

JET PROPULSION LABORATORY

INTEROFFICE MEMORANDUM

314.8-391

25 October 1982

TO: Bob Davis
FROM: John Callahan JJC
SUBJECT: Galileo Interactive Picture Planning Program (GIP)

This memo documents for the first time the work completed on the GIP program. GIP is the first attempt to incorporate dynamic interactive control into Automated Picture Planning.

JC:ac

Distribution

P. Breckheimer
J. Campbell
T. Duxbury
R. Holzman
F. Jordan
W. Kirhofer
J. McDanell
J. Miller
L. Miller
F. Nicholson
W. Owen
D. Roth
S. Synnott
R. Weidner
L. Wood

I. Introduction

GIP is an interactive program developed at the JPL Computer Graphics Lab and runs on an Evans and Sutherland Picture System 2 graphics system hosted by a PDP11/55 minicomputer. The program is an aid in the selection of frames taken for optical navigation purposes, specifically for the Galileo mission. The frames should contain a satellite against a star background at favorable geometries so that camera pointing and spacecraft trajectory may be accurately solved for.

Currently picture planning starts on the Univac 1100/81 computers with programs VUGEOM written by Bill Owen, Jr. (Ref. 1) and PICSEL written by Bob Davis (Ref. 2). VUGEOM computes viewing geometry at fixed time intervals, say 1 hour, and a figure of merit (FOM). PICSEL then takes this information, considers all relevant factors, computes a new FOM, and makes final optical navigation picture selections. An analyst monitors the process working with printout and Calcomp plots.

GIP was written in an effort to help the analyst by (1) providing rapid turnaround, (2) showing data at continuous intervals and (3) giving a dynamic visual understanding of the data. GIP computes viewing geometry like VUGEOM but instead of printout and static plots, GIP drives a CRT picture display. This display is rapidly refreshed providing a movie effect capability. Many different scenes from the spacecraft cameras, the Earth, or an arbitrary external observer may be viewed. The user may interactively center the scene on Jupiter or the spacecraft, or any one of the Galilean satellites. The field of view size may be varied along with right ascension (RA) and declination (DC) pointing angles. Stars (SAO catalog) are displayed in correct magnitude vs. luminosity ratio and cover 98% of the sky down to 10th magnitude. The remaining 2% is covered down to at least 8.9th magnitude. Over 250,000 stars are accessible. The user may label all bodies and stars; important variables such as RA and DC are always displayed.

At present PICSEL output is transferred to the PDP 11/55 by hand and analyzed with the GIP program. In the future VUGEOM, PICSEL and GIP may all be on the same machine. The analyst could then run the programs in order, do his analyses, perhaps rerun VUGEOM and PICSEL, and produce

final picture selections. He would be able to investigate intermediate time points. Since encounter trajectories use only the encounter satellite for optical navigation GIP lends itself well to the problem. The analyst may simply observe the satellite centered in the spacecraft camera as the scene changes with time. GIP will automatically stop at PICSEL selection time points. In addition to this the Picture System has the capability to instantly display VUGEOM/PICSEL plots which at present take hours to produce by Calcomp plotter.

At present GIP is fully operational between 2 January 1988 and 2 January 1992. The user need only supply the start time (calendar date, E.T.), the Jupiter centered, 1950.0 EME state of the spacecraft (Rm, km/sec), and the VUGEOM/PICSEL picture selection time points (calendar date, E.T.). Spacecraft motion is integrated as the program executes. Therefore large time steps are not permitted, and the user should not move time back and forth by large amounts several times. Instead, it is very easy to restart the program. GIP pointing differences from VUGEOM rarely exceed 0.04° .

In the future GIP may be incorporated in the overall Automated Picture Planning Plans (Ref. 3 and 4). This may involve moving GIP to another machine and modifying the program to interface well with VUGEOM and PICSEL. Spacecraft motion should be read from a file such as a PFILE. However spacecraft trajectory requirements are not great since a 1,000 km error at a range of 1,000,000 km represents only a $.06^{\circ}$ field of view error. Bill Owen's simple integrator JUPROP (Ref. 5) is much more accurate than this and could be used to write a file similar to a PFILE.

II. Input/Output

II.1 Input

II.1.1 IG.DAT

IG.DAT is a Fortran binary file created by program IG (initialize GIP). The file contains start time, the states of the Jupiter and Earth barycenters WRT the sun center, the state of the spacecraft WRT Jupiter's center, and the number, times, and satellite to be pointed at for VUGEOM/PICSEL picture choices. IG reads the planetary ephemeris file PLANET.EPH which is DE114.

II.1.2 STAR.CAT

STAR.CAT is a direct access binary file of 7776 blocks (1 block = 512 bytes) containing over 250,000 stars (practically the entire SAO catalog). 98% of the sky is covered to 10th magnitude with the remaining 2% covered to at least 8.9th magnitude. The file was created from tape by program ASFM5 (All Stars File Make -5° field) and is composed of 2592 records. Each record is 3 blocks long and covers a $5^{\circ} \times 5^{\circ}$ field of sky. Contained within a record are the number of stars in the field (up to 255) and the right ascension, declination, magnitude and spectral class for each star. The epoch of the stars is 1 January 1991 referenced to EME 1950.0.

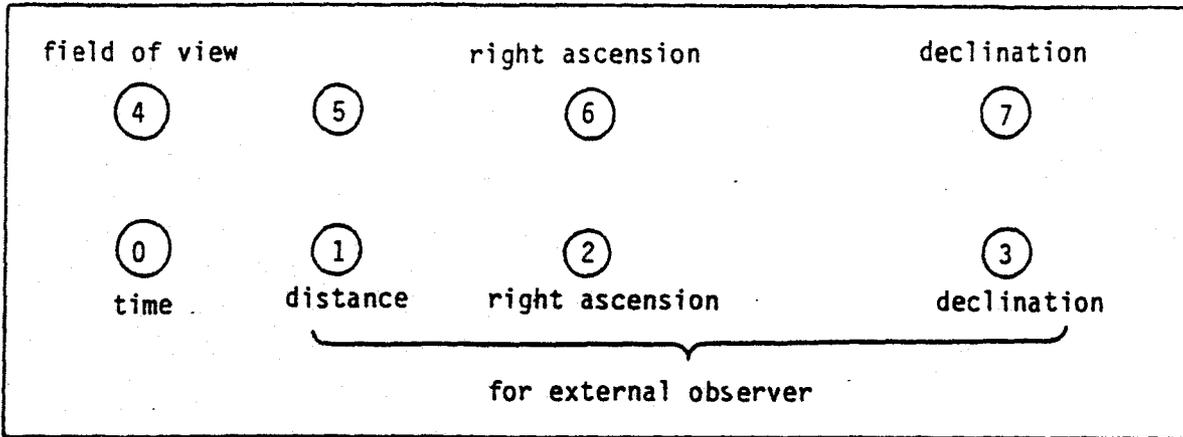
II.1.3 GALSAT.EPH

GALSAT.EPH is another direct access binary file of 12445 blocks and is a converted form of the standard satellite ephemeris file found on the 1100/81's. The file is used to produce states for the Galilean satellites from 2 January 1988 to 2 January 1992. It was created from tape by program SATRD.

II.1.4 Knobs and switches

All but one of the eight knobs, and all of the 16 switches of the Evans and Sutherland Picture System 2 are used to interactively control the GIP program during execution. The knobs and switches are explained on the following pages.

Knobs



knob	explanation
0	Controls time. To increase the delta time at which absolute time is changing turn knob clockwise. Change is non-linear but goes as the cube of the turn. Delta time cannot be turned higher than ± 20 min to protect integrity of spacecraft integrator.
1	If we are observing from the position of an external observer this knob controls the distance from the Jupiter barycenter.
2	Again for an external observer this knob controls the right ascension WRT the Jupiter barycenter.
3	Controls declination WRT the Jupiter barycenter for an external observer.
4	Controls field of view size of scene.
5	Not used.
6	Controls right ascension pointing angle of the observer. An angle is calculated by the program. This knob adjusts that angle.
7	Same as knob 6 only for the declination pointing angle.

Switches

15 O	14 O	13 O	12 O	11 O	10 O	9 O	8 O	7 O	6 O	5 O	4 O	3 O	2 O	1 O	0 O
stop time	names	zero pointing	FOV= 9°	FOV= 1.5°	FOV= .424°	closest body	external observer	R+7= on Earth	S/C	Jupiter and stop program	Callisto	Ganymede	Europa	Io	GIP output file

switch

explanation

0	When set, writes GIPOUT.DAT output file for the scene being viewed. The time, satellite name, and RA and DC are written in formatted form. Time is stopped by the switch.
1	Causes Io to be in the center of field for normal pointing angles.
2	Same as 1 but for Europa.
3	Same as 1 but for Ganymede.
4	Same as 1 but for Callisto.
5	If switch is set will cause program termination. If switch is not set and none of the switches 1-4 or 6 are set then the light for the switch will be lit to signify Jupiter at the center of field (default).
6	Same as 1 but for the spacecraft. However switch 8 must be set.
7	If switch 8 is also set then the scene is that seen from Earth.
8	Causes scene to be that observed by an arbitrary external observer. See knobs 1-3.
9	When set, causes body closest to observer to be in the center of field.
10	Causes field of view of scene to be that of the Galileo cameras - .424°.
11	Field of view set to 1.5°.
12	Field of view set to 9° (no stars displayed).
13	If knobs 6 and 7 have been used this switch nullifies their effect.
14	Labels objects being viewed. For stars, the visual magnitude and spectral class (when known) are displayed.
15	Stops time.

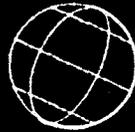
II.2 Output

II.2.1 GIPOUT.DAT

Everytime switch 0 is hit an entry is made in GIPOUT.DAT. Ephemeris time (calendar date), satellite name, and RA and DC (deg) are written in formatted form.

II.2.2 Picture System Monitor (the CRT)

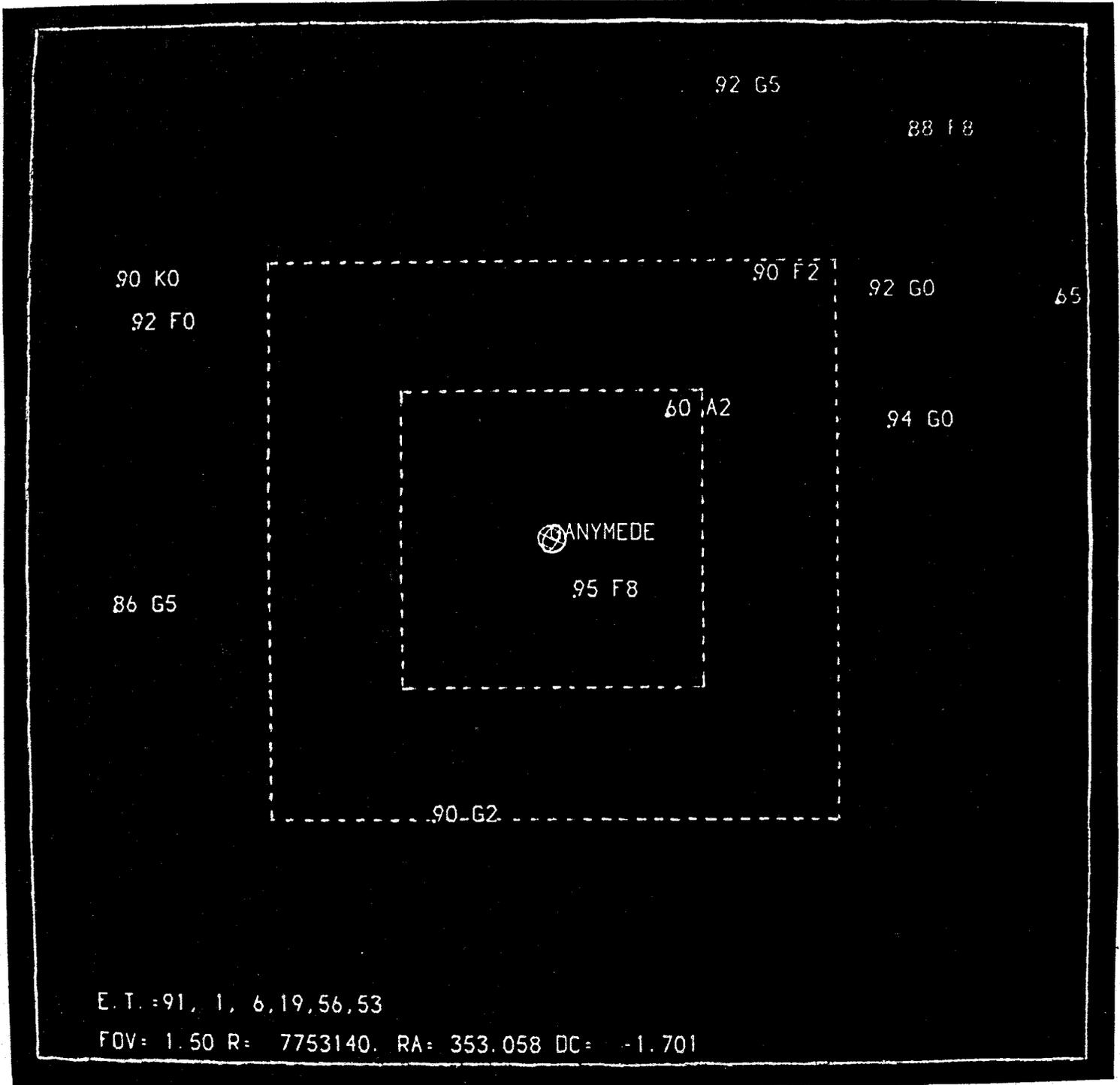
The scenes calculated by GIP are displayed on the Picture System Monitor. The monitor has a resolution of about 4000 x 4000 although the program treats it as 32767x32767. Tens of thousands of vectors may be displayed before the system becomes taxed. Jupiter, the Galilean satellites, the spacecraft, and stars may all be displayed. Also 2 dotted boxes of .424 and .8 degrees are displayed signifying camera field of view and pointing error. At the bottom of the screen are displayed ephemeris time in calendar date (E.T.), field of view in degrees (FOV), distance to observed object in kilometers (R), and right ascension and declination pointing angles in degrees (RA, DC). On the following pages are actual displays from the monitor.



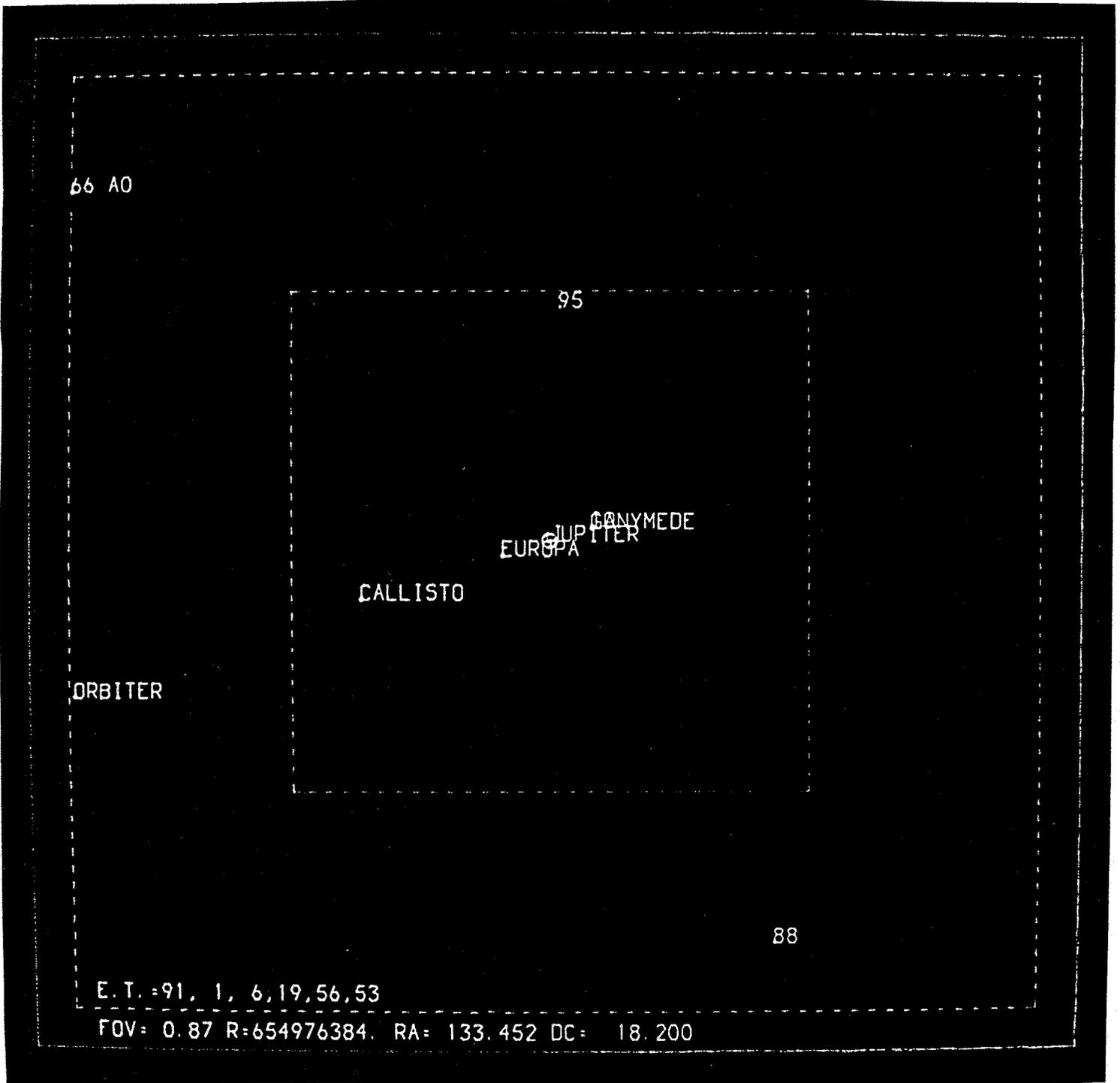
E. T. = 91, 1, 6, 19, 56, 53

FOV= 0.42 R= 7753140. RA= 353.058 DC= -1.701

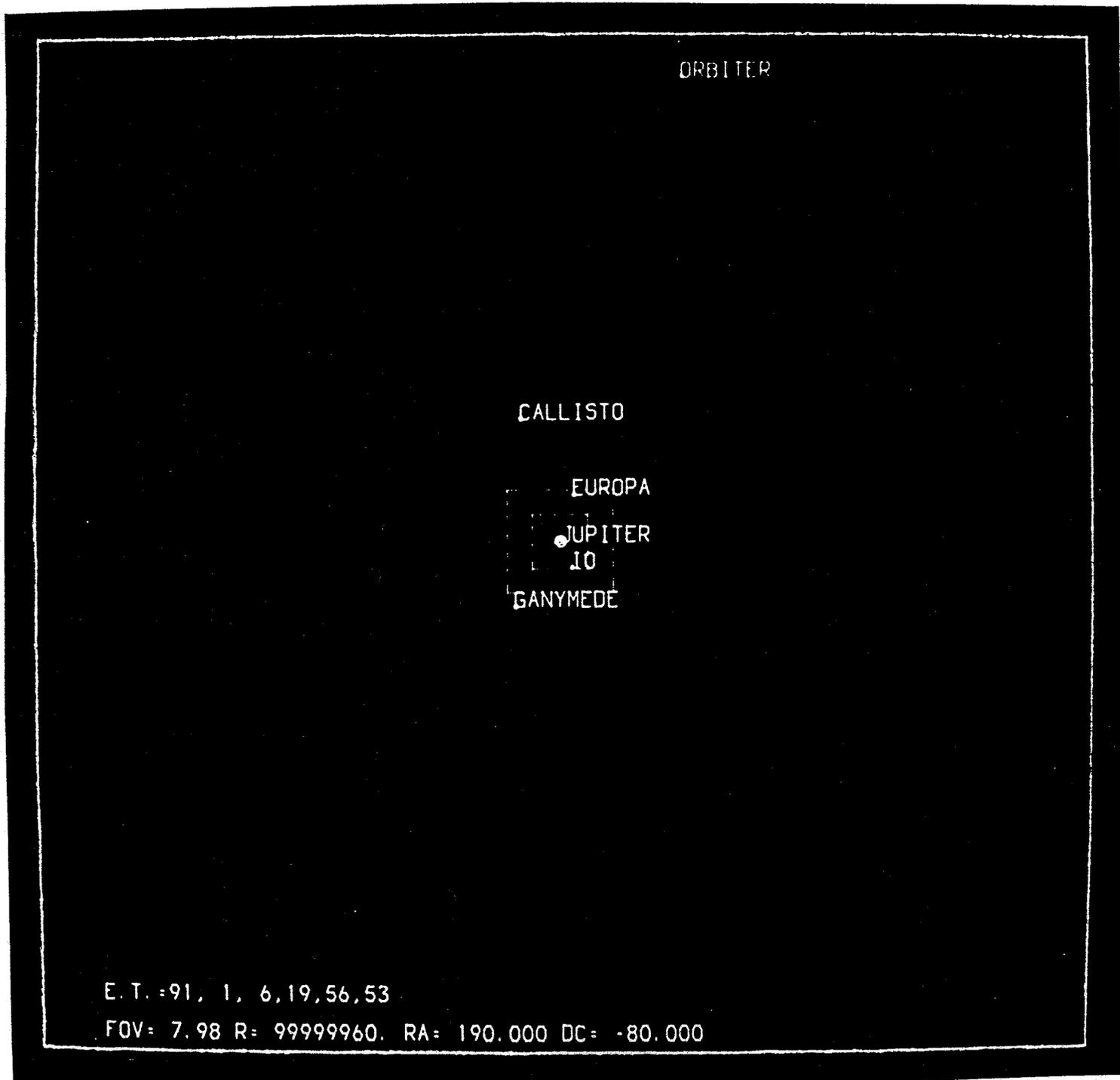
Here GIP was run for the 82-06 (Δ Vega) mission and time stopped at 91, 1, 6, 19, 56, 53. The spacecraft is near apoapsis on its approach before the Ganymede, G4 encounter. The scene is that which would actually be observed by the Galileo cameras. The faint star near Ganymede traversed the field from left to right in 1 hour 27 minutes and 44 seconds. North is always up when viewing the monitor.



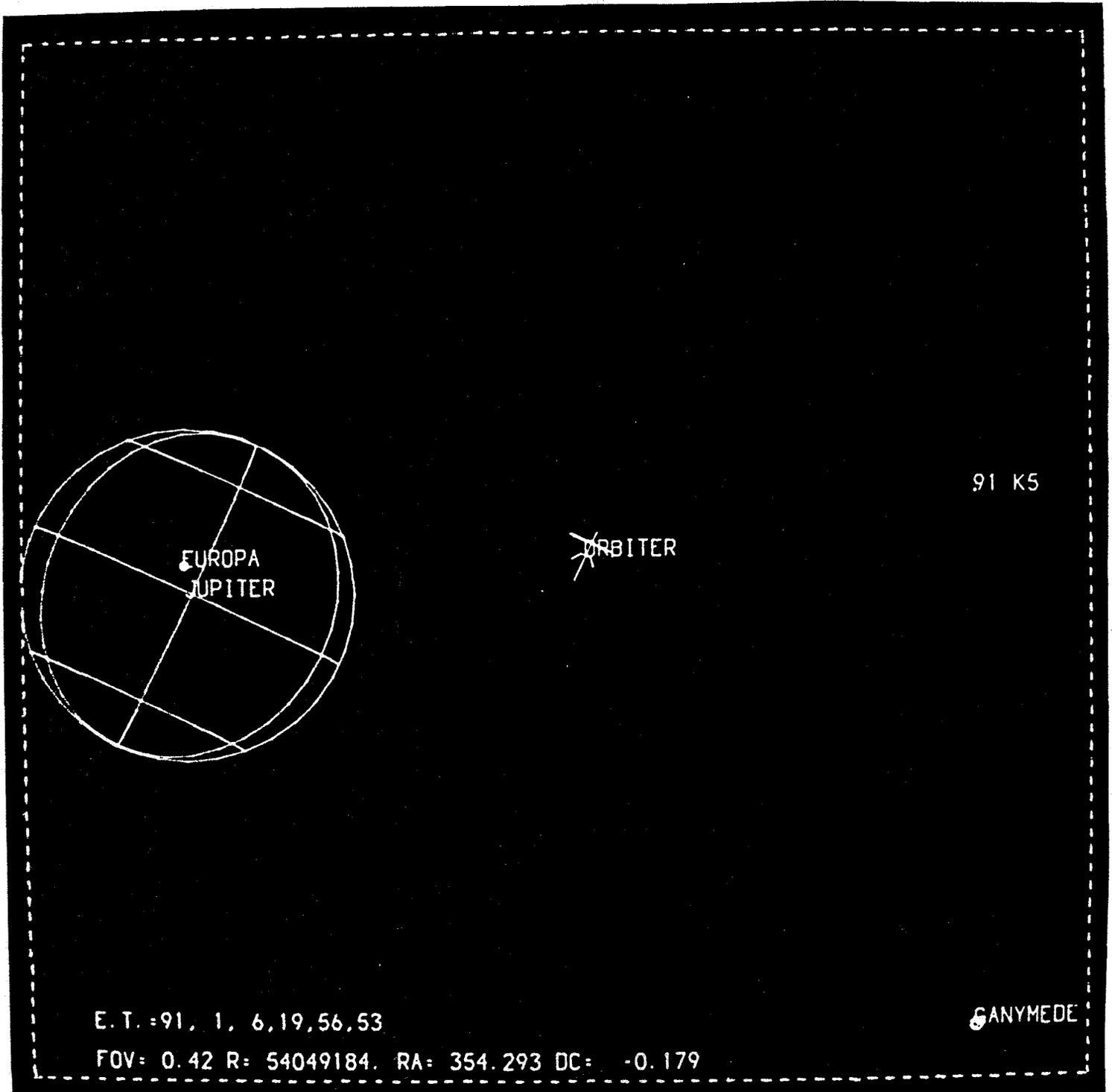
This is the same scene as that shown on the preceding page. However the field of view has been increased from .424 to 1.50 degrees and labels have been added. Note the two dotted boxes which represent camera field and pointing error.



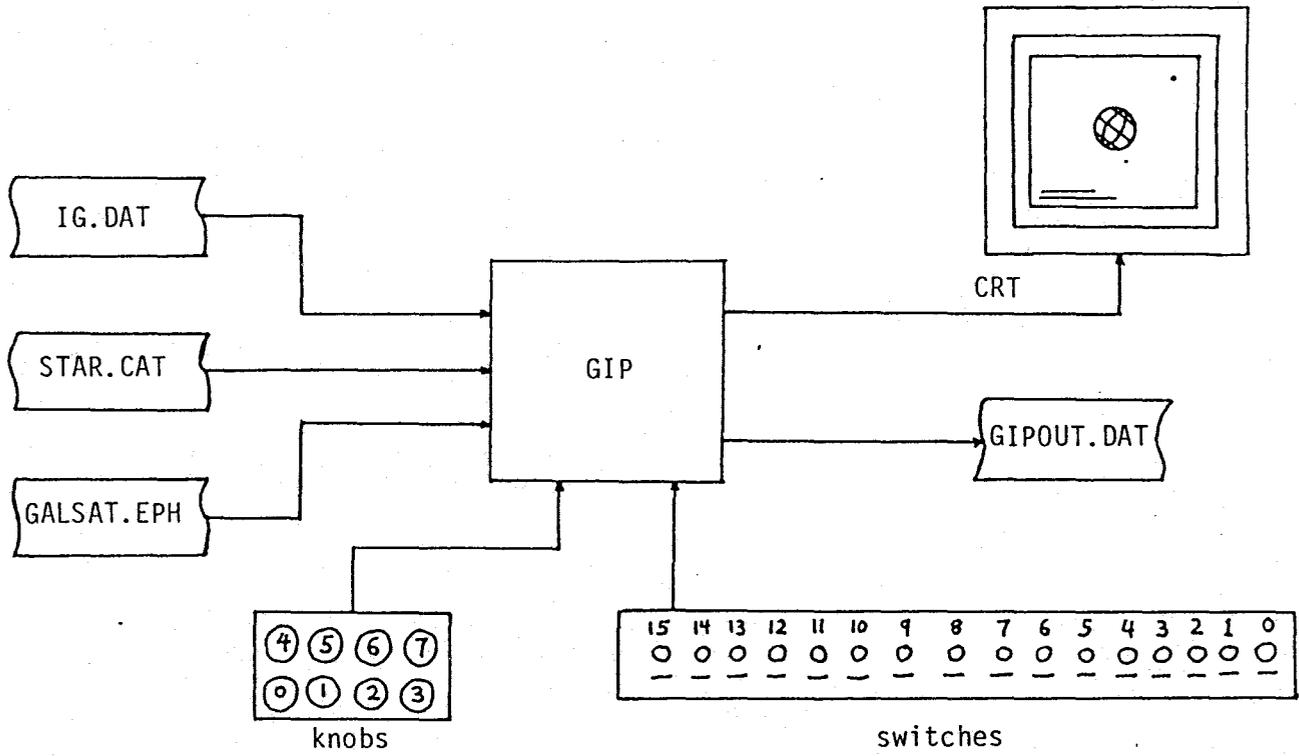
This is the scene as seen from the Earth with a .87 degree field of view.



Here the scene is viewed by an arbitrary external observer at a range close to 100,000,000 km and a high declination. Note that the field of view is large (near 8 degrees) and no stars are displayed.



The external observer has been moved to a position behind the spacecraft looking towards Jupiter, Europa, and Ganymede.



GIP I/O



The Computer Graphics Lab Picture System Workstation.

III. Runstream

Once the initialization program, IG, has been successfully run the user need only type the three letters GIP at his terminal to run GIP. The user should become familiar with the RSX-11m operating system of the PDP 11/55 which drives the Picture System (ref.6 and 7). Special attention should be given to the editor, EEE, and the peripheral interchange program, PIP.

However to get started the user may follow the procedure shown below to run IG and GIP. He should have a start time, the state of the orbiter, and the time points and satellite number from the VUGEOM/PICSEL runs. To sign on type the following

```
> HEL IST
> [password-see me]
> CON 3
```

The password is not echoed by the computer.

Next the user should use the editor to put his data in two files. The time and state should be put in file IG.INP. While the VUGEOM/PICSEL data belongs in file PICSEL.DAT. The following sequence of commands will input the data. An explanation is given to the right.

command	explanation
> EEE IG.INP	Edit file IG.INP
*1,\$D	Erase file
*A	Allow new lines to be input
90 09 06 19 12 00	Input ephemeris time
-9283270.73	X coordinate, 1950.0EME, Jupiter centered, km
282031.87	Y coordinate
521002.60	Z coordinate
-.0700242786948951798	X velocity component, km/sec
-1.20244143001284249	Y velocity component
-.566876185409617164	Z velocity componenet
(carrige return)	Must input a blank line
.	Stop input mode
*W	Write file
*Q	Get out of editor

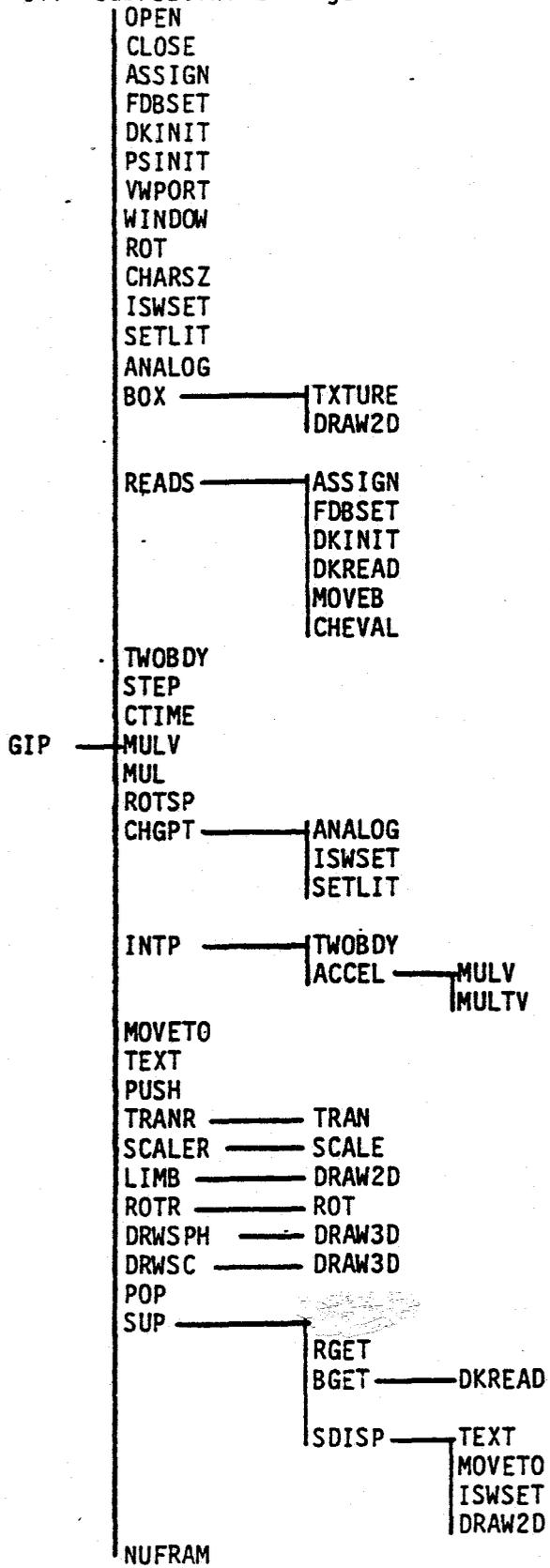
command	explanation
>EEE PICSEL.DAT	Edit file PICSEL.DAT
*1,\$D	Erase file
*A	Allow new lines to be input
90,09,06,20,00,00--3	VUGEOM/PICSEL time and satellite
90,09,06,21,00,00--3	VUGEOM/PICSEL time and satellite
90,09,07,04,00,00--3	VUGEOM/PICSEL time and satellite
90,09,07,05,00,00--3	VUGEOM/PICSEL time and satellite
90,09,07,06,00,00--3	VUGEOM/PICSEL time and satellite
90,09,08,02,00,00--3	VUGEOM/PICSEL time and satellite
90,09,08,03,00,00--3	VUGEOM/PICSEL time and satellite
90,09,08,04,00,00--3	VUGEOM/PICSEL time and satellite
90,09,10,08,00,00--9	No other time points-signified by 9
.	Stop input mode (up to 39 may be input)
*W	Write file
*Q	Get out of editor
>PIP SAVE.INP = IG.INP	If the user wishes to save either IG.INP or PICSEL.DAT, he may use the PIP command to create a new file.
>PIP PICSEL.SAV = PICSEL.DAT	
>IG	Run program IG (initialize GIP)
>GIP	Run GIP

All files are assigned automatically by IG and GIP. GIP will now run and the user may control its execution with the knobs and switches. Be sure switch 5 is initially unset.

Some other useful editor commands are given below.

command	explanation
C;string 1; string 2;p	Change string 1 to string 2 and print
QQ	Do not write file
DP	Delete current line and print next one
n_1, n_2	Print lines n_1 to n_2
n_1	Go to line n_1
/string/	Locate string

IV. Subroutine Linkages



The subroutine linkages for GIP are shown on the preceding page. A brief description of each subroutine is given below. An S by the subroutine name means it is a system subroutine. A PS by the subroutine name means it is a Picture System 2 subroutine. For more information see the reference given or obtain the subroutine listings from me (they are well commented).

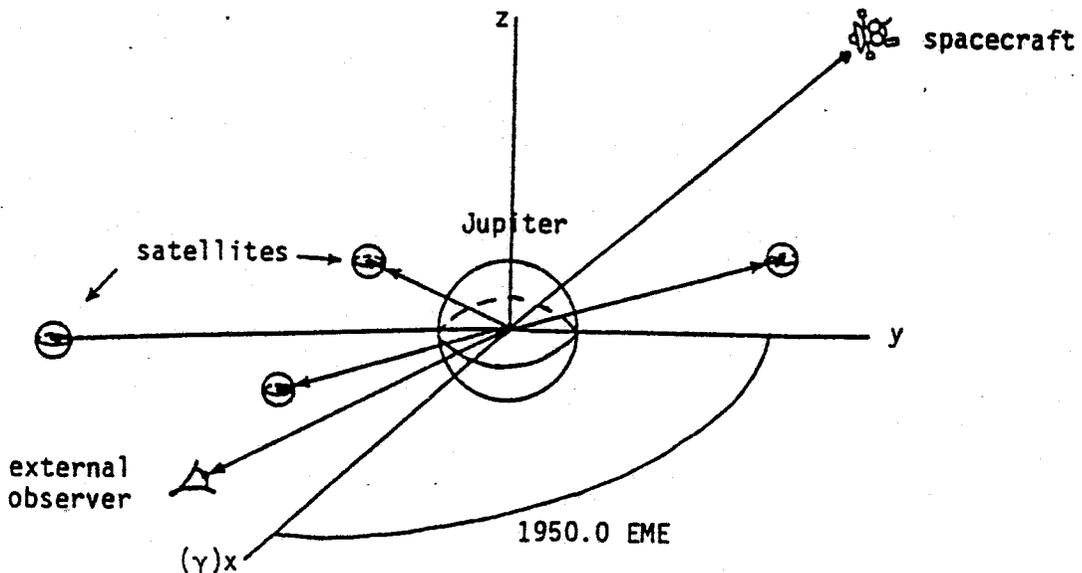
subroutine	description
ACCEL	Calculates sun and Jupiter J2 acceleration perturbation. See listing.
ANALOG (PS)	Gets differential change of knob. See reference 9.
ASSIGN (S)	Assigns files. See reference 8.
BGET	Get buffer for a record from the star file. See listing.
BOX	Draws a 2-D box on the CRT. See listing.
CHARSZ (PS)	gets size of characters displayed by the Picture System. See reference 9.
CHEVAL	Does Chebyshev polynomial evaluation. See listing.
CHGPT	Gets camera pointing angles of right ascension and declination, allows user to modify these angles interactively, and then recomputes unit direction vector. See listing.
CLOSE (S)	Closes files. See reference 8.
CTIME	Converts julian days to calendar date. See listing.
DKINIT (S)	Initializes random access binary I/O. See reference 8.
DKREAD (S)	Reads random access binary file. See reference 8.
DRWSC	Draws simple 3-D spacecraft on CRT. See listing.
DRWSPH	Draws 3-D sphere on CRT. See listing. Also see listing for program MAKSPH which writes code for DRWSPH.
DRAW2D (PS)	Draws 2-D objects on CRT. See reference 9.
DRAW3D (PS)	Draws 3-D objects on CRT. See reference 9.
FDBSET (S)	Sets I/O options for a file. See reference 8.
INTP	Performs simple integration to get state of spacecraft. See listing.
ISWSET (PS)	Returns status of a switch. See reference 9.

subroutine	description
LIMB	Draws a limb for Jupiter and the Galilean satellites. See listing.
MOVEB	Moves byte strings. See listing.
MOVETO (PS)	Moves to a specified location in the data space. See reference 9.
MUL	Multiplies two 3x3 matrices. See listing.
MULTV	Multiplies transpose of 3x3 matrix with a vector. See listing.
MULV	Multiplies a 3x3 matrix with a vector. See listing.
NUFRAM (PS)	Erase current scene; get ready for calculation of next. See reference 9.
OPEN (S)	Opens files. See reference 8.
POP (PS)	Restore PUSHed matrix stack. See reference 9.
PSINIT (PS)	Initializes Picture System. See reference 9.
PUSH (PS)	Saves hardware matrix stack. See reference 9.
READS	Reads satellite ephemeris file. See listing.
RGET	Calculate record number of star file from input RA and DC. See listing.
ROT (PS)	Forms 4x4 left-handed rotation matrix. See reference 9.
ROTR	Prepares data for Picture System subroutine ROT. See listing.
ROTSP	Forms 3x3 right-handed rotation matrix. See listing.
SCALE (PS)	Scales data. See reference 9.
SCALER	Prepares data for Picture System subroutine SCALE. See listing.
SDISP	Display stars. See listing.
SETLIT (PS)	Sets little light over switch on or off. See reference 9.
STEP	Performs simple integration to get state of Earth. See listing.
SUP	Put star field up on CRT. See listing.
TEXT (PS)	Displays text on CRT. See reference 9.
TRAN (PS)	Translates vectors to specified location within data space. See reference 9.

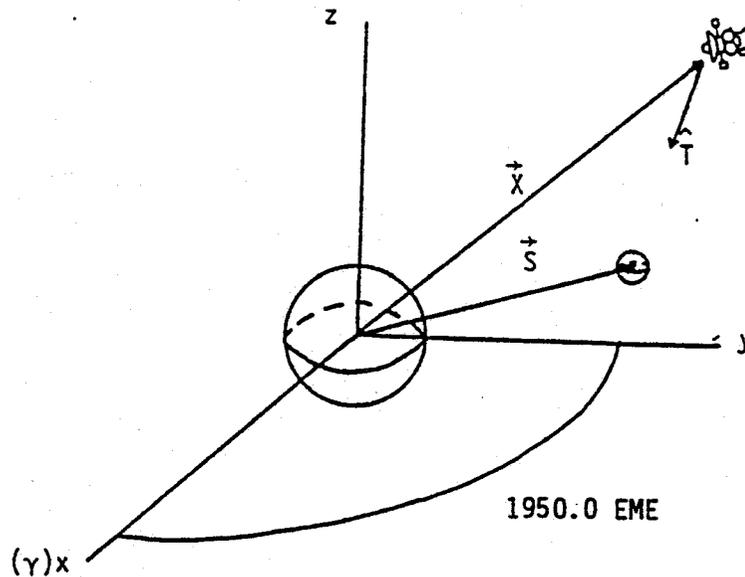
subroutine	description
TRANR	Prepares data for Picture System subroutine TRAN. See listing.
TWOBDY	Calculates two-body motion. See listing.
TXTURE (PS)	Sets texture of lines drawn on the CRT by the Picture System. See reference 9.
VWPORT (PS)	Sets Picture System screen parameters. See reference 9.
WINDOW (PS)	Sets Picture System data space, eye position, and either orthographic or true perspective view. See reference 9.

V. Theory

GIP first obtains states for the Earth and Jupiter barycenters WRT the sun center and then states for Jupiter, the natural satellites, spacecraft, and external observer (or Earth) WRT the Jupiter barycenter. The planet states, originally those from DE114, are propagated about the sun with two body motion. States for the moons and Jupiter are obtained from Lieske's theory. Spacecraft motion is integrated using a multi-conic method similar to reference 5. However the locations of Jupiter and the moons WRT the barycenter are not considered. Considering the Jupiter system we have the situation depicted below.

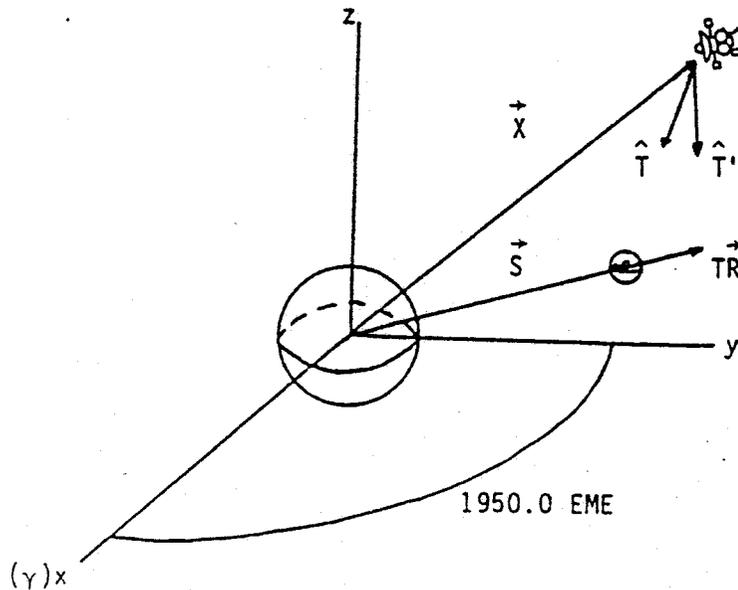


The user decides from where he wishes to view the system. This can be from either the spacecraft or external observer (or Earth). If the spacecraft is chosen then Jupiter and the moons may be viewed. If the external observer is chosen then Jupiter, the moons, and also the spacecraft may be viewed. Next the user decides which body he wishes to have centered in the field of view. Let's assume we are observing one of the moons from the spacecraft. All the distances to the bodies are calculated and stored for future use, and also the unit direction vector to the observed satellite is calculated (\hat{T}).



The program then calculates the right ascension and declination of the \hat{T} vector.

The user may interactively modify the camera pointing and thus the \hat{T} vector to observe the scene with the satellite off center. GIP then computes a translation vector \vec{TR} shown below.



In summary we get \hat{T} :

$$\vec{T} = \vec{S} - \vec{X}$$

$$\hat{T} = \frac{\vec{T}}{|\vec{T}|} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

and the pointing angles α_T (right ascension) and δ_T (declination)

$$\alpha_T = \tan^{-1} (T_y/T_x)$$

$$\delta_T = \sin^{-1} (T_z)$$

Next adjust α_T and δ_T and recompute \hat{T}

$$\alpha'_T = \alpha_T + \Delta\alpha_T$$

$$\delta'_T = \delta_T + \Delta\delta_T$$

$$T'_x = \cos(\alpha'_T) \cos(\delta'_T)$$

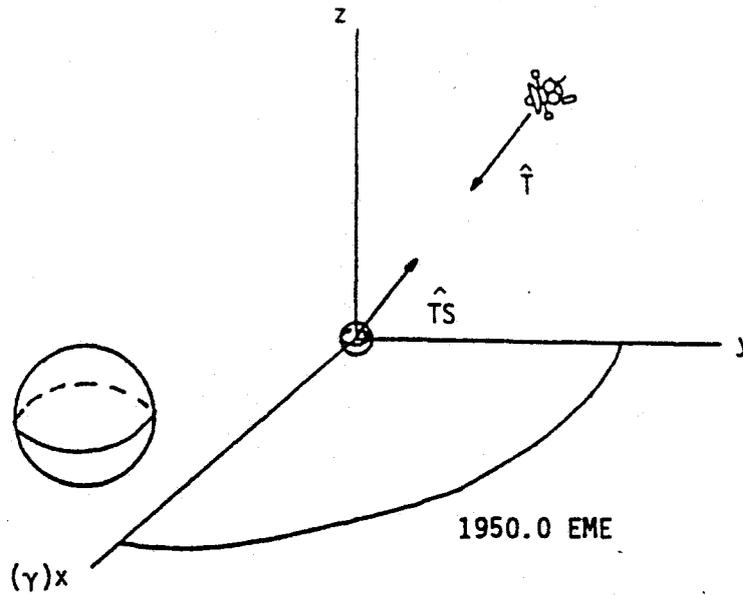
$$T'_y = \sin(\alpha'_T) \cos(\delta'_T)$$

$$T'_z = \sin(\delta'_T)$$

Finally compute the translation vector \vec{TR}

$$\vec{TR} = \vec{X} + \hat{T}' |\vec{T}|$$

The whole system is now translated by the translation vector \vec{TR} . For simplicity let's assume $\vec{TR} = \vec{S}$ and that we are pointed directly at the satellite. We will have the situation shown below.

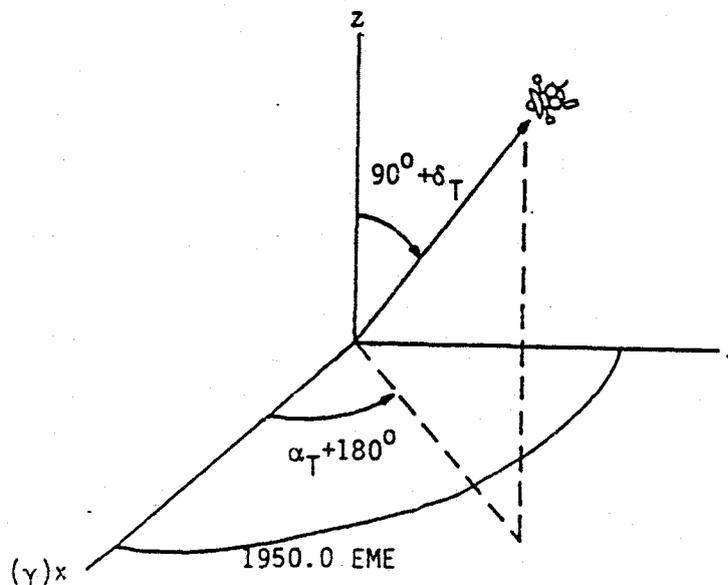


Now the vector pointing from the satellite to the spacecraft, $\hat{T}S$, is of course $-\hat{T}$, and the right ascension, α_{TS} , and declination, δ_{TS} , of $\hat{T}S$ may be obtained from \hat{T} via

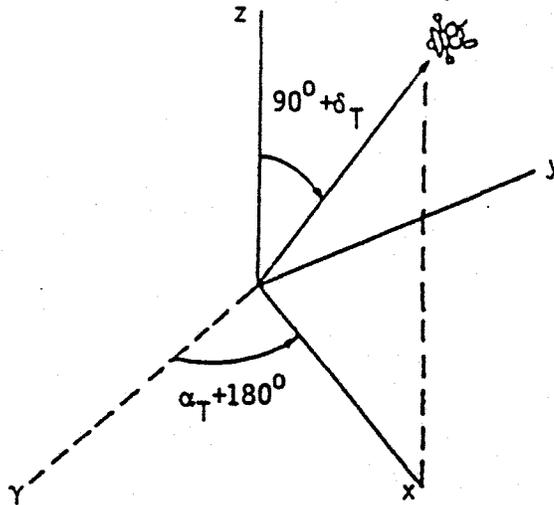
$$\alpha_{TS} = \alpha_T + 180^\circ$$

$$\delta_{TS} = -\delta_T$$

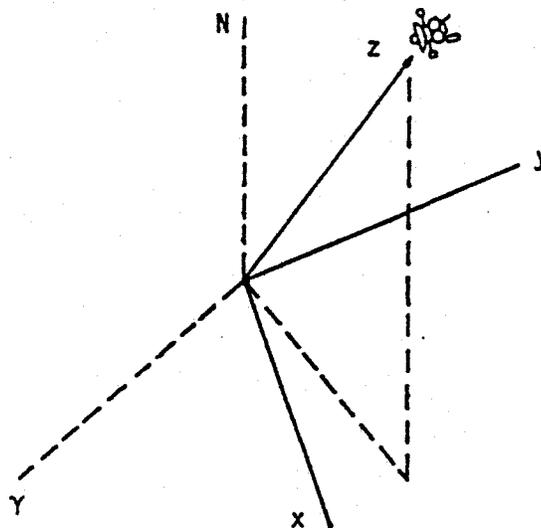
Because the Evan & Sutherland Picture System 2 graphics system puts an observer on the z axis looking towards the origin, we wish to transform the 1950.0 EME coordinates of the bodies to a coordinate system where the z axis is aligned with the $\hat{T}S$ vector. In terms of the RA and DC of the \hat{T} vector we have



First form the transformation matrix $[\alpha_T + 180^\circ]_3$, and transform to

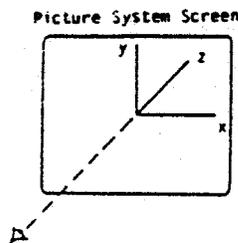


Next form the transformation matrix $[90 + \delta_T]_2$ and transform to

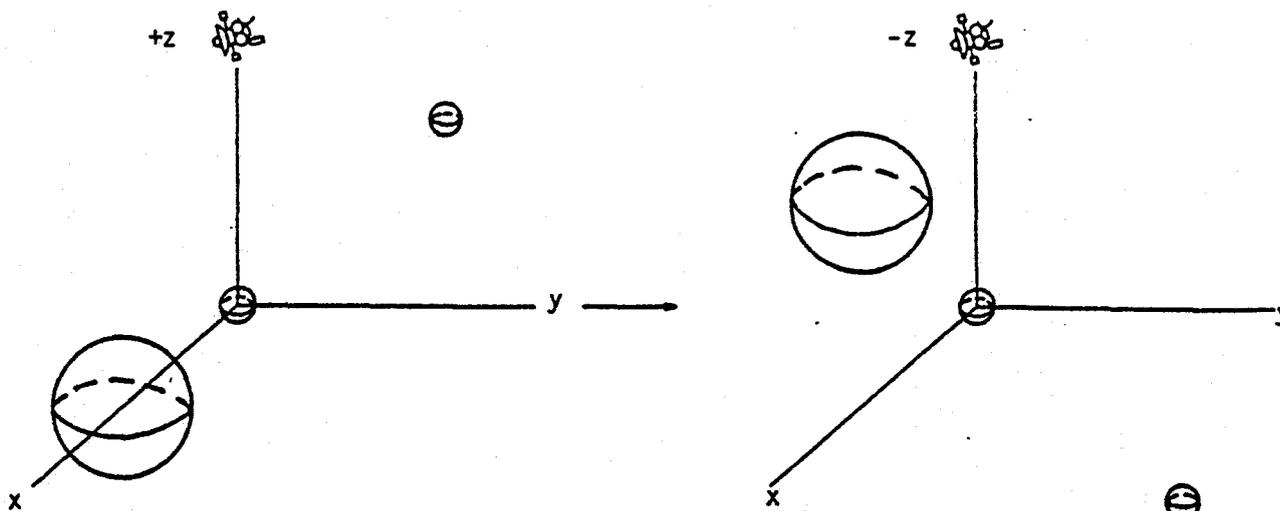


Note that for an observer viewing the system from the spacecraft the north or + declination direction is along the -x axis. Also note that + right ascension is along the -y axis.

The Picture System is a left-handed coordinate system where the eye is placed at -z looking into the screen in a +z sense as shown below



If our right-handed system with eye on the +z axis is mapped directly into a left-handed system and the eye position is changed to -z we will have the situation shown below



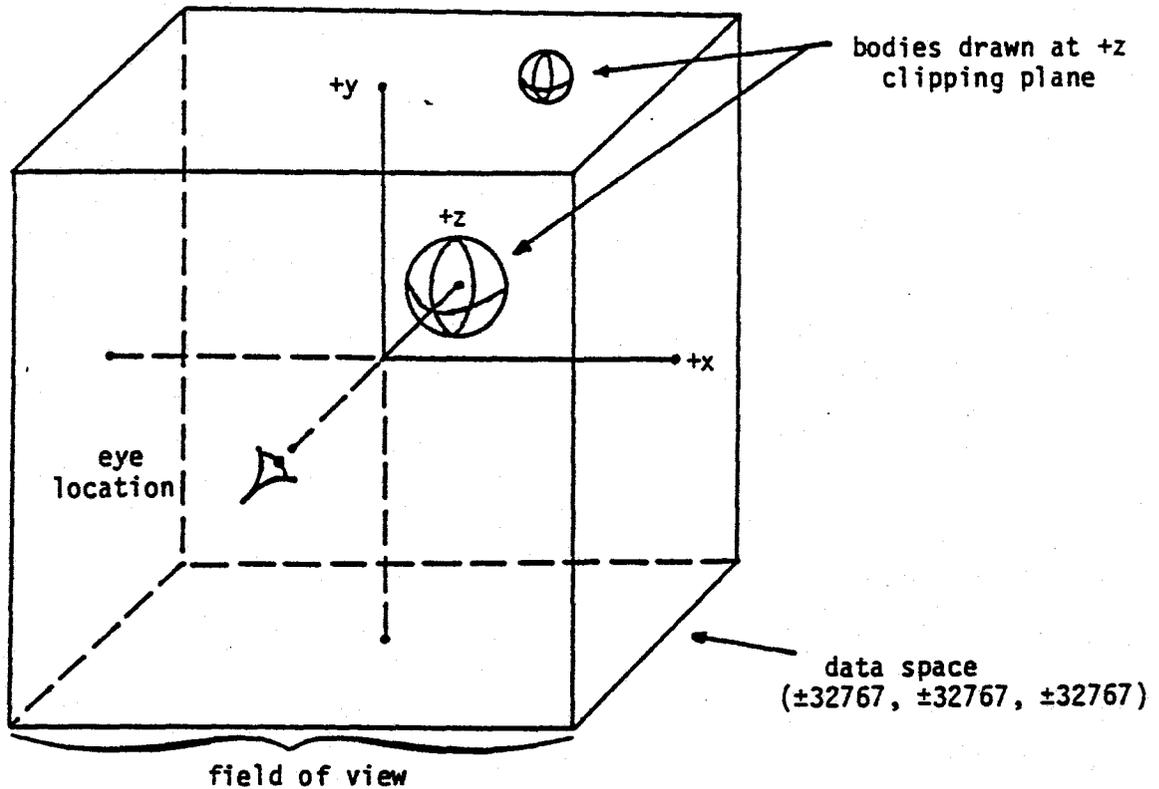
The scene is the same except that the z positions of the observed bodies are inverted. This turns out to be unimportant as explained below.

There are a number of ways data may be drawn by the Picture System, including orthographic and true perspective views. The reader may wish to consult reference 9 on the Picture System for details. The following method was adopted for speed, flexibility (great magnification is possible), and hidden line capability.

Bodies are drawn in an orthographic sense. That is to say all light rays emanating from them appear parallel, and a body is the same size no matter where it is drawn on the z axis. The correct size of a body is calculated by GIP from the distance to it and the field of view of the observer. Picture System x and y coordinates are obtained from the real x and y coordinates also scaled by the distance to the body and field of view.

Picture System x,y,z coordinates vary from ± 32767 . If any of the data lies within this box it is drawn. Data at the edges is clipped and therefore satellites and Jupiter are drawn at $z = +32767$. This allows hidden line clipping. Remember that since the bodies are orthographic the z axis is irrelevant. A limb is drawn for the bodies. Since clipping is not desired for the spacecraft, when drawn, it is placed at $z=0$. We have

then the situation shown below where right-handed, real, transformed coordinates are mapped and drawn on the left-handed Picture System.



Picture System Orthographic Representation of Data

The equations used to get the Picture System representation are shown below. For body distance, R, and field of view, F, get width of field, W, in kilometers at R

$$W = R \tan (F/2)$$

Now get scaling factor, S, to convert kilometers at distance R to Picture System units

$$S = \frac{32767}{W}$$

Convert body radius, r, and x and y coordinates to Picture System body radius r_{PS} , and Picture System x_{PS} and y_{PS} coordinates

$$r_{PS} = (r)S$$

$$x_{PS} = (x)S$$

$$y_{PS} = (y)S$$

Next simply set $z_{PS} = 32767$

The bodies themselves are oriented in the usual way, and then the two previous transforms applied to the whole system are also applied to the bodies. The bodies are all assumed to have Jupiter's pole. Jupiter is given a rotation rate of approximately 10 hours while the satellites are given rotation rates corresponding to their orbital periods. The spacecraft is given a rotation rate of around 1 hour. All bodies are given arbitrary prime meridians. The following transformations apply to go from of date body-fixed to the final right-handed coordinate system.

$$\cancel{[\delta_T + 90]_2 [\alpha_T + 180]_3 [\alpha_J]_3 [-\delta_J]_1 [-\ell]_3} \quad \text{see below}$$

where ℓ is the longitude, α_J and δ_J are the right ascension and declination of Jupiter's pole, and α_T and δ_T are the previously defined right ascension and declination of the camera pointing direction. To account for the fact that the Picture System is a left-handed coordinate system, the z components of the coordinates defining the body are inverted and rotations about the z axis are actually done in an opposite direction as that shown above.

This is a very valid representation of the true scene for all but very large fields of view. For large fields of view ($>20^\circ$) observed bodies would be smaller, in slightly different places, and lines of latitude and longitude on the bodies would be deformed. This is due to the orthographic projection and that for off axis bodies the range instead of the z component is used as a scaling factor.

Now in order to complete the scene we need to add the star background. Recall that the star file is composed of many records each covering a $5^\circ \times 5^\circ$ field. GIP computes which record each of the four corners of the field of view lies in. The stars are then read from the star file, stored in the program, and displayed in such a way as to reduce redundancy and show all

$$[\delta_T + 90]_2 [\alpha_T + 180]_3 \left\{ [90 - \delta_J]_2 [\alpha_J]_3 \right\}^T [-\ell]_3$$

stars in the field of view. Corner number one is considered first and so forth. If a corner other than number one has the same record number as a corner already considered, it is skipped.

If the stars for a record where a corner lies are to be displayed, they are read from the star file, stored in a buffer, and displayed. When the stars for that corner need to be displayed again the star file need not be read unless the record number for the corner has changed. It is not practical to try and display all stars for more than a $3^0 \times 3^0$ field because there are so many.

Star positions on the Picture System are calculated as follows. First the right ascension and declination of the center of the field of view are converted from degrees to radians. Right ascension is made to range from $-\pi$ to $+\pi$. This is done because star coordinates are also given in this way to reduce storage requirements on the computer. The PDP can efficiently handle 2 byte integer words ranging from +32767 to -32767. One will note that this is almost ideal for representing angles in radians varying from $-\pi$ to $+\pi$. The round off error is less than $.003^0$ or only .7% of the camera field of view of $.424^0$. This should be satisfactory for our purposes. Next a conversion factor to go from radians to Picture System units is calculated. Remember that both the x and y axes vary from -32767 to +32767. Therefore to get the conversion factor, C, we have, given field of view F (in degrees),

$$C = 65534 / [F \left(\frac{\pi}{180} \right)]$$

The actual star locations on the Picture System are now calculated, and if they lie within the field of view they are displayed. Remember that + right ascension lies along the -y axis and + declination lies along the -x axis. Therefore to calculate the Picture System x and y coordinates we have

$$x = -(\delta_s - \delta_R)C$$

$$y = -(\alpha_s - \alpha_R)C \cos \delta_s$$

where α_s and δ_s are the right ascension and declination of the star and α_R and δ_R are the right ascension and declination of the center of field. The z coordinate is unimportant and placed at z=0.

In order to correctly display the star magnitudes the standard equation relating magnitude and luminosity is used:

$$l_n/l_m = 100^{(m-n)/5}$$

where m and n are the visual magnitudes of two stars and l_m and l_n are the corresponding luminosities. Now the x,y Picture System space is composed of $(32767)^2 = 1,073,676,289$ pixels (the actual resolution of the CRT is around 4000x4000). A star magnitude of 11.0 is defined as capable of lighting up one pixel element. Therefore the pixel elements a star will light up is given by

$$l = 100^{(11.0-m)/5}$$

where m and l are the visual magnitude and luminosity (in pixel elements) respectively. A short table of m and l is given below.

l	m
1	11.00
2	10.25
3	9.81
4	9.49
5	9.25
6	9.05
7	8.89
8	8.74

Magnitudes between 11 and 5 are given true relative brightness. Magnitudes greater than 11 are given a true brightness of 11. Magnitudes less than 5 are given a true brightness of 5. Pixel elements are lit up in a counter

clockwise manner surrounding the location of the star to obtain the correct brightness.

The entire scene is now completed: all bodies and stars have been drawn. The scene is rotated by 90° so that north is always up when viewing the Picture System monitor. The method outlined here should be easily transportable to simpler raster devices, because body sizes and x,y locations are all done in software.

Acknowledgements

Tom Duxbury originally suggested this problem. Bob Holtzman, group supervisor of the Computer Graphics Lab, and Jim Blinn have been very gracious in allowing me to use the Graphics Lab. Tom Duxbury and Rich Doyle wrote the subroutines READS, MOVEB, CHEVAL, MUL, MULV, and MULTV. Bill Owen supplied the subroutine TWOBODY and offered many valuable suggestions on writing a simple, accurate integrator for the spacecraft.

References

1. "VUGEOM (Version 2) User's Guide," W. M. Owen, Jr., IOM 314.3-330, 22 July 1981
2. "A Program Called PICSEL ...", R. P. Davis, IOM 314.3-321, 11 June 1981
3. "Galileo Picture Planning-Status and Plans," R. P. Davis, IOM 314.8-357, 1 June 1982
4. "Picture Planning 10 Year Plans," R. P. Davis, IOM 314.8-361, 12 May 1982
5. "A Multi-Conic Propagator for the Jupiter System," W. M. Owen, Jr., E.M. 314-239, 15 April 1981
6. "Introduction To RSX-11M", Digital Equipment Corporation, order No. DEC-11-OMIEA-B-D, Maynard, Massachusetts, November 1976
7. "RSX-11 Utilities Procedures Manual," Digital Equipment Corporation, order No. DEC-11-OXMDA-A-D, Maynard, Massachusetts, January 1977

References

8. "Fortran IV-PLUS User's Guide," Digital Equipment Corporation, order No. DEC-11-LFPUA-B-D, Maynard, Massachusetts, December 1975
9. "Picture System 2 User's Manual," Evans & Sutherland Computer Corporation, Document E&S #901129-001 A1, 580 Arapeen Drive, Salt Lake City, Utah, 13 May 1977

JDC:ac