

Naval Research Laboratory

Washington, DC 20375-5320



NRL/FR/6521-92-9508

**Computer Simulation of a Space-Based
Infrared Surveillance Sensor**

**ROBERT L. LUCKE, WILLIAM A. SHAFFER,
AND RUSSELL B. RHODES**

*Advanced Concepts Branch
Optical Sciences Division*

September 7, 1992

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (<i>Leave Blank</i>)	2. REPORT DATE September 7, 1992	3. REPORT TYPE AND DATES COVERED	
4. TITLE AND SUBTITLE Computer Simulation of a Space-Based Infrared Surveillance Sensor		5. FUNDING NUMBERS PE - ONR/RA11W52	
6. AUTHOR(S) R. Lucke, W. Shaffer, and R. Rhodes		8. PERFORMING ORGANIZATION REPORT NUMBER NRL/FR/6521-92-9508	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Research Laboratory Washington, DC 20375-5320		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Technology Arlington, VA 22217-5000		11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited		12b. DISTRIBUTION CODE	
13. ABSTRACT (<i>Maximum 200 words</i>) This report describes a Naval Research Laboratory software package for the simulation of surveillance by a spaceborne IR sensor. The simulation begins with data taken from airborne platforms and calculates the effects of the atmosphere, the time-varying sensor-to-scene observing geometry, and the optics and focal plane array architecture of the sensor to produce accurately simulated imagery. Signal processing techniques are then applied to suppress clutter and to detect moving airborne targets.			
14. SUBJECT TERMS IR surveillance Computer Simulation		15. NUMBER OF PAGES 24	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED		16. PRICE CODE	
18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT SAR	

CONTENTS

1. INTRODUCTION	1
2. THE END-TO-END SIMULATION PACKAGE (EESP)	1
2.1 Item (1) - Read Background Data	3
2.2 Item (2) - Implant Targets	3
2.3 Item (3) - Implant Clouds	3
2.4 Item (4) - Atmospheric Correction	3
2.5 Item (5) - LOS Pointing and Tracking (see Section 3)	3
2.6 Item (6) - Convert Latitude-Longitude to HR Pixels (see Section 4)	4
2.7 Item (7) - Optical Blur Function (see Section 5)	4
2.8 Item (8) - Integrate over Background Scene (see Section 6)	4
2.9 Item (9) - Add Noise	4
2.10 Item (10) - Add Nonuniformity	4
2.11 Item (11) - Signal Processing (see Section 7)	4
2.12 Item (12) - Track Assembly	6
3. LINE-OF-SIGHT POINTING AND TRACKING (ITEM 5)	6
3.1 Orientation of Gimbal in Satellite	12
3.2 Orientation of Satellite for the Fast Steering Mirror (FSM)	12
4. CONSTRUCTION OF THE GROUND FOOTPRINT (ITEM 5)	13
5. OPTICAL BLUR (ITEM 7)	14
6. INTEGRATE OVER BACKGROUND SCENE (ITEM 8)	15
6.1 Ground-Point Fold-In	15
6.2 Location of HR Pixels Relative to Center of Detector	16
6.3 Sum Up the Contributions of Nearby HR Pixels	17
7. REGISTRATION AND RESAMPLING (ITEM 11)	18
7.1 Displacement Estimation	18
7.2 Resampling the Images	18
8. IMAGE DIAGNOSTICS	18
9. CONCLUSION AND RETROSPECTIVE	19
REFERENCES	20

COMPUTER SIMULATION OF A SPACE-BASED INFRARED SURVEILLANCE SENSOR

1. INTRODUCTION

Simulation of a Naval Research Laboratory space-based infrared surveillance sensor must include such basic effects as the time-varying geometry of the satellite-to-earth line of sight (LOS), atmospheric emission and attenuation, the optical blur of the sensor's telescope, and the noise and nonuniform response characteristics of the solid-state detectors. A user-oriented, menu-driven software package called the End-to-End Simulation Package (EESP) has been developed to perform the necessary calculations. In addition to generating accurately simulated imagery, it includes a set of signal processing algorithms for finding moving targets. The EESP computation begins with real IR data (TIMS [1] and Hi-Camp [2] are the primary sources), accepts user inputs for many parameters (e.g., orbital altitude and position of satellite, pixel size, ground and observation point), and ends with target declarations. In between, *all* effects must be taken into account with sufficient accuracy to give valid results.

The simulated sensor is a satellite-borne, scanning, IR radiometer. The satellite is placed in a circular orbit, unrestricted in altitude. Figure 1 shows the basic geometry of the problem. The linear focal plane array is mounted in a telescope which, at the user's option, can either be placed on a gimbal or use an external steering mirror to achieve LOS pointing control [3]. The linear focal plane array consists of 512 detectors arranged "vertically" and scans "horizontally" across the scene through 512 pixels (pixel \equiv resolution element) by means of a scan mirror. Vertical spacing is under user control (so the detectors can be overlapped), as is the sampling rate in the horizontal (scan) direction.

EESP has demonstrated the capability of reducing background clutter to the sensor noise level for highly structured scenes. Unfortunately, this claim cannot be made for scenes that contain clouds, and hence exhibit cloud parallax motion, but EESP has the capability of studying this problem by implanting moving clouds in a scene. Preliminary investigation indicates that, while clutter due to cloud parallax motion cannot be processed out to the level of sensor noise, it can be drastically reduced.

2. THE END-TO-END SIMULATION PACKAGE (EESP)

In this section, the calculations performed by EESP to generate images and detect targets are broken down into twelve categories and/or capabilities. Most of these "items" (as they will be referred to hereafter) require further explication. This is done partly in later sections of this report and partly in other reports which are referred to as needed. The flow chart provided in Fig. 2 shows the functional relations of the calculational items briefly described below. A few other items, which are on the EESP menus for the purpose of doing image diagnostics, are presented in Section 8.

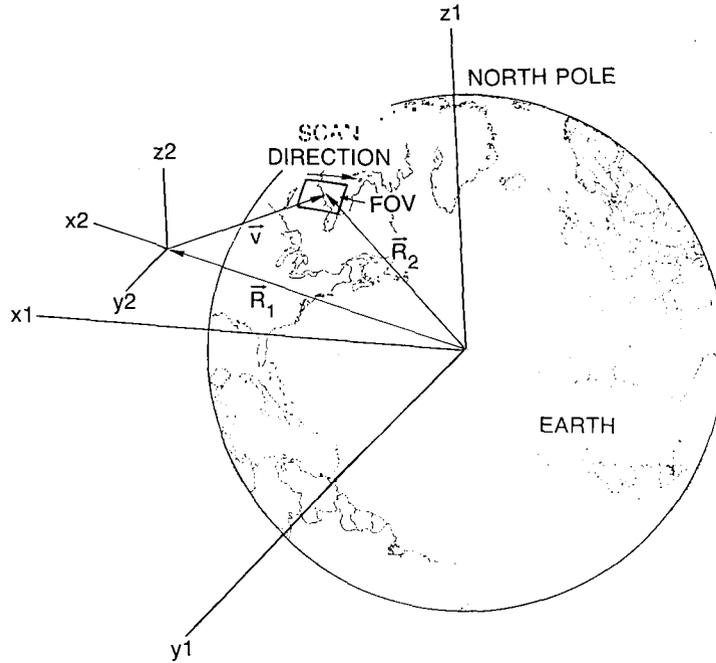


Fig. 1 — The (X_1, Y_1, Z_1) axes are fixed in the body of the Earth. The origin of the (X_2, Y_2, Z_2) axes orbits the Earth; the X_2 direction is always along the local vertical. The satellite is orbiting in the plane of the figure.

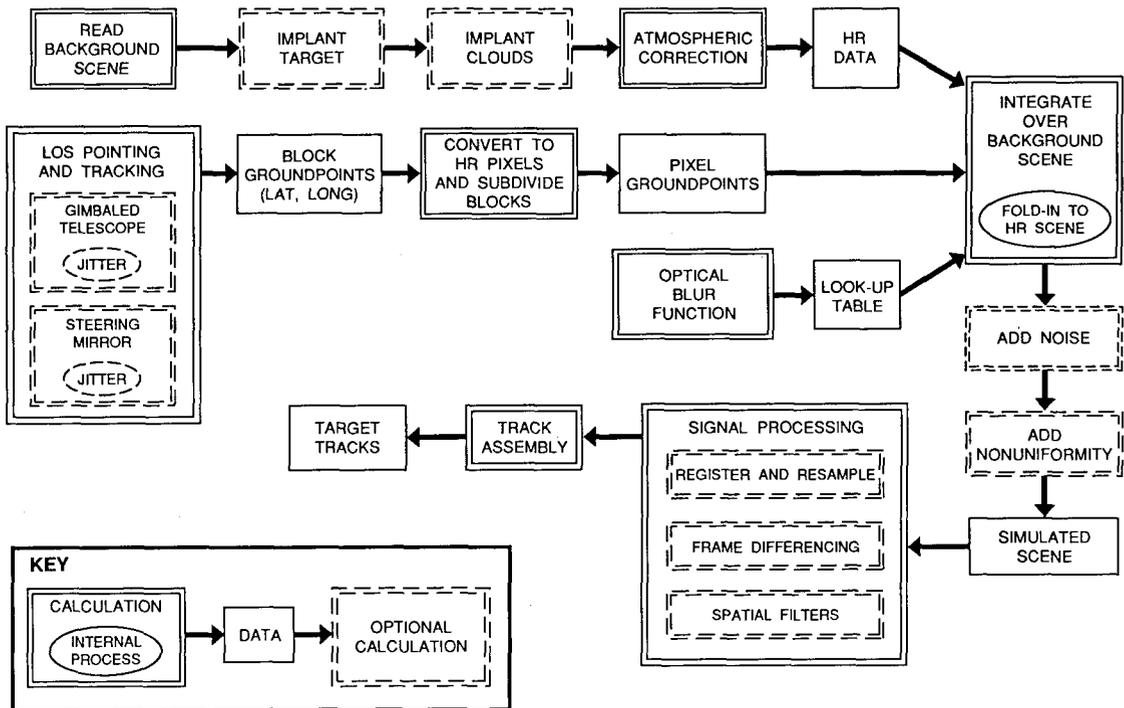


Fig. 2 — Flow chart of the calculations performed by EESP

2.1 Item (1) - Read Background Data

Real infrared data taken in nadir-looking mode from airborne platforms are used as ground truth. TIMS and Hi-Camp are our major sources. The main requirement on this data, other than that it be of good quality, is that it have substantially better resolution than that of the simulated sensor. The 10-m resolution of TIMS and Hi-Camp fulfills this condition with respect to the 50 to 100 m resolution we believe necessary for the space sensor. Test-pattern scenes, such as checkerboards and intensities that are linear functions of position are also available. Hereafter, this background data will be referred to as HR (high resolution) data.

2.2 Item (2) - Implant Targets

Targets can be inserted into the HR data, with user-specified altitude, flight direction, and speed. Targets are modeled by a compact configuration of HR pixels with an appropriate area. Thus, for example, a Backfire bomber, which has an area (viewed from above) of 310 m², is modeled by 3 HR pixels (100 m² per pixel) with unit emissivity and a specified radiance. Since the target is unresolved, a detailed model of its shape is unnecessary.

2.3 Item (3) - Implant Clouds

Very little data exists that can be used to obtain realistic cloud parallax motions. Therefore, software has been written so that artificial clouds can be implanted in the scene. These clouds can be assigned speeds appropriate to their altitudes so that they move across the scene. They emit their own radiation and transmit background radiation according to a user-assigned opacity, which can vary from zero to unity.

2.4 Item (4) - Atmospheric Correction

Atmospheric transmission and path radiance for arbitrary zenith angles are found for the appropriate waveband by using tables of these quantities at particular off-zenith angles (multiples of 10°), for appropriate paths to space (surface-to-space or cloud-to-space), then interpolating as needed. The tables must be calculated separately (with LOWTRAN 6, for example) and entered into the program. The correction equation is

$$D' = (D - R) * T'/T + R'$$

where

D, D' are the original and corrected data, respectively,
 R, R' are the atmospheric path radiances for the aircraft and for the satellite, respectively, and
 T, T' are the atmospheric transmittances for the aircraft and for the satellite, respectively.

Thus, the effect of the atmosphere is first removed from the data for the aircraft altitude and viewing geometry, then reinserted for the altitude and viewing geometry of the satellite.

2.5 Item (5) - LOS Pointing and Tracking (see Section 3)

Double-precision ray-tracing calculations are performed to locate the ground footprints of the array's detectors accurately on the Earth's surface, accounting for the time-dependent geometric effects of the satellite orientation and pointing mechanism. Either a gimbaled telescope or an external steering mirror can be used for LOS pointing; both are dealt with in considerable detail in Section 3. Random LOS jitter can be added as a user-specified rms value (in microradians). At this point, only the locations of the corners of "blocks" (that will later be divided into 64 × 64 pixels) are calculated (see Section 4). They are given in (latitude, longitude) coordinates.

2.6 Item (6) - Convert Latitude-Longitude to HR Pixels (see Section 4)

Block footprint points on the Earth's surface are converted from latitude and longitude to HR pixel coordinates. For this purpose, the HR scene is "pasted" onto the Earth with its lower left-hand corner at the center of the field of view (FOV) (this is explained in detail in Section 6). Once HR coordinates are determined, each block is divided into 4096 (64×64) pixels.

2.7 Item (7) - Optical Blur Function (see Section 5)

Optical blur is simulated by convolving the telescope's blur function (taken to be a Gaussian) with the sensitive area of a square detector. The blurring that results from the scan motion is included by integrating over the positions taken by the blur spot as the scene is scanned across the focal plane. The final blur function (which, because of the scanning motion, is different in the x- and y-directions) is stored in a look-up table for later use.

2.8 Item (8) - Integrate over Background Scene (see Section 6)

For each sample in the simulated scene, the HR pixels that make a significant contribution to a simulated pixel must be located, and their contributions to the sample calculated, as described in detail in Section 6. Location of the relevant HR pixels is accomplished by the "folding in" process described in Section 6.1.

2.9 Item (9) - Add Noise

Detector noise, considered to be white, is added to the simulated data using Gaussian-distributed random numbers with standard deviations specified by the user.

2.10 Item (10) - Add Nonuniformity

Detector-to-detector calibration nonuniformity is added to the simulated data using Gaussian-distributed random numbers with standard deviations specified by the user. Since a scanning sensor is assumed, this means that each horizontal row of samples in the simulated imagery comes from a detector with different responsivity compared to its neighbors. This results, of course, in a "streaky" picture [4].

At this point, a complete simulated picture exists. The example shown in Fig. 3 was obtained from far-IR imagery of the countryside near Mono Lake, CA. A 256×256 -sample simulated image was generated for an assumed satellite altitude of 800 km, with 50 m ground resolution at nadir. However, the scene shown used far off-nadir viewing geometry (grazing angle = 30°), resulting in a ground resolution of $87 \text{ m} \times 175 \text{ m}$ due to the greater range to the tracked ground point and to foreshortening of the scene. There are two samples per pixel in this image; the sampling rate is a user-specified input and is explained in detail in Section 4. Up to 20 pictures can be generated (a larger number may require some code modifications) to obtain a time-sequential string of images for testing signal processing algorithms. The major algorithms in the software package are listed in Sections 2.11 and 2.12.

2.11 Item (11) - Signal Processing (see Section 7)

Signal processing techniques for rendering faint moving targets detectable can be applied in any order or not at all, at the user's option. More algorithms can be added in a modular fashion.

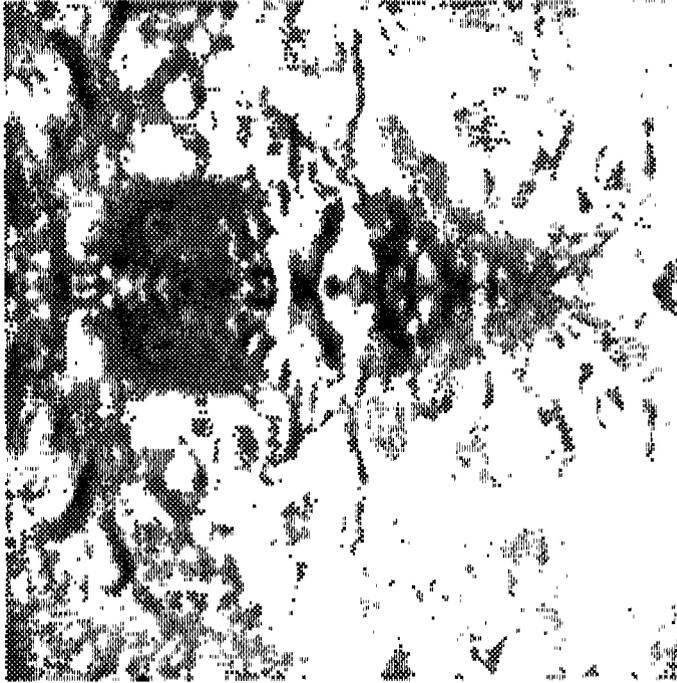


Fig. 3 — A simulated far-IR image of the desert near Mono Lake, CA. The horizontal fold point just above the center is explained in Section 3 (cf., Fig. 10).

2.11.1 Registration and Resampling

Successive pictures are resampled on a common grid. This requires finding the displacement between them [4,5] and applying some sort of interpolator [6,7], and is intended primarily to prepare the data for the Frame Differencing Signal Processor (FDSP). A few additional remarks on registration and resampling are made in Section 7.

2.11.2 Frame Differencing

Subtracting frames of data can be done to any desired order. Stationary backgrounds can thus be subtracted out (with a precision limited by the registration and resampling algorithms indicated above, and by inherent scene distortion) and aircraft in flight detected by their motion: with 50 to 100 m resolution, they are too faint to find any other way. Experience indicates that, for good registration and resampling algorithms, first order differencing is sufficient; higher orders are completely dominated by detector noise and nonuniformity. In the presence of cloud parallax motion, however, second-order differencing appears to be useful.

Figure 4 shows a difference frame generated for the scene shown in Fig. 3, using two images separated in time by 0.5 s. The dark-bright spots ("target dipole") at the center of the picture are the signature of a faint moving target (a Backfire bomber), completely invisible in Fig. 3.

2.11.3 Spatial Filters

Currently, only a median-subtract spatial filter has been implemented. Spatial filters have very little effect on the output of a frame-differencer if scenes can be registered well before differencing; their primary use is for larger, slower, targets than aircraft (e.g., ships).

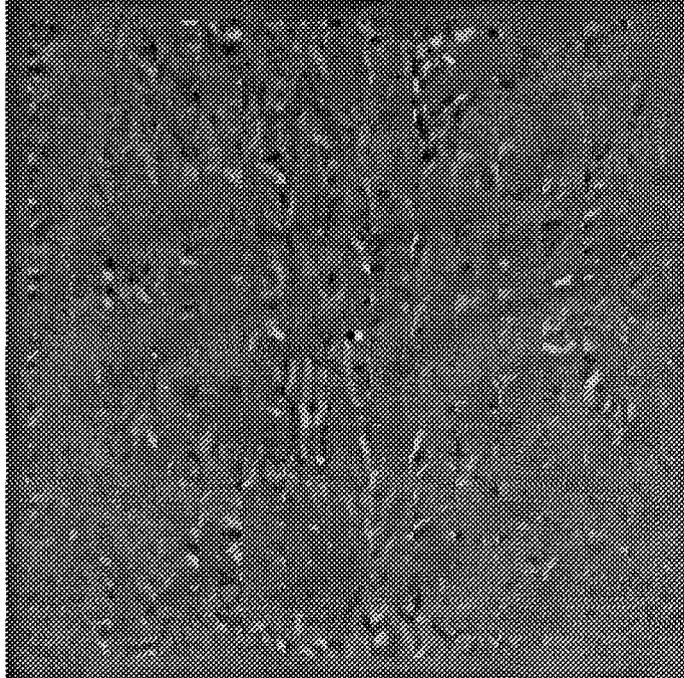


Fig. 4 — The difference between the image of Fig. 3 and another image taken 30 s later. The dipole signature of a moving target is apparent at the center of the scene. Note other targetlike features in the image.

2.12 Item (12) - Track Assembly

The signature of a moving target in a frame differencing signal processor is a “dipole,” a positive contrast feature adjacent to a negative contrast feature. The direction and magnitude of the feature’s separation is determined by the velocity of the target. The tracking algorithm, called the logical tracker [8], searches the first difference frame for all dipoles that correspond to realistic velocities, then checks to see if similar dipoles appear at the places appropriate to their velocities in subsequent difference frames.

Figure 5 shows the result of applying the logical tracker to a sequence of difference frames of which Fig. 3 is the first. The moving target is detected and there are no false alarms, even though Fig. 4 shows a number of point-target-like features.

The items above requiring further explanation are covered in greater depth in the following sections.

3. LINE-OF-SIGHT POINTING AND TRACKING (ITEM 5)

The satellite is placed in a northbound, inclined, circular orbit, at the initial (latitude, longitude) point of (0,0) at $t = 0$. Thus, if the user chooses an orbital inclination of 90° , the satellite is placed in a polar orbit, northbound on the Greenwich Meridian, directly above the equator at $t = 0$ (refer to Fig. 1, which does not, however, show the satellite on the Greenwich Meridian). A different orbital position can be obtained by specifying $t \neq 0$.

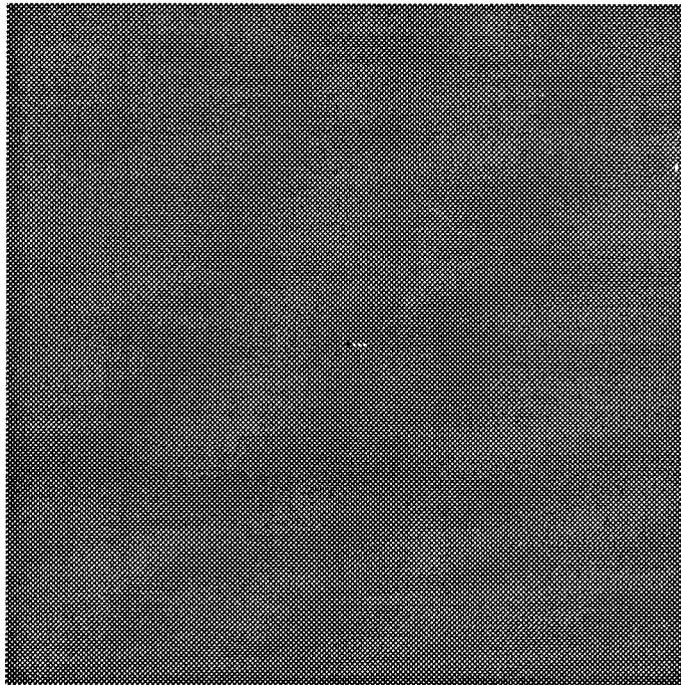


Fig. 5 — The target track obtained from Fig. 4 and subsequent differenced images. The targetlike clutter features of Fig. 4 have been rejected on the basis of their (lack of) motion.

The linear focal plane array is mounted in a telescope which, at the user's option, can either be placed on a gimbal, as shown in Fig. 6, or use a fast steering mirror (FSM), external to the telescope, to achieve LOS pointing control. Except for its optical blur function, the actual properties of the telescope are not used in the calculation. The telescope serves to convert linear measures in the focal plane to angular measures in the FOV; the calculation simply assumes the angular measures with which to begin (mainly this means the angular IFOV of the detectors), and proceeds from there.

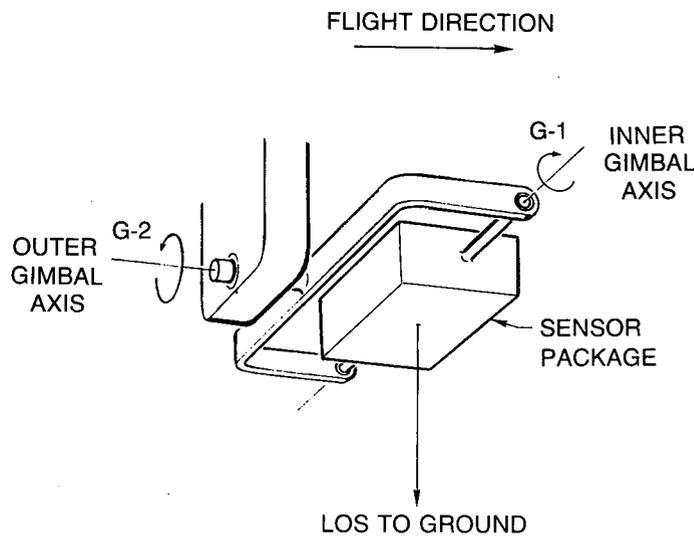


Fig. 6 — The two-axis gimbal moves the sensor package to track the desired ground point. When the pitch, roll, and yaw of the satellite are zero, the outer gimbal axis (G2) is parallel to the flight direction. Scanning is done with a mirror internal to the sensor package.

The linear focal plane array (FPA) consists of 512 pixels (pixel=resolution element=detector size) that are arranged “vertically” and scanned “horizontally” across the scene through 512 pixels by means of a scan mirror. Thus, the “horizontal” and “vertical” directions are the scan and cross-scan directions, respectively. The steering mirror, when used, is separate from the scan mirror. The FPA, telescope, and scan mirror assembly are shown schematically in Fig. 7. Comparison of Figs. 7 and 1 will show the rationale behind the “horizontal” and “vertical” terminology.

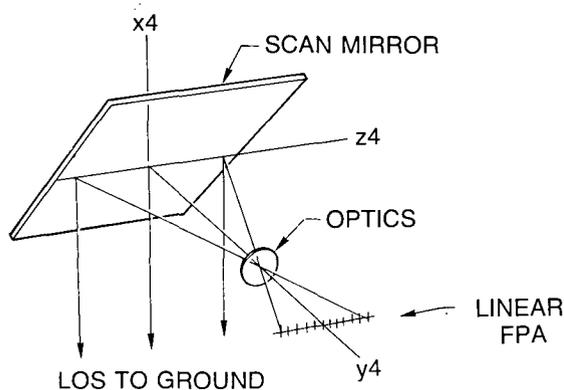


Fig. 7 — The scan mirror rotates about the Z4 axis (which is fixed in the satellite) to scan the LOS from left to right. When the pitch, roll, and yaw of the satellite are zero, Z4 is in the flight direction.

The 512×512 image is the largest the simulator can generate. In fact, the usual practice is to generate only a $64(\text{vert}) \times 256(\text{hor})$ image. One reason for this is to save computing time, because it can take up to 30 min to generate even this small image when the calculation is done for near-horizon viewing geometry (near-nadir viewing takes only 5 to 10 min). The other reason is that the background data used covers only a small part of the footprint of a satellite-borne sensor.

The sensor's FOV is determined by the length of the array and the amplitude of the scan mirror's motion; it is 512×512 pixels. A careful distinction must be made between pixels, which are determined by the size of detectors, and samples, which are determined by how often a detector's output is sampled (in the scan direction) or what the center-to-center spacing of the detectors is (in the cross-scan direction). Physically, of course, the spacing of samples in the vertical direction must be accomplished with some staggered layout of the detectors in the FPA, but this is not something the computer has to worry about. The program can handle more than one sample per pixel (but not less!).

If one sample per pixel is specified, then the final $64(\text{vert}) \times 256(\text{hor})$ image will cover 1/16th of the sensor's FOV. If more than one sample per pixel is specified, then the 64×256 -sample image will cover a smaller fraction of the total FOV. The 64×256 -sample image is chosen so that its right-hand edge coincides with the centerline of the 512×512 pixel FOV.

To solve the pointing and tracking problem, and to calculate ground footprints, a number of coordinate sets and the angular transformations between them will be needed. Four sets of coordinates are used.

(1) (x_1, y_1, z_1) are Earth coordinates, shown in Fig. 1. In terms of radius R , latitude θ (north latitude is positive), and longitude Φ (east longitude is positive), they are

$$x_1 = R \cos\theta \cos\phi,$$

$$y_1 = R \cos\theta \sin\phi,$$

and

$$z_1 = R \sin\theta. \quad (1)$$

For points on the Earth's surface, $R = R_e = \text{Earth radius}$.

(2) (x_2, y_2, z_2) are orbiting coordinates, shown in Fig. 2. Coordinate x_2 is the local vertical, z_2 is the direction of flight. These axes are assumed to rotate so as to track the local vertical as the satellite orbits.

(3) Coordinates (x_3, y_3, z_3) are satellite coordinates. They are related to (x_2, y_2, z_2) by the roll, pitch, yaw (R,P,Y) orientation angles of the satellite, which are zero in nominal orientation. For the gimbaled telescope, LOS pointing is done entirely with the gimbal: R,P,Y remain zero. When the FSM is used, the satellite is maneuvered in pitch and roll to obtain initial orientation (see below).

(4) Coordinates (x_4, y_4, z_4) are sensor coordinates. They are related to (x_3, y_3, z_3) by the gimbal angles G_1 (inner) and G_2 (outer) shown in Fig. 6. When the FSM is used, sensor coordinates and satellite coordinates are the same. Thus, in nominal configuration, the (x_4, y_4, z_4) axes of Fig. 7 coincide with the (x_2, y_2, z_2) axes of Fig. 1.

The following three angles specify the satellite's location; it is assumed that latitude = longitude = 0 at $t = 0$.

- $E_1 = (\text{satellite revolution rate}) \times \text{time} = U_1 t$ is the angular position of the satellite in its orbit, measured from the equator crossing. If the orbital inclination is zero, this is the satellite's latitude.

- $E_2 = (\text{Earth rotation rate}) \times \text{time} = U_2 t$ is the longitude of the equator crossing point of the orbit. If the orbital inclination is zero, this is also the satellite's longitude.

- γ specifies the inclination of the orbit (in degrees). It is actually the complement of the orbital inclination, which is the angle between the plane of the orbit and the equator. Thus, referring to Fig. 1, $\gamma = 0$ describes a polar orbit; for $\gamma = 10^\circ$ (for example), the satellite passes east of the North Pole (yes, we know it's really south, but you get the idea!).

The rotational relations among the four sets of coordinates are given below (translational relations are not needed directly). The sines and cosines of angles are identified by the prefixes "S" and "C," respectively. Thus, for example, $SE_1 \equiv \text{Sin}(E_1)$, $CE_1 \equiv \text{Cos}(E_1)$. Specification of clockwise (CW) or counterclockwise (CCW) rotations about an axis assume that the center of rotation is viewed from the +axis direction. Thus, the usual mathematical definition of a positive rotation coincides with CCW. As will be apparent, the same symbol (e.g. E_1) is often (not always!) used to denote both an angle of rotation and the matrix that effects the rotational transformation. Notations of CW and CCW rotations assume that the angle in question is a positive number and that other angles in the relevant transformation set are zero. For example, E_1 is a CCW rotation around y_1 only if $E_2 = \gamma = 0$; otherwise it is a rotation around some other axis not shown in Fig. 1.

The transformation from Earth to orbiting coordinates is effected as follows:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = E_1 E_\gamma E_2 \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}, \quad (2)$$

where

$$E_1 = \begin{bmatrix} CE_1 & 0 & SE_1 \\ 0 & 1 & 0 \\ -SE_1 & 0 & CE_1 \end{bmatrix} \quad \text{CW about } y_1, \quad (3)$$

$$E_\gamma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\gamma & -s_\gamma \\ 0 & s_\gamma & c_\gamma \end{bmatrix} \quad \text{CW about } x_1, \quad (4)$$

and

$$E_2 = \begin{bmatrix} CE_2 & SE_2 & 0 \\ -SE_2 & CE_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{CCW about } z_1. \quad (5)$$

As noted above, if $\gamma = 0$, then E_γ is the identity matrix, and (E_1, E_2) become (latitude, longitude) of the satellite's position. Naturally, the inverse transform is effected by

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = E_2^T E_\gamma^T E_1^T \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}, \quad (6)$$

where the superscripted T denotes the transpose matrix. The transformation from orbiting to satellite coordinates is given by

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = R P Y \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}, \quad (7)$$

where

$$R = \begin{bmatrix} CR & -SR & 0 \\ SR & CR & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{CW about } z_2, \quad (8)$$

$$P = \begin{bmatrix} CP & 0 & -SP \\ 0 & 1 & 0 \\ SP & 0 & CP \end{bmatrix} \quad \text{CCW about } y_2, \quad (9)$$

and

$$Y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & CY & -SY \\ 0 & SY & CY \end{bmatrix} \quad \text{CW about } x_2. \quad (10)$$

The transformation from satellite to sensor coordinates is effected by rotating through the gimbal angles shown in Fig. 6:

$$\begin{bmatrix} x4 \\ y4 \\ z4 \end{bmatrix} = G1 \ G2 \begin{bmatrix} x3 \\ y3 \\ z3 \end{bmatrix}, \quad (11)$$

where

$$G1 = \begin{bmatrix} CG1 & 0 & -SG1 \\ 0 & 1 & 0 \\ SG1 & 0 & CG1 \end{bmatrix} \quad \text{CCW about } y3 \quad (12)$$

and

$$G2 = \begin{bmatrix} CG2 & SG2 & 0 \\ -SG2 & CG2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{CCW about } z3. \quad (13)$$

At nominal configuration, $G1 = G2 = 0$, the optic axis looks straight out the bottom of the satellite. Positive $G1$ gives a forward look (CCW about $y3$ axis); positive $G2$ gives a leftward look (CCW about $z3$ axis).

The linear array is in the $(y4, z4)$ plane, looking in the $-y4$ direction, as shown in Fig. 7. The scan mirror directs the LOS into the $-x4$ direction and scans in the $(x4, y4)$ plane. If (in nominal configuration) one sits on the satellite, facing forward (the $+z$ direction is forward), and looks down, scanning is done from left to right, as indicated in Fig. 1.

Heretofore, the LOS has referred to the center of the sensor's FOV. Other look directions within the FOV result from specifying the cross-scan location of an arbitrary detector in the linear array by the angle $E1$ ("elevation," not to be confused with $E1 = \text{latitude}$) and the in-scan angular position specified by Az (azimuth). At the center of the scene, $E1 = Az = 0$. The components of the resulting look vector are

$$(vx4, vy4, vz4) = (-CE1 \cdot CAz, -CE1 \cdot SAz, SE1). \quad (14)$$

Orienting the gimbal to point at a specified spot on the ground, or finding where a particular LOS intersects the Earth's surface, is done by solving the vector equation $\mathbf{R}_1 + \mathbf{v} = \mathbf{R}_2$ (see Fig. 1), where \mathbf{R}_1 is the position of the satellite measured from the center of the Earth, \mathbf{R}_2 is the position of the observation point on the Earth, and \mathbf{v} is the vector connecting these two points. Note that $\mathbf{v} = s\hat{\mathbf{v}}$, where s is the distance from the satellite to the Earth observation point, and $\hat{\mathbf{v}}$ is the LOS look direction given in Eq. (14).

To begin the observation, the gimbals must be oriented to direct the LOS to the desired observation point. With \mathbf{R}_1 and \mathbf{R}_2 known, the vector between them is immediately found from $\mathbf{v} = \mathbf{R}_2 - \mathbf{R}_1$, and \mathbf{v} is normalized to $\hat{\mathbf{v}}$, the look direction from the satellite. The solution for $\hat{\mathbf{v}}$ is carried out in Earth coordinates $(x1, y1, z1)$ because both the observed point on the surface of the Earth and the location of the satellite are specified in these coordinates. $\hat{\mathbf{v}}$ is then transformed to satellite coordinates, where its components are $(vx3, vy3, vz3)$, and the gimbal angles required to point the center of the FOV in this direction are given by

$$(vx3, vy3, vz3) = (CG1 \cdot CG2, CG1 \cdot SG2, SG1) \quad (15)$$

or

$$G1 = \arcsin(vz3)$$

$$G2 = \arctan(vy3/vx3). \quad (16)$$

3.1 Orientation of Gimbal in Satellite

Figure 6 is drawn for nominal configuration, that is, $R = P = Y = 0$, so that satellite coordinates (x_3, y_3, z_3) are the same as orbiting coordinates (x_2, y_2, z_2) . When $G_1 = G_2 = 0$, the LOS, after the scan mirror, points straight down, i.e., in the $-z_2$ direction.

3.2 Orientation of Satellite for the Fast Steering Mirror (FSM)

When the FSM is used, the optical layout is as shown in Fig. 7, but with the two-axis FSM in place of the one-axis scan mirror and the FPA-telescope assembly replaced by an FPA-telescope-scan mirror assembly. For an off-nadir observation, the satellite must be oriented so that the LOS is directed to the desired ground point. When the FSM is in its rest position, the LOS points straight out the bottom of the satellite. Thus, orienting the satellite requires a pitch-roll maneuver to point its bottom at the desired ground point. (Yaw could be used but is not: yaw = 0 always.) Then, as time goes on, the mirror steers the LOS to track the fixed ground point; the satellite's orientation is not changed during the observation (except that, as always, it tracks the local vertical). Scanning is done separately by the scan mirror, internal to the telescope.

The satellite attitude solution for this case is done similarly to that for the gimballed sensor, except that sensor coordinates (x_4, y_4, z_4) are never used because the sensor is fixed in the satellite (so $(x_4, y_4, z_4) = (x_3, y_3, z_3)$). Instead, the required look direction \hat{v} is found in orbital coordinates with components (v_x, v_y, v_z) , and the satellite is rotated about its pitch and roll axes to achieve proper orientation. Proper orientation means that the center of the FOV (which always looks straight out the bottom of the satellite) points in the desired direction. The relation between the look direction (in orbiting coordinates) and P and R is

$$(v_x, v_y, v_z) = (-CP \cdot CR, CP \cdot SR, SP), \quad (17)$$

so pitch and roll are found from

$$\begin{aligned} P &= \arcsin(v_z) \\ R &= \arctan(-v_y/v_x). \end{aligned} \quad (18)$$

3.2.1 Mirror Reflections

LOS's from the array are bounced off the scanning mirror and/or the external tracking mirror. In either case, as shown in Fig. 8, the mirror's orientation is described by a normal vector \hat{n} . Given \hat{n} and the incident LOS \hat{v}_1 , the reflected LOS is

$$\hat{v}_2 = v_1 - 2(\hat{v}_1 \cdot \hat{n})\hat{n}. \quad (19)$$

Or, given the two LOS's, the required mirror position is

$$\hat{n} = \frac{\hat{v}_2 - \hat{v}_1}{|\hat{v}_2 - \hat{v}_1|}. \quad (20)$$

3.2.2 Jitter

LOS jitter is implemented for the gimballed telescope by adding randomly chosen angles to G_1 and G_2 . This means, of course, that the directions the LOS moves in response to these jitter angles are mutually perpendicular, but the jitter angle applied to G_2 , say $\delta\alpha$, appears on the LOS as $\delta\alpha \cdot \cos(G_1)$ (G_1 is assumed to be zero in the position shown in Fig. 6). This should be kept in mind if large G_1 's are used. Furthermore, jitter angles are generated only when the block ground points are found (cf. Section 4). This means that only nine (at most) independent pairs of jitter angles are used; they are applied every 64 samples in the scan direction.

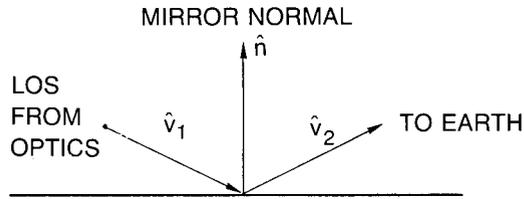


Fig. 8 — The three unit vectors shown represent the mirror normal and the directions of the incident and reflected beams. If any two of these are known, the third can be found from Eq. (19) or Eq. (20).

For the FSM, jitter is implemented by apply the randomly chosen angles to the pitch and roll axes of the satellite. Thus, both appear (essentially) equally on the LOS because it always looks (nearly) straight out the bottom of the satellite, perpendicular to the pitch and roll axes. Otherwise, the implementation is the same as for the gimbaled telescope.

4. CONSTRUCTION OF THE GROUND FOOTPRINT (ITEM 5)

Once the center of the FOV is properly directed, the line of sight from any detector in the focal plane at an arbitrary scan mirror angle (i.e., the LOS to an arbitrary location in the FOV), is given by Eq. (14) as a unit vector in sensor coordinates (x_4, y_4, z_4) . This unit vector is transformed to (x_1, y_1, z_1) coordinates. The point at which this particular LOS intersects the Earth's surface \mathbf{R}_2 must now be found from knowledge of \mathbf{R}_1 and \hat{v} . This is done by writing $\mathbf{v} = s\hat{v}$, noting that $|\mathbf{R}_2|^2 = R_e^2$ (R_e = Earth radius), and solving $|\mathbf{R}_1 + s\hat{v}|^2 = R_e^2$ for s :

$$s = -\mathbf{R}_1 \cdot \hat{v} - \{(\mathbf{R}_1 \cdot \hat{v})^2 - (R_1^2 - R_e^2)\}^{1/2}. \quad (21)$$

(Note that $\mathbf{R}_1 \cdot \hat{v}$ is a negative number because the vectors point in roughly opposite directions.) Finally, $\mathbf{R}_2 = \mathbf{R}_1 + s\hat{v}$ gives the ground intersection point.

LOS ground intersection points are calculated for (corners of) every 64th pixel in the linear optical array, for every 64th sample in the scan direction. That is, ground points are calculated for the corners of blocks that will later be subdivided into 4096 (64×64) pixel locations, one for each of the 4096 samples in that block of the image. The relation between pixels and samples is given by the oversampling factors in the scan (X) and cross-scan (Y) directions, denoted by OSFX and OSFY. Thus, if OSFX = 2, two samples per pixel-length are in the scan direction (i.e., two samples per dwell). With left-to-right scanning, this means the total distance between the left edge of a pixel on the first sample and its right edge on the 64th sample is $(1 + 63/\text{OSFX}) \times d_1$, where d_1 is the horizontal length of a pixel. Similarly, the distance between the top of the first pixel and the bottom of the 64th is $(1 + 63/\text{OSFY}) \times d_2$, where d_2 is the vertical length of a pixel. Horizontal and vertical lengths are equal in the focal plane, but (usually) not when projected on the ground.

For the maximum image size of 512×512 samples (which can cover a considerably smaller ground area than the whole FOV of 512×512 pixels), LOS ground intersection points are contained in two data arrays giving the latitude and longitude of 256 points that divide the ground footprint into 64 separate blocks. These arrays are called X(8,8).C(4) and Y(8,8).C(4); the (8,8) indices label the 64 blocks and the (4) index labels the corners of each block. For the usual image size of 64×256 , only four of these blocks are used. These blocks are then operated on individually by the rest of the program.

The next step is to convert latitudes and longitudes into "HR pixels," that is, units in which the unit of length is the size of a pixel in the background HR data set, and the origin is the fixed stare point, i.e., the center of the 512×512 pixel FOV.

The scan mirror executes a complete scan-and-flyback cycle in a frame period time, usually 0.5 s. Its duty cycle is 80%: the scan motion takes 0.4 s, the flyback takes 0.1 s. The time required to scan through the usual horizontal range of 256 samples is therefore $0.4 \times (256/512)/OSFX$, where OSFX is the oversampling factor in the scan direction. The satellite's position is updated and new gimbal angles (or FSM orientation angles) are calculated for each scan position, that is, for the vertical edges of the 64-sample-long blocks.

The edges of these blocks are then divided into 64 equal increments, which overlap by OSFX and OSFY in the scan and cross-scan directions, respectively. The interior of each block is then gridded into 64×64 overlapping pixel locations, one for each sample. The pixel footprints within a block are all equally spaced. This means that the change in size of the pixel footprints, which should take place gradually over the FOV, instead happens in jumps at the block boundaries. Except, perhaps, very near the horizon, where severe footprint distortions take place, this approximation will not significantly affect imagery.

5. OPTICAL BLUR (ITEM 7)

The optical blur function is chosen to simulate matched optics, which is precisely defined to mean that the diameter of the first zero-intensity circle of the theoretical diffraction pattern for an unobscured circular aperture is equal to the width of the detector (which is square). Since reality will not match this precise definition, the optical blur function is taken to be Gaussian. The standard deviation of the Gaussian is chosen so that 70% (as opposed to the theoretical value of 85%) of the light falls within a detector when the blur spot is centered on it. In detector-width units, this is $\sigma = 0.36$.

Referring to Fig. 9, we see that the contribution to a detector from an HR pixel with unit intensity located distances u, v from the detector's center is

$$D(u,v) = \int_{-u-1/2}^{-u+1/2} \int_{-v-1/2}^{-v+1/2} B(x,y) dx dy, \quad (22)$$

where

$$B(x,y) = \frac{1}{2\pi\sigma^2} \exp \left[-\frac{x^2 + y^2}{2\sigma^2} \right] \quad (23)$$

is the assumed optical blur function, which is Gaussian (with the same σ) in both its arguments. The shape of the HR pixel (which, as shown in Fig. 9, will usually not be square) does not enter this expression - it is sufficiently smaller than the blur circle ($\leq 1/6$) that its shape makes no appreciable difference, so it is treated as a point source. The condition that 70% of the energy in the blur circle falls within a detector becomes $D(0,0) = 0.7$, and is fulfilled for $\sigma = 0.36$.

$D(U,v)$ can be expressed *analytically* as a combination of products of error functions. It can be expressed *usefully*, to a very good approximation, as the same Gaussian form as $B(x,y)$, but with $\sigma_D = 0.50$.

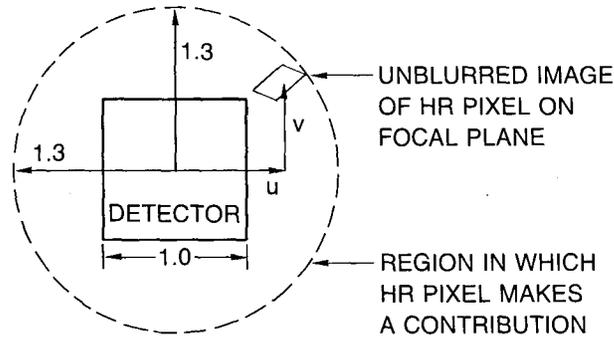


Fig. 9 — Only those high-resolution pixels whose centers lie within 1.3 detector-widths of the center of a particular detector contribute to that detector's output.

The optical blur caused by the scanning motion can be taken into account by noting that the motion is in the u direction only (in the focal plane coordinates of Fig. 9) and that the image on the focal plane moves a distance $1/OSFX$, where unity is the width of a detector. Thus, the contribution of an HR pixel becomes the average of $D(u,v)$ over the distance the image moves:

$$C(u,v) = OSFX \int_{u-1/2 \cdot OSFX}^{u+1/2 \cdot OSFX} D(w,v)dw. \quad (24)$$

Look-up tables are calculated for the function $C(u,v)$ and its derivatives. Note that $C(u,v)$ is separable: $C(u,v) = C_x(u) \cdot C_y(v)$. Thus, for arbitrary u,v ,

$$C(u,v) = [C_x(u_i) + (u-u_i) \cdot C'_x(u_i)] \cdot [C_y(v_j) + (v-v_j) \cdot C'_y(v_j)] \quad (25)$$

where i and j denote values stored in the tables, and $'$ denotes derivative. That is, $C(u,v)$ is given by a two-term Taylor series for both its arguments. By using this approach, values for $C(u,v)$ as a continuous function can be found without doing the integral in Eq. (22) each time.

6. INTEGRATE OVER BACKGROUND SCENE (ITEM 8)

At this point, the ground footprint of every detector is known for every sample that will appear in the simulated scene. For each sample, the program must now locate those HR pixels that will contribute significantly to that sample and calculate their contributions.

6.1 Ground-Point Fold-In

A basic problem with ground truth data is that most existing databases with acceptable resolution come from aircraft-borne instruments, hence do not cover as large an area as a satellite-borne sensor. Typically, as the airplane flies, it generates data covering a scene that is long in one direction but short in the other (although even the long dimension is usually not long enough, rarely being more than 10 to 20 km). These scenes must be used to tile a larger region for our satellite simulations. This is done, conceptually, by "folding out" the aircraft data so that the resulting large scene does not have discontinuities at the edges of the original small scene. In actuality, this folding out of the HR data is done by separately "folding in" the x - and y -indexes of particular points in the simulated scene, as shown schematically in Fig. 10.

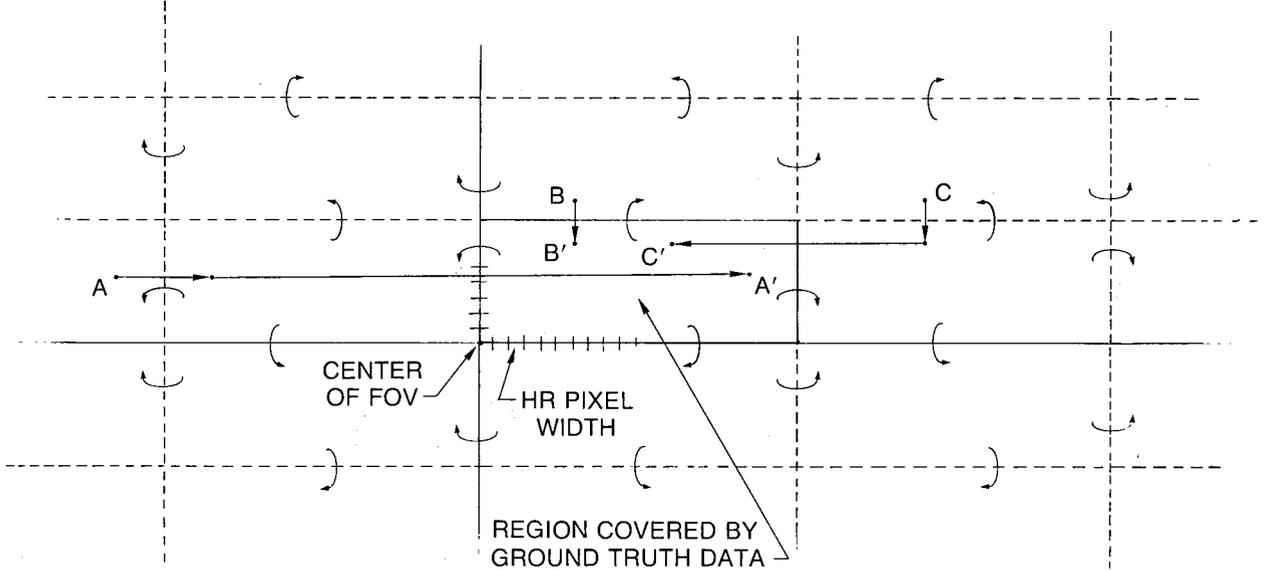


Fig. 10 — A, B, and C represent arbitrary points in the simulated scene. They are folded into the region covered by the high-resolution data in the manner shown.

To see how this is done, we consider the background scene to cover an interval $[0, N]$ along the x axis with pixels labeled n , $1 \leq n \leq N$ (the origin is the center of the sensor's 512×512 pixel FOV). An arbitrary positive point x can be specified by $x = M.d$, where M and d are its integer and decimal parts, respectively. We want to find the new point x' , $0 \leq x' \leq N$, into which x folds. If $0 \leq M \leq N$, then, trivially, $x' = x$. But if $M \geq N$, we must first evaluate $K = T(M/N)$, using the computer's real-to-integer truncation function, $T(\pm \text{integer.decimal}) = \pm \text{integer}$. If K is odd, then the new point is $x' = (K+1)N - x$ (the interval containing x will be reversed when it is folded into the basic $[0, N]$ interval). If K is even, the new point is $x' = x - KN$. For example, taking $N = 100$ and $x = 127.3$, we find that $K = 1$ and $x' = 72.7$, which is correct. Similarly, if $x = 227.3$, then $K = 2$ and $x' = 27.3$.

For an arbitrary negative point, $x = -M.d$, the appropriate formulas are $x' = x + (K+1)N$ for K odd, and $x' = -KN - x$ for K even. In this case, the reversal of the interval occurs for K even. Thus, for $x = -27.3$, $K = 0$ (which is even) and $x' = 27.3$. For $x = -127.3$, $K = 1$ and $x' = 72.7$.

6.2 Location of HR Pixels Relative to Center of Detector

The location of HR pixels must be found in focal plane coordinates. Up to now, the program has located focal plane coordinates on the ground in units of HR pixels. This requires a transformation between the sets of coordinates shown in Fig. 11, where \hat{i} and \hat{j} are perpendicular unit vectors in the plane of HR pixels and \hat{v}_1 and \hat{v}_2 are unit vectors (specifying the edges of detectors) that are perpendicular on the focal plane.

Angles θ_1, θ_2 are defined as shown in Fig. 11, so that

$$\hat{v}_1 = \cos\theta_1 \hat{i} + \sin\theta_1 \hat{j}$$

and

$$\hat{v}_2 = \cos\theta_2 \hat{i} + \sin\theta_2 \hat{j}. \quad (26)$$

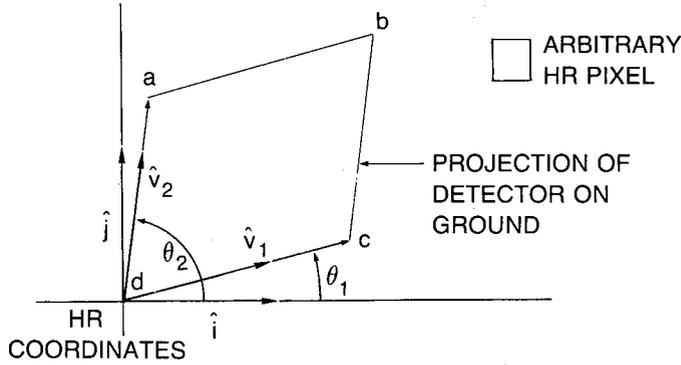


Fig. 11 — This figure shows (schematically) the same elements shown in Fig. 9, but projected on the ground instead of projected on the focal plane.

\hat{v}_1, \hat{v}_2 are not perpendicular, but they span the x, y plane, so an arbitrary HR pixel at $x\hat{i} + y\hat{j}$ can also be described as having the location $a\hat{v}_1 + b\hat{v}_2$, where a and b are found by solving $a\hat{v}_1 + b\hat{v}_2 = x\hat{i} + y\hat{j}$ to get

$$a = \frac{x\sin\theta_2 - y\cos\theta_2}{\sin(\theta_2 - \theta_1)} \tag{27}$$

$$b = - \frac{x\sin\theta_1 - y\cos\theta_1}{\sin(\theta_1 - \theta_2)} \tag{28}$$

Because of the Earth's curvature, a simulated pixel projected on the ground is *slightly* different from a parallelogram, so opposite sides are averaged to find \hat{v}_1 and \hat{v}_2 .

The last step to obtain the focal plane position of an HR pixel is to divide a and b (from Eqs. (27) and (28)), which are in units of HR pixels, by the magnitudes of \hat{v}_1 and \hat{v}_2 to convert them to focal plane units. The resulting numbers (u, v) can now be checked to see if they are within the circle of radius 1.3 that will make a significant contribution to the detector in question.

6.3 Sum Up the Contributions of Nearby HR Pixels

The contributions from all relevant HR pixels must now be added up to find the signal on the detector. These contributions are determined by the location of each HR pixel in relation to the location of the detector ground footprint for the sample under consideration. The necessary values are found from the look-up tables generated for $C(u, v)$, explained in Section 5.

In principle, of course, $C(u, v) \neq 0$ no matter how large u, v are, so all HR pixels should be summed over. To save computing time, a cut-off distance of 1.3 is imposed, as indicated in Fig. 9; only HR pixels whose centers lie inside the circular region $[-1.3 \leq (u^2 + v^2)^{1/2} \leq 1.3]$ are summed over. For $\sigma = 0.36$ ($\sigma_c = 0.50$), this includes about 97% of the area under the function $C(u, v)$, but only about 90% of the power in a real, nearly diffraction-limited, blur function. Thus, about 10% of the optical power is not handled correctly, and the influence of an extremely bright region will not be felt far and wide across the focal plane, as in reality it would.

Since the number of HR pixels that happen to fall inside the acceptable region is variable, the sum of their contributions is normalized as follows:

$$\frac{\sum_{m=1 \dots \text{pixels}} \text{Signal}(m) * \text{Blur}(m)}{\sum_{m=1 \dots n \text{ pixels}} \text{Blur}(m)}, \quad (29)$$

where $\text{Blur}(m)$ is the value of $C(u,v)$ for the HR pixel labeled m . The range of 1.3 is chosen to give extremely accurate simulations, having a precision of about $0.01 \mu\text{F}$ for input values around 500 to $1000 \mu\text{F}$. This is far below the usual noise level of $0.3 \mu\text{F}$. Thus, while extreme precision may be desired for some purposes (exploring the ultimate power of algorithms, for example), computing time can be reduced for ordinary applications by reducing the range to about 1.

7. REGISTRATION AND RESAMPLING (ITEM 11)

To prepare the data for an FDSP, successive images must be accurately registered and resampled. The first step in the process is to determine the displacement between images.

7.1 Displacement Estimation

Because different parts of the scene move by different amounts, the scene is broken up into 32×32 -sample blocks, over which displacement is acceptably uniform. The measured displacement is then used to resample the central 16×16 subblock, and another 32×32 block is done. Over this small region, displacement differs insignificantly from a translation. Displacement is found by assuming the satellite has a navigation and orientation system (composed of star trackers and precision gyroscopes) capable of determining (to within 10 to $20 \mu\text{rad}$) where each pixel is located on the ground. The residual 10 to $20 \mu\text{rad}$ misregistration can then be found by applying a gradient technique to the radiance differences of the two scenes, or by applying a Fourier technique [5]; the former is the more accurate of the two, as currently implemented. Even greater accuracy can be obtained by implementing a direct search that minimizes the clutter in the resulting difference frame [4]. The computer keeps track of exact ground positions for all images, so the accuracy of various resampling methods can be readily assessed.

7.2 Resampling the Images

Once the displacement between two images has been determined, they must be resampled on a common grid. Currently, this is done with filtering interpolators [6], which achieve excellent clutter suppression by applying a spectral filter to the unshifted frame that matches the filtering effect of applying the interpolator to the frame that is shifted. The result, for most cloud-free scenes, is clutter reduction below the level of sensor noise. Usually, only two images are compared (that is, frame differencing is not done to a higher order than one); consequently, some improvement can be realized by implementing Dual Difference Filters, which constitute the optimal approach to this problem [7].

8. IMAGE DIAGNOSTICS

EESP contains some options which do not contribute directly to the generation of images or the detection of targets, but can assist the user in image analysis.

Whole-image histograms can be generated and/or line plots made of individual horizontal or vertical slices through the scene; these features are standard and require no special explanation. Also provided is an option that calculates and displays the standard deviations of 32×32 subblocks of the image. Each

standard deviation is calculated using the subblock mean, so the result is an indication of which parts of the scene contain the most variable structure, hence should be the strongest sources of clutter.

But better measures of clutter are available. The Cumulative Statistics option attempts to give a clear idea of the clutter content of a scene by using as few numbers as possible. It lists the maximum, minimum, mean, and standard deviation (σ) of an image. It also gives Δ_x and Δ_y , which are the root-mean-square values of the difference between adjacent samples. The defining equations for these quantities are

$$\Delta_x = \left\{ \sum_{ij} (s_{i,j+1} - s_{i,j})^2 \right\}^{1/2} \quad (30)$$

and

$$\Delta_y = \left\{ \sum_{ij} (s_{i+1,j} - s_{i,j})^2 \right\}^{1/2}, \quad (31)$$

where s_{ij} denotes the value of sample (i,j) and the sum is over all samples in the image, with the obvious exception of those samples at the edge for which $i + 1$ or $j + 1$ falls outside the image. (This edge effect applies to many of the image processing algorithms used by EESP, especially resampling, and it is for this reason that some displayed images, such as Fig. 4, have the outermost eight samples blanked out or replaced by the scene average.) The reason why Δ_x and Δ_y are given separately is that images need not be isotropic, and certainly won't be when the sampling rates in the x and y directions are different. Note that j labels the x , or horizontal, direction, i labels the y , or vertical, direction ($i = j = 1$ at the upper left-hand corner of the image).

Δ has been found to be a better indicator of scene clutter than σ because it is a *local* measure of variability. A scene that is substantially brighter in one region than another can have a large value for σ because it exhibits large variations from the mean, but, if the radiance grades smoothly from one region to the other, it will not present a real challenge to a Frame Differencing Signal Processor (FDSP). Clutter leakage through an FDSP depends exclusively on the local properties of the image. Sharp edges, for example, often "leak through" an FDSP, producing artifacts in the differenced image. (However, our work has shown that, given reasonable sampling rates and good registration and resampling algorithms, even the clutter leakage from sharp edges can be reduced to below the level of sensor noise.)

The Δ 's are, of course, closely related to the radiance gradients in the scene. If $R(x,y)$ represents the scene radiance and $G_x = dR/dx$, then $\Delta_x \approx G_x \cdot dx$, where dx is the separation between samples in the x direction. Similarly, $\Delta_y \approx G_y \cdot dy$. Thus, Δ is an rms measure of the local rate of change in a scene, and this is a much more important quantity in determining its clutter characteristics than is σ .

9. CONCLUSION AND RETROSPECTIVE

The major features of the End-to-End Simulation Package (EESP) have been described. It is a versatile, easily modified means of generating accurately simulated imagery and of assessing signal processing algorithms. In another report, EESP has been used to demonstrate the feasibility of IR detection of bomber-sized airborne targets from space-borne platforms, with high reliability and very low false alarm rates [4,8].

The software package described here is the culmination of an intensive, five-year effort by the Advanced Concepts Branch personnel. Its earliest roots trace back to work done a decade ago at NRL to assess the prospective usefulness of the Defense Advanced Research Projects Agency (DARPA) proposed demonstration of IR detection from space, called Teal Ruby. An important lesson of this work

is that the analyst and the programmer must each be aware of the other's problems. Early failures to observe this rule resulted in problems which, when corrected, cut the running time needed for a VAX 750 to generate a nadir-looking frame from two hours to fifteen minutes!

The first author has been involved in this effort from the beginning and is acutely aware of the tremendous amount of analytical and programming work that has been expended to achieve the current level of capability. But when all is said and done, I am reminded of the immortal words of Omar Khayyam:

Up from Earth's center through the seventh gate,
I arose and on the throne of Saturn sate.
And many a knot unravelled on the road,
But not the master-knot of human fate.

REFERENCES

1. F.D. Palluconi and G. R. Meeks, "Thermal Infrared Multispectral Scanner (TIMS): An Investigator's Guide to TIMS Data," JPL 85-32, Jet Propulsion Laboratory, Pasadena, CA, June 1985.
2. B. A. Buckley and M. L. Burnell, "Hi-Camp II User's Handbook," Lockheed Rept. LMSC-F044351 Rev. A, Oct. 1987.
3. R.L. Lucke, "Hardware Image Motion Compensation for a Scanning, Space-Borne Sensor," *Infrared Phys.* **30**, 121-141, 1990.
4. W. Shaffer and R. Lucke, "Simultaneous Registration and Nonuniformity Correction of Space-Based IR Images for a Scanning Sensor," NRL Report NRL/FR/6521-92-9390, May 1992.
5. A. Schaum and M. McHugh, "Analytic Methods of Image Registration: Displacement Estimation and Resampling," NRL Report 9298, Feb. 1991.
6. R. Lucke and A. Stocker, "Filtering Interpolators for Frame Differencing Signal Processors," submitted to *IEEE Trans. Sig. Proc.*
7. A.P. Schaum, "Dual Differencing Filtering for Change Detection in Digital Imagery," *IEEE Image Proc.*, in press.
8. W.A. Shaffer, R.L. Lucke, and M.J. McHugh, "A Logical Tracker for the Detection of Moving Targets in IR Images," NRL Report (in publication).