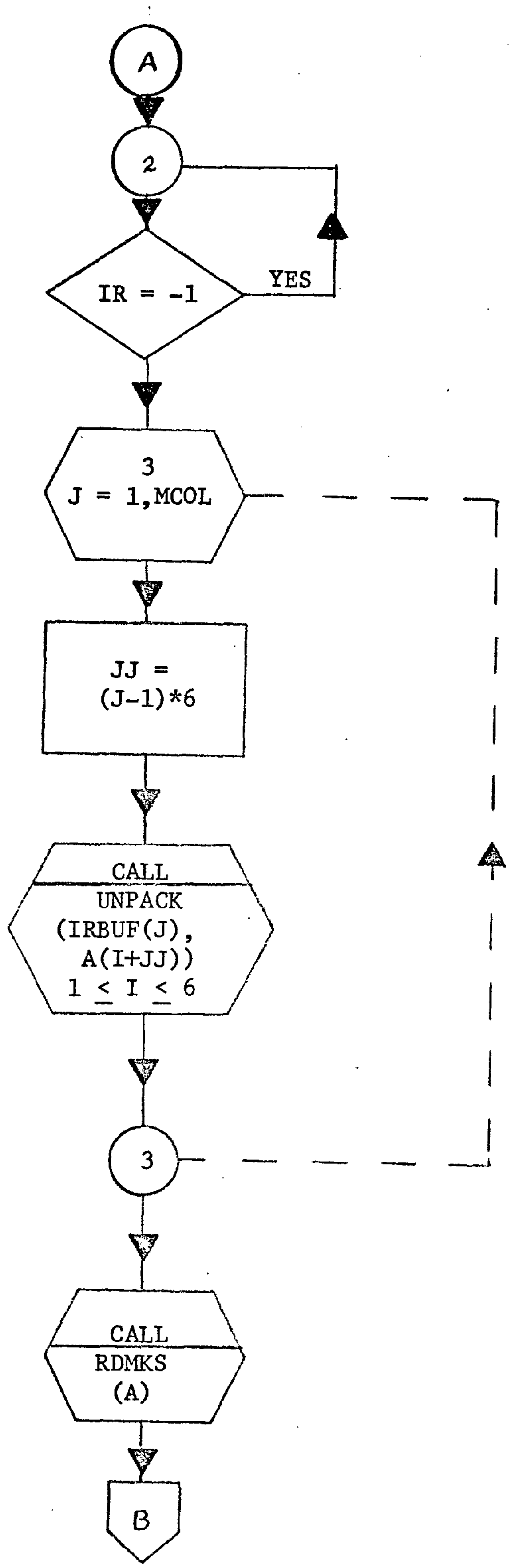


Fig. 9.3.5.2. DETAILED FLOW CHART - FILTER (Continued)

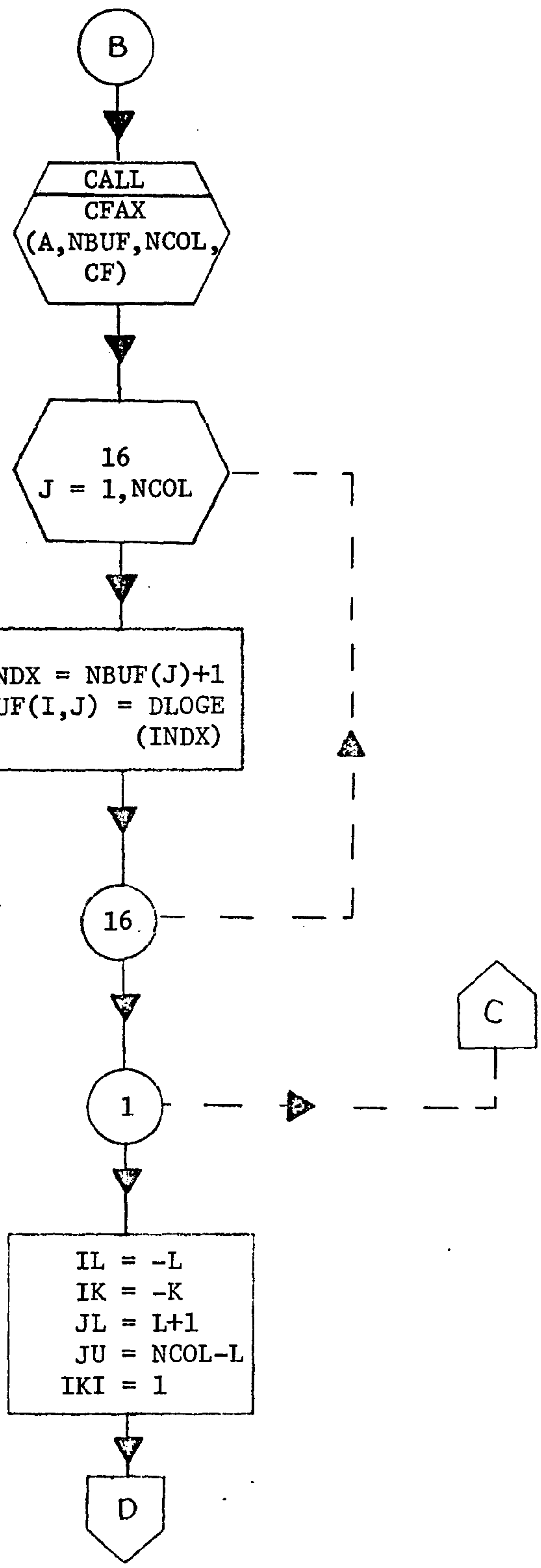


Fig. 9.3.5.2. DETAILED FLOW CHART - FILTER (Continued)

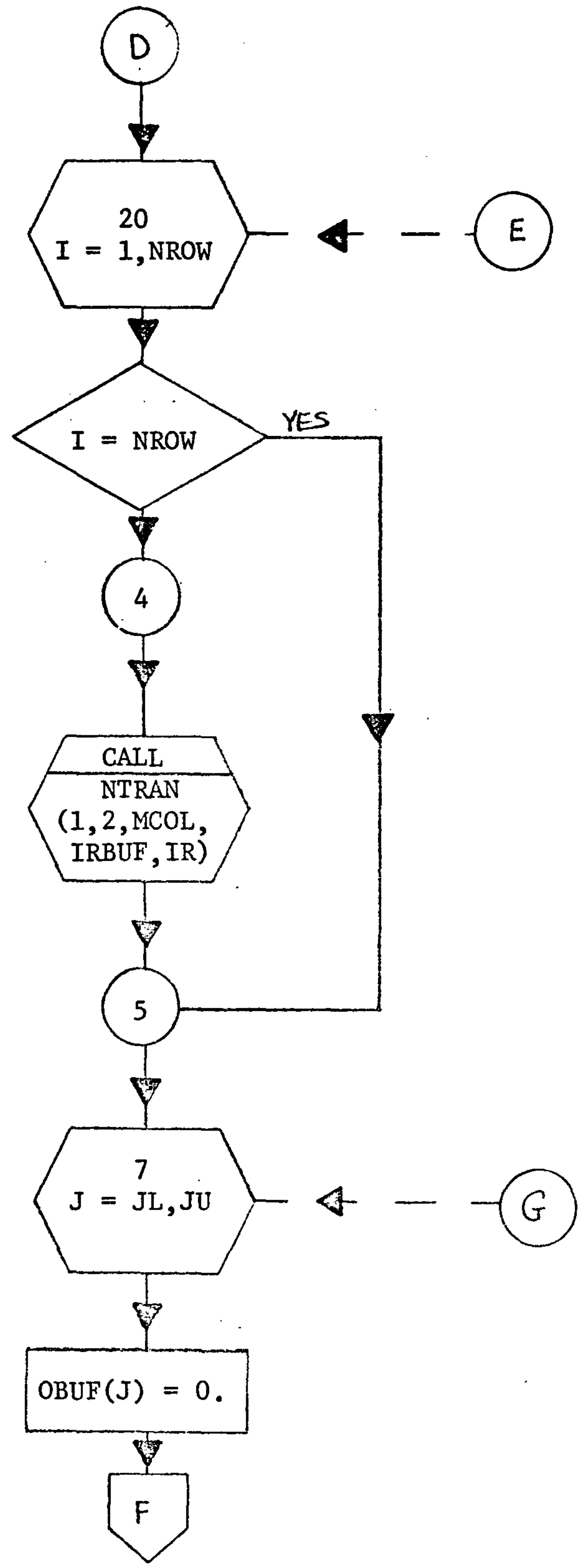
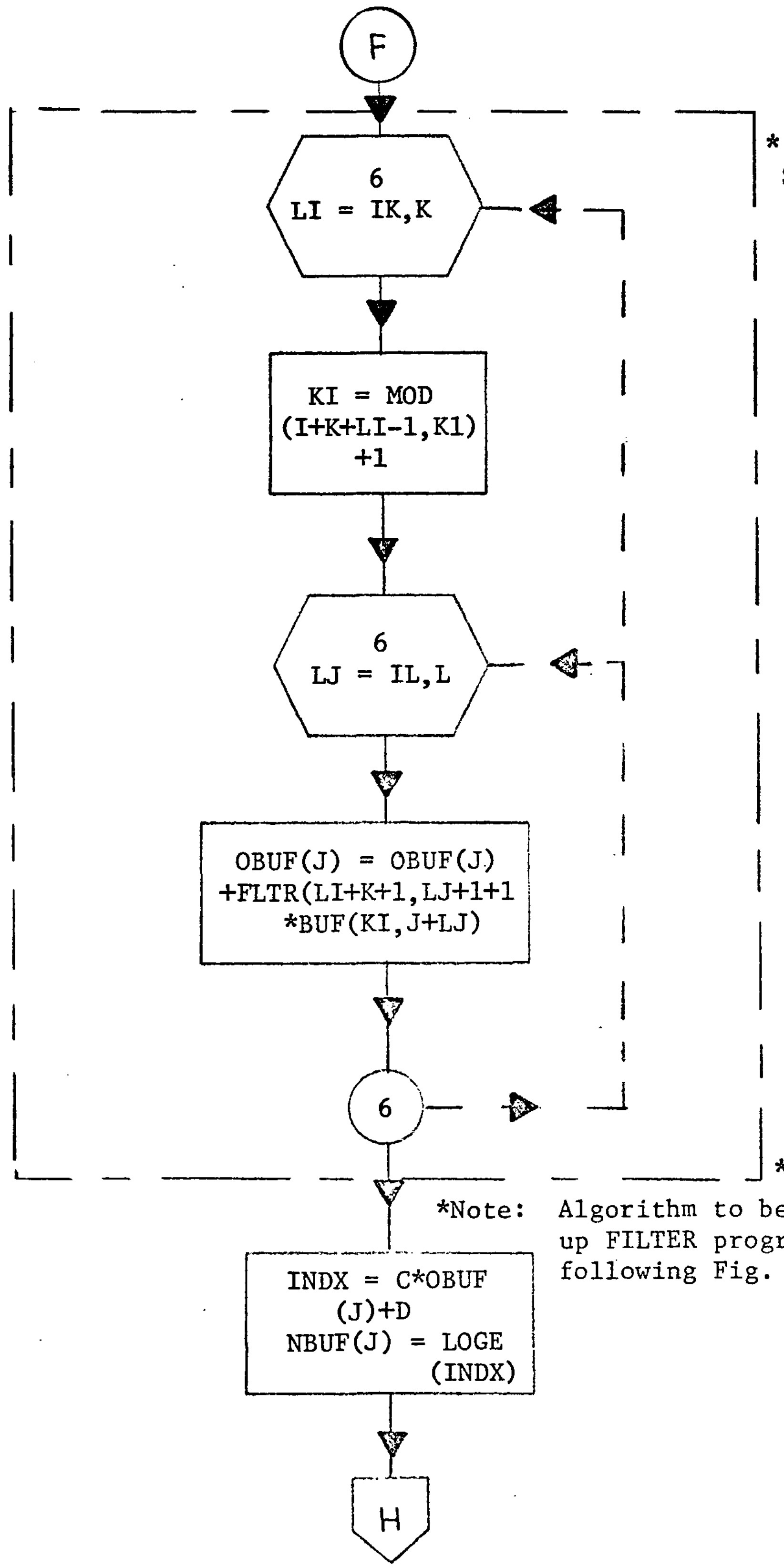


Fig. 9.3.5.2. DETAILED FLOW CHART - FILTER (Continued)



*Note: Algorithm to be modified to speed-up FILTER program; see document following Fig. 9.3.5.2.

Fig. 9.3.5.2. DETAILED FLOW CHART - FILTER (Continued)

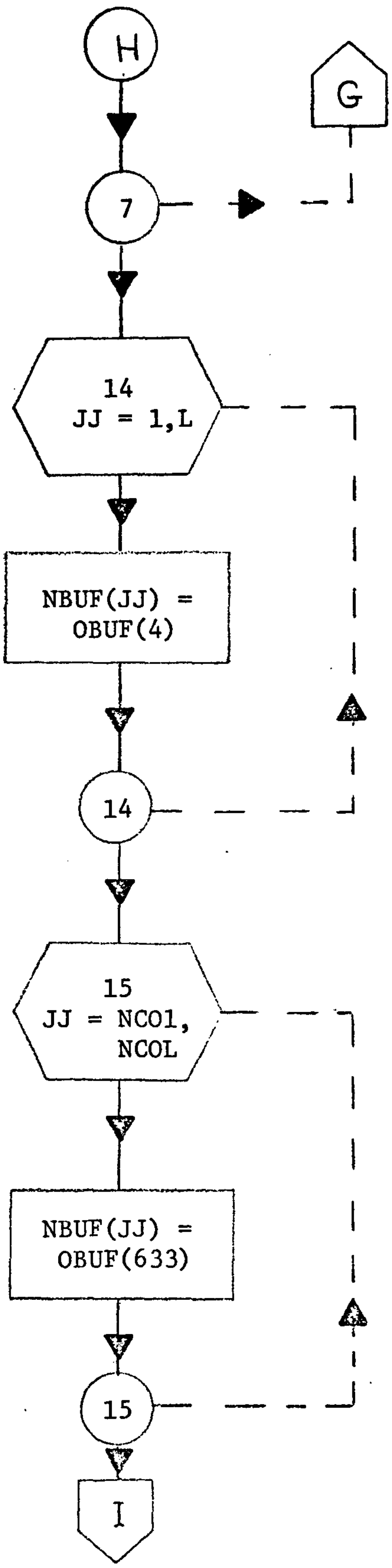


Fig. 9.3.5.2. DETAILED FLOW CHART - FILTER (Continued)

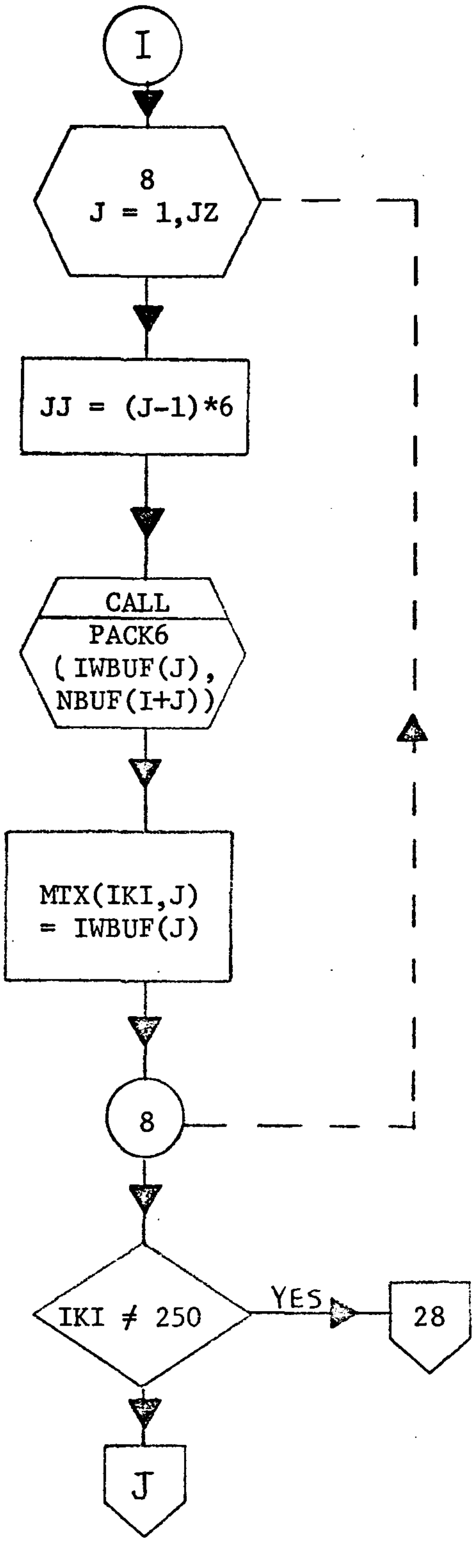


Fig. 9.3.5.2. DETAILED FLOW CHART - FILTER (Continued)

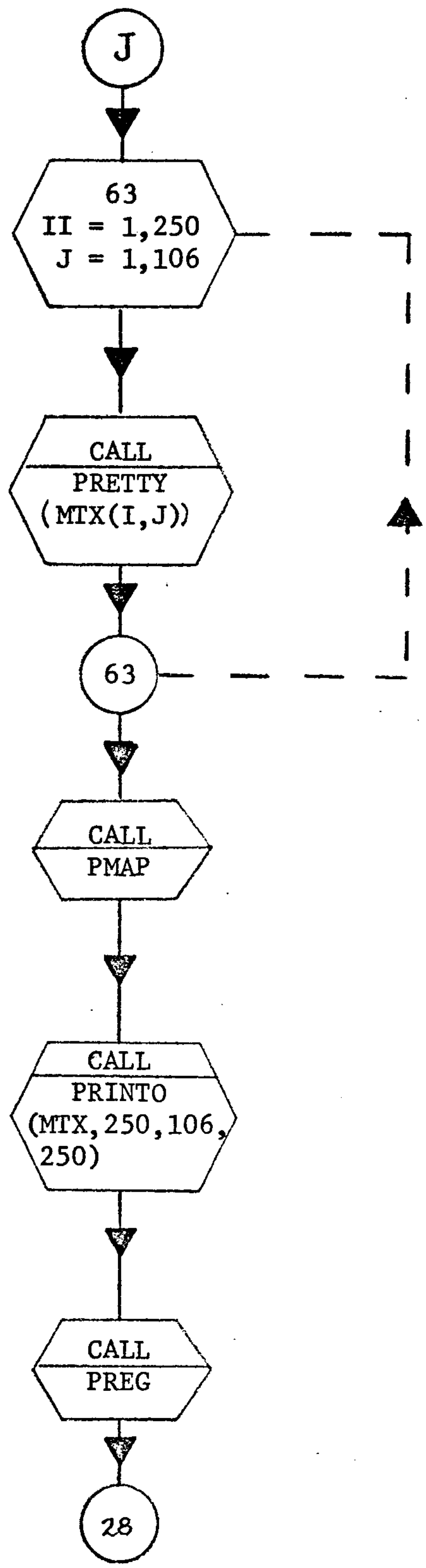


Fig. 9.3.5.2. DETAILED FLOW CHART - FILTER (Continued)

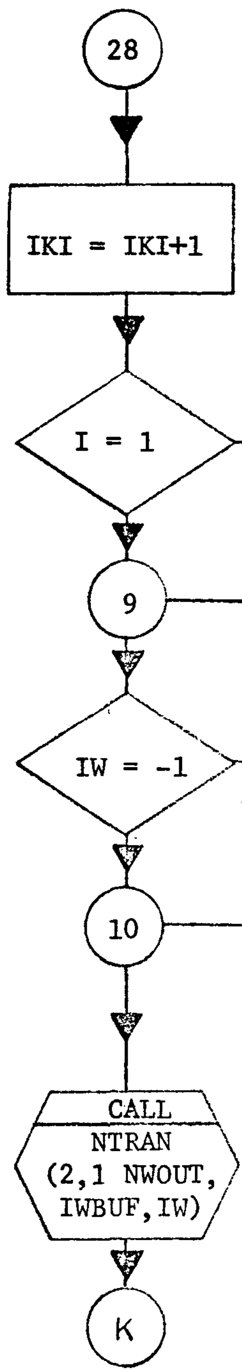


Fig. 9.3.5.2. DETAILED FLOW CHART - FILTER (Continued)

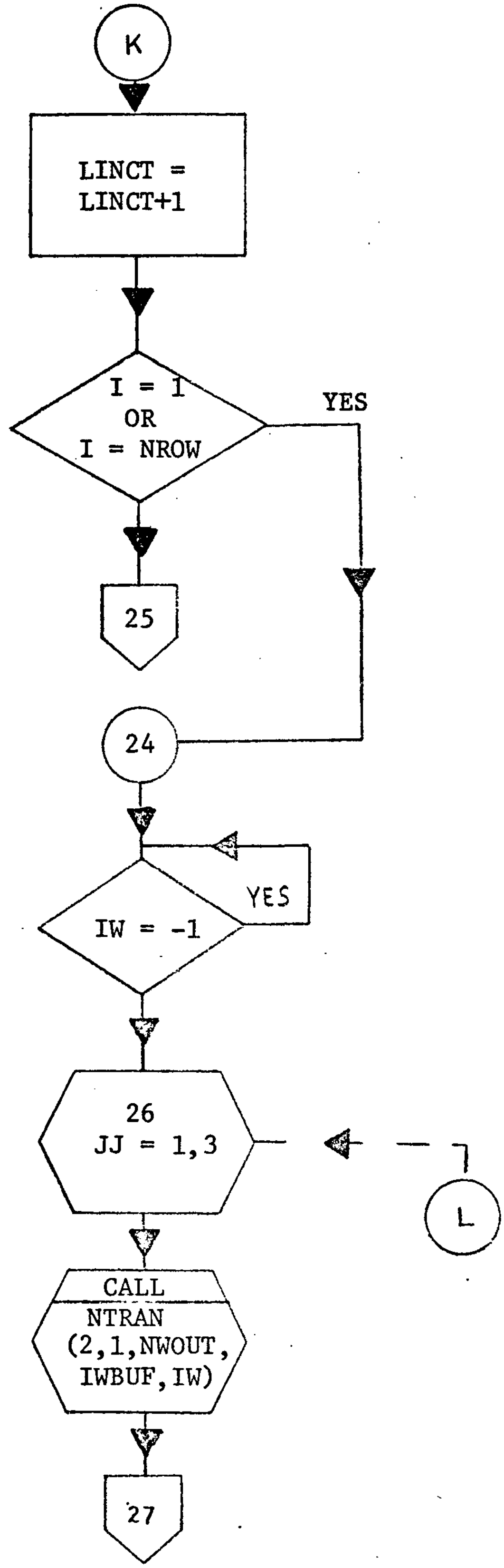


Fig. 9.3.5.2. DETAILED FLOW CHART - FILTER (Continued)

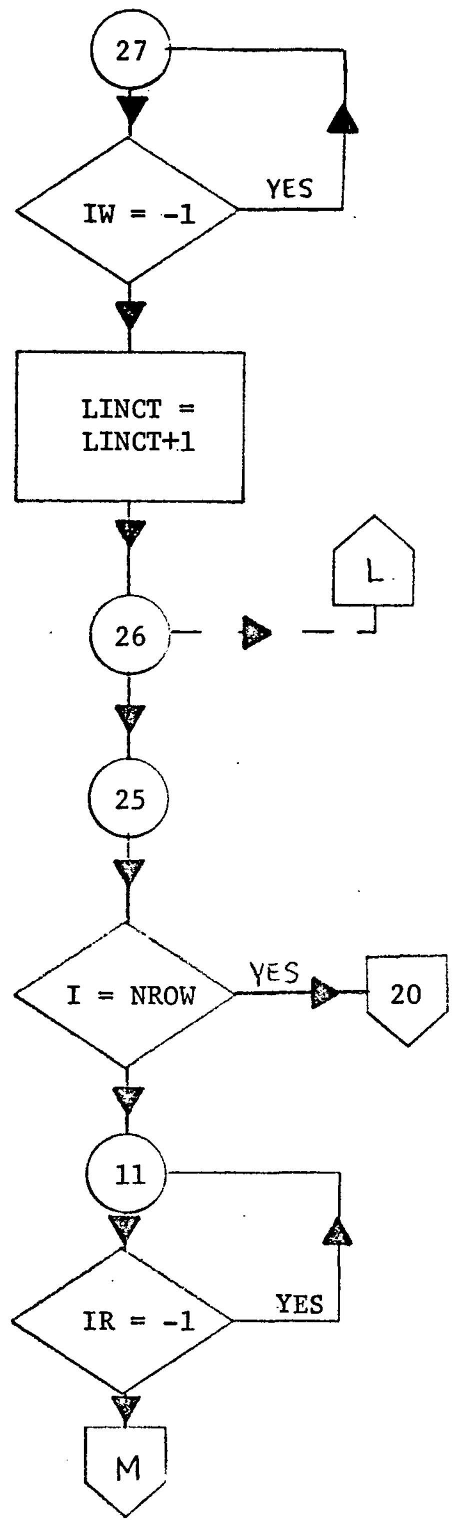


Fig. 9.3.5.2. DETAILED FLOW CHART - FILTER (Continued)

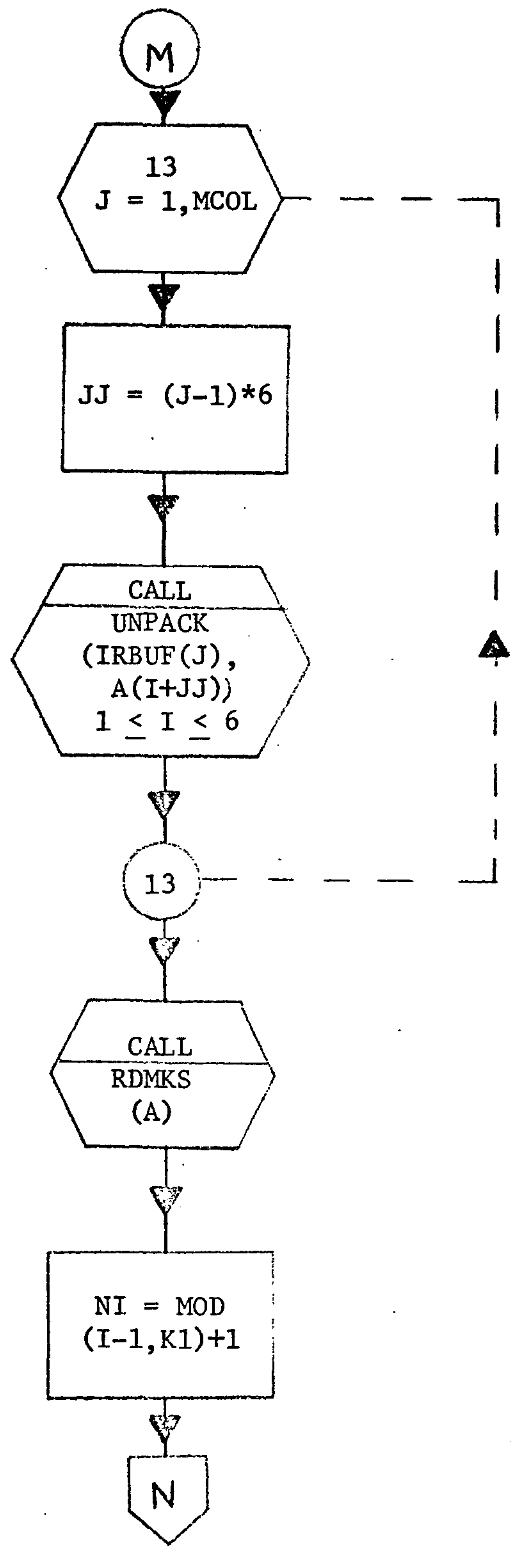


Fig. 9.3.5.2. DETAILED FLOW CHART - FILTER (Continued)

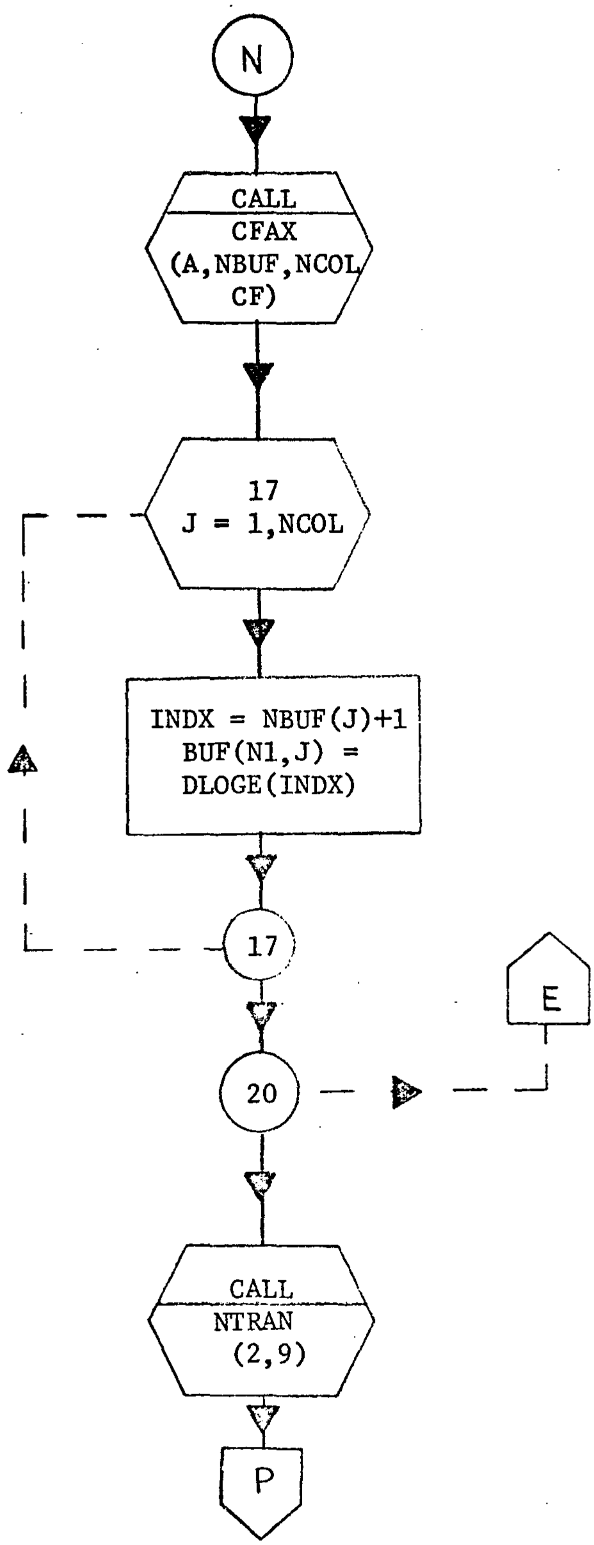


Fig. 9.3.5.2. DETAILED FLOW CHART - FILTER(Continued)

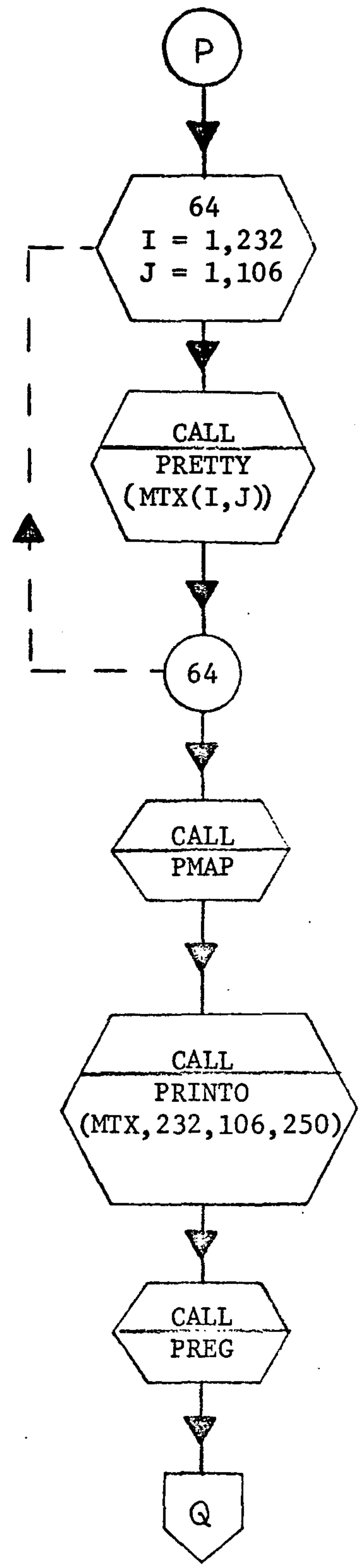


Fig. 9.3.5.2. DETAILED FLOW CHART - FILTER (Continued)

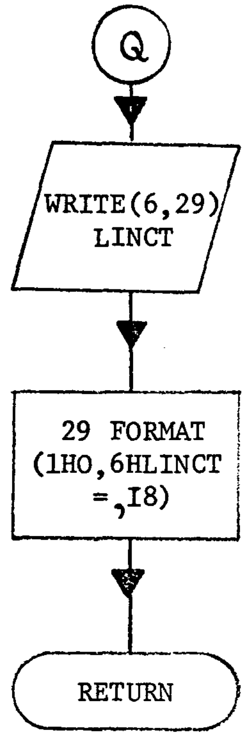
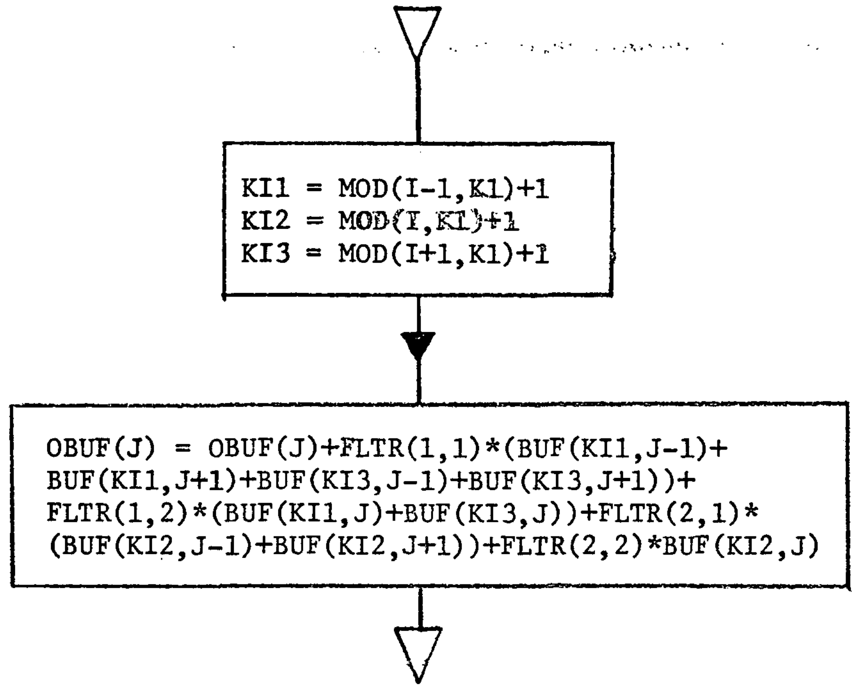


Fig. 9.3.5.2. DETAILED FLOW CHART - FILTER (Concluded)



The inclusion of the above specific modification into the filter program makes use of the symmetry of the filter and reduces the number of multiplications from nine to four. This modification is for K=L=1 to decrease the execution time for a 3x3 filter. The corresponding obvious modification can be made for larger filters, with an even greater gain in efficiency (for example, for a 7x7 filter, instead of 49 multiplications, only 16 will be needed).

9.3.6 SUBROUTINE CFAX

IDENTIFICATION

Name/Title	- CFAX (Correction factors application)
Author/Date	- R. L. Wendt, December 1970
Organization	- ASR
Machine Identification	- UNIVAC 1108
Source Language	- FORTRAN V

PURPOSE/METHOD

The subroutine CFAX applies correction factors as multiplicative factors. The factors given to CFAX come from program CFR (§9.3.3. above).

USAGE

Calling Sequence:

CALL CFAX (A, NBUF, NCOL, CF),

where the variables A, NBUF, NCOL and CF are as described in Table 9.3.2.

DETAILED FLOW CHART - See Fig. 9.3.6.1.

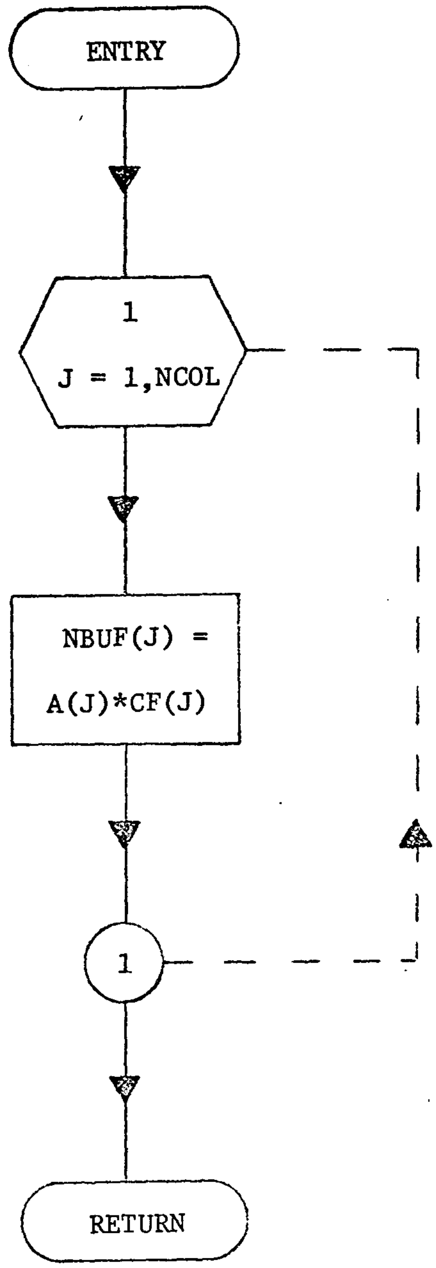


Fig. 9.3.6.1. DETAILED FLOW CHART - CFAX

9.3.7 SUBROUTINE PRETTY

IDENTIFICATION

Name/Title	-	PRETTY
Author/Date	-	This modification: R. L. Wendt, March 1970.
Organization	-	ASR
Machine Identification	-	UNIVAC 1108
Source Language	-	EXEC II ASSEMBLY LANGUAGE

PURPOSE/METHOD

This routine is a modification of the standard LO subroutine of the same name in that two changes are made. First, the original routine failed to protect all index registers; this defect is corrected here, since the filter algorithm requires an unusually large number of index registers. Second, instead of numbers, we here use symbols more directly related to the brightness of the scene. The second difference is indicated as follows:

```
LDD     L, 14     A3, 05
        L, 14     A3, 05
        L, 14     A3, 61
        L, 14     A3, 32
        L, 14     A3, 36
        L, 14     A3, 34
        L, 14     A3, 40
        L, 14     A3, 39
```

(LDD is a reference point for a jump instruction.)

9.3.8 SUBROUTINE PRINTO

IDENTIFICATION

Name/Title	- PRINTO
Author/Date	- This modification: R. L. Wendt, May 1970
Organization	- ASR
Machine Identification	- UNIVAC 1108
Source Language	- FORTRAN V

PURPOSE/METHOD

This routine is a modified version of PRINTM used in the analysis program. The routine prints 20 words per page and over prints every other line.

USAGE

Calling Sequence:

CALL PRINTO (A, NC, NR, MAXR), where A is as described in Table 9.3.2, NC is the number of rows, NR is the number of (packed) elements per row, and MAXR is the maximum of the number of rows. (This routine is designed to fit between PMAP and PREG.)

DETAILED FLOW CHART - See Fig. 9.3.8.1.

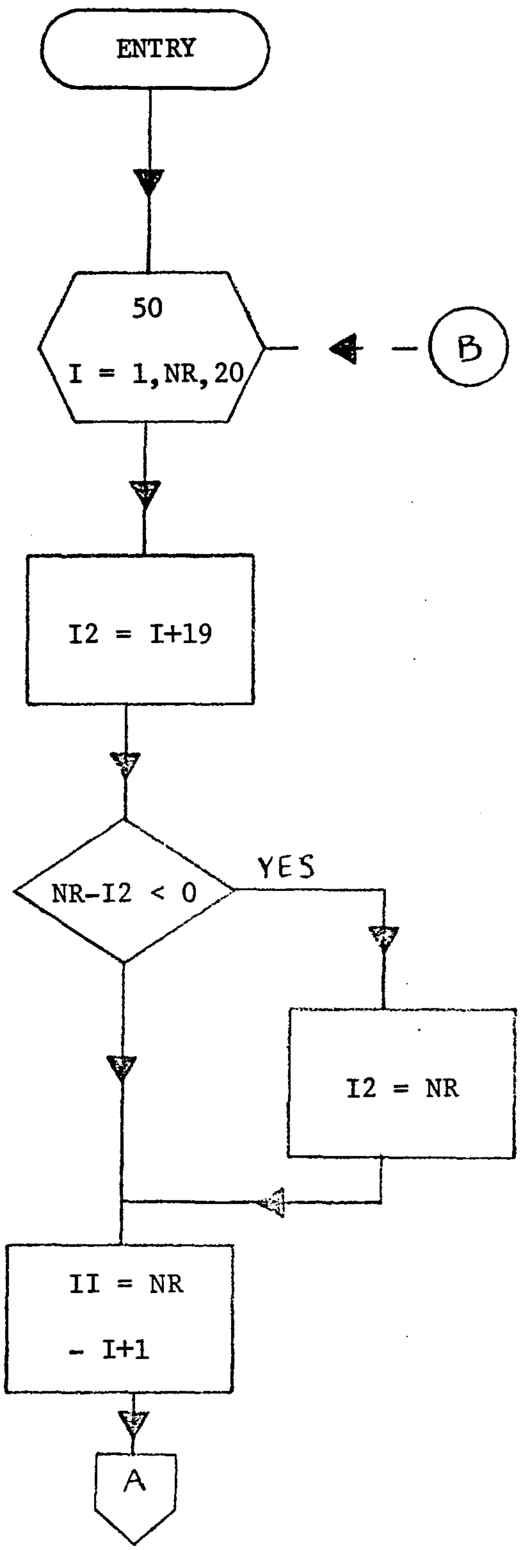


Fig. 9.3.8.1. DETAILED FLOW CHART - PRINTO

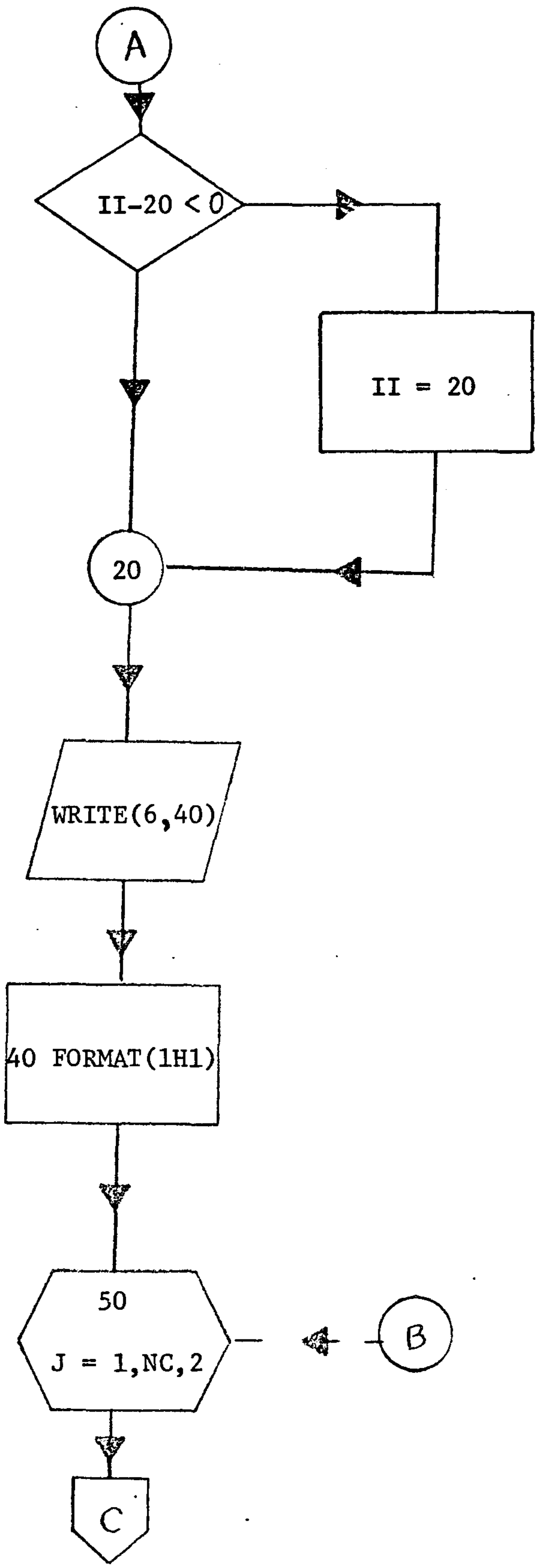


Fig. 9.3.8.1. DETAILED FLOW CHART - PRINTO (Continued)

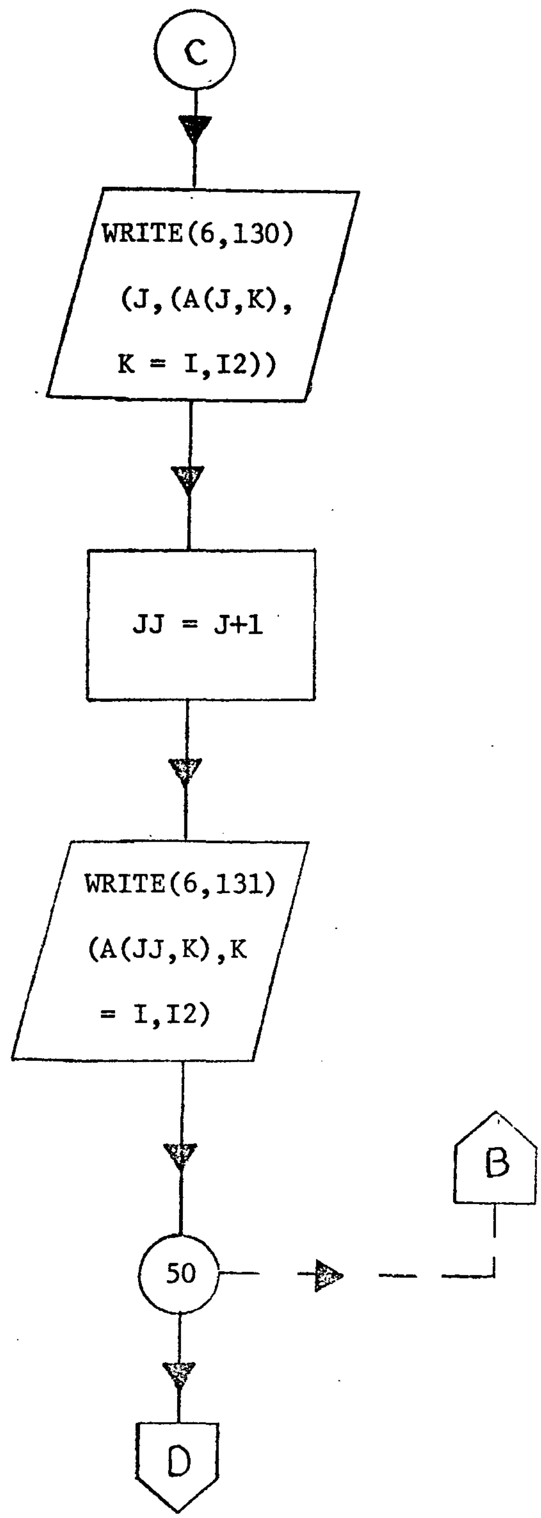


Fig. 9.3.8.1. DETAILED FLOW CHART - PRINTO (Continued)

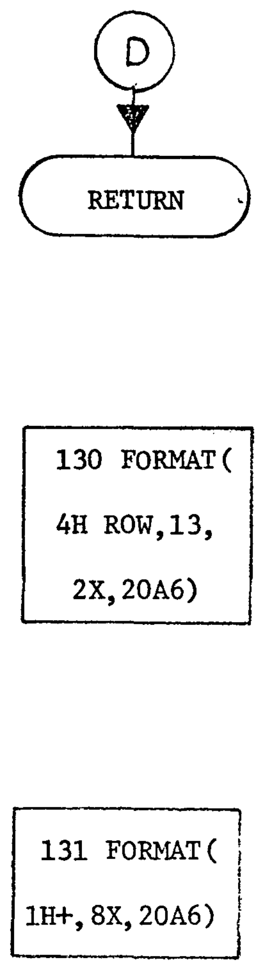


Fig. 9.3.8.1. DETAILED FLOW CHART - PRINTO (Concluded)

COMPUTER PROGRAM DOCUMENTATION

Program COPYT

Project B

by

Jack Bryant

and

R. L. Wendt

Prepared by

Applied Scientific Research, Inc.

Houston, Texas

Under Contract NAS 9-10577

For

MAPPING SCIENCES BRANCH

National Aeronautics and Space Administration

Manned Spacecraft Center

Houston, Texas

September, 1971

10. COMPUTER PROGRAM DOCUMENTATION - COPYT

10.1. PROGRAM DESCRIPTION

This program reads a SEDIT or EDIT tape via NTRAN on unit 1, produces a copy on unit 2 and another copy written one line per record on unit 3.

10.2. USAGE

10.2.1. DESCRIPTION OF INPUT

Tape input - One SEDIT or EDIT tape read from unit 1. There is no other input.

10.2.2. DESCRIPTION OF OUTPUT

Tape output - The principle output is two tapes; one tape (on unit 2) is a copy of the input tape. The other tape (from unit 3) is a copy with one 106 word line per record instead of 94 106 word lines per record. This tape is used as input for the filter program.

Printed output - The header is printed, this being the only normal output. In case of various errors, the appropriate status words are printed.

10.2.3. RUN PREPARATION, EXECUTION CHARACTERISTICS

Card deck setup - See Fig. 10.2.3.1.

Subroutine requirements - NTRAN to handle tapes.

Subroutine OLPRSC, see §10.3.3.

Run time - The program runs at tape read/write speed.

10.3. REFERENCE INFORMATION

DETAILED FLOW CHART - See Fig. 10.3.1.

DESCRIPTION OF VARIABLES - See Table 10.3.2.

DESCRIPTION OF SUBROUTINE - See §10.3.3.

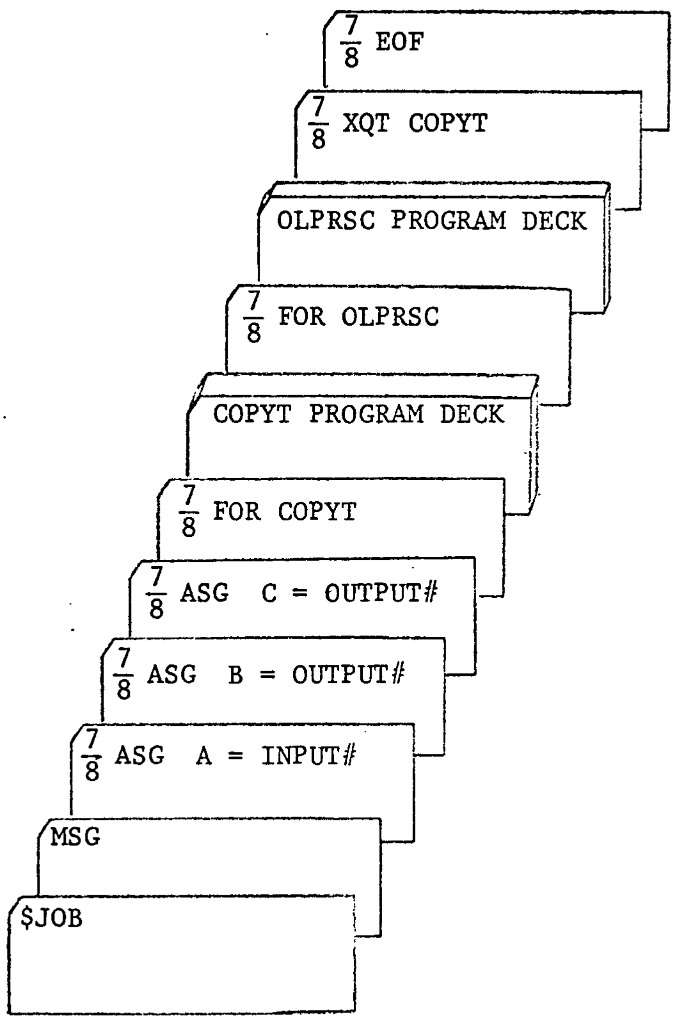


Fig. 10.2.3.1. DECK SETUP (UNIVAC 1108) - COPYT.

I - Index on the number of framelets per reel.

N - Index on number of files per framelet including header.

IB - Array used in I/O by NTRAN.

IR,IW - Status words from NTRAN.

J - Index used in subroutine OLPRSC for writing the one line records of LO data.

Table 10.3.2. DESCRIPTION OF PROGRAM VARIABLES - COPYT

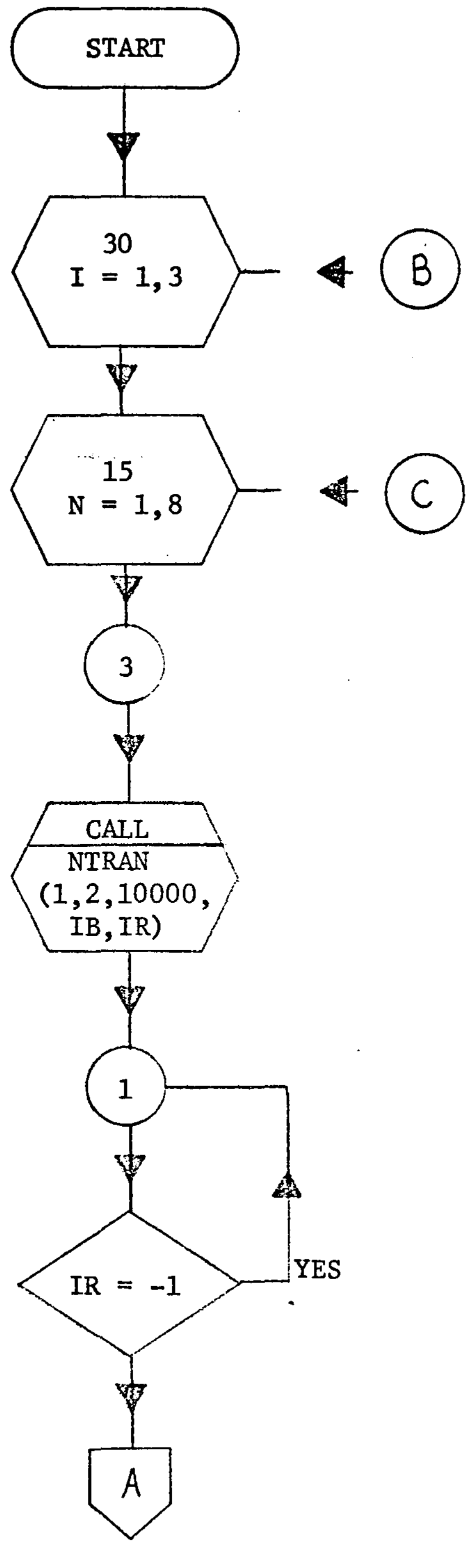


Fig. 10.3.1. DETAILED FLOW CHART - COPYT

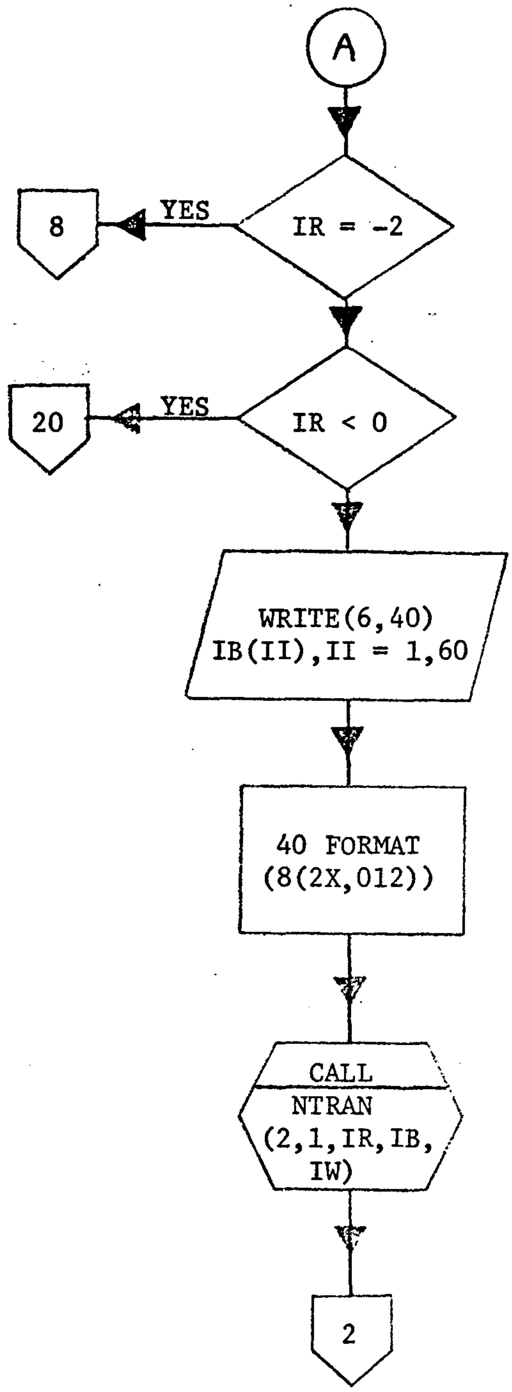


Fig. 10.3.1. DETAILED FLOW CHART - COPYT (Continued)

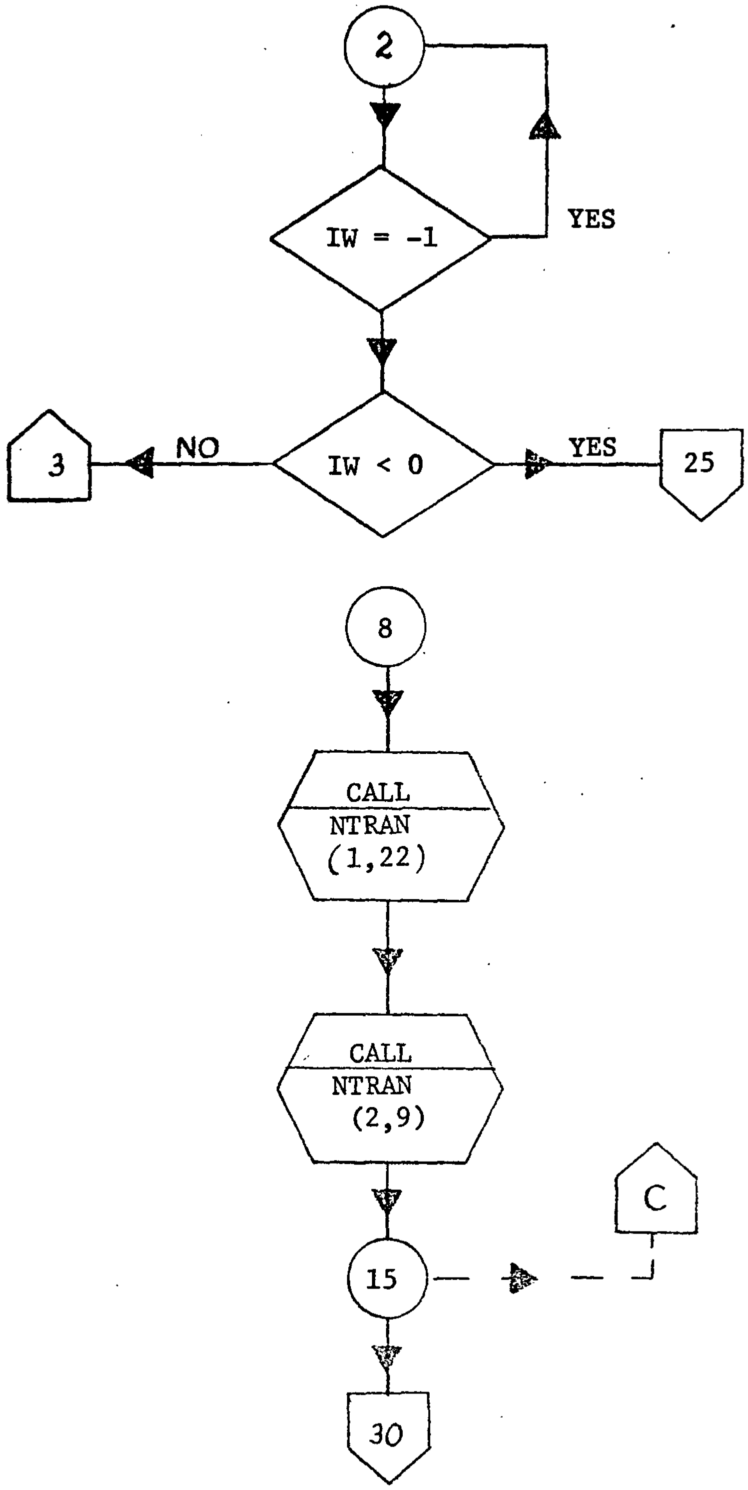


Fig. 10.3.1. DETAILED FLOW CHART - COPYT (Continued)

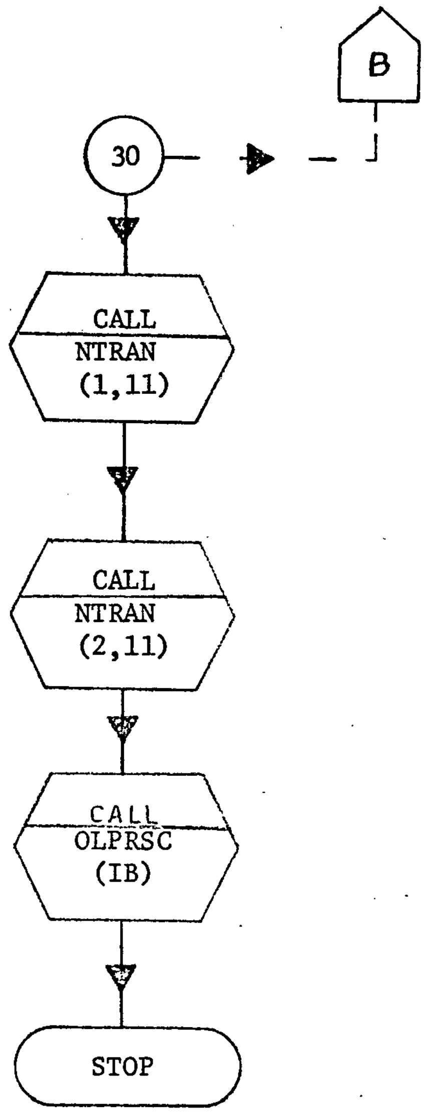


Fig. 10.3.1. DETAILED FLOW CHART - COPYT (Continued)

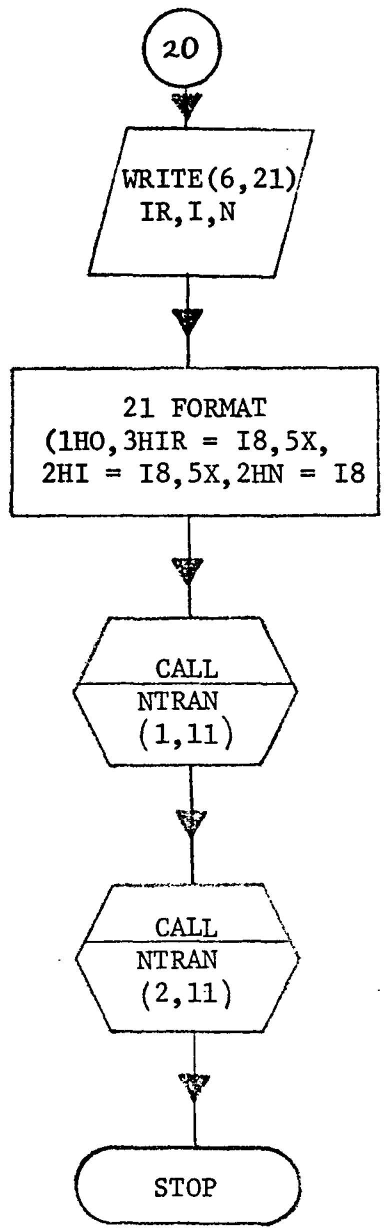


Fig. 10.3.1. DETAILED FLOW CHART - COPYT (Continued)

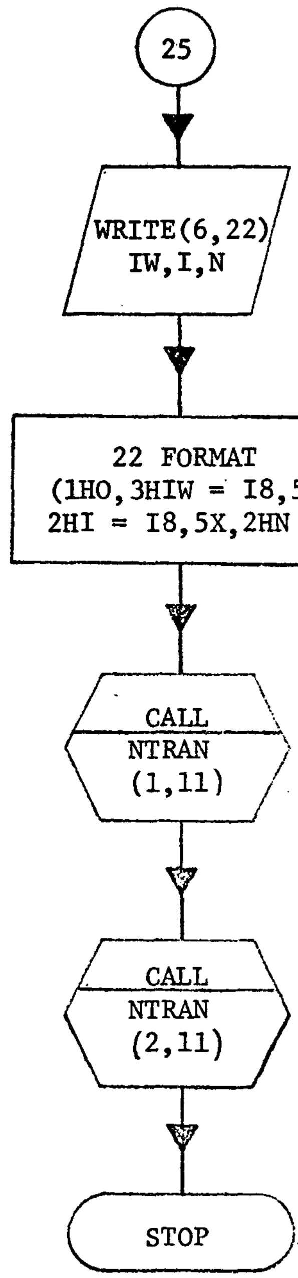


Fig. 10.3.1. DETAILED FLOW CHART - COPYT (Concluded)

10.3.3. SUBROUTINE OLPRSC

IDENTIFICATION

Name/Title	- OLPRSC (One line per record SEDIT copy)
Author/Date	- R. L. Wendt, August 1971
Organization	- ASR
Machine Identification	- UNIVAC 1108
Source Language	- FORTRAN V

PURPOSE/METHOD

This subroutine produces a copy of the tape on unit 2, written in 94 line records, on unit 3 written one line per record. The header and general format of the output tape are the same as the input tape.

USAGE

Calling sequence:

CALL OLPRSC (IB)

The variable IB, along with other variables used by OLPRSC, are described in Table 10.3.2.

DETAILED FLOW CHART - See Fig. 10.3.3.1.

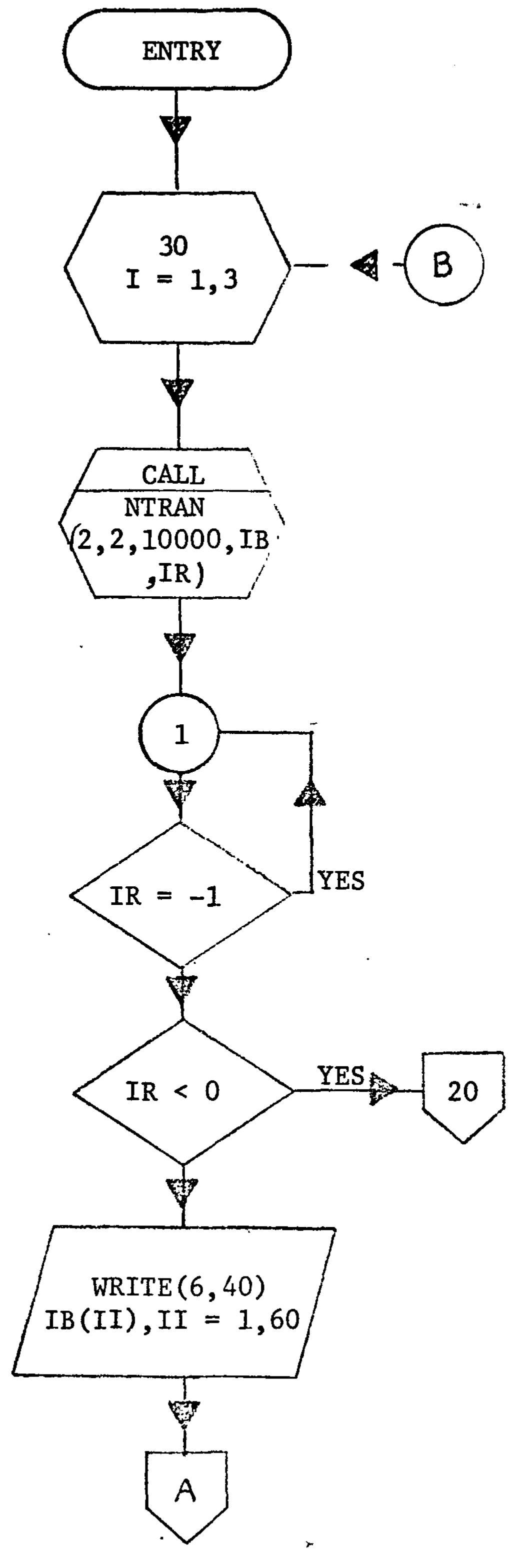


Fig. 10.3.3.1. DETAILED FLOW CHART - OLPRSC

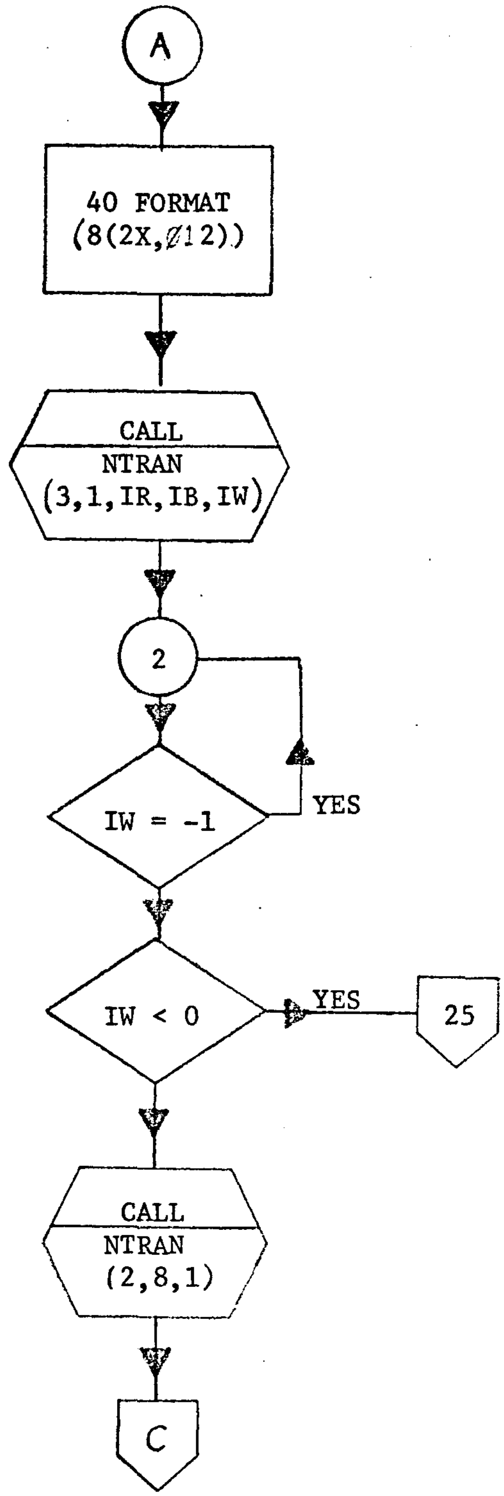


Fig. 10.3.3.1. DETAILED FLOW CHART - OLPRSC (Continued)

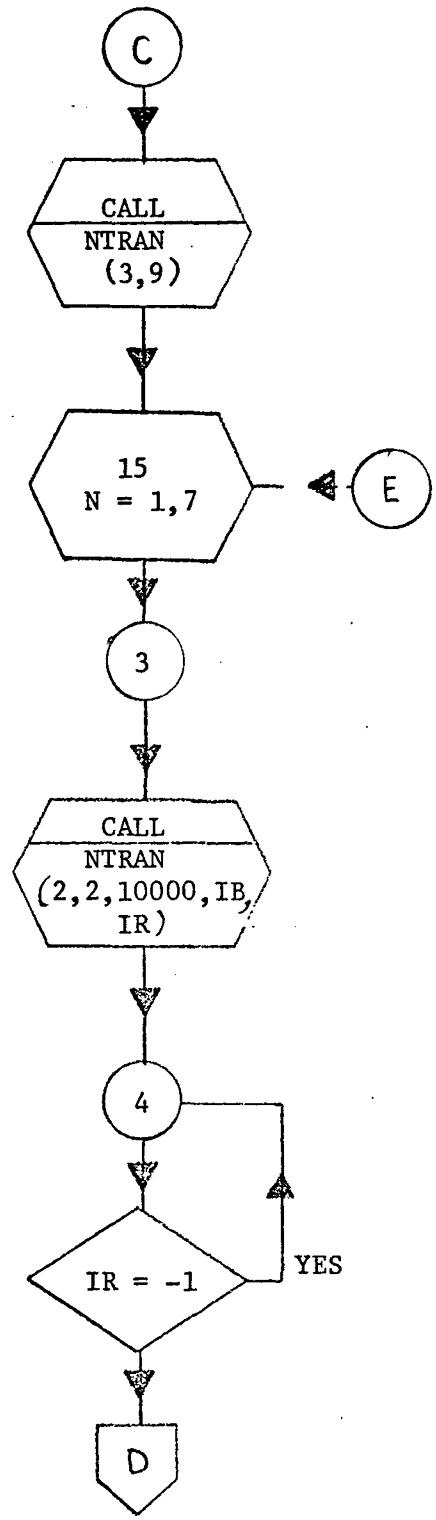


Fig. 10.3.3.1. DETAILED FLOW CHART - OLPRSC (Continued)

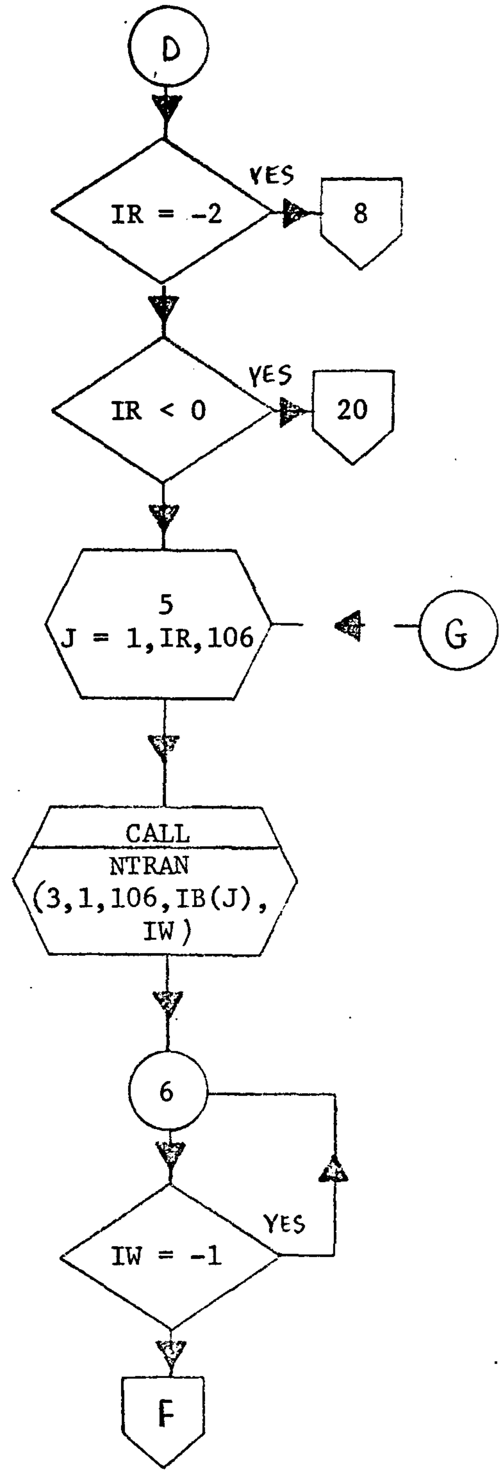


Fig. 10.3.3.1. DETAILED FLOW CHART - OLPRSC (Continued)

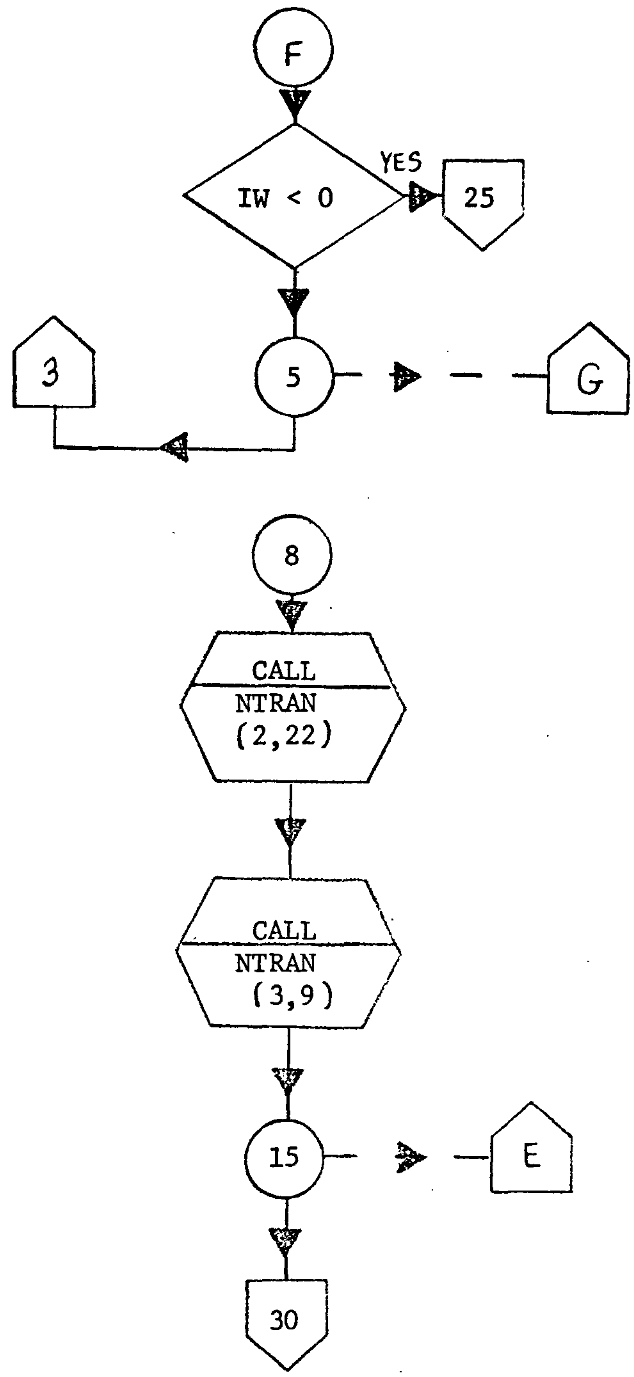


Fig. 10.3.3.1. DETAILED FLOW CHART - OLPRSC (Continued)

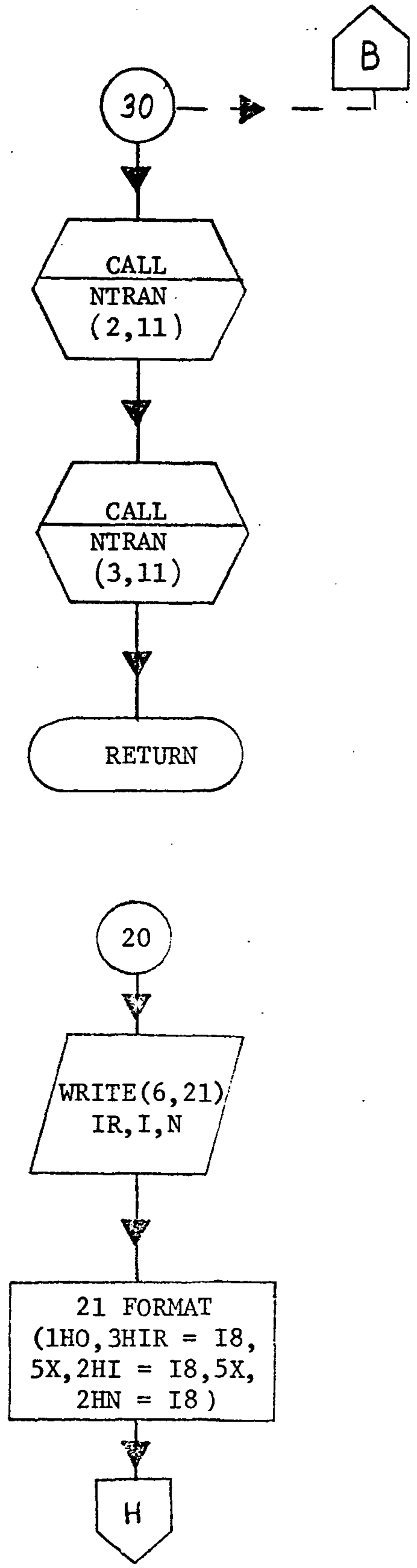


Fig. 10.3.3.1.DETAILED FLOW CHART - OLPRSC (Continued)

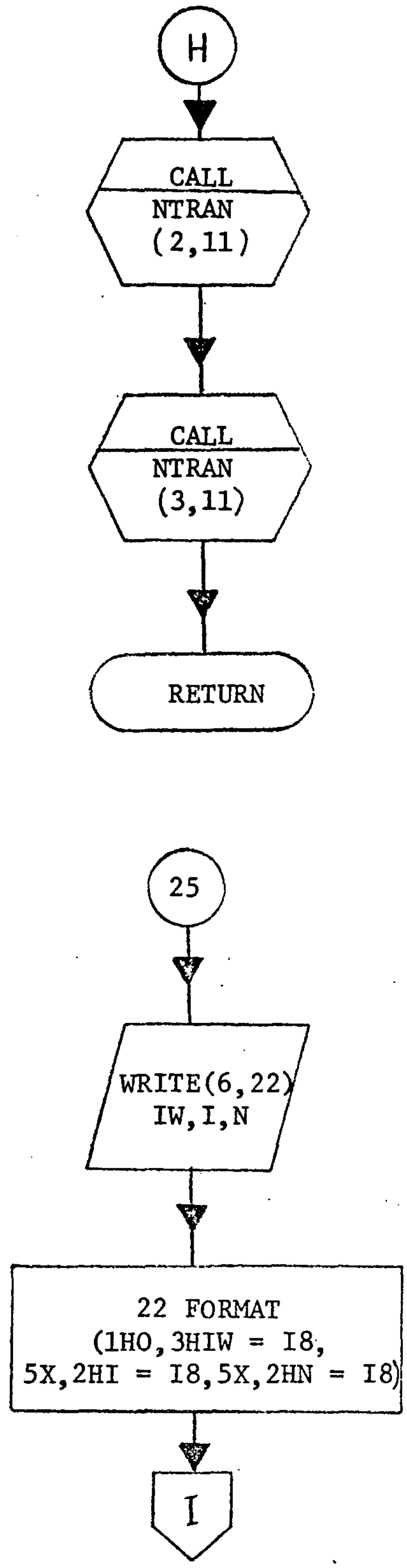


Fig. 10.3.3.1. DETAILED FLOW CHART - OLPRSC (Continued)

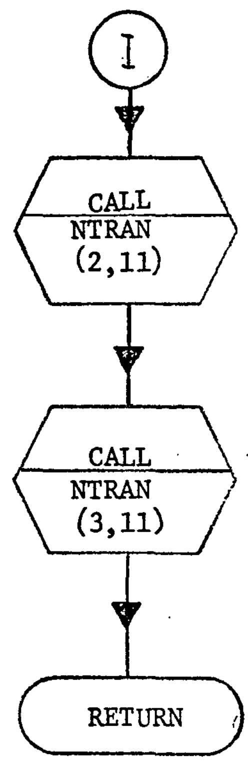


Fig. 10.3.3.1. DETAILED FLOW CHART - OLPRSC (Concluded)

COMPUTER PROGRAM DOCUMENTATION

MERGE NOTCH, DATA, EDGE DATA

Program MRGNDE

Project A

by

Jack Bryant

and

R. L. Wendt

Prepared by

Applied Scientific Research, Inc.

Houston, Texas

Under Contract NAS 9-10577

For

MAPPING SCIENCES BRANCH

National Aeronautics and Space Administration

Manned Spacecraft Center

Houston, Texas

August, 1971

11. COMPUTER PROGRAM DOCUMENTATION - MRGNDE

11.1. PROGRAM DESCRIPTION

This program,while belonging properly to Project A, is reported on here because of the strong connection with §§10 and 12, in that its main function is to read an input tape and write three tapes with the ultimate purpose being to write the data one line per record while preserving the header and general structure of the tape.

The program begins by initializing parameters and counters. The header of each framelet is read and written on the output tape as it occurs followed by an end of file mark.

Records of 94 lines in length are read from the input tape and are written on the output tape a line at a time. A count of the number of lines is recorded. As the end of file marks are encountered for the notch, sub framelets and edge data a counter is incremented and compared against the number of data blocks for each framelet. When equality occurs an end of file mark is written on the output tape, the second end of file mark after the edge data is skipped, the output unit swaps reels and rewinds. In this case the process is repeated for the remaining framelets in 027, 025 where 028 is the first framelet on the edited tape. When the process is complete both units, input and output,are rewound.

This process gives the output tapes the same format (with the exception of one-line per record) as full framelet tapes we made from SEDIT tapes, and makes their use in the JOIN program possible. Initially, these tapes were also used with the tests of the filter program. For a description of the format of the tapes made here, see the report on Project A, §10.4.1.2, p. 147.

11.2. USAGE

11.2.1. DESCRIPTION OF INPUT

Data tape from Surveyor 3, LO III, FR 154 H

Framelets 028, 027,025 Edited.

11.2.2. DESCRIPTION OF OUTPUT

Tape output - Three tapes; merged data tape one line per record, used as data tape for JOIN program.

Printed output - The only normal output in the value of the line counter at the completion of the data read. In the event of an error on read or write, the appropriate status word is written.

11.2.3. RUN PREPARATION

Card deck setup - See Fig. 11.2.3.1.

Subroutine requirements - NTRAN tape handling routine.

11.2.4. EXECUTION CHARACTERISTICS

Run time - The program runs at tape read/write speed.

11.3. GENERAL FLOW CHART

See Fig. 11.3.1.

11.4. REFERENCE INFORMATION

11.4.1. DETAILED FLOW CHART

See Fig. 11.4.1.1.

11.4.2. DESCRIPTION OF VARIABLES

See Table 11.4.2.1.

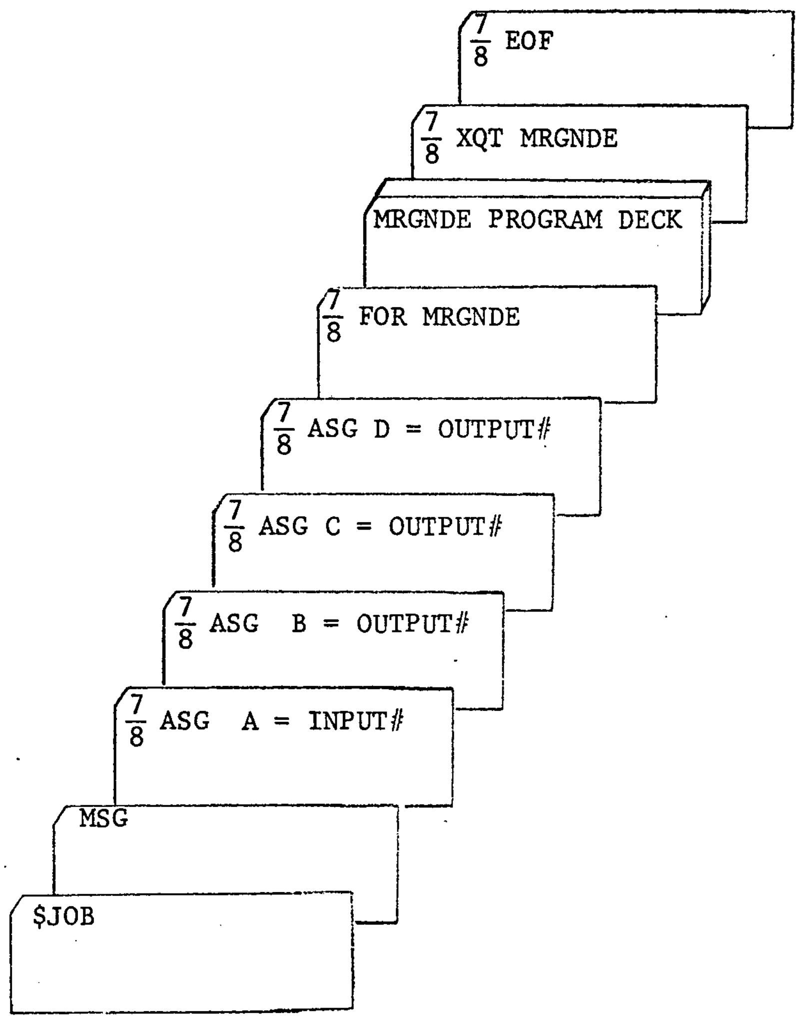


Fig. 11.2.3.1. DECK SETUP (UNIVAC 1108) - MRGNDE

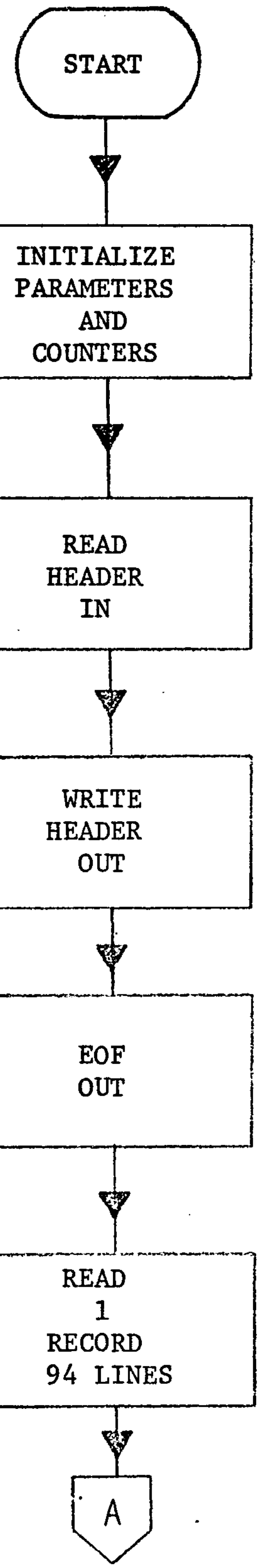


Fig. 11.3.1. GENERAL FLOW CHART - MRGNDE

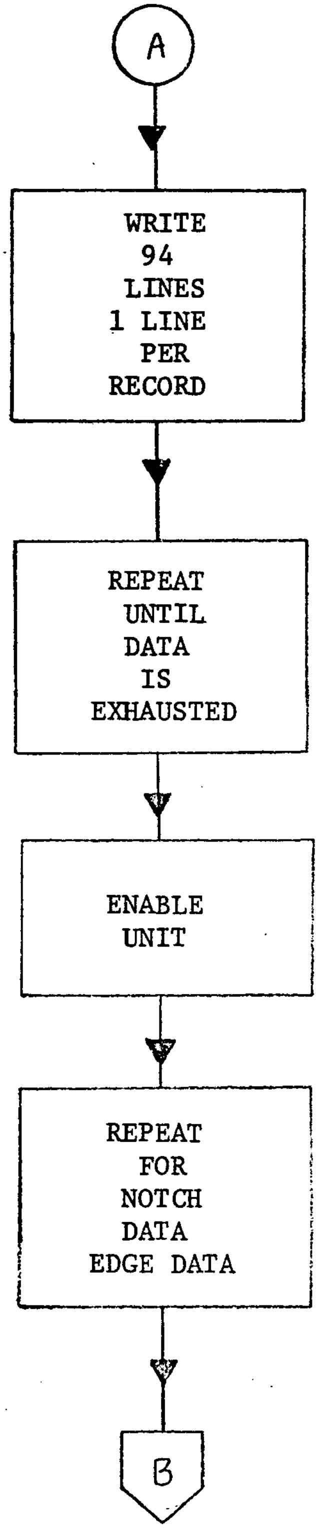


Fig. 11.3.1. GENERAL FLOW CHART - MRGNDE (Continued)

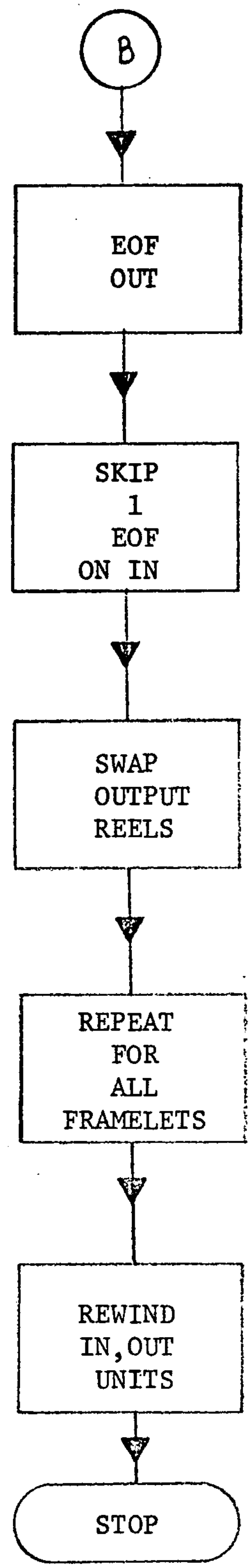


Fig. 11.3.1. GENERAL FLOW CHART - MRGNDE (Concluded)

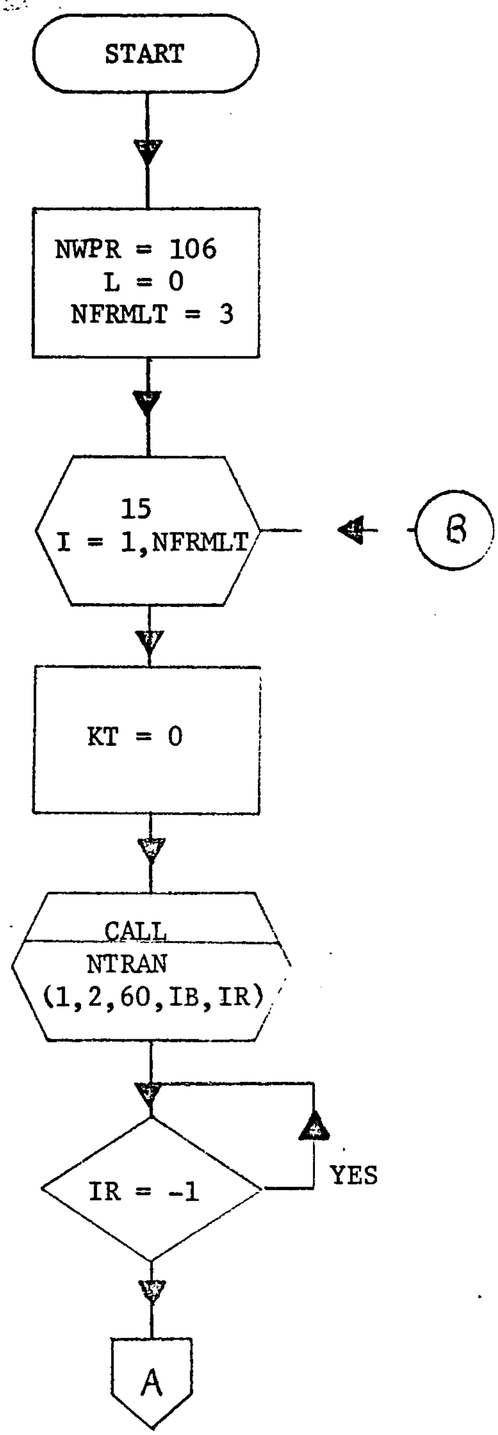


Fig. 11.4.1.1. DETAILED FLOW CHART - MRGNDE

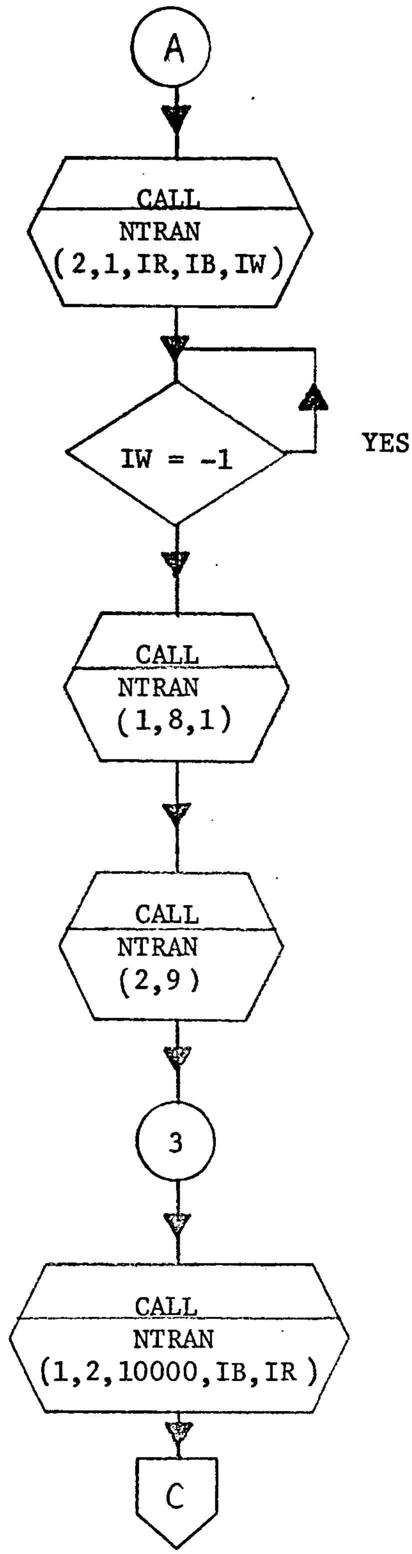


Fig. 11.4.1.1. DETAILED FLOW CHART - MRGNDE (Continued)

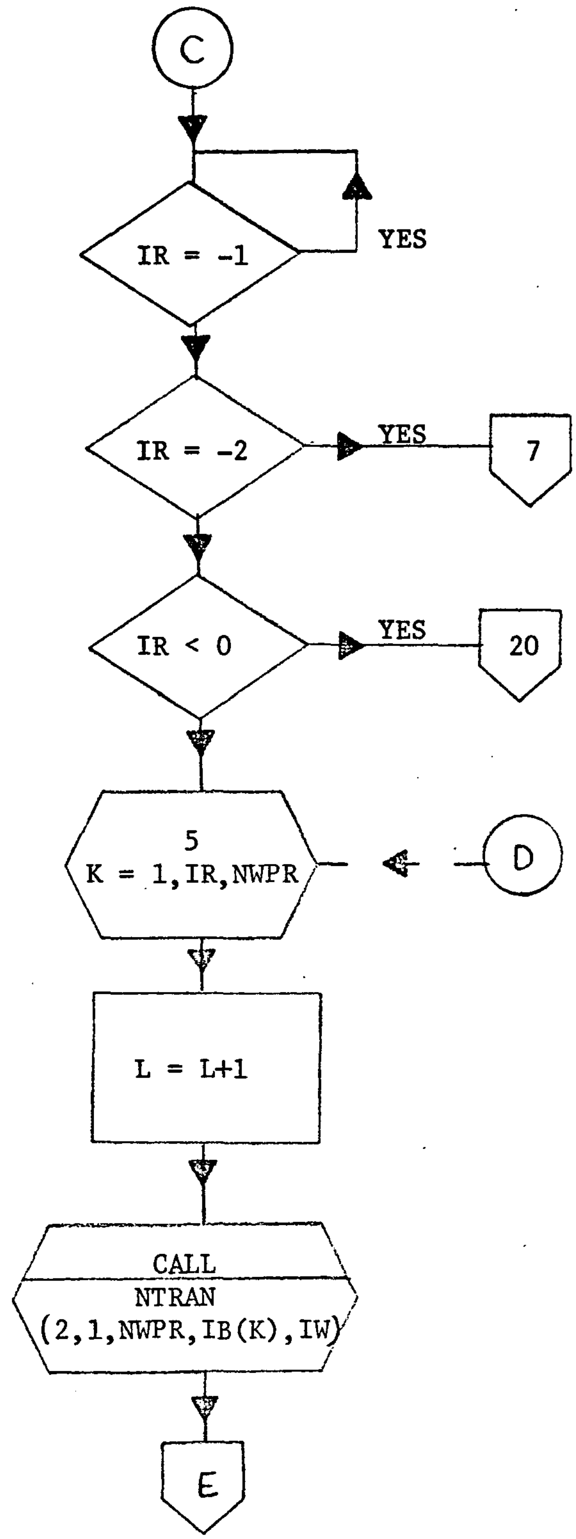


Fig. 11.4.1.1. DETAILED FLOW CHART - MRGNDE (Continued)

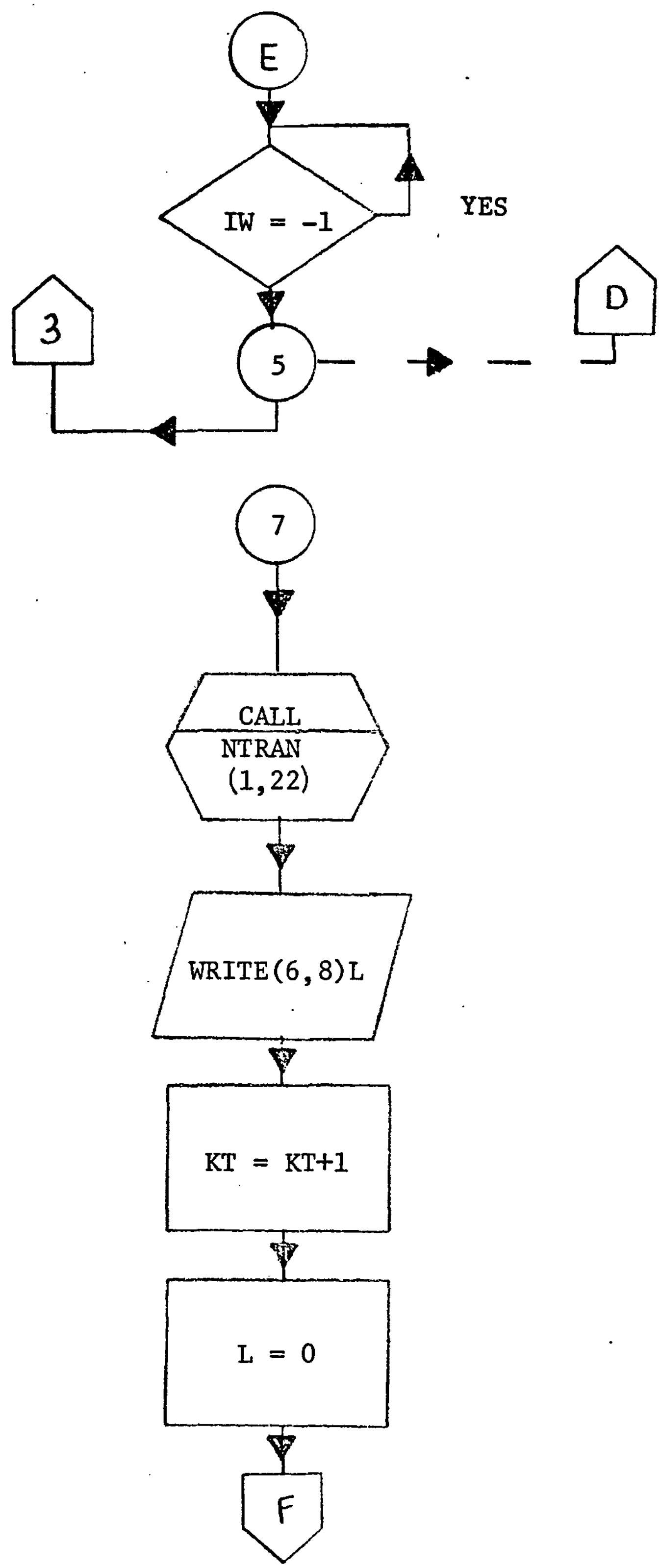


Fig. 11.4.1.1. DETAILED FLOW CHART - MRGNDE (Continued)

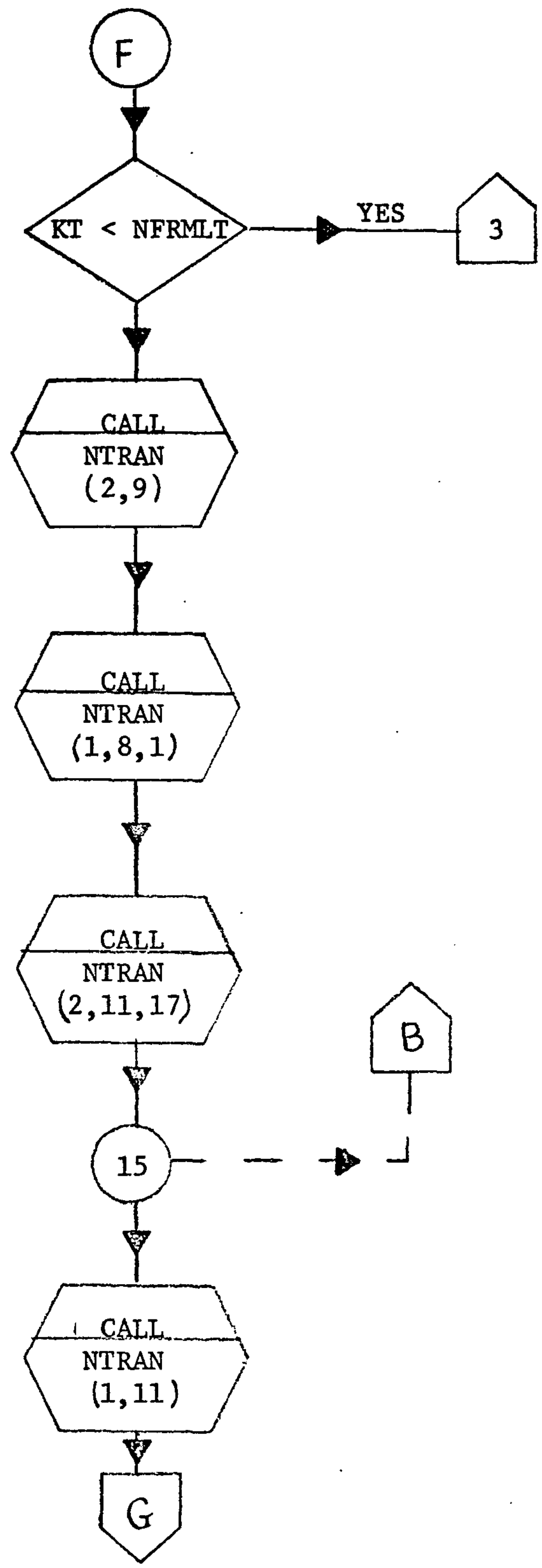


Fig. 11.4.1.1. DETAILED FLOW CHART - MRGNDE (Continued)

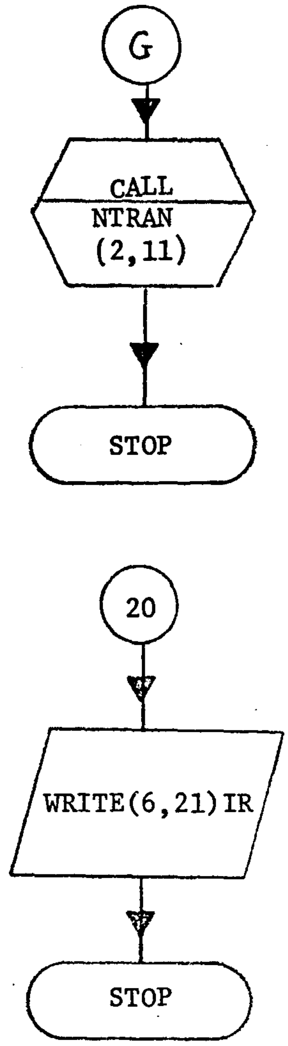


Fig. 11.4.1.1. DETAILED FLOW CHART - MRGNDE (Concluded)

NWPR - Number of words per row of framelet.

NFRMLT - Number of framelets.

L - Line counter.

KT - Framelet counter.

IB - Array; used in read and write by NTRAN.

IR,IW - Status words; input and output for NTRAN.

Table 11.4.2.1. DESCRIPTION OF VARIABLES - MRGNDE

COMPUTER PROGRAM DOCUMENTATION

MERGE SEDIT OR EDIT WITH FILTERED TAPE

Program MCSF

Project B

by

Jack Bryant

and

R. L. Wendt

Prepared by

Applied Scientific Research, Inc.

Houston, Texas

Under Contract NAS 9-10577

For

MAPPING SCIENCES BRANCH

National Aeronautics and Space Administration

Manned Spacecraft Center

Houston, Texas

August, 1971

12. COMPUTER PROGRAM DOCUMENTATION - MCSF

12.1. PURPOSE

This program merges a SEDIT or EDIT tape with output of the filter program into a data tape which is suitable for use by the Lunar Orbiter Analysis Program. The merged tape has one chit which is filtered to be processed by the Analysis Program to study the contour map. All data corrections are performed by the filter program (see §9) and thus need not be done in the analysis program.

12.2. USAGE

12.2.1. DESCRIPTION OF INPUT

Tape input: Unit 1 - SEDIT or EDIT tape.

Unit 2 - Output tape from FILTER (§9).

Card input: IFILE - The file number of the filtered chit.

12.2.2. DESCRIPTION OF OUTPUT

Tape output: Unit 3 - Merged tape.

Printed output: The header is printed, normally. Should errors in tape format or read/write appear, the appropriate status word is written.

12.2.3. RUN PREPARATION

Card deck setup - See Fig. 12.2.3.1.

Required subroutines - NTRAN tape handling routine.

12.2.4. EXECUTION CHARACTERISTICS

The program executes at tape read/write speed; most of the time charged for the job is probably spent in hanging tapes.

12.3. REFERENCE INFORMATION

DETAILED FLOW CHART - See Fig. 12.3.1.

DESCRIPTION OF VARIABLES - See Table 12.3.2.

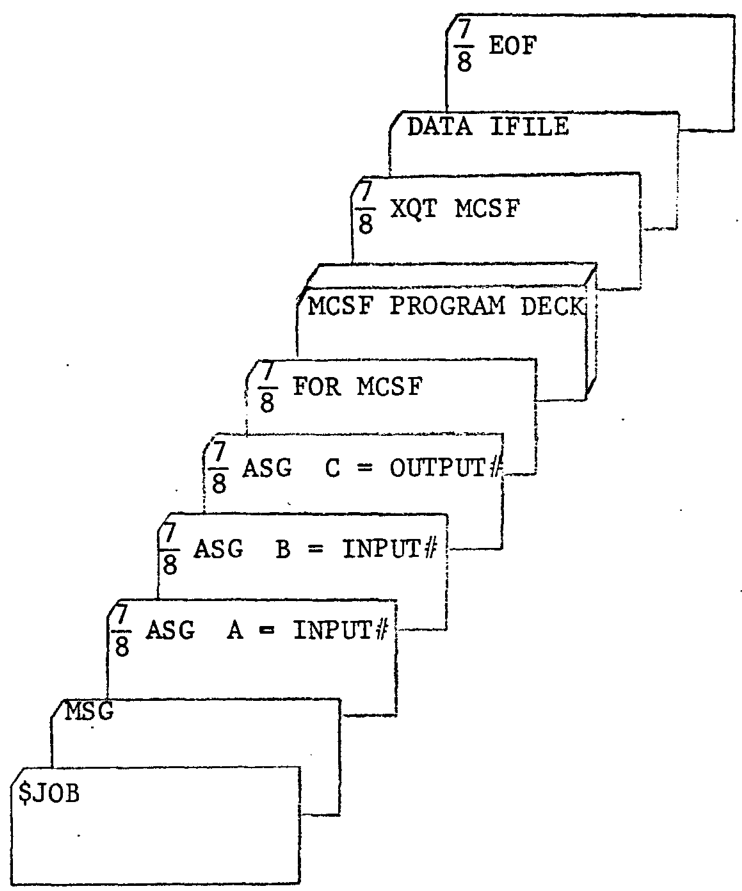


Fig. 12.2.3.1. CARD DECK SETUP (UNIVAC 1108) - MCSF

IB - Array used both on input and output.

KB - Array used in reading filter output tape and used to pack IB with 94 lines before writing records.

I - Index on number of files per tape.

IFILE - The file number of the filtered chit.

INDX,INDX1 - Indices used in packing the filtered chit into IB to provide records of 94 lines.

NWDS - The residual record of a chit,not being a full 94 lines, has upon reading a file mark in loop 16;(N-1)*106 words or N-1 lines; usually 80 lines if a chit is 738 lines.

Table 12.3.2. DESCRIPTION OF VARIABLES - MCSF

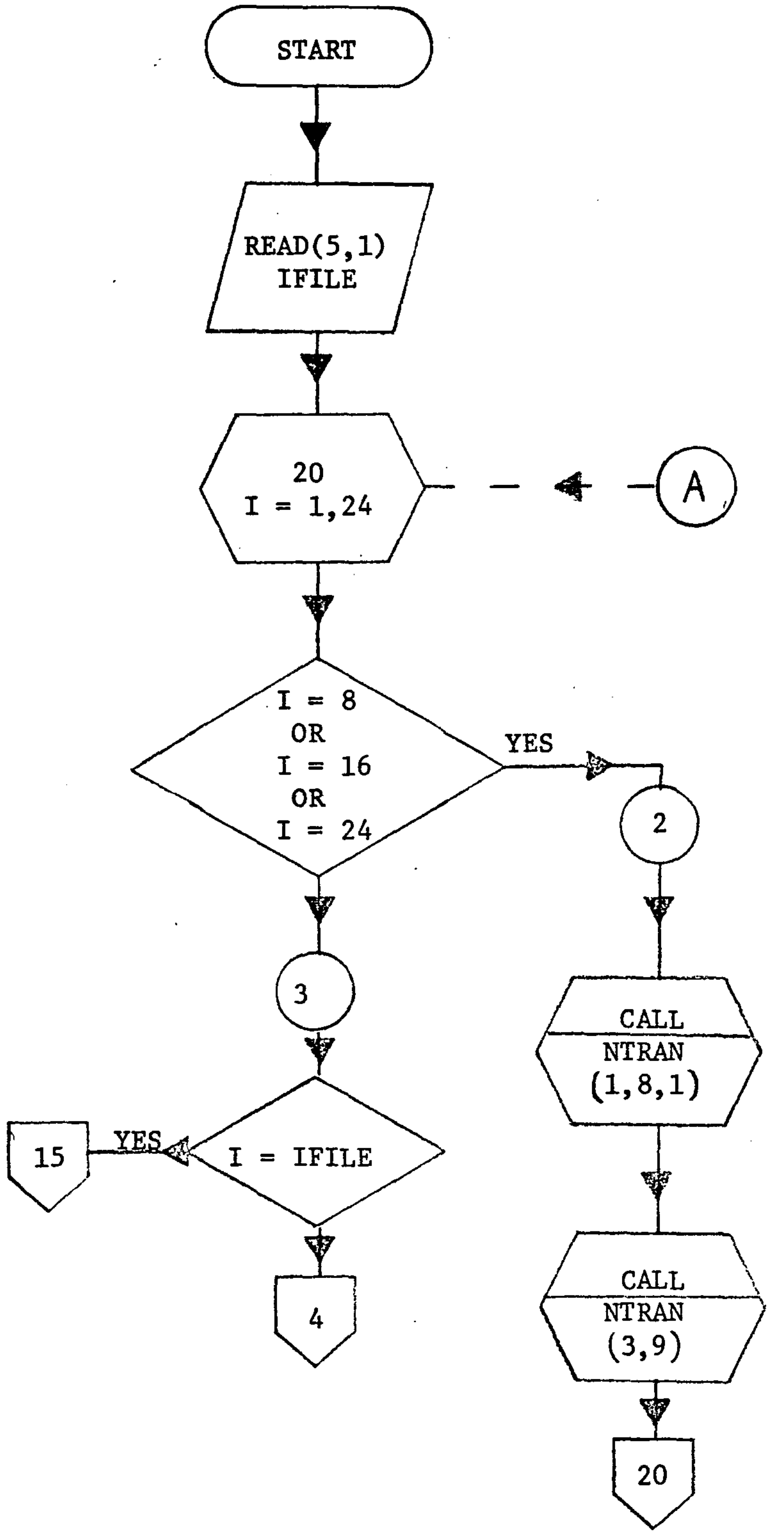


Fig. 12.3.1. DETAILED FLOW CHART - MCSF

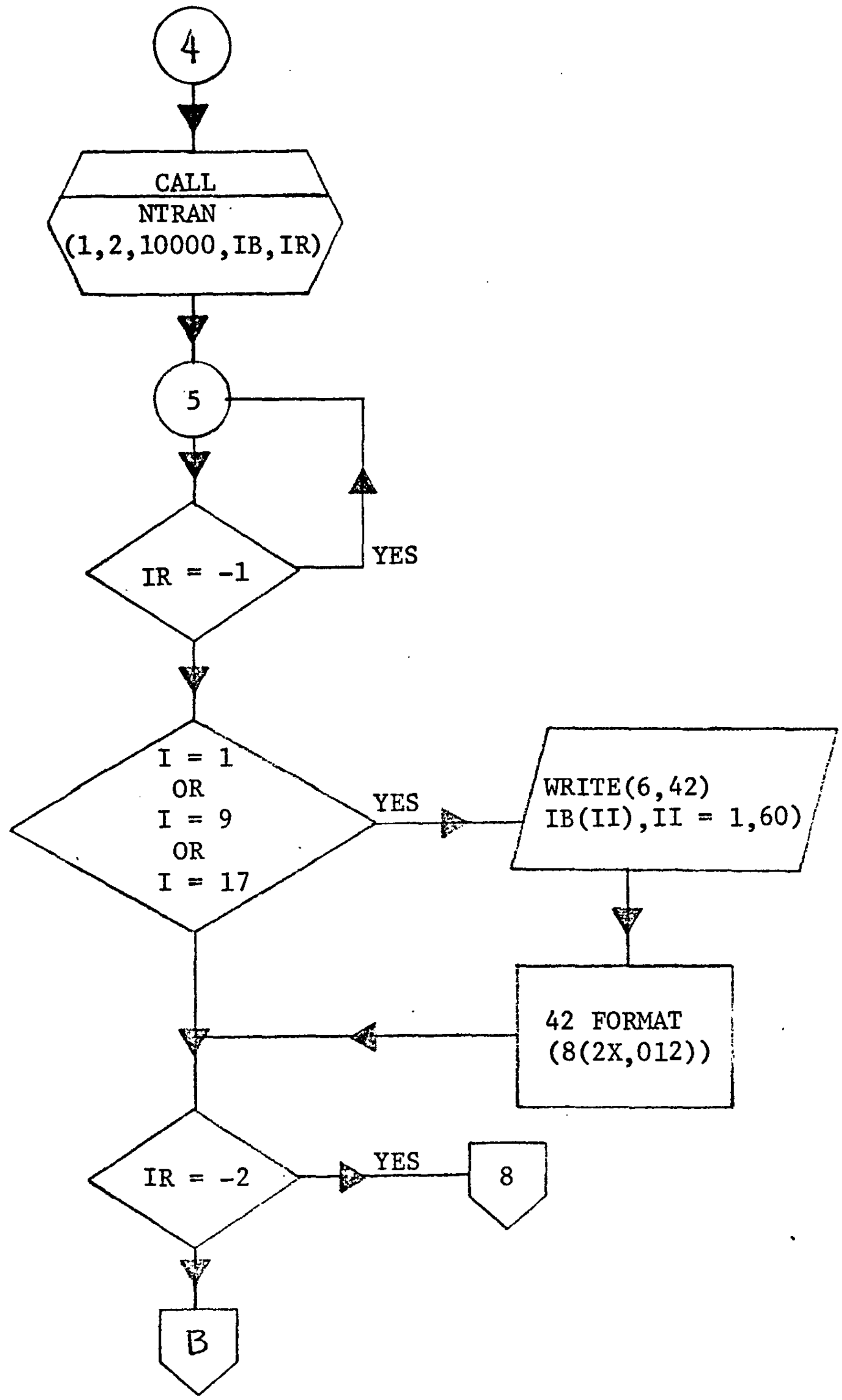


Fig. 12.3.1. DETAILED FLOW CHART - MCSF (Continued)

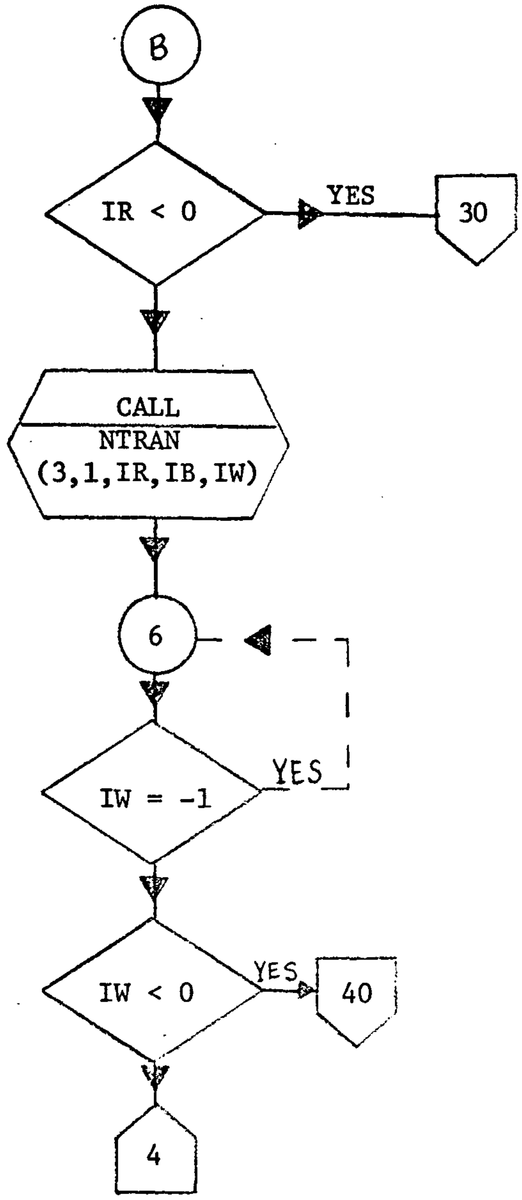


Fig. 12.3.1. DETAILED FLOW CHART - MCSF (Continued)

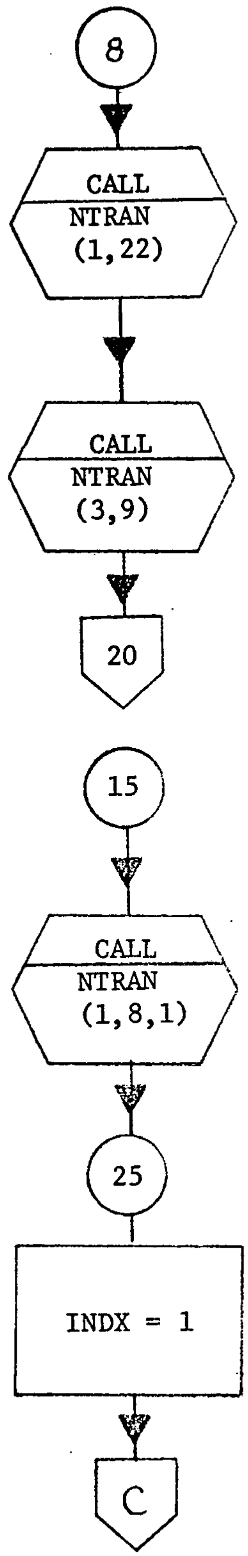


Fig. 12.3.1. DETAILED FLOW CHART - MCSF (Continued)

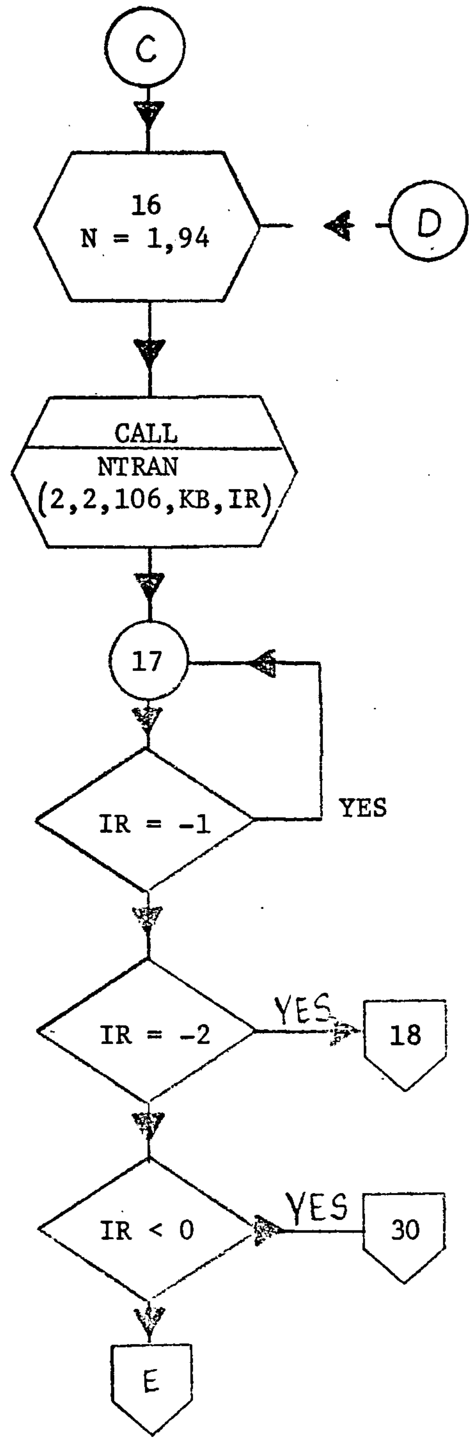


Fig. 12.3.1. DETAILED FLOW CHART - MCSF (Continued)

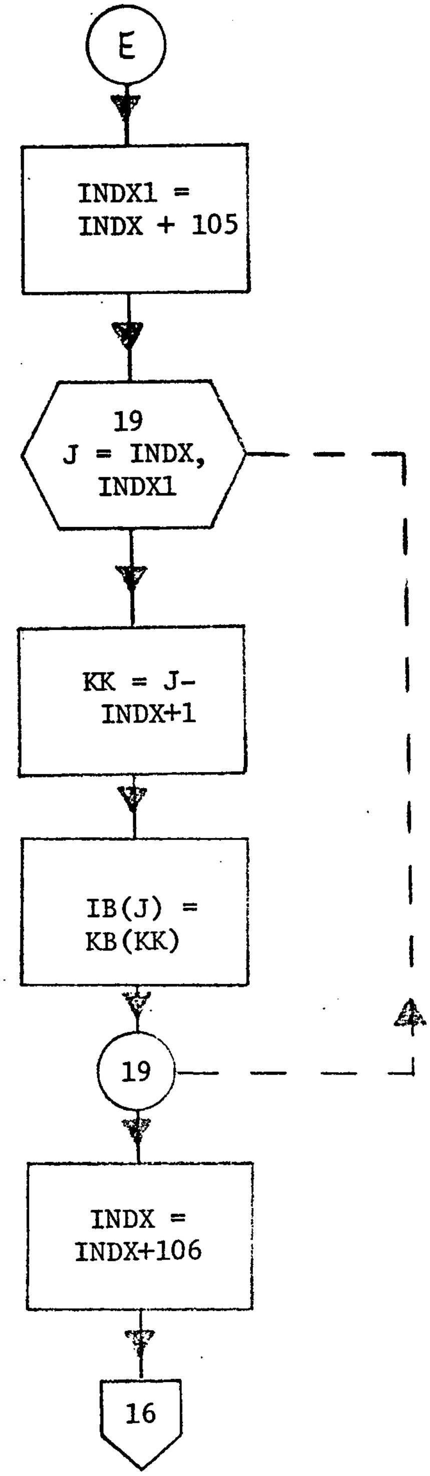


Fig. 12.3.1. DETAILED FLOW CHART - MCSF (Continued)

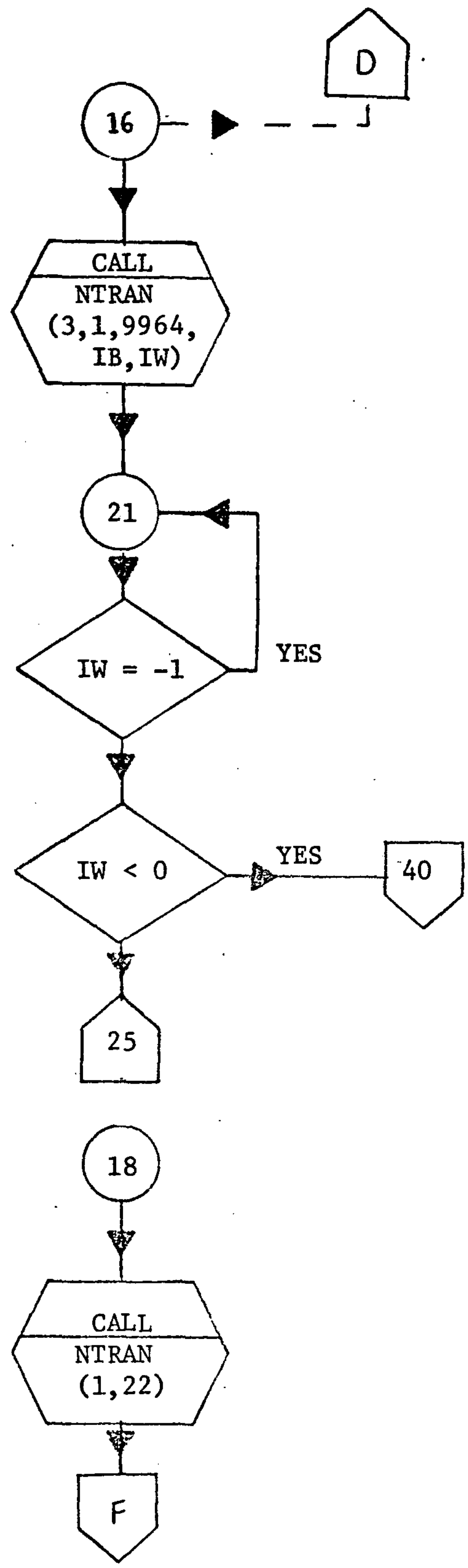


Fig. 12.3.1. DETAILED FLOW CHART - MCSF (Continued)

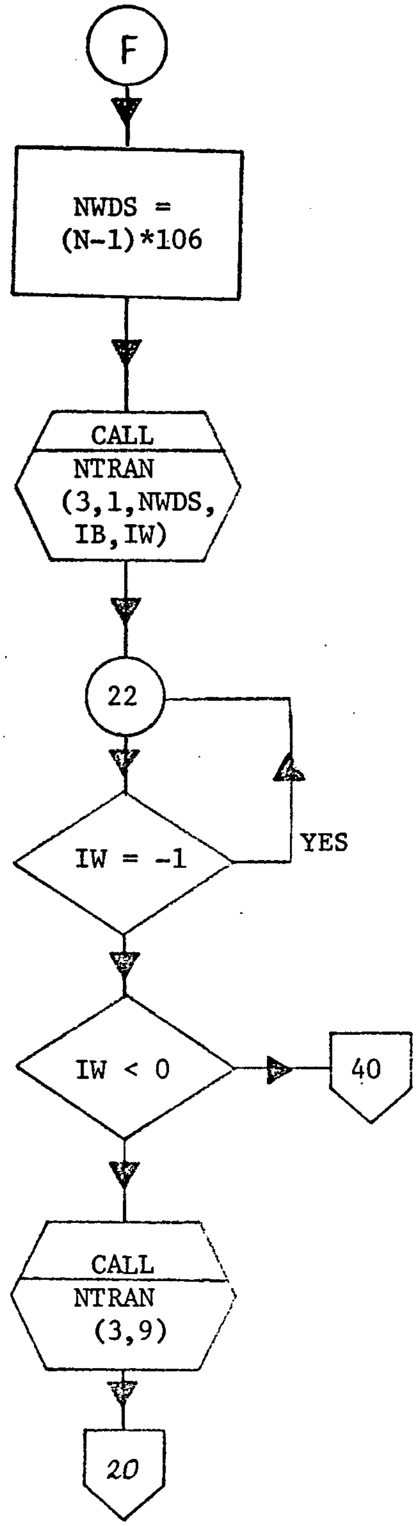


Fig. 12.3.1. DETAILED FLOW CHART - MCSF (Continued)

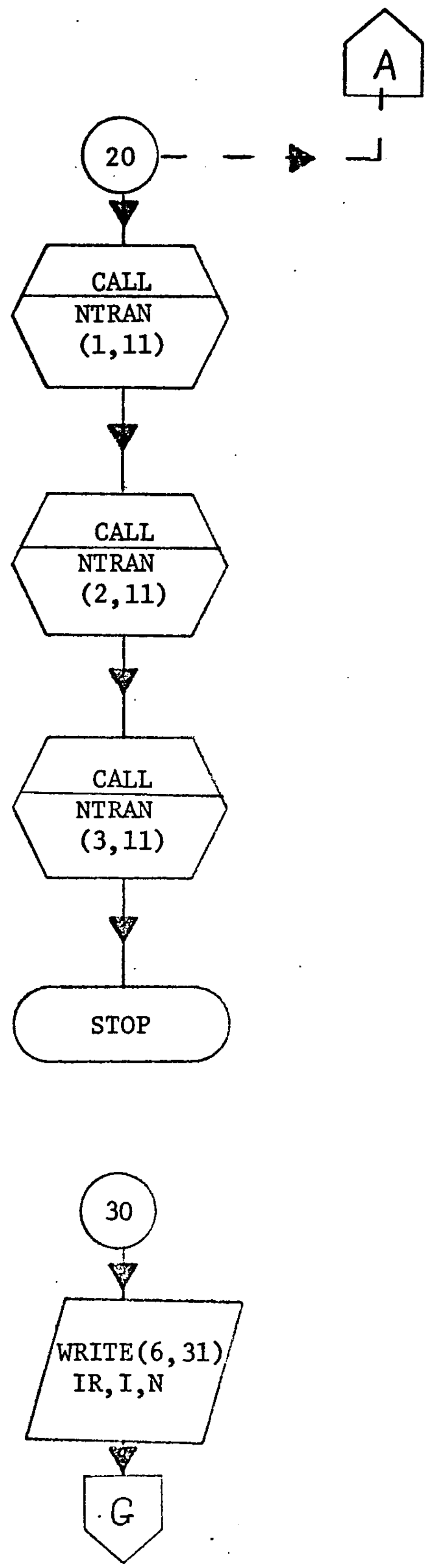


Fig. 12.3.1. DETAILED FLOW CHART - MCSF (Continued)

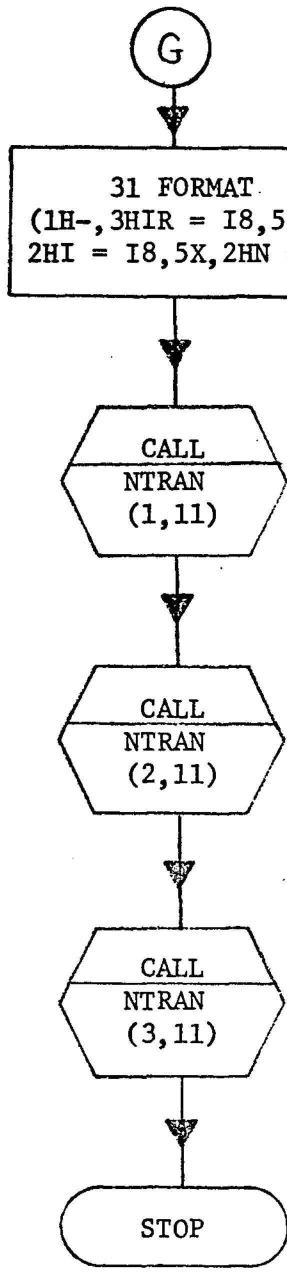


Fig. 12.3.1. DETAILED FLOW CHART - MCSF (Continued)

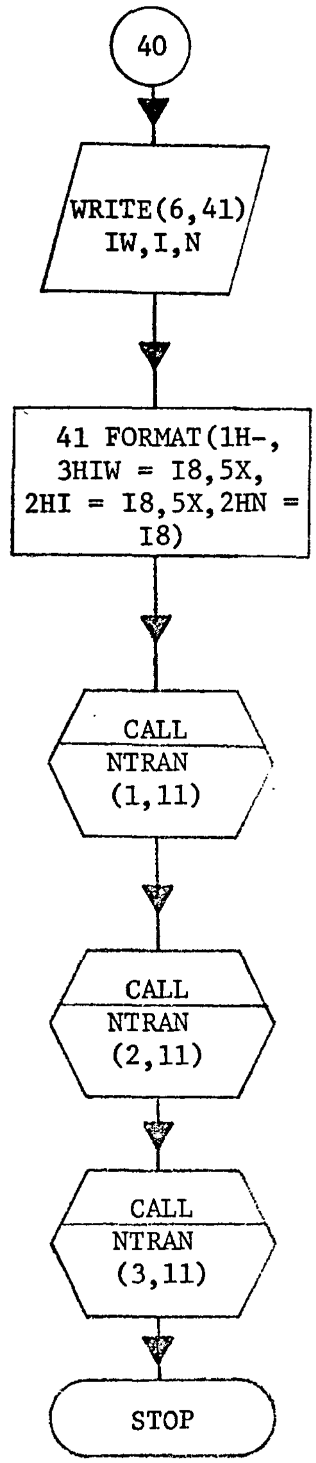


Fig. 12.3.1. DETAILED FLOW CHART - MCSF (Concluded)

Part III PHOTOMETRIC FUNCTION

An examination of the products of the Analysis Program suggested that the photometric function was biased for low brightness values. This conjecture received little support, however, from the 3D plot (using the pinplot machine at MSC). We initially tried changing values in the photometric function table, but this lead to unsatisfactory results. Changes which should have corrected a bias for low brightness values caused the contours to bunch up where the slope was about 2 - 3 degrees.

At the same time, work on the D log E curve was yielding better contour maps using the old photometric function. Accordingly, we concluded that the photometric function used by the Analysis Program is acceptable.

The 3D pinplot mentioned above is in the care of Dr. N. W. Naugle, 1410 Winding Road, College Station, Texas 77840.

Part IV THE ANALYSIS PROGRAM

1. Introduction

The study (Statistical Error Analysis of Photometric Data) which constitutes Project C of Contract NAS 9-10577 was intended to investigate the errors in the Analysis Program. Initial work of the study, begun in March 1970, involved the current Analysis Program in use then. We selected the version appropriate to LO III since that program was being used in Projects A and B. Considerable time was spent preparing working documentation of the critical subprograms, for the authors of these programs never furnished more than scant general flow charts and vague descriptions of the usage. This is particularly true of subroutine RECPO; this program was written in SLEWTH II assembly language, and performs the integration along radial lines thru the zero phase point.

This preliminary work had reached a certain state of completion when we learned of major changes being completed in the Analysis Program. We include here a table of subprograms of the Analysis Program as the program was in March, 1970 and as it was in December, 1970. One can see at a glance that a complete revamp of the Analysis Program had been undertaken, the main direction of which seemed to be the elimination of SLEWTH II assembly language programs of over twenty instructions.

Old Subprogram	New Subprogram	Type	Change
ALNTAB		ASM	omitted
	AUTOCO	FOR	new
BLOCKD	BLOCKD	FOR	
CONGRD	CONGRD	FOR	
CONTUR	CONTUR	FOR	
COUNT		FOR	omitted
DANCE	DANCE	FOR	
DMPBUF		FOR	omitted
DSPLAY	DSPLAY	FOR	
DUMMY	DUMMY	FOR	changed
FDATA		FOR	omitted
FILTER	FILTER	FOR	
FIX10		ASM	omitted
FIX30		ASM	omitted
FLT10		ASM	omitted
FLT30		ASM	omitted
FPOINT		FOR	omitted
	FPH2	FOR	new
	FPH3	FOR	new
	FPH4	FOR	new
	FPH5	FOR	new
	FPM2	FOR	new
	FPM3	FOR	new
	FPM4	FOR	new
	FPM5	FOR	new

Table 1.1 Subprograms of Analysis Program

Old Subprogram	New Subprogram	Type	Change
	PRO	FOR	new
	PSDPAK	FOR	new
QUIK3L		FOR	omitted
	RADREC	FOR	new
RDMKS	RDMKS	FOR	changed
READ		FOR	omitted
READPO		FOR	omitted
	REC1	FOR	new
	RECPAC	FOR	new
RECPO		ASM	rewritten in
	RECPO	FOR	FORTRAN
	RECRUN	FOR	new
RMS	RMS	FOR	
	RPREC	FOR	new
	RPSD	FOR	new
SIMML	SIMML	FOR	
STATGR	STATGR	FOR	
SUB1	SUB1	FOR	
SUMF	SUMF	ASM	
TABR		FOR	omitted
TANSB	TANSB	FOR	changed
	TRANS	FOR	new
	TRICON	FOR	new
	TRISEL	FOR	new
UMP	UMP	ASM	
FRAMEV		FOR	to FRAME

Table 1.1 Subprograms of Analysis Program (continued)

Old Subprogram	New Subprogram	Type	Change
	FRAME	FOR	
GDATA	GDATA	FOR	
GEER		FOR	omitted
GEER1		FOR	omitted
GET	GET	ASM	
HANPSD	HANPSD	FOR	updated
HAZMAP		FOR	omitted
HMOUTP		FOR	omitted
INPUT	INPUT	FOR	
INTSUB	INTSUB	FOR	
	MAXI	FOR	new
NCHI	NCHI	ASM	
PACK	PACK	ASM	
PACK6	PACK6	ASM	
PHISB	PHISB	FOR	corrected
	PLOT3	FOR	new
PLOTV		FOR	omitted
	PLOTQ	FOR	new
PMAP	PMAP	ASM	
POREC		ASM	rewritten in
	POREC	FOR	FORTRAN
PREG	PREG	ASM	
PRETTY	PRETTY	ASM	
PRINTM	PRINTM	FOR	
UNPACK	UNPACK	ASM	
WRITIH		FOR	omitted
WRRL		FOR	omitted
	XPOHI	FOR	new
	XPOME	FOR	new
XPOSUR		FOR	omitted

Table 1.1 Subprograms of Analysis Program (concluded)

Subroutines RECPO and POREC, as written in SLEWTH II, incorporated some of the tightest programming we have ever seen. The program maximized speed by, for example, carrying computations only so far as they were significant, and saving partial results which even FORTRAN V would miss. However, the fact that these subroutines were essentially undocumented, and were written in an unfamiliar language, made the Analysis Program hard to use. (If one does not understand the program in use, then one has difficulty tracing an apparent error and separating data errors, usage errors and programming errors.) Thus, it was possibly a good idea to rewrite the Analysis Program in FORTRAN V, since no expert programmers were available who could understand the SLEWTH II subroutines. This change caused the Analysis Program to require about twice the run time, but this may not have been considered excessive.

However, the programming change also cleaned up the Analysis Program and made the error analysis part of Project C rather simple; it no longer seemed like our kind of a job. Indeed, much of the work (working documentation of RECPO and POREC, preparation of flow charts) had already been done when we learned of the change in the Analysis Program. By that time, the difficulties of Project A, coupled with its greater intrinsic interest, had taken over the effort of the workers. By the time these difficulties were overcome, the Analysis Program was not being used for anything. It therefore seemed foolish to continue work on error analysis, and, instead, we report on what findings we made using the Analysis Program in Projects A and B.

We discovered several changes relevant to Project B and one relevant to Project A. The whole sequence of D log E - CFAC - RDMKS corrections is done inefficiently and inaccurately in the Analysis Program. For no time spent, while a chit is being read, one could perform all these corrections PLUS apply a 3 x 3 filter and take every other element - every other row (EOEEOR). The time saving results from the fact that all corrections can be performed at once, in one pass, at tape read speed. This work is reported on in 2 below.

In Project A, the final result was the join of an approximate 9 chit area. The geometry of the situation made it possible to prepare a "simulated" chit (about 2.71 x 2.71 times the size of a real chit) by selecting every 2.71 element. (The details are reported on in 3.) We attempted to make the Analysis Program do a contour map of this area and found that changes in the state vectors XX, YY and ZZ, and in DX, DY (multiplying each by 2.71) made a beautiful contour map of this nearly 9 chit area. There seems to be no limit to this technique; the assumptions made in the analysis procedure are clearly valid for at least a 10 x 10 chit size area.

Incidently to this part of our report we describe the method of making the tape which made Fig. 12.3, p. 212 of the report on Project A.

In Project B we found that filtered data made nicer contour maps than unfiltered data. This is probably due to ailaising in the sampling process, since the EOEEOR is only good up to about 60 1/mm and we found significant noise in the digitizer well above this frequency. The details are reported on in 4.

2. Improved Corrections. In this section, we summarize a special application of work done for Projects A and B. In order to understand the reasoning leading to the improved corrections in the Analysis Program, we need to review the method and order of the corrections in the Analysis Program.

The Analysis Program initially selects EOEEOR in subroutine GDATA, called before any other corrections are made. The EOE selection is performed by SLEWTH II subroutine UMP; EOR in by a DO logs and NTRAN. (The arithmetic in DO loop 4 is excessively complicated, but this is a minor point.) The Analysis Program now calls subroutine RDMKS. This subroutine creates a smear on each side of the pciture where the drummarks used to be, replacing the drummark region by linear interpolation of the drummark area boundary points. (Using the algorithm developed for Project A would speed and substantially improve this subroutine, but we shall see an even better improvement is possible.) In case there is a mirror in the camera setup (IMED = 0) (usually!), the rows have to be turned over. This, also, represents wasteful computation. Next, subroutine XPOHI (or XPOHM) is called, resulting in more calculations and needless UNPACK - PACK6 combinations. In addition, XPOHI does a bewildering amount of statistice, which seems to be seldom wanted.

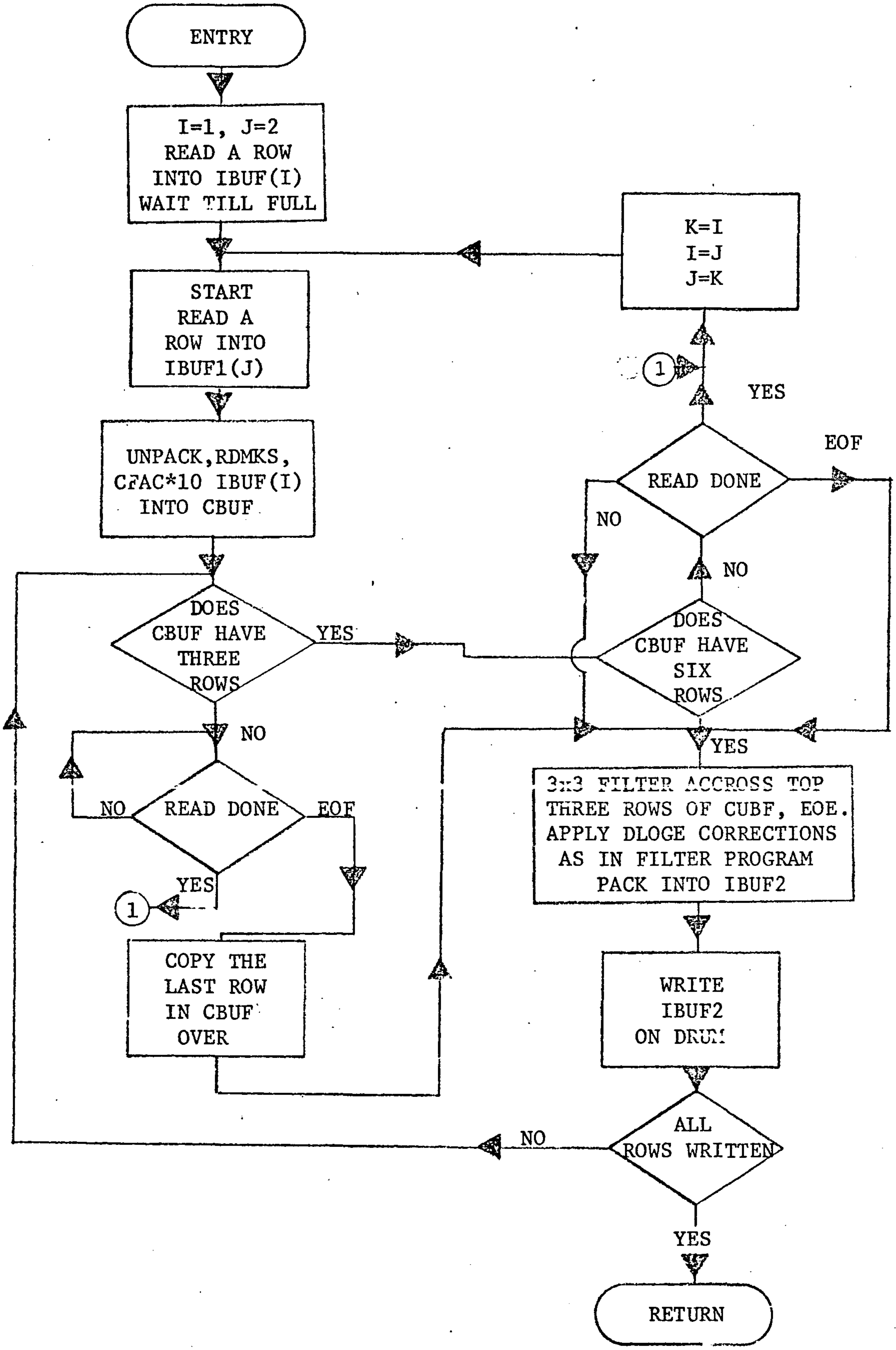
The point of this summary of how the Analysis Program gets started is that, for each chit, nearly 100,000 needless UNPACK - PACK6 calls are made in a total of 14 DO loops and thru subroutines. This only takes ten or fifteen seconds, probably not critical, but the point is that no time at all should be taken, since all the corrections can be done,along with the application of a fast 3 x 3 filter, while the data is being read from tape.

The general flow chart of an algorithm which performs this is given in Fig. 2.1. The large box ('3 x 3 FILTER ...') should be self-explanitory to one familiar with Projects A and B. We summarize:

RDMKS -

Replaces IBUF(1) by IBUF(2).

Replaces IBUF(8) and IBUF(9) by



NOTE: IBUF1 IS A DOUBLE INPUT BUFFER. CBUF IS A 636 x 6 CIRCULAR BUFFER. IBUF2 IS A 53 x 1 OUTPUT BUFFER

Fig. 2.1 General flow chart, corrections and filtering.

MAXO (IBUF(8), IBUF(7)) and MAXO (IBUF(9), IBUF(7)).

Call this operation REP(8, 9, 7). The algorithm then performs REP(10, 11, 12)(completing the removal of the left drummarks), and then REP(625, 626, 624), REP(627, 628, 629)(completing the removal of the right drummarks).

CFAC*10 -

This is the modified correction factor routine developed in the filter program. The reasons for the change is not apparent until the D log E corrections are applied. The net effect is to improve the tonal resolution at the end of DLOGE curve where the most rapid changes occur.

3 x 3 FILTER -

This filter is a symmetric 3 x 3 filter with an upper frequency cutoff of about 60 1/mm and with X2 enhancement at about 20 - 30 1/mm. The filter, being symmetric, should be applied using the more efficient code developed for Project B which makes use of the symmetry. The reason the filter is so rapidly applied is not only owing to its symmetry; also, only EOEEOR need be filtered.

3. NINE CHIT ANALYSIS

The JOIN program documented in Project A, took three framelets 2068 rows long and made one joined area about three chits wide and long. (The exact width was 1878 elements. This is less than 3 x 636 = 1908 because of overlap and the fact that the width must be a multiple of six.) Because the chits were staggared, 2012 rows only were formed in the join of thru 2068 row subframelets. The program SELECT was written to select every 2.71 x 2.71 element from the join. The general flow chart of this program is given in Fig. 3.1. Prior to performing SELECT, the data (JOIN of 2068) was filtered by the 7 x 7 filter given in

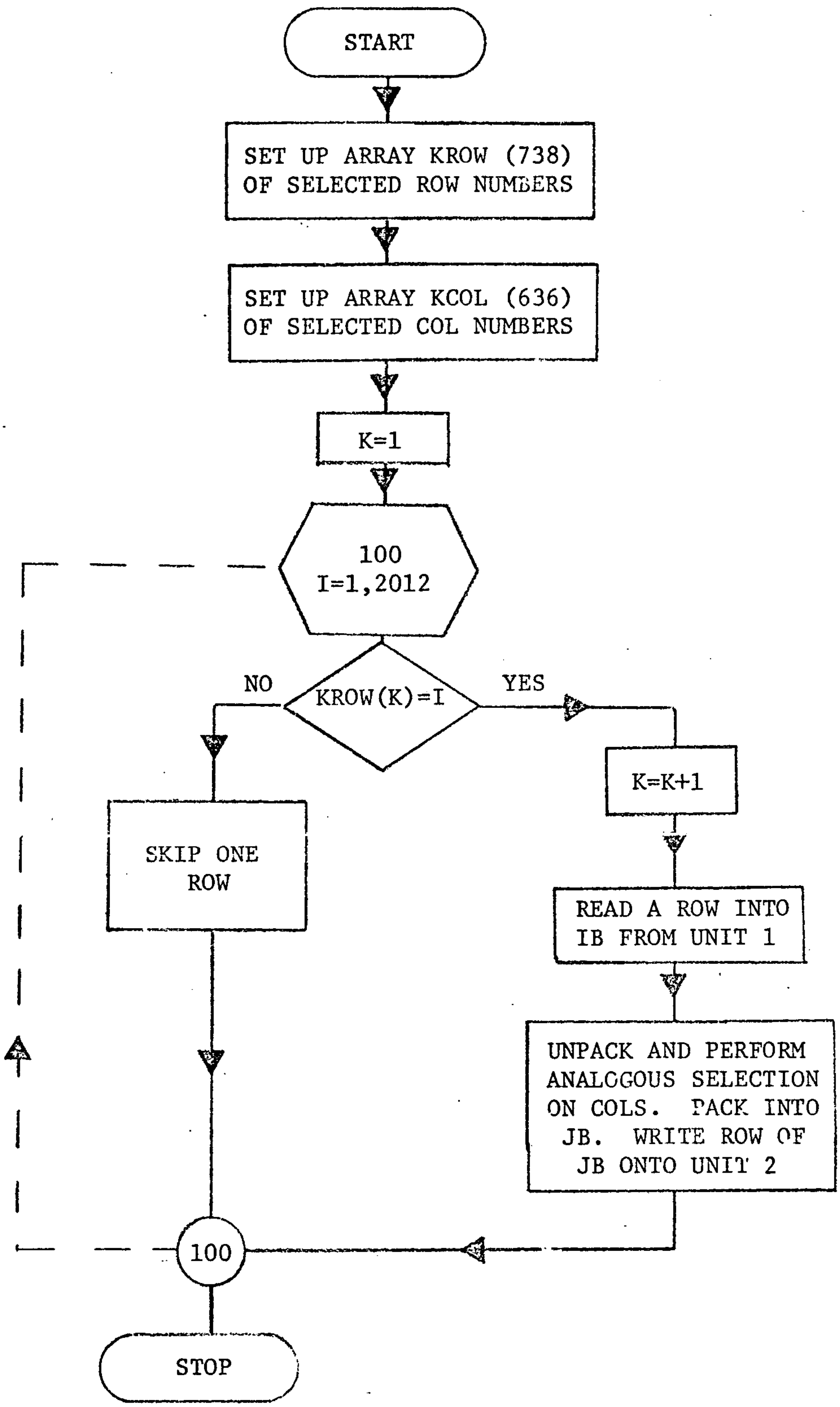


Fig. 3.1 General Flow Chart, SELECT

	COL 1	COL 2	COL 3
ROW 1	.00160905	.00482329	.01029977
ROW 2	.00375408	.01317102	.02676457
ROW 3	.00688308	.02329252	.04565517
ROW 4	.00842219	.02798151	.05422945
ROW 5	.00688303	.02329251	.04565516
ROW 6	.00375408	.01317102	.02676456
ROW 7	.00160904	.00482329	.01029978

COL 4	COL 5	COL 6	COL 7
.01317883	.01029977	.00482329	.00160905
.03355888	.02676457	.01317102	.00375408
.05661355	.04565517	.02329252	.00688308
.06702120	.05422945	.02798151	.00842219
.05661354	.04565516	.02329251	.00688308
.03355887	.02676455	.01317102	.00375408
.01317884	.01029978	.00482329	.00160904

Fig. 3.2 Filter, applied before SELECT

	COL 1	COL 2	COL 3
ROW 1	.00235325	.00068320	.00142543
ROW 2	.00100271	.00314601	.01906304
ROW 3	.00063853	.01400500	.05979526
ROW 4	.00091490	.02135989	.08430406
ROW 5	.00063853	.01400501	.05979523
ROW 6	.00100271	.00314601	.01906305
ROW 7	.00235325	.00068320	.00142543

COL 4	COL 5	COL 6	COL 7
.00299262	.00142543	.00068320	.00235325
.03132849	.01906304	.00314601	.00100271
.09145999	.05979526	.01400501	.00063853
.12683061	.08430406	.02135989	.00091490
.09145994	.05979522	.01400501	.00063853
.03132849	.01906305	.00314601	.00100271
.00299262	.00142543	.00068320	.00235325

Fig. 3.3 Filter, applied after SELECT

(Not Reproducible)

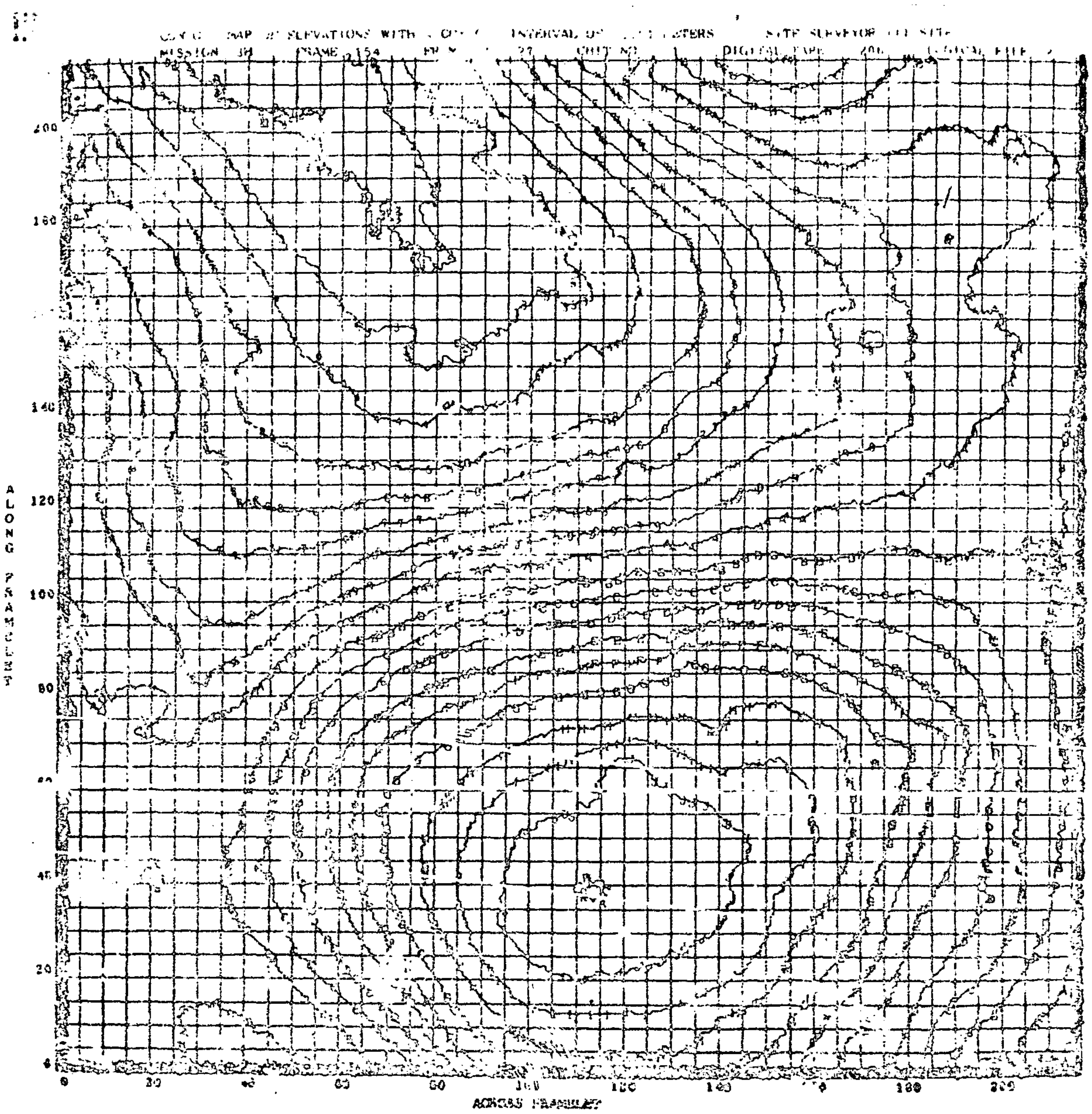
Fig. 3.4 Contour map, r.i.ie chit area.

Fig. 3.2. After the selection, the filter algorithm for chit size data was applied with the filter in Fig. 3.3. To summarize, we performed the following operations: JOIN, FILTER with Fig. 3.2, SELECT every 2.71 x 2.71, FILTER with Fig. 3.3, and sent the resulting picture (now 636 x 738) to JPL-IPL for reproduction.

The same tape was merged with a header and edge data so that it could be analyzed by the Analysis Program as we have done with other filtered data. By changing the state vectors (by multiplying by 2.71), and by multiplying DX and DY by 2.71, we were able to produce the contour map in Fig. 3.4. The MSC number of this tape is 49634.

The picture of the JOIN area in Project A (Fig. 12.3, p. 212) was made by a similar (although much simpler) process. We first filtered to cutoff above 60 1/mm; then EOEEOR was selected (making a picture 929 x 1006, which became 924 x 1002 on packing) and this tape was sent to JPL-IPL for display.

4. Filtered Input for the Analysis Program. Using the filter given in Fig. 3.3, we filtered the SURVEYOR III SITE chit, and prepared the contour map in Fig. 4.1. (The analysis program was modified to remove corrections we had already performed.) This filter has parameters a = 1.35, b = 40. It is interesting that the filtered chit contour map shows the SURVEYOR III crater to be about 16.5 meters deep, vs. 12.5 meters for our best other map. The contours are obviously smoother and easier to interpret. Since filtering could have been incorporated with no additional computer time spent, it is unfortunate that this was not done.



CONTOUR IDENT.	
1	-17.04
2	-15.22
3	-13.41
4	-11.59
5	-9.78
6	-7.97
7	-6.15
8	-4.34
9	-2.52
A	-.71
B	1.10
C	2.91
D	4.73
E	6.54
F	8.36
G	10.17
H	11.98
I	13.80
J	15.61
K	17.43
L	19.24
M	21.06
N	22.87
O	24.69
P	26.50
Q	28.31
R	30.13
S	31.94
T	33.76
U	35.57

Fig. 4.1 Contour map -- new correction factors, D log E , and filtered