A Mapping Method for VISSR Date

Seiichiro Kigawa*

Abstract

Image mapping is used to process Visible and Infrared Spin Scan Radiometer (VISSR) image data, i.e., each pixel of the VISSR image data must correspond to its respective position on earth, thus making it necessary to transform between geodetic and VISSR frame coordinates. This report describes a coordinate transformation method which uses orbit and attitude prediction data to determine the position on the earth (geodetic coordinates) that corresponds to a VISSR image pixel (VISSR frame coordinates). On the other hand, it can also be conversely used to determine the VISSR image pixel which corresponds to a position on the earth. Another significant feature of the presented transformation method is that it calculates important information which can be utilized in other digital processing techniques, e.g., infrared (IR) digital image processing requires the satellite zenith distance, and visible (VIS) digital image processing uses the sun zenith distance, distance to the sun, and sun glint information. The applicable theory and sample coordinate transformation programs are presented. These programs were designed for a small-scale computer system which can utilize VISSR archive data that is stored at the Meteorological Satellite Center (MSC), and also Streched-VISSR data that is broadcasted via satellite.

1. Introduction

Image mapping is used to process Visible and Infrared Spin Scan Radiometer (VISSR) image data, i.e., each pixel of the VISSR image data must correspond to its respective position on earth, thus making it necessary to transform between geodetic and VISSR frame coordinates. Coordinate transformation allows converting the geodetic coordinates (latitude, longitude, height) to VISSR frame coordinates (line, pixel) and vice versa. This report describes a coordinate transformation method that uses orbit and attitude prediction data to determine the position on the earth which corresponds to a VISSR image pixel. On the other hand, it can also be conversely used to determine the VISSR image pixel which corresponds to a position on earth.

Another significant feature of the presented transformation method is that it calculates important information which can be utilized in other digital processing techniques, e.g., infrared (IR) digital image processing requires the satellite zenith distance, and visible (VIS) digital image processing uses the sun zenith distance, distance to the sun, and sun glint information. This information can easily be supplied because the positions of the sun, satellite, and earth reference point are all calculated with this coordinate transformation process.

The applicable theory and sample coordinate transformation programs are presented. These programs were designed for a small-scale computer system which can utilize VISSR archive data that is stored at the Meteorological Satellite Center (MSC), and also Streched-VISSR data that is broadcasted via satellite. (The mapping method for the S-VISSR image data are described in THE GMS USERS' GUIDE (Second Edition) APPENDIX F that is published by the MSC.)

* Meteorological Satellite Center.

2. Coordinate Transformation Theory

All parameters used for the VISSR image coordinate transformation are defined in Table 1, whereas Fig. 1-1 to 1-4 show applicable transformation flow charts.

The transformation consists of three stages: (1) The transformation from geodetic to VISSR coordinates, (2) The transformation from VISSR to the geodetic coordinates, and (3) The subsequent computation of information required for digital image processing. The information necessary for digital image processing are the sun and satellite zenith distances, sun and satellite azimuth angles, distances to the sun and satellite, satellite-sun digression, and sun glint data. The transformation from the geodetic to the VISSR coordinates (Fig. 1-2) necessitates a calculation reiteration because the scanning time corresponding to a point on the earth is unknown.

Table 1. Parameters Used for Coordinate Transformation

a. Coordinate Transformation Parameters

- t_s : Observation start time (UTC represented in MJD)
- P : Stepping angle along line (rad)
- Q : Sampling angle along pixel (rad)
- I_c : Center line number of VISSR frame
- J_c : Center pixel number of VISSR frame
- n : Number of sensors
- M_x : VISSR misalignment angle around x-axis (rad)
- M_y : VISSR misalignment angle around y-axis (rad)
- M_z : VISSR misalignment angle around z-axis (rad)
- [M] : VISSR misalignment matrix (3×3)
- R_e : Equatorial radius of the earth (m)
- f : Flattening of the earth

b. Attitude Parameters (33 sets at 5-minute intervals)

- t_n : Prediction time (UTC represented in MJD)
- α_r : Angle between z-axis and satellite spin axis projected on yz-plane in mean of 1950.0 coordinates (rad)
- δ_r : Angle between satellite spin axis and yz-plane (rad)
- β : β-angle (rad), i.e., angle between the sun and earth center on the z-axis vertical plane
- ω : Spin rate of satellite (rpm)

c. Orbital Parameters (9 sets at 5-minute intervals)

- t_n : Prediction time (UTC represented in MJD)
- X : X component of satellite position in the earth-fixed coordinates (m)
- Y : Y component of satellite position in the earth-fixed coordinates (m)
- Z : Z component of satellite position in the earth-fixed coordinates (m)
- θ_g : True Greenwich sidereal time (rad)
- α_s : Right ascension from satellite to the sun in the earth-fixed coordinates (rad)
- δ_s : Declination from satellite to the sun in the earth-fixed coordinates (rad)
- $[N_p]$: Nutation and precession matrix (3×3)

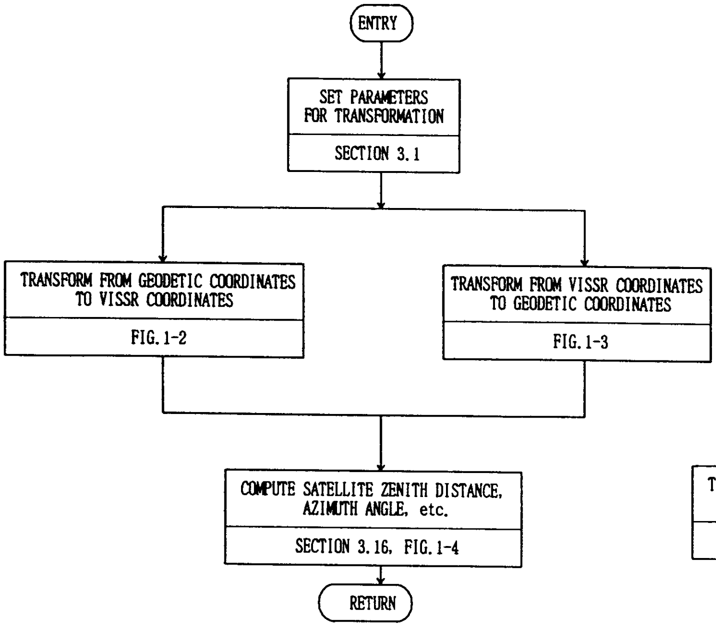


Fig. 1 − 1　Flow chart of coordinate transformation.

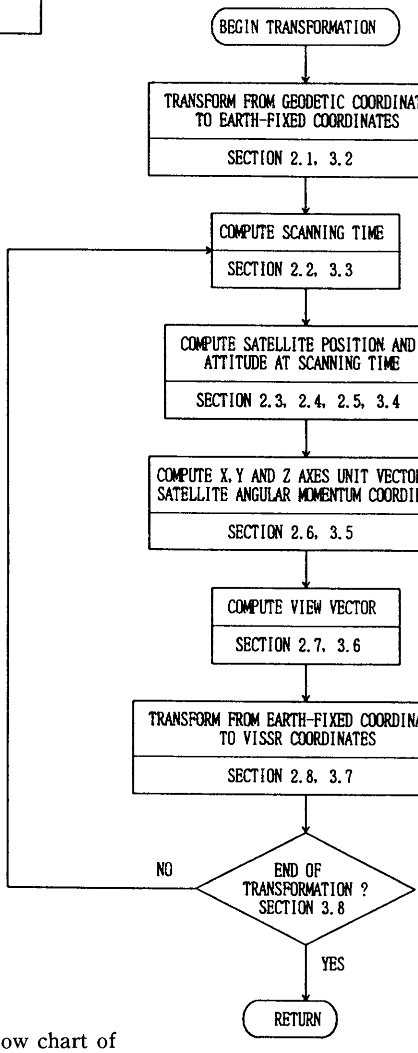


Fig. 1 − 2　Flow chart of transformation from geodetic to VISSR coordinates.

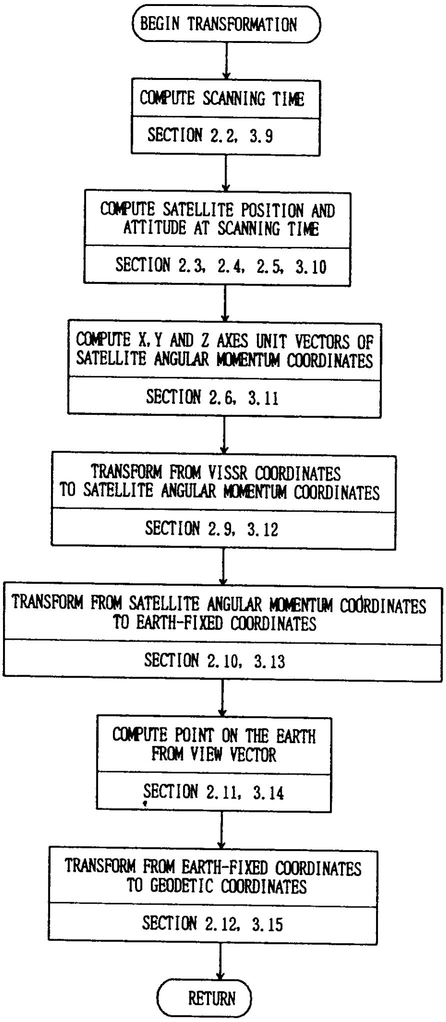


Fig. 1 − 3 Flow chart of transformation from VISSR to geodetic coordinates.

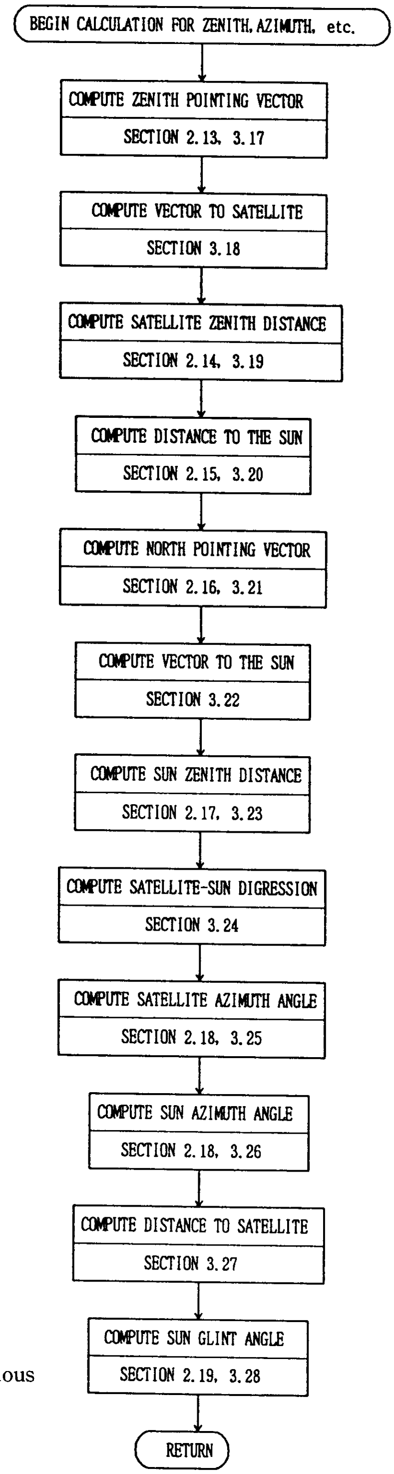


Fig. 1 − 4 Flow chart to calculate various transformation parameters.

2．1． Geodetic to Earth-fixed Transformation

The transformation from geodetic (ϕ, λ, h) to earth-fixed coordinates (X_e, Y_e, Z_e) is given by

$$\left.\begin{aligned} X_e &= (R_N+h)\cos\phi\cos\lambda \\ Y_e &= (R_N+h)\cos\phi\sin\lambda \\ Z_e &= \{R_N(1-e^2)+h\}\sin\phi \end{aligned}\right\} \quad (1)$$

where

$$R_N = \frac{R_e}{(1-e^2\sin^2\phi)^{0.5}} \quad (2)$$

ϕ : geodetic latitude, with north (+) and south (−)

λ: longitude, with east (+) and west (−)

h: height

with flattening of the earth f being related to eccentricity e by the below relation.

$$e^2 = 2f-f^2 \quad (3)$$

2．2 Scanning Time

Scanning time of a picture element (I, J) is given by

$$t_{IJ} = \frac{[(I-1)/n]+QJ/2\pi}{1440\omega} + t_s \quad (4)$$

where t_{IJ} is the scanning time represented in Modified Julian Date (MJD), I and J are line and pixel number of the point of interest, and [ ] denotes Gauss' notation.

2．3 Satellite Position and Attitude at Scanning Time

The orbit and attitude prediction data (α_r, δ_r, β, X, Y, Z, θ_g, α_s, δ_s) is interpolated to obtain values which correctly correspond to the scanning time. Interpolation is not necessary to determine the nutation and precession matrix $[N_P]$, thus prediction times occurring just prior to the scanning time can be employed.

Any parameter W of the orbit and attitude prediction data at time t_{IJ} is interpolated as follows,

$$W = W_0 + \frac{W_1-W_0}{t_1-t_0}(t_{IJ}-t_0) \quad (5)$$

where W_0, W_1 are 5-min data prediction intervals, and t_1, t_0 are the prediction times represented in MJD.

2．4 Mean of 1950.0 to True of Date Transformation

The transformation from the mean of 1950.0 coordinates $\mathbf{X_M}$ to the true of date coordinates $\mathbf{X_T}$ is given by

$$\mathbf{X_T} = [N_P]\cdot\mathbf{X_M} \quad (6)$$

where $[N_P]$ is the nutation and precession matrix.

2．5 True of Date to Earth-fixed Transformation

The true of date coordinates $\mathbf{X_T}$ are transformed into the earth-fixed coordinates $\mathbf{X_E}$ as

$$\mathbf{X_E} = [B]\cdot\mathbf{X_T} \quad (7)$$

where

$$[B] = \begin{bmatrix} \cos\theta_g & \sin\theta_g & 0 \\ -\sin\theta_g & \cos\theta_g & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

with θ_g being the true Greenwich sidereal time.

2．6 Axis Direction Unit Vectors of Satellite Angular Momentum Coordinates

Figure 2 shows the satellite's angular momentum coordinates, with the origin representing the satellite's center of gravity, the x-axis the direction of the vector which is rotated $\mathbf{S_s}$' around the z-axis to obtain the β angle ($\mathbf{S_s}$' is the sun direction vector projected onto the z-axis vertical plane), the y-axis which is used to form a right-handed coordinate system, and the z-axis which indicates the direction of the angular momentum vector.

The x, y, and z direction unit vectors of the satellite angular momentum coordinates which are transformed into the earth-fixed coordinates are defiened as

z-axis, $\mathbf{S_p}$:

$$\mathbf{S_p} = [B]\cdot[N_P]\cdot\begin{bmatrix} \sin\delta_r \\ -\cos\delta_r \cdot \sin\alpha_r \\ \cos\delta_r \cdot \cos\alpha_r \end{bmatrix} \quad (9)$$

x-axis, $\mathbf{S_x}$:

$$\mathbf{S_x} = \frac{\mathbf{S_p}\times\mathbf{S_s}}{|\mathbf{S_p}\times\mathbf{S_s}|}\sin\beta + \frac{\mathbf{S_p}\times\mathbf{S_s}}{|\mathbf{S_p}\times\mathbf{S_s}|} \times \mathbf{S_p}\cos\beta \quad (10)$$

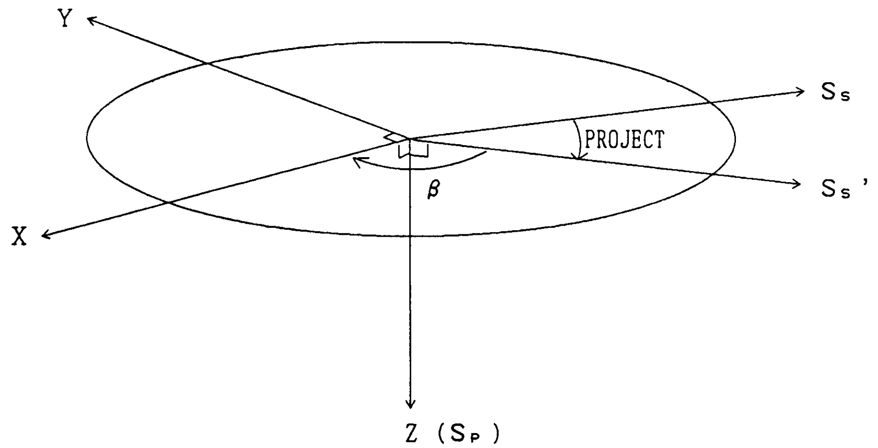


Fig. 2 Satellite angular momentum coordinates.

y-axis, $\mathbf{S_y}$:

$$\mathbf{S_y} = \mathbf{S_p} \times \mathbf{S_x} \tag{11}$$

where $\mathbf{S_s}$ is the vector from the satellite to the sun.

$$\mathbf{S_s} = \begin{bmatrix} \cos\delta_s \cdot \cos\alpha_s \\ \cos\delta_s \cdot \sin\alpha_s \\ \sin\delta_s \end{bmatrix} \tag{12}$$

2. 7 View Vector

The view vector $\mathbf{X_E}$ is directed from the satellite (X, Y, Z) to the point of interest (X_e, Y_e, Z_e) in the earth-fixed coordinates, and is expressed as

$$\mathbf{X_E} = \begin{bmatrix} X_e - X \\ Y_e - Y \\ Z_e - Z \end{bmatrix} \tag{13}$$

2. 8 Earth-fixed to VISSR Frame Transformation

Line number I and pixel number J of the point of interest in the VISSR frame coordinates are given by

$$\theta_L = \cos^{-1} \frac{\mathbf{X_E} \cdot \mathbf{S_p}}{|\mathbf{X_E}|\ |\mathbf{S_p}|} \tag{14}$$

$$I = \frac{(\pi/2 - \theta_L) - M_y}{P} + I_c \tag{15}$$

$$\mathbf{V_A} = \mathbf{S_p} \times \mathbf{X_E} \tag{16}$$

$$\mathbf{V_B} = \mathbf{S_y} \times \mathbf{V_A} \tag{17}$$

$$\theta_p = \cos^{-1} \frac{\mathbf{S_y} \cdot \mathbf{V_A}}{|\mathbf{S_y}|\ |\mathbf{V_A}|} \tag{18}$$

$$T_F = \mathbf{S_p} \cdot \mathbf{V_B} \tag{19}$$

if $T_F < 0$ then $\theta_p = -\theta_p$

$$J = \frac{\theta_p + M_z - (\pi/2 - \theta_L)\tan M_x}{Q} + J_c \tag{20}$$

2. 9 VISSR Frame to Satellite Angular Momentum Transformation

The vector $\mathbf{X_s}$ is directed from the satellite to the point of interest in the satellite angular momentum coordinates, and is expressed as

$$\mathbf{X_s} = \begin{bmatrix} \cos Q\,(J - J_c) & -\sin Q(J - J_c) & 0 \\ \sin Q\,(J - J_c) & \cos Q(J - J_c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot [M] \cdot \begin{bmatrix} \cos P\,(I - I_c) \\ 0 \\ \sin P\,(I - I_c) \end{bmatrix} \tag{21}$$

where I and J are line and pixel number of the point of interest in the VISSR frame coordinates.

2. 10 Satellite Angular Momentum to Earth-fixed Transformation

The satellite angular momentum coordinates $\mathbf{X_s}$ are transformed into the earth-fixed coordinates $\mathbf{X_E}$ as follows

$$\mathbf{X_E} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = [S]\cdot \mathbf{X_s} \tag{22}$$

where

$$[S] = [\mathbf{S_x}, \mathbf{S_y}, \mathbf{S_p}] \tag{23}$$

2. 11 View Vector to Point on the Earth

The point of interest on the earth is computed by the unit view vector $\mathbf{X_E}$ and satellite position (X, Y, Z) in the earth-fixed coordinates.

The view vector directed from the satellite to the point of interest is

$$\mathbf{X_E} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \tag{24}$$

$$k = \frac{-b \pm (b^2 - ac)^{0.5}}{a} \tag{25}$$

where

$$\left. \begin{aligned} a &= (1-f)^2(u_x^2 + u_y^2) + u_z^2 \\ b &= (1-f)^2(Xu_x + Yu_y) + Zu_z \\ c &= (1-f)^2(X^2 + Y^2 - R_e^2) + Z^2 \end{aligned} \right\} \tag{26}$$

Among the two solutions for k, the smaller absolute value is employed.

If the value of $b^2 - ac$ is negative, the view vector does not cross the earth surface, thus the point of interest in the earth-fixed coordinates is given by

$$\left. \begin{aligned} X_e &= X + ku_x \\ Y_e &= Y + ku_y \\ Z_e &= Z + ku_z \end{aligned} \right\} \tag{27}$$

2. 12 Earth-fixed to Geodetic Transformation

The transformation from the earth-fixed (X_e, Y_e, Z_e) to the geodetic coordinates (ϕ, λ) is given by

$$\phi = \tan^{-1}\left[\frac{Z_e}{(1-f)^2(X_e^2 + Y_e^2)^{0.5}}\right] \tag{28}$$

$$\lambda = \tan^{-1}\left[\frac{Y_e}{X_e}\right] \tag{29}$$

2. 13 Zenith Pointing Vector

The unit vector pointing to the zenith at subject $\mathbf{H}$ is given by

$$\mathbf{H} = \begin{bmatrix} \cos\phi\cos\lambda \\ \cos\phi\sin\lambda \\ \sin\phi \end{bmatrix} \tag{30}$$

where the subject is defined by the point of interest on the earth (Fig.3).

2. 14 Satellite Zenith Distance

The satellite zenith distance at the subject, Z_{SAT}, is computed by the vector $\mathbf{H}$ and the vector

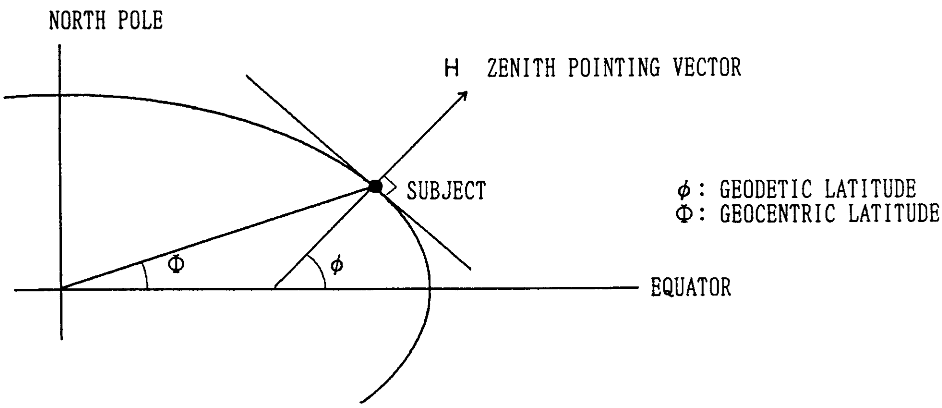


Fig. 3 Subject zenith pointing vector along the geodetic vertical.

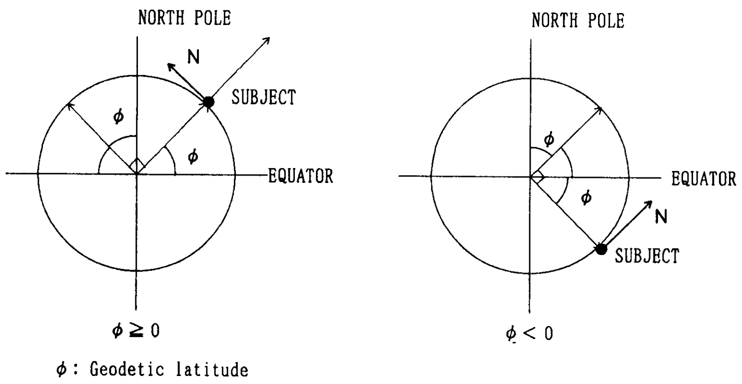
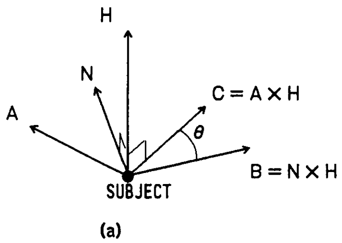


Fig. 4 Horizontal plane of vector that points north.



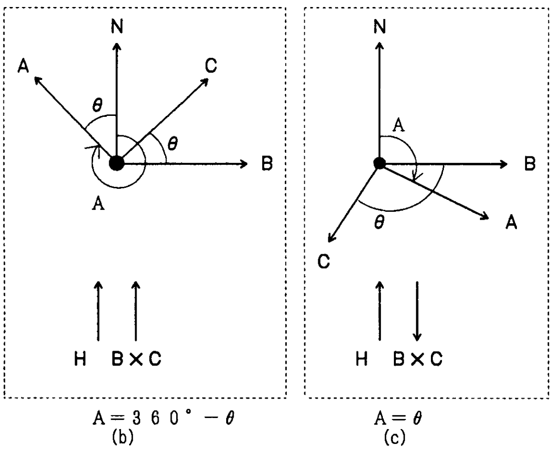


Fig. 5 Azimuth angle calculation.

(a) **A** : vector to the sun or satellite
H : zenith pointing vector
N : north pointing vector

(b) Azimuth angle A of the vector **A** is 360° $-\theta$
in the case where **H** and **B**×**C** are in the same direction.

(c) Azimuth angle A of the vector **A** is θ
in the case where **H** and **B**×**C** are in opposite directions.

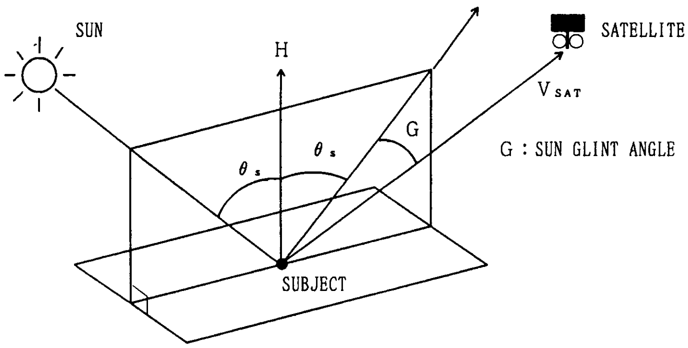


Fig. 6 Sun glint angle, i.e., the angle between the vector
of the sun's rays reflected at the subject and the vector from the subject to the satellite.

from the subject to the satellite $\mathbf{V}_{SAT}$.

$$Z_{SAT} = \cos^{-1}\frac{\mathbf{H}\cdot\mathbf{V}_{SAT}}{|\mathbf{H}|\ |\mathbf{V}_{SAT}|} \tag{31}$$

2. 15 Distance to the Sun

The distance from the earth to the sun is given by

$$\left.\begin{array}{l} A_M = 315^\circ.253 + 0^\circ.98560027 t_{IJ} \\ R_{SUN} = 1.00014 - 0.01672\cos A_M - 0.00014\cos 2A_M \end{array}\right\} \tag{32}$$

where t_{IJ} is the scanning time represented in MJD, and Rsun is expressed in astronomical units.

2. 16 North Pointing Vector

The vector in the horizontal plane that points north at the subject **N** is given by following equations (Fig.4).

$$\left.\begin{array}{l} \phi_N = 90^\circ - \phi \\ \lambda_N = \lambda - 180^\circ \end{array}\right\} \ \phi \geqq 0 \tag{33}$$

$$\left.\begin{array}{l} \phi_N = 90^\circ + \phi \\ \lambda_N = \lambda \end{array}\right\} \ \phi < 0 \tag{34}$$

if $\lambda_N \leqq -180^\circ$ then $\lambda_N = \lambda_N + 360^\circ$

$$\mathbf{N} = \begin{bmatrix} \cos\phi_N\cos\lambda_N \\ \cos\phi_N\sin\lambda_N \\ \sin\phi_N \end{bmatrix} \tag{35}$$

2. 17 Sun Zenith Distance

The sun zenith distance at the subject, Z_{SUN}, is computed by the vector **H** and the vector from the subject to the sun, $\mathbf{V}_{SUN}$.

$$Z_{SUN} = \cos^{-1}\frac{\mathbf{H}\cdot\mathbf{V}_{SUN}}{|\mathbf{H}|\ |\mathbf{V}_{SUN}|} \tag{36}$$

2. 18 Sun/Satellite Azimuth Angle

Azimuth angle A of a vector **A** at the subject is computed by the vector pointed to zenith **H** and the vector pointed north **N** at the subject (Fig. 5(a)−(c)). The vector **A** is either $\mathbf{V}_{SUN}$ or $\mathbf{V}_{SAT}$.

$$\mathbf{B} = \mathbf{N}\times\mathbf{H} \tag{37}$$

$$\mathbf{C} = \mathbf{A}\times\mathbf{H} \tag{38}$$

$$\theta_1 = \cos^{-1}\frac{\mathbf{B}\cdot\mathbf{C}}{|\mathbf{B}|\ |\mathbf{C}|} \tag{39}$$

$$\mathbf{D} = \mathbf{B}\times\mathbf{C}$$

$$\theta_2 = \cos^{-1}\frac{\mathbf{H}\cdot\mathbf{D}}{|\mathbf{H}|\ |\mathbf{D}|} \tag{41}$$

if $\theta_2 = 0^\circ$ then $A = 360^\circ - \theta_1$

if $\theta_2 = 180^\circ$ then $A = \theta_1$

2. 19 Sun Glint Angle

The sun glint angle, G (Fig.6), is defined as the angle between the vector of the sun's rays reflected at the subject and the vector from the subject to the satellite, being given by

$$\theta_S = \cos^{-1}\frac{\mathbf{H}\cdot\mathbf{V}_{SUN}}{|\mathbf{H}|\ |\mathbf{V}_{SUN}|} \tag{42}$$

$$\mathbf{S}_G = \mathbf{H}\cos\theta_S - \frac{\mathbf{H}\cdot\mathbf{V}_{SUN}}{|\mathbf{H}|\ |\mathbf{V}_{SUN}|} \times \mathbf{H}\sin\theta_S \tag{43}$$

$$G = \cos^{-1}\frac{\mathbf{S}_G\cdot\mathbf{V}_{SAT}}{|\mathbf{S}_G|\ |\mathbf{V}_{SAT}|} \tag{44}$$

3. Sample Programs

Sample programs are presented which are represented in FORTRAN (FORTRAN 77), and are applicable for both VISSR archive data that is stored at the MSC and S-VISSR data that is broadcasted via satellite.

Users must enter the parameters used in the coordinate transformation into a common block, where in this case the block name is MMAP1 having an array size of 672×4 words.

The sample programs listings are given at the end of this section.

COMMON/MMAP1/MAP(672,4)

MAP(1,1)～MAP(672,1)…… COORDINATE TRANSFORMATION PARAMETERS

MAP(1,2)～MAP(672,2)…… ATTITUDE PREDICTION DATA

MAP(1,3)～MAP(672,3)……ORBIT PREDICTION DATA 1

MAP(1,4)～MAP(672,4)……ORBIT PREDICTION DATA 2

a. VISSR Archive Data

VISSR IR ARCHIVE DATA (3rd or 5th block)

COORDINATE TRANSFORM. PARAMETERS SEGMENT	ATTITUDE PREDICTION DATA	(reserved)	ORBIT PREDICTION DATA 1	ORBIT PREDICTION DATA 2	(reserved)
672 words	672 words	408 words	672 words	672 words	408 words

VISSR VIS ARCHIVE DATA (3rd or 5th block)

OPERATIONAL MODE SEGMENT	S/DB OPERATION INFORMATION SEGMENT	IR DATA CALIBRATION TABLE	VIS DATA CALIBRATION TABLE	(reserved)
672 words	672 words	672 words	672 words	688 words

COORDINATE TRANSFORM. PARAMETERS SEGMENT	ATTITUDE PREDICTION DATA	ORBIT PREDICTION DATA 1	ORBIT PREDICTION DATA 2	(reserved)
672 words	672 words	672 words	672 words	688 words

b. S-VISSR Data

If the orbit and attitude prediction data in the documentation sector of the S-VISSR data is used, the sample programs can be executed exactly as described. The parameters in the documentation sector of the S-VISSR which are entered into the common block are shown in Tables 2~ 4.

Table 2　Coordinate Transformation Parameters Segment

Position (word)	Type	Contents	S-VISSR Orbit and Attitude Data Block Position (word)	Type
5-6	R*8	Observation start time	1-6	R*6.8
7	R*4	VIS channel stepping angle along line (rad)	7-10	R*4.8
8	R*4	IR channel stepping angle along line (rad)	11-14	R*4.8
11	R*4	VIS channel sampling angle along pixel (rad)	15-18	R*4.10
12	R*4	IR channel sampling angle along pixel (rad)	19-22	R*4.10
15	R*4	VIS channel center line number of VISSR frame	23-26	R*4.4
16	R*4	IR channel center line number of VISSR frame	27-30	R*4.4
19	R*4	VIS channel center pixel number of VISSR frame	31-34	R*4.4
20	R*4	IR channel center pixel number of VISSR frame	35-38	R*4.4
27	R*4	Number of sensors of VIS channel	39-42	R*4.0
28	R*4	Number of sensors of IR channel	43-46	R*4.0
31	R*4	VIS total line number of VISSR frame	47-50	R*4.0
32	R*4	IR total line number of VISSR frame	51-54	R*4.0
35	R*4	VIS pixel number of one line	55-58	R*4.0
36	R*4	IR pixel number of one line	59-62	R*4.0
39	R*4	VISSR misalignment angle around x-axis (rad)	63-66	R*4.10
40	R*4	VISSR misalignment angle around y-axis (rad)	67-70	R*4.10
41	R*4	VISSR misalignment angle around z-axis (rad)	71-74	R*4.10
42	R*4	Element of VISSR misalignment matrix on row 1 and column 1	75-78	R*4.7
43	R*4	- row 2 and column 1	79-82	R*4.10
44	R*4	- row 3 and column 1	83-86	R*4.10
45	R*4	- row 1 and column 2	87-90	R*4.10
46	R*4	- row 2 and column 2	91-94	R*4.7
47	R*4	- row 3 and column 2	95-98	R*4.10
48	R*4	- row 1 and column 3	99-102	R*4.10
49	R*4	- row 2 and column 3	103-106	R*4.10
50	R*4	- row 3 and column 3	107-110	R*4.7
131-132	R*8	Attitude parameters - Daily mean of satellite spin rate (rpm)	241-246	R*6.8

Table 3 Attitude Prediction Data Segment

Position (word)	Type	Contents	S-VISSR Attitude Prediction Data Sub-Block Position (word)	Type
13-672		33 sets of 20 words Attitude predciction data (relative position as below)		
0-1	R*8	Prediction time (UTC represented in MJD)	1-6	R*6.8
4-5	R*8	Angle between z-axis and satellite spin axis projected on yz-plane in mean of 1950.0 coordinates (rad)	13-18	R*6.8
6-7	R*8	Angle between satellite spin axis and yz-plane in mean of 1950.0 coordinates (rad)	19-24	R*6.11
8-9	R*8	β-angle (rad)	25-30	R*6.8
10-11	R*8	Spin rate: spin speed of satellite (rpm)	31-36	R*6.8

Table 4 Orbit Prediction Data Segment

Position (word)	Type	Contents	S-VISSR Orbit Prediction Data Sub-Block Position (word)	Type
13-642		9 sets of 70 words Orbit prediction data (relative position as below)		
0-1	R*8	Prediction time (UTC represented in MJD)	1-6	R*6.8
16-17	R*8	X component of satellite position in the earth-fixed coordinates (m)	49-54	R*6.6
18-19	R*8	Y component of satellite position in the earth-fixed coordinates (m)	55-60	R*6.6
20-21	R*8	Z component of satellite position in the earth-fixed coordinates (m)	61-66	R*6.6
28-29	R*8	Greenwich sidereal time in true of date coordinates (deg)	85-90	R*6.8
34-35	R*8	Right ascension from the satellite to the sun in the earth-fixed coordinates (deg)	103-108	R*6.8
36-37	R*8	Declination from the satellite to the sun in the earth-fixed coordinates (deg)	109-114	R*6.8
		Element of nutation and precession matrix		
38-39	R*8	- row 1 and column 1	129-134	R*6.12
40-41	R*8	- row 2 and column 1	135-140	R*6.14
42-43	R*8	- row 3 and column 1	141-146	R*6.14
44-45	R*8	- row 1 and column 2	147-152	R*6.14
46-47	R*8	- row 2 and column 2	153-158	R*6.12
48-49	R*8	- row 3 and column 2	159-164	R*6.14
50-51	R*8	- row 1 and column 3	165-170	R*6.14
52-53	R*8	- row 2 and column 3	171-176	R*6.14
54-55	R*8	- row 3 and column 3	177-182	R*6.12

```
C***********************************************************************
      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  CONTENTS
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) SATELLITE-SUN DIGRESSION (DEGREES)
C                      (6) SATELLITE DISTANCE (METER)
C                      (7) SUN DISTANCE (KILOMETER)
C                      (8) SUN GLINT ANGLE (DEGREES)
C   DSCT     O   R*8   SCAN TIME (MJD)
C   IRTN     O   I*4   RETURN CODE
C                        0=O.K.
C                        1=IMODE PARAMETER ERROR
C                        2=RLAT PARAMETER ERROR
C                        3=MISSING NUMBER
C                        4=PIXEL OF POINT IS OUT OF VERTICAL RANGE OF
C                          VISSR FREAM
C                        5=PIXEL OF POINT IS OUT OF HORIZONTAL RANGE
C                          OF VISSR FREAM
C                        6=APPOINTED POSITION IN THE GEOGRAPHICAL
C                          CO-ORDINATES IS OUT OF RANGE OF VISSR
C                          OBSERVATION AREA
```

```
C                        7=APPOINTED PIXEL IN THE VISSR IMAGE
C                          CO-ORDINATES IS IN SPACE
C                        8=MISSING NUMBER
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!!!!!!!!!!!!!!!!! CONSTANT SET !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      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)                                          [3.1 ]
      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                                      [3.2 ]
        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
        RI0      = RFCL-ATAN(SIN(SNGL(DLAT))/(6.610689-COS(SNGL(DLAT))))
     &             /RSTEP
        RTIM     = DTIMS+DBLE(RI0/SENS/1440.)/DSPIN                    [3.3 ]
C
  100   CONTINUE
        CALL  MGI100(RTIM,CDR,SAT,SP,SS,BETA)                          [3.4 ]
        CALL  MGI220(SP,SS,SW1)                                        [3.5 ]
        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)                                      [3.6 ]
        SLV(2)   = STN1(2)-SAT(2)
        SLV(3)   = STN1(3)-SAT(3)
        CALL  MGI200(SLV,SL)                                           [3.7 ]
        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-RI0).GE.EPS)  THEN                                   [3.8 ]
          RTIM   = DBLE(AINT((RI-1.)/SENS)+RJ*RSAMP/SNGL(DPAI))/
     &             (DSPIN*1440.D0)+DTIMS
          RI0    = 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))/   [3.9 ]
     .             (DSPIN*1440.D0)+DTIMS
        CALL  MGI100(RTIM,CDR,SAT,SP,SS,BETA)                          [3.10]
        CALL  MGI220(SP,SS,SW1)                                        [5.11]
        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)))                     [3.12]
            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)            [3.13]
            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)                                         [3.14]
            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)+         [3.15]
                       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 ET CETERA) !!!!!!!!!!! [3.16]
          STN2(1)    = DCOS(DLAT)*DCOS(DLON)                             [3.17]
          STN2(2)    = DCOS(DLAT)*DSIN(DLON)
```

```
      STN2(3)    = DSIN(DLAT)
      SLV(1)     = SAT(1)-STN1(1)                                     [3.18]
      SLV(2)     = SAT(2)-STN1(2)
      SLV(3)     = SAT(3)-STN1(3)
      CALL  MGI200(SLV,SL)
C
      CALL  MGI230(STN2,SL,DSATZ)                                     [3.19]
      IF(DSATZ.GT.HPAI)  IRTN = 7
C
      SUNM     = 315.253D0+0.98560027D0*RTIM                          [3.20]
      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                                           [3.21]
        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                                [3.22]
      SW1(2)  = SLV(2)+SS(2)*SDIS*1.D3
      SW1(3)  = SLV(3)+SS(3)*SDIS*1.D3
      CALL  MGI200(SW1,SW2)                                           [3.23]
      CALL  MGI230(STN2,SW2,DSUNZ)
      CALL  MGI230(SL,SW2,DSSDA)                                      [3.24]
      CALL  MGI240(SL,STN2,STN3,DPAI,DSATA)                           [3.25]
      CALL  MGI240(SW2,STN2,STN3,DPAI,DSUNA)                          [3.26]
      DSATD    = DSQRT(SLV(1)*SLV(1)+SLV(2)*SLV(2)+SLV(3)*SLV(3))     [3.27]
C
      CALL  MGI230(SW2,STN2,DSUNG)                                    [3.28]
      CALL  MGI220(STN2, SW2,SW3)
      CALL  MGI220(SW3,STN2, 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
C************************************************************************
      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(10,33),ATT1(3),ATT2(3),ATT3(3),NPA(3,3),
C                        ‾‾ 10 FOR S-VISSR
     .           ORBT1(35,8),ORBT2(35,8),SAT(3),SP(3),SS(3)
C                            ‾‾‾‾‾‾‾‾‾‾‾ NOT USED FOR S-VISSR
      INTEGER*4 MAP(672,4)
C
      EQUIVALENCE (MAP(13,3),ORBT1(1,1)),(MAP(13,4),ORBT2(1,1))
C                                        ‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾ NOT USED FOR S-VISSR
      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
 ..............................................................................
      IF(RTIM.GE.ORBT1(1,8).AND.RTIM.LT.ORBT2(1,1))  THEN
        CALL  MGI110
     .       (8,RTIM,CDR,ORBT1,ORBT2,SAT,SITAGT,SUNALP,SUNDEL,NPA)
        GO TO  1200
      ENDIF
      DO 1100 I=1,7
        IF(RTIM.GE.ORBT2(1,I).AND.RTIM.LT.ORBT2(1,I+1))  THEN
          CALL  MGI110
     .         (I,RTIM,CDR,ORBT2,ORBT1,SAT,SITAGT,SUNALP,SUNDEL,NPA)
          GO TO  1200
        ENDIF
 1100 CONTINUE                                     NOT USED FOR S-VISSR
 ..............................................................................
      WRITE(6,*) '  ** ORBIT DATA ERROR **'
 1200 CONTINUE
C
      DO 3000 I=1,33-1
C                 ‾‾ 9 FOR S-VISSR
        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
          GO TO  3001
        ENDIF
 3000 CONTINUE
      WRITE(6,*) '  ** ATTITUDE DATA ERROR **'
 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
C***********************************************************************
      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
        SUNALP    =(ORBTA(18,I)+(ORBTA(18,I+1)-ORBTA(18,I))*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)
      ELSE
        DELT=(RTIM-ORBTA(1,8))/(ORBTB(1,1)-ORBTA(1,8))
        SAT(1)    = ORBTA( 9,8)+(ORBTB( 9,1)-ORBTA( 9,8))*DELT
        SAT(2)    = ORBTA(10,8)+(ORBTB(10,1)-ORBTA(10,8))*DELT
        SAT(3)    = ORBTA(11,8)+(ORBTB(11,1)-ORBTA(11,8))*DELT
        SITAGT    =(ORBTA(15,8)+(ORBTB(15,1)-ORBTA(15,8))*DELT)*CDR
        SUNALP    =(ORBTA(18,8)+(ORBTB(18,1)-ORBTA(18,8))*DELT)*CDR
        SUNDEL    =(ORBTA(19,8)+(ORBTB(19,1)-ORBTA(19,8))*DELT)*CDR
        NPA(1,1) = ORBTA(20,8)
        NPA(2,1) = ORBTA(21,8)
        NPA(3,1) = ORBTA(22,8)
        NPA(1,2) = ORBTA(23,8)
        NPA(2,2) = ORBTA(24,8)
        NPA(3,2) = ORBTA(25,8)
        NPA(1,3) = ORBTA(26,8)
        NPA(2,3) = ORBTA(27,8)
        NPA(3,3) = ORBTA(28,8)                         NOT USED FOR S-VISSR
      ENDIF
      RETURN
      END
C***********************************************************************
      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
C**********************************************************************
      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
C**********************************************************************
      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
C**********************************************************************
      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
C**********************************************************************
      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
```

VISSR 画像の座標変換プログラム

木川　誠一郎
気象衛星センター　システム管理課

画像位置合わせとは，VISSR 画像データの各画素と地球上の緯経度との対応づけをすることである。画像位置合わせを行うためには，VISSR 画像の座標系と地球上の緯経度の座標系との変換を行う必要がある。

ここでは，VISSR 画像の座標変換プログラムを紹介する。このプログラムを用いることにより，地球上の任意の地点に対応する VISSR 画像上の画素を知ることができる。逆に，VISSR 画像上の任意の画素に対応する地球上の地点の緯経度を知ることもできる。また，座標変換と同時に衛星天頂角や太陽天頂角等の画像解析に重要な情報を容易に算出できる。

このプログラムは気象衛星センターの VISSR 保存データとストレッチド VISSR データの両方に使用できる。