A Median Filter Subroutine

In the near future the IUE Spectra Image Processing System will be modified to incorporate median filtering of the background spectrum (Turnrose, Harvel, and Bohlin 1979). Such filtering will remove more cleanly the discrepant points, such as reseaux, radiation spots, tube blemishes, and Lyman alpha spillover. To aid the Guest Observer who is reprocessing his SWP spectra because of the ITF error to take advantage of median filtering, we present here FORTRAN IV coding for a subroutine, SMOOTH. SMOOTH will perform median filtering on a spectrum in the IUE format.

SMOOTH will filter a spectrum of up to 6ØØ samples with a width of 1 to 31 points. The first two entries in the input data array are ignored because these contain no samples in the IUE format.

Near the ends of the spectrum, the width of the filter is expanded or contracted automatically so the the window just reaches the first or last valid data point. This variable width assures the maximum filtering while avoiding manufacturing data at the beginning and end of the spectrum.

In addition to producing a classical median filtering, SMOOTH can be used to produce a smoothed median. In this case, the samples in the spectrum are replaced by an average of a specifiable number of points from the middle of the ordered list.

The input parameters and options are explained in detail in the comments at the beginning of the subroutine. The filtered spectrum is output in the same array that contained the unfiltered spectrum. An error flag is also returned.

The algorithm has been tested only for odd values of the filter width. It should also work for even values of the filter width.

F. H. Schiffer, III

A. V. Holm

Reference

Turnrose, B.E., Harvel, C.A., and Bohlin, R.C. 1979, NASA IUE Newsletter, No. 7, p. 9.

```
      SUBROUTINE  SMOOTH(DATA,NPNTS,NMED,NSAM,NERR)
C     THIS SUBROUTINE WILL APPLY A MEDIAN FILTER TO THE DATA
C       WITHIN THE ARRAY DATA.
C     DATA - A REAL ARRAY CONTAINING THE DATA, THE FIRST TWO
C          WORDS OF THE ARRAY ARE NOT INCLUDED IN THE FILTERING
C     NPNTS - THE NUMBER OF DATA POINTS TO BE INCLUDED IN THE
C          FILTERING PROCESS.
C     NMED -  THE WIDTH OF THE MEDIAN FILTER.
C     NSAM -  THE NUMBER OF POINTS WITHIN THE ORDERED LIST TO
C          BE AVERAGED TO GET THE SMOOTHED VALUE.  NSAM=1
C          CORRESPONDS TO THE CLASSICAL MEDIAN FILTER.
C     NERR -  AN ERROR FLAG RETURNED TO INDICATE THE STATUS OF
C          THE REQUESTED SMOOTHING.
C          NERR=Ø  THE SMOOTHING WAS DONE WITH NO ERRORS.
C          NERR=1  THE MEDIAN WIDTH WAS ONE = NO SMOOTHING.
C          NERR=2  THE MEDIAN WIDTH WAS EVEN.  NO MEDIAN VALUE
C                    EXISTS.  DATA AVERAGED TO GET MEDIAN VALUE.
C          NERR=4  SAMPLING WIDTH EQUALS MEDIAN WIDTH.  SMOOTHING
C                    EQUIVALENT TO BOX FILTER NSAM WIDE.
C          NERR=8  SAMPLING WIDTH EQUAL TO ZERO.  MEDIAN POINT USED.
C          NERR=16 MEDIAN WIDTH WIDER THAN THE DATA ARRAY.
C          NERR=1ØØ  MEDIAN WIDTH TOO WIDE FOR SUBROUTINE ARRAYS.
C          NERR=2ØØ  SAMPLING WIDTH WIDER THAN MEDIAN WIDTH.
C          NERR=4ØØ  MEDIAN WIDTH TOO SMALL ( LESS THAN ONE).
C          NERR=8ØØ  SAMPLING WIDTH TOO SMALL (LESS THAN ZERO).
C        IF NERR RETURNED IS LESS THAN 1ØØ THE FILTERING WAS DONE
C          SUBJECT TO THE WARNINGS GIVEN.  IF NERR IS GREATER
C          THAN 99 NO  SMOOTHING WAS DONE.
C
C     THE SUBROUTINE USES TWO ARRAYS - QUE AND ORDER - TO SORT THE
C        DATA.  THE SIZE OF THESE ARRAYS DETERMINES THE MAXIMUM
C        FILTER WIDTH ALLOWED, WHICH IS ALSO CONTAINED IN THE
C        VARIABLE MAXW.
C
C     DECLARATIONS
      REAL DATA,QUE,ORDER,X,Y,Z
      INTEGER NPNTS,NMED,NSAM,NERR,I,J,KB,KE,L,M,MAXW,NQUE
      DIMENSION  DATA(602),QUE(31),ORDER(31)
      DATA MAXW/31/
C
C     MAXIMUM  FILTER WIDTH SET AT 31 POINTS
C
C     ERROR CHECKING
C
      NERR=Ø
      IF (NMED.GT.MAXW) NERR=1ØØ
      IF (NSAM.GT.NMED) NERR=NERR+2ØØ
      IF (NMED.LT.1)     NERR=NERR+4ØØ
      IF (NSAM.LT.Ø)     NERR=NERR+8ØØ
      IF (NPNTS.LE.Ø)    NERR=NERR+16ØØ
C       ERROR EXIT - FATAL ERRORS
      IF (NERR.GT.Ø)  RETURN
      IF (NMED.EQ.1)     NERR=1
      IF (NMED/2.EQ.(NMED+1)/2)  NERR=NERR+2
      IF (NSAM.EQ.NMED) NERR=NERR+4
      IF (NSAM.EQ.Ø)     NERR=NERR+8
      IF (NMED.GE.NPNTS)NERR=NERR+16
C
```

```
C         INITIALIZE THE ARRAYS
      DO 59 M=1,NMED
      QUE(M)=DATA(3)
      ORDER(M)=DATA(3)
59    CONTINUE
      NQUE=1
      L=NPNTS+2
      IMED=1
      DO 499 M=3,L
      Y=ORDER(IMED)
      X=QUE(NQUE)
      IF (M+IMED/2.GT.L) GO TO 85
C         NOT CONTRACTING
      Y=DATA(M+IMED/2)
      IF (IMED.GE.NMED) GO TO 85
      IF (M-IMED/2.LE.3) GO TO 85
C         EXPANDING
84    X=ORDER(IMED)+1.Ø
      IMED=IMED+1
      QUE(NQUE)=Y
      NQUE=NQUE+1
      IF (NQUE.GT.NMED) NQUE=1
      GO TO 2ØØ
C         NORMAL OR CONTRACTING
85    QUE(NQUE)=Y
      NQUE=NQUE+1
      IF (NQUE.GT.NMED) NQUE=1
      IF (X-Y) 1ØØ,3ØØ,2ØØ
C         MOVE DATA TOWARDS BEGINNING OF ORDERED LIST
1ØØ   DO 1Ø9 J=2,IMED
      IF (ORDER(J).LE.X) GO TO 1Ø9
      IF (ORDER(J).LT.Y) GO TO 1Ø5
      ORDER(J-1)=Y
      GO TO 199
1Ø5   ORDER(J-1)=ORDER(J)
1Ø9   CONTINUE
      ORDER(IMED)=Y
199   GO TO 3ØØ
C         MOVE DATA TOWARDS END OF ORDERED LIST
2ØØ   DO 2Ø9 J=2,IMED
      IF (ORDER(J-1).LE.Y) GO TO 2Ø9
      Z=ORDER(J-1)
      ORDER(J-1)=Y
      Y=Z
      IF (X.LE.Z) GO TO 299
2Ø9   CONTINUE
      ORDER(IMED)=Y
299   CONTINUE
3ØØ   CONTINUE
      IF (M+IMED/2-L) 3Ø3,3Ø6,3Ø1
C         CONTRACTING MEDIAN WIDTH
3Ø1   IMED=IMED-2
```

```
      Y=ORDER(IMED+1)
      X=QUE(NQUE)
      GO TO 85
3Ø3   IF (IMED.EQ.NMED) GO TO 3Ø6
      IF (M-(IMED-1)/2.LE.3) GO TO 3Ø6
C       EXPANDING MEDIAN WIDTH
      Y=DATA(M+(IMED+1)/2)
      GO TO 84
3Ø6   CONTINUE
C       REPLACE DATA(M) WITH MEDIAN FROM ORDERED LIST
      KB=(IMED-NSAM+2)/2
      KE=(IMED+NSAM+1)/2
      IF (KB.LT.1) KB=1
      IF (KE.GT.IMED) KE=IMED
      X=Ø.Ø
      DO 3Ø9 J=KB,KE
      X=X+ORDER(J)
3Ø9   CONTINUE
      IF (KE-KB.GT.Ø) X=X/(KE-KB+1)
      DATA(M)=X
499   CONTINUE
      RETURN
      END
```