CORRELATION STACK

SEISMIC DATA PROCESSING TECHNIQUE

Ralph Knapp

August 1, 1991

KGS Open-file Report 91-28

CORRELATION STACK

Correlation stack is a standard processing package that evaluates traces within a common-midpoint (CMP) gather according to how well they correlate with a pilot trace and weights the trace appropriately prior to stack. [Note that CMP and CDP are equivalent nomenclature.] The result is that the traces that best fit the pilot trace influence the stacked results the most, thereby reducing noise from non-reflection signal.

A correlation stack has been implemented on the Data General MV 20 000 computer of the Kansas Geological Survey. This version evaluates traces on a window-by-window basis. Thus, if only a portion of the trace is of good quality, only that portion will influence the stack. For instance, groundroll, which may exist on only a portion of the trace will not stack, but the rest of the trace might. Window length is user-defined and should be large enough to well-represent a series of reflections. To enhance run speed, not every point of the trace is evaluated, and a user-defined step is used. The step should be on the order of 1/5 to 1/10 the length of the window. Further, dynamic adjustments are allowed to the trace. Prior to stack the portion of the trace may be time shifted to improve the stack. The shift helps adjust for normal moveout (NMO) error or statics error, although for best results, statics and NMO must be done as well as possible prior to correlation stack. The limit of the shift is user-defined and should be only a fraction of a cycle or equal to maximum static and NMO errors expected. A cutoff value for correlation can be user-defined. This parameter is not too critical but should be a value between 0.0 and about 0.8. It should definitely not be less than zero, and for poorer quality

data it should probably be equal to 0.0. Finally, a user-defined parameter to make the stack weight equal to a power of the correlation coefficient is provided. This value should be 2 or 4 and it improves the discrimination of the correlation stack by making the weight of a window with good correlation much larger than the weight of a window with poor correlation. Figure 1 shows how this parameter functions.

Figure 2 shows the results of correlation stack. Figure 3 is the results of the best stack available prior to correlation stack. Figure 4 is the section of the pilot traces, a running mix of Figure 3. Compare Figures 2 and 3 and note the improvements. Most dramatic is that shallow reflections from less than 60 ms are brought out better either because of the dynamic timing correction or the muting of noise that was reducing the effectiveness of the stack of Figure 3. At about CMP 475 and 65 ms the stack of a thin bed is dramatically improved, probably due to slight timing corrections that allow the high frequencies to stack. Deeper reflections (greater than 100 ms) are improved on the correlation stack because of the reduction of random noise. In general, Figure 2 shows much improvement in the signal-noise-ratio and in the coherency and resolution of reflections.

Table 1 reviews the parameters of the correlation stack. Table 2 shows a sample input deck and the execution procedure for the example shown in Figure 2. Table 3 is an example of the input deck and execution procedure for the pilot trace. The program is listed in Appendix 1.

APPENDIX 1. Program CSTAK

```
      PROGRAM CSTAK
*
* CORRELATION STACK 91-07-09
*
      integer*2 j,reel1(1600),k,reel2(200),trh(120),TRH1(120)
      INTEGER*2 NSAMP,INCRE,ITYPE
      real*4 hamp(2048),HAMP1(2048),HAMPOUT(2048),WT(2048),SWT(2048)
      REAL*4 SHIFT(2048)
      integer param
      character*30 datain,datain1,dataout,filein,fileout
      EQUIVALENCE(RANGE,TRH(19))
*     l=param('ga',2,0,FILEOUT,lerror)
*     OPEN(6,FILE=FILEOUT)
      l=param('ga',1,0,FILEIN ,lerror)
      OPEN(5,FILE=FILEIN, PAD='YES')
      READ(5,'(A30)') DATAIN,DATAIN1,DATAOUT
* DATAIN  - moveout corrected, statics corrected data ready for stack
* DATAIN1 - pilot traces: stacked and mixed section
* DATAOUT - stacked output section
      PRINT *,' CORRELATION STACK 91-07-09'
      PRINT *,' INPUT DATA = ', DATAIN
      PRINT *,' PILOT TRACE = ', DATAIN1
      PRINT *,' OUTPUT DATA = ', DATAOUT
      READ (5,*) XLENGTH,STEP,SEARCH,SMOOTH,cut,power
      PRINT *,' LENGTH OF CORRELATION (MS) = ',XLENGTH
      PRINT *,' STEP (MS) = ', STEP
      PRINT *,' MAXIMUM SEARCH LENGTH (MS) = ', SEARCH
      PRINT *,' LENGTH OF SMOOTHING (MS) = ', SMOOTH
      print *,' CORRELATION CUTOFF = ',CUT
      PRINT *,' CORRELATION POWER = ', POWER
      open(1,file=dataIN ,iointent= 'INput',mode='binary',recfm='dynamic',form='unfo
      open(3,file=dataIN1,iointent= 'INput',mode='binary',recfm='dynamic',form='unfo
      open(2,file=dataOUT,iointent='OUTput',mode='binary',recfm='dynamic',form='unfo
      READ(3)J,(REEL1(I),I=1,J),K,(REEL2(I),I=1,K)
      PRINT *,' J K = ',J,K
      NSAMP=REEL2(11)
      ITYPE=REEL2(13)
      INCRE=REEL2(9)
      READ(1)J,(REEL1(I),I=1,J),K,(REEL2(I),I=1,K)
      PRINT *,' J K = ',J,K
      IERR=0
      IF(INCRE.NE.REEL2(9))  IERR=1
      IF(NSAMP.NE.REEL2(11)) IERR=1
      IF(ITYPE.NE.REEL2(13)) IERR=1
      IF(IERR.NE.0) THEN
            PRINT *,' PARAMETER MISMATCH OF INPUT FILES: '
            PRINT *,' NSAMP = ', NSAMP, REEL2(11)
            PRINT *,' ITYPE = ', ITYPE, REEL2(13)
            PRINT *,' INCRE = ', INCRE, REEL2(9)
            STOP
      ENDIF
*
      PRINT *,' NSAMP = ',NSAMP
      PRINT *,' INCRE = ',INCRE
      WRITE(2)J,(REEL1(I),I=1,J),K,(REEL2(I),I=1,K)
      SI=FLOAT(INCRE)/1000000.
      PRINT *,'    SI = ',SI
      LENGTH=IFIX(XLENGTH/1000./SI+.5)        ! IN POINTS
      ISTEP =IFIX(STEP   /1000./SI+.5)
      ISMOO =IFIX(SMOOTH /1000./SI+.5)
      ISEAR =IFIX(SEARCH /1000./SI+.5)
*
      PRINT *,' LENGTH = ',LENGTH
      PRINT *,'  ISTEP = ',ISTEP
      PRINT *,'  ISMOO = ',ISMOO
```

```
      PRINT *,'  ISEAR = ',ISEAR
*
      trh(12)=0                                              !PRESET
1     CALL TREAD(TRH1,HAMP1,NSAMP,ITYPE,3,IERR)! READ PILOT TRACE
      IF(IERR.NE.0) GO TO 10000
      DO 101 I=1,NSAMP                          ! PRESET VALUES
      WT(I)=0.
      SWT(I)=0.
      SHIFT(I)=0.
101   HAMPOUT(I)=0.
      if(trh(12).ge.trh1(12)) go to 102 ! IF PILOT MATCHES DATA
11    CALL TREAD(TRH,HAMP,NSAMP,ITYPE,1,IERR)! READ DATA TRACE
      IF(IERR.NE.0) GO TO 10000
102   continue                                  ! START PROCESS
      IF(TRH(12).LT.TRH1(12)) GO TO 11          ! READ NEXT DATA
      IF(TRH(12).GT.TRH1(12)) GO TO 1000        ! ALL DATA READ
      if(trh(15).ne.1) go to 11                 ! DEAD TRACE
*
      I1=LENGTH/2+1+ISEAR                       ! FIRST POINT OF DATA
      I2=NSAMP-(LENGTH/2+1)                     ! LAST  POINT
      DO 200 I=I1,I2,ISTEP                      ! DO OVER ALL DATA
      ISIGN=+1                                  ! STEP DIRECTION
      IFLAG=0                                   ! FIRST MATCH
      ISHIFT=0                                  ! PRESET SHIFT VALUE
      N=NSAMP
      III=I
      CALL CROSS(CORR0,HAMP(I+ISHIFT),HAMP1(I),LENGTH)
201   ISHIFT=ISHIFT+ISIGN
      CALL CROSS(CORR1,HAMP(I+ISHIFT),HAMP1(I),LENGTH)
      IF(CORR1.LT.CORR0) THEN                   ! CHANGE DIRECTION OF
      ISIGN=-ISIGN                              !     SEARCH
      IF(IFLAG.NE.0) GO TO 202                  ! IF ITS NOT THE FIRST
      ENDIF                                     !     MATCH
      IFLAG=1
      CORR0=CORR1
      IF(IABS(ISHIFT).LE.ISEAR) GO TO 201! IF WITHIN SEARCH BOUNDS
202   CONTINUE
      ISHIFT=ISHIFT+ISIGN                       ! SHIFT VALUE SET
      WT(I)=0.
      SHIFT(I)=0.
      IF((IABS(ISHIFT).LE.ISEAR) .AND. (CORR0.GE.CUT)) THEN
              WT(I)=CORR0**POWER
              SHIFT(I)=FLOAT(ISHIFT)
              ENDIF
      IF(I.EQ.I1) THEN                          ! FIRST VALUE OF DATA
              DO 2021 II=1,I1-1                 ! FILL IN
              WT(II)=WT(I1)
              SHIFT(II)=SHIFT(I1)
2021          CONTINUE
              ENDIF
      DO 2023 II=-(ISTEP+1)/2+1,NSAMP-I ! SET TO END OF TRACE
      WT(I+II)=WT(I)
2023  SHIFT(I+II)=SHIFT(I)
200   CONTINUE
      n=nsamp
      CALL FILT(WT,N   ,ISMOO)
      CALL FILT(SHIFT,N,ISMOO)
      DO 300 I=1,NSAMP
      IF(I+FLOAT(SHIFT(I)+.5).LT.1) GO TO 300    ! OUTSIDE ARRAY
      IF(I+FLOAT(SHIFT(I)+.5).GT.NSAMP) GO TO 300! OUTSIDE ARRAY
      HAMPOUT(I)=HAMPOUT(I)+HAMP(I+FLOAT(SHIFT(I)+.5))*WT(I)
      SWT(I)=SWT(I)+WT(I)
300   CONTINUE
      GO TO 11
*
1000  DO 1001 I=1,NSAMP
```

```
        IF(SWT(I).EQ.0.) THEN
                HAMPOUT(I)=0.
        ELSE
                HAMPOUT(I)=HAMPOUT(I)/SWT(I)
        ENDIF
1001    CONTINUE
        TRH1(88)=1
        J=120+2*NSAMP
        PRINT *,' WRITE ',TRH1(12)
        write(2)j,trh1,(hampOUT(I),I=1,NSAMP)
        go to 1
10000   close(1)
        close(5)
        close(6)
        stop
        end
*
        SUBROUTINE CROSS(CORR,A,B,L)
        real*8 ab,a2,b2
        DIMENSION A(1),B(1)
        XL=FLOAT(L+1)
        AB=0.
        A2=0.
        B2=0.
        CORR=0.
        DO 1 I=-L/2,L/2
        AB=AB+(A(I+1)/XL)*(B(I+1)/XL)
        A2=A2+(A(I+1)/XL)**2
        B2=B2+(B(I+1)/XL)**2
1       CONTINUE
        IF (A2.LT.1.E-8 .OR. B2.LT.1.E-8) RETURN
        CORR=AB/dSQRT(A2)/dSQRT(B2)
        RETURN
        END
*
        SUBROUTINE FILT(A,N,L)
        DIMENSION A(1),B(2048)
        XL=L
        DO 10 I=L/2+1,N-L/2
        B(I)=0.
        AMX=0.
        AMN=99.
        DO 1 II=-L/2,L/2
                AMX=AMAX1(A(I+II),AMX)
                AMN=AMIN1(A(I+II),AMN)
1       B(I)=B(I)+A(I+II)/XL
10      B(I)=B(I)-(AMX+AMN)/XL  ! REMOVE MAX AND MIN VALUES
        DO 2 I=L/2+1,N-L/2
2       A(I)=B(I)
        DO 3 I=1,L/2
        A(I)=B(L/2+1)
3       A(N-I+1)=B(N-L/2)
        RETURN
        END
```

TABLE 1. CORRELATION STACK PARAMETERS

LENGTH - length of the correlation window. Should be about twice the width of a "reflection" cluster. [ms]

STEP - step function for the window. There should be 5 to 10 steps per window. [ms]

SEARCH - width of the limit of dynamic correction. In general, should not be more than half a cycle. Can be larger to accommodate large errors, in which case LENGTH should be larger by at least a factor of two to avoid mis-correlation. [ms]

SMOOTH - length of smoothing or averaging of weights and shifts. Use of this parameter avoids problems due to spikes in the values and provides a smooth correction function. The subroutine FILT uses SMOOTH. FILT is robust in that it throws out maximum and minimum extremes prior to smoothing. [ms]

CUT - cutoff value for the correlation. In general, CUT should be zero, but for good data it can have larger values. Remember that perfect correlation has a coefficient of 1.0, so CUT = 1.0 will remove all data. CUT = 0.8 is probably a reasonable value.

POWER - The correlation coefficient is raised to POWER to determine the weight. This improves the discrimination of the correlation stack and is the most powerful parameter. Should be equal to 2 or 4.

TABLE 2. EXECUTION AND SAMPLE INPUT DECK FOR CORRELATION STACK.

execution: X :UDD:GEOPHYSICS:CSTAK, *input, list*

input:

RWK.CMP.GATH	*CMP gather data, fully processed except for stack.*
RWK.STAK.RMIX7	*pilot traces, derived from running mix of stack.*
RWK.CSTAK	*output stacked results.*
30 2 3 30 0 4	*LENGTH, STEP, SEARCH, SMOOTH, CUT, POWER*

TABLE 3. EXECUTION AND SAMPLE INPUT DECK FOR PILOT SECTION.

execution: X :UDD:GEOPHYSICS:RMIX, *input, list*

input:

RWK.CMP.STAK	*input section: stacked version of RWK.CMP.GATH*
RWK.STAK.RMIX7	*output section: 7 trace tapered running mix section*
7 0	*number of traces to mix, do not honor boundaries*
1 2 3 4 3 2 1	*mix weights*

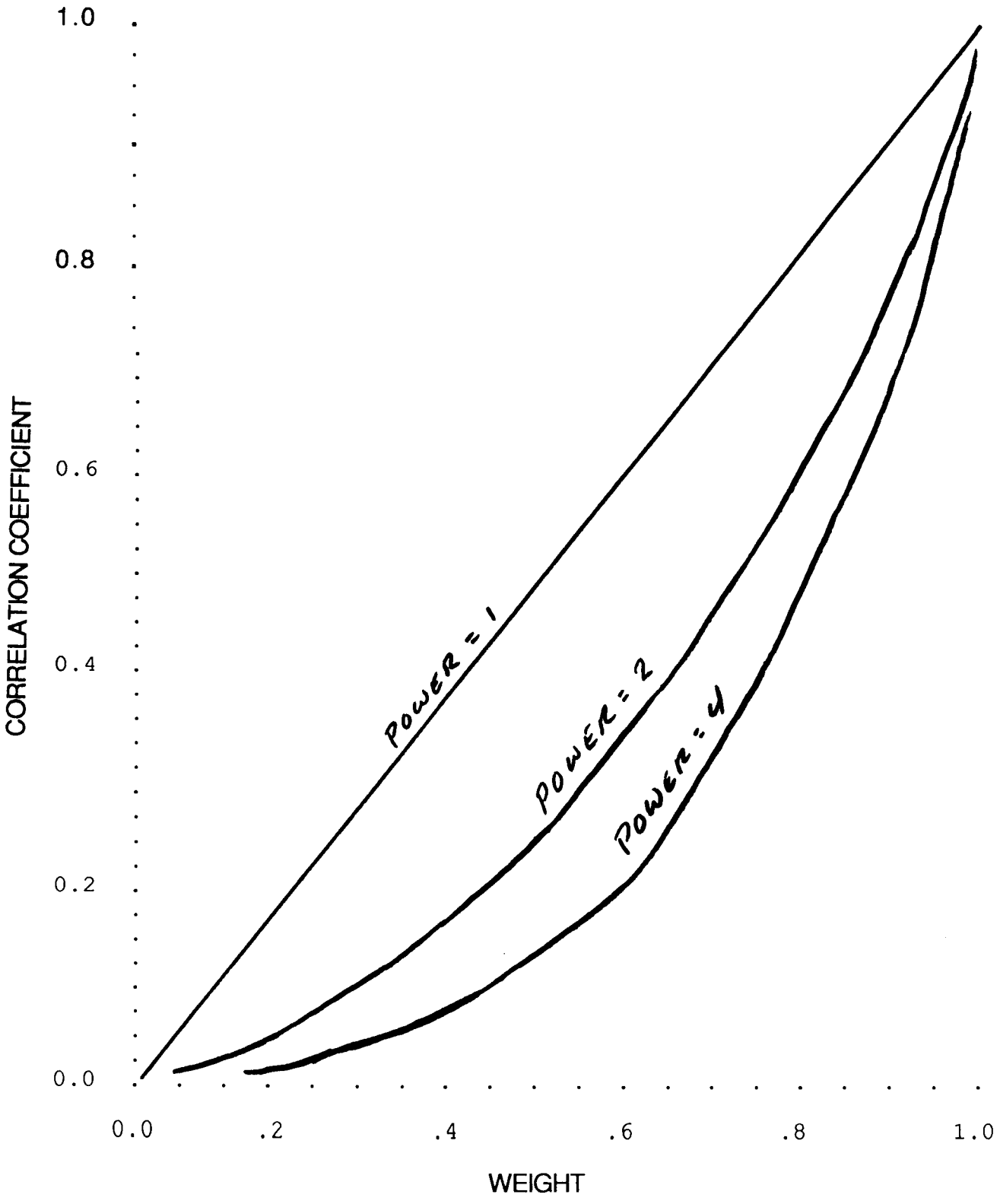

FIGURE 1. Weighting values as determined from the correlation coefficient for different values of POWER.

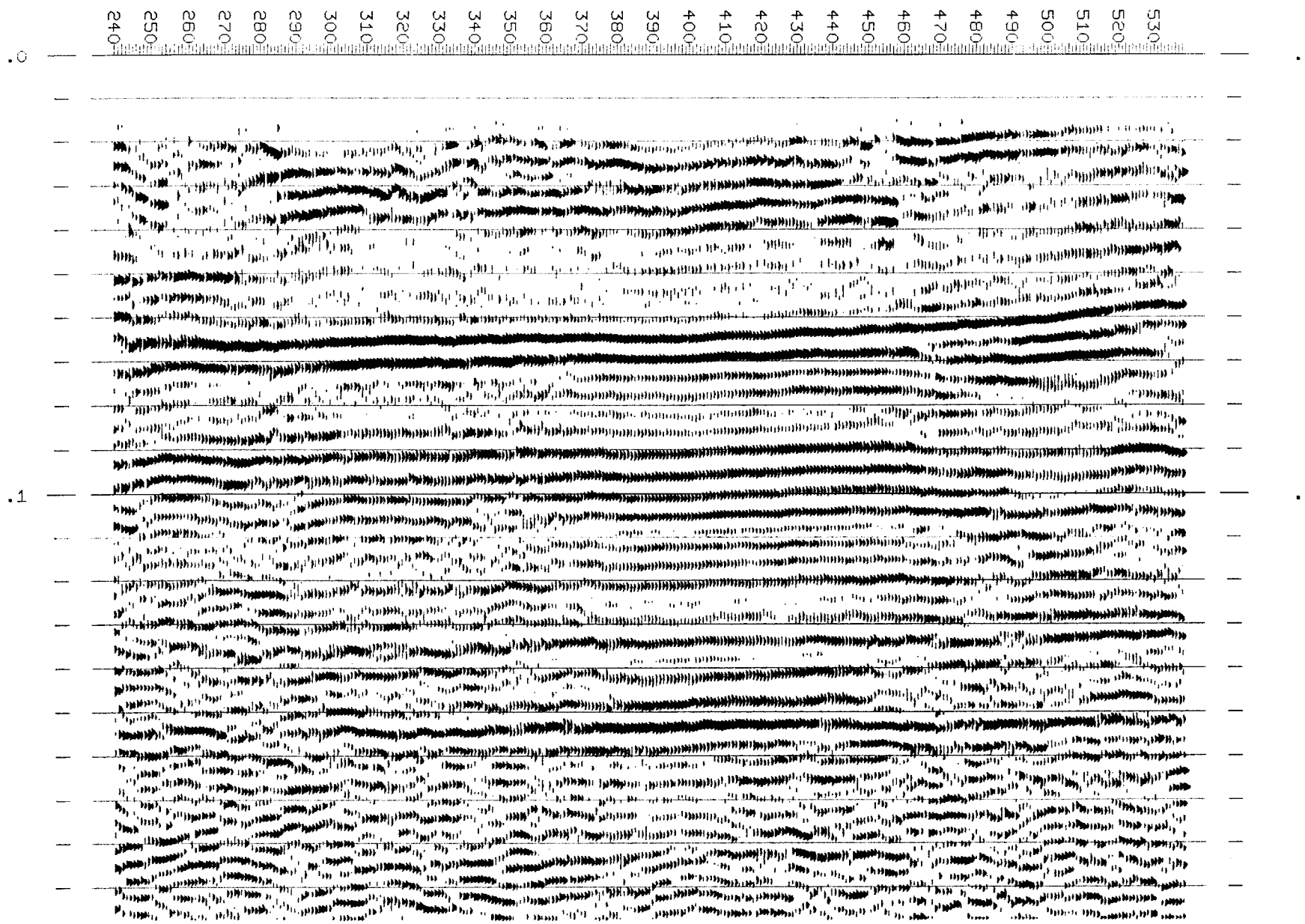

FIGURE 2. Correlation stack section.

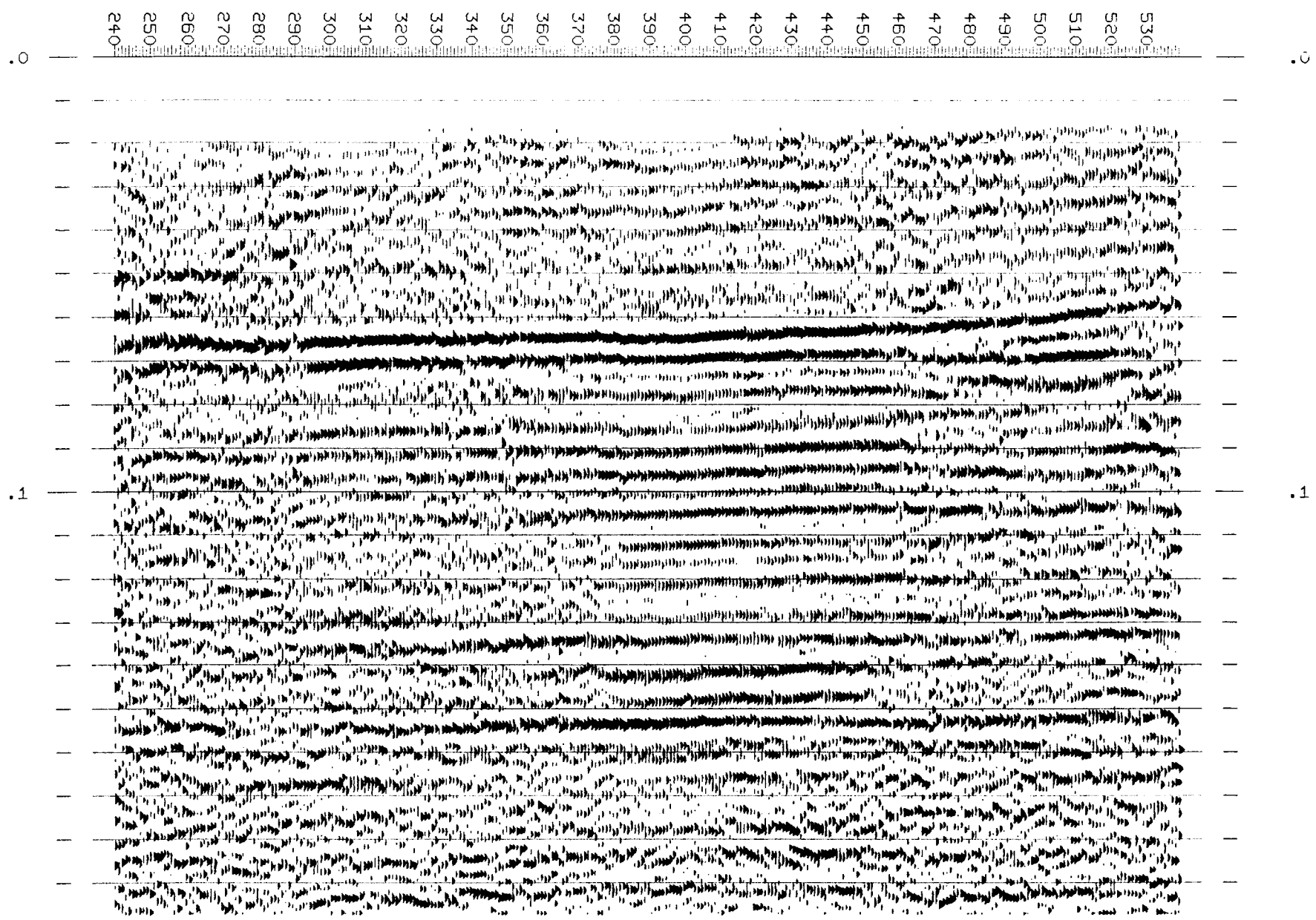

FIGURE 3. Final stack section of data without correlation stack.

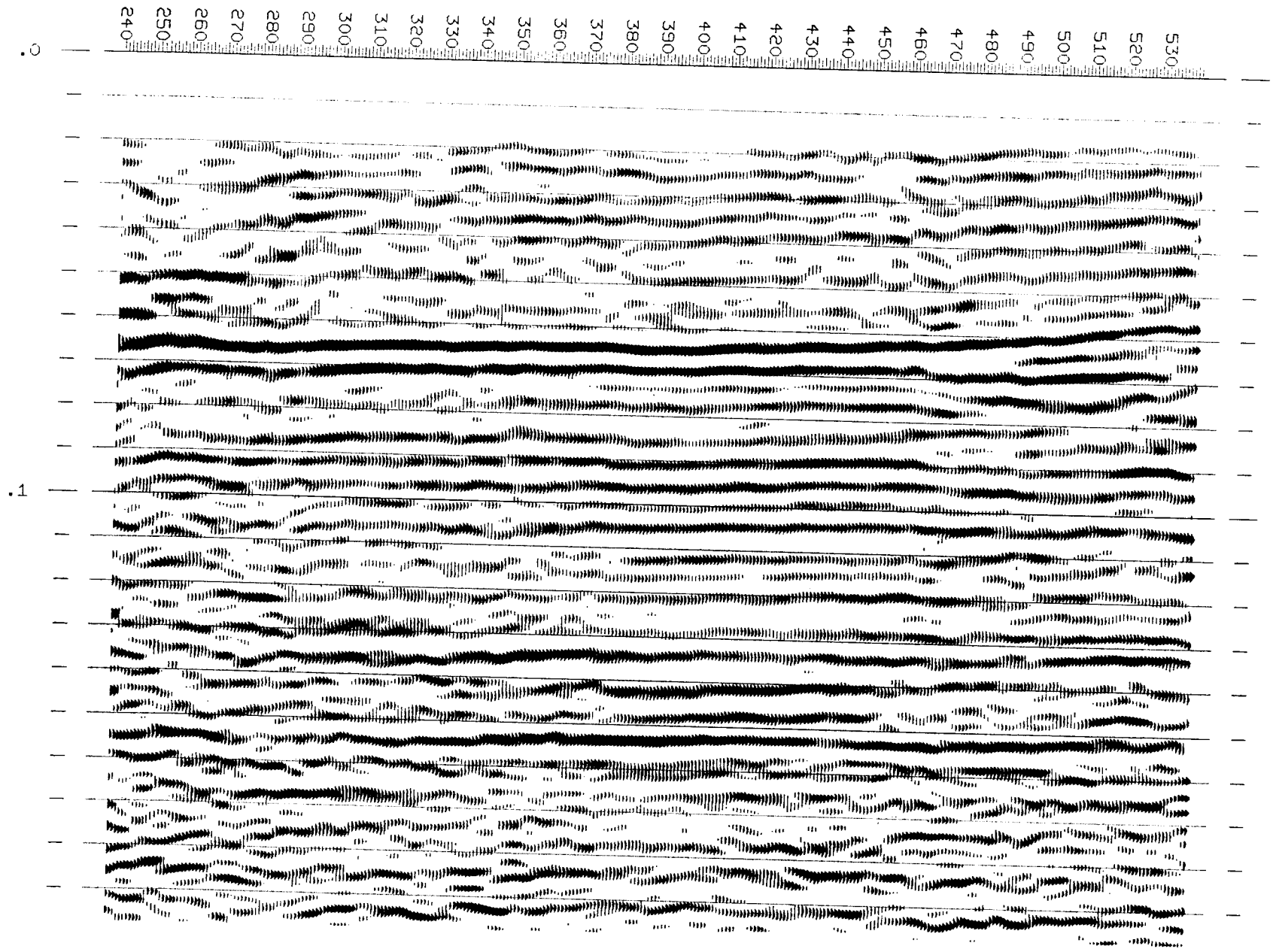

FIGURE 4. Pilot trace section.