Fine Tuning of Stretched-VISSR Image Mapping

Seiichiro Kigawa*

Abstract

Stretched-VISSR image mapping error may increase after satellite orbit control (station-keeping maneuver) or satellite attitude control (attitude maneuver). To solve this problem, a fine tuning technique of the Stretched-VISSR image for user's computer systems has been developed. This technique can be applied to both full disc images and half disc (northern hemisphere) images. The applicable theory and sample fine tuning programs written in FORTRAN are presented.

1. Introduction

Image mapping is used to process Visible and Infrared Spin Scan Radiometer (VISSR) image data, where each pixel of the VISSR image data must correspond to its respective position on earth. To improve the accuracy of image mapping, the Meteorological Satellite Center (MSC) developed an image mapping fine tuning technique called Distortion Data Determination (DDD) which is based on detection of the earth's edge from VISSR infrared image. DDD can be used to correct the processes of all products at the MSC, but it can not correct Stretched-VISSR data because of the broadcast time schedule. Therefore, the image mapping error of the Stretched-VISSR data may increase after satellite orbit control (station-keeping maneuver) or satellite attitude control (attitude maneuver). To solve this problem, a fine tuning technique of the Stretched-VISSR image for user's computer systems was newly developed. The accuracy of the newly-developed technique is within 1 pixel (infrared image). Moreover, this technique can be applied to both fulldisc images and half disc (northern hemisphere) images. This technique is very simple because it was designed for any small-scale computer system which can utilize the Stretched-VISSR data that is broadcasted via satellite. The applicable theory and sample fine tuning programs written in FORTRAN are presented.

---

* Meteorological Satellite Center

(Received December 1,1992;Revised January 12,1992)

2 Applicable Theory

2.1 Mapping Parameter Compilation

The following data are extracted from the documentation sector of the Stretched-VISSR data :

(a)Orbit and attitude prediction data block

(b)Constant parameters for simplified mapping block

(c)Parameters for simplified mapping block

2.2 Earth Edge Data Compilation

The earth edge data (actual earth edge) which meets the following conditions is extracted from Spacecraft and CDAS Status blocks in the documentation sector.

(a)SSL≧scan line number≧SSL−10 (for equator zone)

or

PIXEL No.	100	101	102	103	104	105
MGIVSR* RETURN CODE	6	6	6	0	0	0
SPACE OR EARTH	SPACE	SPACE	SPACE	EARTH	EARTH	EARTH

⇧ ESTIMATED EARTH EDGE (WEST)

* MGIVSR: Coordinate Transformation routine

Fig.1 Estimated earth edge

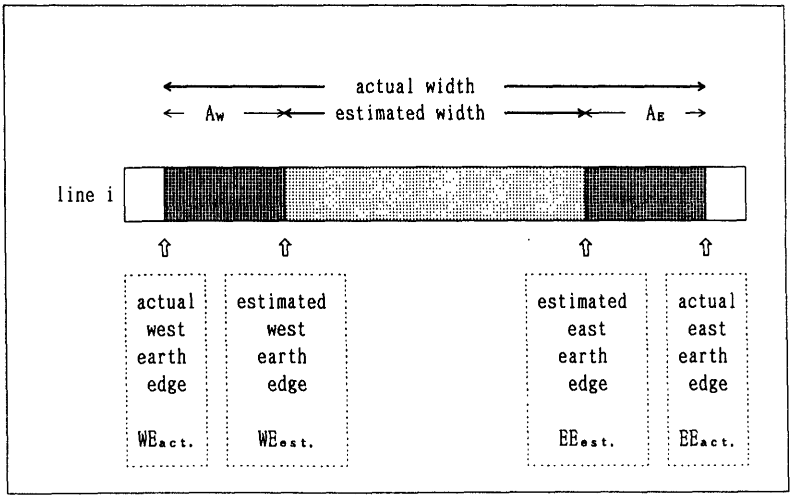
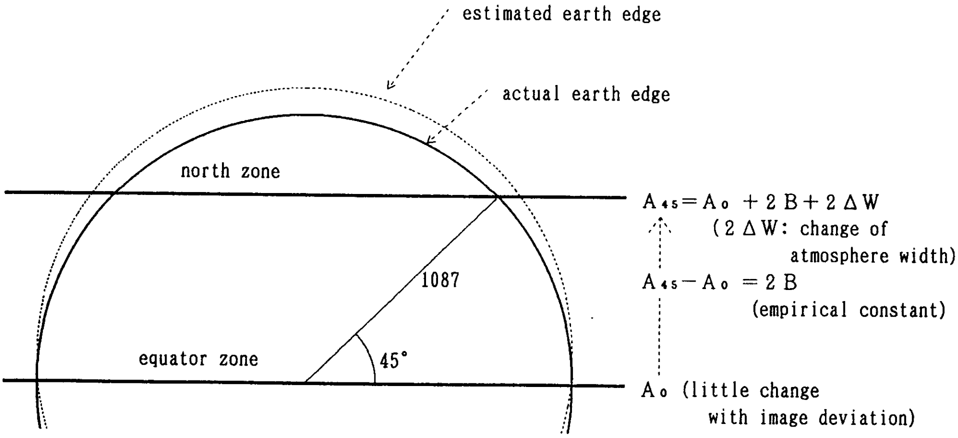


Fig.2 Atmosphere width



(b) $(SSL-1087/\sqrt{2})+5 \geqq$ scan line number $\geqq (SSL-1087/\sqrt{2})-5$ (for north zone)

where

SSL : sub-satellite line number

2.3 Mapping Parameter Correction

(a)Estimated Earth Edge

Estimated east and west earth edges are calculated from orbit and attitude data with the Coordinate Transformation routine (Fig.1).

(b)Atmosphere Width

An atmosphere width, A (Fig.2), is defined as the difference between the actual width of the earth and the estimated width of the earth on a no-deviation image which is given by :

$$A = A_W + A_E$$
$$= (WE_{est.} - WE_{act.}) + (EE_{act.} - EE_{est.})$$
$$= (EE_{act.} - WE_{act.}) - (EE_{est.} - WE_{est.}) \qquad (1)$$

where

A_w : west atmosphere width

A_E : east atmosphere width

$EE_{act.}$: actual east earth edge point

$WE_{act.}$: actual west earth edge point

$EE_{est.}$: estimated east earth edge point

$WE_{est.}$: estimated west earth edge point

(c)Atmosphere Width Change with Earth Image Deviation

The atmosphere width in the equator zone shows little change with deviation of the earth image. On the other hand, the difference of the atmosphere width between the equator zone and the north zone has empirically been shown to be constant. Therefore, the change of the atmosphere width on the north zone, $2\triangle W$, due to the north-south deviation of the earth image from its predicted position is given by (Fig.3):

$$\triangle W = (A_{45} - A_0)/2 - B \qquad (2)$$

where

A_{45} : atmosphere width in the north zone

A_0 : atmosphere width in the equator

zone (little change with image deviation)

B : empirical constant(+0.769)

(d)North-South Deviation

The rate of change of the earth width in the north zone (45 deg latitude) is almost 2 pixels per line, so 1 line of deviation of the earth image from its predicted position causes a 2-pixel change in the atmosphere width in the north zone. Therefore, the half of change of the atmosphere width in the north zone, $\triangle W$, is almost equal to the deviation of the earth image. The north-south deviation of the earth image, $\triangle i$, is given by :

$$\triangle i \fallingdotseq -\triangle W = \sqrt{(1087^2-(1087/\sqrt{2}+\triangle W)^2)} - \sqrt{(1087^2-(1087/\sqrt{2})^2)} \quad (3)$$

where

$\triangle i$: with north(−) and south(+)

(e)East-West Center

The actual east-west center of the earth image, $C_{act.}$, is given by

$$C_{act.} = (WE_{act.} + EE_{act.})/2 \quad (4)$$

and the estimated east-west center of the earth image, $C_{est.}$, is given by

$$C_{est.} = (WE_{est.} + EE_{est.})/2 \quad (5)$$

(f)East-West Deviation

The east-west deviation of the earth image, $\triangle j$, is given by :

$$\triangle j = C_{act.} - C_{est.} \quad (6)$$

where

$\triangle j$: with east(+) and west(−)

(g)Orbit and Attitude Prediction Data Correction

VISSR misalignment angle around the Y-axis and Z-axis in the orbit and attitude prediction data are corrected with the deviation of the earth image. The corrective angles of the VISSR misalignment angle around the Y-axis and Z-axis are given by :

$$\triangle Y = -\triangle i \times P \quad (7)$$

$$\triangle Z = +\triangle j \times Q \quad (8)$$

where

$\triangle Y$: corrective angle around Y-axis (rad)

$\triangle Z$: corrective angle around Z-axis (rad)

P : IR channel stepping angle along line (rad)

Q : IR channel sampling angle along pixel (rad)

(h)Coordinate Transformation Table Correction

The line number and pixel number in the Coordinate Transformation Table for the simplified mapping are corrected with the deviation of the earth image. The correct line number is obtained by adding the north-south deviation, $\triangle i$, to the line number and the correct pixel number is obtained by adding the east-west deviation, $\triangle j$, to the pixel number.

3. Sample Programs

Sample programs are presented which are written in FORTRAN(FORTRAN 77), and are applicable for the Stretched-VISSR data that is broadcasted via satellite. After fine tuning, the maximum coordinate transformation error is 140 μradian (infrared image: 1 pixel). The sample program listings are given at the end of this report.

4. Conclusion

The fine tuning technique of the Stretched-VISSR image was newly developed. If this technique is used at a Stretched-VISSR user's computer system, the increase in image mapping error after satellite orbit or attitude control is reduced to within acceptable limits.

ストレッチドVISSR画像位置合わせの精度向上

木川　誠一郎
気象衛星センターシステム管理課

ストレッチドVISSRデータは、観測と同時にリアルタイムでユーザ局に配信されているため歪補正処理の結果を反映できず、衛星の軌道・姿勢制御の直後に画像ずれ（画素と緯経度の対応を決定する座標変換の誤差が大きくなること）が生じることがある。ストレッチドVISSRの画像ずれを防ぐために、ストレッチドVISSR受信局で処理可能な簡易歪補正プログラムを開発した。このプログラムを使用すると赤外1画素程度の精度で画像位置合わせが可能になる。

```
C***********************************************************************
C                                                                     C
C              +=========+    +--------------+ BLOCK LENGTH :        C
C              ! S-VISSR !<----! S-VISSR DATA !  9174 BYTES           C
C              ! NAV.    !    +--------------+ (FIXED LENGTH)        C
C              ! DATA    !    UNIT=10  (DISK)                        C
C              ! CHECK   !          !                                C
C +---------+  ! PROGRAM !          !                                C
C ! LISTING !<----! <SV0000>!          !                                C
C +---------+  +=========+          !                                C
C UNIT=6                            !                                C
C                        +--    +----------------------------+       C
C                        !    1 !  DOCUMENTATION SECTOR DATA / !     C
C                        !      !  IR1, IR2, IR3 DATA         !       C
C                        !      +----------------------------+       C
C                        !    2 !  VIS 1 DATA                 !       C
C                        !      +----------------------------+       C
C                1 SCAN LINE  3 !  VIS 2 DATA                 !       C
C                ==> 5 BLOCKS   +----------------------------+       C
C                        !    4 !  VIS 3 DATA                 !       C
C                        !      +----------------------------+       C
C                        !    5 !  VIS 4 DATA                 !       C
C                        +--    +----------------------------+       C
C                               <----------9174 BYTES---------->     C
C                                                                     C
C                                       1 NOV. 1992  S. KIGAWA        C
C***********************************************************************
      PROGRAM  SV0000
C-----------------------------------------------------------------------
C     FINE TUNING OF STRETCHED-VISSR IMAGE MAPPING
C-----------------------------------------------------------------------
      INTEGER*4  ISMT(25,25,4),JSMT(25,25,4),IX(25)/25*0/
      INTEGER*4  JWEL1/0/,JWEL2/0/,ITC/0/,LAEDG(2),LEEDG(2)
      REAL*4     WEL1(100)/100*0/,WEL2(100)/100*0/,
     .           WEL3(100)/100*0/,WEL4(100)/100*0/
      REAL*4     RINF(8)
      REAL*8     DSCT
      CHARACTER  CSMT(2500)*1,COBAT(3200)*1
      CHARACTER  CBUF(9174)*1,DATID*2,SCTID*2,COND*128,MAPC*64,
     .           TEXTID*4,MAPTBL*100,OBAT*128,MANAM*410,SPARE*1459,
     .           SCTCD1*2,SCTCD2*2,SCTCD3*2,CMAPC*64
      EQUIVALENCE ( CBUF(   1)(1:1), DATID(1:1)  )
      EQUIVALENCE ( CBUF(   3)(1:1), SCTID(1:1)  )
      EQUIVALENCE ( CBUF(   5)(1:1), COND(1:1)   )
      EQUIVALENCE ( CBUF( 131)(1:1), MAPC(1:1)   )
      EQUIVALENCE ( CBUF( 195)(1:1), TEXTID(1:1) )
      EQUIVALENCE ( CBUF( 199)(1:1), MAPTBL(1:1) )
      EQUIVALENCE ( CBUF( 299)(1:1), OBAT(1:1)   )
      EQUIVALENCE ( CBUF( 427)(1:1), MANAM(1:1)  )
      EQUIVALENCE ( CBUF( 837)(1:1), SPARE(1:1)  )
      EQUIVALENCE ( CBUF(2296)(1:1), SCTCD1(1:1) )
      EQUIVALENCE ( CBUF(4589)(1:1), SCTCD2(1:1) )
      EQUIVALENCE ( CBUF(6882)(1:1), SCTCD3(1:1) )
C                                      *OPEN FILE
      OPEN(UNIT=10,ACCESS='DIRECT',RECL=9174,IOSTAT=IOS)
      IF( IOS.NE.0 )                                      GO TO  9000
C                                      *GET MAPPING DATA
      DO 1000 IBLK=1,2500*5,5
C                                      +READ S-VISSR DATA
        READ(UNIT=10,REC=IBLK,FMT='(91(100A1),74A1)',IOSTAT=IOS)  CBUF
        IF( IOS.NE.0 )                                    GO TO  8000
C                                      +DOCUMENTAION SECTOR ?
        IF( ICHAR(SCTID(1:1)).NE.0 .OR. ICHAR(SCTID(2:2)).NE.0 )
     .                                                    GO TO  1000
C                                      +SET TEXT ID
        ITLN1 = ICHAR( TEXTID(2:2) )
C                                      +ALREADY SET ?
        IF( IX(ITLN1+1).NE.0 )                            GO TO  1000
C                                      +SET SIMPLIFIED MAPPING DATA
        CMAPC(1:64) = MAPC(1:64)
        DO 1100 I1=1,100
          CSMT(ITLN1*100+I1)(1:1) = MAPTBL(I1:I1)
 1100   CONTINUE
C                                      +SET ORBIT/ATTITUDE DATA
        DO 1200 I2=1,128
          COBAT(ITLN1*128+I2)(1:1) = OBAT(I2:I2)
 1200   CONTINUE
C                                      +SET TEXT ID FLAG
        IX(ITLN1+1) = 1
C                                      +ALL DATA ?
        KTLN = IX( 1)+IX( 2)+IX( 3)+IX( 4)+IX( 5)+IX( 6)+IX( 7)+IX( 8)
     .         +IX( 9)+IX(10)+IX(11)+IX(12)+IX(13)+IX(14)+IX(15)+IX(16)
     .         +IX(17)+IX(18)+IX(19)+IX(20)+IX(21)+IX(22)+IX(23)+IX(24)
     .         +IX(25)
        IF( KTLN.EQ.25 )                                  GO TO  2000
 1000 CONTINUE
C
 2000 CONTINUE
C                                      *GET SIMPLIFIED MAPPING TABLE
      CALL  SV0200( CSMT, ISMT )
C                                      *GET ORBIT/ATTITUDE TABLE
      CALL  SV0300( COBAT, JSMT )
C                                      *GET SSP LINE/PIXEL
      CALL  SV0110( 4, CMAPC(17:20), ISSPA )
      CALL  SV0110( 4, CMAPC(21:24), ISSPO )
      RSSPA = FLOAT(ISSPA)*1.E-3
      RSSPO = FLOAT(ISSPO)*1.E-3
 2100 CONTINUE
      CALL  MGIVSR(2,RSSPP,RSSPL,RSSPO,RSSPA,0.0,RINF,DSCT,JR)
      ISSPL = NINT(RSSPL)
      ISSPP = NINT(RSSPP)
C                                      *GET EARTH-EDGE
      DO 3000 IBLK=1,2500*5,5
C                                      +READ S-VISSR DATA
        READ(UNIT=10,REC=IBLK,FMT='(91(100A1),74A1)',IOSTAT=IOS)  CBUF
        IF( IOS.NE.0 )                                    GO TO  8000
C                                      +IR1 SECTOR ?
        IF( ICHAR(SCTID(1:1)).NE.0 .OR. ICHAR(SCTID(2:2)).NE.0 )
     .                                                    GO TO  3000
C                                      +GET SCAN COUNT
        L0   = ICHAR(COND( 9: 9))
        L3   = MOD(L0,16)*100+L0/16*1000
        L0   = ICHAR(COND(10:10))
        L1   = MOD(L0,16)+L0/16*10
        ILIN = L3+L1
C                                      +JUDGE SCAN COUNT
        ILL1 = ILIN-(ISSPL-1087/SQRT(2.0))
        ILL2 = ILIN-(ISSPL)
        IF( ILL2.GT.0 )                                   GO TO  4000
                                IFLG = 0
        IF( IABS(ILL1).LE. 5 )  IFLG = 1
        IF( IABS(ILL2).LE.10 )  IFLG = 2
        IF( IFLG.EQ.0 )                                   GO TO  3000
C                                      +GET ACTUAL EARTH-EDGE
        CALL  SV0110( 2, COND(11:12), LAEDG(1) )
        CALL  SV0110( 2, COND(13:14), LAEDG(2) )
C                                      +GET ESTIMATED EARTH-EDGE
        DO 3010 I1=1,2
          IP =  128+LAEDG(I1)
          IM = -128+LAEDG(I1)
          CALL  MGIVSR(-2,FLOAT(IP),FLOAT(ILIN),
     .                 RLON,RLAT,0.0,RINF,DSCT,JP)
          CALL  MGIVSR(-2,FLOAT(IM),FLOAT(ILIN),
     .                 RLON,RLAT,0.0,RINF,DSCT,JM)
          DO 3020 I2=1,8
            IO = (IP+IM)/2
            CALL  MGIVSR(-2,FLOAT(IO),FLOAT(ILIN),
     .                   RLON,RLAT,0.0,RINF,DSCT,JO)
            IF( JP.NE.JO )  THEN
              IM = IO
              JM = JO
            ELSEIF( JM.NE.JO )  THEN
              IP = IO
              JP = JO
            ELSE
            ENDIF
 3020     CONTINUE
          IF( I1.EQ.1 )  LEEDG(1) = IP
          IF( I1.EQ.2 )  LEEDG(2) = IM
 3010   CONTINUE
C                                      *GET EARTH WIDTH
        IF( IFLG.EQ.1 )  THEN
          JWEL1        = JWEL1 + 1
          WEL1(JWEL1)  = (LAEDG(2)-LAEDG(1))-(LEEDG(2)-LEEDG(1))
          WEL3(JWEL1)  = FLOAT(LAEDG(2)+LAEDG(1))/2.0
     .                   -FLOAT(LEEDG(2)+LEEDG(1))/2.0
        ENDIF
        IF( IFLG.EQ.2 )  THEN
          JWEL2        = JWEL2 + 1
          WEL2(JWEL2)  = (LAEDG(2)-LAEDG(1))-(LEEDG(2)-LEEDG(1))
          WEL4(JWEL2)  = FLOAT(LAEDG(2)+LAEDG(1))/2.0
     .                   -FLOAT(LEEDG(2)+LEEDG(1))/2.0
        ENDIF
 3000 CONTINUE
 4000 CONTINUE
C                                      *GET EARTH IMAGE DEVIATION
      IF(JWEL1.EQ.0 .OR. JWEL2.EQ.0)                      GO TO  8000
      RME3 = 0.0
      RME2 = 0.0
      DO 5000 I2=1,JWEL2
        RME2 = RME2 + WEL2(I2)
        RME3 = RME3 + WEL4(I2)
```

```
 5000 CONTINUE
      RME2 = RME2/FLOAT(JWEL2)
      RME1 = 0.0
      DO 5100 I1=1,JWEL1
        RME1 = RME1 + WEL1(I1)-RME2
        RME3 = RME3 + WEL3(I1)
 5100 CONTINUE
      RME1 = RME1/FLOAT(JWEL1)
      RME1 = RME1/2.0-0.769
      RME1 = SQRT(1087.*1087.-(1087./SQRT(2.0)+RME1)**2)
     .       -SQRT(1087.*1087.-(1087./SQRT(2.0)      )**2)
      RME3 = RME3/FLOAT(JWEL2+JWEL1)
      WRITE(6,FMT='(A16,F6.2)')  ' LINE  ERROR ==>',RME1
      WRITE(6,FMT='(A16,F6.2)')  ' PIXEL ERROR ==>',RME3
C                                         *CORRECT MAPPING DATA
      IF( ITC.EQ.0 .AND.
     .    ABS(RME1).LE. 3.0 .AND. ABS(RME3).LE. 3.0 )       GO TO  8000
      IF( ABS(RME1).GT.50.0 .OR.  ABS(RME3).GT.50.0 )       GO TO  8000
      CALL  SV0400( JSMT, RME1, RME3 )
C                                         *DO FINE CORRECTION
      IF( ABS(RME1).GT. 3.0 .OR. ABS(RME3).GT. 3.0 )  THEN
        JWEL1 = 0
        JWEL2 = 0
        ITC   = ITC + 1
                                                            GO TO  2100
      ENDIF
C                                         *CLOSE FILE
 8000 CONTINUE
      CLOSE(UNIT=10)
 9000 CONTINUE
      STOP
      END
      SUBROUTINE  SV0100( IWORD, IPOS, C, R4DAT, R8DAT )
C---- -------------------------------------------------------------
C     TYPE CONVERT ROUTINE ( R-TYPE )
C---- -------------------------------------------------------------
      INTEGER*4  IWORD,IPOS,IDATA1
      CHARACTER  C(*)*1
      REAL*4     R4DAT
      REAL*8     R8DAT
      R4DAT = 0.0
      R8DAT = 0.D0
      IF( IWORD.EQ.4 )  THEN
        IDATA1 = ICHAR( C(1)(1:1) )/128
        R8DAT  = DFLOAT( MOD(ICHAR(C(1)(1:1)),128) )*2.D0**(8*3)+
     .           DFLOAT(      ICHAR(C(2)(1:1))      )*2.D0**(8*2)+
     .           DFLOAT(      ICHAR(C(3)(1:1))      )*2.D0**(8*1)+
     .           DFLOAT(      ICHAR(C(4)(1:1))      )
        R8DAT  = R8DAT/10.D0**IPOS
        IF( IDATA1.EQ.1 )  R8DAT = -R8DAT
        R4DAT  = SNGL( R8DAT )
      ELSEIF( IWORD.EQ.6 )  THEN
        IDATA1 = ICHAR( C(1)(1:1) )/128
        R8DAT  = DFLOAT( MOD(ICHAR(C(1)(1:1)),128) )*2.D0**(8*5)+
     .           DFLOAT(      ICHAR(C(2)(1:1))      )*2.D0**(8*4)+
     .           DFLOAT(      ICHAR(C(3)(1:1))      )*2.D0**(8*3)+
     .           DFLOAT(      ICHAR(C(4)(1:1))      )*2.D0**(8*2)+
     .           DFLOAT(      ICHAR(C(5)(1:1))      )*2.D0**(8*1)+
     .           DFLOAT(      ICHAR(C(6)(1:1))      )
        R8DAT  = R8DAT/10.D0**IPOS
        IF( IDATA1.EQ.1 )  R8DAT = -R8DAT
        R4DAT  = SNGL( R8DAT )
      ENDIF
      RETURN
      END
      SUBROUTINE  SV0110( IWORD, C, I4DAT )
C---- -------------------------------------------------------------
C     TYPE CONVERT ROUTINE ( I-TYPE )
C---- -------------------------------------------------------------
      INTEGER*4  IWORD,I4DAT
      CHARACTER  C(*)*1
      I4DAT = 0
      IF( IWORD.EQ.2 )  THEN
        I4DAT  =     ICHAR( C(1)(1:1) )*2**(8*1)+
     .               ICHAR( C(2)(1:1) )
      ELSEIF( IWORD.EQ.4 )  THEN
        I4DAT  =     ICHAR( C(1)(1:1) )*2**(8*3)+
     .               ICHAR( C(2)(1:1) )*2**(8*2)+
     .               ICHAR( C(3)(1:1) )*2**(8*1)+
     .               ICHAR( C(4)(1:1) )
      ENDIF
      RETURN
      END
      SUBROUTINE  SV0200( CSMT, ISMT )
C-----------------------------------------------------------------
C     SIMPLIFIED MAPPING DATA PROCESSING ROUTINE                  C
C-----------------------------------------------------------------
      CHARACTER  CSMT(2500)*1
      INTEGER*4  ISMT(25,25,4)
      DO 2100 IL1=1,25
        DO 2200 IL2=1,25
          ILAT   = 60-(IL1-1)*5
          ILON   = 80+(IL2-1)*5
          IL3    = (IL1-1)*100+(IL2-1)*4+1
          ILINE1 = ICHAR(CSMT(IL3  )(1:1))*256+ICHAR(CSMT(IL3+1)(1:1))
          IPIXE1 = ICHAR(CSMT(IL3+2)(1:1))*256+ICHAR(CSMT(IL3+3)(1:1))
          ISMT(IL2,IL1,1) = ILAT
          ISMT(IL2,IL1,2) = ILON
          ISMT(IL2,IL1,3) = ILINE1
          ISMT(IL2,IL1,4) = IPIXE1
 2200   CONTINUE
 2100 CONTINUE
      RETURN
      END
      SUBROUTINE  SV0300( COBAT ,JSMT )
C-----------------------------------------------------------------
C     ORBIT AND ATTITUDE DATA PROCESSING ROUTINE                  C
C-----------------------------------------------------------------
      COMMON /MMAP1/MAP
      INTEGER*4  MAP(672,4)
      CHARACTER  COBAT*3200
      INTEGER*4  JSMT(25,25,4)
      REAL*4     R4DMY,RESLIN(4),RESELM(4),RLIC(4),RELMFC(4),SENSSU(4),
     .           VMIS(3),ELMIS(3,3),RLINE(4),RELMNT(4),RINF(8)
      REAL*8     R8DMY,DSPIN,DTIMS,ATIT(10,33),ORBT1(35,8),DSCT
C
      EQUIVALENCE (MAP( 5,1),DTIMS),     (MAP( 7,1),RESLIN(1))
      EQUIVALENCE (MAP(11,1),RESELM(1)), (MAP(15,1),RLIC(1))
      EQUIVALENCE (MAP(19,1),RELMFC(1)), (MAP(27,1),SENSSU(1))
      EQUIVALENCE (MAP(31,1),RLINE(1)),  (MAP(35,1),RELMNT(1))
      EQUIVALENCE (MAP(39,1),VMIS(1)),   (MAP(42,1),ELMIS)
      EQUIVALENCE (MAP(131,1),DSPIN)
      EQUIVALENCE (MAP(13,3),ORBT1(1,1)),(MAP(13,2),ATIT(1,1))
C
      DO 1000 I=1,4
        DO 1100 J=1,672
          MAP(J,I) = 0
 1100   CONTINUE
 1000 CONTINUE
C
      CALL  SV0100( 6, 8, COBAT(  1:  6), R4DMY     ,  DTIMS )
      CALL  SV0100( 4, 8, COBAT(  7: 10), RESLIN(1),  R8DMY )
      CALL  SV0100( 4, 8, COBAT( 11: 14), RESLIN(2),  R8DMY )
      CALL  SV0100( 4,10, COBAT( 15: 18), RESELM(1),  R8DMY )
      CALL  SV0100( 4,10, COBAT( 19: 22), RESELM(2),  R8DMY )
      CALL  SV0100( 4, 4, COBAT( 23: 26), RLIC(1)  ,  R8DMY )
      CALL  SV0100( 4, 4, COBAT( 27: 30), RLIC(2)  ,  R8DMY )
      CALL  SV0100( 4, 4, COBAT( 31: 34), RELMFC(1),  R8DMY )
      CALL  SV0100( 4, 4, COBAT( 35: 38), RELMFC(2),  R8DMY )
      CALL  SV0100( 4, 0, COBAT( 39: 42), SENSSU(1),  R8DMY )
      CALL  SV0100( 4, 0, COBAT( 43: 46), SENSSU(2),  R8DMY )
      CALL  SV0100( 4, 0, COBAT( 47: 50), RLINE(1) ,  R8DMY )
      CALL  SV0100( 4, 0, COBAT( 51: 54), RLINE(2) ,  R8DMY )
      CALL  SV0100( 4, 0, COBAT( 55: 58), RELMNT(1),  R8DMY )
      CALL  SV0100( 4, 0, COBAT( 59: 62), RELMNT(2),  R8DMY )
      CALL  SV0100( 4,10, COBAT( 63: 66), VMIS(1)  ,  R8DMY )
      CALL  SV0100( 4,10, COBAT( 67: 70), VMIS(2)  ,  R8DMY )
      CALL  SV0100( 4,10, COBAT( 71: 74), VMIS(3)  ,  R8DMY )
      CALL  SV0100( 4, 7, COBAT( 75: 78), ELMIS(1,1), R8DMY )
      CALL  SV0100( 4,10, COBAT( 79: 82), ELMIS(2,1), R8DMY )
      CALL  SV0100( 4,10, COBAT( 83: 86), ELMIS(3,1), R8DMY )
      CALL  SV0100( 4,10, COBAT( 87: 90), ELMIS(1,2), R8DMY )
      CALL  SV0100( 4, 7, COBAT( 91: 94), ELMIS(2,2), R8DMY )
      CALL  SV0100( 4,10, COBAT( 95: 98), ELMIS(3,2), R8DMY )
      CALL  SV0100( 4,10, COBAT( 99:102), ELMIS(1,3), R8DMY )
      CALL  SV0100( 4,10, COBAT(103:106), ELMIS(2,3), R8DMY )
      CALL  SV0100( 4, 7, COBAT(107:110), ELMIS(3,3), R8DMY )
      CALL  SV0100( 6, 8, COBAT(241:246), R4DMY     , DSPIN )
C
      DO 2000 I=1,10
        J = (I-1)*64+257-1
        CALL  SV0100(6, 8,COBAT( 1+J: 6+J),R4DMY,ATIT(1,I))
        CALL  SV0100(6, 8,COBAT(13+J:18+J),R4DMY,ATIT(3,I))
        CALL  SV0100(6,11,COBAT(19+J:24+J),R4DMY,ATIT(4,I))
        CALL  SV0100(6, 8,COBAT(25+J:30+J),R4DMY,ATIT(5,I))
        CALL  SV0100(6, 8,COBAT(31+J:36+J),R4DMY,ATIT(6,I))
 2000 CONTINUE
C
      DO 3000 I=1,8
        J = (I-1)*256+897-1
        CALL  SV0100(6, 8,COBAT(  1+J:  6+J),R4DMY,ORBT1( 1,I))
        CALL  SV0100(6, 6,COBAT( 49+J: 54+J),R4DMY,ORBT1( 9,I))
```

```
      CALL  SV0100(6, 6,COBAT( 55+J: 60+J),R4DMY,ORBT1(10,I))
      CALL  SV0100(6, 6,COBAT( 61+J: 66+J),R4DMY,ORBT1(11,I))
      CALL  SV0100(6, 8,COBAT( 85+J: 90+J),R4DMY,ORBT1(15,I))
      CALL  SV0100(6, 8,COBAT(103+J:108+J),R4DMY,ORBT1(18,I))
      CALL  SV0100(6, 8,COBAT(109+J:114+J),R4DMY,ORBT1(19,I))
      CALL  SV0100(6,12,COBAT(129+J:134+J),R4DMY,ORBT1(20,I))
      CALL  SV0100(6,14,COBAT(135+J:140+J),R4DMY,ORBT1(21,I))
      CALL  SV0100(6,14,COBAT(141+J:146+J),R4DMY,ORBT1(22,I))
      CALL  SV0100(6,14,COBAT(147+J:152+J),R4DMY,ORBT1(23,I))
      CALL  SV0100(6,12,COBAT(153+J:158+J),R4DMY,ORBT1(24,I))
      CALL  SV0100(6,16,COBAT(159+J:164+J),R4DMY,ORBT1(25,I))
      CALL  SV0100(6,12,COBAT(165+J:170+J),R4DMY,ORBT1(26,I))
      CALL  SV0100(6,16,COBAT(171+J:176+J),R4DMY,ORBT1(27,I))
      CALL  SV0100(6,12,COBAT(177+J:182+J),R4DMY,ORBT1(28,I))
 3000 CONTINUE
C
      DO 4100 IL1=1,25
       DO 4200 IL2=1,25
        RLAT   = FLOAT( 60-(IL1-1)*5 )
        RLON   = FLOAT( 80+(IL2-1)*5 )
        CALL  MGIVSR(2,RPIX,RLIN,RLON,RLAT,0.0,RINF,DSCT,IRTN)
        JSMT(IL2,IL1,1) = NINT( RLAT )
        JSMT(IL2,IL1,2) = NINT( RLON )
        JSMT(IL2,IL1,3) = NINT( RLIN )
        JSMT(IL2,IL1,4) = NINT( RPIX )
 4200  CONTINUE
 4100 CONTINUE
C
      RETURN
      END
      SUBROUTINE  SV0400( ISMT , RDL, RDP )
C-----------------------------------------------------------------------
C     MAPPING TABLE CORRECTION ROUTINE                                 C
C-----------------------------------------------------------------------
      COMMON /MMAP1/MAP
      INTEGER*4  MAP(672,4)
      INTEGER*4  ISMT(25,25,4)
      REAL*4     RESLIN(4),RESELM(4),VMIS(3),ELMIS(3,3),RINF(8)
      REAL*8     R8DMY,DSPIN,DTIMS,ATIT(10,33),ORBT1(35,8),DSCT
C
      EQUIVALENCE (MAP( 7,1),RESLIN(1))
      EQUIVALENCE (MAP(11,1),RESELM(1))
      EQUIVALENCE (MAP(39,1),VMIS(1)),  (MAP(42,1),ELMIS)
C                                     *CORRECT ORBIT/ATTITUDE DATA
      VMIS(2)     = VMIS(2)-(RDL)*RESLIN(2)
      VMIS(3)     = VMIS(3)+(RDP)*RESELM(2)
       WRITE(6,FMT='(A16,F12.8)') ' CORRECT Y-MIS :',VMIS(2)
       WRITE(6,FMT='(A16,F12.8)') ' CORRECT Z-MIS :',VMIS(3)
      CA          = COS(VMIS(1))
      CB          = COS(VMIS(2))
      CC          = COS(VMIS(3))
      SA          = SIN(VMIS(1))
      SB          = SIN(VMIS(2))
      SC          = SIN(VMIS(3))
      ELMIS(1,1)  =  CC*CB
      ELMIS(2,1)  =  -SC*CB
      ELMIS(3,1)  =  SB
      ELMIS(1,2)  =   CC*SB*SA+SC*CA
      ELMIS(2,2)  =  -SC*SB*SA+CC*CA
      ELMIS(3,2)  =  -CB*SA
      ELMIS(1,3)  =   CC*SB*CA+SC*SA
      ELMIS(2,3)  =  -SC*SB*CA+CC*SA
      ELMIS(3,3)  =   CB*CA
C                                     *CORRECT MAPPING TABLE
      DO 1000 IL1=1,25
        DO 1100 IL2=1,25
         ISMT(IL2,IL1,3) = ISMT(IL2,IL1,3)+NINT(RDL)
         ISMT(IL2,IL1,4) = ISMT(IL2,IL1,4)+NINT(RDP)
 1100   CONTINUE
 1000 CONTINUE
C
      RETURN
      END
      SUBROUTINE  MGIVSR( IMODE,RPIX,RLIN,RLON,RLAT,RHGT,
     .                    RINF,DSCT,IRTN)
C
C***********************************************************************
C***********************************************************************
C***********************************************************************
C***********************************************************************
C
C  THIS PROGRAM CONVERTS GEOGRAPHICAL CO-ORDINATES (LATITUDE,LONGITUDE,
C  HEIGHT) TO VISSR IMAGE CO-ORDINATES (LINE,PIXEL) AND VICE VERSA.
C
C  THIS PROGRAM IS PROVIDED BY THE METEOROLOGICAL SATELLITE CENTER OF
C  THE JAPAN METEOROLOGICAL AGENCY TO USERS OF GMS DATA.
```

```
C
C                                                    MSC TECH. NOTE NO.23
C                                                            JMA/MSC 1991
C
C***********************************************************************
C***********************************************************************
C***********************************************************************
C***********************************************************************
C
C***********************************************************************
C             I/O  TYPE
C   IMODE      I   I*4   CONVERSION MODE & IMAGE KIND
C                          IMAGE KIND
C                                GMS-4 GMS-5
C                          1,-1  VIS   VIS
C                          2,-2  IR    IR1
C                          3,-3  --    IR2
C                          4,-4  --    WV
C                          CONVERSION MODE
C                            1 TO 4    (LAT,LON,HGT)=>(LINE,PIXEL)
C                           -1 TO -4   (LAT,LON    )<=(LINE,PIXEL)
C   RPIX      I/O  R*4   PIXEL OF POINT
C   RLIN      I/O  R*4   LINE OF POINT
C   RLON      I/O  R*4   LONGITUDE OF POINT (DEGREES,EAST:+,WEST:-)
C   RLAT      I/O  R*4   LATITUDE OF POINT (DEGREES,NORTH:+,SOUTH:-)
C   RHGT       I   R*4   HEIGHT OF POINT (METER)
C   RINF(8)    O   R*4   (1) SATELLITE ZENITH DISTANCE (DEGREES)
C                        (2) SATELLITE AZIMUTH ANGLE (DEGREES)
C                        (3) SUN ZENITH DISTANCE (DEGREES)
C                        (4) SUN AZIMUTH ANGLE (DEGREES)
C                        (5) SATELLTE-SUN DIPARTURE ANGLE (DEGREES)
C                        (6) SATELLTE DISTANCE (METER)
C                        (7) SUN DISTANCE (KIRO-METER)
C                        (8) SUN GRINT ANGLE (DEGREES)
C   DSCT       O   R*8   SCAN TIME (MJD)
C   IRTN       O   I*4   RETURN CODE (0=O.K.)
C
C!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
C
C   COMMON /MMAP1/MAP(672,4)
C
C     1. COORDINATE TRANSFORMATION PARAMETERS SEGMENT
C                                                 MAP(1,1)-MAP(672,1)
C     2. ATTITUDE PREDICTION DATA SEGMENT         MAP(1,2)-MAP(672,2)
C     3. ORBIT PREDICTION DATA 1 SEGMENT          MAP(1,3)-MAP(672,3)
C     4. ORBIT PREDICTION DATA 2 SEGMENT          MAP(1,4)-MAP(672,4)
C***********************************************************************
C
C!!!!!!!!!!!!!!! DEFINITION !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      COMMON /MMAP1/MAP
C
      REAL*4     RPIX,RLIN,RLON,RLAT,RHGT,RINF(8)
      INTEGER*4  MAP(672,4),IRTN
C
      REAL*4     EPS,RI0,RI,RJ,RSTEP,RSAMP,RFCL,RFCP,SENS,RFTL,RFTP
      REAL*4     RESLIN(4),RESELM(4),RLIC(4),RELMFC(4),SENSSU(4),
     .           VMIS(3),ELMIS(3,3),RLINE(4),RELMNT(4)
      REAL*8     BC,BETA,BS,CDR,CRD,DD,DDA,DDB,DDC,DEF,DK,DK1,DK2,
     .           DLAT,DLON,DPAI,DSPIN,DTIMS,EA,EE,EF,EN,HPAI,PC,PI,PS,
     .           QC,QS,RTIM,TF,TL,TP,
     .           SAT(3),SL(3),SLV(3),SP(3),SS(3),STN1(3),STN2(3),
     .           SX(3),SY(3),SW1(3),SW2(3),SW3(3)
      REAL*8     DSCT,DSATZ,DSATA,DSUNZ,DSUNA,DSSDA,DSATD,SUNM,SDIS,
     .           DLATN,DLONN,STN3(3),DSUNG
C
C!!!!!!!!!!!!!!! EQUIVALENCE !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      EQUIVALENCE (MAP( 5,1),DTIMS),     (MAP( 7,1),RESLIN(1))
      EQUIVALENCE (MAP(11,1),RESELM(1)), (MAP(15,1),RLIC(1))
      EQUIVALENCE (MAP(19,1),RELMFC(1)), (MAP(27,1),SENSSU(1))
      EQUIVALENCE (MAP(31,1),RLINE(1)),  (MAP(35,1),RELMNT(1))
      EQUIVALENCE (MAP(39,1),VMIS(1)),   (MAP(42,1),ELMIS)
      EQUIVALENCE (MAP(131,1),DSPIN)
C
C***********************************************************************
C
      PI    =  3.141592653D0
      CDR   =  PI/180.D0
      CRD   =  180.D0/PI
      HPAI  =  PI/2.D0
      DPAI  =  PI*2.D0
      EA    =  6378136.D0
      EF    =  1.D0/298.257D0
      EPS   =  1.0
C!!!!!!!!!!!!!!! PARAMETER CHECK !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      IRTN  =  0
      IF(ABS(IMODE).GT.4)  IRTN=1
```

```
      IF(ABS(RLAT).GT.90. .AND. IMODE.GT.0)  IRTN=2
      IF(IRTN.NE.0)  RETURN
C!!!!!!!!!!!!!!!!!! VISSR FRAME INFORMATION SET !!!!!!!!!!!!!!!!!!!!!!!!!!
      LMODE   = ABS(IMODE)
      RSTEP   = RESLIN(LMODE)
      RSAMP   = RESELM(LMODE)
      RFCL    = RLIC(LMODE)
      RFCP    = RELMFC(LMODE)
      SENS    = SENSSU(LMODE)
      RFTL    = RLINE(LMODE)+0.5
      RFTP    = RELMNT(LMODE)+0.5
C!!!!!!!!!!!!!!!!!! TRANSFORMATION (GEOGRAPHICAL=>VISSR) !!!!!!!!!!!!!!!!
      IF( IMODE.GT.0 .AND. IMODE.LT.5 )  THEN
        DLAT    = DBLE(RLAT)*CDR
        DLON    = DBLE(RLON)*CDR
        EE      = 2.D0*EF-EF*EF
        EN      = EA/DSQRT(1.D0-EE*DSIN(DLAT)*DSIN(DLAT))
        STN1(1) = (EN+DBLE(RHGT))*DCOS(DLAT)*DCOS(DLON)
        STN1(2) = (EN+DBLE(RHGT))*DCOS(DLAT)*DSIN(DLON)
        STN1(3) = (EN*(1.D0-EE)+DBLE(RHGT))*DSIN(DLAT)
C
        RIO     = RFCL-ATAN(SIN(SNGL(DLAT))/(6.610689-COS(SNGL(DLAT))))
     .            /RSTEP
        RTIM    = DTIMS+DBLE(RIO/SENS/1440.)/DSPIN
C
  100   CONTINUE
        CALL  MGI100(RTIM,CDR,SAT,SP,SS,BETA)
C-----------------------------------------------------------------------
        CALL  MGI220(SP,SS,SW1)
        CALL  MGI220(SW1,SP,SW2)
        BC      = DCOS(BETA)
        BS      = DSIN(BETA)
        SW3(1)  = SW1(1)*BS+SW2(1)*BC
        SW3(2)  = SW1(2)*BS+SW2(2)*BC
        SW3(3)  = SW1(3)*BS+SW2(3)*BC
        CALL  MGI200(SW3,SX)
        CALL  MGI220(SP,SX,SY)
        SLV(1)  = STN1(1)-SAT(1)
        SLV(2)  = STN1(2)-SAT(2)
        SLV(3)  = STN1(3)-SAT(3)
        CALL  MGI200(SLV,SL)
        CALL  MGI210(SP,SL,SW2)
        CALL  MGI210(SY,SW2,SW3)
        CALL  MGI230(SY,SW2,TP)
        TF      = SP(1)*SW3(1)+SP(2)*SW3(2)+SP(3)*SW3(3)
        IF(TF.LT.0.D0)  TP=-TP
        CALL  MGI230(SP,SL,TL)
C
        RI      = SNGL(HPAI-TL)/RSTEP+RFCL-VMIS(2)/RSTEP
        RJ      = SNGL(TP)/RSAMP+RFCP
     .            +VMIS(3)/RSAMP-SNGL(HPAI-TL)*TAN(VMIS(1))/RSAMP
C
        IF(ABS(RI-RIO).GE.EPS)  THEN
          RTIM  = DBLE(AINT((RI-1.)/SENS)+RJ*RSAMP/SNGL(DPAI))/
     .            (DSPIN*1440.D0)+DTIMS
          RIO   = RI
          GO TO  100
        ENDIF
        RLIN    = RI
        RPIX    = RJ
        DSCT    = RTIM
        IF(RLIN.LT.0 .OR. RLIN.GT.RFTL)  IRTN=4
        IF(RPIX.LT.0 .OR. RPIX.GT.RFTP)  IRTN=5
C
C!!!!!!!!!!!!!!!!!! TRANSFORMATION (VISSR=>GEOGRAPHICAL) !!!!!!!!!!!!!!!!
      ELSEIF(IMODE.LT.0 .AND. IMODE.GT.-5)  THEN
C
        RTIM    = DBLE(AINT((RLIN-1.)/SENS)+RPIX*RSAMP/SNGL(DPAI))/
     .            (DSPIN*1440.D0)+DTIMS
        CALL  MGI100(RTIM,CDR,SAT,SP,SS,BETA)
        CALL  MGI220(SP,SS,SW1)
        CALL  MGI220(SW1,SP,SW2)
        BC      = DCOS(BETA)
        BS      = DSIN(BETA)
        SW3(1)  = SW1(1)*BS+SW2(1)*BC
        SW3(2)  = SW1(2)*BS+SW2(2)*BC
        SW3(3)  = SW1(3)*BS+SW2(3)*BC
        CALL  MGI200(SW3,SX)
        CALL  MGI220(SP,SX,SY)
        PC      = DCOS(DBLE(RSTEP*(RLIN-RFCL)))
        PS      = DSIN(DBLE(RSTEP*(RLIN-RFCL)))
        QC      = DCOS(DBLE(RSAMP*(RPIX-RFCP)))
        QS      = DSIN(DBLE(RSAMP*(RPIX-RFCP)))
        SW1(1)  = DBLE(ELMIS(1,1))*PC+DBLE(ELMIS(1,3))*PS
        SW1(2)  = DBLE(ELMIS(2,1))*PC+DBLE(ELMIS(2,3))*PS
        SW1(3)  = DBLE(ELMIS(3,1))*PC+DBLE(ELMIS(3,3))*PS
        SW2(1)  = QC*SW1(1)-QS*SW1(2)
        SW2(2)  = QS*SW1(1)+QC*SW1(2)
        SW2(3)  = SW1(3)
        SW3(1)  = SX(1)*SW2(1)+SY(1)*SW2(2)+SP(1)*SW2(3)
        SW3(2)  = SX(2)*SW2(1)+SY(2)*SW2(2)+SP(2)*SW2(3)
        SW3(3)  = SX(3)*SW2(1)+SY(3)*SW2(2)+SP(3)*SW2(3)
        CALL  MGI200(SW3,SL)
        DEF     = (1.D0-EF)*(1.D0-EF)
        DDA     = DEF*(SL(1)*SL(1)+SL(2)*SL(2))+SL(3)*SL(3)
        DDB     = DEF*(SAT(1)*SL(1)+SAT(2)*SL(2))+SAT(3)*SL(3)
        DDC     = DEF*(SAT(1)*SAT(1)+SAT(2)*SAT(2)-EA*EA)+SAT(3)*SAT(3)
        DD      = DDB*DDB-DDA*DDC
        IF(DD.GE.0.D0 .AND. DDA.NE.0.D0)  THEN
          DK1     = (-DDB+DSQRT(DD))/DDA
          DK2     = (-DDB-DSQRT(DD))/DDA
        ELSE
          IRTN    = 6
          GO TO  9000
        ENDIF
        IF(DABS(DK1).LE.DABS(DK2))  THEN
          DK    = DK1
        ELSE
          DK    = DK2
        ENDIF
        STN1(1) = SAT(1)+DK*SL(1)
        STN1(2) = SAT(2)+DK*SL(2)
        STN1(3) = SAT(3)+DK*SL(3)
        DLAT    = DATAN(STN1(3)/(DEF*DSQRT(STN1(1)*STN1(1)+
     .            STN1(2)*STN1(2))))
        IF(STN1(1).NE.0.D0)  THEN
          DLON  = DATAN(STN1(2)/STN1(1))
          IF(STN1(1).LT.0.D0 .AND. STN1(2).GE.0.D0)  DLON=DLON+PI
          IF(STN1(1).LT.0.D0 .AND. STN1(2).LT.0.D0)  DLON=DLON-PI
        ELSE
          IF(STN1(2).GT.0.D0)  THEN
            DLON=HPAI
          ELSE
            DLON=-HPAI
          ENDIF
        ENDIF
        RLAT    = SNGL(DLAT*CRD)
        RLON    = SNGL(DLON*CRD)
        DSCT    = RTIM
      ENDIF
C
C!!!!!!!!!!!!!!!!!! TRANSFORMATION (ZENITH/AZIMUTH) !!!!!!!!!!!!!!!!!!!!!!
      STN2(1)   = DCOS(DLAT)*DCOS(DLON)
      STN2(2)   = DCOS(DLAT)*DSIN(DLON)
      STN2(3)   = DSIN(DLAT)
      SLV(1)    = SAT(1)-STN1(1)
      SLV(2)    = SAT(2)-STN1(2)
      SLV(3)    = SAT(3)-STN1(3)
      CALL  MGI200(SLV,SL)
C
      CALL  MGI230(STN2,SL,DSATZ)
      IF(DSATZ.GT.HPAI)  IRTN = 7
C
      SUNM     = 315.253D0+0.985600D0*RTIM
      SUNM     = DMOD(SUNM,360.D0)*CDR
      SDIS     = (1.00014D0-0.01672D0*DCOS(SUNM)-0.00014*DCOS(2.D0*
     .           SUNM))*1.49597870D8
C
      IF(DLAT.GE.0.D0) THEN
        DLATN    = HPAI-DLAT
        DLONN = DLON-PI
        IF(DLONN.LE.-PI)  DLONN=DLONN+DPAI
      ELSE
        DLATN    = HPAI+DLAT
        DLONN    = DLON
      ENDIF
      STN3(1) = DCOS(DLATN)*DCOS(DLONN)
      STN3(2) = DCOS(DLATN)*DSIN(DLONN)
      STN3(3) = DSIN(DLATN)
      SW1(1)  = SLV(1)+SS(1)*SDIS*1.D3
      SW1(2)  = SLV(2)+SS(2)*SDIS*1.D3
      SW1(3)  = SLV(3)+SS(3)*SDIS*1.D3
      CALL  MGI200(SW1,SW2)
      CALL  MGI230(STN2,SW2,DSUNZ)
      CALL  MGI230(SL,SW2,DSSDA)
      CALL  MGI240(SL,STN2,STN3,DPAI,DSATA)
      CALL  MGI240(SW2,STN2,STN3,DPAI,DSUNA)
      DSATD    = DSQRT(SLV(1)*SLV(1)+SLV(2)*SLV(2)+SLV(3)*SLV(3))
C
C
      CALL  MGI200(STN1,SL)
      CALL  MGI230(SW2,SL,DSUNG)
```

FINE TUNING PROGRAM OF STRETCHED-VISSR IMAGE MAPPING (4/5)　　JAPAN METEOROLOGICAL AGENCY / METEOROLOGICAL SATELLITE CENTER　1992

```
      CALL  MGI220(SL, SW2,SW3)
      CALL  MGI220(SW3,SL, SW1)
      WKCOS=DCOS(DSUNG)
      WKSIN=DSIN(DSUNG)
      SW2(1)=WKCOS*SL(1)-WKSIN*SW1(1)
      SW2(2)=WKCOS*SL(2)-WKSIN*SW1(2)
      SW2(3)=WKCOS*SL(3)-WKSIN*SW1(3)
      CALL  MGI230(SW2,SLV,DSUNG)
C
      RINF(6)  =  SNGL(DSATD)
      RINF(7)  =  SNGL(SDIS)
      RINF(1)  =  SNGL(DSATZ*CRD)
      RINF(2)  =  SNGL(DSATA*CRD)
      RINF(3)  =  SNGL(DSUNZ*CRD)
      RINF(4)  =  SNGL(DSUNA*CRD)
      RINF(5)  =  SNGL(DSSDA*CRD)
      RINF(8)  =  SNGL(DSUNG*CRD)
C!!!!!!!!!!!!!!!! STOP/END !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 9000 CONTINUE
      RETURN
      END
      SUBROUTINE  MGI100(RTIM,CDR,SAT,SP,SS,BETA)
      COMMON /MMAP1/MAP
      REAL*8    ATTALP,ATTDEL,BETA,CDR,DELT,RTIM,SITAGT,SUNALP,SUNDEL,
     .          WKCOS,WKSIN
      REAL*8    ATIT(10,10),ATT1(3),ATT2(3),ATT3(3),NPA(3,3),
     .          ORBT1(35,8),SAT(3),SP(3),SS(3)
      INTEGER*4 MAP(672,4)
C
      EQUIVALENCE (MAP(13,3),ORBT1(1,1))
      EQUIVALENCE (MAP(13,2),ATIT(1,1))
C
      DO 1000 I=1,7
        IF(RTIM.GE.ORBT1(1,I).AND.RTIM.LT.ORBT1(1,I+1))  THEN
          CALL  MGI110
     .         (I,RTIM,CDR,ORBT1,ORBT2,SAT,SITAGT,SUNALP,SUNDEL,NPA)
          GO TO  1200
        ENDIF
 1000 CONTINUE
 1200 CONTINUE
C
      DO 3000 I=1,33-1
        IF(RTIM.GE.ATIT(1,I) .AND. RTIM.LT.ATIT(1,I+1))  THEN
          DELT = (RTIM-ATIT(1,I))/(ATIT(1,I+1)-ATIT(1,I))
          ATTALP = ATIT(3,I)+(ATIT(3,I+1)-ATIT(3,I))*DELT
          ATTDEL = ATIT(4,I)+(ATIT(4,I+1)-ATIT(4,I))*DELT
          BETA   = ATIT(5,I)+(ATIT(5,I+1)-ATIT(5,I))*DELT
          IF( (ATIT(5,I+1)-ATIT(5,I)).GT.0.D0 )
     .      BETA   = ATIT(5,I)+(ATIT(5,I+1)-ATIT(5,I)-360.D0*CDR)*DELT
          GO TO  3001
        ENDIF
 3000 CONTINUE
 3001 CONTINUE
C
      WKCOS    = DCOS(ATTDEL)
      ATT1(1)  = DSIN(ATTDEL)
      ATT1(2)  = WKCOS        *(-DSIN(ATTALP))
      ATT1(3)  = WKCOS        *DCOS(ATTALP)
      ATT2(1)  = NPA(1,1)*ATT1(1)+NPA(1,2)*ATT1(2)+NPA(1,3)*ATT1(3)
      ATT2(2)  = NPA(2,1)*ATT1(1)+NPA(2,2)*ATT1(2)+NPA(2,3)*ATT1(3)
      ATT2(3)  = NPA(3,1)*ATT1(1)+NPA(3,2)*ATT1(2)+NPA(3,3)*ATT1(3)
      WKSIN    = DSIN(SITAGT)
      WKCOS    = DCOS(SITAGT)
      ATT3(1)  = WKCOS*ATT2(1)+WKSIN*ATT2(2)
      ATT3(2)  =-WKSIN*ATT2(1)+WKCOS*ATT2(2)
      ATT3(3)  = ATT2(3)
      CALL  MGI200(ATT3,SP)
C
      WKCOS    = DCOS(SUNDEL)
      SS(1)    = WKCOS        *DCOS(SUNALP)
      SS(2)    = WKCOS        *DSIN(SUNALP)
      SS(3)    = DSIN(SUNDEL)
C
      RETURN
      END
      SUBROUTINE  MGI110
     .          (I,RTIM,CDR,ORBTA,ORBTB,SAT,SITAGT,SUNALP,SUNDEL,NPA)
      REAL*8    CDR,SAT(3),RTIM,ORBTA(35,8),ORBTB(35,8)
      REAL*8    SITAGT,SUNDEL,SUNALP,NPA(3,3),DELT
      INTEGER*4 I
      IF(I.NE.8)  THEN
        DELT=(RTIM-ORBTA(1,I))/(ORBTA(1,I+1)-ORBTA(1,I))
        SAT(1)    = ORBTA( 9,I)+(ORBTA( 9,I+1)-ORBTA( 9,I))*DELT
        SAT(2)    = ORBTA(10,I)+(ORBTA(10,I+1)-ORBTA(10,I))*DELT
        SAT(3)    = ORBTA(11,I)+(ORBTA(11,I+1)-ORBTA(11,I))*DELT
        SITAGT    =(ORBTA(15,I)+(ORBTA(15,I+1)-ORBTA(15,I))*DELT)*CDR
        IF( (ORBTA(15,I+1)-ORBTA(15,I)).LT.0.D0 )
     .    SITAGT    =(ORBTA(15,I)+(ORBTA(15,I+1)-ORBTA(15,I)+360.D0)
     .              *DELT)*CDR
        SUNALP    =(ORBTA(18,I)+(ORBTA(18,I+1)-ORBTA(18,I))*DELT)*CDR
        IF( (ORBTA(18,I+1)-ORBTA(18,I)).GT.0.D0 )
     .    SUNALP    =(ORBTA(18,I)+(ORBTA(18,I+1)-ORBTA(18,I)-360.D0)
     .              *DELT)*CDR
        SUNDEL    =(ORBTA(19,I)+(ORBTA(19,I+1)-ORBTA(19,I))*DELT)*CDR
        NPA(1,1)  = ORBTA(20,I)
        NPA(2,1)  = ORBTA(21,I)
        NPA(3,1)  = ORBTA(22,I)
        NPA(1,2)  = ORBTA(23,I)
        NPA(2,2)  = ORBTA(24,I)
        NPA(3,2)  = ORBTA(25,I)
        NPA(1,3)  = ORBTA(26,I)
        NPA(2,3)  = ORBTA(27,I)
        NPA(3,3)  = ORBTA(28,I)
      ENDIF
      RETURN
      END
      SUBROUTINE  MGI200(VECT,VECTU)
      REAL*8  VECT(3),VECTU(3),RV1,RV2
      RV1=VECT(1)*VECT(1)+VECT(2)*VECT(2)+VECT(3)*VECT(3)
      IF(RV1.EQ.0.D0)  RETURN
      RV2=DSQRT(RV1)
      VECTU(1)=VECT(1)/RV2
      VECTU(2)=VECT(2)/RV2
      VECTU(3)=VECT(3)/RV2
      RETURN
      END
      SUBROUTINE  MGI210(VA,VB,VC)
      REAL*8  VA(3),VB(3),VC(3)
      VC(1)= VA(2)*VB(3)-VA(3)*VB(2)
      VC(2)= VA(3)*VB(1)-VA(1)*VB(3)
      VC(3)= VA(1)*VB(2)-VA(2)*VB(1)
      RETURN
      END
      SUBROUTINE  MGI220(VA,VB,VD)
      REAL*8  VA(3),VB(3),VC(3),VD(3)
      VC(1)= VA(2)*VB(3)-VA(3)*VB(2)
      VC(2)= VA(3)*VB(1)-VA(1)*VB(3)
      VC(3)= VA(1)*VB(2)-VA(2)*VB(1)
      CALL  MGI200(VC,VD)
      RETURN
      END
      SUBROUTINE  MGI230(VA,VB,ASITA)
      REAL*8  VA(3),VB(3),ASITA,AS1,AS2
      AS1= VA(1)*VB(1)+VA(2)*VB(2)+VA(3)*VB(3)
      AS2=(VA(1)*VA(1)+VA(2)*VA(2)+VA(3)*VA(3))*
     .    (VB(1)*VB(1)+VB(2)*VB(2)+VB(3)*VB(3))
      IF(AS2.EQ.0.D0)  RETURN
      ASITA=DACOS(AS1/DSQRT(AS2))
      RETURN
      END
      SUBROUTINE  MGI240(VA,VH,VN,DPAI,AZI)
      REAL*8  VA(3),VH(3),VN(3),VB(3),VC(3),VD(3),DPAI,AZI,DNAI
      CALL  MGI220(VN,VH,VB)
      CALL  MGI220(VA,VH,VC)
      CALL  MGI230(VB,VC,AZI)
      CALL  MGI220(VB,VC,VD)
      DNAI = VD(1)*VH(1)+VD(2)*VH(2)+VD(3)*VH(3)
      IF(DNAI.GT.0.D0)  AZI=DPAI-AZI
      RETURN
      END
```


FINE TUNING PROGRAM OF STRETCHED-VISSR IMAGE MAPPING (5/5) JAPAN METEOROLOGICAL AGENCY / METEOROLOGICAL SATELLITE CENTER 1992