Deepspace Interplanetary Navigation Operations Colorado Research EXplorer (DINO C-REx)

DINO C-REx Technical Memorandum

Document ID: DINO_C-REx-Image Generation

SYSTEMS ENGINEERING REPORT 4.12: STAR SIMULATION

Prepared by	Matt Muszynski

Status: Initial Version
Scope/Contents
Description of the DINO C-REx method for simulating stars and limiting them to only those in the camera FOV.

Rev:	**Change Description**	**By**
1.0	Initial Release. Much of the material comes from the Spring 2017 Camera Module Continuity Documentation	Matt Muszynski

Contents

Table 2: Columns Present in Full Tycho-2 Database

Name	Type	Length	Used by DINO C-REx?
_RAJ2000	F10.6	Right ascension (FK5, Equinox=J2000.0) at Epoch=J2000, proper motions taken into account (computed by VizieR, not part of the original data)	No
_DEJ2000	F10.6	Declination (FK5, Equinox=J2000.0) at Epoch=J2000, proper motions taken into account (computed by VizieR, not part of the original data)	No
_DEJ2000	F10.6	Declination (FK5, Equinox=J2000.0) at Epoch=J2000, proper motions taken into account (computed by VizieR, not part of the original data)	No
TYC1	I4	[1,9537]+= TYC1 from TYC or GSC (1)	No
TYC2	I5	[1,12121] TYC2 from TYC or GSC (1)	No
TYC3	I1	[1,3] TYC3 from TYC (1)	No
pmRA	F10.6	[1,3] TYC3 from TYC (1)	No
pmRA	F7.1	[-4418.0,6544.2]? prop. mot. in RA*cos(dec)	No
pmDE	F7.1	[-5774.3,10277.3]? prop. mot. in Dec	No
BTmag	F6.3	[2.183,16.581]? Tycho-2 BT magnitude (7)	**Yes**
VTmag	F6.3	[1.905,15.193]? Tycho-2 VT magnitude (7)	**Yes**
HIP	I6	[1,120404]? Hipparcos number [NULL integer written as an empty string]	No
RA (ICRS)	F12.8	Observed Tycho-2 Right Ascension, ICRS	**Yes**
DE (ICRS)	F12.8	Observed Tycho-2 Declination, ICRS	**Yes**

1 Overview

In order to use an opnav camera in a Basilisk simulation, the user must instantiate an opnav camera object in Python and define its fundamental characterstics. This is fully documented in SER 4.10. All of these should be set during scenario setup in the Python layer.

2 Tycho-2 Star Catalog

The stellar data used in the image generation portion of DINO C-REx is derived from Tycho-2 mission data retrieved from NASA's High Energy Astrophysics Science Archive Research Center (HEASARC).

2.1 HEASARC Tycho-2 Data Reduction

Because the HEASARC data retrieved from the VizieR Astronomical Server is designed to be used for high precision astrophysical applications beyond the scope of DINO C-REx, it contains several columns that were not needed. Table 2 show the data columns that are used for DINO C-REx's stellar data file, tycho_small.csv. Informed by Deep Space 1's dimmest observation capability of magnitude 9.5, all stars dimmer than 10th magnitude have also been eliminated from tycho_small.db. The text file originally retrieved from VizieR (asu.tsv), and a SQLite file containing its full data (tycho.db) have been retained in case future operations require data that was removed to create tycho_small.csv.

2.2 Coordinate Frame

HEASARC's Tycho-2 data is presented in the International Clestial Reference Frame (ICRF), which is derived from the positions of distant bright radio sources (mostly quasars) measured with Very Long

Baseline Interferometry (VLBI). The ICRF represents an order of magnitude higher accuracy than J2000, and the two are in agreement up to the precision of the latter. For this reason, DINO C-REx does not distinguish between the two, and uses the Tycho-2 data as if it were reported in J2000.

3 DINO C-REx Stellar Database

On top of the HEASARC data described above, the DINO C-REx stellar database (tycho.db) also includes columns for calculated temperature and solid angle subtended by source. Each of these columns is an estimate for the true value which is not well known for every star. In short, each is estimated via a numerical approximation that assumes all stars in the model are perfect blackbodies. See SER 4.1: Magnitude to Flux Conversion for more details.

4 Camera Field of View Calculation

DINO C-REx's camera model is based off of simple optics, with a thin lens approximation. It is assumed that all objects imaged are infinitely far away because all celestial objects will be many orders of magnitude farther away than any realistic camera focal length, illustrated in figure **??**. This allows the assumption[1] that in terms of the detector size (c) and the camera's focal length (f), the field of view of the camera in one dimension (γ) is

$$\gamma = 2\tan^{-1}\left(\frac{c}{2f}\right) \tag{1}$$

If we take c to be the diagonal of the detector illustrated in figure 5, and assume that the field of view of the camera will have the same relative dimensions as the detector itself, we can derive the angular dimensions of the camera's field of view (α, β) in terms of the detector dimensions (a, b) and γ.

$$\alpha = \frac{\gamma a}{\sqrt{a^2+b^2}} \tag{2a}$$

$$\beta = \frac{\gamma b}{\sqrt{a^2+b^2}} \tag{2b}$$

This field of view can then be used to map background stars as described in section **??**. Camera FOV calculation is accomlished in C++ via the function OpnavCamera::Calculate_FOV().

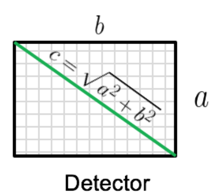

Fig. 1: Illustration of camera detector dimensions along with the intermediary calculation of the diagonal, used to alculate the FOV

5 Adding stars to Camera Object

Stars are added to the camera object upon initialization. Nominally the data is loaded from tycho.db (unless a different db is specified). Right Ascension, Declination, VTmag and BTmag are loaded, and unit vectors pointing from the spacecraft to each star in inertial space are calculated. Estimated stellar temperature is also loaded from the database[2]. Unit vector coordinates and Temperatures are saved

[1] https://www.edmundoptics.com/resources/application-notes/imaging/understanding-focal-length-and-field-of-view/
[2] Estimated temperature is not part of the original Tycho database, but was added by the DINO C-REx team using the process described in SER 4.1.

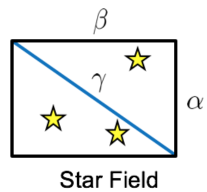

Fig. 2: Illustation of the camera FOV. α, β, and γ are the same dimensions as a, b, and c in fig 5)

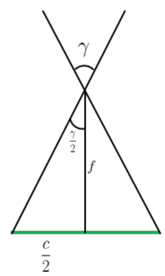

Fig. 3: 2D projection of the diagonal c through the camera lens

into camera object attributes to be used later in creating images. Right ascension, declination, and magnitudes are also saved as camera attributes, but are only used for debugging.

6 Rotation to Camera Frame and FOV Limiting

Before we can determine which stars are in the camera's field of view, all stellar unit vectors ($\hat{n}_*$) must be rotated into the camera's coordinate frame. This involves two rotations, the first from inertial into spacecraft body coordinates, and second from body to camera coordinates. The first is acheived by the $[BN]$ direction cosine matrix, created from the spacecraft MRP, and the second is achieved by the $[CB]$ direction cosine matrix, entered by the user when initializing the camera object. The full conversion is shown in eq. 4.

$$\hat{b}_* = [BN]\hat{n_*} \tag{3}$$

$$\hat{c}_* = [CB]\hat{b}_* = [CB][BN]\hat{n_*} \tag{4}$$

From here, similar triangles are used to determine if each star is in the field of view. The star is in the field of view if the ratio of half the detector width ($\alpha/2$) to focal length (f) is greater than or equal to the ratio of camera frame unit vector coordinates c_2 to c_1 *and* the ratio of half the detector height ($\beta/2$) to focal length (f) is greater than or equal to the ratio of camera frame unit vector coordinates c_3 to c_1 as illustrated in figure 6.

7 Pixel/Line Conversion

Once unit vectors in the camera frame have been calculated, stars can be moved into pixel line space. A similar method to that described above for FOV limiting is used. It is assumed that the ratio of unit vector coordinate values is equal to the ratio of distance from the center of the FOV to the star's location in each of the pixel and line direction. Multiplying that ratio by the number of pixels or lines as appropriate gives the offset in that dimension from the center of the field of view, and adding half the number of pixels/lines moves the center of the coordinate system form the center of the field of view to the top left corner, unifying the camera system with the conventions for image processing and

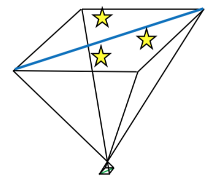

Fig. 4: 3D illustration of the projection of the camera detector through the lens up to the sky.

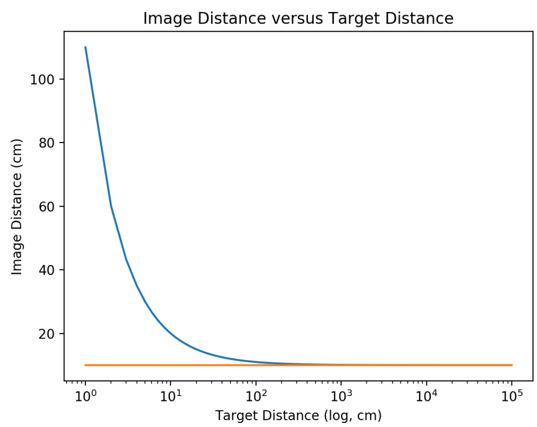

Fig. 5: Image distance quickly approaches object distance, giving justification for using the approximations in this derivation.

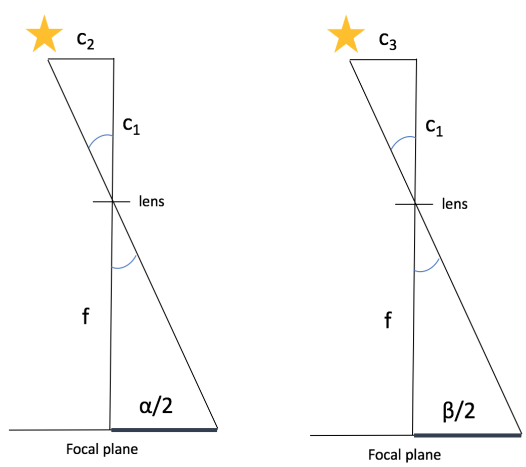

Fig. 6: Demonstration of FOV limiting in the height and width dimensions of the FOV.

navigation pixel/line coordinates.

$$p = -f\frac{c_2}{c_1}\frac{n_{pix}}{width_{detector}} + \frac{n_{pix}}{2} \tag{5}$$

$$l = -f\frac{c_3}{c_1}\frac{n_{pix}}{height_{detector}} + \frac{n_{line}}{2} \tag{6}$$

8 Intensity Calculation

Finally, now that stars are placed on the detector, their intensities can be calculated. This involves calculating a Planck curve at the wavelength values in lambdaSet (using the calculated temperature for the star, see SER 4.1), and multiplying by solidAngleSubtentedBySource (see SER 4.1). This gives a flux value in Watts$/m^2$ at each wavelength value in lambdaSet. Dividing each of these values by the energy per photon at that wavelength ($E_{photon} = hc/\lambda$), we convert to photons per second per m^2. Finally, multiplying by effective area of the detector and integration timestep, we get the total number of photons incident on the detector due to the star. This will further be refined by applying a point spread function (see SER 4.7) and noise model (SER 4.5).