Camera Estimation for Orbiting Pushbrooms

Rajiv Gupta and Richard I. Hartley

General Electric Corporate R&D
River Rd, Schenectady, NY 12309, USA

Abstract: Several space-borne cameras use pushbroom scanning to acquire imagery. Traditionally, modeling and analyzing these sensors has been computationally intensive due to the motion of the orbiting satellite with respect to the rotating earth, and the non-linearity of the mathematical model involving orbital dynamics. A new technique for mapping a 3-D point to its corresponding image point that leads to fast convergence is described. Besides computational efficiency, experimental results also confirm the accuracy of the model in mapping 3-D points to their corresponding 2-D points.

1 Pushbroom Sensors

The pushbroom principle is commonly used in satellite cameras for acquiring 2-D images of the Earth surface. SPOT satellite's HRV camera is a well-known example of a pushbroom system [1]. In general terms, a pushbroom camera consists of an optical system projecting an image onto a linear array of sensors. At any time only those points are imaged that lie in the plane defined by the optical center and the line containing the sensor array. This plane will be called the instantaneous view plane or simply *view plane* (see [2] for details).

This optical system is mounted on the satellite and as the satellite moves, the view plane sweeps out a region of space. The sensor array, and hence the view plane, is approximately perpendicular to the direction of motion. The magnitude of the charge accumulated by each detector cell during some fixed interval, called the *dwell time*, gives the value of the pixel at that location. Thus, at regular intervals of time 1-D images of the view plane are captured. The ensemble of these 1-D images constitutes a 2-D image. It should be noted that one of the image dimensions depends solely on the sensor motion.

It is well known that the standard photogrammetric bundle adjustment typical of aerial imagery does not work for satellite imagery [3, 2]. Even if one were to model the ortho-perspective nature of the imagery, classical space resectioning is unable to separate the correlation among the unknown parameters. For accuracy, and in fact convergence, a pushbroom camera model must explicitly take into account the constraints imposed by: (1) the Kepler's Laws, (2) the rotation of the earth, and (3) the constraints imposed by the ephemeris data.

For satellite cameras the task of finding the image coordinates of a point in space is relatively complex and computationally intensive because many of intermediate steps force the use of approximate or iterative schemes. For instance, there is no closed-form expression determining the angular position of an orbiting satellite given its time of flight from any given point in its orbit (e.g., time of flight from perigee). Because of this, the exact computation of the image produced by a pushbroom sensor has traditionally been a time consuming task.

This paper describes a general methodology for estimating parameters of a pushbroom camera that alleviates the problems mentioned above. A new technique for efficiently mapping a 3-D point to its corresponding 2-D image coordinate is described. Despite the non-linearity of the mathematical model, our scheme exhibits fast convergence: in most cases we obtained the desired accuracy in one or two iterative steps. Experimental results also confirm the accuracy of the model in mapping 3-D points to their corresponding 2-D points.

The model described here has been implemented for the SPOT satellite's HRV cameras. Even though some of the terminology used refers specifically to SPOT, the model is applicable to all pushbroom cameras. To that extent, SPOT is just an example application.

The overall camera parameter estimation process can be divided into two main tasks, a modeling task and an optimization task.

Modeling Task. Before we can estimate the parameters of a camera, we have to implement a software model of the camera. This software model transforms a point in the world coordinate system (given, for example, as [lat, lon, elevation]) into a pixel location (u, v) of the same point, in accordance with parameters and mechanisms of the camera. A camera modeling routine essentially mimics the operation of the camera as it transforms a ground point into a pixel in the image.

Optimization Task. If a routine to transform any 3-D point (x, y, z) into its image coordinates (u, v) is available, one can formulate the camera parameter estimation problem as a global optimization problem. In this formulation, the parameters of the camera are the unknown variables while the *ground control points* and the ephemeris information collected by the on-board sys-

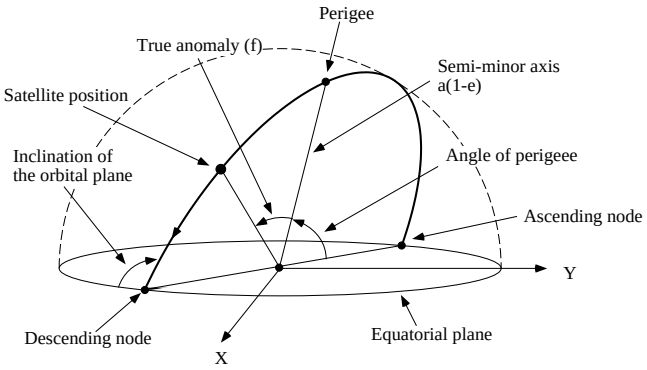

Figure 1: The six orbital parameters.

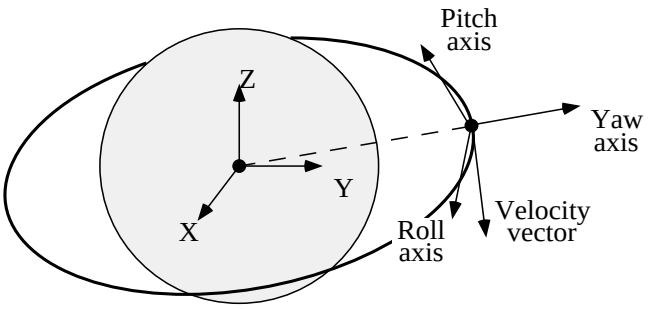

Figure 2: The local orbital frame.

tems provide inputs and constraints. The overall task is to compute a set of camera parameters that minimize the least-squared-error between the given and computed pixel values for each ground control point while abiding by all the orbital constraints.

The optimization method used in our implementation is based partly on the well known Levenberg-Marquardt (LM) parameter estimation algorithm [4]. Our extensions to the basic LM algorithm include methods to handling sparsity in the Jacobian matrix and its customization for the camera parameter estimation problem. These implementation details are beyond the scope of this paper.

2 Camera Parameters

All the parameters needed for modeling the camera have a predetermined nominal value which is known prior to the launch. For some parameters, since they are continuously monitored by the on-board systems, a more accurate value is provided to the user as ephemeris and other auxiliary information. Nevertheless, for the sake of greater accuracy it has proven necessary to refine the ephemeris data and estimate all the parameters simultaneously by solving the overall mapping problem using ground-control points and orbital constraints. It is useful to classify the camera parameters into three classes: *known parameters*, *independent parameters*, and *dependent parameters*.

Known Parameters. Constants for the planet around which the satellite is rotating (typically Earth) are assumed to be known a priori. These parameters include: semi-major, semi-minor axes and the eccentricity of the planet (which is assumed to be ellipsoidal with uniform density), planet's GM constant (gravitational constant times mass), length of sidereal day, and the longitude of the first descending node.

Independent Parameters. The exact position of a satellite in its orbit is fully described by the following six parameters (Fig. 1 and [5]): (1) semi-major axis of the orbital ellipse a, (2) orbital eccentricity e, (3) inclination of orbital plane with respect to the equatorial plane i, (4) geocentric angle between the perigee and the ascending node ω, (5) longitude of the ascending node λ_{AN_k} or the descending node λ_{DN_k}), and (6) true anomaly f.

The first two parameters determine the elliptical shape of the orbit. The third and fourth parameters fix this ellipse in space with respect to the equatorial plane. The fifth parameter, λ_{AN_k} (or, equivalently, λ_{DN_k}), registers the k-th orbital track with the rotating earth. Because of the rotation of earth, the equator crossing of the satellite drifts westward by a fixed known amount with each revolution. The true anomaly f denotes the angular position of the satellite from the perigee. In this list, this is the only time dependent parameter; all others can be assumed to be fixed for any given track of the satellite.

The above orbital parameters specify the position of the camera platform. In order to specify the orientation or the pose of the camera the following reference frames are needed.

A **Local Orbital Frame** is defined at every point in the orbit as follows (see Fig. 2). The origin of the frame is the satellite's center of mass; the *yaw axis* is the geocentric vector pointing radially away from the Earth center; the *roll axis* is in the orbital plane perpendicular to the yaw axis, along the velocity vector; and *pitch axis* is perpendicular to both yaw and roll axes.

The **Satellite Attitude Reference Frame** is fixed with the satellite. Nominally it is aligned with the local orbital reference frame as follows: the X axis is along the pitch axis, the Y axis is aligned with the roll axis and the Z axis is aligned with the yaw axis. The angles between the attitude frame and local orbital plane are used to orient the satellite.

The complete orientation of the satellite is computed in two parts: (1) the attitude or the *look direction* of each pixel in the detector array within the satellite attitude reference frame, and (2) the orientation of the attitude reference frame with respect to the local orbital reference frame.

First we specify the *look direction* of each detector element. It is customary to specify the look direction by two angles: Ψ_x and Ψ_y (Fig 3). Ψ_x represents the rota-

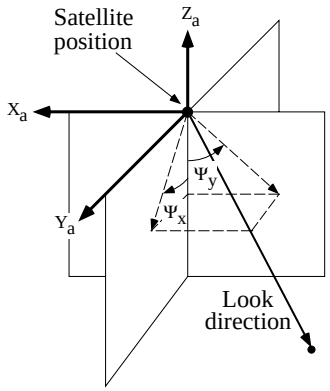

Figure 3: Look angles Ψ_x and Ψ_y.

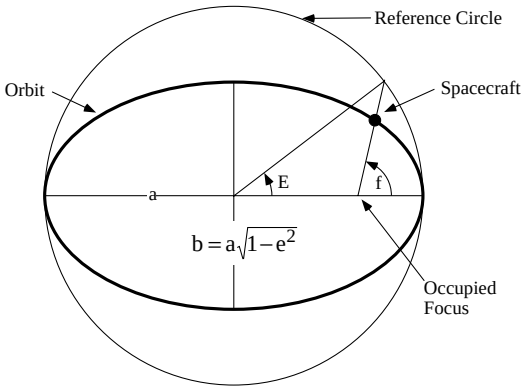

Figure 4: Parameters of an ellipse.

tion that causes the satellite to look forward or backward along the direction of flight; Ψ_y is the rotation perpendicular to it. More precisely, the first angle Ψ_x is the angle made by the orthogonal projection of the look direction in the Y-Z plane with the negative Z axis of the satellite attitude reference frame. If the camera is pointed towards the nadir, this angle is zero; a non-zero Ψ_x makes the satellite look forward or backward along the ground track. Similarly, Ψ_y is the angle that the orthogonal projection of the look direction vector, projected in the X-Z plane, makes with the negative Z axis. In nadir viewing, Ψ_y is zero for the central pixel; it gradually increases for detectors looking eastward, and decreases for detectors looking westward (see [1]).

Given Ψ_x and Ψ_y, the unit vector along the look direction in the attitude reference frame is given by $U = [\tan\Psi_y, \tan\Psi_x, 1]/\sqrt{1+\tan^2\Psi_x+\tan^2\Psi_y}$. The look direction of the pth pixel in the attitude reference frame can be computed from that of the first and the Nth pixel by interpolation using $U_p = (1-\frac{p-1}{N-1})U_1 + (\frac{p-1}{N-1})U_N$, where U_1 and U_N are the look directions vectors for the first and the N-th pixels.

The orientation of the satellite attitude reference frame can be specified by three rotation angles, $RotX_i$, $RotY_i$, and $RotZ_i$, of the attitude frame with respect to the local orbital frame, for each row i in the image. Nominally, $RotX_i$, $RotY_i$, and $RotZ_i$ are zero at every point in the orbit. These parameters are continuously monitored by the attitude control system and their rate of change (instead of the actual value) is reported as a part of the auxiliary information gathered during image acquisition.

We assume that $\frac{d(RotX_i)}{dt}$, $\frac{d(RotY_i)}{dt}$, and $\frac{d(RotZ_i)}{dt}$ are available for each row, either directly, or through interpolation. Under this assumption, the drift of the attitude frame with respect to the local orbital plane can be computed by integration if that for the first row of imagery is known. This gives rise to three new independent variables $RotX_1$, $RotY_1$, and $RotZ_1$, the rotation angles for the first row.

Besides the above parameters, other independent camera parameters include (1) time from the perigee to the scene center, (2) the dwell time for the detectors (i.e. the time between image lines), and (3) the image coordinates of the center pixel.

Dependent Parameters. These parameters can either be computed from the independent parameters or are measured directly by the on-board systems. The list includes such measurable and given parameters as the ephemeris information (positions and velocities at different points in the orbit), ground control points (i.e., associations between [lat, lon, elevation] and (u, v) in the image), and rates of change of $RotX_i$, $RotY_i$, and $RotZ_i$. Dependent parameters are used to impose constraints on the solution.

3 Tracking the Satellite

A satellite provides a stable and, more importantly, a predictable platform as one can employ constraints dictated by Kepler's laws. This section details the procedures for computing the angular position of a satellite given its travel time from perigee and vice versa. The various elliptical parameters are defined in Fig. 4 [5].

Elapsed Time from True Anomaly. The true anomaly f can be converted into the eccentric anomaly E using, $\cos E = \frac{(\cos f + e)}{(1+e\cos f)}$ where e is the eccentricity of the orbit. A direct relationship exists between the mean anomaly M and the eccentric anomaly E, viz., $M = E - e\sin E$. Thus, from Kepler's second law, one can compute the elapsed time as $t = M/\bar{\omega}$. In this equation, the mean angular velocity $\bar{\omega}$ can be computed using Kepler's third law: $\bar{\omega} = \sqrt{\frac{GM_e}{a^3}}$, where a is the semi-major axis of the orbit, G is the universal gravitational constant, and M_e is the mass of the earth.

True Anomaly from Elapsed Time. One is tempted to back-trace the computation described above to compute the true anomaly from the elapsed time. However, the equation that relates M and E is not explicit in M. To overcome this one must either linearize it to express E in terms of M, or compute E iteratively.

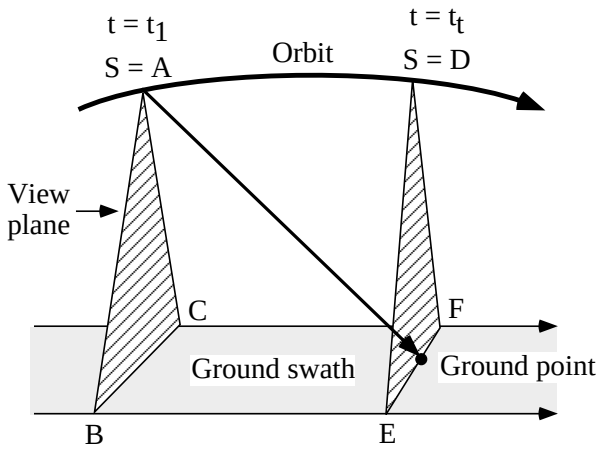

Figure 5: Bringing the ground point in the view plane.

Once again, Mean Angular Velocity can be computed from Kepler's 3rd law, $\bar{\omega} = \sqrt{(GM_e/a^3)}$. The mean anomaly, then, is given by $M = t \times \bar{\omega}$, where t is the elapsed time. One can first compute, from mean anomaly M, a rough value for the eccentric anomaly E using a series expansion of $M = E - e \sin E$. With this coarse value as the starting point, we can solve for fixed point iteratively as follows.

```
old_E = 0.0;
while(fabs(E - old_E) > EPSILON) {
   old_E = E;
   E = M + e * sin(old_E); }
```

In practice, the above iteration rarely takes more than one or two iterative steps. Finally, to compute the true anomaly f from eccentric anomaly E, we use the identity $\tan f = \frac{\sqrt{1-e^2}\sin E}{\cos E - e}$.

4 The Mapping Algorithm

Fig. 5 depicts the mapping of a 3-D ground point to its corresponding image point. The satellite's initial position S in the orbit at time $t = 0$ is marked A. At this time instant, everything in the intersection of the view plane ABC and the ground swath is observed by the satellite. The satellite will observe the point (x, y, z) from the point D in the orbit: at this point, the new view plane (DEF) passes through the ground point. Thus, from a known starting position such as A, the satellite must be moved to point D. We accomplish this by making the angle between the view plane and the ray SX equal to zero, where S denotes the instantaneous position of the satellite in the orbit. The algorithm executes the following steps.

1. Initialization. We always starts at the scene center (i.e., the point A in the above description is the point in the orbit where the central row of the image was acquired). The true anomaly at A is computed using the time from perigee to the scene center and the satellite is moved there. Recall that the time from perigee to the scene center is an independent parameter; Computation of anomaly, given the travel time, has already been detailed in Section 3.

The satellite is moved from point A to D in two main steps called the *coarse* and *fine pointing modes*.

2. Move the View Plane: Coarse Pointing Mode. In this mode, it is assumed that the satellite is a perfectly stable platform and any drifts in its attitude are ignored. In other words, the local orbital frame and the satellite attitude frame are assumed to be perfectly aligned with each other at every point in the orbit.

Assume that the satellite is flying in a straight line as shown in Fig. 6. Let the instantaneous position of the satellite be $t = t_1$ as shown. At this time instant, one can compute the angle between the ray and the view plane in a straight-forward manner. Instead of working with the angle discrepancy between the look direction and the view plane, we work with its complement, viz., the angle between the look direction and the direction of the motion of the satellite. We want to move the satellite to its target position at $t = t_t$ where the angle is Θ_t. In order to accomplish this, we move the satellite a small time step Δt to a new position t_2. At this new time instant, we recompute the position, the velocity vector, and the angle Θ_2. Using sine law, $\frac{\Delta t}{\sin(\Theta_2 - \Theta_1)} = \frac{\chi}{\sin\Theta_1}$ and $\frac{\delta t}{\sin(\Theta_t - \Theta_2)} = \frac{\chi}{\sin(180 - \Theta_t)}$. Eliminating the unknown χ, we get

$$\delta t = \frac{\sin(\Theta_t - \Theta_2)}{\sin\Theta_t} \frac{\sin\Theta_1}{\sin(\Theta_2 - \Theta_1)} \Delta t. \qquad (1)$$

It can be shown that no matter where t_2 is with respect to t_1 and t_t, the above equation holds. Thus we can bring the satellite to bear any desired look angle Θ_t.

In practice when the satellite is actually moved by $\Delta t + \delta t$, the angular discrepancy becomes smaller but is not exactly zero. This is because of the assumption concerning the straight-line motion of the satellite. The above step is repeated till the angular discrepancy is as close to zero as is numerically meaningful.

3. Move the View Plane: Fine Pointing Mode. Here, the attitude drifts measured by on-board system are taken into account. To do this, the drift in yaw, roll and pitch angles of the attitude frame with respect to the local orbital frame are determined from the rate of change in the satellite's orientation measured by on-board systems. Typically, the rates of change of yaw, roll and pitch angles are given only at few select points in the orbit; for others, they are interpolated. From this data, the actual drift in yaw, roll and pitch angles at any point in the orbit is computed via integration. To account for these angles, the satellite is moved as before; The only difference is that at each iterative step the look angles computed are modified by the drift in satellite's attitude determined above.

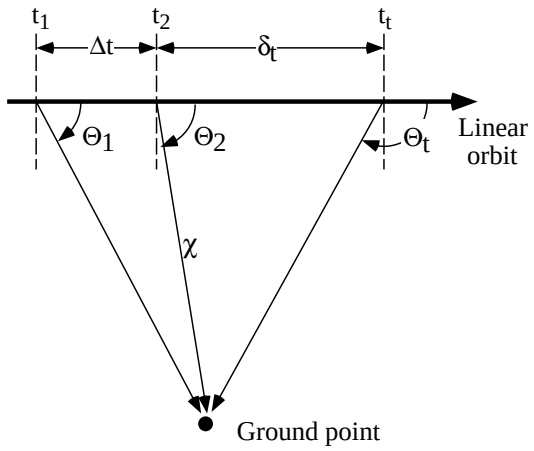

Figure 6: Angular discrepancy and satellite motion.

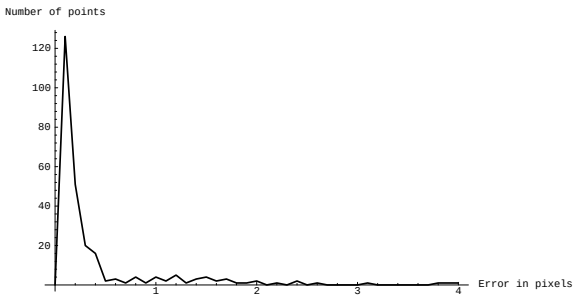

Figure 7: Error in re-projected points.

4. Computation of u and v. Once the satellite has been brought to a place where the 3-D ground point lies in the instantaneous view plane, the travel time contains the information about the u coordinate. If Ppx and Ppy denote image center then $u = Ppx + (travel\_time - center\_travel\_time)/dwell\_time$.

Within the instantaneous view plane, the fraction of field of view angle in the cross-flight direction that is being cut by the look direction needed to see `p3d` contains the information about the v coordinate. Let $viewing\_angle\_ratio$ denote this fraction. The v coordinate is given by $v = 1 + 2(Ppy - 1)viewing\_angle\_ratio$.

Parameter	Actual	Nominal
Semi-major axis a	7182980	7203322 m
Eccentricity e	0.00103919	0.001327417
Inclination i	98.77000	98.73727 deg
Perigee angle w	90.00000	71.39493 deg
Longitude of DN	-131.31380	-131.2472 deg
Look angle Ψ_{x1}	0.54666602	0.8728483 deg
Look angle Ψ_{xn}	0.56695408	0.8895469 deg
Look angle Ψ_{y1}	0.22989154	0.2299154 deg
Look angle Ψ_{yn}	27.1112440	27.11084 deg
Time perigee to ctr	1238.47153	1237.909 sec
Dwell time	0.00150400	0.001503339 sec

Table 1: Estimated vs. nominal parameter values.

5 Experimental Results

A pair of SPOT stereo images of the Los Angeles area were used to calibrate the corresponding cameras. A set of 25 ground control points and 100 image to image match points were used for calibration. The algorithm took about 1 minute on a SPARC 10 to solve for both cameras.

Table 1 shows the estimated independent camera parameters, and their nominal values, for one of the two cameras. As can be seen that a considerable variation exists between the actual and the nominal values. If the ideal values for independent parameters are used for 3-D to 2-D mapping, we get an RMS error of 48.92 pixels. On the ground, this is equivalent to having a discrepancy of about 489 meters.

Fig. 7 shows the error distribution of the re-projected points. As can be seen, about 90% of the points have a reprojection error of less than 1.0 pixel and over 95% are with in 2 pixel error. Points with larger than two pixel errors were manually confirmed to be outliers arising from errors in the matching procedures (i.e., these point pairs were mistakenly identified as match points). The overall RMS error with which a ground point can be mapped to its corresponding image point it 0.73 pixels.

References

[1] SPOT Image Corporation, 1897, Preston White Dr., Reston, VA 22091-4368. *SPOT User's Handbook*, 1990.

[2] Rajiv Gupta. Simple camera models for orbiting pushbroom sensors. In *The proceedings of 1995 Mobile Mapping Symp*, The Ohio State Universiy, May 1995.

[3] Ashley P. Tam. *Terrain elevation extraction from digital SPOT satellite imagery.* PhD thesis, Masters Thesis, Dept. of Surveying Engineering, Calgary, Alberta, July 1990.

[4] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 1988.

[5] C. C. Slama, editor. *Manual of Photogrammetry.* American Society of Photogrammetry, Falls Church, VA, fourth edition, 1980.

[6] Richard I. Hartley and Rajiv Gupta. Linear pushbroom cameras. In *Computer Vision - ECCV '94, Volume I, LNCS-Series Vol. 800, Springer-Verlag*, pages 555–566, May 1994.