気象衛星センター　技術報告第32号　1996年11月

Image Navigation of Multi-functional Transport Satellite

Seiichiro Kigawa*

Abstract

Multi-functional Transport Satellite (MTSAT) will be launched in 1999. Stretched-VISSR in GMS system for Medium-scale Data Utilization Station (MDUS) users will be replaced by High Resolution Imager Data (HiRID). The HiRID format will be based on the GMS-5 S-VISSR format. There will be no change in parameters for image navigation.

This report describes the concept of the image navigation for MTSAT based on GMS image navigation. The applicable theory and sample programs for the coordinate transformation written in FORTRAN are presented in the Appendix.

1. What is "Image Navigation"?

Image Navigation involves making each pixel of Imager (Visible and Infrared Radiometer of Multi-functional Transport Satellite) or VISSR (Visible and Infrared Spin Scan Radiometer) correspond to its position on the earth. Image navigation is essential to process Imager or VISSR data with image data calibration.

2. What is "MTSAT"?

This report describes the concept of the image navigation for MTSAT based on GMS (Geostationary Meteorological Satellite of Japan) image navigation.

JMA is going on procedures for procurement and production of MTSAT as a successor to GMS-5 in cooperation with the Civil Aviation Bureau (CAB), Ministry of Transport of Japan since 1994. MTSAT has two functions: one is for meteorological services in JMA and the other for air-traffic control services in CAB. MTSAT will be launched in geostationary orbit at 140 east longitude around August 1999.

The fundamental specification of the meteorological mission of MTSAT is shown in Table 1. As for the meteorological mission, the specifications of GMS-5 have been fundamentally succeeded and an infrared sensor with a wavelength of 3.5 μm has been added.

The resolution at the sub-satellite point of the Stretched-VISSR (S-VISSR) of GMS-5 is 1.25 km for visible channel and 5 km for infrared channels. It is necessary that the High Resolution Imager Data (HiRID: in place of S-VISSR) of MTSAT have the same format as S-VISSR while GMS-5 is in geostationary orbit as a back-up satellite. The HiRID format will be based on the GMS-5 S-VISSR format. The spatial resolution of HiRID will be 1.25 km for the visible channel and 5 km for infrared channels. The signal quantization of HiRID will be 6 bits for the visible channel and 10

---

* Meteorological Satellite Center

(1996年8月8日受領、1996年9月9日受理)

Table 1. Fundamental Specifications of MTSAT

Design Life	more than 5 years for meteorological mission more than 10 years for air-traffic control mission
Orbital Position	±0.1°north-south and east-west from 140°E
Imaging Period	within 27.5 minutes
Imaging Channels	1 Visible and 4 Infrared
Signal Quantization	10 bits for both Visible and Infrared
Spatial Resolution	1 km for Visible, 4 km for Infrared
Telecommunication Function	● Transmission of raw Imager data ● Relay of High Resolution Imager Data (HiRID: in place of S-VISSR) ● Relay of WEFAX and LRIT signals ● Relay of DCP signals

bits for infrared channels. The additional data and bits will be put into unused area in the current GMS-5 S-VISSR format. This will hardly affect the current Medium-scale Data Utilization Station (MDUS) users as long as they use only the data that are included in the current GMS-5 S-VISSR format. MDUS users who plan to utilize the additional new data will need to modify the processing system of MDUS.

3. Image Navigation of GMS/VISSR

3.1 Parameters of Image Navigation

Figure 1 shows the concept of VISSR navigation. The ultimate purpose of the image navigation is to transform between geodetic and VISSR frame coor-

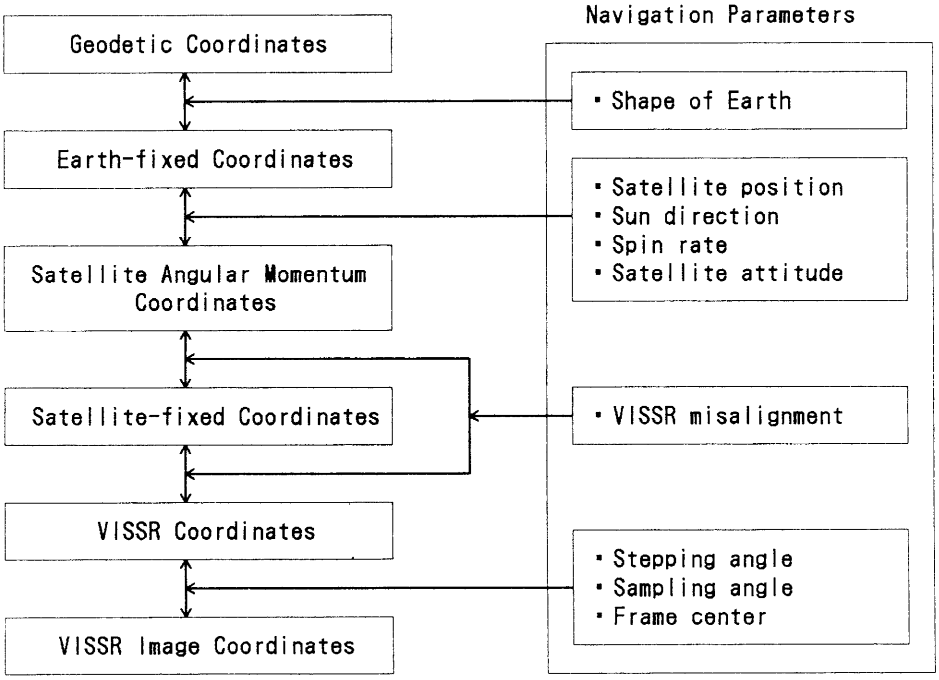


Figure 1 Schematic diagram showing the concept of Navigation

dinates. Therefore, image navigation is often called "coordinate transformation."

The parameters used for image navigation are the shape of the earth, satellite position, sun direction, spin rate, satellite attitude, VISSR misalignment and constants of VISSR frame. It is possible to perform the transformation shown in Figure 1 using these parameters.

3.2 Making of Parameters

The parameters used for image navigation are made in the Image Processing, Orbit and Attitude Processing of DPC (Data Processing Center). Figure 2 shows the concept of the source of parameters for image navigation.

3.2.1 Shape of Earth

It is necessary to use an ellipsoid of revolution (spheroid) to transform from geodetic coordinates (i.e., latitude, longitude, height) to earth-fixed coordinates. The shape of the Earth includes the parameters of a spheroid from the Image constant data file.

3.2.2 Satellite Position

The satellite position is calculated by the range

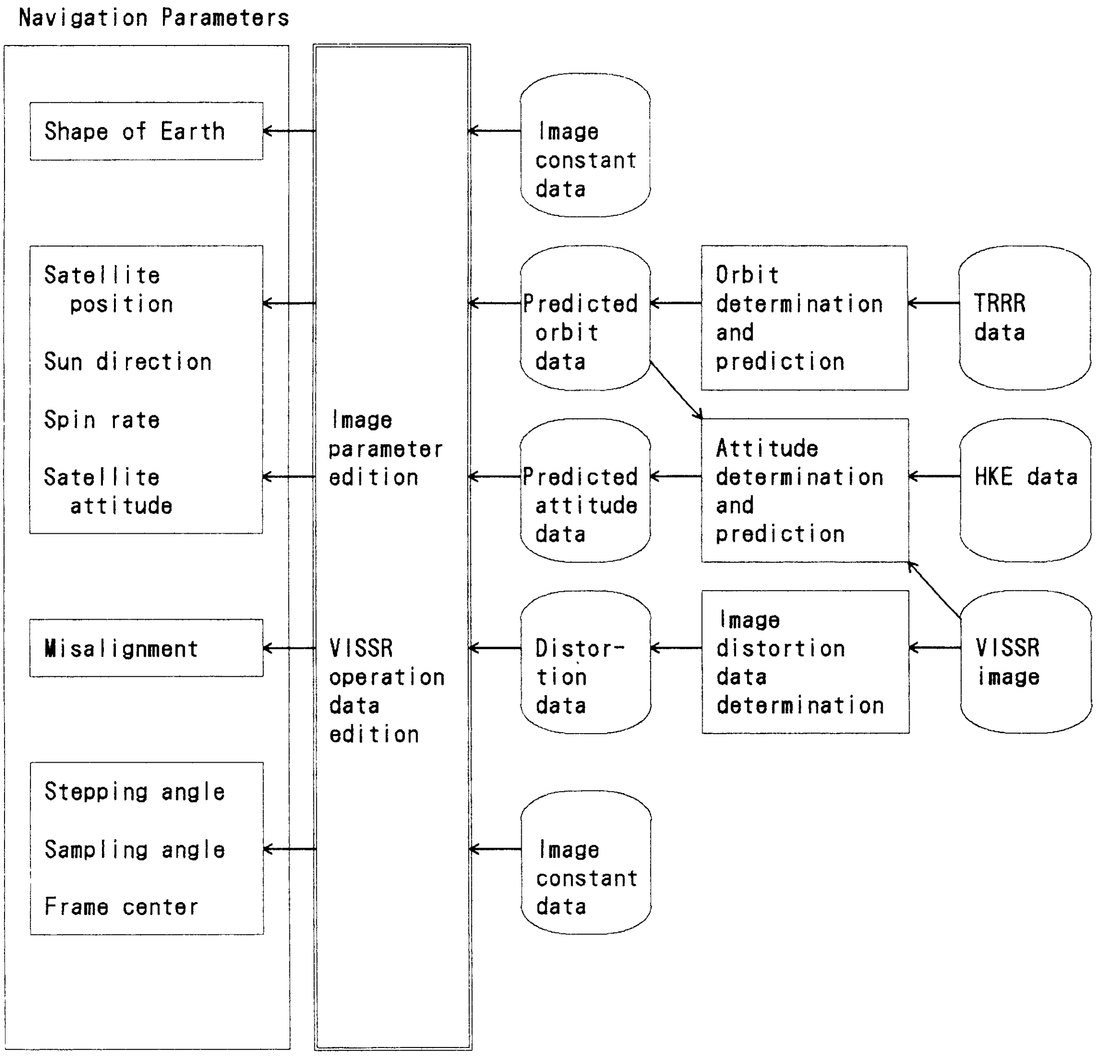


Figure 2 Schematic diagram showing the concept of the source of parameters for Navigation

data of TRRR (Trilateration Range and Range Rate). The TRRR system consists of three stations: CDAS, Ishigaki island and Crib Point in Australia for the measurement of the distance between satellite and stations. The TRRR is performed once every 6 hours. Each TRRR measurement is performed for a period of 6 minutes. Orbit Processing determines the orbital parameters of the satellite using the range data and predicts the position of the satellite at the future time using determined orbital parameters. The satellite position used for image navigation is the predicted position of the satellite. GMS system is able to provide the predicted position within 100 meters of error.

3.2.3 **Spin Rate**

The spin rate is the period of satellite rotation. It is calculated by the data of the sun sensor on the satellite. Satellite telemetry contains the sun sensor data. Attitude Processing picks up the sun sensor data from the House Keeping Edit file for the satellite telemetry and calculates the spin rate. The spin rate used for image navigation is the 1-day mean value.

3.2.4 **Sun Direction**

The sun direction used for image navigation is provided from the orbit processing for the reference direction of the satellite attitude.

3.2.5 **Satellite Attitude**

The satellite attitude used for image navigation consists of spin-axis direction and β angle. The spin-axis direction is determined by landmark positions on VISSR image. The attitude processing

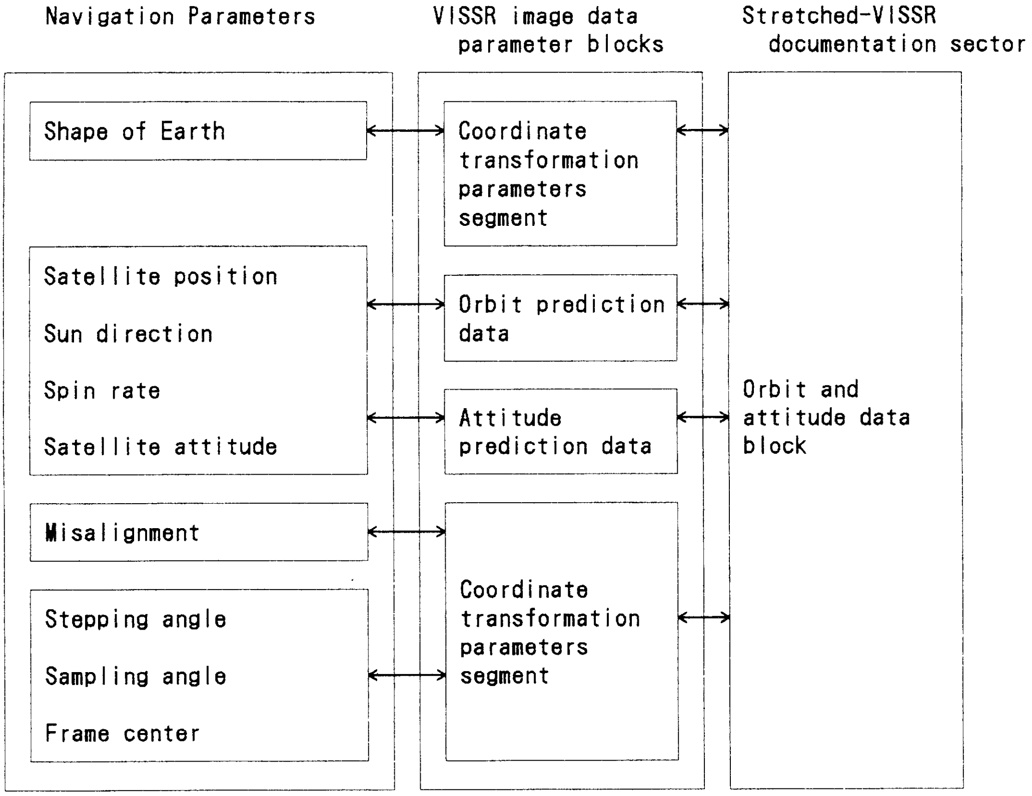


Figure 3 Schematic diagram showing the concept of Navigation parameters in VISSR data

extracts the landmark positions using the reference map data of 99 reference points from visible image. The attitude processing determines the spin-axis direction using the landmark positions and predicts the future spin-axis direction. The spin-axis direction used for image navigation is the predicted spin-axis direction within 0.01 degrees of error.

3.2.6 Misalignment

The misalignment angles had been used for the furnishing alignment angles of VISSR for the satellite originally, however, are used for the tuning of the image navigation.

The initial values of the misalignment angles are determined in the attitude processing. However, there is the degradation of the image navigation accuracy because it is difficult to predict the satellite attitude completely, for example, after thruster firing for satellite station keeping. To solve this problem, a fine tuning technique called Image Distortion Data Determination based on detection of the earth's edge from VISSR infrared image is used. The fine tuning process extracts the position of the earth in the infrared image and calculates the deviation of the earth image. The misalignment angles are corrected with the deviation of the earth image to improve the image navigation accuracy. The misalignment angles used for image navigation are corrected angles after the fine tuning process.

3.2.7 Constant of VISSR frame

The stepping angle and sampling angle of VISSR frame mean pixel lattice intervals in the north-south and east-west directions, respectively. These parameters and VISSR frame center positions are defined in the Image constant data file.

3.3 Storing of Parameters

The parameters used for image navigation are stored in the Orbit and attitude data block of S-VISSR documentation sector. Figure 3 shows the concept of the image navigation parameter storing.

3.4 Image Navigation Accuracy

The image navigation error of the S-VISSR is 1 to 1.5 pixel RMSE in normal operation. The Image Distortion Data Determination can be used to correct the processes of all products at the MSC (Meteorological Satellite Center), but it can not correct S-VISSR because of the broadcast time schedule. Therefore, the image navigation error of the S-VISSR 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 S-VISSR for MDUS user's computer system was developed (Kigawa: 1993).

4. Image Navigation of MTSAT

4.1 Parameters of Image Navigation

The attitude parameters of MTSAT are deferent from GMS attitude parameters because the attitude control methods of MTSAT and GMS are deferent, i.e., GMS is spin-stabilized satellite and MTSAT is three-axis-stabilized satellite. However, the HiRID will use the same parameters as S-VISSR for image navigation. The attitude parameters of MTSAT will convert to the GMS attitude parameters of the HiRID. This will hardly affect the current MDUS users.

4.2 Making of Parameters

The HiRID documentation that includes parame-

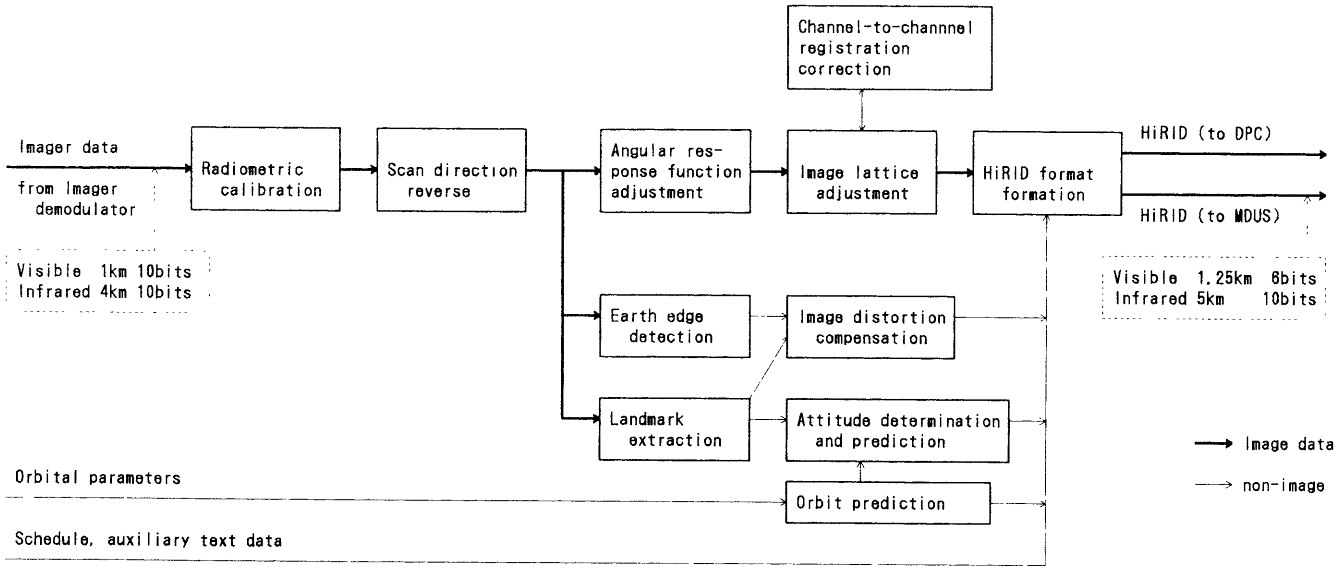


Figure 4 Ground Processing for MTSAT image navigation

ters for image navigation will be created by the Image Processing System in CDAS. Figure 4 shows the concept of the ground system for MTSAT image navigation.

4.3 Storing of Parameters

There will be no change in the Orbit and attitude data block format of HiRID documentation sector from the S-VISSR format.

4.4 Image Navigation Accuracy

To reduce the error of the image navigation, real-time landmark extraction and fine tuning for Imager data navigation will be performed in CDAS for MTSAT. The specification of the image navigation is less than 35μrad (1.25 km) RMSE.

4.5 Notice

This section describes some information for MDUS users based on the current HiRID plan. Formal and detailed information will be announced by JMA headquarters in the near future.

(1) Image navigation for additional channels
MDUS users will be able to use coordinate transformation for infrared channel 1 in place of coordinate transformation for new channel (3.5-4.0μm). This means pixels located at the same position in the frames of both channels show the same position on the ground.

(2) Parameters Update
The parameters for the image navigation of HiRID will be updated by real time navigation (landmark extraction and fine tuning using the earth's edge). It is important to note that updated and corrected parameters will be sent about 8 minutes after HiRID dissemination start time. The parameters that will be sent for the first 8 minutes will be predicted parameters. (It takes 11 minutes for the northern hemisphere HiRID dissemination and 22 minutes for full disk HiRID dissemination.) Figure 5 shows the current plan of HiRID format and its documentation updates.

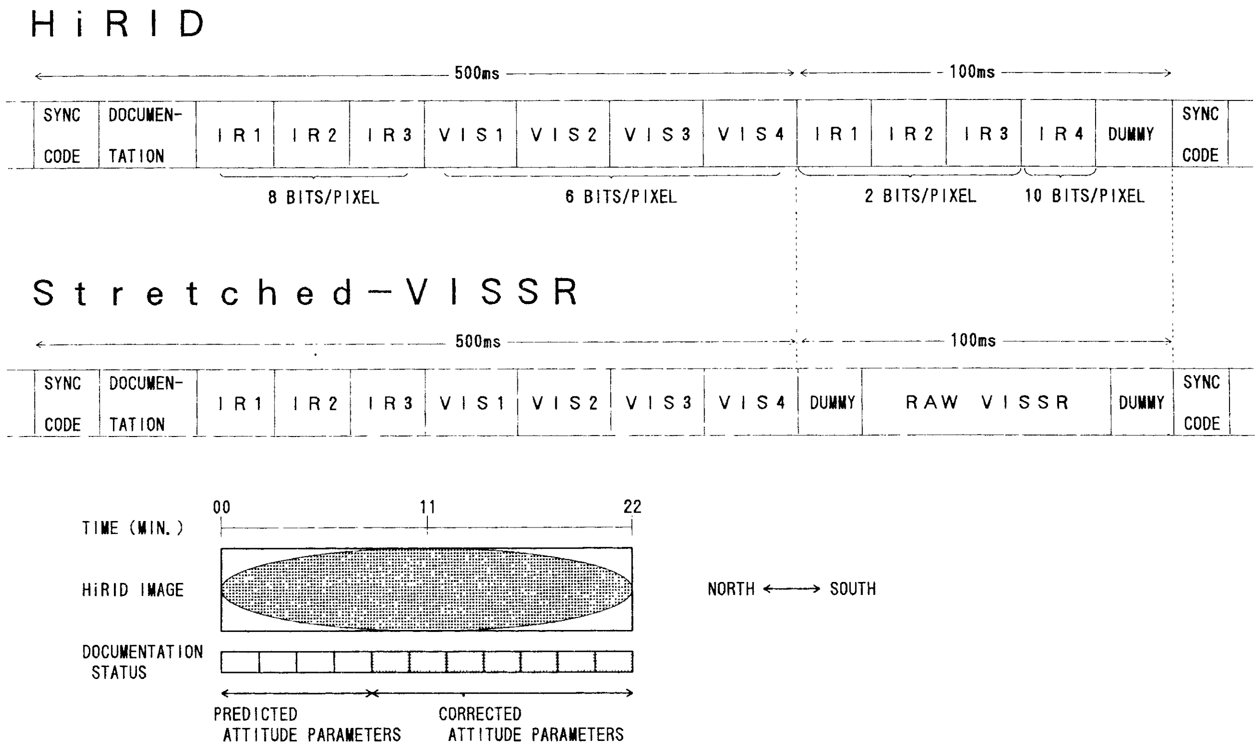


Figure 5 Current Plan for HiRID

5. Coordinate transformation for S-VISSR and HiRID

The applicable theory and sample programs for the coordinate transformation written in FORTRAN are presented in the Appendix. The sample programs are applicable for both S-VISSR and HiRID.

Reference

Kigawa, S., 1993. Fine Tuning of Stretched-VISSR Image Mapping. Meteorological Satellite Center Technical Note, No. 26, March 1993, p 1-10.

運輸多目的衛星のナビゲーション

木川誠一郎*

運輸多目的衛星は、1999年に打上げられる予定である。GMS システムにおけるストレッチド VISSR は、高分解能イメージャデータ（HiRID: High Resolution Imager Data）に置き換えられる予定である。高分解能イメージャデータのフォーマットは、ストレッチド VISSR を拡張したフォーマットになる予定であり、画像データのナビゲーション（画像位置合わせ）に使用するパラメータは変わらない予定である。

このレポートでは、運輸多目的衛星における画像のナビゲーションの概念を解説する。ナビゲーションに使用する座標変換プログラムを付録に添付している。

APPENDIX

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

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 vertocal 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 cpmponent of salellite position in the earth-fixed coordinates (m)
Z : Z cpmponent of salellite 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)

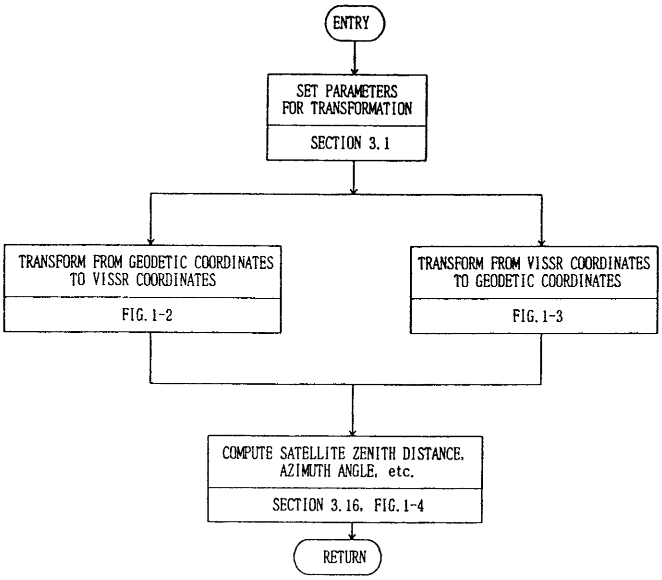


Fig. 1-1 Flow chart of coordinate transformation.

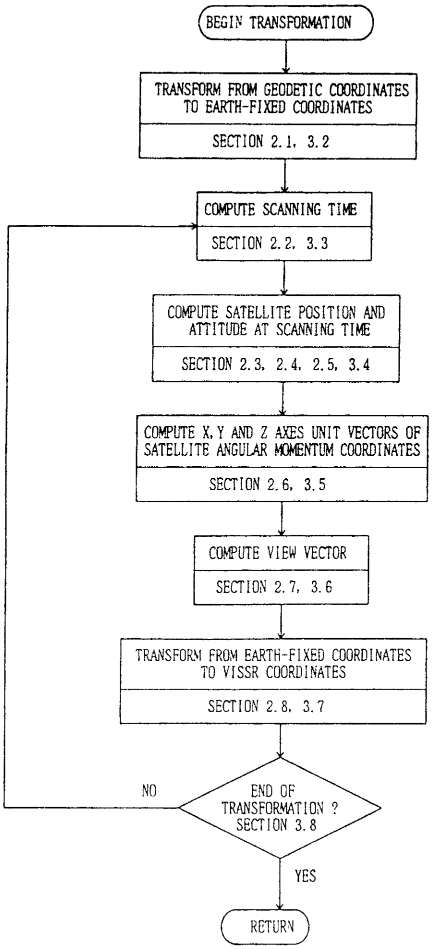


Fig. 1-2 Flow chart of transformation from geodetic to VISSR coordinates.

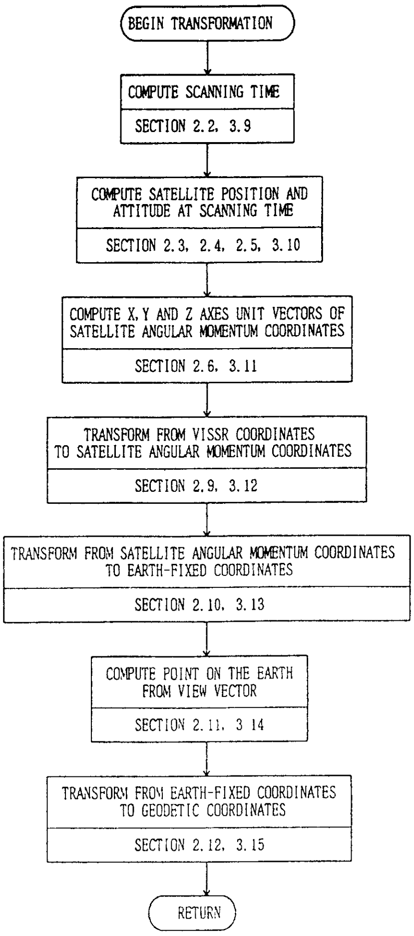


Fig. 1-3 Flow chart of transformation from VISSR to geodetic coordinates.

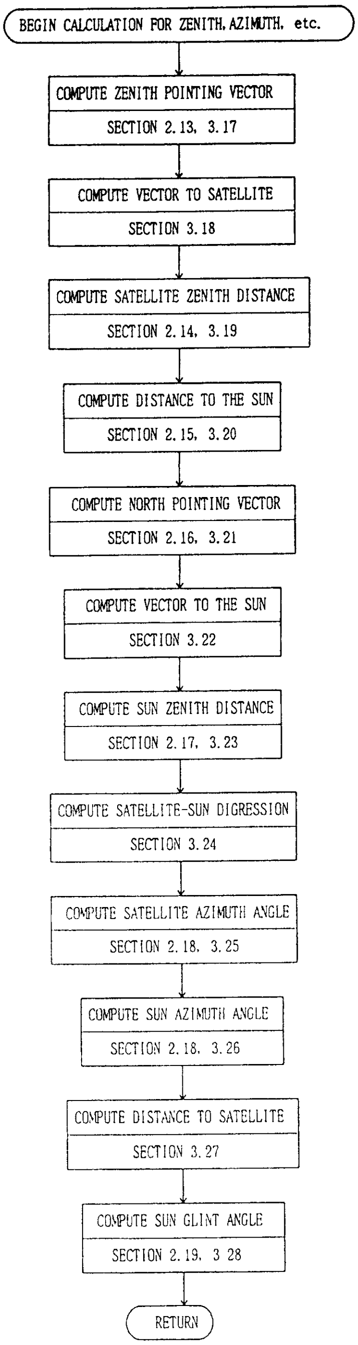


Fig. 1-4 Flow chart to calculate various transformation parameters.

VISSR image pixel which corresponds to a position on earth.

Another significant feature of 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 Stretched-VISSR (and HiRID: High Resolution Imager Data) that is broadcasted via satellite. This appendix is the latest version of "A Mapping Method for VISSR Data" (Kigawa: 1991, Meteorological Satellite Center Technical Note, No. 23).

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 informatiom necessary for digital image processing are the sun and satellite zenith deistances, 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.

2.1 Geodetic to Earth-fixed Transformation

The transformatiom 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. Interpolantion 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 tramsformed 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

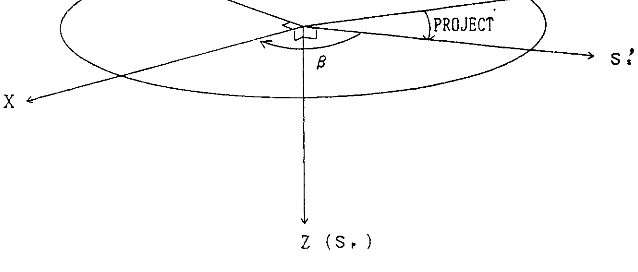


Fig. 2 Satellite angular momentum coordinates.

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 defined as

z-axis, $\mathbf{S}_p$:

$$\mathbf{S}_p = [B] \cdot [N_p] \cdot \begin{bmatrix} \sin\delta_r \\ -\cos\delta_r \cdot \sin\alpha_r \\ \cos\delta_r \cdot \cos\alpha_r \end{bmatrix} \tag{9}$$

X-aixs, $\mathbf{S}_x$:

$$\mathbf{S}_x = \frac{\mathbf{S}_p \times \mathbf{S}_s}{|\mathbf{S}_p \times \mathbf{S}_s|} \sin\beta + \frac{\mathbf{S}_p \times \mathbf{S}_s}{|\mathbf{S}_p \times \mathbf{S}_s|} \times \mathbf{S}_p \quad \cos\beta \tag{10}$$

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 augular momentum coordinates, and is expressed as

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

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

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

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

where

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

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

$$\left.\begin{aligned} X_e &= X + k u_x \\ Y_e &= Y + k u_y \\ Z_e &= Z + k u_z \end{aligned}\right\} \tag{27}$$

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 **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).

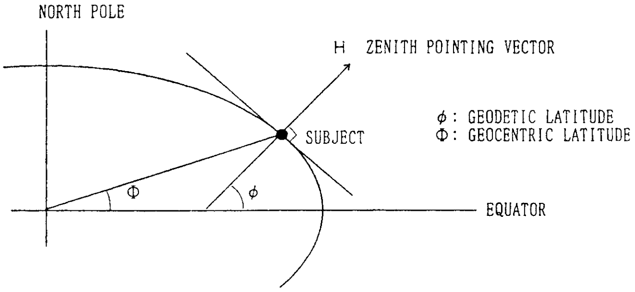


Fig. 3 Subject zenith pointing vector along the geodetic vertical.

2.14 Satellite Zenith Distance

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

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

where t_{IJ} is the scanning time represented in MJD, and R_{SUN} 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).

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

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

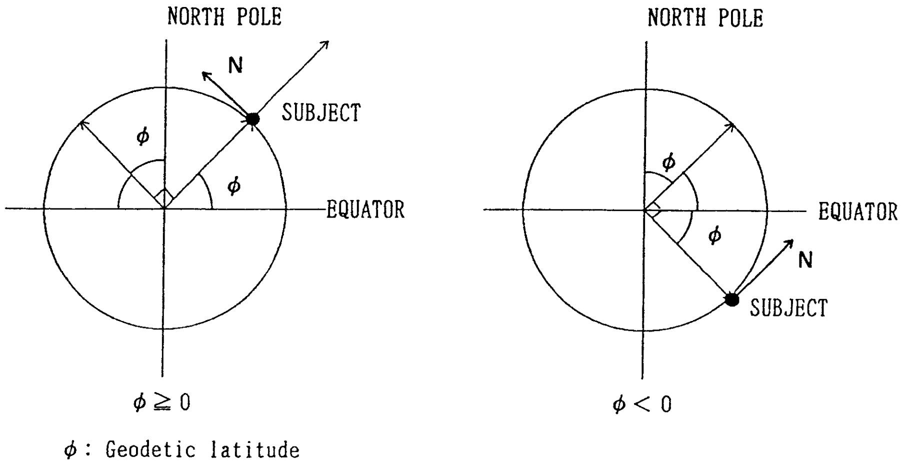


Fig. 4 Horizontal plane of vector that points north.

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 $\mathbf{V}_{SAT}$.

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

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

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

$$\mathbf{D} = \mathbf{B} \times \mathbf{C} \tag{40}$$

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

and

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, beging given by

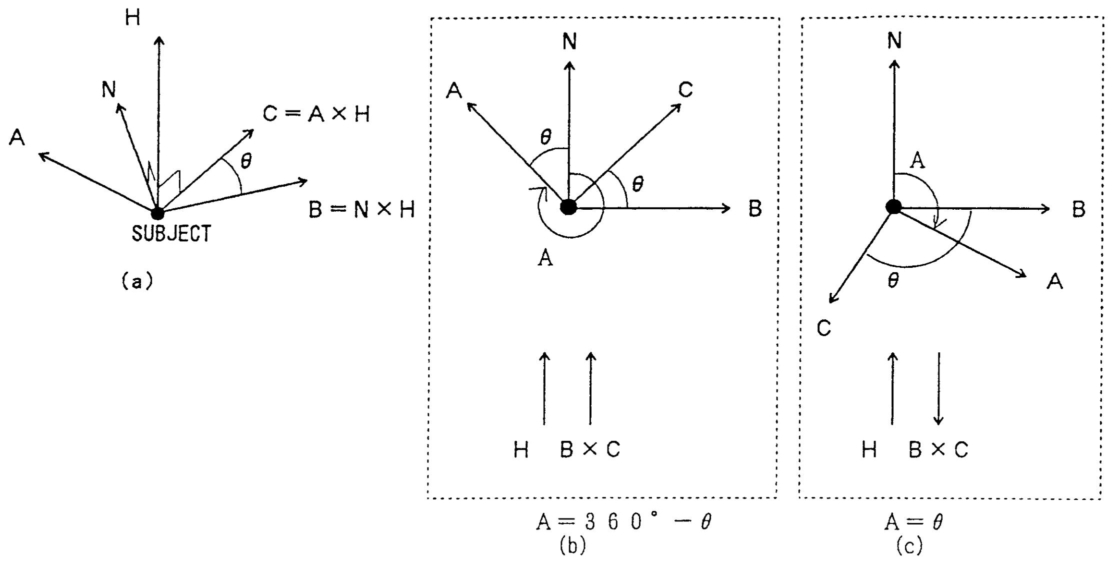
$$\theta_S = \cos^{-1} \frac{\mathbf{H} \cdot \mathbf{V}_{SUN}}{|\mathbf{H}|\,|\mathbf{V}_{SUN}|} \tag{42}$$

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

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

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 (and HiRID) data that is broadcasted via satellite.



(a)

A＝３６０°－θ
(b)

A＝θ
(c)

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 $360^\circ-\theta$

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.

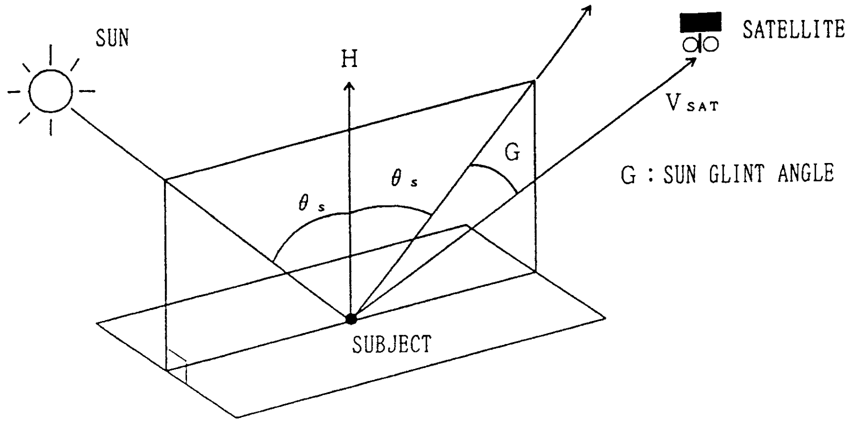


Fig. 6 Sun glint angle, i,e., the angle between the vector of the sun's rays reflected at the subject and the vector from the subject to the satellite.

SAMPLE PROGRAMS

```
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                                                                      C
C***********************************************************************
      PROGRAM  SV0000
C-----------------------------------------------------------------------
C     GMS-5 S-VISSR AND MTSAT HiRID NAVIGATION
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=801*5,2500*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                                       *EXAMPLE POSITION
      RLAT = 35.00
      RLON = 140.00
C                                       *GET LINE & PIXEL
      CALL  MGIVSR(1,RPVIS,RLVIS,RLON, RLAT, 0.0,RINF,DSCT,JR)
      CALL  MGIVSR(2,RPIR1,RLIR1,RLON, RLAT, 0.0,RINF,DSCT,JR)
      CALL  MGIVSR(3,RPIR2,RLIR2,RLON, RLAT, 0.0,RINF,DSCT,JR)
      CALL  MGIVSR(4,RPWV ,RLWV ,RLON, RLAT, 0.0,RINF,DSCT,JR)
C                                       *OUTPUT LINE & PIXEL
      WRITE(6,*)  'VISIBLE  LINE & PIXEL : ', RLVIS, RPVIS
      WRITE(6,*)  'IR1(IR4) LINE & PIXEL : ', RLIR1, RPIR1
      WRITE(6,*)  'IR2      LINE & PIXEL : ', RLIR2, RPIR2
      WRITE(6,*)  'WV(IR3)  LINE & PIXEL : ', RLWV , RPWV
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
```

GMS-5 S-VISSR AND MTSAT HiRID IMAGE NAVIGATION (1/5)

JAPAN METEOROLOGICAL AGENCY / METEOROLOGICAL SATELLITE CENTER 1996

```
          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, 8, COBAT( 11: 14), RESLIN(3),  R8DMY )
      CALL  SV0100( 4, 8, COBAT( 11: 14), RESLIN(4),  R8DMY )
      CALL  SV0100( 4,10, COBAT( 15: 18), RESELM(1),  R8DMY )
      CALL  SV0100( 4,10, COBAT( 19: 22), RESELM(2),  R8DMY )
      CALL  SV0100( 4,10, COBAT( 19: 22), RESELM(3),  R8DMY )
      CALL  SV0100( 4,10, COBAT( 19: 22), RESELM(4),  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(111:114), RLIC(3)  ,  R8DMY )
      CALL  SV0100( 4, 4, COBAT(115:118), RLIC(4)  ,  R8DMY )
      CALL  SV0100( 4, 4, COBAT( 31: 34), RELMFC(1),  R8DMY )
      CALL  SV0100( 4, 4, COBAT( 35: 38), RELMFC(2),  R8DMY )
      CALL  SV0100( 4, 4, COBAT(119:122), RELMFC(3),  R8DMY )
      CALL  SV0100( 4, 4, COBAT(123:126), RELMFC(4),  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( 43: 46), SENSSU(3),  R8DMY )
      CALL  SV0100( 4, 0, COBAT( 43: 46), SENSSU(4),  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( 51: 54), RLINE(3) ,  R8DMY )
      CALL  SV0100( 4, 0, COBAT( 51: 54), RLINE(4) ,  R8DMY )
      CALL  SV0100( 4, 0, COBAT( 55: 58), RELMNT(1),  R8DMY )
      CALL  SV0100( 4, 0, COBAT( 59: 62), RELMNT(2),  R8DMY )
      CALL  SV0100( 4, 0, COBAT( 59: 62), RELMNT(3),  R8DMY )
      CALL  SV0100( 4, 0, COBAT( 59: 62), RELMNT(4),  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  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  MTSAT
C                           1,-1  VIS   VIS    VIS
C                           2,-2  IR    IR1   IR1,IR4
C                           3,-3  --    IR2    IR2
C                           4,-4  --    WV     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,RIO,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)                                       [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
        RIO     = RFCL-ATAN(SIN(SNGL(DLAT))/(6.610689-COS(SNGL(DLAT))))
     .            /RSTEP
        RTIM    = DTIMS+DBLE(RIO/SENS/1440.)/DSPIN               [3.3]
C
  100   CONTINUE
        CALL  MGI100(RTIM,CDR,SAT,SP,SS,BETA)                    [3.4]
C-----------------------------------------------------------------------
        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-RIO).GE.EPS)  THEN                             [3.8]
          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                          [3.9]
        CALL  MGI100(RTIM,CDR,SAT,SP,SS,BETA)                    [3.10]
        CALL  MGI220(SP,SS,SW1)                                  [3.11]
        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)+
     .              STN1(2)*STN1(2))))                           [3.15]
        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) !!!!!!!!!!!!!!![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+0.985600D0*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
C
      CALL  MGI200(STN1,SL)                                   - [3.28]
      CALL  MGI230(SW2,SL,DSUNG)
      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
```