A Progress Report
NASA Grant No. NGT-51055
September 1, 1993 - December 15, 1993

DETERMINING THE 3-D STRUCTURE AND MOTION OF OBJECTS USING A SCANNING LASER RANGE SENSOR

Submitted to:

Dr. Jon D. Erickson
Chief Scientist
Automation and Robotics Division
Mail Code ER
NASA - Lyndon B. Johnson Space Center
Houston, TX 77058

Submitted by:

N. Nandhakumar
Assistant Professor

Philip W. Smith
Graduate Student

SEAS Report No. UVA/528476/EE94/101
December 1993

DEPARTMENT OF ELECTRICAL ENGINEERING

SCHOOL OF
ENGINEERING
& APPLIED SCIENCE

University of Virginia
Thornton Hall
Charlottesville, VA 22903

A Progress Report
NASA Grant No. NGT-51055
September 1, 1993 - December 15, 1993

DETERMINING THE 3-D STRUCTURE AND MOTION OF OBJECTS USING A SCANNING LASER RANGE SENSOR

Submitted to:

Dr. Jon D. Erickson
Chief Scientist
Automation and Robotics Division
Mail Code ER
NASA - Lyndon B. Johnson Space Center
Houston, TX 77058

Submitted by:

N. Nandhakumar
Assistant Professor

Philip Smith
Graduate Student

Department of Electrical Engineering
SCHOOL OF ENGINEERING AND APPLIED SCIENCE
UNIVERSITY OF VIRGINIA
CHARLOTTESVILLE, VIRGINIA

Report No. UVA/528476/EE94/101
December 1993

Copy No._____

TABLE OF CONTENTS

1 Problem Review

In order for the EVAHR robot to autonomously track and grasp objects, its vision system must be able to determine the 3-D structure and motion of an object from a sequence of sensory images. This task is accomplished by the use of a laser radar range sensor which provides dense range maps of the scene. Unfortunately, the currently available laser radar range cameras use a sequential scanning approach which complicates image analysis. Although many algorithms have been developed for recognizing objects from range images, none are suited for use with single beam, scanning, time-of-flight sensors because all previous algorithms assume instantaneous acquisition of the entire image. This assumption is invalid since the EVAHR robot is equipped with a sequential scanning laser range sensor. If an object is moving while being imaged by the device, the apparent structure of the object can be significantly distorted due to the significant non-zero delay time between sampling each image pixel. If an estimate of the motion of the object can be determined, this distortion can be eliminated; but, this leads to the motion-structure paradox - most existing algorithms for 3-D motion estimation use the structure of objects to parameterize their motions. The goal of this research is to design a rigid-body motion recovery technique which overcomes this limitation. The method being developed is an iterative, linear, feature-based approach which uses the non-zero image acquisition time constraint to accurately recover the motion parameters from the distorted structure of the 3-D range maps. Once the motion parameters are determined, the structural distortion in the range images is corrected.

2 Accomplishments 9/1/93 - 12/15/93

In the initial phase of this research, the following issues were addressed:

- The motion recovery algorithm was developed.
- The behavior of the algorithm in the absence of correspondence error was verified using a computer simulation in which the correspondence between object features in successive image frames was known exactly.
- The behavior of the algorithm in the presence of feature correspondence errors was studied using simulated laser radar imagery with correspondence performed by a human observer.
- The stability and convergence of the iterative algorithm were studied and a formal, analytical proof of its stability was developed.
- The above research resulted in the submission of a paper to the *1994 IEEE Conference on Computer Vision and Pattern Recognition.*

2.1 The Motion Recovery Algorithm

- **Initialize**
  - Use the sequence of reflectance and range images to extract and establish correspondence between N object point pairs $(p^k_{i,j}, p'^k_{i',j'})$, $k = 0, 1, \ldots, N-1$.
  - Compute the time between sampling for the kth pair of points, Δt_k, for each value of $k = 0, 1, \ldots, N-1$ and calculate the average acquisition delay time, $t_{av} = \frac{1}{N}\sum_{k=0}^{N-1} \Delta t_k$.
  - Set $\mathbf{A}_{k,0} = 0$ for each value of $k = 0, 1, \ldots, N-1$, and set the iteration counter $m = 0$.
  - Use the relation $\frac{p'^k_{i',j'}}{\Delta t_k} \simeq \mathbf{R}_0(t_{av})\frac{p^k_{i,j}}{t_{avg}} + \mathbf{T}_{u,0}$ to estimate $(\mathbf{R}_0(t_{av}), \mathbf{T}_{u,0})$.
- **Loop**

- Use $(\mathbf{R}_m(t_{av}), \mathbf{T}_{u,m})$ to estimate the velocity of the object, $(\dot{\Theta}_m, \dot{\Phi}_m, \dot{\Upsilon}_m, \dot{X}_m, \dot{Y}_m, \dot{Z}_m)$.
- Use the rotational velocity estimates to compute $\mathbf{R}_m(\Delta t_k)$ and $\mathbf{A}_{k,m}$, $k = 0, 1, \ldots, N-1$.
- Use the relation $\frac{p'^k_{i',j'}}{\Delta t_k} - \mathbf{A}_{k,m}\frac{p^k_{i,j}}{\Delta t_k} = \mathbf{R}_{m+1}(t_{av})\frac{p^k_{i,j}}{\Delta t_k} + \mathbf{T}_{u,m+1}$ to form a system of linear equations. Solve this system to refine the estimate of the motion transformation $(\mathbf{R}_{m+1}(t_{av}), \mathbf{T}_{u,m+1})$.
- Let m = m+1 and repeat the loop until the change in any of the motion parameters is below some user defined threshold.

- **Structural Distortion Removal**
  - For each point $p_{i,j}$, determine its 3-D coordinate at time $t = 0$ using the relation $v_{p_{i,j}} = \mathbf{R}^{-1}(\delta t)[p_{i,j} - \mathbf{T}(\delta t)]$ where $\delta t = T_s(i + nj)$ and T_s is the inter-pixel sampling time.

2.2 Simulation with Exact Feature Correspondence

In order to verify the behavior of the motion recovery algorithm without the effects of feature correspondence error, a computer simulation was developed using MATLAB and an IBM RS6000 workstation. The test data were generated by defining eight points on an object and eight different transformations based on the Euler angle representation of the rotation matrix. Each transformation was calculated using the same rotational and translational velocities, but different, randomly chosen sampling delay times, Δt_k. The rotational and translational velocities used in this test were randomly chosen to be $\dot{\Theta} = 1.3$ rads/sec, $\dot{\Phi} = 0.9$ rads/sec, $\dot{\Upsilon} = -1.2$ rads/sec, $v_x = 90$ units/sec, $v_y = -80$ units/sec, and $v_z = 30$ units/sec. Each point was then "set in motion" using its respective transformation as is shown in Figure 1(b). The result of these transformations thus produced the set $\{p'^k_{i',j'}\}$ while the original object points formed the set $\{p^k_{i,j}\}$. The iterative algorithm described above was applied to these data sets, and the resulting velocity estimates were then used to reconstruct the actual three dimensional structure of the object.

As can be seen in Figure 3, the algorithm converged rapidly to the proper rotational and translational velocities after only a small number of iterations. Figure 2 shows the result of reconstructing the true structure of the object using the velocity estimates produced by the iterative technique. The structural distortion caused by the different times of rotation for each point has been eliminated.

2.3 Experiment Using Simulated Laser Radar Images

The iterative motion recovery algorithm was then tested using simulated laser radar images of an object moving against a background of constant depth. The purpose of these tests was to determine the effects of correspondence errors on the convergence of the algorithm. Figure 4 shows a two frame sequence of simulated laser range images (taken with a frame rate of 2 *frames/sec*) of four "reflectors" on the same object undergoing rigid body motion. Correspondence between the points was determined manually. The three dimensional points extracted from the images in Figures 4(a) and (b) were then used to form the sets $\{p^k_{i,j}\}$ and $\{p'^k_{i',j'}\}$, respectively. The iterative motion recovery algorithm was then applied to these data sets, and the resulting velocity estimates were then used to reconstruct the actual three dimensional structure of the object.

The actual rotational and translational velocities of the object and their estimates after a number of iterations are shown in Table 1. The convergence of the estimates is shown in Figure 5. While the velocity estimates still converge to a finite value, there is measurable error in the estimate of the velocities. These discrepancies between the estimated and the actual velocities are due to errors in determining point correspondence between the two range images. The version of the iterative algorithm employed in this test used a simple least squares algorithm to update the estimates of $R_m(t_{av})$ and $T_{u,m}$ at each iteration level, producing considerable cumulative error in each velocity estimate. If a more robust method for solving systems of linear equations is employed, the effect of these accumulated errors could be minimized further. Note, however, that the iterative algorithm provided more accurate measures of the object's velocity than the initial estimate provided.

2.4 Stability of the Iterative Estimation Algorithm

In order for the algorithm described in Section 2 to provide reasonable estimates of the rotational and translational velocities of objects moving within the range field, the adaptive law governing the estimator must be stable in the sense of Lyapunov. Below two theorems are included which will be employed to prove that the iterative procedure is stable.

Theorem 1 *If matrices* $\mathbf{Q}_1, \mathbf{Q}_2 \in \mathcal{C}^{n\times n}$ *are such that* $\mathbf{Q}_1 = [\mathbf{P}_1 \vdots \mathbf{P}_2]$ *and* $\mathbf{Q}_2 = [\mathbf{P}_1 \vdots 0]$ *where* $\mathbf{P}_1 \in \mathcal{C}^{n\times m}$ *and* $\mathbf{P}_2 \in \mathcal{C}^{n\times(n-m)}$, *then* $\mathbf{Q}_1^{-1}\mathbf{Q}_2 = diag(\mathbf{I}_{m\times m}, 0_{(n-m)\times(n-m)})$ *where* $\mathbf{I}_{m\times m}$ *is the* $m \times m$ *identity matrix.*

Proof:

$$\begin{aligned}\mathbf{Q}_2 &= [\mathbf{P}_1 \vdots 0] \\ &= [\mathbf{P}_1 \vdots \mathbf{P}_2]\begin{bmatrix} \mathbf{I}_{m\times m} & 0_{m\times(n-m)} \\ \cdots & \cdots \\ 0_{(n-m)\times m} & 0_{(n-m)\times(n-m)} \end{bmatrix} \\ &= \mathbf{Q}_1\begin{bmatrix} \mathbf{I}_{m\times m} & 0_{m\times(n-m)} \\ 0_{(n-m)\times m} & 0_{(n-m)\times(n-m)} \end{bmatrix}\end{aligned}$$

and thus,

$$\mathbf{Q}_1^{-1}\mathbf{Q}_2 = \begin{bmatrix} \mathbf{I}_{m\times m} & 0_{m\times(n-m)} \\ 0_{(n-m)\times m} & 0_{(n-m)\times(n-m)} \end{bmatrix} \quad \text{Q.E.D.}$$

Theorem 2 *The equilibrium state* $x=0$ *of a linear, time-invariant, discrete time system of the form* $x[k+1] = \mathbf{E}x[k] + \mathbf{F}u[k]$ *where* $\mathbf{E} \in \mathcal{C}^{n\times n}$, $\mathbf{F} \in \mathcal{C}^{n\times m}$, $x[k] \in \mathcal{R}^n$, *and* $u[k] \in \mathcal{R}^m$, *is stable if and only if the spectral radius* $|\rho(\mathbf{E})| \leq 1$ *and every eigenvalue of* $\mathbf{E}$ *with a magnitude of one is a simple zero of the minimum polynomial of* $\mathbf{E}$.

Proof: the proof is omitted here because it is beyond the scope of this report and can be found in almost any graduate level text in modern systems control.

In order to use theorem 2 to determine the stability of the iterative motion recovery process, the algorithm's adaptive law must be expressed in the form of a linear, time invariant discrete time system. The relation $\frac{p'^k_{i',j'}}{\Delta t_k} - \mathbf{A}_{k,m}\frac{p^k_{i,j}}{\Delta t_k} = \mathbf{R}_{m+1}(t_{av})\frac{p^k_{i,j}}{\Delta t_k} + \mathbf{T}_{u,m+1}$ used to update the estimate of the motion transformation in the iteration loop can be rewritten to yield

$$p'^k_{i',j'} - \mathbf{A}_k[m]p^k_{i,j} = \mathbf{R}_a[m]p^k_{i,j} + \Delta t_k \mathbf{T}_u[m]. \tag{1}$$

by letting $\mathbf{R}_a[m] = \mathbf{R}_m(t_{av})$, $\mathbf{R}_k[m] = \mathbf{R}_m(\Delta t_k)$, $\mathbf{A}_k[m] = \mathbf{A}_{k,m}$, and $\mathbf{T}_u[m] = \mathbf{T}_{u,m}$, where m represents the mth iteration step. This equation produces three equations in twelve unknowns, so a set of at least four three dimensional points must be used to solve for $R_a[m]$. Thus,

$$\begin{bmatrix} p'^1_{i',j'} \\ p'^2_{i',j'} \\ p'^3_{i',j'} \\ p'^4_{i',j'} \end{bmatrix} - \begin{bmatrix} \mathbf{A}_1[m] & 0 & 0 & 0 \\ 0 & \mathbf{A}_2[m] & 0 & 0 \\ 0 & 0 & \mathbf{A}_3[m] & 0 \\ 0 & 0 & 0 & \mathbf{A}_4[m] \end{bmatrix} =$$

$$\begin{bmatrix} \mathbf{R}_a[m] & 0 & 0 & 0 \\ 0 & \mathbf{R}_a[m] & 0 & 0 \\ 0 & 0 & \mathbf{R}_a[m] & 0 \\ 0 & 0 & 0 & \mathbf{R}_a[m] \end{bmatrix}\begin{bmatrix} p^1_{i,j} \\ p^2_{i,j} \\ p^3_{i,j} \\ p^4_{i,j} \end{bmatrix} + \begin{bmatrix} \mathbf{T}_u[m] & 0 & 0 & 0 \\ 0 & \mathbf{T}_u[m] & 0 & 0 \\ 0 & 0 & \mathbf{T}_u[m] & 0 \\ 0 & 0 & 0 & \mathbf{T}_u[m] \end{bmatrix}\begin{bmatrix} \Delta t_1 \\ \Delta t_2 \\ \Delta t_3 \\ \Delta t_4 \end{bmatrix} \tag{2}$$

can be rewritten in terms of the unknown elements of $\mathbf{R}_a[m]$ and $\mathbf{T}_u[m]$ to yield

$$\bar{p}' - \bar{\mathbf{A}}[m]\bar{p} = \mathbf{W}_1\bar{r}[m] \tag{3}$$

where

- $\bar{p}' = \left[\begin{array}{cccc} (p_{i',j'}^{\prime 1})^T & (p_{i',j'}^{\prime 2})^T & (p_{i',j'}^{\prime 3})^T & (p_{i',j'}^{\prime 4})^T \end{array} \right]^T$
- $\bar{\mathbf{A}}[m] = \mathrm{diag}(\mathbf{A}_1[m], \mathbf{A}_2[m], \mathbf{A}_3[m], \mathbf{A}_4[m])$
- $\bar{p} = \left[\begin{array}{cccc} (p_{i,j}^1)^T & (p_{i,j}^2)^T & (p_{i,j}^3)^T & (p_{i,j}^4)^T \end{array} \right]^T$
- $\mathbf{W}_1 = \begin{bmatrix} (p_{i,j}^1)^T & 0 & 0 & \Delta t_1 & 0 & 0 \\ 0 & (p_{i,j}^1)^T & 0 & 0 & \Delta t_1 & 0 \\ 0 & 0 & (p_{i,j}^1)^T & 0 & 0 & \Delta t_1 \\ (p_{i,j}^2)^T & 0 & 0 & \Delta t_2 & 0 & 0 \\ 0 & (p_{i,j}^2)^T & 0 & 0 & \Delta t_2 & 0 \\ 0 & 0 & (p_{i,j}^2)^T & 0 & 0 & \Delta t_2 \\ (p_{i,j}^3)^T & 0 & 0 & \Delta t_3 & 0 & 0 \\ 0 & (p_{i,j}^3)^T & 0 & 0 & \Delta t_3 & 0 \\ 0 & 0 & (p_{i,j}^3)^T & 0 & 0 & \Delta t_3 \\ (p_{i,j}^4)^T & 0 & 0 & \Delta t_4 & 0 & 0 \\ 0 & (p_{i,j}^4)^T & 0 & 0 & \Delta t_4 & 0 \\ 0 & 0 & (p_{i,j}^4)^T & 0 & 0 & \Delta t_4 \end{bmatrix}$
- $\bar{r}[m] = \left[\begin{array}{ccccccccccc} r_{11}[m] & r_{12}[m] & r_{13}[m] & r_{21}[m] & r_{22}[m] & r_{23}[m] & r_{31}[m] & r_{32}[m] & r_{33}[m] & t_1[m] & t_2[m] & t_3[m] \end{array} \right]^T$ where $r_{ij}[m]$ represents the (i,j)th element of $\mathbf{R}_a[m]$ and $t_l[m]$ represents the lth element of $\mathbf{T}_u[m]$.

However, since $\mathbf{A}_k[m] = \mathbf{R}_k[m-1] - \mathbf{R}_a[m-1]$, the matrix $\bar{\mathbf{A}}[m]$ can be expressed as $\bar{\mathbf{A}}[m] = \bar{\mathbf{R}}_k[m-1] - \mathbf{R}_a[m]\mathbf{I}_{12\times 12}$ where $\bar{\mathbf{R}}_k[m-1] = \mathrm{diag}(\mathbf{R}_1[m-1], \mathbf{R}_2[m-1], \mathbf{R}_3[m-1], \mathbf{R}_4[m-1])$. Thus, by substitution in equation 3

$$\begin{aligned} \bar{p}' - (\bar{\mathbf{R}}_k[m-1] - \mathbf{R}_a[m-1]\mathbf{I}_{12\times 12})\bar{p} &= \mathbf{W}_1\bar{r}[m] \\ \bar{p}' - \bar{\mathbf{R}}_k[m-1]\bar{p} + \mathbf{R}_a[m-1]\mathbf{I}_{12\times 12}\bar{p} &= \mathbf{W}_1\bar{r}[m] \\ \bar{p}' - \bar{\mathbf{R}}_k[m-1]\bar{p} + \mathbf{W}_2\bar{r}[m-1] &= \mathbf{W}_1\bar{r}[m]. \end{aligned} \tag{4}$$

where

$$\mathbf{W}_2 = \begin{bmatrix} (p_{i,j}^1)^T & 0 & 0 & 0 & 0 & 0 \\ 0 & (p_{i,j}^1)^T & 0 & 0 & 0 & 0 \\ 0 & 0 & (p_{i,j}^1)^T & 0 & 0 & 0 \\ (p_{i,j}^2)^T & 0 & 0 & 0 & 0 & 0 \\ 0 & (p_{i,j}^2)^T & 0 & 0 & 0 & 0 \\ 0 & 0 & (p_{i,j}^2)^T & 0 & 0 & 0 \\ (p_{i,j}^3)^T & 0 & 0 & 0 & 0 & 0 \\ 0 & (p_{i,j}^3)^T & 0 & 0 & 0 & 0 \\ 0 & 0 & (p_{i,j}^3)^T & 0 & 0 & 0 \\ (p_{i,j}^4)^T & 0 & 0 & 0 & 0 & 0 \\ 0 & (p_{i,j}^4)^T & 0 & 0 & 0 & 0 \\ 0 & 0 & (p_{i,j}^4)^T & 0 & 0 & 0 \end{bmatrix}.$$

A slight rearrangement of equation 4 yields the following linear, time invariant discrete difference equation in terms of the unknown parameter vector, $\bar{r}[m]$,

$$\bar{r}[m+1] = \mathbf{E}\bar{r}[m] + \mathbf{F}u[m] \tag{5}$$

where

$$\mathbf{E} = \mathbf{W}_1^{-1}\mathbf{W}_2 \tag{6}$$

$$\mathbf{F} = \left[-\mathbf{W}_1^{-1} \ \vdots \ \mathbf{W}_1^{-1} \right] \tag{7}$$

$$u[m] = \left[\bar{r}_k[m]^T \ \ \bar{p}'^T \right]^T \tag{8}$$

$$\bar{r}_k[m] = \bar{\mathbf{R}}_k[m-1]\bar{p}. \tag{9}$$

By careful observation of the definitions of $\mathbf{W}_1$ and $\mathbf{W}_2$, it is noted that they are of the form

$$\mathbf{W}_1 = [\mathbf{P}_1 \ \vdots \ \mathbf{P}_2], \quad \mathbf{W}_2 = [\mathbf{P}_1 \ \vdots \ 0].$$

Thus, theorem 1 can be used to show that

$$\mathbf{E} = \mathbf{W}_1^{-1}\mathbf{W}_2 = \mathrm{diag}(\mathbf{I}_{9\times 9}, 0_{3\times 3}) \tag{10}$$

So, since $\mathbf{E}$ is an idempotent matrix, the spectrum, $\sigma(\mathbf{E}) \in \{0, 1\}$ and the spectral radius, $|\rho(\mathbf{E})| \leq 1$. Thus, since the eigenvalue $\lambda_1 = 1$ is a simple zero of $\mathbf{E}$ and $|\rho(\mathbf{E})| \leq 1$, the iterative motion recovery method presented in the previous subsection is stable in the sense of Lyapunov as per theorem 2. The estimate of the rotational and translational parameters is therefore always finite, since the input vector $u[m]$ is always bounded due to the fact that $\bar{p}, \bar{p}'$ are constant vectors and $\bar{\mathbf{R}}_k[m-1]$ is a unitary matrix.

3 Future Work

In the coming year, the following theoretical and experimental issues will be considered. First, the proposed algorithm will be tested using real data sets taken from the EVAHR's laser mapping sensors as soon as the system is again operational. Segmentation and correspondence will, at first, be done manually so that severe point mismatches won't have to be considered in the initial algorithm development. After analyzing the algorithm's performance on real data, the continued research will focus on the theoretical proof of convergence for the motion transformation parameters using the iterative approach described. From the analytical study and the experimentation with the actual EVAHR data sets, the conditions under which this algorithm can be used will be more concretely determined. Before the system is implemented, however, the computational aspects of the algorithm, including methods of reducing the computational complexity while increasing the accuracy of estimating the motion transformation, the numerical behavior of the various matrices, and the error tolerance in the initial estimation will be examined thoroughly. After these preliminary studies, segmentation and correspondence procedures for determining the necessary sets of point correspondence pairs for the motion estimation algorithm will be designed and developed in order to make the motion estimation procedure completely autonomous. An approximate schedule of the work is shown below.

Proposed Schedule for 1/1/94 - 12/31/94

- Spring 1994 - Test the algorithm using real data sets obtained from the EVAHR's laser ranging system. Continue to develop a proof of convergence , and begin to study the numerical/computational issues for efficient and robust implementation of the motion estimation procedure.
- Summer 1994 - Visit NASA JSC, and continue to study numerical/implementation issues. Begin development of automatic correspondence techniques.
- Fall 1994 - Continue development of the automatic correspondence technique and begin to implement a complete system.

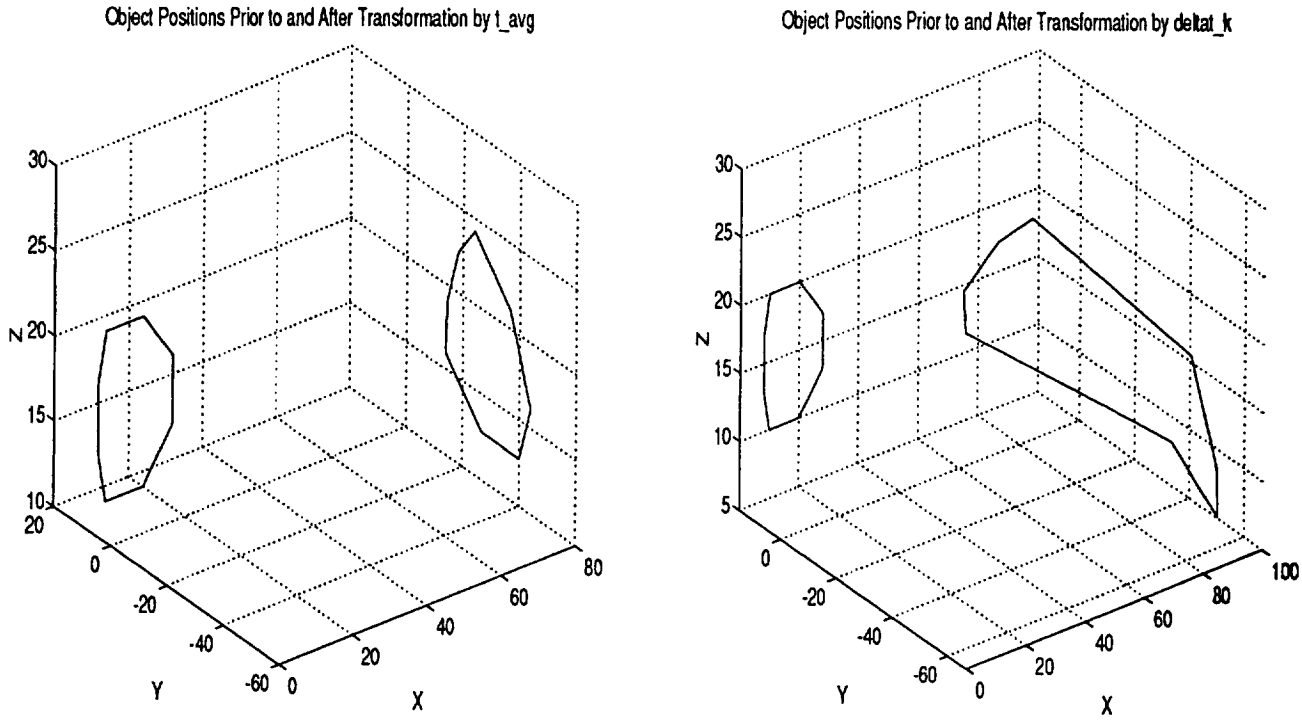


Figure 1: (a) the original object and its pose after each point is transformed using a relation based on t_{avg}; (b) the original object and its pose after each point is transformed using its own relation based on Δt_k.

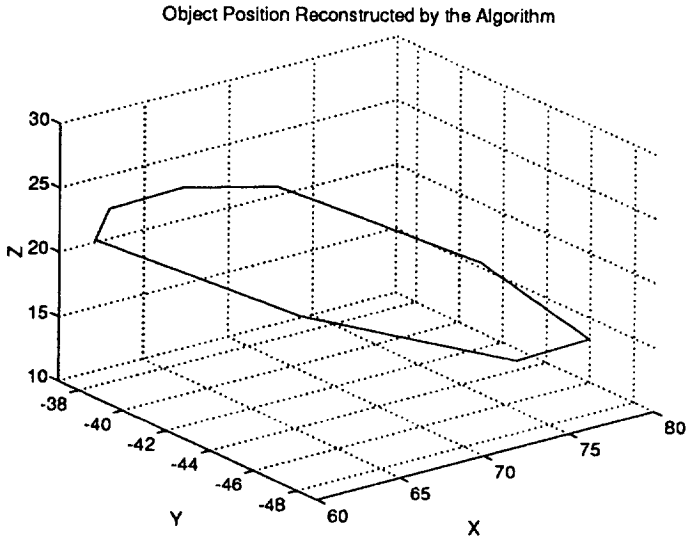


Figure 2: Object structure reconstructed using the velocity estimates generated by the iterative algorithm. The structural distortion present in the transformed object has been eliminated.

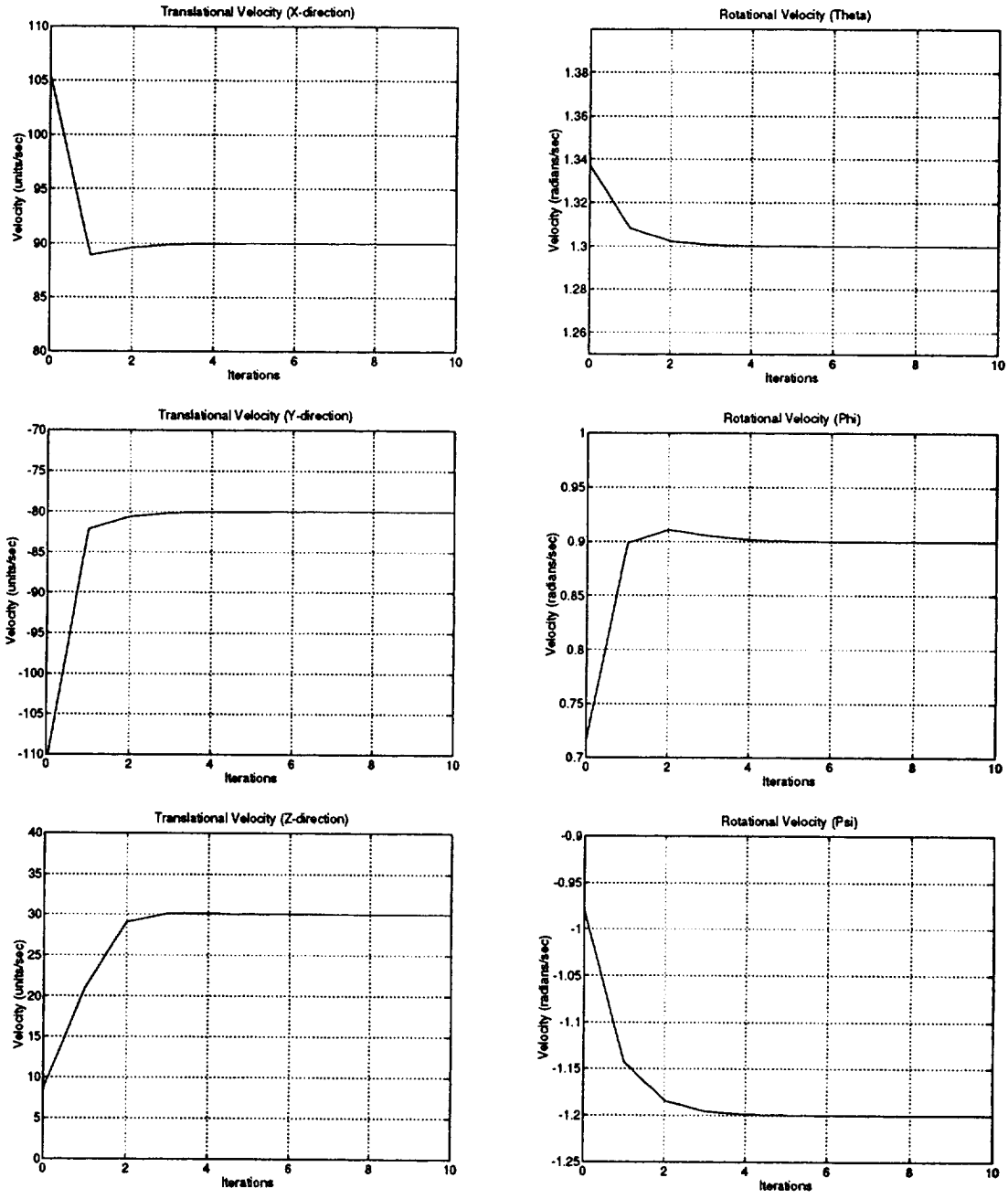
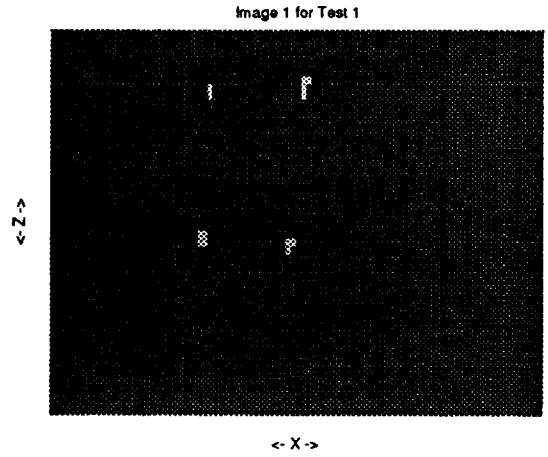
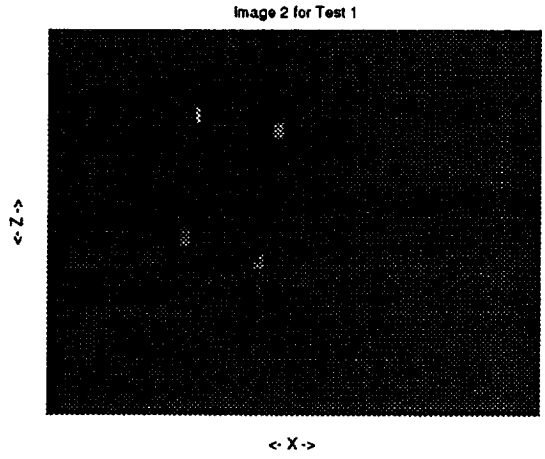


Figure 3: The velocity estimates produced using the iterative method. Each estimate converges to the object's actual velocity.



(a)



(b)

Figure 4: Laser radar range image sequence of four blocks moving with same velocity

	Actual Velocity	Estimated Velocity	Error
v_x (m/sec)	-7	-7.14	.14
v_y (m/sec)	15	15	0
v_z (m/sec)	-3	-2.64	.36
$\dot{\Theta}$ (rads/sec)	0	0	0
$\dot{\Upsilon}$ (rads/sec)	0	0	0
$\dot{\Phi}$ (rads/sec)	.2	.2	0

Table 1: A comparison of the actual velocity and estimated velocity determined by the iterative algorithm for the simulated laser range images.

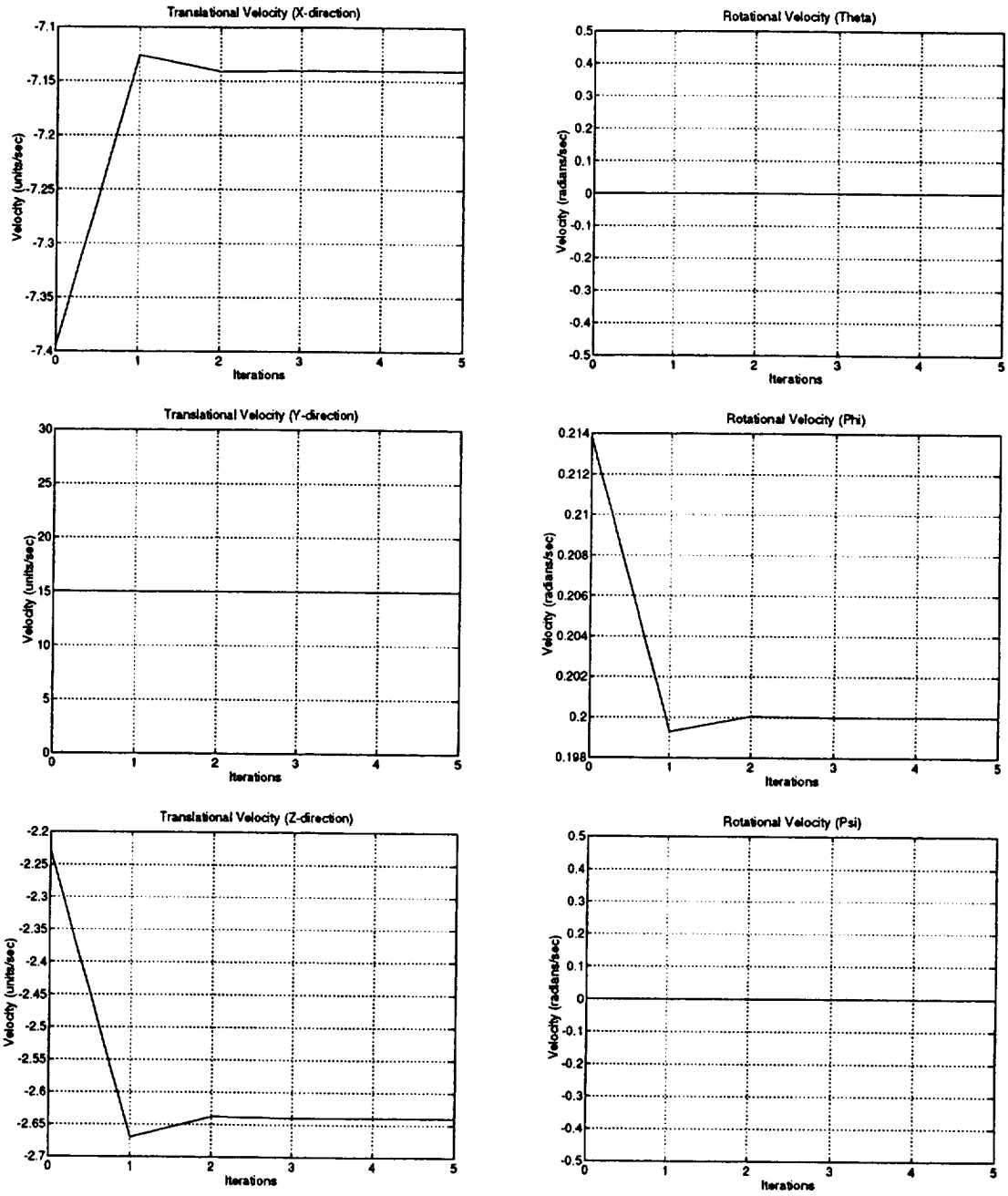


Figure 5: The velocity estimates produced using the iterative method. Each estimate converges to a value that is close to the object's true velocity. (See Table 1.)

DISTRIBUTION LIST

1 - 3 Dr. Jon D. Erickson
Chief Scientist
Automation and Robotics Division
Mail Code ER
NASA - Lyndon B. Johnson Space Center
Houston, TX 77058

4 Dr. Stanley H. Goldstein
Director, University Programs
Mail Code AHU
NASA - Lyndon B. Johnson Space Center
Houston, TX 77058

5 - 6* NASA Scientific and Technical Information Facility
P.O. Box 8757
Baltimore/Washington International Airport
Baltimore, MD 21240

7 - 8 N. Nandhakumar

9 - 10 F. O'Bryant, Clark Hall

11** SEAS Postaward Administration

12 SEAS Preaward Administration

*Mail 1 copy bound and 1 unbound

**Only the cover letter

JO#5480:pna