NASA TECHNICAL MEMORANDUM

NASA TM-78133

A FAST ROUTINE FOR COMPUTING MULTIDIMENSIONAL HISTOGRAMS

By Robert R. Jayroe, Jr.
Data Systems Laboratory

October 1977

NASA

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

TABLE OF CONTENTS

LIST OF TABLES

TECHNICAL MEMORANDUM 78133

A FAST ROUTINE FOR COMPUTING MULTIDIMENSIONAL HISTOGRAMS

I. INTRODUCTION

Usually, a table look up procedure can be used to calculate histograms for data containing up to two variables or having two dimensions, provided that the range of each variable is known and a satisfactory class interval can be selected that does not exceed the computer memory allocation. For example, suppose that data representing two variables (two-dimensional data) have been digitized to six bits so that each variable has an integer range of from 0 to 63. The resulting joint histogram could have as many as $64^2 = 4096$ different elements, but in most cases not all of them will occur. All of the different elements would occur, however, if the two variables were independent and completely random. To account for that possibility, a software routine would have to specify that 4096 memory locations be set aside for counting the number of occurrences (population) of each different element.

If N(I, J) is the population counter, and I and J are the data values (0 to 63) of the two variables, then N(I, J) is increased by one every time the paired values of I and J are encountered in the data. Thus, the paired data values are the indices (table) of the population counter and there is a one to one correspondence, hence the name table look up. Usually a table look up procedure wastes memory locations, since many of the different paired data values I and J never occur, but the procedure is extremely fast.

The search procedure does not usually require as many memory locations, but it is a much slower approach. Thus, the tradeoff between the two procedures is one of computer memory requirements versus computer time needed to perform the search.

For the search procedure, not only is a population counter needed, but another variable for identifying the different elements of paired values, I and J, that have occurred is also needed. For the table look up procedure, the indices

of the population counter were the identifiers. In the search case, let N(K) be the population counter of the Kth different element of paired values I and J. Then, let M(K, L) be the identifier of the Kth different element (two-dimensional vector) with L equal to one for the first component and L equal to two for the second component. Thus, the value of M(K,1) and M(K,2) would be the respective I and J paired values of the Kth different element. For every pair of I and J values, the list of M(K,1) and M(K,2) has to be searched to determine if the paired I and J values, respectively, are equal to any of the K identifiers, M(K,1) and M(K,2). If I and J were the same as the Lth identifier, the search would be terminated and the population counter N(L) would be incremented by one. However, if the paired values of I and J were different from all of the K identifiers, this new element of paired values would be added to the list by setting M(K+1,1)=I, M(K+1,2)=J, and setting the population counter N(K+1)=1. Thus, the list grows as new paired values are found and the search usually takes longer and longer.

With multispectral image data, the memory location requirements are usually too large to use a pure table look up procedure, and too much time is required to use a pure search procedure. Therefore, the problem is to find an efficient combination of a table look up and search procedure for calculating multidimensional histograms.

II. LANDSAT HISTOGRAM EXAMPLE USING A PURE SEARCH PROCEDURE

In one Landsat image there are 7 581 600 picture elements (Pels), and each Pel, which corresponds to a particular location of approximately 80 m resolution in the ground scene, is represented by a four-dimensional vector. The four components of this vector correspond to the reflected light intensity in each of the four different spectral images acquired by the sensor at a particular ground scene location. All four spectral images are digitized to six bits, but three of the images are radiometrically corrected to seven bits. Hence, in three of the images the data can range from 0 to 127, while in the fourth the data can range from 0 to 63. The maximum number of different four-dimensional vectors that could be generated with this combination of integers is $128^3 \times 64 =$ 131 096 512. However, since there are only 7 581 600 Pels in an image, and not all of the vectors will be different, it is estimated that there could be as many as a million, or at least several hundred thousand, different four-dimensional vectors in one Landsat image. Thus, the memory requirements for a four-dimensional histogram are huge regardless of what kind of procedure is used.

A pure search procedure was written for an IBM-360/75 that would terminate after the desired amount of data was examined, or 10 000 different four-dimensional vectors were found, or if the computer time exceeded a specified amount. In the terminology of the Introduction, there were 10 000 N(K)'s and 10 000 M(K, L)'s with L going from one to four. Table 1 shows the results of the search procedure as a function of the number of Pels, the number of different vectors and computer time.

TABLE 1. SEARCH PROCEDURE COMPUTER TIMES

Number of Pels	Number of Vectors	Computer Time (s)
22 932	1975	111
22 932	1995	128
22 932	2234	134
22 932	2254	135
22 932	3172	174
22 932	3850	270
22 932	6213	517
22 932	6576	529
22 932	8039	721
12 000	2395	108
60 000	5073	686
120 000	6738	1565
132 000	6912	1735

Two different cases are considered in Table 1. One is where there are a moderate number of different vectors per number of Pels, and the other is where there are many more Pels than vectors. The first case is a test site containing 22 932 Pels, but the image data were acquired over the test site at nine different times and therefore contain a different number of vectors in each acquisition. The second case is a different test site, and the search program was started at the same Pel location each time, but the total number of Pels examined was increased with each run. In the first case the computer time is more dependent on the number of vectors, while in the second case it appears to be more dependent on the number of Pels. Table 1 illustrates that for this type search procedure, the computer time will be highly independent on the image test site data and difficult to estimate.

The main reason the pure search procedure is so time consuming is that each new and different vector will be compared with all of the previous vectors that have been found before it is inserted in the table, and the vectors are not ordered in any particular sequence, except for the order in which they occurred in the image data. Thus, the problem is to invent some ordering procedure so that the number of searches can be drastically reduced.

III. COMBINATION TABLE LOOK UP AND SEARCH PROCEDURE

As a means of ordering the different vectors, it would be convenient if each different four-dimensional vector could be represented by a unique single number for a table look up procedure, but yet not exhaust and waste memory locations. It would be even more convenient if these single numbers could be consecutive integers starting with one and ending with a maximum number that can be determined from the amount of memory locations that are allowable. In the opening paragraph of Section II, it was shown that it is not practical to assign a unique number to each different vector since it is not possible to count all the vectors with most computer systems. However, it is possible to count a considerable number of different vectors at a reasonable rate if each different vector is assigned a single integer number that is approximately unique.

To describe this approximation, the use of the terms divisor, base, and multiplier need to be discussed. Let I_1, I_2, I_3, and I_4 be the components of a four-dimensional vector or the value of the image data at a particular Pel location for the four spectral images, and assume that the I values are 83, 37,

64, and 52, respectively. Each component value is divided by a divisor and the remainder is computed. For example, if the divisor is 10, then the remainders R_1, R_2, R_3, and R_4 would be 3, 7, 4, and 2, respectively. The base is used to convert the four integer remainders to one number. If the base is 10, then R_1 is multiplied by 10^0, R_2 by 10^1, R_3 by 10^2 and R_4 by 10^3, and the resulting numbers are added together. For the particular example given, the vector 83, 37, 64, 52 would be converted to the single number 2473. Using a divisor and base equal to 10, the largest single number that could be obtained for a four-dimensional vector is 9999. These single integer numbers are not unique since there are instances where different vectors will be converted to the same number, and to help distinguish them, a multiplier is needed. Suppose for the case where the divisor and base are equal to 10, that it is desirable to count up to 40 000 different vectors. The multiplier is computed by dividing 40 000 by 9999 and rounding off to the lowest integer value, which in this case is four. Thus, the vector 83, 37, 64, 52 is converted to the number 2473 by the divisor and base, and that number is converted to 9892 by the multiplier, four. The multiplier by itself does not solve the duplication problem, but it does provide the capability for drastically reducing the number of searches that are required when duplicate indices or table values are encountered.

Thus far, the table look up part of the histogram procedure has been described. The search procedure starts after the vector has been converted to a single table number, K, and is based first on the population counter, N(K), and then on the actual vector components, M(K, L), where L = 1, 2, 3, 4. If the population counter N(K) is zero for a particular K value, then the population counter is set equal to one and the components of the vector are put into M(K, L). If the population counter N(K) is not equal to zero, then the vector components are compared with the components that are in M(K, L). If the vector components are identical to the M(K, L) components, then the population counter is incremented by one and the search is terminated. Otherwise, the table value K is incremented by one and the search procedure is repeated with the population counter N(K+1). The search procedure is terminated when an empty space is found in the table, an identical vector is found, or when the table is completely filled. The index K can be incremented by one until the end of the table is reached, which in the previous example was a value of 40 000. If the value 40 000 is reached, K is set to one and incremented from the beginning of the table. This procedure is used until 40 000 different vectors are found, and the program terminates when vector 40 001 is encountered.

The multiplier performs the function of providing additional consecutive locations in the table for different vectors having the same index K. If the multiplier is four, then K is an integral multiplier of four, which means that there are three K values that cannot be computed from the vector data between every two consecutive K values that can be computed from the vector data. For example, consider the two vectors 83, 37, 64, 52 and 84, 37, 64, 52, which only differ by a value of one in the first component. The corresponding K values are 9892 and 9896, and the K values 9893, 9894, and 9895 cannot naturally occur in the vector data. The three K values that do not naturally occur could, for example, be the table values for the vectors 73, 67, 44, 32; 53, 57, 14, 32; 23, 17, 84, 52 all of which have the index 9892. Thus, the index K tells where to start searching in the table and usually the search does not have to continue very long. If the multiplier is N, there will be N consecutive table locations available for different vectors having the same index K, and the naturally occurring indices are equally spaced. The preceding statement is true provided the divisor and base are the same number. Occasionally, the situation arises where there are many different vectors in the data that all have the same index. When this happens, some of the vectors may be placed in the table at considerable distances from their index value and the search time for these vectors will increase. This situation can be remedied somewhat by making the base larger than the divisor. This has the effect of inserting additional spaces of indices that do not naturally occur in the vector data at table values that are integral multiples of the base times the multiplier. The number of these spaces that are inserted at each integral multiple is equal to the difference between the base and divisior times the multiplier. Making the base smaller than the divisor is not profitable, since it usually increases the probability of occurrence of index duplication in the vector data and makes the search time longer.

The choice of a divisor, base, and multiplier will mainly depend on the amount of computer storage that is available, but in some cases may depend on the distribution of the data as previously discussed. A choice can be made by using the relationship between the maximum number of different vectors K, and the divisor D, base B, and multiplier M which is given by

$$K \geq M \cdot (D - 1) \cdot (1 + B + B^2 + B^3) \quad \text{with} \quad B \geq D \quad \text{and} \quad M \geq 1 \quad .$$

This particular combination table look up and search procedure generally proceeds at a fast pace until the table of vectors starts to fill up. For Landsat imagery it appears reasonable to assume that the table size should be approximately 20 percent of the total number of Pels examined for small test areas, and this constraint can be considerably relaxed for test areas containing several hundred thousand or more Pels. For image data during the growing season, the vector table needs to be made several times larger.

Table 2 shows the results of the combination table look up and search procedure for various Landsat test areas. In this case the computer time appears to be mainly dependent on the number of Pels and the distribution of the vector data, provided the table size is large enough, and mostly independent of the number of different vectors.

TABLE 2. COMBINATION TABLE LOOK UP AND SEARCH PROCEDURE COMPUTER TIMES

Number of Pels	Number of Vectors	Computer Time (s)	Table Size (Vectors)
21 504	450	7	30 000
21 504	3 552	7	30 000
21 952	472	6	40 000
21 952	3 598	7	40 000
51 000	1 118	14	40 000
51 000	11 179	15	30 000
310 800	11 180	80	30 000
310 800	19 602	76	30 000
310 800	27 847	96	40 000
312 000	33 670	56	60 000
708 000	54 905	457	60 000
1 440 000	25 001	347	40 000
1 440 000	27 696	505	40 000
1 440 000	31 751	424	40 000

IV. APPLICATIONS

The main application of the multidimensional histogram computation is multiimage or multivariable signal processing, provided the number of different vectors does not reach an unmanageable size. As more images are added to the data set or as the dimensionality increases, the number of different vectors will increase drastically and some averaging process, such as a clustering approach, may be needed to reduce the number of different vectors to a manageable size. In cases where the number of different vectors is much smaller than the number of Pels, considerable savings in processing costs can be achieved by extracting the multidimensional histogram information and reformatting the image data. The reformatting takes the form of placing the vectors at the beginning of the data, together with the number of times each vector occurs, followed by the image with one number at each Pel location that identifies the vector that belongs there. This type of format permits each vector to be processed once, instead of processing every Pel, and the results of the processing can then be applied to each Pel with more cost effectiveness by using a table look up procedure. This type of format is efficient for all types of processing, except those that require spatial averaging such as filtering and interpolation. The types of processing that require spatial information could be performed by reconstructing the image data to its original form, but more different vectors would be generated as a result of the spatial processing. The main types of processing that would benefit from such a format include classification inventories, image ratioing and differencing, density stretching, lower order distribution computations, and other types of image transformations. The number of different vectors contained in an image could also be used as a measure of image complexity, and as a criterion for evaluating various image processing techniques.

APPENDIX A

FORTRAN PROGRAM LISTING

```
LEVEL 21.8 ( JUN 74 )                    OS/360  FORTRAN H

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=56,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002            LOGICAL*1 N(40000,4),M(4),A(784)
ISN 0003            DIMENSION NPOP(40000),NN(120)
                C
                C     THE PROGRAM COMPUTES A 4 DIMENSIONAL HISTOGRAM.
                C     THE VECTORS ARE READ IN ONE RECORD AT A
                C     TIME USING THE VARIABLE A.
                C     N(I,J)=THE J COMPONENTS OF THE I DIFFERENT VECTORS IN THE TABLE
                C     NPOP(I)=THE NUMBER OF OCCURRENCES OF THE ITH VECTOR IN THE TABLE
                C     NN(I)=THE NUMBER OF DIFFERENT VECTORS THAT OCCUR I TIMES
                C     M=THE COMPONENTS OF A VECTOR
                C     NBAND=NUMBER OF SPECTRAL IMAGES OR VECTOR DIMENSION
                C     NPIX=NUMBER OF PELS PER RECORD
                C     NBITE=NUMBER OF 8 BIT BYTES PER RECORD, THE ARGUMENT OF A
                C     NREC=NUMBER OF RECORDS TO EXAMINE
                C     KOUNT=A VARIABLE USED TO COUNT THE NUMBER OF DIFFERENT VECTORS
                C     KNT=A VARIABLE USED TO COUNT THE NUMBER OF PICTURE ELEMENTS(PELS)
                C     IMOD=DIVISOR
                C     JMOD=BASE
                C     JFAC=MULTIPLIER
                C     NSKIP=NUMBER OF RECORDS TO BE SKIP
                C
                C     STRTTM=SUBROUTINE TO START CLOCK FOR PRINTING OUT COMPUTER TIME
ISN 0004            CALL STRTTM
                C
                C     WRITE TITLE FOR OUTPUT PAGE
ISN 0005            WRITE(6,700)
ISN 0006          700 FORMAT(' HASH',9X,'LACIE FILE 1',/)
                C
ISN 0007            READ(5,701)NSKIP,NBAND,NPIX,NBITE,NREC,JFAC,IMOD,JMOD
ISN 0008          701 FORMAT(8I5)
ISN 0009            WRITE(6,702)NSKIP,NPIX,NREC,JFAC,IMOD,JMOD
ISN 0010          702 FORMAT(' RECORDS SKIPPED=',I5,' PIXELS USED=',I5,
                   1' RECORDS USED=',I5,'MULTIPLIER=',I5,' DIVISOR=',I5,' BASE=',I5,/)
ISN 0011            KNT=0
ISN 0012            KOUNT=0
                C
                C     MEAL=A VARIABLE USED TO PRINT OUT THE PEL NUMBER AT EACH 100TH
                C     AND 1000TH NEW VECTOR
ISN 0013            MEAL=100
                C
                C     SKIP NSKIP RECORDS
ISN 0014            IF(NSKIP.EQ.0)GO TO 81
ISN 0016            DO 80 I=1,NSKIP
ISN 0017            READ (1)DIA
ISN 0018          80 CONTINUE
                C
                C     IFEAT=MAXIMUM NUMBER OF DIFFERENT VECTORS ALLOWED
ISN 0019          81 IFEAT=40000
                C
                C     INITIALIZE ALL VECTOR POPULATIONS TO ZERO
ISN 0020            DO 83 I=1,IFEAT
ISN 0021            NPOP(I)=0
```

```
ISN 0022        83 CONTINUE
                C
                C     LOOP 300 READ IN NREC RECORDS
ISN 0023              DO 300 I=1,NREC
                C
                C     READ IN ONE RECORD
ISN 0024              READ (10)A
                C
                C     ACCUMULATE FOUR DIMENSIONAL HISTOGRAM FOR EACH
                C     RECORD.  J, JA, JB,JC ARE BYTE COUNTERS FOR EACH PIXEL.
ISN 0025        91 DO 200 J=1,NBITE,NBAND
                C
                C     COUNT EVERY PICTURE ELEMENT (PEL)
ISN 0026              KNT=KNT+1
                C
                C     GET EACH VECTOR OUT OF A AND PUT INTO M
ISN 0027        91 JA=J+1
ISN 0028              JB=JA+1
ISN 0029              JC=JB+1
ISN 0030              M(1)=A(J)
ISN 0031              M(2)=A(JA)
ISN 0032              M(3)=A(JB)
ISN 0033              M(4)=A(JC)
                C
                C     4 DIMENSIONAL HISTOGRAM ROUTINE (COMBINATION TABLE LOOK UP/SEARCH
                C     PROCEDURE)
                C     BEGIN TABLE LOOK UP PART
ISN 0034              L=0
ISN 0035              DO 99 NB=1,NBAND
ISN 0036              ID=M(NB)-IMOD*(M(NB)/IMOD)
ISN 0037              L=ID+L*JMOD
ISN 0038        99 CONTINUE
ISN 0039              L=L*JFAC+1

                      END TABLE LOOK UP PART, START SEARCH PART
ISN 0040        100 IF(NPOP(L).NE.0)GO TO 110
                C
                C     HAVE FOUND A NEW VECTOR, INCREMENT VECTOR COUNTER
ISN 0042              KOUNT=KOUNT+1
                C
                C     MN IS A FEATURE VECTOR COUNTER THAT INDENTIFIES
                C     EVERY 100TH AND 1000TH VECTOR
ISN 0043              MN=KOUNT-(KOUNT/MFAC)*MFAC
ISN 0044              IF (MN.NE.0) GO TO 101
ISN 0046              IF(KOUNT.GE.9900)MFAC=100
ISN 0048              X=KNT
ISN 0049              Y=KOUNT
ISN 0050              X=X/Y
                C
                C     PRINT PEL NUMBER FOR EVERY 100TH AND 1000TH VECTOR
ISN 0051              WRITE(6,800)I,KNT,KOUNT,X
ISN 0052        800 FORMAT(1X,'SCAN=',I5,2X,'PEL.NO.=',I9,2X,'VECT.NO.=',I5,10X,
                   &'P/V=',F15.6)
                C
```

```
                C     CHECK TO SEE IF LIMIT ON NUMBER OF VECTORS
                C     IS EXCEEDED
ISN 0053          101  IF (KOUNT.GT.IFEAT)GO TO 400
                C
                C     SET POPULATION OF NEW VECTOR TO ONE
ISN 0055            NPOP(L)=1
                C
                C     PUT NEW VECTOR INTO TABLE
ISN 0056            DO 105 NB=1,NBAND
ISN 0057            N(L,NB)=M(NB)
ISN 0058          105 CONTINUE
ISN 0059            GO TO 200
                C
                C     CHECK TO SEE IF VECTOR IS IN TABLE
ISN 0060          110 DO 120 NB=1,NBAND
ISN 0061            IF(N(L,NB).NE.M(NB))GO TO 130
ISN 0063          120 CONTINUE
                C
                C     VECTOR IS IN TABLE, INCREMENT POPULATION COUNTER
ISN 0064            NPOP(L)=NPOP(L)+1
ISN 0065            GO TO 200
                C
                C     VECTOR IS NOT ITHE SAME AS THE ONE WITH INDEX L,
                C     TRY THE NEXT INDEX
ISN 0066          130 L=L+1
                C
                C     CHECK TO SEE IF INDEX IS LARGER THAN END OF TABLE
                C     IF SO, SET INDEX TO ONE AND START AT BEGINNING OF TABLE
ISN 0067            IF(L.GT.IFEAT)L=1
ISN 0069            GO TO 110
                C
                C     RETURN TO NEXT VECTOR
ISN 0070          200 CONTINUE
                C
                C     RETURN TO NEXT RECORD
ISN 0071          300  CONTINUE
                C
                C     HAVE COMPLETED HISTOGRAM
                C     PRINT LAST NEW VECTOR AND PEL NUMBER THAT OCCURRED
ISN 0072          400 X=KNT
ISN 0073            Y=KOUNT
ISN 0074            X=X/Y
ISN 0075            I=I-1
ISN 0076            WRITE(6,800)I,KNT,KOUNT,X
                C
                C     WRITE OUT FEATURE VECTORS THAT OCCUR
                C     AT LEAST 1000 TIMES
ISN 0077            DO 450 I=1,IFEAT
ISN 0078            IF (NPOP(I).LT.1000)GO TO 450
ISN 0080            WRITE (6,810) (N(I,NC),NC=1,4),NPOP(I)
ISN 0081          810 FORMAT(1X,4(I4),'POP.=',I7)
ISN 0082          450 CONTINUE
                C
                C     POPULATION DISTRIBUTION IN LOGARITHMIC INCREMENTS
```

```
ISN 0083            DO 500 I=1,120
ISN 0084            NN(I)=0
ISN 0085          500 CONTINUE
ISN 0086            DO 550 I=1,IFEAT
ISN 0087            IF(NPOP(I).LE.0)GO TO 550
ISN 0089            II=NPOP(I)/1000
ISN 0090            IF(II.LT.1)GO TO 510
ISN 0092            II=II+100
                C
                C     THE ABOVE COUNTS THE NUMBER OF VECTORS THAT OCCUR 1000S OF TIMES
ISN 0093            GO TO 540
ISN 0094          510 II=NPOP(I)/100
ISN 0095            IF (II.LT.1)GO TO 530
ISN 0097            II=II+99
                C
                C     THE ABOVE COUNTS THE NUMBER OF VECTORS THAT OCCUR 100S OF TIMES
ISN 0098            GO TO 540
ISN 0099          530 II=NPOP(I)
                C
                C     THE ABOVE COUNTS THE NUMBER OF VECTORS THAT OCCUR FROM 1-99 TIMES
ISN 0100          540 NN(II)=NN(II)+1
ISN 0101          550 CONTINUE
ISN 0102            DO 560 I=1,99
ISN 0103            IF(NN(I).LT.1)GO TO 560
ISN 0105            X=NN(I)*100
ISN 0106            X=X/Y
                C
                C     PRINT THE NUMBER OF VECTORS THAT OCCUR 1-99 TIMES
ISN 0107            WRITE(6,820)I,NN(I),X
ISN 0108          820 FORMAT(1X,'NO.OCCURRENCES=',I5,1X,'HOW MANY=',I9,' PERCENT=',F8.4)
ISN 0109          560 CONTINUE
ISN 0110            J=0
ISN 0111            DO 570 I=100,139
ISN 0112            IF (NN(I).LT.1)GO TO 570
ISN 0114            J=J+100
ISN 0115            X=NN(I)*100
ISN 0116            X=X/Y
                C
                C     PRINT THE NUMBER OF VECTORS THAT OCCUR 100S OF TIMES
ISN 0117            WRITE(6,820)J,NN(I),X
ISN 0118          570 CONTINUE
ISN 0119            J=0
ISN 0120            DO 580 I=140,170
ISN 0121            IF(NN(I).LT.1)GO TO 590
ISN 0123            J=J+1000
ISN 0124            X=NN(I)*100
ISN 0125            X=X/Y
                C
                C     PRINT THE NUMBER OF VECTORS THAT OCCUR 1000S OF TIMES
ISN 0126            WRITE(6,820)J,NN(I),X
ISN 0127          580 CONTINUE
                C
                C     CALL TIME ROUTINE AND PRINT ELAPSED TIME
ISN 0128            CALL GETETM
ISN 0129            CALL PRTETM
ISN 0130            REWIND 10
ISN 0131            STOP
ISN 0132            END
```

APPENDIX B

SAMPLE OUTPUT RESULTS

```
HASH     LACIE FILE 1

RECORDS SKIPPED=     0 PIXELS USED=  196 RECORDS USED=  112MULTIPLIER=     4 DIVISOR=     9 BASE=    10

SCAN=     2  PEL.NO.=       213  VECT.NO.=  100          P/V=      2.12999916
SCAN=     3  PEL.NO.=       430  VECT.NO.=  200          P/V=      2.14999967
SCAN=     4  PEL.NO.=       679  VECT.NO.=  300          P/V=      2.26333332
SCAN=     6  PEL.NO.=       988  VECT.NO.=  400          P/V=      2.46999931
SCAN=     7  PEL.NO.=      1314  VECT.NO.=  500          P/V=      2.62799931
SCAN=    10  PEL.NO.=      1912  VECT.NO.=  600          P/V=      3.18666649
SCAN=    12  PEL.NO.=      2281  VECT.NO.=  700          P/V=      3.25857367
SCAN=    15  PEL.NO.=      2795  VECT.NO.=  800          P/V=      3.49374962
SCAN=    18  PEL.NO.=      3448  VECT.NO.=  900          P/V=      3.83111095
SCAN=    22  PEL.NO.=      4157  VECT.NO.= 1000          P/V=      4.15699959
SCAN=    25  PEL.NO.=      4749  VECT.NO.= 1100          P/V=      4.31727219
SCAN=    30  PEL.NO.=      5762  VECT.NO.= 1200          P/V=      4.80166626
SCAN=    34  PEL.NO.=      6651  VECT.NO.= 1300          P/V=      5.11613372
SCAN=    43  PEL.NO.=      8298  VECT.NO.= 1400          P/V=      5.92714214
SCAN=    48  PEL.NO.=      9306  VECT.NO.= 1500          P/V=      6.20399952
SCAN=    52  PEL.NO.=     10107  VECT.NO.= 1600          P/V=      6.31687453
SCAN=    56  PEL.NO.=     10810  VECT.NO.= 1700          P/V=      6.35882282
SCAN=    62  PEL.NO.=     12119  VECT.NO.= 1800          P/V=      6.73277760
SCAN=    68  PEL.NO.=     13266  VECT.NO.= 1900          P/V=      6.98210526
SCAN=    72  PEL.NO.=     13939  VECT.NO.= 2000          P/V=      6.96949959
SCAN=    75  PEL.NO.=     14672  VECT.NO.= 2100          P/V=      6.98666573
SCAN=    78  PEL.NO.=     15162  VECT.NO.= 2200          P/V=      6.89181805
SCAN=    80  PEL.NO.=     15646  VECT.NO.= 2300          P/V=      6.80260849
SCAN=    83  PEL.NO.=     16197  VECT.NO.= 2400          P/V=      6.74074573
SCAN=    85  PEL.NO.=     16546  VECT.NO.= 2500          P/V=      6.61839962
SCAN=    87  PEL.NO.=     16983  VECT.NO.= 2600          P/V=      6.53192234
SCAN=    89  PEL.NO.=     17408  VECT.NO.= 2700          P/V=      6.44741677
SCAN=    92  PEL.NO.=     17897  VECT.NO.= 2800          P/V=      6.39178562
SCAN=    94  PEL.NO.=     18304  VECT.NO.= 2900          P/V=      6.33930969
SCAN=    97  PEL.NO.=     18873  VECT.NO.= 3000          P/V=      6.29099941
SCAN=    99  PEL.NO.=     19328  VECT.NO.= 3100          P/V=      6.23483849
SCAN=   101  PEL.NO.=     19794  VECT.NO.= 3200          P/V=      6.18562412
SCAN=   104  PEL.NO.=     20323  VECT.NO.= 3300          P/V=      6.15848446
SCAN=   107  PEL.NO.=     20855  VECT.NO.= 3400          P/V=      6.13382339
SCAN=   110  PEL.NO.=     21393  VECT.NO.= 3500          P/V=      6.11142826
SCAN=   112  PEL.NO.=     21952  VECT.NO.= 3598          P/V=      6.10116673
NO.OCCURRENCES=     1 HOW MANY=      1565 PERCENT= 43.4964
NO.OCCURRENCES=     2 HOW MANY=       582 PERCENT= 16.1756
NO.OCCURRENCES=     3 HOW MANY=       351 PERCENT=  9.7554
NO.OCCURRENCES=     4 HOW MANY=       205 PERCENT=  5.6976
NO.OCCURRENCES=     5 HOW MANY=       122 PERCENT=  3.3908
NO.OCCURRENCES=     6 HOW MANY=        87 PERCENT=  2.4180
NO.OCCURRENCES=     7 HOW MANY=        77 PERCENT=  2.1401
NO.OCCURRENCES=     8 HOW MANY=        54 PERCENT=  1.5008
NO.OCCURRENCES=     9 HOW MANY=        53 PERCENT=  1.4730
NO.OCCURRENCES=    10 HOW MANY=        42 PERCENT=  1.1673
NO.OCCURRENCES=    11 HOW MANY=        28 PERCENT=  0.7782
NO.OCCURRENCES=    12 HOW MANY=        35 PERCENT=  0.9728
NO.OCCURRENCES=    13 HOW MANY=        33 PERCENT=  0.9172
NO.OCCURRENCES=    14 HOW MANY=        28 PERCENT=  0.7782
NO.OCCURRENCES=    15 HOW MANY=        22 PERCENT=  0.6115
NO.OCCURRENCES=    16 HOW MANY=        19 PERCENT=  0.5281
NO.OCCURRENCES=    17 HOW MANY=        19 PERCENT=  0.5281
NO.OCCURRENCES=    18 HOW MANY=        16 PERCENT=  0.4447
NO.OCCURRENCES=    19 HOW MANY=        17 PERCENT=  0.4725
NO.OCCURRENCES=    20 HOW MANY=         4 PERCENT=  0.1112
NO.OCCURRENCES=    21 HOW MANY=        15 PERCENT=  0.4169
NO.OCCURRENCES=    22 HOW MANY=        15 PERCENT=  0.4169
```

```
NO.OCCURRENCES=   23 HOW MANY=        5 PERCENT=  0.1390
NO.OCCURRENCES=   24 HOW MANY=        8 PERCENT=  0.2223
NO.OCCURRENCES=   25 HOW MANY=        6 PERCENT=  0.1668
NO.OCCURRENCES=   26 HOW MANY=       12 PERCENT=  0.3335
NO.OCCURRENCES=   27 HOW MANY=       12 PERCENT=  0.3335
NO.OCCURRENCES=   28 HOW MANY=       10 PERCENT=  0.2779
NO.OCCURRENCES=   29 HOW MANY=        2 PERCENT=  0.0556
NO.OCCURRENCES=   30 HOW MANY=        4 PERCENT=  0.1112
NO.OCCURRENCES=   31 HOW MANY=        7 PERCENT=  0.1946
NO.OCCURRENCES=   32 HOW MANY=        7 PERCENT=  0.1946
NO.OCCURRENCES=   33 HOW MANY=        7 PERCENT=  0.1946
NO.OCCURRENCES=   34 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=   35 HOW MANY=        5 PERCENT=  0.1390
NO.OCCURRENCES=   36 HOW MANY=        2 PERCENT=  0.0556
NO.OCCURRENCES=   37 HOW MANY=        3 PERCENT=  0.0834
NO.OCCURRENCES=   38 HOW MANY=        5 PERCENT=  0.1390
NO.OCCURRENCES=   39 HOW MANY=        6 PERCENT=  0.1668
NO.OCCURRENCES=   40 HOW MANY=        4 PERCENT=  0.1112
NO.OCCURRENCES=   41 HOW MANY=        4 PERCENT=  0.1112
NO.OCCURRENCES=   42 HOW MANY=        3 PERCENT=  0.0834
NO.OCCURRENCES=   43 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=   44 HOW MANY=        2 PERCENT=  0.0556
NO.OCCURRENCES=   45 HOW MANY=        6 PERCENT=  0.1668
NO.OCCURRENCES=   46 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=   47 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=   48 HOW MANY=        3 PERCENT=  0.0834
NO.OCCURRENCES=   49 HOW MANY=        2 PERCENT=  0.0556
NO.OCCURRENCES=   50 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=   52 HOW MANY=        5 PERCENT=  0.1390
NO.OCCURRENCES=   53 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=   54 HOW MANY=        6 PERCENT=  0.1668
NO.OCCURRENCES=   55 HOW MANY=        2 PERCENT=  0.0556
NO.OCCURRENCES=   58 HOW MANY=        2 PERCENT=  0.0556
NO.OCCURRENCES=   60 HOW MANY=        5 PERCENT=  0.1390
NO.OCCURRENCES=   61 HOW MANY=        2 PERCENT=  0.0556
NO.OCCURRENCES=   62 HOW MANY=        2 PERCENT=  0.0556
NO.OCCURRENCES=   63 HOW MANY=        3 PERCENT=  0.0834
NO.OCCURRENCES=   64 HOW MANY=        2 PERCENT=  0.0556
NO.OCCURRENCES=   65 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=   66 HOW MANY=        2 PERCENT=  0.0556
NO.OCCURRENCES=   67 HOW MANY=        2 PERCENT=  0.0556
NO.OCCURRENCES=   68 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=   69 HOW MANY=        2 PERCENT=  0.0556
NO.OCCURRENCES=   70 HOW MANY=        2 PERCENT=  0.0556
NO.OCCURRENCES=   71 HOW MANY=        3 PERCENT=  0.0834
NO.OCCURRENCES=   73 HOW MANY=        3 PERCENT=  0.0834
NO.OCCURRENCES=   76 HOW MANY=        2 PERCENT=  0.0556
NO.OCCURRENCES=   77 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=   78 HOW MANY=        3 PERCENT=  0.0834
NO.OCCURRENCES=   79 HOW MANY=        2 PERCENT=  0.0556
NO.OCCURRENCES=   80 HOW MANY=        2 PERCENT=  0.0556
NO.OCCURRENCES=   81 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=   82 HOW MANY=        2 PERCENT=  0.0556
NO.OCCURRENCES=   83 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=   84 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=   85 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=   87 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=   89 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=   92 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=   94 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=   97 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=   99 HOW MANY=        1 PERCENT=  0.0278
NO.OCCURRENCES=  100 HOW MANY=       10 PERCENT=  0.2779
NO.OCCURRENCES=  200 HOW MANY=        1 PERCENT=  0.0278

 ELAPSED TIME   --   MINUTES -  0  SECONDS -  5.11
```

APPROVAL

A FAST ROUTINE FOR COMPUTING MULTIDIMENSIONAL HISTOGRAMS

By Robert R. Jayroe, Jr.

The information in this report has been reviewed for security classification. Review of any information concerning Department of Defense or Atomic Energy Commission programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

This document has also been reviewed and approved for technical accuracy.

J. T. POWELL
Director, Data Systems Laboratory

☆U.S. GOVERNMENT PRINTING OFFICE 1977—740-193/136 REGION NO. 4