(NASA-TM-82169) INTERPCLATING FOR THE LOCATION OF REMOTE SENSOR LATA (NASA) 23 p HC A02/MF A01 CSCL 05B

N82-13469

Unclas
G3/43 04371

NASA

Technical Memorandum 82169

Interpolating for the Location of Remote Sensor Data

E. F. Puccinelli and J. Kornfield

JULY 1981

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

TM 82169

INTERPOLATING FOR THE LOCATION OF REMOTE SENSOR DATA

by

Edward F. Puccinelli
NASA/Goddard Space Flight Center
Code 931
Greenbelt, Maryland 20771

and

Jack Kornfield
National Science Foundation
1800 G. St.
Washington, D.C. 20550

July 1981

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

INTERPOLATING FOR THE LOCATION OF REMOTE SENSOR DATA

Edward F. Puccinelli
NASA/Goddard Space Flight Center

and

Jack Kornfield
National Science Foundation

ABSTRACT

An interpolation algorithm is presented as a practical alternative to common interpolation and approximation methods when applied to the problem of determining the location of remote sensor data. This algorithm is based upon knowledge of the geometry of the problem and is shown to be inherently more accurate than common interpolation schemes which may be applied to all types of data. A practical location problem is used to demonstrate its accuracy and computational cost.

INTERPOLATING FOR THE LOCATION OF REMOTE SENSOR DATA

INTRODUCTION

A common problem encountered in the analysis of remote sensor data is the determination of the location of large volumes of data. The number of computations for the location of any single field of view is in itself not great. When this number is multiplied by the high sampling rate common to many of today's scanners, the cost in terms of time and money may become intolerable. This fact has lead analysts to turn to the use of interpolation and approximation methods to aide in determining the location of fields of view at a reasonable cost. Usually, accuracy of location is sacrificed for speed of computation.

First, this paper will investigate the possible reasons for location inaccuracy which arise when using interpolation or approximation schemes. The results of the discussion will be used to introduce an interpolation algorithm which serves as a practical alternative to the more traditional methods of spline interpolation, polynomial interpolation, least squares polynomial approximation, etc. This new algorithm will then be compared to spline interpolation in an actual application.

BACKGROUNDS, ASSUMPTIONS AND DEFINITIONS

A simple example serves to introduce the interpolation problem. A remote sensor will be assumed to be on board a spacecraft orbiting the earth. The discussion which follows is not dependent on this assumption and the conclusions are equally valid for sensors on board aircraft or for bodies other than the earth.

Other assumptions that will be made for the sake of clarity of the discussion are associated with the sensor and the spacecraft orientation. The spacecraft axes will be considered to have 0° pitch, 0° yaw and 0° roll attitude when the yaw axis is normal to the ellipsoid representing the earth surface and the pitch axis is coincident with the cross product of the normal to the surface and the spacecraft velocity vector. The sensor axes will be assumed coincident with the spacecraft axes. Figure 1 illustrates the geometry under these assumptions.

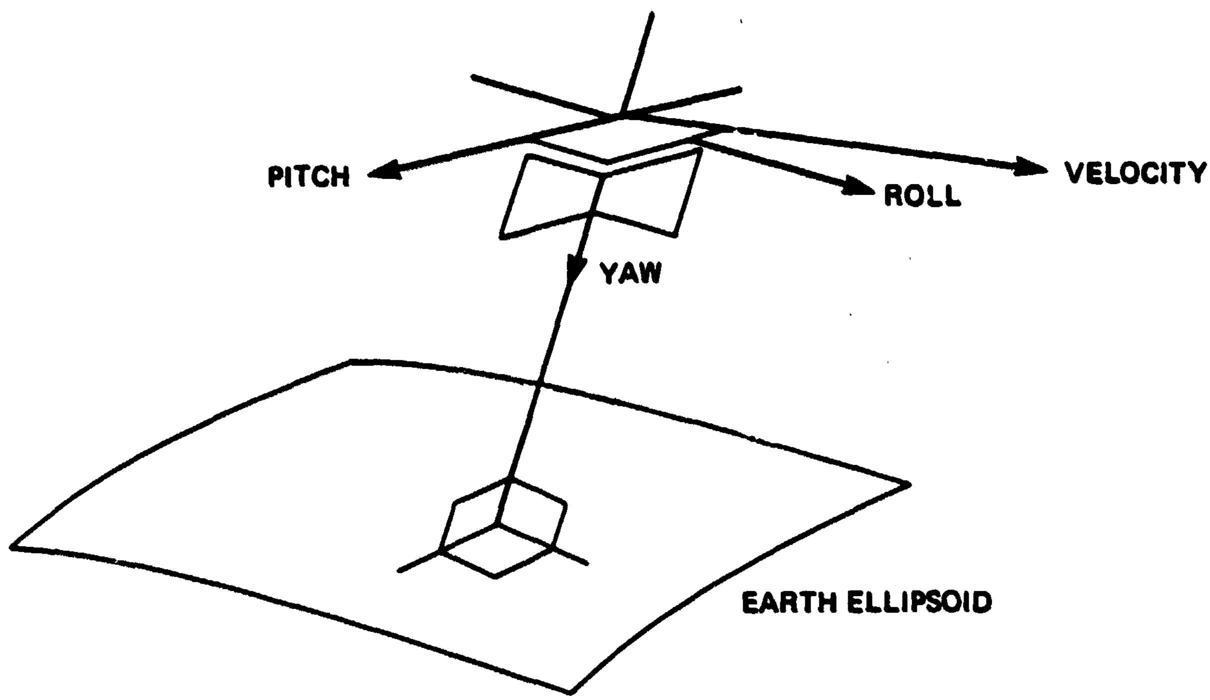

Figure 1. The Sensor and Spacecraft Axes Orientation for 0° Pitch, Yaw and Roll.

It is also assumed that the sensor rotates in the pitch-yaw plane so that in one rotation the sensor returns data for n fields of view. These n fields of view will be defined as being one scan of the sensor. The first field of view of the scan will be defined as occuring at time t_1 and the last at time t_n. No assumption is made about the angular rate of motion of the sensor about the roll axis or about the sampling rate.

Again, these assumptions made for the sake of clarity do not restrict the generalization of the following discussion to other types of sensors. Figure 2 illustrates the second set of assumptions.

To lay a framework for discussing interpolation errors it is necessary to establish the information needed in order to locate a single field of view. The following is a list of the data

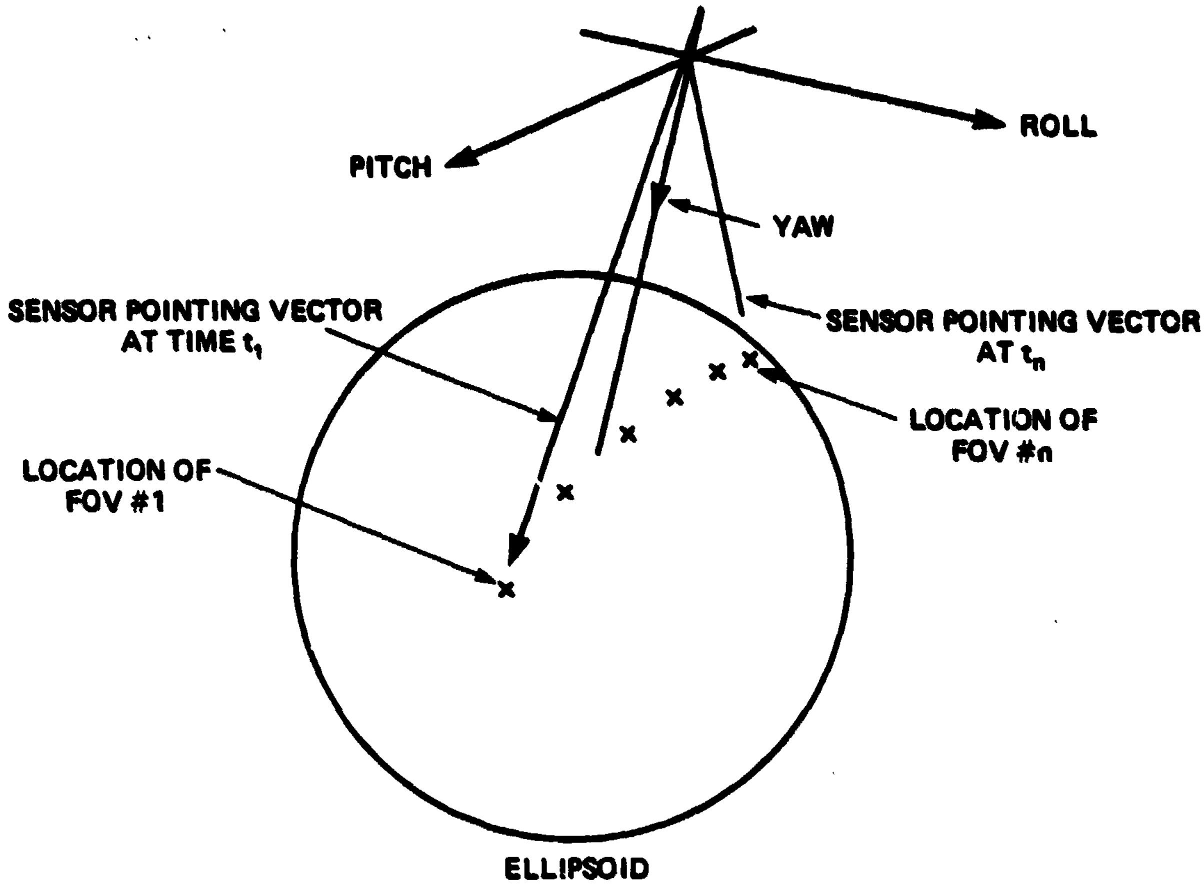

Figure 2. A Single Scan as Viewed by a Plane Rotating Sensor on Board a Stationary Spacecraft.

necessary for this task. (All information, except when otherwise specified is measured relative to an earth fixed cartesian coordinate system whose origin is coincident with the center of the earth.)

1. An equation describing the earth surface. Call this E.
2. The position of the spacecraft as a vector function of time. Call this $\vec{s}(t)$.
3. The velocity of the spacecraft as a vector function of time. Call this $\vec{V}(t)$.
4. The attitude of the spacecraft as a vector function of time. Call this $\vec{a}(t)$ where $\vec{a}(t) =$ (pitch (t), roll (t), yaw (t))^T.
5. A vector function $\vec{w}(t)$ which describes the sensor pointing direction at time t relative to the sensor axes. (Recall that the sensor axes and the spacecraft axes are assumed coincident for the sake of simplicity. Were they not, their relationship would be a sixth piece of information.)

Reference 1 describes in detail a method of determining a vector function $\vec{f}$ of these five parameters which yields the earth fixed location $\vec{e}$ of sensor data.

$$\vec{e}(t) = \vec{f}(E, \vec{s}(t), \vec{v}(t), \vec{a}(t), \vec{w}(t)) \tag{1}$$

If the four functions of time $(\vec{s}(t), \vec{v}(t), \vec{a}(t), \vec{w}(t))$ and equation E were known exactly at any time t, then the vector $\vec{e}(t)$ would be the exact location of the data sensed at t. If from scan start t_1 to scan end t_n the input functions were continuous then $\vec{f}$ would yield a continuous trace of the scan on the earth surface E.

Evaluation of $\vec{f}$ at discrete times $t_1, t_2, \ldots, t_n$ when the sensor is recording data will produce discrete points which are the locations of the individual fields of view in a single scan. This set of locations $\{\vec{e}(t_1), \vec{e}(t_2), \ldots, \vec{e}(t_n)\}$ is defined as a scan trace.

Of course, these five input parameters are hardly ever known accurately which in itself leads to error in the determination of any single location. An analysis of this type of error is beyond the scope of this paper. The discussion here is confined to developing an interpolation method which permits rapid determination of the location of the scan trace.

THE INTERPOLATION PROBLEM

In practice when locating remote sensor data the usual procedure is to determine several locations in a scan trace as accurately as possible. This subset of locations will be referred to as anchor points.* An interpolation or approximation algorithm is then applied to the anchor points in an effort to determine the location of the remainder of the scan trace. Ignoring the errors in location of any individual field of view due to imprecise or inaccurate measurements of the five necessary inputs, the location problem can be expressed as follows:

*The anchor points may be expressed in either cartesian coordinates or latitudes and longitudes. Usually, it is computationally advantageous to work with latitudes and longitudes, however, for the sake of being able to envision the problem in three dimensions the discussion will refer to cartesian coordinates. A practical example is presented later for both systems.

Given a set of anchor points determine a vector function $\vec{g}$ such that

$$\|\vec{f}(t) - \vec{g}(t)\|_2 < \delta \tag{2}$$

for all $t \in \{t_1, t_2, \ldots, t_n\}$.

The vector norm $\|\vec{V}\|_2$ is defined as the Euclidean length of the vector $\vec{V}$.

Note that expressing the problem in this fashion allows for the use of either interpolation or approximation methods for finding $\vec{g}$.

SOURCES OF INTERPOLATION ERRORS

There are two main reasons why $\|f(t) - g(t)\|_2$ may exceed δ between anchor points $\vec{e_i} = \vec{e}(t_i)$ and $\vec{e_j} = \vec{e}(t_j)$.

First, $\vec{g}(t)$ may not coincide with the earth surface. An exaggerated but simple example will demonstrate this point. Suppose $\vec{g}(t)$ is the straight line through two anchor points $\vec{e_i}$ and $\vec{e_j}$ and that the sensor scan plane does not contain the earth center but is actually sensing data near the earth horizon. Figure 3 shows how the difference between $\vec{f}$ and $\vec{g}$ might be larger than a specified tolerance.

It may be argued that if approximation or interpolation is performed on latitude and longitude coordinates then the resulting locations are implicitly on the earth surface. However, as Figure 3 shows, if the scan plane is not near the earth center there may still be a significant error in approximating $\vec{f}$.

The second principle source of error is dependent on the behavior of the four functions $\vec{s}(t), \vec{v}(t), \vec{a}(t), \vec{w}(t)$. Respectively, these describe the position, velocity and attitude of the spacecraft and the motion of the sensor relative to its axes. Consider the following idealized example.

1. Let E describe the earth surface exactly.
2. Let (t_i, t_j) be the time interval in one scan during which locations are sought.
3. Assume that the spacecraft does not move during (t_i, t_j) and is exactly over the equator and the 0° longitude line.

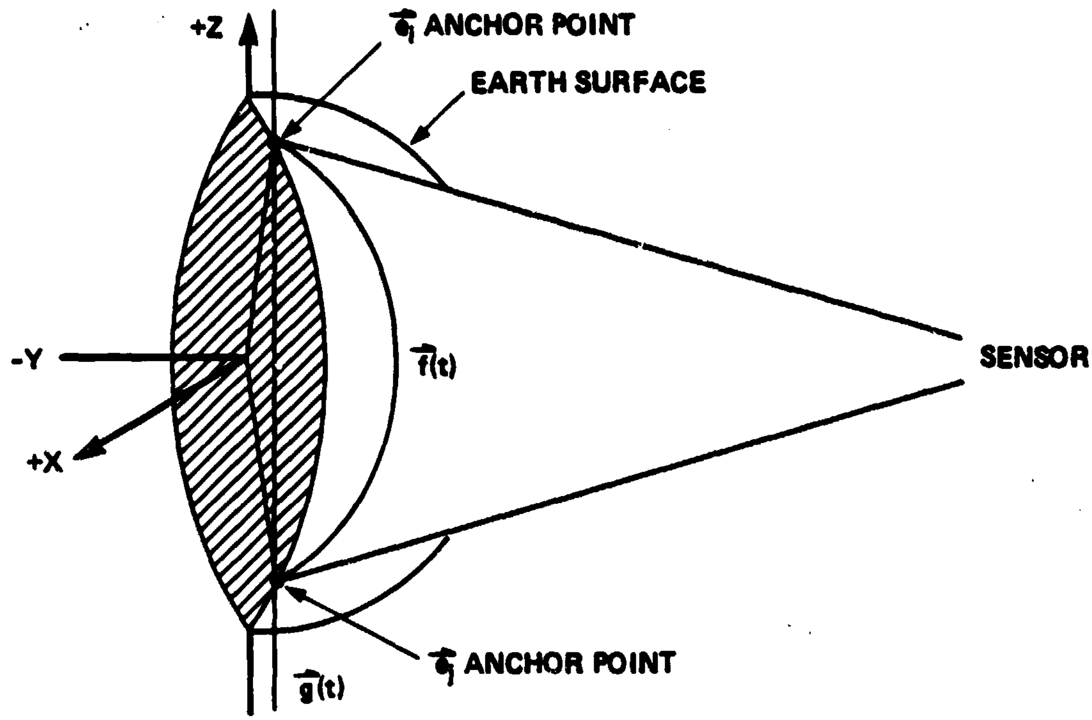

Figure 3. Error in Approximating the Earth Surface.

4. Assume that the velocity vector orients the spacecraft roll axis so that it points due north.
5. Let the scanner pointing vector rotate at a constant rate about the roll axis in the pitch-yaw plane.

Figure 4 shows how the scan trace will coincide with the equator under these conditions if the attitude were perfectly zero in all axes over (t_i, t_j). This figure also shows what the scan trace would be if roll and yaw were constantly zero but pitch varied sinusoidaly as described by

$$\text{pitch}(t) = p \cdot \sin\left[\frac{2(t - t_i)}{t_j - t_i} - 1\right]$$

If only two or three anchor points are used in the interval (t_i, t_j) the scan trace for constant attitude is well approximated but there is little hope for a good approximation in the case of varying pitch attitude.

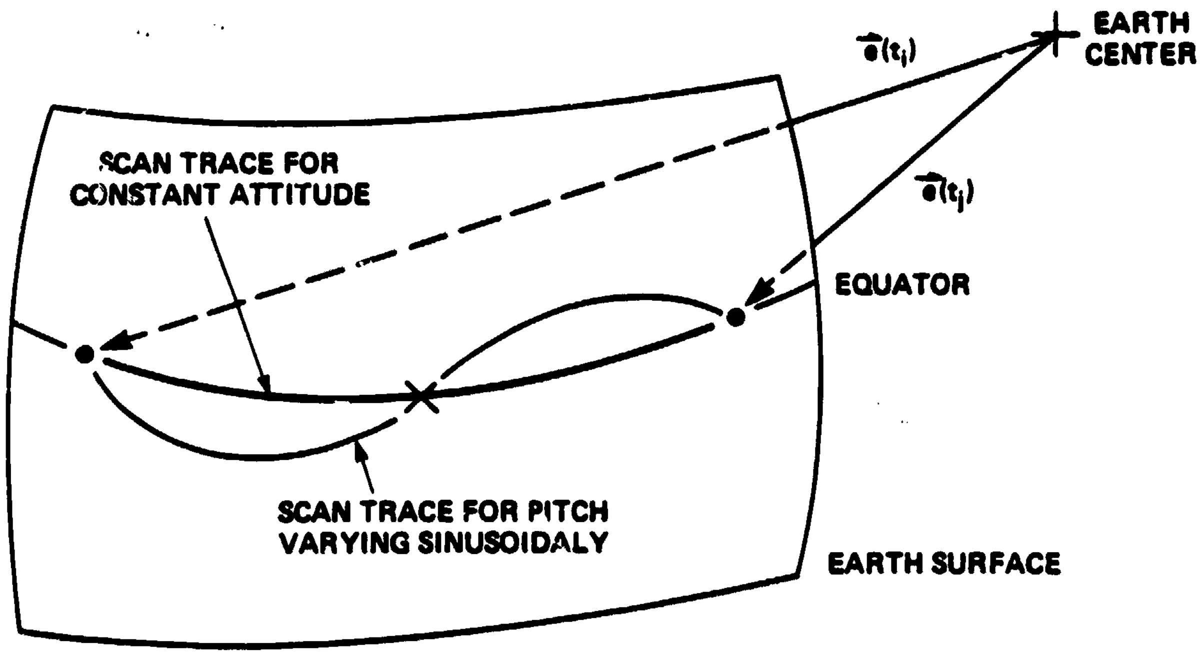

Figure 4. Two Possible Scan Traces.

When the idealized restrictions are removed and the spacecraft position and velocity are allowed to vary with time as well as all three attitude parameters it is not difficult to envision very complicated scan traces on the earth's surface.

The trick is to select a sufficient number of anchor points in the interval (t_i, t_j) to allow for accurate interpolation or approximation. In order to get an idea of what is a sufficient number the analyst must know the expected behavior of the functions $\vec{s}(t)$, $\vec{v}(t)$, $\vec{a}(t)$ and $\vec{w}(t)$.

The usual approach to solving the basic problem described by inequality (2) is for the analyst to solve explicitly for m anchor points in a scan of n fields of view and use them to produce an interpolation or approximation function $\vec{g}$. Common methods include polynomial and spline interpolation as well as least square approximations. Linear interpolation is in itself computationally the least expensive of these methods. However, accuracy requirements may necessitate the computation of so many anchor points that it may be less costly to use another method which requires fewer anchor points to produce the same accuracy.

An important point to note is that all of these methods attempt to approximate the scan trace function $\vec{f}$ without using any information about the component functions of $\vec{f}$.

This fact leads to the concept of developing an interpolation algorithm which approximates the component functions rather than $\vec{f}$ itself. Using these approximate values, $\vec{f}$ is then evaluated exactly.

AN ALTERNATIVE TO COMMON INTERPOLATION AND APPROXIMATION METHODS

This new algorithm is described as follows:

1. Within the time interval of a single scan (t_1, t_n) select a time interval (t_i, t_j) for which

$$\|\vec{f}(E, \vec{s}(t), \vec{v}(t), \vec{a}(t), \vec{w}(t)) - \vec{f}(E, \vec{s}_c, \vec{v}_c, \vec{a}_c, \vec{k}(t))\|_2 < \delta \qquad (3)$$

   where $\vec{s}_c$, $\vec{v}_c$ and $\vec{a}_c$ are constant approximations of the spacecraft position, velocity and attitude and $\vec{k}(t)$ is an approximation in an earth fixed cartesian coordinate system of $\vec{w}(t)$ the sensor pointing direction function. These approximations are described in the following steps. Note that determining (t_i, t_j) may well be impossible to do precisely but is in fact equivalent to selecting anchor points for any of the previously mentioned interpolation and approximation methods.

2. The approximation to the position, velocity and attitude are

$$s_c = s(t_h)$$

$$v_c = v(t_h)$$

$$a_c = a(t_h)$$

   where

$$t_h = (t_i + t_j)/2$$

3. Next, the vector function $\vec{w}(t)$ must be approximated. First determine the anchor points $\vec{e}_i$ and $\vec{e}_j$ by evaluating $f(E, \vec{s}(t_d), \vec{v}(t_d), \vec{a}(t_d), \vec{w}(t_d))$ for $d = i, j$. Then form the two unit vectors

$$\vec{k}_d = (\vec{e}_d - \vec{s}_h)/\|(\vec{e}_d - \vec{s}_h)\|_2,\ d = i, j.$$

These two unit vectors may be pictured as originating from the constant spacecraft position $\vec{s}_h$ and pointing toward the two anchor points. Using these two vectors and knowledge of how the sensor moves with respect to the spacecraft axes the function $\vec{k}(t)$ is required to approximate the motion and satisfy

$$\vec{k}(t_d) = \vec{k}_d \text{ for } d = i, j$$
$$\|\vec{k}(t)\|_2 = 1 \text{ for } t \in (t_i, t_j)$$

Further definition of $\vec{k}(t)$ is dependent upon the motion of the sensor of interest. An example of the complete definition of $\vec{k}(t)$ for a sensor which rotates about the spacecraft roll axis is given in the next section.

4. The final step of this algorithm is to evaluate

$$\vec{e}(t) = \vec{f}(E, \vec{s}_c, \vec{v}_c, \vec{a}_c, \vec{k}(t)) \text{ for } t \in (t_i, t_j).$$

This is done by finding the intersection of the line in space which approximates the sensor pointing direction with the earth surface. The line is given by

$$L(u) = \vec{s}_h + u\vec{k}(t)$$

and the earth surface by

$$E(u_1, u_2) = \begin{pmatrix} a \cos u_1 \cos u_2 \\ a \sin u_1 \cos u_2 \\ c \sin u_2 \end{pmatrix}$$

Equating $L(u)$ to $E(u_1, u_2)$, the parameter u may be solved for by multiplying the third equation by a/c, then squaring both sides of the equation, and finally adding all three equations together. This eliminates the parameters u_1 and u_2 and gives a quadratic equation in u.

$$Au^2 + 2Bu + C = 0$$

where

$$q = a^2/c^2$$

$$A = k_x^2(t) + k_y^2(t) + qk_z^2(t) \quad (4)$$

$$B = s_{xh}k_x(t) + s_{yh}k_y(t) + qs_{zh}k_z(t) \quad (5)$$

$$C = s_{xh}^2 + s_{yh}^2 + q(s_{zh}^2 - C^2) \quad (6)$$

It should be noted that solutions for u represent the distance from the spacecraft to the intersection(s) of L(u) with the earth's surface. Therefore the smaller value is the one of interest giving

$$u = \frac{-B - \sqrt{B^2 - AC}}{A}$$

The situation is shown in Figure 5. The location $\vec{e}$ is then evaluated as

$$\vec{e} = \vec{s}_h + u\vec{k}(t).$$

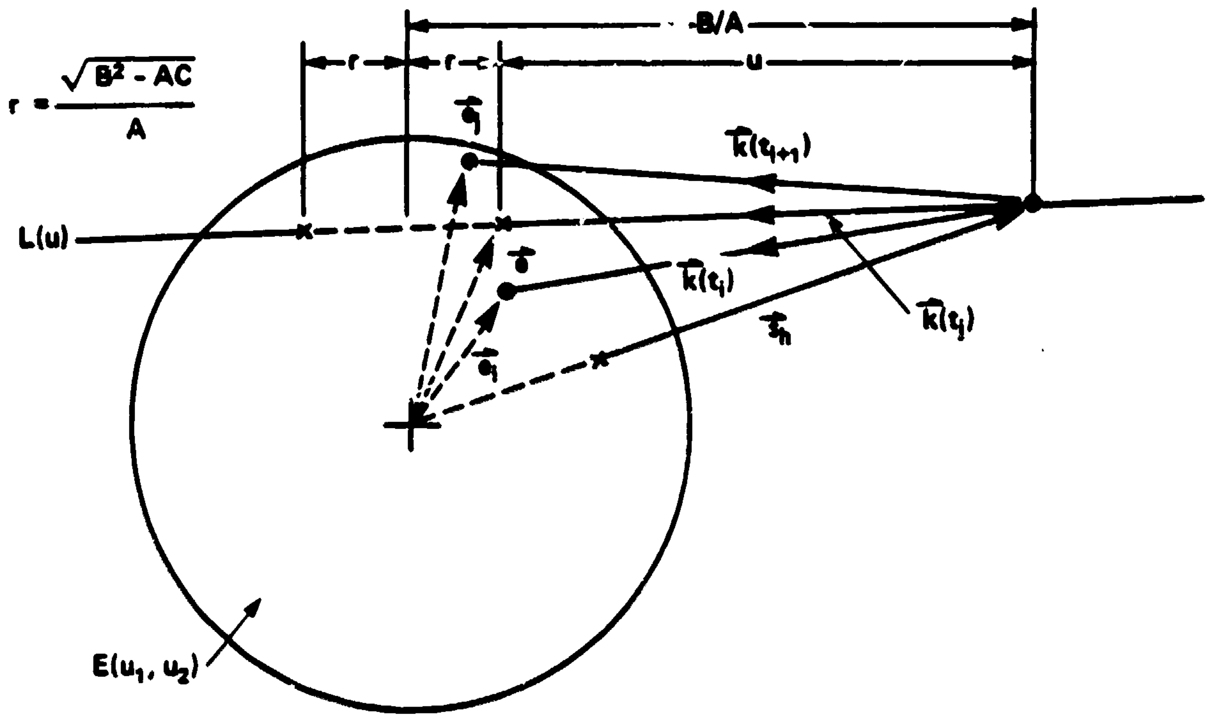

Figure 5. Determination of an Approximate Location $\vec{e}_j$.

In summary, the proposed algorithm involves

a. selection of anchor points in a scan

b. assumption of a single spacecraft position velocity and attitude between any two anchor points.

c. the formation of a function $\vec{k}(t)$ which approximates the sensor motion between the two anchor points.

d. determination of the intercept of $\vec{k}(t)$ with the earth surface.

The basic concept is that the exact function $\vec{f}(B, \vec{s}(t), \vec{v}(t), \vec{a}(t), \dot{\vec{w}}(t))$ is being approximated by making approximations of its component functions.

Incidentally, the algorithm may also be useful over time intervals (t_i, t_j) where the attitude is not approximately constant. For instance, for a scanner rotating in a plane about the spacecraft roll axis, the roll attitude rate should be approximately constant. The pitch and yaw attitudes should be approximately constant. The motion of the spacecraft about the roll axis will then be represented by the scanner pointing vector $\vec{k}(t)$.

A COMPARISON OF SPLINE INTERPOLATION AND THE PROPOSED ALGORITHM

In order to evaluate the proposed algorithm, a count of the operations needed to locate one scan trace will be made. A similar count will be made for the spline interpolation technique. After a comparison of efficiency is made the results of a test to compare accuracy will be described.

Let m_s be the number of anchor points needed in order for cubic spline interpolation on latitude and longitudes to produce locations to an accuracy of δ. Let n_s be the remaining number of fields of view in the scan to be located by the spline interpolation, (e.g. $m_s + n_s$ = total number of fields of view to be located in one scan). Let m_p and n_p be similar numbers for the proposed algorithm. Table 1 shows the number of operations necessary to determine the location of any single anchor point when evaluating the function $\vec{f}$ as described in reference 1.

Table 2 shows the number of operations involved in each step of the procedure when locating a scan trace of cubic spline interpolation of latitude and longitude coordinates.

a. The longitude values must be checked and adjusted as necessary to avoid discontinuity problems in steps 2 and 3. These operations are not included in the count.

b. The number of operations was determined from the code of reference 2, specifically, subroutines ICSICV and ICSEVU.

The number of operations required by the proposed algorithm will be dependent upon how the function $\vec{k}(t)$ is formulated to approximate the motion of the sensor of interest. In order to present a comparison with spline interpolation we will proceed assuming the case of a sensor which rotates in a plane about the spacecraft roll axis. The formulation of $\vec{k}(t)$ for this case is presented as follows:

1. Form the unit vectors k_i and k_j as described in the previous section.
2. Compute the angle α between the two vectors.

$$\alpha = \cos^{-1} (\vec{k}_i \cdot k_j)$$

3. Let

$$\beta = \frac{t - t_i}{t_j - t_i} \alpha$$

4. Then

$$\vec{k}(t) = \sin (\alpha - \beta) \vec{k}_i + \sin \beta \vec{k}_j$$

is a function which approximates $\vec{w}(t)$. Note that it is not of unit length as formally required, but since its maximum length is less than 2 there is no ambiguity in solving for the intersection of L(u) and E. The function $\vec{k}(t)$ is not normalized in order to save computations.

The number of operations for the proposed algorithm based upon the given formulation of $\vec{k}(t)$ is given in Table 3.

In Table 3, operations which would only need to have been computed once, no matter how many scans were being approximated, were not counted. Other methods of formulating $\vec{k}(t)$ might be found for a specific problem which involve fewer operations.

Tables 2 and 3 were set up to produce locations in latitude and longitude coordinates. This is advantageous to spline interpolation when comparing the totals. Had it been required to produce locations in cartesian coordinates then in Table 2 the counts in step 1 would be slightly reduced while the counts in steps 2 and 3 would be multiplied by a factor of 3/2 to reflect interpolation on three coordinates as opposed to two. Table 3 would be altered by deleting step 4. The comparison is shown in Table 4. Note that constant terms have been purposely dropped.

Table 4 shows that it is not obvious which method is more efficient without further information about actual values of m_s, n_s, m_p, n_p, and the time each type of operation takes on a specific computer. In practice, one must determine the values m_s and m_p for a specific problem and then refer to the table to establish which method is faster.

In order to compare the accuracy of the two methods, an exact computer model was developed for the scan trace of the Temperature Humidity Infrared Radiometer (THIR) on board the Nimbus series spacecraft. The parameters of the Nimbus-6 sun synchronous orbit used by the model were

a. 99.15° inclination

b. 7333.16km semi-major axis

c. .001 eccentricity

d. 104.16 min period

The modeled characteristics of the sensor are identical to those previously assumed. In this case, the sensor rotates at 48 rpm in a plane about the spacecraft roll axis at a linear rate. Each field of view is sampled for .22ms with a delay of .98ms between samples (reference 4) so that an entire scan from earth edge is approximately comprised of 343 samples. The spacecraft axes were oriented as assumed previously so that for 0° yaw, pitch and roll, the yaw axis is normal to the earth ellipsoid and the pitch axis is normal to the velocity and yaw vectors. The earth ellipsoid was assumed to have an equatorial radius of 6378.144km and a polar radius of 6356.759km. For the time period of one scan the attitude was held at zero in all axes. The model then evaluated $\vec{f}$ at 343 times. The solutions $\vec{e}_1, \vec{e}_2, \ldots \vec{e}_{343}$ were considered to be exact.

The location accuracy requirements were chosen to be $\delta = 3.6$km. The location volume requirements were for 93 fields to be located in each scan such that they were approximately evenly spaced in distance on the earth surface.

A computer program was written which used $\vec{e}_1, \vec{e}_{172}, \vec{e}_{343}$ as anchor points. The cubic spline through those points was evaluated at the remaining 340 locations and compared to the exact solutions. The number of anchor points was then incremented by five and selected so that they were evenly spaced along the scan on the earth surface. Interpolated locations were again compared to exact locations. This procedure was repeated incrementing the number of anchor points by five each time until the maximum difference between the exact solution and the interpolated solution at any field of view was less than 3.5 km. The result was that 23 anchor points produced an error of 6 km while 28 anchor points reduced the error to 3 km. The error measurement was made first for spline interpolation of latitudes and longitudes and then for cartesian coordinates. No significant difference in error measurement was formed.

Next the proposed method was applied. It was found that only two anchor points e_1 and e_{343} gave a maximum error of .5 km!

Applying $m_s = 28$, $m_p = 2$, $n_s = 65$ and $n_p = 91$ to Table 4 produces the comparison shown in Table 5.

The proposed method is more efficient when interpolating on latitude–longitude and cartesian coordinates. In both cases the maximum error was less by a factor of 6 when the proposed method was used.

It should be noted that if all 343 fields of view had to be located rather than 93 evenly spaced in distance, then Table 4 would have shown spline interpolation to be faster.

CONCLUSIONS

The proposed algorithm for location of remote sensor data is a practical alternative to common interpolation and extrapolation techniques. If applied to the same anchor points used for one of the traditional methods it can produce more accurate locations. An important feature is that it requires the determination of fewer anchor points. Its efficiency is dependent upon the number of anchor points required to produce a desired accuracy and the number of locations sought and may be easily evaluated using Table 4.

In practice, for situations where extremely large volumes of locations are desired a hybrid approach may prove advantageous. If one considers linear interpolation to be the fastest interpolation possible then it would be logical to determine the anchor points necessary in a scan to produce locations to a specified accuracy using this method. However, instead of determining these anchor points directly by evaluating $\vec{f}(E, \vec{s}(t), \vec{v}(t), \vec{a}(t), \vec{w}(t))$ use the proposed scheme or perhaps one of the traditional methods to determine the anchor points needed for linear interpolation.

Table 1
Operations in Locating One Anchor Point

Solution Coordinates	Mults.	Adds	$\sqrt{\ }$	Trig.	Trig.$^{-1}$
Lat-Lon	95	43	5	14	3
Cartesian	90	42	4	14	1

Table 2
Number of Operations When Using Spline Interpolation

Step	Mults.	Adds	$\sqrt{\ }$	Trig.	Trig.$^{-1}$
1. Locate m_s anchor points in lat-lon coordinates[a].	95 m_s	43 m_s	5 m_s	14 m_s	3 m_s
2. Determine the coefficients of the cubic splines for latitude and longitude between each pair of anchor points[b].	32 m_s -38	38 m_s -48	0	0	0
3. Evaluate the spline functions at n_s times for latitude and longitude values[b].	6 n_s	8 n_s	0	0	0
Totals	6 n_s + 127 m_s - 38	8 n_s + 81 m_s - 48	5 m_s	14 m_s	3 m_s

a. The longitude values must be checked and adjusted as necessary to avoid discontinuity problems in steps 2 and 3. These operations are not included in the count.
b. The number of operations was determined from the code of reference 2. Specifically subroutines ICSICV and ICSEVU.

Table 3
Number of Operations When Using the Proposed Algorithm

Step	Mults.	Adds	$\sqrt{\ }$	Trig.	Trig.$^{-1}$
1. Locate m_p anchor points in cartesian coordinates	90 m_p	42 m_p	14 m_p	14 m_p	m_p
2. Compute the $m_p - 1$ angles α	$3(m_p - 1)$	$2(m_p - 1)$			$m_p - 1$
3. Form the m_p vectors $\vec{k}$	4 m_p	5 m_p	m_p		
4. Form n_p values of β	2 n_p	n_p			
5. Evaluate $\vec{k}(t) = \sin(\alpha - \beta)\vec{k_i} + \sin(\beta)\vec{k_j}$ n_p times	6 n_p	4 n_p		2 n_p	
6. Find the intersection of L(u) and E a total of n_p times. (See equations 4, 5, 6)	8 n_p + $4(m_p - 1)$	4 n_p + $3(m_p - 1)$			
7. Evaluate the location $\vec{e} = \vec{s_h} + u\vec{k}(t)$ a total of n_p times	3 n_p	3 n_p			
8. Convert the n_p locations to latitude and longitude values	5 n_p + 5 m_p	$n_p + m_p$	$n_p + m_p$		2 n_p + 2 m_p
Totals	24 n_p + 106 m_p - 7	13 n_p + 53 m_p - 5	n_p + 6 m_p	2 n_p + 14 m_p	2 n_p + 4 m_p - 1

Table 4
Comparison of the Total Operations

Coordinate System Desired for Locations	Method	Mults.	Adds	$\sqrt{\ }$	Trig.	$Trig.^{-1}$
Lat-Lon	SPLINE	$6 n_s + 127 m_s$	$8 n_s + 81 m_s$	$5 m_s$	$14 m_s$	$3 m_s$
	PROPOSED	$24 n_p + 106 m_p$	$13 n_p + 53 m_p$	$n_p + 6 m_p$	$2 n_p + 14 m_p$	$2 n_p + 4 m_p$
Cartesian	SPLINE	$9 n_s + 138 m_s$	$12 n_s + 99 m_s$	$4 m_s$	$14 m_s$	m_s
	PROPOSED	$19 n_p + 101 m_p$	$12 n_p + 52 m_p$	$3 m_p$	$2 n_p + 14 m_p$	$2 m_p$

Table 5
Comparison of Methods for an Actual Situation

Coordinate System Desired for Locations	Method	Mults.	Adds	$\sqrt{\ }$	Trig.	$Trig.^{-1}$
Lat-Lon	SPLINE	3946	2788	140	392	84
	PROPOSED	2396	1289	103	210	190
Cartesian	SPLINE	4449	3552	112	392	28
	PROPOSED	1931	1196	6	210	4

REFERENCES

1. Puccinelli, E. F., "Ground Location of Satellite Scanner Data", Photogrammetric Engineering and Remote Sensing, Vol. 42, No. 4, April 1976, pp. 537–543.

2. IMSL Library 1 Reference Manual, Vol. 1, Edition 5, 1975.

3. Nimbus 6 User's Guide, NASA-Goddard Space Flight Center, February 1975.

4. THIR Timing Study, Operations Research Incorporated, Memorandum from S. Toth, R. Prince to C. Rice, R. Shapiro, June 1977.