Estimating the Position of a Sphere from Range Images

Daniel Clouse
Curtis Padgett
Jet Propulsion Laboratory, California Institute of Technology,
4800 Oak Grove Dr., Pasadena, CA 91109-8099
daniel.clouse@jpl.nasa.gov

Abstract— This paper describes two algorithms for accurately locating a spherical object in space using data collected from a scanning laser range finder. This work is designed to support the capture phase of a future rendezvous and sample return mission to Mars. The first algorithm finds the parameters of the sphere which optimally fit the set of data points. This method is shown to be quite slow, and unlikely to meet the accuracy requirements of the mission. The second algorithm makes a set of almost-independent estimates of the position of the center of the sphere, one from each scan line of data. These estimates are combined using regression to produce an aggregate estimate of the sphere position. This second algorithm is both fast enough, and accurate enough to meet mission requirements.

TABLE OF CONTENTS

1. INTRODUCTION

A future rendezvous and sample return (RSR) mission will attempt to return rock and soil samples from Mars to Earth. The current mission scenario requires retrieval of a 16 cm diameter spherical orbital sample canister (OS) from Mars orbit. In the terminal approach phase of this docking maneuver the principal sensor used to determine the direction and distance to the OS will be a scanning laser range finder.

The terminal approach phase covers a range of distances from 5 km to 1 meter. During this interval, the laser range finder is required to return independent position estimates at one-half second intervals. These estimates will be filtered by guidance and control software to provide a continuous, highly accurate estimate of the position of the OS.

The laser range finder requires algorithms to accurately determine the 3 degree of freedom (DOF) position of the OS throughout the terminal approach phase. This document describes two algorithms useful in the final 20 meters, and analyzes their performance in simulation. The requirements are for accuracy of 8 mm standard deviation in each of the X, Y, and Z directions. The algorithm must also run on a 12.5 MHz processor, producing a position estimate every one-half second.

The first algorithm determines the OS position by finding the moving sphere which best fits the data. We call this algorithm Sphere Fitting. This algorithm is able to meet the accuracy requirements much of the time, but produces high error in some conditions. Though it may be possible to modify the algorithm to produce more accurate results, the algorithm also suffers from being quite slow. Were this algorithm chosen, further work would be required to speed up the algorithm to allow it to run in real time on a 12.5 MHz processor.

The second algorithm determines the OS position by generating a 3 DOF position estimate from each scan line. We call this algorithm Scan Line Extent. This algorithm is more accurate than Sphere Fitting, and also much faster. There should be no problem running this algorithm in real time.

In the next section, we describe the two algorithms. Section 3 lists the assumptions upon which our evaluation of the algorithms are based. Section 4 describes the simulation methodology used to evaluate the performance of the two algorithms. Section 5 reports the expected performance characteristics of the two algorithms based on the simulation results. Section 6 lists our conclusions.

2. DESCRIPTION OF ALGORITHMS

Two algorithms for determining the position of the OS at close range have been developed and evaluated. These are the Sphere Fitting algorithm and the Scan Line Extent algorithm. In this section, we describe each of these.

Sphere Fitting

The sphere fitting algorithm is based on a method commonly used for fitting circles in 2D images. We extended

0-7803-7231-X/01/$10.00/
IEEEAC paper #013, Updated Dec. 5, 2001

the algorithm to work in 3 spatial dimensions, and also to handle a moving sphere.

Generally, the pixels in an image are assumed to be acquired during the same instant in time. This is not true for the "pixels" in the range image from a laser range finder. In our application, the "pixels," or more accurately measurements, are acquired over the course of $\frac{1}{2}$ second. During this time, the target is expected to move. Each measurement contains the spherical coordinates (angle, angle, range) to a point in space, and the time at which the measurement was acquired.

First we convert the spherical coordinates to Cartesian coordinates. The foreground measurements on the OS are easily distinguishable from the distant background measurements by thresholding. In this algorithm, we ignore all the background points. This gives us a set of k points each of the form (x_i, y_i, z_i, t_i). The t_i in each point is the time stamp.

One common circle fitting algorithm (see [1]), minimizes the following cost function C to find optimal values for the center (X, Y) and radius R of the circle.

$$C(X,Y,R) = \sum_{i=1}^{k} \left[\sqrt{(x_i - X)^2 + (y_i - Y)^2} - R \right]^2 \quad (1)$$

Here, $\sqrt{(x_i - X)^2 + (y_i - Y)^2}$ is the distance from the measurement (x_i, y_i) to the putative center of the circle (X, Y). When the cost function is minimized (i.e. when X, Y, and R are "correct"), this distance should be very close to the radius of the circle R.

To reduce the number of square roots, and thus speed the computation of the cost function, the following function may be minimized instead.

$$C(X,Y,R) = \sum_{i=1}^{k} \left[(x_i - X)^2 + (y_i - Y)^2 - R^2 \right]^2 \quad (2)$$

In our case, the value of R is a known constant, $8cm$. This simplifies the problem a bit. However, this simplicity is offset by the complexity added by the third dimension, and the fact that the sphere is moving. The concept of a constant center (X, Y, Z) of the sphere is no longer valid. Now the constant center is replaced by a function of time, as in equations 3 through 5.

$$X(t) = X_0 + \dot{X}t \quad (3)$$

$$Y(t) = Y_0 + \dot{Y}t \quad (4)$$

$$Z(t) = Z_0 + \dot{Z}t \quad (5)$$

Here, (X_0, Y_0, Z_0) represents the position of the center of the OS at time 0; $(\dot{X}, \dot{Y}, \dot{Z})$ represents the velocity of the OS. The velocity is assumed constant throughout the $\frac{1}{2}$ second measurement interval. This assumption is discussed in section 3. The position of the center of the sphere at time t is then given by $(X(t), Y(t), Z(t))$.

Using the notation of equations 3 through 5, the cost function we have chosen to minimize in the Sphere Fitting algorithm is given in equation 6.

$$C(X_0, Y_0, Z_0, \dot{X}, \dot{Y}, \dot{Z}) = \sum_{i=1}^{k} [(x_i - X(t_i))^2 + (y_i - Y(t_i))^2 + (z_i - Z(t_i))^2 - R^2]^2 \quad (6)$$

Minimization is accomplished by performing conjugate gradient descent, as in [2]. This is an iterative procedure which uses both cost function evaluations, and evaluations of the derivative to find a set of parameters $(X_0, Y_0, Z_0, \dot{X}, \dot{Y}, \dot{Z})$ which produce a locally minimum cost.

Scan Line Extent

The Sphere Fitting algorithm ignores the chronological order in which measurements are collected. Not so with the Scan Line Extent algorithm. This algorithm recognizes that the raster scan pattern used by the laser ranger device may be broken up into scan lines. Without loss of generality, we assume the horizontal scan is faster than the vertical scan, so that azimuth position changes very quickly compared to elevation.

Each scan line which encounters the OS produces an approximately linear set of measurements across the face of the sphere. Though the OS may move a significant amount in $\frac{1}{2}$ second, the movement of the OS in the $\frac{1}{100}^{\text{th}}$ second required to collect a scan line is expected to be no more than a millimeter. Since this is smaller than the accuracy of the measurements, we treat all points in the scan line as if they were collected at the same instant of time.

At near distances, the two angular components of a collected point provide much more accurate measurements than does the range component. This algorithm attempts to leverage the angular components to improve accuracy. Consecutive measurements are very close together in azimuth. The first and last measurements of a scan line which encounter the OS give very accurate estimates of the azimuth of the edges of the OS.

This edge azimuth information is used in two ways. First, the average of the two azimuths gives an estimate of the azimuth of the center of the OS at the time at which the scan line was collected. Second, the difference between the two edge azimuths tells the angular extent of the OS.

The angular extent is used in two ways. First, the angular extent is used to combine the range components of all scan

line measurements into an accurate estimate of the distance to the center of the OS. Second, once the distance to the center of the OS is known, the angular extent may be used to produce an estimate of the elevation difference between the scan line and the OS center.

In this way, each scan line which encounters the OS produces an estimate of the coordinates of the center of the spherical OS at the time at which the scan line was collected. Regression is used to produce a combined estimate of (X_t, Y_t, Z_t), the position of the OS at time t, and $(\dot{X}, \dot{Y}, \dot{Z})$, the velocity of the OS during the measurement interval. A more detailed description of this algorithm may be found in appendix A.

3. Assumptions

The development of the two algorithms, and the methodology used to test them are based on a number of assumptions. These are discussed in the following sections.

Motion is Linear

Both algorithms assume that the relative velocity of the OS is constant with respect to the sensor during any $\frac{1}{2}$ second time interval in which a set of measurements is taken. There are a number of ways in which this assumption is actually false.

First, the assumption is false because, according to our requirements, the sensor may be rotating at a rate of 6 mrad/sec around each of the three axes. Our analysis, in section 3, suggests that this induces a negligible departure from linearity during any measurement time interval, and therefore will have insignificant influence on algorithm performance.

Second, the assumption is false due to the influence of gravity gradient accelerations. However, over the course of the one-half second interval, this influence is expected to be insignificant as well.

Third, the assumption is false during any measurement interval in which an acceleration is applied to the spacecraft. Presumably, the spacecraft control software knows when accelerations are applied, and can ignore the suspect measurements taken during these maneuvers.

Magnitude of Linear Motion

In this section, we present analysis to support our contention that spacecraft rotations produce only negligible departures from linear motions. We also calculate the magnitude of the apparent linear motion of the OS at various distances in the rendezvous scenario.

According to our requirements, the maximum expected relative velocity between the spacecraft and the OS is 1.5 cm/sec in the x and y directions, and 3 cm/sec in z. Additionally, a maximum of 6 mrad/sec rotation is expected around each spacecraft axis. The laser ranger is pointing up the z axis, and we assume it is mounted 1 meter away from the spacecraft center of mass along the x-axis.

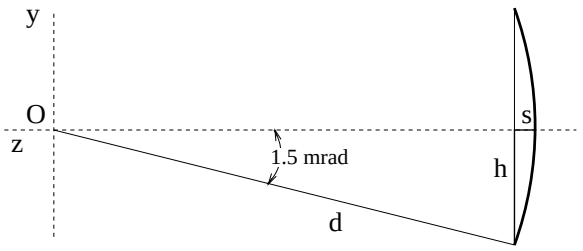

Figure 1. Aligning reference frame with scan line

First, let us look at how close to linear the apparent motion of the OS will be under these constraints. All the motions are linear except for the spacecraft rotations. We demonstrate that these rotations induce very nearly linear relative motions using figure 1. In this figure the laser ranger is positioned at the origin, and the curve represents the apparent motion of the OS relative to the sensor during a $\frac{1}{2}$ second interval due to rotation of the sensor about the x-axis. During this period of time the angular change is 3 mrad. In order to illustrate the configuration, the angle shown in the figure is much larger than this. The distance to the OS is given by d. We want to know the relationship between h, half the arc chord, and s the largest offset from the arc to the arc chord. If h is large compared to s, then the apparent motion is very close to linear. The following equations calculate the proportion $\frac{2h}{s}$.

$$h = d \sin 1.5\text{mrad} \tag{7}$$

$$s = d - d \cos 1.5\text{mrad} \tag{8}$$

$$\frac{2h}{s} = \frac{2 \sin 0.0015}{1 - \cos 0.0015} = 2667 \tag{9}$$

The divergence from linearity is only 1 part in 2667 indicating that the motion is very nearly linear. This result is independent of the distance to the OS, and is the same regardless of the axis of rotation. At a distance of 20 meters, $s = 2.3 \times 10^{-5}$ meters represents the error in range introduced by the assumption of linearity. This is far below the required level of noise. Analysis of rotations about the other two axes also yield extremely small errors.

Next, we calculate the magnitude of the linear motions in each of three dimensions under worst-case conditions. These calculations will be used in the simulations reported in sections 4 and 5.

First, we look at the worst case motion per second in the z-dimension. There are two contributors to a change in z: the 3.0 cm/sec translation in the z direction, and the 6 mrad/sec

Dist. (meters)	Rot. about Y, X (cm/sec)	Rot. about Z (cm/sec)	X, Y Translation (cm/sec)	Total X (cm/sec)	Total Y (cm/sec)
1	0.6	0.6	1.5	2.7	2.7
2	1.2	0.6	1.5	4.1	4.1
5	3.0	0.6	1.5	5.1	5.1
10	6.0	0.6	1.5	8.1	8.1
15	9.0	0.6	1.5	11.1	11.1
20	12.0	0.6	1.5	14.1	14.1

Table 1. Maximum linear motion in x and y

rotation about the y-axis. Rotations about the x- and z-axes produce no change in the distance to the OS. However, since the laser is mounted 1 meter away from the center of mass along the x-axis, a rotation about the y-axis will tend to move the sensor nearer to or farther from the target — a change in z. The magnitude of this change due to a 6 mrad/sec rotation about the y-axis is 0.6 cm/sec. Note that this motion is independent of the distance to the target, as is the 3.0 cm/sec translation in the z direction. The maximum linear motion in the z-dimension is the sum of these two, 3.6 cm/sec.

For the x- and y-dimensions, the magnitude of the linear motion changes with distance to the OS for some of the rotations. Therefore, we present table 1, which tabulates the maximum linear motion in the x and y directions at various target distances.

The first column gives the distance to the OS. Columns two, three, and four list the contributions from various sources of motion – rotations and translations. These are summed to produce the maximum linear motion in the x- and y-dimensions, listed in the last two columns..

Note that the calculations for x and y are symmetrical. A rotation about x contributes the same change in the y-dimension as a rotation about y contributes to x. The magnitude of these changes are listed in column two. A rotation about z contributes the same to each of x and y. This contribution is constant with respect to the distance to the target, and is based on the offset of the sensor from the center of mass (1 meter). This contribution due to a rotation about z is given in column three. Finally, the 1.5 cm/sec translation along the axis of interest (x or y) is listed in column four.

These maximum linear motion calculations are used to establish the amount of motion to be applied in the simulations reported in sections 4 and 5.

Entire OS is Within Field of Regard

Both algorithms assume that the entire OS is within the field of regard during the measurement interval. The consequences of this assumption being false are different for the two algorithms.

The impact of failure of this assumption on the Scan Line Extent algorithm are potentially quite severe. Any scan line in which either edge of the OS cannot be found is rejected by the algorithm, and is therefore useless. If the OS intersects the left or right edges of the field of regard many scan lines will be rejected. If the left or right edge bisects the sphere, all scan lines will be rejected. The impact on the accuracy of the algorithm will be catastrophic.

An intersection of the OS with the top or bottom of the field of regard has a far less deleterious effect. In this case, many scan lines will be preserved, and the impact will be smaller. The results reported in section 5 at a distance of 1 meter demonstrate this case. At 1 meter, the OS fills the entire 10 by 10 degree field of regard. The laser ranger requires a full second to scan an area this large, so only half of the OS can be scanned within the $\frac{1}{2}$ second measurement interval. Even though only half of the OS is being scanned, enough good scan lines are produced to generate an acceptably accurate estimation of the OS position.

We expect the impact of violation of this assumption on the Sphere Fitting algorithm to be less severe than on the Scan Line Extent algorithm. The Sphere Fitting algorithm treats each measurement identically. Although the algorithm attributes no special status to points near the edge of the sphere, these points are more informative for determination of the X and Y components of the center position estimate. We therefore expect some degradation of performance if edge points are lost.

The evaluation of the algorithms in section 5 are based on the assumption that the OS in its entirety is visible within the field of regard. Very little study has been done on how the two algorithms degrade when this assumption is violated. However, significant degradation in accuracy should be expected if this assumption is violated.

Laser Radar Characteristics

We assume the laser radar used to acquire range images is capable of generating pulses at 10000 MHz. We assume the scan rate in the x-dimension is 1000 degrees per second, and 10 degrees per second in the y-dimension. We assume the range measurements returned by the laser ranger are accurate to $\sigma = 2.0$ cm. We assume the x, and y encoders

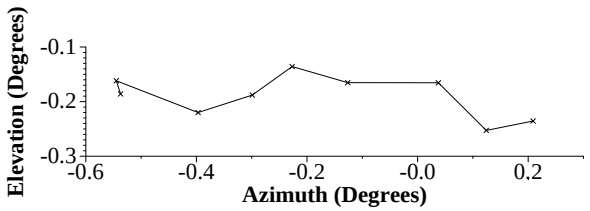

Figure 2. Scan line at 10 meters with noise added.

Dist. (meters)	$\frac{1}{2}$ Angle Extent (degrees)	Frames Per Trial
1	5.22	0.48
2	2.65	0.95
5	1.12	2.23
10	0.60	4.14
15	0.43	5.77
20	0.35	7.20

Table 2. Frame Size and Number of Frames at Different Distances

are capable of reporting the scanner angle to an accuracy of $\sigma = 0.78$ mrad.

4. Simulation Methodology

In order to produce the error estimates presented in section 5, we built a simple simulator to produce range data. The simulator generates a raster scan pattern and collects ranges to a simulated moving sphere. We do not attempt to faithfully reproduce the interaction between the Gaussian beam and the surface of the sphere. Rather, a ray is cast in the current scanner direction, and the range is reported as the distance from the ray origin to its intersection to the sphere. This range data is used as input to the two algorithms.

The scan pattern generated is a perfect raster scan pattern using the scan rates and pulse rate as reported in section 3. The scan rates are of constant velocity with instantaneous accelerations at the ends of scan lines and at the top and bottom of frames. The pulse rate is also constant with no added noise.

After the ray intersection is calculated, we add random Gaussian noise to each component of the simulated position measurement. The magnitude of noise added to each of azimuth, elevation, and range is as specified in section 3. Figure 2 shows the azimuth and elevation of the points in a scan line across the center of the sphere at a distance of 10 meters.

The simulator assumes that the velocity of the sphere relative to the sensor is constant throughout the scanning period. The velocity simulated is that given given in section 3. Note that the relative sphere velocity used in a particular simulation is based on the mean distance to the sphere. See table 1.

In order to estimate the noise on the position estimates of each algorithm, we run 200 trials at each of 6 different distances. During each trial, the algorithm receives $\frac{1}{2}$ second of data, taken as the sphere moves up, to the right (from the point of view of the sensor) and toward the sensor. The starting position of each trial is chosen so that the center of the sphere will move through the point (0, 0, Z) at time 0.25 seconds, where Z is the distance for the current trial. Each $\frac{1}{2}$ second scan starts at the upper left hand corner and starts scanning right and down.

The angular extent of each scan is chosen to be 105% of that required to view the complete sphere at all times during the measurement interval. The size of the frame increases as the distance to the sphere is reduced. The number of frames which can be completed in $\frac{1}{2}$ second then necessarily must decrease. Table 2 gives the calculated half angular extent, and the number of frames at each of 6 different distances. Note that at 1 and 2 meters, less than a full frame is collected. Also, note that at 1 meter, the half angular extent exceeds 5 degrees, the baseline for the current scanner design.

5. Simulation Performance

The simulation technique described in section 4 allows us to evaluate the performance of the algorithms. For each $\frac{1}{2}$ second of range data, each algorithm produces an estimate of the position of the sphere half-way through the time interval, and its velocity throughout the interval. Each such estimate may be compared to the true simulation position and velocity in order to ascertain the accuracy of the two algorithms.

For each algorithm, and for each of six different distances, we gathered 200 such position and velocity estimates, each based on a $\frac{1}{2}$ second of data with independent added noise. This allows us to estimate, for each of the x-, y-, and z-dimensions, the bias of the algorithm, and the noise around the bias.

The bias μ is merely the mean of the differences between the estimates and the truth as given in equation 10.

$$\mu = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i) \tag{10}$$

The noise σ is the square root of the variance of the estimates about the truth as denoted in equation 11.

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{11}$$

In these two equations, $\hat{x}_i$ is the algorithm's estimate (e.g.

of the x-position of the sphere), for the i^{th} set of data, and x_i is the truth for that data set[1].

If the bias can be characterized accurately, it can often be removed by simply subtracting the bias from the algorithm's estimate. For this reason, the noise is generally considered to be the more important of the two measures.

Sphere Fitting Algorithm

Figure 3 plots the position error of the sphere fitting algorithm. This graph shows that both the bias and the noise of the x-coordinate are well within requirements at all distances. However, the z-coordinate has problems at all distances, while the y-coordinate is within specifications except at a distance of one meter.

This pattern shown in the bias graph suggests that the best fitting sphere center found by the algorithm tends to be pulled in the direction of the highest density of measured points. Also, the noise graph shows that there tends to be more noise along the direction in which the center is pulled.

This is not a problem in the x-dimension. Because the scanning pattern is fast in the x-dimension, the density of points is about equal on either side of the true sphere center along the x-dimension. For this reason, there is little bias, and little noise along the x-dimension.

At most distances, there is little bias or noise in the y-dimension because there are about as many measurements to the top of the sphere as to the bottom. At short distances, we no longer are able to gather a complete frame in the measurement interval. The bias of the algorithm then shows up as a tendency to move the center up or down, towards whichever side of the sphere was scanned. The noise also increases in the same dimension as the bias.

The z-dimension has a problem at all distances because measurements are only made to the closer face of the sphere. Along the z-dimension there is always a higher density of points on the close side of the sphere center than on the far side. The algorithm is thus biased towards reporting the sphere as being closer than it actually is. In addition, there is consistently high noise along this dimension.

Table 3 shows the maximum and minimum CPU times among the 200 trials at each distance. Note that there is quite a variation between the fastest and slowest times at each distance. This is because the fitting algorithm used is an iterative algorithm that finishes only when the required level of accuracy is attained.

[1] Note that this implies that the truth may change with the data set. In fact, the trajectory of the sphere is identical in all $n = 200$ trials. However, in order to reduce noise, the algorithm is allowed to choose the time at which it will report the position. The value of x_i used in equations 10 and 11 is the truth at the time used for $\hat{x}_i$ by the algorithm.

Dist. (meters)	Min Time (seconds)	Max Time (seconds)
1	0.08	0.87
2	0.11	0.28
5	0.09	0.18
10	0.08	0.20
15	0.04	0.13
20	0.06	0.12

Table 3. CPU times for single trials of Sphere Fitting Algorithm

Timings were done on a 200 MHz Pentium Pro. One of our requirements is that the algorithm must run in 0.5 seconds on a 12.5 MHz processor. If we multiply each of the numbers in table 3 by 16 to approximate the speed of the algorithm on the slower processor, we see that even the fastest runs reported in the table would not complete in the requisite time.

Our analysis of this algorithm indicates that it is too slow, it is too noisy, and has a number of biases. Since the bias problems are so easily characterized, it should not be too difficult to solve them. Likewise, we may find ways to speed up the algorithm, either by optimizing the code, or by starting the iteration process with better initial estimates of the true center position. It may even be possible to reduce the noise. However, we have not attempted to do any of these things because we have found the performance of the Scan Line Extent algorithm to be far superior.

Scan Line Extent Algorithm

Figure 4 shows the position error of the Scan Line Extent algorithm at various distances. Once again, these errors are collected from 200 trials at each distance. First, note that the scale of these graphs is much different than for those of figure 3. All of the noise values reported here are under 8 mm, and therefore meet our requirement.

The noise on the x component is smallest. This is to be expected, since in this algorithm the x component is measured more directly than the other two components. Both the y and z components depend heavily on measurements made in the x-dimension, and therefore any noise in the x-dimension would likely be reflected in y and z as well.

The increase in noise (and also in bias) at one meter in the y-dimension is likely an indication of the influence of less than a full frame of data being collected at this distance. It may be possible to reduce this influence by using a weighted regression scheme rather than the straightforward linear regression currently used to combine the estimates from different scan lines.

Looking at the plot of biases, the only disturbing trend is in the z component. The algorithm tends to underestimate

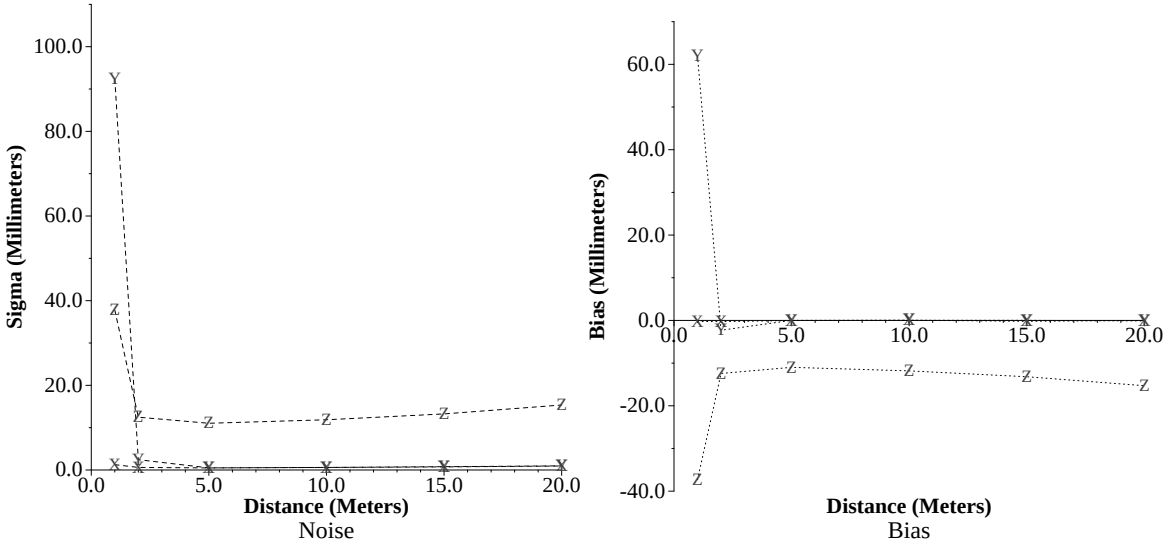

Figure 3. Position error of Sphere Fitting algorithm

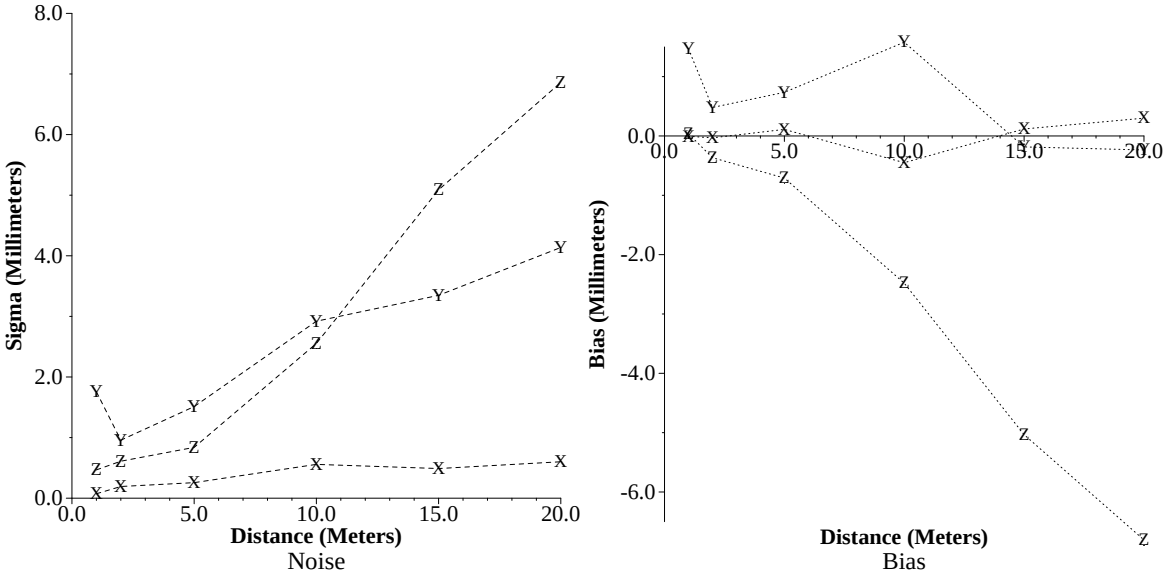

Figure 4. Position error of Scan Line Extent algorithm

the distance to the sphere, and this trend increases with increased distance. Further investigation of the algorithm may reveal the cause of this bias, and supply a remedy. However, even without a deeper understanding of the behavior, we expect much of this bias can be removed by making an adjustment based on the z estimate. This adjustment would grow with z, effectively countering much of the bias inherent in the algorithm.

In addition to a position estimate, the Scan Line Extent algorithm returns an estimate of the velocity of the sphere as well. We have no specified requirements to meet on this estimate. However, if it is desired that the time of the position estimate we supply be other than that of the most accurate time chosen by the algorithm, such an estimate can be made using the velocity estimate. This may result in an increase in noise and perhaps in bias. To give an idea of how significant this may be, figure 5 plots the noise and bias on the velocity estimate of the Scan Line Extent algorithm collected from 200 trials at each distance.

Table 4 shows the minimum and maximum CPU times required across 200 trials for the Scan Line Extent Algorithm at each of six distances. Notice that there is comparatively little difference between the minimum and maximum times reported at any given distance. Unlike the Sphere Fitting Algorithm, this algorithm is not iterative. Approximately the same number of computations is done for each run at a given distance. There is some expected variation due to differences in the number of data points collected at varying distances, and differences in the number of scan lines.

If we multiply the worst case time in this table (0.0107) by 16 to account for the difference in processing speed be-

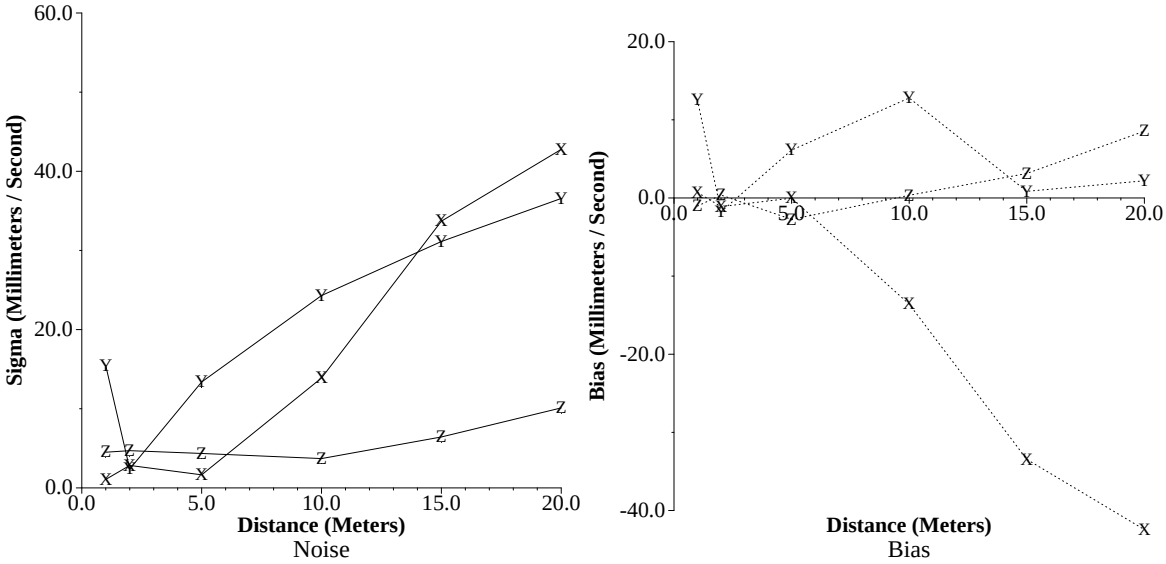

Figure 5. Velocity error of Scan Line Extent algorithm

Dist. (meters)	Min Time (seconds)	Max Time seconds
1	0.0063	0.0070
2	0.0062	0.0077
5	0.0075	0.0096
10	0.0085	0.0107
15	0.0086	0.0095
20	0.0084	0.0093

Table 4. CPU times for single trials of Sphere Fitting Algorithm

tween the 200 MHz machine on which these timings were performed, and the target 12.5 MHz machine, we see that the resulting time (0.1712 seconds) is still comfortably under the 0.5 second limit set by our requirements.

The Scan Line Extent algorithm turns out to be both faster and more accurate than the Sphere Fitting algorithm.

6. Conclusions

For this study, we have developed two algorithms to interpret range images of a moving sphere produced by a scanning laser range finder. Though both algorithms are successful to some degree, we have found that the Scan Line Extent algorithm is both faster and more accurate than the Sphere Fitting algorithm. In addition, we have demonstrated that, under a number of assumptions discussed herein, the Scan Line Extent algorithm is capable of meeting both the noise and CPU time requirements of the RSR mission.

7. Acknowledgments

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Appendix

1. Detailed Description of the Scan Line Extent Algorithm

A scan line consists of the measurements of the positions of a set of m points in a line across the face of the sphere. Each point i consists of a measurement of azimuth (α_i), elevation (ϵ_i), and range (r_i) to the point.

For each point, we also have the time (t_i) at which the measurement was taken. Although the sphere is in motion during the scan, we assume that the movement of the sphere during the scanning of a single line is insignificant compared to the other sources of error. Consequently, we treat all the positions in the scan line as if they were collected at the same moment in time.

The scanning methodology used assures us that the configuration of points is very nearly linear and approximately parallel with the x-axis. However, to improve the alignment with the x-axis, we apply a rotation around the z-axis. As shown in figure 6, to find the rotation angle, we fit a regression line to the original scan line points to predict the elevation component of each point ϵ_i from the azimuth α_i. The slope of the regression line θ is the rotation angle we use to do the alignment. The intercept of the regression line ϕ is the mean azimuth of the scan line after the rotation is applied.

The equations for rotating the original point $(\alpha_i, \epsilon_i, r_i)$ to produce a point $(\dot{\alpha}_i, \dot{\epsilon}_i, \dot{r}_i)$ in the new space follow. Note that the range measurement $\dot{r}_i$ is not changed by this transformation.

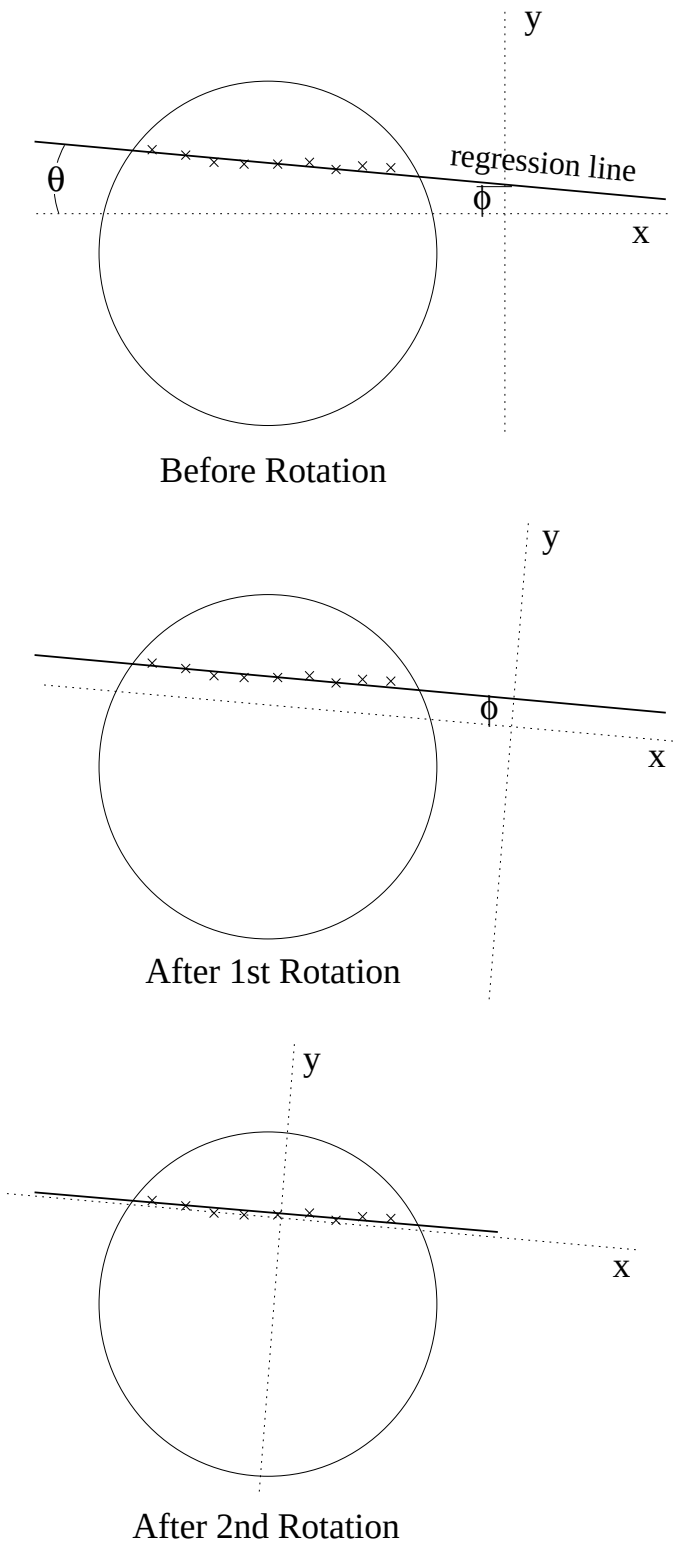

Figure 6. Aligning reference frame with scan line

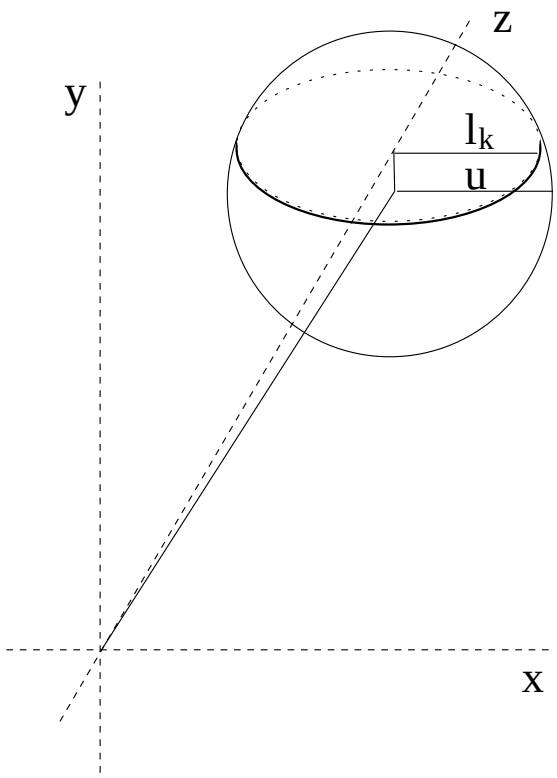

Figure 7. Position of sphere in final transformed space

$$\dot{\alpha}_i = \alpha_i \cos\theta - \epsilon_i \sin\theta \tag{12}$$

$$\dot{\epsilon}_i = \alpha_i \sin\theta + \epsilon_i \cos\theta \tag{13}$$

$$\dot{r}_i = r_i \tag{14}$$

A second rotation is applied to the space to align the scan line with the x-axis. The rotation moves the scan line so that it is parallel to the x-axis, and so that its center is on the z-axis. The center of the scan line is defined as the mid-point between the two ends. Once again, the range is not changed by this transformation. Figure 7 shows the position of the sphere in the final transformed space.

$$\ddot{\alpha}_i = \dot{\alpha}_i - \frac{\dot{\alpha}_m + \dot{\alpha}_1}{2} \tag{15}$$

$$\ddot{\epsilon}_i = \dot{\epsilon}_i - \phi \tag{16}$$

$$\ddot{r}_i = \dot{r}_i \tag{17}$$

The azimuth and elevation components of a measurement are much more accurate than is the range. Therefore, we concentrate on using these angular measurements as effectively as possible. Note that we have already used the average of the two end points to run the y-axis through the center of the sphere. This produces the trivial estimate of the azimuth of the center of the sphere $\ddot{A}$ in the transformed space given in equation 18. This estimate turns out to be

very accurate. To get an estimate in the original space the two transformations will eventually be inverted.

$$\ddot{A} = 0 \tag{18}$$

Next, we leverage the angle measurements to produce an accurate estimate of the range to the center of the sphere. The angular extent in azimuth λ, as calculated in equation 19 of the scan line could be used to establish the distance to the sphere if only we knew the scan line length l e.g. in centimeters. Unfortunately, we do not know l. Note that l is the radius of the circle defined by the scan line slicing through the sphere. We do know the radius of the sphere u since this is a constant 8 cm. However, the scan line does not run through the center of the sphere, so its length will be somewhat less than this.

$$\lambda = \frac{\ddot{\alpha}_m - \ddot{\alpha}_1}{2} \tag{19}$$

Instead, we make a rough estimate s_0 of the distance to the center of the scan line slice circle. From this, we can make a first estimate of the unknown scan line length l_0 as in equation 21. The estimate of s_0 that we use is the maximum range of the two extreme scan line points.

$$s_0 = \max(\ddot{r}_1, \ddot{r}_m) \tag{20}$$

$$l_0 = s_0 \tan\frac{\lambda}{2} \tag{21}$$

This estimate of l_0 is robust to errors in s_0. At a distance of 1 meter, an error of 5 cm in s_0 results in an error of only 2 mm in l_0. The estimate is even more robust as the sphere moves farther away.

Now, it is possible to produce an improved estimate s_1 of the distance to the center of the scan line slice circle based on s_0, l_0, and the range $\ddot{r}_i$ and azimuth $\ddot{\alpha}_i$ measurements of all the scan line points. This method may be generalized to produce an improved s_k and l_k from s_{k-1} and l_{k-1} using the equations below. (See figure 8 for a definition of terms.) However, we have found that s_1 provides an adequate estimate without further iteration.

$$l_k = s_{k-1} \tan\frac{\lambda}{2} \tag{22}$$

$$d_i = \ddot{r}_i \cos\ddot{\alpha}_i \tag{23}$$

$$x_i = \ddot{r}_i \sin\ddot{\alpha}_i \tag{24}$$

$$z_{ik} = \sqrt{l_k^2 - x_i^2} \tag{25}$$

$$s_k = \frac{1}{m}\sum_{i=1}^{m}[d_i + z_{ik}] \tag{26}$$

Note that s_k is not an estimation of the range to the center of the sphere $\ddot{R}$, but an estimate of the range to the center of the circle defined by the intersection of the scan line plane with the sphere. If we knew the elevation of the center of the sphere $\ddot{E}$, we could calculate the range component of the center of the sphere $\ddot{R}$ using the following equation.

$$\ddot{R} = s_k \cos\ddot{E} \tag{27}$$

We estimate $\ddot{E}$ using the following set of equations. See also figure 9.

$$y = \pm\sqrt{u^2 - l_k^2} \tag{28}$$

$$\ddot{E} = \pm \arctan\frac{y}{s_k} \tag{29}$$

One problem with this set of equations is that it is not clear whether the center of the sphere is above or below the scan line. In other words, the sign of y and of $\ddot{E}$ is ambiguous. If we are limited to the information available in a single scan line, it is not possible to distinguish between these two cases.

We solve this problem by looking at the lengths of all the scan lines in a frame. Generally, the elevation of longest scan line will be close to that of the center of the sphere. Scan lines with larger elevations are above the sphere center, so the signs in equations 28 and 29 are negative. Conversely, scan line with smaller elevations are below the sphere center, so the signs are positive.

In order to return our center position estimate to the original space, we must reverse the rotations of equations 12 through 17. This is done using the following equations.

$$\dot{A} = \ddot{A} + \frac{\dot{\alpha}_m + \dot{alpha}_1}{2} \tag{30}$$

$$\dot{E} = \ddot{E} + \phi \tag{31}$$

$$A = \dot{A}\cos(-\theta) - \dot{E}\sin(-\theta) \tag{32}$$

$$E = \dot{A}\sin(-\theta) + \dot{E}\cos(-\theta) \tag{33}$$

The method described above produces, for each of n scan lines j, an independent estimate of the position of the sphere center. To distinguish between the position estimates returned from different scan lines, we add a subscript, so (A_j, E_j, R_j) refers to the center position estimate produced from the j^{th} scan line. The time stamp for this position estimate measurement T_j is taken as the mean of the times for the two end points of the scan line.

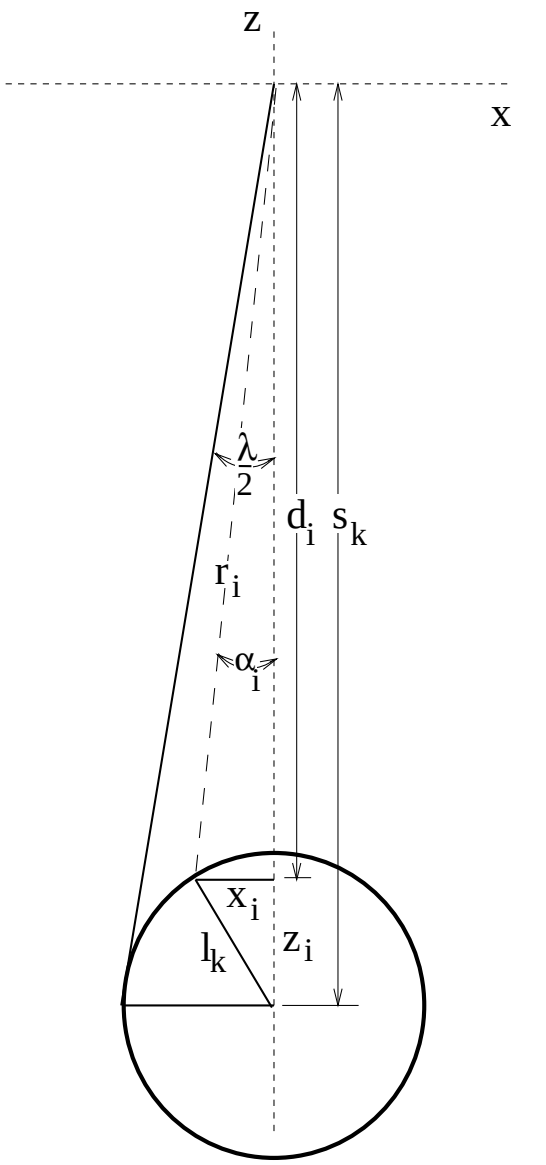

Figure 8. The configuration as viewed from top

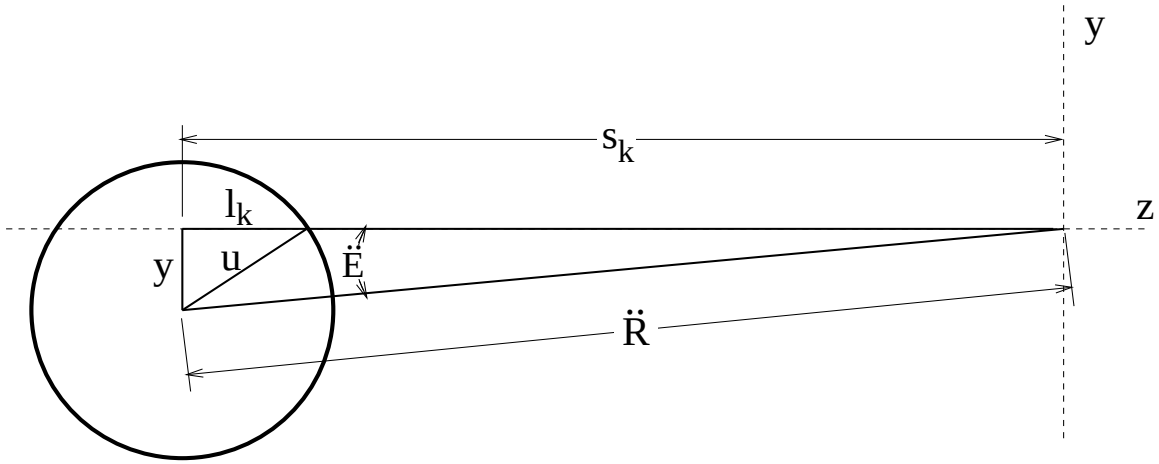

Figure 9. The configuration as viewed from side

$$T_j = \frac{t_{j1} + t_{jm_j}}{2} \tag{34}$$

We combine this set of timed independent position estimates using linear regression. First, we convert the spherical coordinates to Cartesian coordinates using the following equations.

$$X_j = R_j \sin A_j \cos E_j \tag{35}$$

$$Y_j = R_j \sin E_j \tag{36}$$

$$Z_j = R_j \cos A_j \cos E_j \tag{37}$$

Next, for each of the three spatial dimensions, we perform standard linear regression. In this discussion, we use the X-dimension as an example. Equations for the Y and Z dimensions are analogous.

Linear regression finds the velocity $\dot{X}$ and starting position X_0 which reduce the sum of squared errors for the following linear prediction of the position of each X_j from the corresponding time T_j.

$$X_j = \dot{X} T_j + X_0 \tag{38}$$

The following equations define the best fitting line.

$$\dot{X} = \frac{n \sum_{j=1}^{n} X_j T_j - \sum_{j=1}^{n} X_j \sum_{j=1}^{n} Y_j}{n \sum_{j=1}^{n} T_j^2 - (\sum_{j=1}^{n} T_j)^2} \tag{39}$$

$$X_0 = \frac{\sum_{j=1}^{n} X_j - \dot{X} \sum_{j=1}^{n} T_j}{n} \tag{40}$$

The output of the three linear regressions is an estimate of the starting position (X_0, Y_0, Z_0) and the velocity $(\dot{X}, \dot{Y}, \dot{Z})$ of the sphere center. From this we can predict the position of the sphere at any point in time. However, the most accurate prediction will be at the average scan line time $\bar{T} = \frac{1}{n} \sum_{j=1}^{n} T_j$.

REFERENCES

[1] Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vision*, volume 1. Addison-Wesley, Reading, MA, 1992.

[2] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, MA, 2nd edition, 1992.

Daniel Clouse *received the B.A. degree in Computer Science from the University of California, Berkeley in 1982, the M.S. degree in Computer Science from the University of California, San Diego (UCSD) in 1992, and the Ph.D. degree in Cognitive Science and Computer Science from UCSD in 1998. He is a Member of the Technical Staff at the Jet Propulsion Laboratory, California Institute of Technology in the Machine Vision group of the Autonomy and Control Section. His interests include vision processing, neural networks, language translation, and word sense disambiguation.*

Curtis Padgett *received his Ph.D. from the Computer Science and Engineering Department at the University of California at San Diego (UCSD) in 1998. He received his M.S. degree from the same department in 1992. He has been employed at the Jet Propulsion Laboratory, California Institute of Technology, Pasadena since 1993. He is a Member of the Technical Staff in the Machine Vision group of the Autonomy and Control Section where he works on remote sensing applications for space systems. His research interests include algorithm optimization, machine vision, and artificial intelligence applied in classification and pattern recognition tasks.*