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 4.6: TIME INTEGRATION MODEL

Prepared by	Matt Muszynski

Status: Initial Version
Scope/Contents
Description of the DINO C-REx camera module's object model as it applies to time integration.

Rev:	**Change Description**	**By**
1.0	Initial Release	Matt Muszynski

Contents

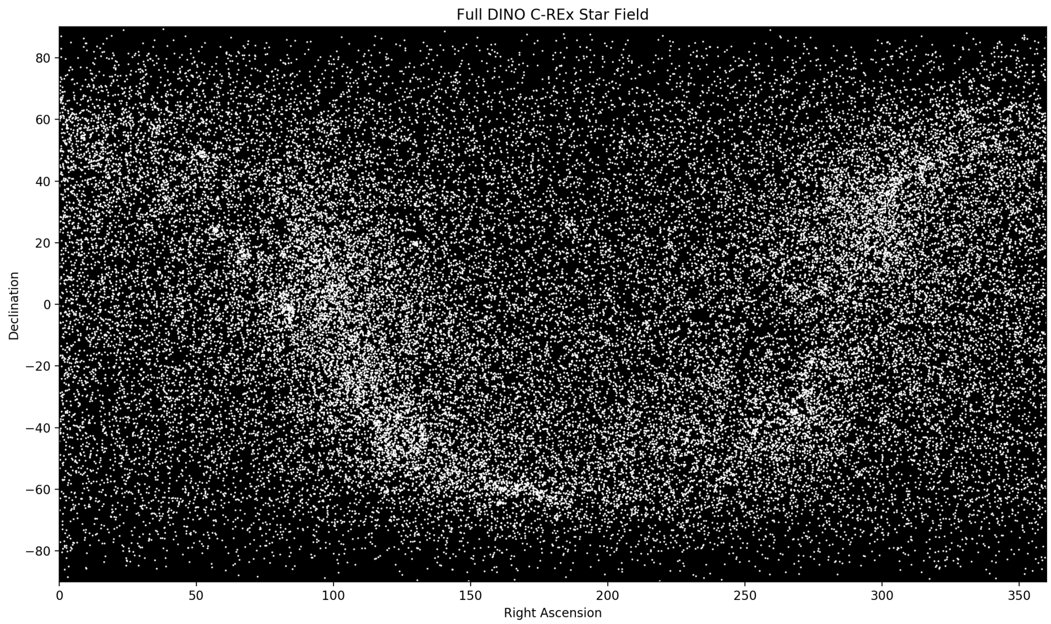

Fig. 1: All Stars in the DINO C-REx Database.

1 Overview

This document gives an overview of how the DINO C-REx camera model handles time integration of images. This functionality was added in order to achieve requirements 4.8.4, 4.8.4.1, and 4.8.4.2:

4.8.4 The system shall model spacraft pointing error during exposures

4.8.4.1 The system shall model spacecraft jitter during exposures

4.8.4.2 The system shall model spacecraft drift during exposures

In order to achieve the smearing behavior characteristic of spacecraft motion during an exposure, the camera module employs a discrete time integration, creating a still frame (called a scene) for each time step of the Basilisk simulation. As the propagation progresses and the camera attitude changes, the stars and beacons in each frame change. When the discrete images are summed at the end of an exposure, the difference in location from frame to frame creates the characteristic smear.[1]

2 Object Model

The object model for DINO C-REx's camera module was created mostly with the optimization of time integration in mind.[2] The model has three classes, each holding more specific data than the last:

[1] However, if the camera slews too far between any two frames, we may see artifacts of this integration method (discrete dots in the final image rather than streaks. For this reason, DINO C-REx has defined maximum slew rates at which it can take a realistic picture. See SER 4.2.

[2] See SER 4.8 for a full description of the camera object model. This document only gives as much detail as is needed to understand the time integration method it describes.

Table 2: camera.updateState() Action Based on Current and Previous takeImage Message.

Current takeImage	Previous takeImage	Action
0	0	None
1	0	Create new image object, record current attitude
1	1	Record current attitude
0	1	Close current image, calculate and sum scene objects, publish detectorArray for Image Processing

camera, image, and scene.

2.1 Camera Class

The camera object is used to hold the basic physical information about the camera and stars in an image. The user defines which spacecraft the camera is a part of and its orientation relative to the body frame of the spacecraft[1] When a camera object is initialized, it loads information for the location and brightness of all stars in the DINO C-REx stellar database, tycho.db. The camera class is designed to hold only information about objects that will persist through any image. For this reason, the camera class has no knowledge of beacons, as they will appear in different locations depending on where they and the spacecraft are in space.

2.2 Image Class

The image class is designed to hold all information that will persist through an entire exposure. This includes a subset of the stars that are stored by the camera object it is a part of as well as the beacons that are visible at any time during the exposure. Because the time scale of exposures is very small compared to the time it takes for a beacon to move appreciably across the sky, it is assumed that all beacons remain stationary for the entire exposure. For this reason, all beacons are added to images. As the exposure progresses, attitudes are recorded to the current image object. Once it is over, the image is computed. This includes copying the subset of stars that are visible at any time during the exposure from the camera into the image object, removing stars that are occulted by beacons, and adding simulated beacons.

2.3 Scene Class

After all objects visible to the camera at any time during the exposure are added to the image object, scene objects can be made. One scene object is made for each propagation time step of the exposure. A subset of star and beacon facet attributes are copied from the data in the image to which the scene belongs. This subset includes only those stars and beacons visible to the camera at the timestep at which the scene is generated.

3 The Nav Executive and the takeImage message

The process for time integration in completely controlled by the navigation executive via the takeImage message. takeImage is a boolean that is published at each time step by the nav executive and is recorded to the camera via camera.updateState(), which takes further action depending on what the message is and what it was at the last time step. There are four actions that may be taken by camera.updateState() based on the current and previous takeImage message.Figure 2 shows the behavior of camera.updateState() under all four conditions.

[1] Among other attributes that are not relevant here, see Doxygen documentation.

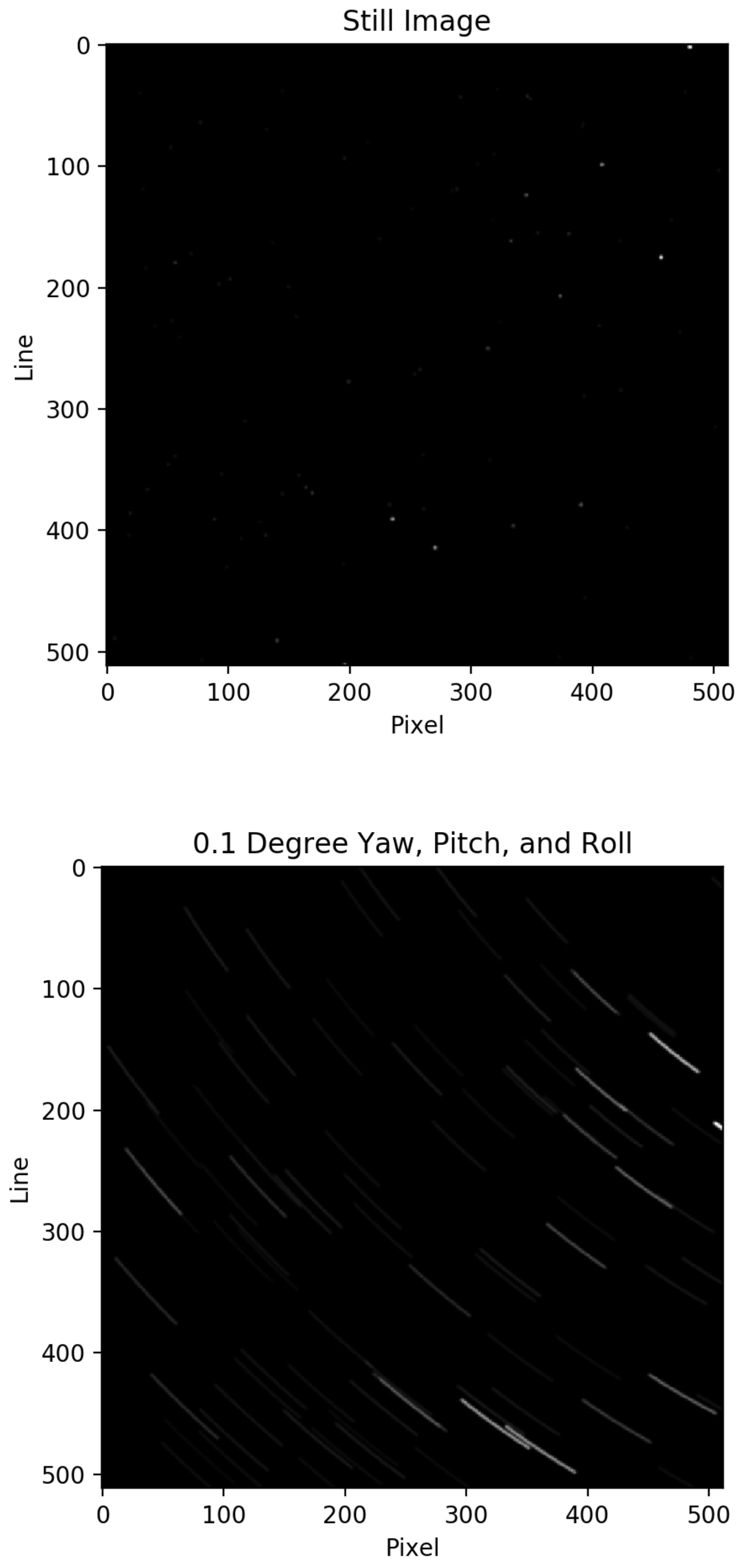

Fig. 2: A single scene (above) and a full image created by summing many scenes.