ROBERT B. FORREST
*School of Surveying**
University of New South Wales
Kensington, N.S.W., Australia

Simulation of Orbital Image-Sensor Geometry

A Fortran program is presented, the parameters are listed, and the algorithms are described for the simulation of the Landsat MSS.

ABSTRACT: *The essential dynamic aspects of orbital image-sensor geometry are modeled in a Fortran computer program presented and described in the paper. Parameters are listed for the Landsat* MSS, *and the program is sufficiently versatile that it can be modified to simulate other image-sensor geometrics and formats. The program is organized to provide Earth-surface coordinate outputs (latitude/longitude, local Cartesian) for any number of image coordinate inputs (pixel row/column numbers). Individual terrain heights may be specified for each image point of interest.*

INTRODUCTION

A FORTRAN computer simulation of orbital image-sensor geometry is presented. The program models satellite, Earth-rotation and sensor dynamic behavior and provides the Earth-surface positions for any number of specified image locations. The user specifies the image-center latitude, longitude, and terrain height. In the form listed, the parameters used are those for the Landsat MSS. A few changes allow the coverage and distortions of other satellite orbits and image sensors to be analyzed. Thus, the program may be of use to those interested in studying the geometry and positional aspects of image data from future satellite imaging systems (SPOT, MOS, Landsat D). The form listed has been useful in showing the true ground coverage of a single MSS image, as well as the effects of varying satellite attitude on the coverage.

The program was written during an analysis of the geometric corrections applied to the electron beam recorder (EBR) used at the National Aeronautics and Space Administration Goddard Space Flight Center (NASA/GSFC) to produce Landsat MSS film images. To evaluate the adequacy of the EBR corrections, some theoretical values were needed. A modified form of the program described here provided those values. The simulation gives a precision of about one meter on the ground, more than adequate for any presently planned satellite imaging programs. Concepts discussed by Widger (1966), Kratky (1971, 1974), Forrest (1974), and Puccinelli (1976) are incorporated. Readers unfamiliar with the Landsat orbit and the MSS sensor are referred to any of several related papers in this and other journals in the early 1970's.

* Through June, 1981.

MSS COORDINATE SYSTEM

Input image locations are specified in row-column pixel coordinates, r and c. As shown in Figure 1, r increases in the direction of Landsat motion, from north to south; and c increases in the direction of MSS scan sweep, from west to east. The simulation assigns integral row-column coordinate values to pixel centers. Each pixel extends from -0.5 to $+0.5$ unit in each direction from its center. For one entire framed MSS image (or *scene* as it is regrettably called), $0.5 \leq r \leq 2340.5$ and $0.5 \leq c \leq 3240.5$.

Most calculations for an image are referenced to the center. The simulation considers the image center pixel coordinates ($r = 1170.5, c = 1620.5$) to occur halfway between scan sweeps 195 and 196 of a framed Landsat image (Figure 1). This is a consequence of the even number of scan sweeps printed in a standard EBR framed MSS image. As Figure 2 shows, because there are six lines in each scan sweep, the image center normally is sensed slightly *before* the satellite passes over it. The satellite nadir at the moment the image center is sensed will vary, depending on the parameters applied in the simulation.

0099-1112/81/4708-1187$02.25/0

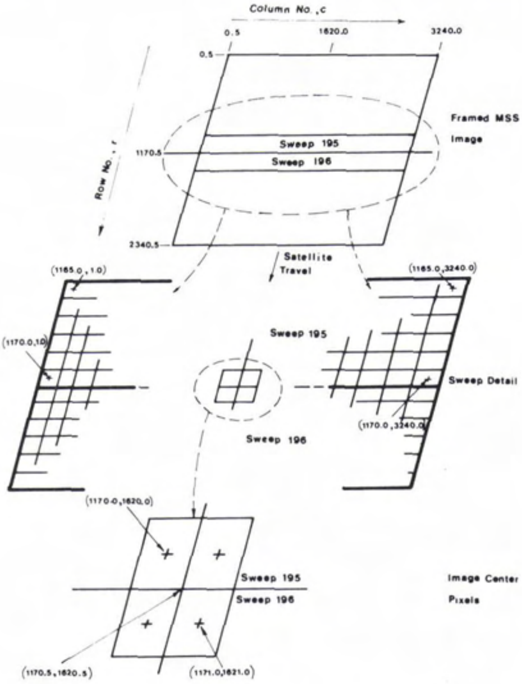

FIG. 1. Pixel coordinate system and details.

Figure 2 shows an ideal situation for two consecutive scan sweeps. Satellite attitude/altitude variations, MSS detector placement, and sweep-cycle time all act to modify the ideal geometry. Ground details at or near the fore or aft edges of a scan sweep may be imaged twice, once, or not at all.

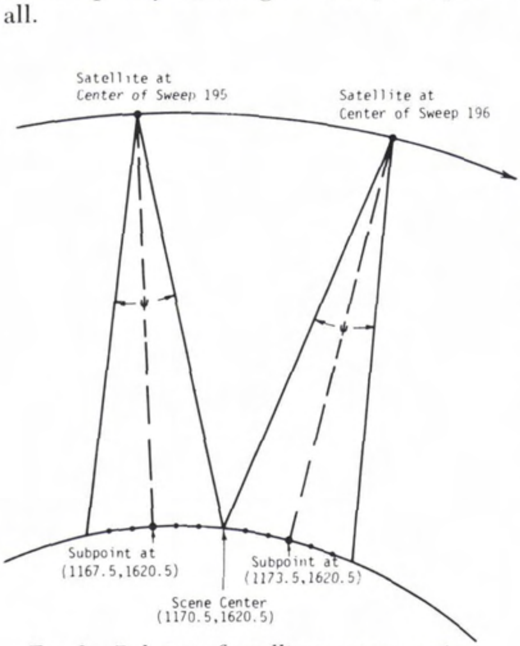

FIG. 2. Relation of satellite to scene center.

The following sensor- and format-related parameters are defined within the computer program. The values in brackets are those shown in the program listing for the Landsat MSS.

- t_{sw} = Time for a complete scanner sweep cycle, seconds (1/13.62).
- $\dot{p}$ = Active scan sweep-rate, pixels/second (100417.5).
- Q_0, Q_1, Q_2, Q_3 = Coefficients of the sweep-rate nonlinearity correction in pixel column units, modeled as a cubic polynomial of sensed pixel column coordinate (0, -0.01733, 1.6043×10^{-5}, -3.3011×10^{-9}).
- s = Scan sweeps per image (390).
- l = Lines per scan sweep (6).
- p = Total number of pixels in direction of scan for a single scan line (3240).
- η = Maximum angular coverage in direction of scan, radians (0.2).
- ψ = Angular coverage of a single scan sweep normal to the direction of scan, radians (0.000514).

EARTH ORBIT GEOMETRY

Figure 3 shows the orbit geometry at image-center time, t_0. In the figure, a sphere of radius equal to that of the satellite's assumed circular orbit is centered on the reference Earth ellipsoid, also assumed to be the center of mass for the Earth. Earth-centered coordinates X,Y,Z have plus X passing through the equator at 0 degrees longitude, and plus Z through the North Pole. Time is measured from the ascending node, the point at which an individual orbit customarily is considered to begin. Angle γ represents longitude of the descending node at time t_0; the fixed Landsat WRS orbital paths are not used. The spherical angle, i, is orbital inclination, about 99 degrees for Landsat's retrograde orbit. The azimuth of the orbital velocity vector at t_0 is $\alpha_0 = \beta_0 + \pi$, and geocentric latitude is Φ_0'.

The relationships between ellipsoidal and spherical parameters are shown in Figure 4, a meridional section at t_0. Altitude, H, is measured normal to the ellipsoid. The satellite velocity vector is normal to the radius of orbit, which in general is not parallel to the geodetic vertical.

The following orbital and ellipsoidal parameters are defined within the computer program, with the values used for the listing given in brackets:

- a = Semimajor axis of the reference Earth ellipsoid, in metres (6,378,165).
- e^2 = Square of ellipsoid eccentricity (0.0066935113).
- i = Orbital inclination of satellite, radians (1.72787596).

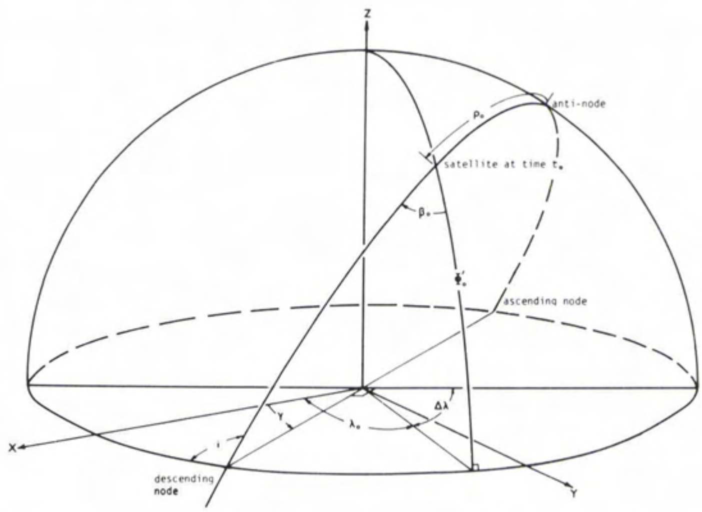

FIG. 3. Orbital geocentric sphere.

R = Radius of orbit, metres (7,285,600); the program can be used for slightly elliptical orbits, taking the effective geocentric distance at image center as R.

ω_s = Satellite angular velocity, radians per second (0.0010152871).

ω_e = Earth angular velocity, radians per second, including orbit precession effects (7.2722052×10^{-5}).

PROGRAM DESCRIPTION

The program as listed (Figure 5) was written for a minicomputer (Digital Equipment Corp. PDP 11/40) which requires the use of double precision throughout to obtain the desired significance. The program has been organized for ease of understanding, rather than efficient execution.

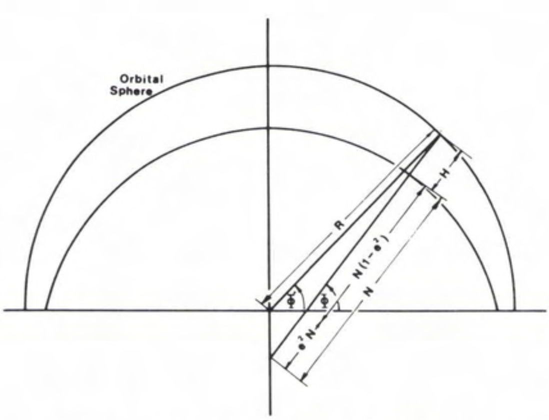

FIG. 4. Geocentric and geodetic reference.

INPUTS

For each exercise of the program, the user provides a data file which includes:

(1) Geodetic latitude and longitude of the image center, Φ_0 and λ_0, in decimal degrees.
(2) Image-center terrain height above sea level in metres, h_0.
(3) Sensor attitude angles ω_0, ϕ_0, κ_0 at the time image center is sensed, in decimal degrees. Zero values correspond to alignment of the sensor coordinate axes (x,y,z in Figure 6) with the local vertical and satellite heading. (For those unfamiliar with this photogrammetric expression for attitude, the sensor tilt angles are Eulerian, defined in the order ω—identical to roll, then ϕ—analogous to pitch, and finally κ—analogous to yaw. The standard sign convention for positive ω, ϕ, and κ corresponds to aircraft left wing up, nose down, and nose left, respectively).
(4) Linear attitude changes with time, $\dot{\omega}$, $\dot{\phi}$, $\dot{\kappa}$, in decimal degrees per second.
(5) A flag which shows whether the image-center terrain height is to be used for each image point processed (flag = 0), or whether heights will be provided for each separate pixel considered (flag = 1).
(6) A list of image coordinates for which Earth-surface locations are desired, as pixel values r and c (Figure 1). If the flag in (5) above is set, terrain heights above sea level in metres also must be included for each image point.

OUTPUTS

The program copies the input image-center information on the output data file. Then, for each image point specified, the program provides:

```
C  RCLL--EARTH-SURFACE COORDS. FROM ORBITAL SENSOR PIXELS FOR
C      SPECIFIED IMAGE CENTER (TERRAIN HEIGHTS OPTIONAL)
      IMPLICIT REAL*8(A-H),(O-Z)
      COMMON A1,E2,XINC,R,WS,WE,TPERSW,PDOT,
     +Q0,Q1,Q2,Q3,SWIM,PPL,ETA,PSI,LPS,HALFPI
      DIMENSION OEZ(3),DOE(3),XE(3),XFC(3),XCAP(3),D(3,3)
      CALL SETFIL(1,'MSS1.DAT',IERR,'DK:',0)
      CALL SETFIL(2,'MSS2.DAT',IERR,'DK:',0)
      DTOR=.0174532925
      HALFPI=90.*DTOR
C*****IMAGE SENSOR/FORMAT PARAMETERS
      TPERSW=1./13.62
      SWIM=390.
      LPS=6
      PPL=3240.
      ETA=0.2
      PSI=.000514
      PDOT=100417.54
      Q0=0.
      Q1=-0.01733
      Q2=1.6043D-05
      Q3=-3.3011D-09
C*****EARTH ELLIPSOID AND EFFECTIVE EARTH ANG. VELOCITY
      A1=6378165.
      E2=.0066935113
      WE=7.2722052D-05
C*****CIRCULAR ORBIT VALUES: RADIUS, INCL., ANG. VELOCITY
      R=7285600.
      XINC=99.*DTOR
      WS=.0010152871
C*****IF INH=1,SEPARATE TERRAIN HTS. FOR EACH PIXEL;
C*****IF INH=0,ALL PIXELS ARE AT IMAGE-CENTER TERRAIN HT (HH).
C*****OEZ(I) AND DOE(I) ARE IN ORDER: OMEGA,PHI,KAPPA
      READ(1,203) PHZ,XLZ,HH,(OEZ(I),I=1,3),(DOE(J),J=1,3),INH
  203 FORMAT(3D15.8/3D15.8/3D15.8/I2)
      WRITE (2,200)
  200 FORMAT (' CENTER:ROW,COL/LAT,LONG,HT/OM,PHI,KAPPA/RATES')
C*****IMAGE-CENTER PIXEL COORDS AS ROW/COL
      ROW=SWIM*LPS/2.+0.5
      COL=PPL/2.+0.5
      WRITE (2,204) ROW,COL
      WRITE (2,203) PHZ,XLZ,HH,(OEZ(I),I=1,3),(DOE(J),J=1,3)
C*****ESTABLISH ROTATION FROM EARTH TO LOCAL COORDS.
      PHZ=PHZ*DTOR
      XLZ=XLZ*DTOR
      CALL MSSA(PHZ,XLZ,TZ,GAMMA,BETA,H)
      CALL ROT2(D,BETA,PHZ,XLZ)
C*****MAKE CENTER ON IMAGE MATCH PHZ,XLZ ON GROUND
      PHS=PHZ
      XLS=XLZ
      DO 1 K=1,3
      CALL MSSA(PHS,XLS,TZ,GAMMA,BETA,H)
      CALL MSSB(ROW,COL,TZ,GAMMA,OEZ,DOE,PH,XL,XFC,HH)
      PHS=PHS+PHZ-PH
    1 XLS=XLS+XLZ-XL
C*****PROCESS EACH PIXEL IN FILE
      WRITE (2,201)
  201 FORMAT('  INDIV. PIXELS:'/'   ROW',5X,'COL',9X,
     + 'LAT',8X,'LONG',5X,'HT',7X,'X',7X,'Y',7X,'Z')
    2 IF(INH.EQ.0) GO TO 5
      READ (1,204,END=99) ROW,COL,HH
      GO TO 6
    5 READ (1,204,END=99) ROW,COL
  204 FORMAT(2F7.1,F6.0)
    6 CALL MSSB(ROW,COL,TZ,GAMMA,OEZ,DOE,PH,XL,XE,HH)
      DO 3 I=1,3
      XCAP(I)=0.
      DO 3 J=1,3
    3 XCAP(I)=XCAP(I)+D(I,J)*(XE(J)-XFC(J))
      PH=PH/DTOR
      XL=XL/DTOR
      WRITE (2,202) ROW,COL,PH,XL,HH,(XCAP(I),I=1,3)
  202 FORMAT(2F8.1,2X,2F11.6,F6.0,2X,3F8.0)
      GO TO 2
   99 END


      SUBROUTINE MSSA(PHZ,XLZ,TZ,GAMMA,BETA,H)
C***LAT/LONG OF IMAGE CENTER IN; TIME OF IMAGING, LONG. OF
C   DESC. NODE, AND (AZIMUTH-180 DEGREES) OUT.
      IMPLICIT DOUBLE PRECISION (A-H),(O-Z)
      COMMON A1,E2,XINC,R,WS,WE,TPERSW,PDOT,
     +Q0,Q1,Q2,Q3,SWIM,PPL,ETA,PSI,LPS,HALFPI
      XN=A1/DSQRT(1.-E2*DSIN(PHZ)*DSIN(PHZ))
      SP=XN*E2*DSIN(PHZ)*DCOS(PHZ)/R
      PHPZ=PHZ-DATAN2(SP,DSQRT(1.-SP*SP))
      H=R*DCOS(PHPZ)/DCOS(PHZ)-XN
      CR=DSIN(PHPZ)/DSIN(XINC)
      IF(DABS(CR).GT.1.) CR=DABS(CR)/CR
      SR=DSQRT(1.-CR*CR)
      TZ=(DATAN2(SR,CR)+HALFPI)/WS-(TPERSW/2.)
      DL=-DATAN2(SR,CR*DCOS(XINC))
      GAMMA=XLZ-HALFPI+DL
      BETA=-DATAN2(DCOS(XINC),DSIN(XINC)*SR)
      RETURN
      END


      SUBROUTINE MSSB(ROW,COL,TZ,GAMMA,OEZ,DOE,PH,XL,XE,HH)
C***PIXEL ROW/COL IN; GEODETIC LAT,LONG,X,Y,Z OUT
      IMPLICIT DOUBLE PRECISION (A-H),(O-Z)
      COMMON A1,E2,XINC,R,WS,WE,TPERSW,PDOT,
     + Q0,Q1,Q2,Q3,SWIM,PPL,ETA,PSI,LPS,HALFPI
      DIMENSION XS(3),OEZ(3),DOE(3),A(3,3),B(3,3)
      DIMENSION XPP(3),XP(3),OE(3),X(3),XE(3)
      SI=DSIN(XINC)
      CI=DCOS(XINC)
      N=(ROW+DABS(ROW)*(LPS-0.501)/ROW)/LPS
      COLP=Q0+COL*(1.+Q1+COL*(Q2+COL*Q3))
C****** MOMENT AT WHICH PIXEL WAS IMAGED
      T=TZ+TPERSW*(N-SWIM/2.)+(COLP-PPL/2.-0.5)/PDOT
      RHO=WS*T-HALFPI
      SP=DCOS(RHO)*SI
      PHP=DATAN2(SP,DSQRT(1.-SP*SP))
C***GEODETIC LATITUDE OF SATELLITE BY ITERATION
      PHS=PHP
      DO 2 I=1,3
      XN=A1/DSQRT(1.-E2*DSIN(PHS)*DSIN(PHS))
      SP=XN*E2*DSIN(PHS)*DCOS(PHS)/R
    2 PHS=PHP+DATAN2(SP,DSQRT(1.-SP*SP))
C***** SATELLITE ALTITUDE,LONGITUDE,HEADING
      H=R*DCOS(PHP)/DCOS(PHS)-XN
      DL=-DATAN2(DSIN(RHO),DCOS(RHO)*CI)
      XLS=GAMMA+HALFPI-DL-WE*(T-TZ)
      BETA=-DATAN2(CI,SI*DSIN(RHO))
C***ROW/COLP TO SENSOR COORDINATE VECTOR
      TH=ETA*(COLP-PPL/2.-0.5)/PPL
      X(1)=PSI*(ROW-N*LPS+LPS/2.-0.5)/LPS
      X(2)=DSIN(TH)
      X(3)=-DCOS(TH)
C***** ROTATE AND SCALE TO LOCAL-VERTICAL UNIT VECTOR
      DO 1 I=1,3
    1 OE(I)=(OEZ(I)+DOE(I)*(T-TZ))*.0174532925
      CALL ROTAT (A,OE)
      SC=1./DSQRT(1.+X(1)*X(1))
      DO 3 I=1,3
      XP(I)=0.
      DO 3 J=1,3
    3 XP(I)=XP(I)+A(J,I)*SC*X(J)
C***ROTATION FROM EARTH TO LOCAL-VERTICAL COORDS
      CALL ROT2(B,BETA,PHS,XLS)
C***IMAGE/SATELLITE FROM LOCAL-VERTICAL TO EARTH COORDS.
      DO 4 I=1,3
      XS(I)=B(3,I)*(XN+H)
      XPP(I)=0.
      DO 4 J=1,3
    4 XPP(I)=XPP(I)+B(J,I)*XP(J)
C***SCALE IMAGE VECTOR TO TERRAIN HT. ABOVE ELLIPSOID
      V2=(A1+HH)*(A1+HH)*1.D-12
      UU=DSQRT(A1*A1*(1.-E2))+HH
      U2=UU*UU*1.D-12
      XS(3)=XS(3)-DSIN(PHS)*XN*E2
      AK=U2*(XPP(1)*XPP(1)+XPP(2)*XPP(2))+V2*XPP(3)*XPP(3)
      BK=U2*(XPP(1)*XS(1)+XPP(2)*XS(2))+V2*XS(3)*XPP(3)
      CK=U2*(XS(1)*XS(1)+XS(2)*XS(2))+V2*(XS(3)*XS(3)-U2*1.D12)
      U=(-BK-DSQRT(BK*BK-AK*CK))/AK
C****** EARTH-CENTERED COORDS., LAT/LONG
      XE(1)=XS(1)+U*XPP(1)
      XE(2)=XS(2)+U*XPP(2)
      XE(3)=XS(3)+U*XPP(3)
      S=DSQRT(XE(1)*XE(1)+XE(2)*XE(2))
      PH=DATAN2(XE(3)*(A1+HH),S*(A1*(1.-E2)+HH))
      XL=DATAN2(XE(2),XE(1))
      RETURN
      END


      SUBROUTINE ROTAT(A,OE)
C**** 3-DIMENSIONAL ROTATION MATRIX 'A'
      IMPLICIT REAL*8(A-H),(O-Z)
      DIMENSION A(3,3),OE(3)
      SW=DSIN(OE(1))
      CW=DCOS(OE(1))
      SP=DSIN(OE(2))
      CP=DCOS(OE(2))
      SK=DSIN(OE(3))
      CK=DCOS(OE(3))
      A(1,1)=CP*CK
      A(2,1)=-CP*SK
      A(3,1)=SP
      A1=SW*SP
      A(3,2)=-SW*CP
      A(3,3)=CW*CP
      A(1,2)=CW*SK+A1*CK
      A(2,2)=CW*CK-A1*SK
      A1=CW*SP
      A(1,3)=SW*SK-A1*CK
      A(2,3)=SW*CK+A1*SK
      RETURN
      END


      SUBROUTINE ROT2(A,X,Y,Z)
C****** ROTATION MATRIX FOR LAT/LONG/AZIMUTH
      IMPLICIT DOUBLE PRECISION (A-H),(O-Z)
      DIMENSION A(3,3)
      SX=DSIN(X)
      CX=DCOS(X)
      SY=DSIN(Y)
      CY=DCOS(Y)
      SZ=DSIN(Z)
      CZ=DCOS(Z)
      A(1,1)=SX*SZ+CX*SY*CZ
      A(1,2)=-SX*CZ+CX*SY*SZ
      A(1,3)=-CX*CY
      A(2,1)=-CX*SZ+SX*SY*CZ
      A(2,2)=CX*CZ+SX*SY*SZ
      A(2,3)=-SX*CY
      A(3,1)=CY*CZ
      A(3,2)=CY*SZ
      A(3,3)=SY
      RETURN
      END
```

FIG. 5. Fortran program listing.

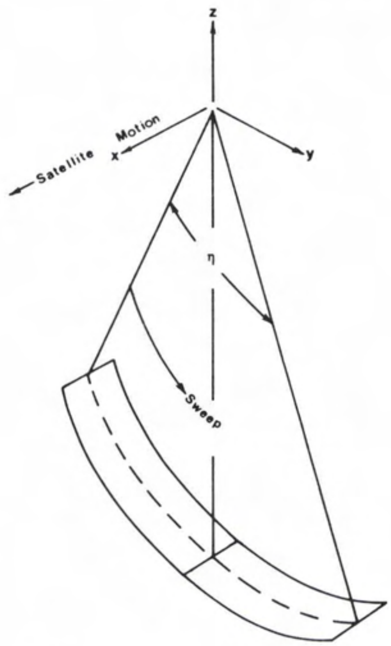

FIG. 6. Sensor coordinate system.

(1) Geodetic latitude and longitude in decimal degrees, and
(2) Local Cartesian coordinates $\hat{X}$, $\hat{Y}$, and $\hat{Z}$ in metres. Within the reference plane, the $\hat{X}$, $\hat{Y}$, $\hat{Z}$ origin is at image center, with positive $\hat{X}$ in the direction of satellite heading and $\hat{Z}$ positive upward.

LIMITATIONS

An optical-mechanical west-to-east scanning sensor is modeled in the program as listed. With a few changes a fixed linear array can be simulated. The internal MSS imaging geometry is not modeled in great detail, except for mirror sweep rate. MSS detector offsets or placement errors are not included, nor are the so-called commutation effects caused by the sampling sequence. Such effects generally are seen as simple image-space pixel translations. The listed program assumes a circular or near-circular orbit, as stated earlier. It is not suitable for orbit inclination of exactly zero, or for images centered on the night side of an orbit.

ORGANIZATION

The main program, RCLL (Row/Column to Lat/Long), first reads the image center data from the input file and sets up the rotation from X,Y,Z to $\hat{X},\hat{Y},\hat{Z}$. Next, subroutines MSSA and MSSB are called iteratively to obtain the fundamental image-center quantities t_0,γ, and β_0 (Figure 3). RCLL then reads and processes each input image point in turn. Subroutine MSSB returns the latitude, Φ, and longitude, λ, for the specified pixel coordinates and terrain elevation, and also provides the X,Y,Z coordinates. RCLL develops $\hat{X},\hat{Y},\hat{Z}$ and writes the final ground coordinate values to the output data file.

ANALYSIS

SUBROUTINE MSSA

From the given image-center coordinates, Φ_0 and λ_0, this routine develops three fundamental orbit-related quantities:

t_0, the time in seconds at which Φ_0 and λ_0 were actually sensed,
γ, longitude of the descending node at t_0, and
α_0, azimuth of the orbital path at t_0.

First, the geocentric latitude, Φ_0', is found which corresponds to Φ_0: i.e.,

$$\Phi_0' = \Phi_0 - \sin^{-1}(Ne^2 \sin \Phi_0 \cos\Phi_0/R)$$

where

$$N = a/(1 - e^2 \sin^2\Phi_0)^{1/2}.$$

Then, satellite altitude above the ellipsoid is

$$H = R \cos \Phi_0'/\cos \Phi_0 - N.$$

Next, the orbital arc distance, ρ_0, from the anti-node (Figure 2) is found:

$$\rho_0 = \cos^{-1}(\sin \Phi_0'/\sin i)$$

and then

$$t_0 = (\rho_0 + \pi/2)/\omega_s - t_{sw}/2, \quad (1)$$

may be calculated directly. The $t_{sw}/2$ term results from the definition of t_0 as the time image center is sensed, not the time the satellite is passing over image center (Figure 2).

Continuing, the longitude difference at t_0 between image center and the anti-node is

$$\Delta\lambda = \pi - \tan^{-1}(\tan \rho_0/\cos i),$$

so that

$$\gamma = \lambda_0 - \pi/2 + \Delta\lambda. \quad (2)$$

Finally, the orbital azimuth at t_0 is $\alpha_0 = \beta_0 + \pi$, where

$$\beta_0 = - \tan^{-1}(\cot i/\sin \rho_0). \quad (3)$$

SUBROUTINE MSSB

This routine calculates latitude, Φ, and longitude, λ, of a point on the Earth at specified pixel coordinates, r and c, and terrain height, h. Earth-centered coordinates X,Y,Z (Figure 1) also are returned. The image-center values for t_0 and γ are required, as well as parameters for ellipsoid, satellite orbit, and image sensor and format. The order of computation is

(1) Time, t, at which the pixel r,c was imaged;
(2) Attitude angles ω, ϕ, and κ for the satellite at t;

(3) Satellite latitude, longitude, and altitude Φ_s, λ_s, and H at t;
(4) Image-sensor coordinates x,y,z rotated to a local vertical reference;
(5) Rotation from local-vertical to Earth-centered coordinates; and
(6) Calculation of Φ, λ.

Each of these is briefly discussed below.

Time. The program uses a "continuous pixel" in the direction of scan sweep: if $c = 1816.3$, the time is found which corresponds exactly to 1816.3 pixel-times from start of scan sweep. This convention is admittedly somewhat artificial, since pixel sampling only takes place at integral pixel times, 1816.0 in this example. If the distinction is important, input c values should be limited to integer values.

The sweep number relative to the first scan sweep in the image is

$$n = \left[r + |r|(l - 0.501)/r\right]/l.$$

Sweep-rate nonlinearity is removed next:

$$c' = c + Q_0 + Q_1c + Q_2c^2 + Q_3c^3,$$

where c' is the equivalent constant-sweep-rate column coordinate.

Then

$$t = t_0 + t_{sw}(n - s/2) + (c' - p/2 - 0.5)/\dot{p}$$

where the three terms represent time for image center, time for whole scan sweeps, and time within sweep n.

Attitude. Satellite attitude at t is calculated using ω_0, ϕ_0, κ_0 and $\dot{\omega}$, $\dot{\phi}$, $\dot{\kappa}$. Subroutine ROTAT returns a three-dimensional rotation matrix, **A**, which indicates the alignment of the image sensor Cartesian coordinates x,y,z with respect to the local vertical and satellite heading.

Satellite Location. Geocentric satellite latitude at t is

$$\Phi'_s = sin^{-1}(\cos \rho_s \sin i),$$

where

$$\rho_s = \text{orbital arc distance from anti-node} = \omega_s t - \pi/2.$$

An iterative technique is used to develop geodetic latitude of the satellite, Φ_s. Initial approximation $\bar{\Phi}_1 = \Phi'_s$ is entered in

$$\bar{\Phi}_{i+1} = \Phi'_s + \sin^{-1}(N_i e^2 \sin \bar{\Phi}_i \cos \bar{\Phi}_i/R),$$

with

$$N_i = a/(1 - e^2 \sin^2 \bar{\Phi}_i)^{1/2},$$

and the determination repeated twice, with $\bar{\Phi}_3$ taken as Φ_s. Then satellite altitude is

$$H = R \cos \Phi_s'/\cos \Phi_s - N.$$

Satellite longitude at t is

$$\lambda_s = \gamma - \pi/2 + \tan^{-1}(\tan \rho_s/\cos i) - \omega_e(t - t_0),$$

in which the last term adjusts for Earth rotation between times t and t_0. Finally, orbital heading, β_s, at t is found using the same form as Equation 3.

Sensor Coordinate Vector. The simulation converts pixel coordinates r,c' to right-hand Cartesian coordinates x,y,z (Figure 6). The x axis coincides with the scan sweep axis, positive in the forward satellite direction. The z axis coincides with the scanner's optical axis at sweep center and is positive upward. Using the scanner's field-of-view parameters ψ and η,

$$x = \psi(r - nl + l/2 - 0.5)/l,$$
$$y = \sin\theta, \text{ and}$$
$$z = -\cos\theta,$$

where

$$\theta = \eta(c' - p/2 - 0.5)/p.$$

The sensor-coordinate vector (x,y,z) then is transformed to a satellite-vertical unit vector by

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \mathbf{A}^T \frac{1}{(1 + x^2)^{1/2}} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Ground Coordinate Rotation. The values for Φ_s, λ_S, and β_s at t can be used to define the orientation of the local vertical vector (x',y',z') to the Earth-centered X,Y,Z coordinates. Subroutine ROT2 returns the rotation matrix, $\boldsymbol{B}$, from which

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \boldsymbol{B}^T \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

Location of the Point. Satellite coordinates at t can be defined as $(0,0, N + H)$ in a local-vertical coordinate system aligned with the x',y',z' axes, but with origin at the curvature center of the prime vertical, N, in Figure 4. After rotation through $\boldsymbol{B}^T$ and an offset in Z, the Earth-centered coordinates of the satellite at t are

$$X_s = b_{31}(N + H),$$
$$Y_s = b_{32}(N + H), \text{ and}$$
$$Z_s = b_{33}(N + H) - N e^2 \sin \Phi_s.$$

To find the place where the extended (x'',y'',z'') vector pierces the ellipsoid at terrain height h, a modified version of Puccinelli's (1976) method is used. A parametric form for the vector from ellipsoid center to the desired point would be

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = (a + h) \begin{bmatrix} \cos\lambda & \cos\bar{\Phi} \\ \sin\lambda & \cos\bar{\Phi} \\ & \sin\bar{\Phi} \end{bmatrix} \quad (4)$$

where $\bar{\Phi}$ is not in general equal to Φ. A second form is

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_s + U \begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} \quad (5)$$

where U is an unknown scale factor which must be applied to the unit vector (x'',y'',z). By equating the two sets of expressions (Equations 4 and 5), three equations are obtained in λ, $\hat{\Phi}$, and U. Eliminating the two angles, a quadratic in U is obtained:

$$U^2u^2\left[(x''^2 + y''^2) + v^2z''^2\right] + 2U\left[u^2(X_sx'' + Y_sy'') + v^2Z_sz''\right] + u^2(X_s^2 + Y_s^2) + v^2(Z_s^2 - u^2) = 0,$$

where

$$u = a(1 - e^2)^{1/2} + h, \text{ and}$$
$$v = a + h.$$

The quadratic is solved for the smaller root, and the final Earth-centered coordinates are found using Equation 5. Finally, latitude and longitude are calculated:

$$S = (X^2 + Y^2)^{1/2},$$
$$\Phi = \tan^{-1}\left[(Z/S)\,(a + h)/(a(1 - e^2) + h)\right],$$
$$\lambda = \tan^{-1}(Y/X)$$

Comments

The program has proven to be of benefit in establishing the actual shape on the Earth for a single MSS scene. The details of a typical shape are not always obvious. For example, Landsat heading is due west at the anti-node and about nine degrees west of south at the equator. Thus, one might expect the ground coverage for a single northern-hemisphere scene to be slightly compressed on the east side and expanded on the west side from its nominal dimensions. The simulation shows that the reverse effect takes place, as the Earth rotation and orbit inclination effects override the azimuth change effect.

To see the ground coverage of a linear array image-sensor with the same orbit and image format as Landsat, make the following parameter changes: $t_{sw} = 1/81.72$, $s = 2340$, $l = 1$, $\psi = 8.57 \times 10^{-5}$, $\dot{p} = 10^{20}$, and $Q_0 = Q_1 = Q_2 = Q_3 = 0$.

Of course, the program as listed addresses only half the general positioning problem. Often of more interest is the inverse transformation, i.e., calculating the image location which corresponds to a given Earth-surface position. Because of the critical nature of time in the calculations, it has proven convenient to solve such problems iteratively, using a modified portion of program RCLL as a subprogram. Some arbitrary pixel coordinates are used as initial estimates, and the resulting Earth-surface positions calculated by RCLL are used to improve the estimates.

Acknowledgments

Dr. A. P. Colvocoresses of the U.S. Geological Survey and Mr. G. J. Grebowsky of NASA/GSFC provided valuable information, cooperation, and stimulating discussions in connection with the work described here. My thanks also to the anonymous reviewer who suggested simplified forms for two of the equations. Much of the work described here was performed during a temporary research appointment with the EROS Program Coordinator, Topographic Division, U.S. Geological Survey, Reston, Virginia.

References

Forrest, R. B., 1974. Geometric Correction of ERTS-1 MSS Images. *Proceedings, International Society of Photogrammetry Commission III Symposium*, Stuttgart, September 1974, ISBN 3 7696 85199.

Kratky, V., 1971. Precision Processing of ERTS Imagery. *Proceedings of the American Society of Photogrammetry Fall Convention*, 1971, pp. 481-514.

———, 1974. Cartographic Accuracy of ERTS. *Photogrammetric Engineering*, 40:2, pp. 203-212.

Puccinelli, E. F., 1976. Ground Location of Satellite Scanner Data. *Photogrammetric Engineering and Remote Sensing*, 42:4, pp. 537-543.

Widger, W. K., Jr., 1966. Orbits, Altitudes, Viewing Geometry, Coverage and Resolution Pertinent to Satellite Observations of the Earth and its Atmosphere. *Proceedings, Fourth Remote Sensing Symposium*, University of Michigan Institute of Science and Technology, Ann Arbor, pp. 489-537.

(Received 28 June 1979; revised and accepted 9 December 1980)

Bibliography Available

The *Bibliography of Remote Sensing in Forestry 1950-1978* has recently been completed by its authors, Brian J. Myers and Ian E. Craig. Its purpose is to provide ready reference to the voluminous literature documenting the use of remote sensing in forestry. The authors do not claim to have included all possible articles on the subject, but do hope that it is reasonably comprehensive within its terms of reference. The sources from which these articles have been drawn were restricted to material which would normally be readily available to the user through a forestrý library. The sources include most English language forestry, photogrammetric, and remote sensing journals and the proceedings of regular relevant symposia, but not monographs, foreign language journals, or reports and publications of limited circulation.

Copies of the *Bibliography* may be obtained free of charge from

Mr. Brian J. Myers
Division of Forest Research
CSIRO
P.O. Box 4008
Canberra, A.C.T.
2600 Australia