

# Vision-Based Relative Pose Estimation for Autonomous Rendezvous And Docking

Jed M. Kelsey, Jeffrey Byrne, Martin Cosgrove, Sanjeev Seereeram, Raman K. Mehra  
Scientific Systems Company, Inc.  
500 W. Cummings Park, Ste. 3000  
Woburn, MA 01801, USA  
781-933-5355  
{jed,jbyrne,martin,sanjeev,rkm}@ssci.com

*Abstract*—Autonomous rendezvous and docking is necessary for planned space programs such as DARPA ASTRO, NASA MSR, ISS assembly and servicing, and other rendezvous and proximity operations. Estimation of the relative pose between the host platform and a resident space object is a critical ability. We present a model-based pose refinement algorithm, part of a suite of algorithms for vision-based relative pose estimation and tracking. Algorithms were tested in high-fidelity simulation and stereo-vision hardware testbed environments. Testing indicated that in most cases, the model-based pose refinement algorithm can handle initial attitude errors up to about 20 degrees, range errors exceeding 10% of range, and transverse errors up to about 2% of range. Preliminary point tests with real camera sequences of a 1/24 scale Magellan satellite model using a simple fixed-gain tracking filter showed potential tracking performance with mean errors of  $< 3$  degrees and  $< 2\%$  of range.

tion include laser scanning and ranging sensors such as LIDAR. These sensors can provide good relative position accuracy at distances of up to several kilometers, as well as relative orientation cues (based on object shape) at shorter distances. However, these sensors are typically very expensive and power-hungry, and often have a small field of regard.

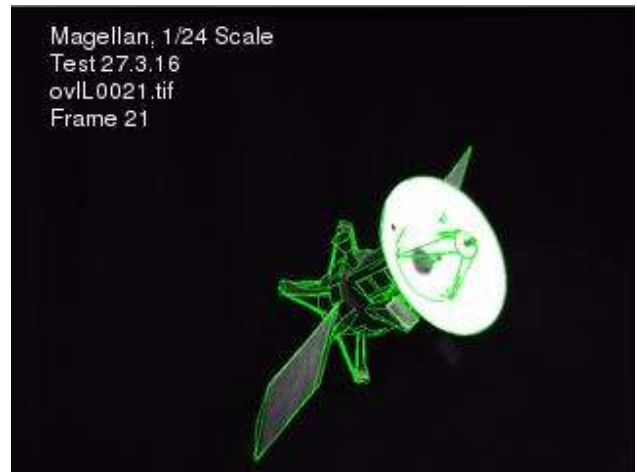
## TABLE OF CONTENTS

1	INTRODUCTION .....	1
2	POSE ESTIMATION OVERVIEW .....	2
3	MODEL-BASED POSE REFINEMENT .....	3
4	TEST ENVIRONMENT .....	6
5	ALGORITHM TESTS AND EVALUATION .....	6
6	CONCLUSIONS .....	18
7	ACKNOWLEDGEMENTS .....	18

## 1. INTRODUCTION

Guidance, Navigation and Control (GN&C) technologies for autonomous rendezvous and docking, formation flying, or proximity operations of spacecraft or other vehicles require accurate, up-to-date measurements and estimates of relative range and attitude while in close formation and during rendezvous or proximity operations.

Active sensors used for relative pose measurement and es-



**Figure 1.** Model-Based Tracking of Magellan 1/24 Scale Spacecraft over a Real Camera Sequence. Green overlay shows tracked model pose.

Inexpensive camera sensors, coupled with advanced image processing and tracking algorithms, can provide a cost-effective and accurate sensing capability to obtain full 6DOF relative pose information during proximity operations. The cameras may serve multiple functions, such as providing easily-interpreted visual cues for human observers and improving situational awareness in supervisory control applications.

A number of existing image processing algorithms for pose estimation and tracking utilize fiducial markings on the tracked object. However, retrofitting an existing satellite with these fiducial markings is impractical at best. Other pose estimation algorithms use an object surface model representation to track the object, avoiding the necessity for fiducial markings while taking advantage of the known structural configuration (including components effectively acting as fiducials).

This paper describes the results of analysis, implementation and testing of a component model-based pose estimation module intended for vision-based guidance, navigation and control applications including automated rendezvous and docking. A high level block diagram of the model-based pose refinement and tracking architecture is shown in Figure 2. The architecture is based on the Iterative-Recursive Least Squares (IRLS) pose refinement algorithm of Drummond and Cipolla [5].

We have integrated several vision-based algorithms for relative pose estimation and tracking into a prototype Vision System for Autonomous Rendezvous and Docking (VISARD). The VISARD algorithms and software were analyzed and tested using both synthesized camera imagery—based on various 3D CAD models, including an upper-stage Delta II Rocket Body—and real camera imagery from a stereo-vision hardware testbed, using scanned/digitized 1/24-scale Magellan (Figure 1) and 1/72-scale Soyuz orbiter models [20].

## 2. POSE ESTIMATION OVERVIEW

Pose estimation is the process of estimating the rigid transform that aligns a model frame with a sensor, body, or world reference frame. Image-based pose and motion estimation can be broadly classified into *model-based* and *non-model-based* techniques.

*Non-model-based* techniques do not assume *a priori* knowledge of the tracked object’s shape, textures, or other visual attributes, and typically attempt to process an image sequence to decouple the scene structure from the sensor motion. Examples from the computer vision community include stereo based egomotion estimation [15], differential or optic flow techniques [18], and structure from motion [19].

These methods can be used to solve for optimal camera motion given continuous image velocity or discrete feature correspondence measurements under epipolar or motion field constraints. They also allow general object or camera egomotion to be estimated on a frame-to-frame basis without necessarily constructing or maintaining a structural model of the tracked object.

One challenge faced by many feature-based pose tracking systems is that automatic tracking often fails for target feature points which change shape due to perspective projection, appear and disappear due to occlusion, or move against a changing textured background. Many of the algorithms may have difficulties with image-to-model feature correlation, feature persistence for tracking over more than a few frames, foreground-background segmentation, and detection and correction of correlation and tracking errors.

*Model-based* techniques address these issues by taking advantage of *a priori* knowledge of the object whose pose and motion is to be estimated. This knowledge typically con-

sists of a CAD description of the object incorporating known structure, shape, textures, transmittance/reflectance, or other visual attributes. 2D or 3D appearance models corresponding to one or more objects may be used to estimate the model pose from image measurements.

Techniques used in model-based pose estimation include feature-based model tracking [6], model-to-image registration using various point-set registration algorithms [1], [7], [17], simultaneous pose and correspondence determination [4], template matching [9], contour tracking [8], and articulated object tracking [11].

### *Model-Based Pose Estimation & Tracking*

Model-based pose estimation and tracking can be decomposed into a three-stage process. The first stage is initial *relative pose estimation*. This stage is used to *bootstrap*, or initialize, the pose refinement and tracking stages. Given some *a priori* model of the object (e.g. fiducial point locations, a surface structure representation, or a viewpoint-dependent appearance model), the system computes a candidate initial relative pose for the object to be tracked.

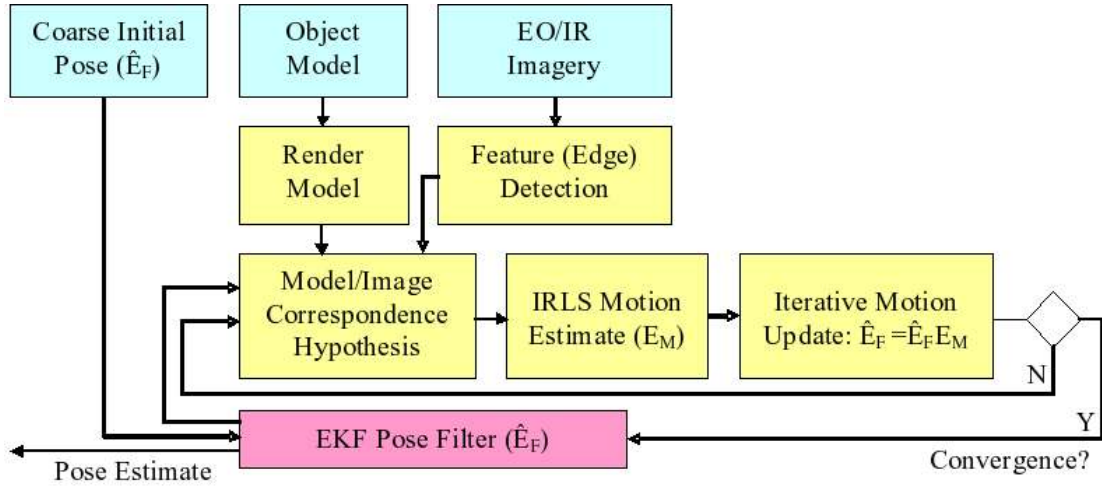
This step is performed principally when the tracker requires initialization, either at the start of tracking, or whenever loss of tracking lock occurs. The initial pose estimate may be obtained using such techniques as Active Appearance Models [14], [12] and robust PCA/Eigentracking [2], [10], [16] or simultaneous pose and correspondence determination (soft-POSIT) [4].

The second stage is *pose refinement*. Given a coarse initial pose estimate, perhaps obtained as the predicted pose from a pose tracking stage, the model-based pose refinement algorithm implementation discussed in this paper renders a view of a model (based on a CAD model or predetermined set of features) and uses an iterative reweighted least squares approach to match the rendered model with image measurements along a set of lines normal to the rendered model edges.

A solution to the iterative reweighted least squares problem is an estimate of the best fit Euclidean transformation, which provides an estimate of the translation and rotation of the model relative to the camera frame. The implementation is based on Drummond and Cipolla’s work on real-time visual pose estimation for complex structures with known models [5].

The third stage, *pose tracking*, utilizes an extended Kalman filter to track the relative pose estimate over time using a known motion model, providing robust estimates and predictions of pose and pose error covariance.

This paper focuses primarily on an implementation of the model-based pose refinement algorithm. Benefits of the implemented model-based pose refinement approach include the following:

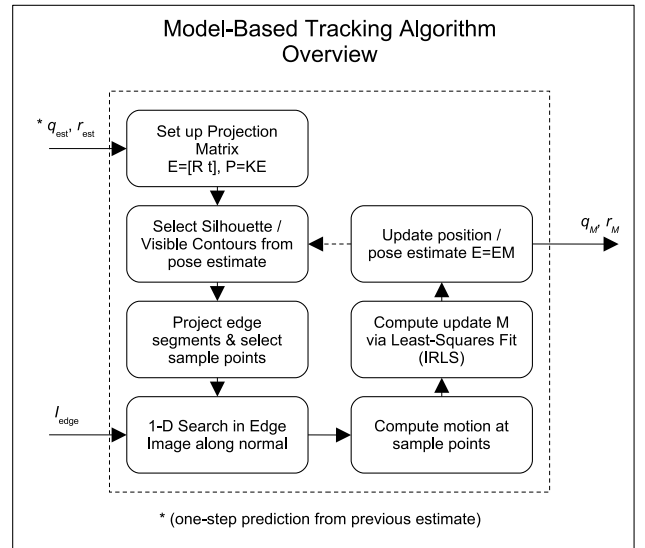


**Figure 2.** Pose Refinement and Tracking Architecture. Blue=Initial Conditions, Yellow=Pose Refinement, Red=Pose Tracking

- The model-based pose estimation algorithm works with any number of cameras and does not require active sensors. Use is not restricted to parallax stereo camera configurations, as long as the relative camera orientations are known.
- The algorithm is designed to handle perspective distortion and self-occlusion of the 3D model during its projective *view rendering* step. Visibility algorithms such as backface culling and hidden line/surface removal are incorporated into the algorithm.
- The algorithm provides additional “features” (modeled edges and connectivity information, etc.) to improve a model match. This results in a global correspondence which reduces the effect of measurement outliers due to the aperture problem inherent in local feature matches.
- Fiducial markings can be naturally integrated into the 3D surface model representation. Note that given sufficient visibility and object shape complexity, the model-based pose estimation/refinement and tracking approach works with “non-cooperative” objects which do not have specific fiducial markings; thus there is no need to retrofit existing objects with separate fiducial markings.
- The model fit provides an error estimation for pose validation. A model fit error above a given threshold can signal the tracker to begin re-initialization.
- Fusion of pose estimates from multiple viewpoints (e.g. wingtip, pylon, and tail cameras on an aircraft) may be accommodated either natively in the pose estimation algorithm or via sensor fusion in the tracking filter. Fusion of the pose estimates requires knowledge of inter-camera alignment.
- The tracking filter may include motion models for object dynamics to improve the model correspondence and tracking performance. The motion model can thus take advantage of *a priori* knowledge of relative or absolute orbital dynamics, platform and target (aero)dynamics, or may use a simplified constant-velocity motion model.

### 3. MODEL-BASED POSE REFINEMENT

The implemented model-based pose refinement algorithm utilizes an Iterative Reweighted Least-Squares (IRLS) formulation to minimize the correspondence error between projected model contours (silhouette or potentially visible interior edges) and the edges detected in one or more camera images [5]. The fit provides relative pose pseudo-“measurements” for use in the tracking filter. A simplified version of the approach, assuming a single camera, is illustrated in Figure 3.



**Figure 3.** Model-Based Pose Refinement Algorithm

The algorithm takes as inputs at each invocation the current predicted estimate of relative pose (in the figure, the orientation quaternion  $q_{est}$  and the relative position  $r_{est}$ ) and one or more processed camera images  $I_{edge}$ . Potentially visible silhouette and internal edges are selected according to the estimated pose and projected onto the image plane.

Sample points are selected at regular intervals along the projected model edges and a one-dimensional search is performed orthogonal to the projected model edge to locate the nearest “edge” (gradient local maximum) in the camera image.

An IRLS optimization approach is used to find the best fit of the model pose to the camera image (i.e. to minimize the distances to detected edges at each sample point). The fit results in an incremental “motion” approximated as a Euclidean transformation consisting of a rotation matrix  $R$  and a translation vector  $t$ .

The algorithm typically converges in only a few IRLS iterations, depending on initial pose estimate accuracy. The resulting pose parameters (expressed in Figure 3 as the attitude quaternion  $q_M$  and relative position vector  $r_M$ ) are extracted from the Euclidean transformation matrix  $E$  after the IRLS iterations are completed.

Mathematically, the approach is as follows. A *perspective camera* can be modeled as a projective transform  $M$  of a scene point  $P = (x, y, z, 1)$  to an image point  $p = (u, v, 1)$ . Homogeneous coordinates are used here to facilitate calculations in the projective space (note the augmentation of the three-dimensional vector  $P$  and two-dimensional vector  $p$  by a 1).

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \frac{1}{Z} M \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (1)$$

The intermediate representation  $(X, Y, Z)^T = Z(u, v, 1)^T$  represents an equivalence relation wherein 3D scene points  $P_i = (x_i, y_i, z_i, 1)^T$  are *equivalent* (they map to the same point  $p = (u, v, 1)^T$  in 2D space) if application of the projective transform  $M$  results in vectors differing only by a scalar factor  $c_i$ , e.g.  $MP_i = (c_i u, c_i v, c_i)^T = c_i(u, v, 1)^T$ .

$M$  can be factored into the product of the intrinsic calibration matrix  $K$  (an affine transformation that represents the skew  $s$ , focal lengths and aspect ratio  $f_u/f_v$ , and center of projection  $(u_0, v_0)$  of the camera) and an extrinsic matrix  $E = (R \ t)$ , which represents the orthonormal rotation matrix  $R$  and translation vector  $t$  of the camera relative to the world (or alternatively an object’s) reference frame.

$$M = KE = \begin{bmatrix} f_u & s & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t_x \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Rigid motion of the camera relative to the target structure can be represented by an additional Euclidean transformation  $E_M$  such that the updated projective transform after the motion becomes

$$p' = \left( \frac{1}{Z'} M' \right) P = \left( \frac{1}{Z'} M E_M \right) P. \quad (3)$$

The goal of model based pose estimation is to measure the Euclidean transformation  $E_M$ , using only image measurements and prior knowledge of model points  $P$  and the projective transform  $M$ .

The Euclidean transform  $E_M$  is obtained from the exponential map

$$E_M = \exp\left(\sum_i \alpha_i G_i\right) \quad (4)$$

where  $G$  forms an orthonormal basis for a vector space of derivatives of 6DOF rigid body motions, and  $\alpha_i$  are the six coefficients in this basis.  $E_M$  can be linearized under the assumption that the interframe motion is small. Using a Taylor series expansion truncated at the linear term:

$$E_M \approx I + \sum_i \alpha_i G_i \quad (5)$$

where  $I$  is the identity matrix. This simplifies the motion estimation problem to finding the coefficients  $\alpha_i$  that best describe the motion.

A motion estimate first requires a correspondence hypothesis between the model and the image. Define a set of sample points  $\Xi$  such that each  $\xi \in \Xi$  corresponds to a sample point  $P^\xi$  on the model, relative to the model reference frame. A projection  $p^\xi$  of  $P^\xi$  can be found using (3), where the projection of all non self-occluding  $\xi \in \Xi$  is a *rendering* of the model for a given initial model pose estimate  $E_M$  and calibrated projection matrix  $M$ .

Selection of these sample points is determined by visibility checking algorithms, which can use backface culling, hardware-accelerated Z-buffers, and other constructs such as fast binary space partition (BSP) trees to dynamically determine visible model features for rendering and processing in real-time.

For each  $p^\xi$ , define the vector  $\hat{n}^\xi$  normal to the projected model edge. Search along the line spanned by  $\hat{n}^\xi$  in the intensity image  $I$  for the nearest brightness edge with intensity gradient above a given threshold, as determined by application of the Sobel edge operator  $S(I)$ :

$$S(I) = \sum_{x,y} \sum_{j,k} K(j,k) I(x-j, y-k) \quad (6)$$

using kernels

$$K_h = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (7)$$

$$K_v = \begin{bmatrix} -1 & 2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (8)$$

such that the magnitude of the edge ( $|S|$ ) is

$$|S| = \sqrt{S_h^2(I) + S_v^2(I)} \quad (9)$$

where  $S_h$  corresponds to the Sobel edge operator with kernel  $K_h$ . The distance  $d^\xi$  is the Euclidean distance between the model point  $\xi$  and the nearest edge along the normal line  $\hat{n}^\xi$ . This distance defines a correspondence between the model and the image.

Using the estimates  $d^\xi$  from the hypothesized model correspondence, it can be shown that the least squares solution to the coefficients  $\alpha$  is

$$v_i = \sum_{\xi} s(d^\xi) d^\xi (L_i^\xi \bullet \hat{n}^\xi) \quad (10)$$

$$C_{ij} = \sum_{\xi} s(d^\xi) (L_i^\xi \bullet \hat{n}^\xi) (L_j^\xi \bullet \hat{n}^\xi) \quad (11)$$

such that

$$\alpha_i = C_{ij}^{-1} v_j \quad (12)$$

and  $L_i^\xi$  is the linearized projected derivative of translation or rotation in the  $i^{\text{th}}$  direction. The weighting function

$$s(d^\xi) = \frac{1}{c + |d^\xi|} \quad (13)$$

provides a weight that depends on  $d$  and changes with each iteration.  $s(d^\xi)$  may be considered a “saliency” measure for each sample point  $\xi$ , and may be further extended according to predicted model edge strength or other heuristics. The constant  $c$  is typically chosen to be approximately one standard deviation of the inlying data.

The motion estimate  $E_M$  is generated from the motion coefficients  $\alpha_i$  using (4). To improve the robustness of the motion estimate to outliers, the least squares solution is iterated. On each iteration, the motion estimate  $E_M$  is reapplied to the camera projection model (3), which generates a reweighting from (13) and a new reweighted least squares solution in (12).

The new solution in (12) is iterated until either a fixed number of iterations or some other heuristic threshold is reached (for example, when the matrix norm of motion difference between two subsequent iterations is below a given threshold).

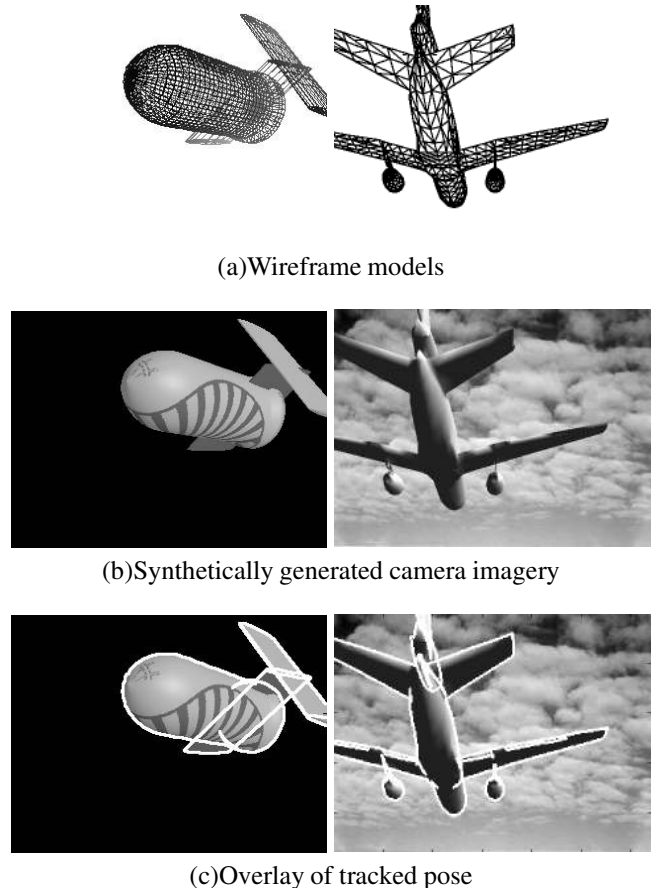
Note that the pose refinement algorithm requires an initial estimate of relative pose. A one-shot pose estimation algorithm can be used to “bootstrap” the algorithm with a coarse initial pose estimate, from which the pose refinement algorithm can proceed.

Once the pose is initialized and tracking commences, the output of a pose tracking filter can provide the predicted relative pose as the initial alignment for the next invocation of the model-based pose refinement algorithm.

#### Example Scenarios

The model-based pose refinement algorithm has been applied to a number of different scenarios and camera image sequences. Figure 4 shows examples of this approach for several application scenarios. On the left is a simple “aerospace”

object whose relative pose is being tracked by the algorithm. On the right, a model of a DC-10 tanker is being tracked in an Autonomous Aerial Refueling (AAR) scenario.



**Figure 4.** Model-Based Pose Estimation for Example Aerospace Object (left) and Autonomous Air Refueling Scenario (right)

Figure 4(a) shows the wireframe models for the tracked objects. In these examples, the models are used for both rendering synthetic camera images and for the model-based pose refinement algorithm. The sample objects are constructed using texture-mapped quad-mesh or triangle-mesh representations, and exhibit different curvatures and textures.

Figure 4(b) shows synthetically-generated camera images for the two objects. The tanker (right) is shown from a potential refueling viewpoint. Note the textured surface of the aerospace object (left) and the textured cloud background in the AAR scenario. These textures result in detection of “false” edges during processing, stressing the pose refinement algorithm and potentially leading to outlier measurements.

The camera images and predicted model pose are passed to the pose refinement algorithm, which generates updated pose estimates. The projected model edges for the refined and registered pose are shown in Figure 4(c) as white overlays on the original camera intensity images. The examples shown exhibit good correspondence after several iterations, given an

initial pose error of 10–15 degrees.

#### 4. TEST ENVIRONMENT

In this section we briefly describe the hardware and software testing facilities and data sources used to test and validate the model-based pose refinement and tracking algorithms.

The VISARD Analysis Testbed consists of a software toolbox and a Hardware-In-the-Loop (HIL) testbed. The toolbox includes algorithm and utility function categories listed in Table 1.

**Table 1.** VISARD Software Toolbox Functions

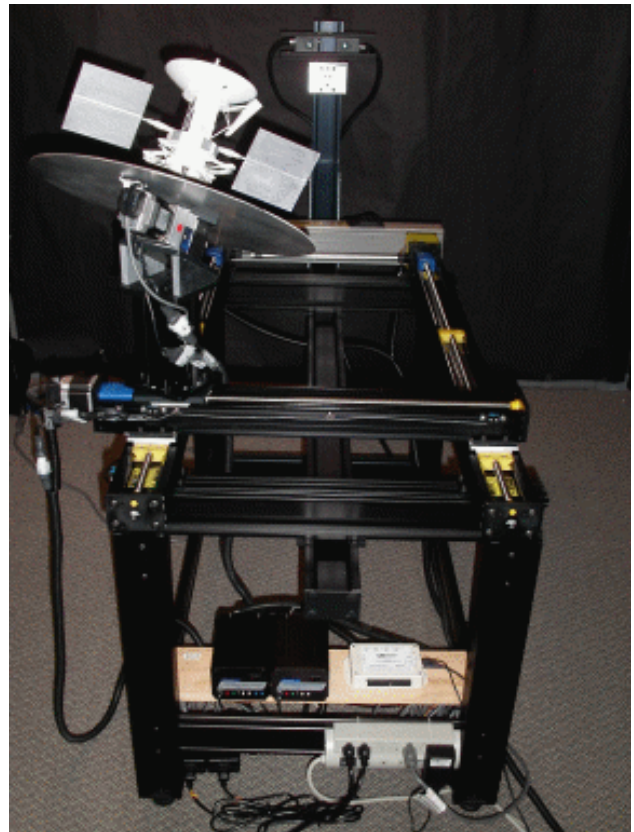
VISARD Software Toolbox Functions
View Synthesis Algorithms
Image Preprocessing Algorithms
Model-Based Pose Refinement / Estimation Algorithms
Pose Tracking Filter (Extended Kalman Filter)
Feature-Based Stereo Tracking Algorithms
Additional Algorithms and Utilities:
Image preprocessing for tracking algorithms
3D object model import/export/manipulation
Algorithm testing routines
VISARD HIL Testbed interface
Simulink blockset
etc.

A key aspect of the VISARD Analysis Testbed is the ability to synthesize or capture real camera image sequences along with synchronized ground truth information. For synthesized camera imagery, ground truth information is directly available from the simulated pose states. The HIL Testbed provides stereo camera imagery and ground truth pose information for small models using a motion-controlled gantry.

##### *HIL Testbed*

The VISARD Hardware-In-the-Loop (HIL) Testbed includes a stereo camera pair which feeds a live stereo video stream to a PC-based frame grabber/vision processor. The hardware operates under controlled lighting conditions to emulate an orbital/space visibility environment for basic testing of the algorithms using scaled-down object mockups.

The VISARD HIL testbed features a 3DOF motion-controlled gantry, stereo image capture, encoder feedback for ground truth, a controlled lighting environment, and a control and data acquisition interface to the MATLAB simulation and analysis environment. See Figure 5 for a picture of the HIL testbed setup with a custom-built Magellan 1/24 scale model mounted.



**Figure 5.** VISARD HIL Testbed

#### 5. ALGORITHM TESTS AND EVALUATION

This section outlines a number of tests and results obtained using the implemented VISARD pose refinement and tracking algorithms. The algorithms were first tested using synthesized imagery and later applied to real video sequences captured using the VISARD HIL Testbed. Ground-truth relative pose information was captured along with the stereo camera imagery.

The goal of this testing was to characterize the performance of the pose refinement and tracking algorithms with respect to a number of parameters:

- Errors in Initial Pose Estimate
- Model-Based Pose Refinement and Tracking Algorithm Parameters
- Sensor Characteristics
- Characteristics of Tracked Object
- Various Environmental Factors

##### *Ground Truth Motion Profiles*

Several “ground truth” motion profiles were designed to provide a range of motions realizable using the testbed, and to ensure repeatability and uniformity between synthetically-generated and testbed-acquired imagery. A brief description of several of the profiles is included in Table 2.



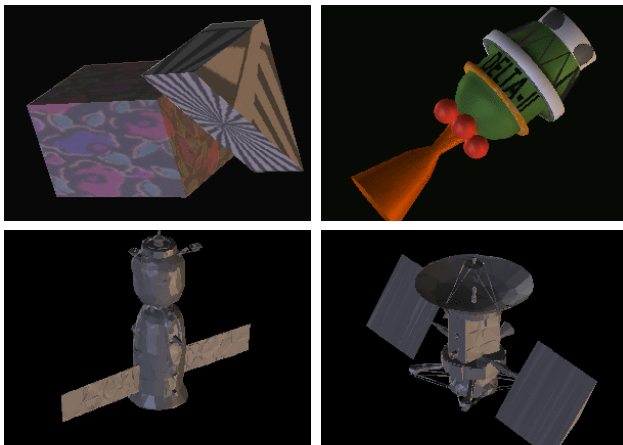
**Table 2.** Motion Profiles Used For Algorithm Testing

Profile ID	Description
1	Sequence of constant-velocity motions
2	Concatenation of profiles 3, 4, and 5
3	Straight-line departure down centerline, constant-velocity rotation
4	Lissajous-type translational profile, constant-velocity rotation
5	Spiral (sinusoidal) approach profile, reversed direction rotation
6	Another Lissajous translational profile, constant-velocity rotation

*Overview of Algorithm Testing*

During algorithm testing, real and synthesized stereo camera image sequences were generated for various scenarios requiring relative pose estimation and tracking. Test scenarios included space-based rendezvous and proximity operations, aircraft mid-air refueling, and ground-based visual servoing.

Figures 6–8 show several examples of the types of objects used for algorithm testing. These objects include various objects of different shapes and complexities, a 1/20 scale Delta II second-stage rocket body model, a 1/72 scale Soyuz satellite model, and a Magellan 1/24 scale model. The tracked model pose for one frame of selected test camera image sequences is shown in Figures 7–8 as a green overlay on the original camera image.

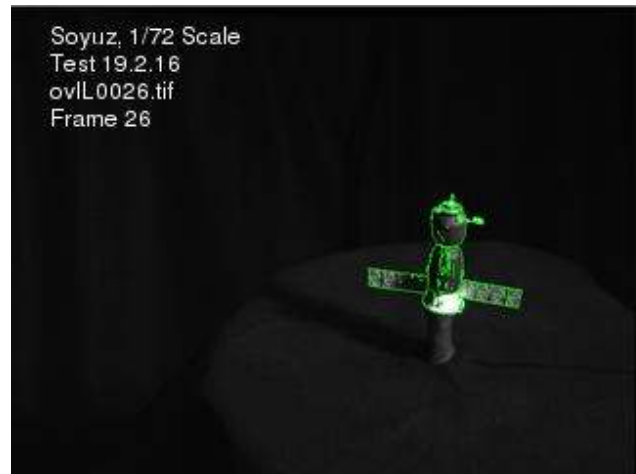


**Figure 6.** A few of the object models used for testing. Clockwise from top left: (a) simple block-type object (sscMbox), (b) Delta II second-stage rocket body, (c) 1/24 scale Magellan spacecraft, and (d) 1/72 scale Soyuz spacecraft models.

Real camera image sequences were obtained from the HIL testbed for the Soyuz and Magellan models. Synthesized camera imagery was used for the Delta II model and for a number of other models of varying types, shapes, and complexity. Ground truth information was obtained from the view synthesis module or using the hardware-in-the-loop 3DOF



(a) Tracked Delta II 2nd Stage Rocket Body Model



(b) Tracked Soyuz 1/72 Scale Satellite Model

**Figure 7.** Additional Examples of Model-Based Pose Refinement & Tracking. Green overlay of tracked model pose shows good correspondence.

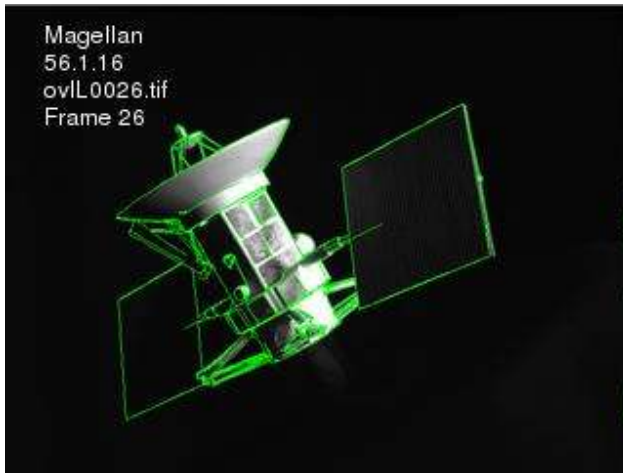
gantry testbed.

*Effect of Initial Pose Errors on Algorithm Performance*

Table 3 summarizes some of the test parameters used to analyze the effect of initial pose errors on pose refinement algorithm performance.

**Table 3.** Test Information

<b>Test Setup:</b> Effect of Initial Pose Errors	
Algorithm:	Model-Based Tracker (MBT) Pose Refinement Algorithm
Image Sequence(s):	202 (synthetic, 640x480)
Motion profile:	2
Model:	sscMbox
Test Case(s):	3,4,5
Tests IDs shown:	202.100.[3,4,5]



**Figure 8.** Model-Based Tracking of Magellan 1/24 Scale Spacecraft using real stereo camera imagery obtained using SSCI’s Hardware-in-the-Loop Testbed. Green overlay of tracked model pose shows good correspondence.

*Analysis and Discussion*—For Test Cases 3, 4, and 5, errors were injected into initial pose estimates to characterize the performance of the MBT pose refinement algorithm.

Relatively small initial attitude or position errors at close range may translate to large apparent movements in the image plane, reducing pose estimation performance. For this reason, absolute error distributions for initial position estimates were specified as percentages of object range.

For Test Case 3 (initial attitude error), the initial pose estimate was generated by rotating the true attitude about a random axis. The amount of rotation was randomly selected from a uniform distribution (0 to 30 degrees).

Figure 9 shows the effect of initial attitude errors on the refined pose attitude for synthetic Test Sequence #202. Note that the algorithm is able to handle large initial attitude errors of up to 23 degrees for this model before the mean measurement error begins to exceed 5 degrees.

For Test case 4 (transverse position error), the true pose was perturbed by a random position error applied in the (X,Y) plane (orthogonal to the cameras’ line of sight). For this case, errors in X and Y were independently selected from a uniform distribution over  $[-1,1] \times 3\%$  of range in each of the two axes.

Figure 10 demonstrates the effect of transverse errors on the refined pose attitude for this sequence. This shows that for the given object at this scale and range (between 0.58 and 1.53 m in this sequence), the mean error in measured attitude does not exceed 5 degrees until the transverse position error exceeds 3.5% of the object range.

For Test case 5 (range error), the true pose was perturbed by a random position error applied along the sensor Z axis (range),

parallel to the cameras’ line of sight. For this case, range errors were selected from a uniform distribution over  $[-1,1] \times 10\%$  of range.

Figure 11 shows similar results for initial range errors as for transverse errors; however, in this case, the algorithm easily handles injected range errors up to and exceeding 10% of range. Further testing would need to be performed to determine a “cutoff” range of operation for this setup. □

*Effect of Number of IRLS Iterations on Algorithm Performance*

Table 4 summarizes some of the test parameters used to analyze the effect of the number of IRLS iterations on pose refinement algorithm performance.

**Table 4.** Test Information

<b>Test Setup:</b> Effect of IRLS Iterations	
Algorithm:	Model-Based Tracker (MBT) Pose Refinement Algorithm
Image Sequence(s):	202 (synthetic, 640x480)
Motion profile:	2
Model:	sscMbox
Test Case(s):	3, 6–8
Tests IDs shown:	202.100.[3,6–8]

*Analysis and Discussion*— The number of IRLS iterations used by the Model-Based Pose Refinement algorithm directly affects the accuracy of the refined pose and the CPU processing requirements.

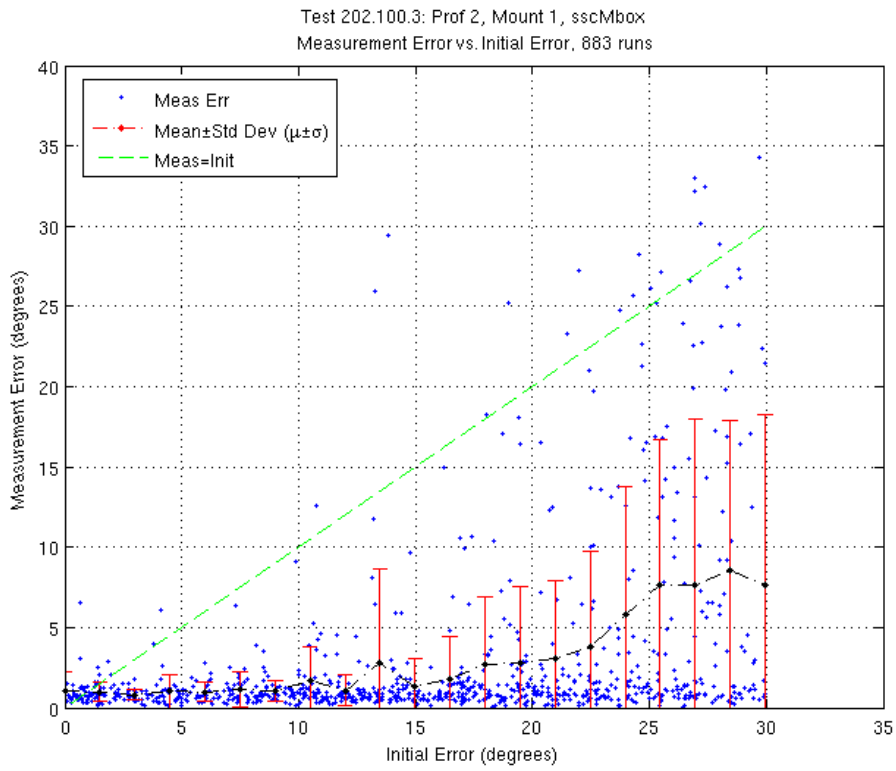
Table 5 shows the specified iteration limits and the average number of iterations per frame of the test sequence for the test cases shown. Note that for Test Case 7, the average number of iterations was only half the specified maximum; the heuristic allows the tracker to bypass extra iterations if the measurement has already “converged.”

**Table 5.** Effect of Adjusting IRLS Iteration Limits

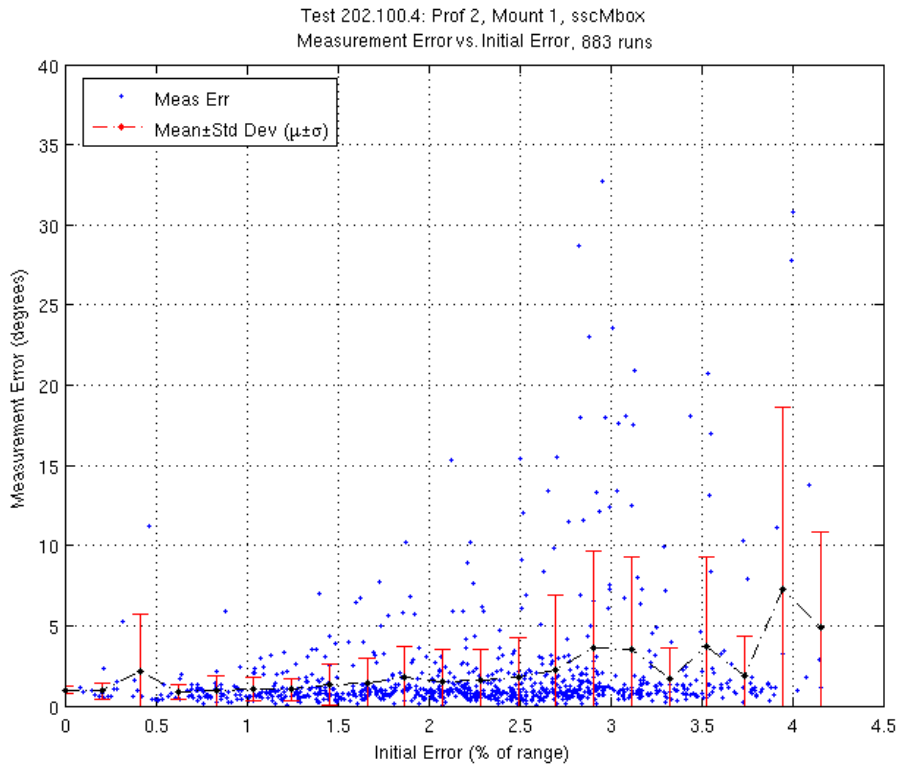
Test ID	Point Sep.	IRLS # Iter	Mean # Iter	Init Err, Meas Err $\approx 5^\circ$
6	3	2	2	13
8	3	8	8	21-23
3	4	$\leq 10$	8.29	23
7	3	$\leq 25$	13.26	27

When the initial error is small, the process quickly converges within one or two iterations (see Figure 12(a), in which the algorithm is limited to 2 iterations). Note that initial attitude errors of below about 10 degrees result in only a few degrees of measurement error for this test case. Although measurement error is reduced for larger initial attitude errors, more iterations are required to further improve measurement accuracy. On average, about 10 degrees of the initial error is removed within the first two iterations.

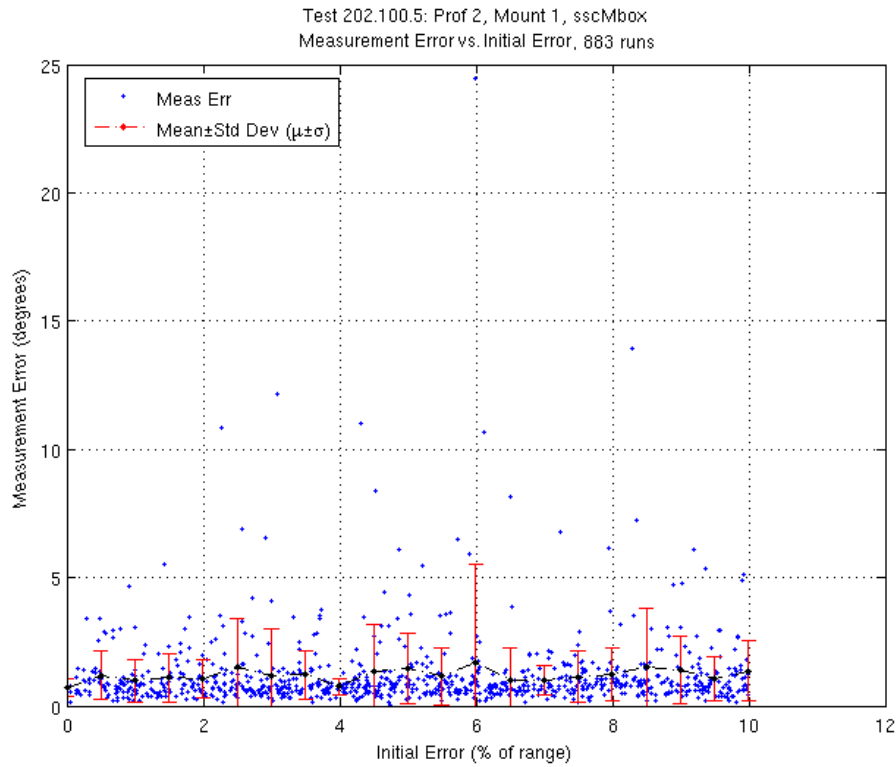




**Figure 9.** Test 202.100.3, Initial Attitude Errors



**Figure 10.** Test 202.100.4, Initial Transverse Position Errors



**Figure 11.** Test 202.100.5, Initial Range Errors.

Increasing to 8 iterations in Test Case 8 (Figure 12(b)) provides a larger reduction in error, so that the same measurement errors result given twice the initial attitude error (about 20 degrees). This comes at about a fourfold increase in processor computation, which seems to increase approximately linearly with the number of IRLS iterations performed. Test Case 7 (Figure 13(b)) shows that given after another 5 iterations on average, measurement errors are only slightly reduced, again at an additional overall expense in processing time. Test cases 3 and 7 (Figures 13(a) and 13(b) respectively) use a heuristic metric to allow early termination of the IRLS iterations when convergence is attained. □

#### *Model-Based Pose Refinement on Real Image Sequence*

Table 6 summarizes some of the test parameters used to analyze the effect of initial pose errors on pose refinement algorithm performance.

*Analysis and Discussion*—For this test, an initial attitude error was injected to characterize the performance of the model-based pose refinement algorithm. Real camera imagery of the Magellan 1/24 scale model was obtained using the HIL testbed. The range of motion for motion profile 2 was 0.63–1.58 m at 1/24 scale, corresponding to full-scale motion between 15–38 m. The injected initial attitude error, uniformly-distributed over the interval [0,30] degrees, was applied about a random axis.

**Table 6.** Test Information

<b>Test Setup: Initial Pose Error on Real Image Sequence</b>	
Algorithm:	Model-Based Tracker (MBT) Pose Refinement Algorithm
Image Sequence(s):	60 (Real image sequence, 640x480)
Motion profile:	2
Model:	Magellan 1/24 Scale Model (scanned, 5000 polygons)
Test Case(s):	3
Tests IDs shown:	60.3.3

Measurement errors for the full range of attitude initialization errors (uniformly sampled from 0 to 30 degrees) exhibited the following characteristics:

Attitude:  $\mu=6.5$  deg,  $\sigma=8.4$  deg (max 35.7 deg).

Position:  $\mu=1.6$  cm,  $\sigma=2.6$  cm (max 20.9 cm).

Over the reduced interval of 0-15 degrees initial error, the pose refinement errors showed a significant improvement:

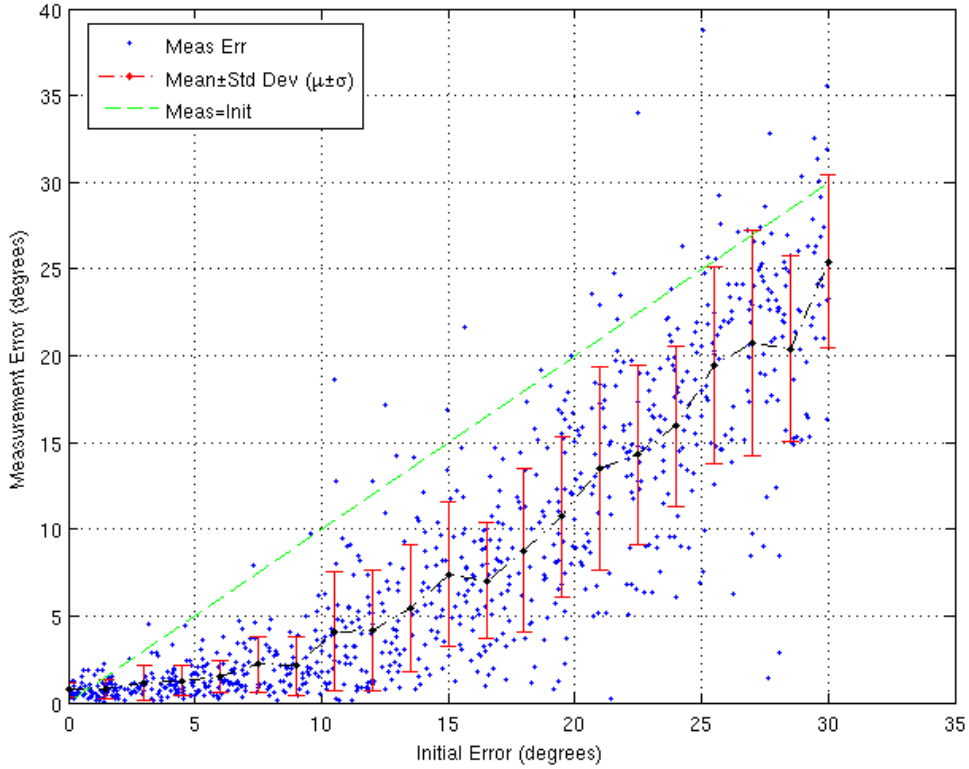
Attitude:  $\mu=1.4$  deg,  $\sigma=1.7$  deg (max 15.0 deg).

Position:  $\mu=4.7$  mm,  $\sigma=4.6$  mm (max 3.9 cm).

The average number of IRLS iterations per sample for this test was 9.7.

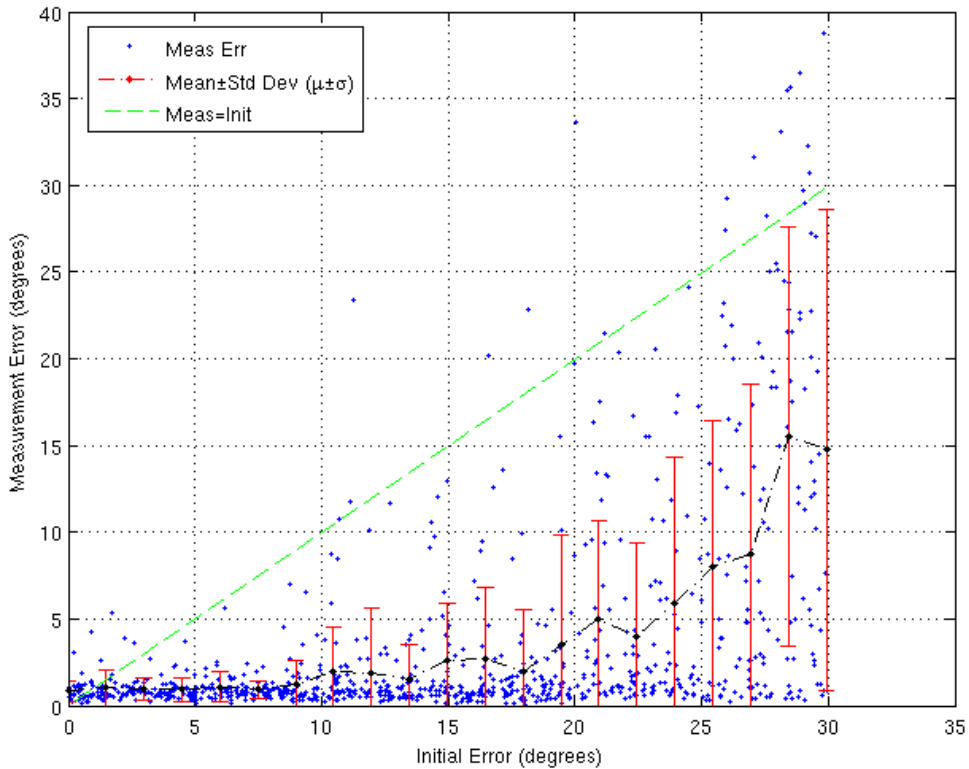
Figure 14 shows the attitude measurement error with respect to initial attitude error for Test 60.3.3. Note the “knee” of the

Test 202.100.6: Prof 2, Mount 1, sscMbox  
Measurement Error vs. Initial Error, 883 runs



(a) Test 202.100.6, 2 IRLS Iterations

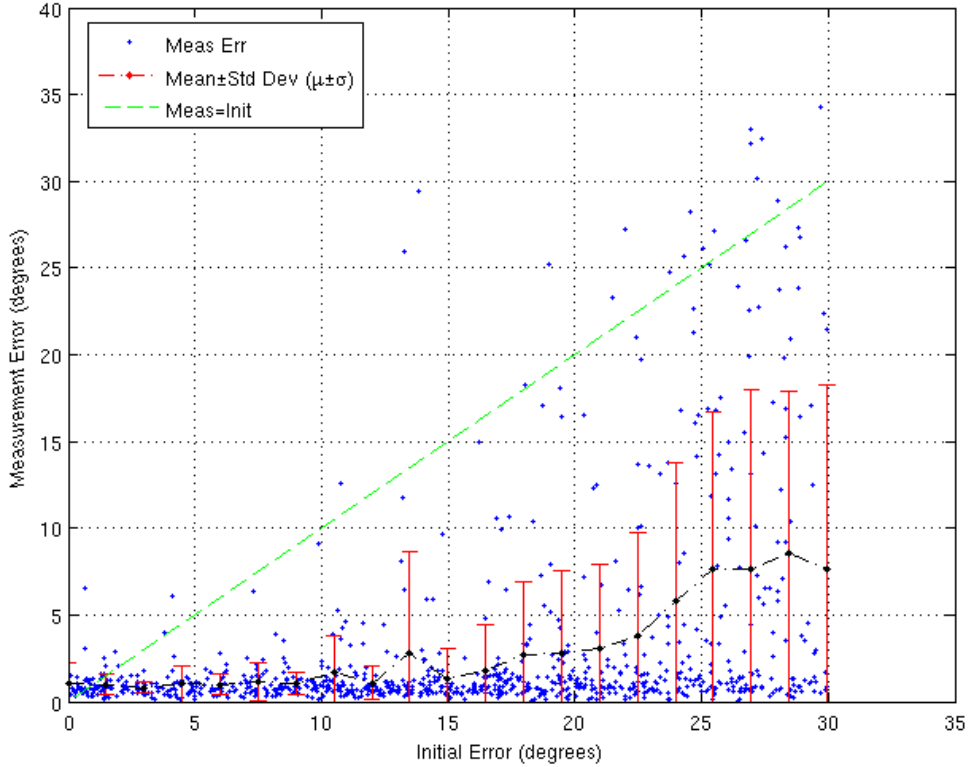
Test 202.100.8: Prof 2, Mount 1, sscMbox  
Measurement Error vs. Initial Error, 883 runs



(b) Test 202.100.8, 8 IRLS Iterations

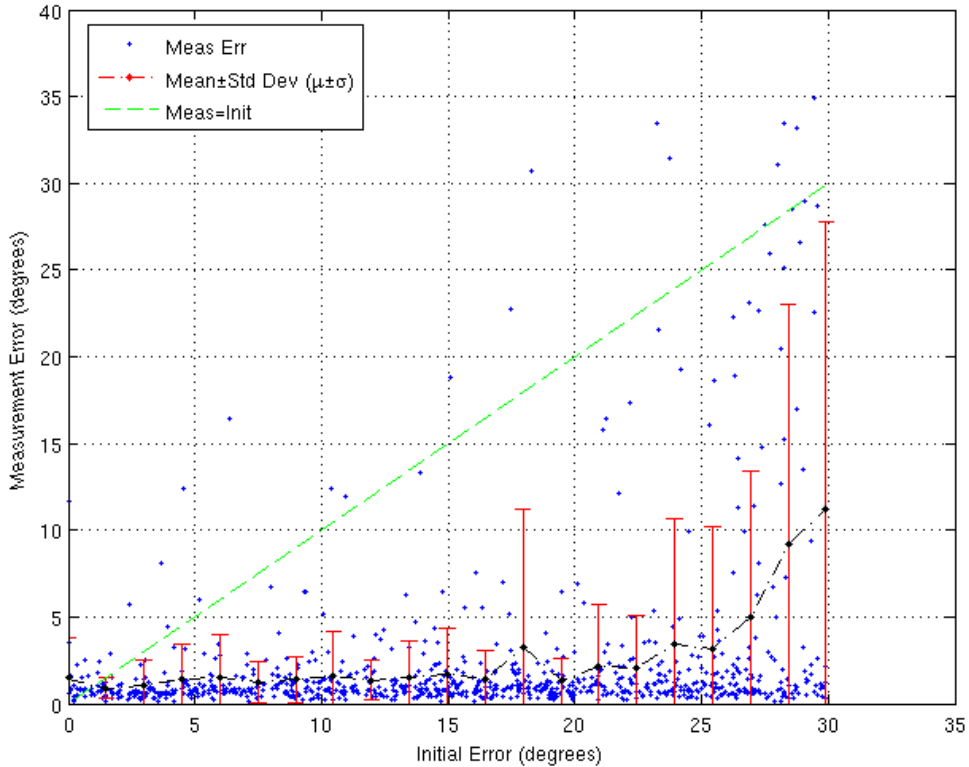
**Figure 12.** Effect of IRLS Iteration Limits on Algorithm Performance

Test 202.100.3: Prof 2, Mount 1, sscMbox  
Measurement Error vs. Initial Error, 883 runs



(a) Test 202.100.3, Max 10 IRLS Iterations (Averaged 8.3)

Test 202.100.7: Prof 2, Mount 1, sscMbox  
Measurement Error vs. Initial Error, 883 runs



(b) Test 202.100.7, Max 25 IRLS Iterations (Averaged 13.3)

**Figure 13.** Effect of IRLS Iteration Limits on Algorithm Performance (continued)

curve around 15 degrees; the algorithm performs much better for initial angular errors below this level. □

### Model-Based Pose Tracking on Real Image Sequence

Table 7 summarizes some of the test parameters used to analyze the effect of initial pose errors on pose refinement algorithm performance.

**Table 7.** Test Information

<b>Test Setup:</b> Model-Based Tracking, Real Image Sequence	
Algorithm:	Model-Based Tracker (MBT) Pose Refinement Algorithm
Tracking Filter:	Simple fixed-gain recursive IIR linear tracking filter
Image Sequence(s):	27 (Real image sequence, 640x480)
Motion profile:	1
Model:	Magellan 1/24 Scale Model (scanned, 5000 polygons)
Test Case(s):	16
Tests IDs shown:	27.3.16

*Analysis and Discussion*—This test shows some tracking results for real camera imagery obtained using the VISARD HIL testbed with the Magellan 1/24 scale spacecraft model. The motion profile (Test Profile 1) consisted of a series of connected linear (constant-velocity) motion segments. The range of motion was between 0.57 m and 1.53 m at 1/24 scale, corresponding to a full-scale range between 14 m and 37 m respectively.

The maximum angular velocity exhibited by this motion profile was 4 degrees per frame. At a processing rate of 25 frames per second (fps), the effective angular rate would be 100 deg/s. The maximum translational velocity was approximately 5.8 mm/frame, or about 3.48 m/s at full scale and 25 fps.

The position measurement error for the tracked sequence exhibited the following characteristics, where  $\mu$  and  $\sigma$  are the mean and standard deviations, respectively:

Attitude:  $\mu=0.86$  deg,  $\sigma=0.45$  deg (max 3.28 deg).

Position:  $\mu=4.3$  mm,  $\sigma=2.5$  mm (max 20.9 mm).

Figure 15 summarizes the tracking performance with respect to initial attitude error, while Figure 16 and Figure 17 show the frame-by-frame tracked motion profile and measurement errors respectively. The average number of IRLS iterations per frame for this test was 4.07. □

### Overall Pose Refinement Algorithm Performance

Table 8 summarizes the effective model-based pose refinement and tracking performance given uniformly-distributed errors in the initial attitude estimate. The data analyzed and summarized in the table are taken from the HIL-generated real camera imagery sequences utilizing the Magellan 1/24 scale spacecraft model.

**Table 8.** Pose Refinement Algorithm Performance

Angular Initialization Error (deg)	Mean Angular Measurement Error (deg)	Approximate Standard Deviation (deg)
0 - 2	1.783	1.338
0 - 4	1.797	1.361
0 - 6	1.795	1.354
0 - 8	1.851	1.478
0 - 10	1.952	1.635
0 - 12	2.129	1.905
0 - 14	2.411	2.391
0 - 16	2.837	3.045
0 - 18	3.392	3.866
0 - 20	4.119	4.926
0 - 22	5.021	6.049
0 - 24	6.080	7.199
0 - 26	7.199	8.279
0 - 28	8.335	9.270
0 - 30	9.498	10.225

Note that the means and standard deviations in Table 8 result from a uniformly-distributed angular initialization error. The mean angular measurement error for all initialization errors between 0 and 10 degrees, for example, was 1.952 degrees. This information is shown graphically in Figure 18.

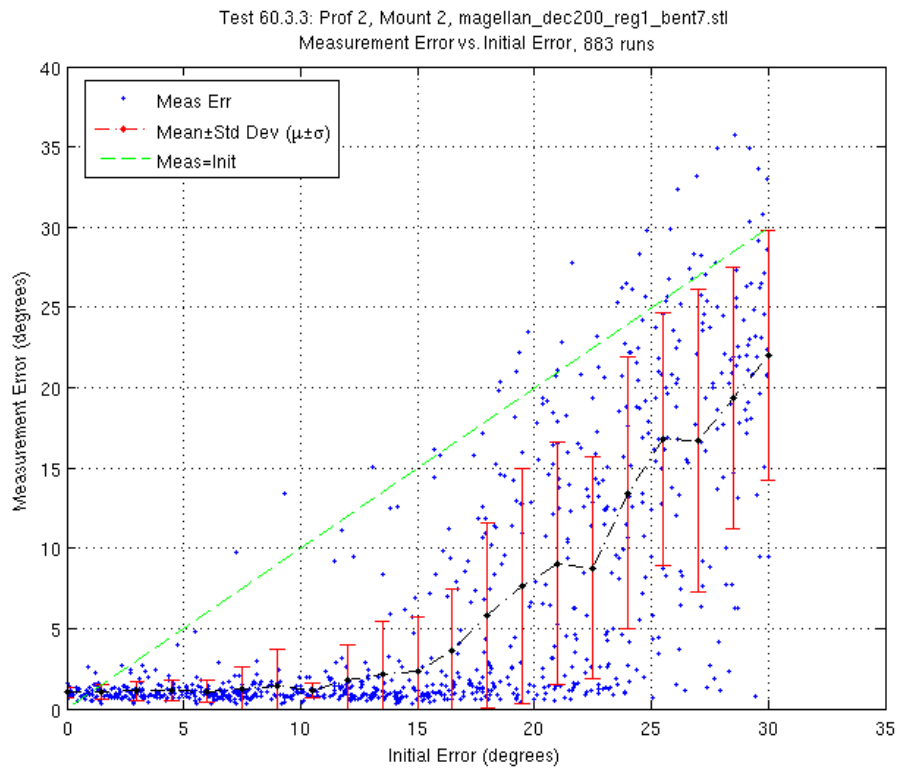
The lower bound (about 2 degrees) of the mean angular measurement error with small initialization errors is due to a combination of several factors, including systematic errors in experiment hardware (imager resolution, object mounting tolerances, etc.) and algorithm parameters (sample point density, search length, etc.) □

### Discussion and Additional Observations

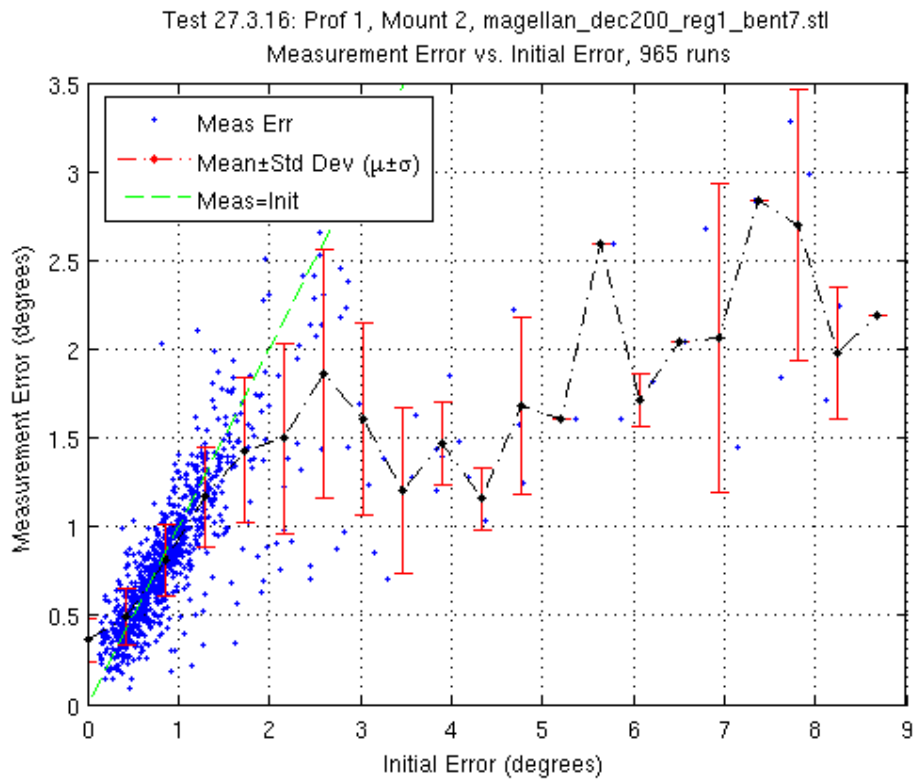
*Axially Symmetric Objects*—Items exhibiting an axially symmetric structure, such as the Delta II rocket body model (as well as the Soyuz spacecraft, when viewed from certain directions) demonstrate relatively few cues for determining the amount of rotation about their principal axes. For distant or high-noise cases, or for long sequences exhibiting large changes in angular velocity, angular estimate errors about this axis may accumulate when tracking the pose for these objects for an extended period of time. However, a low-value 2DOF “pointing error” can often be maintained even with slippage about this primary axis.

*Tests for Real Camera Sequences*—Results obtained for real camera image sequences using the hardware testbed were very similar to those obtained for synthesized imagery. The Magellan model, for instance, was successfully tracked over a wide range of lighting conditions and camera gain / exposure / brightness settings using both synthesized and real video streams. However, performance using the synthetically-generated imagery was for most model shapes noticeably bet-

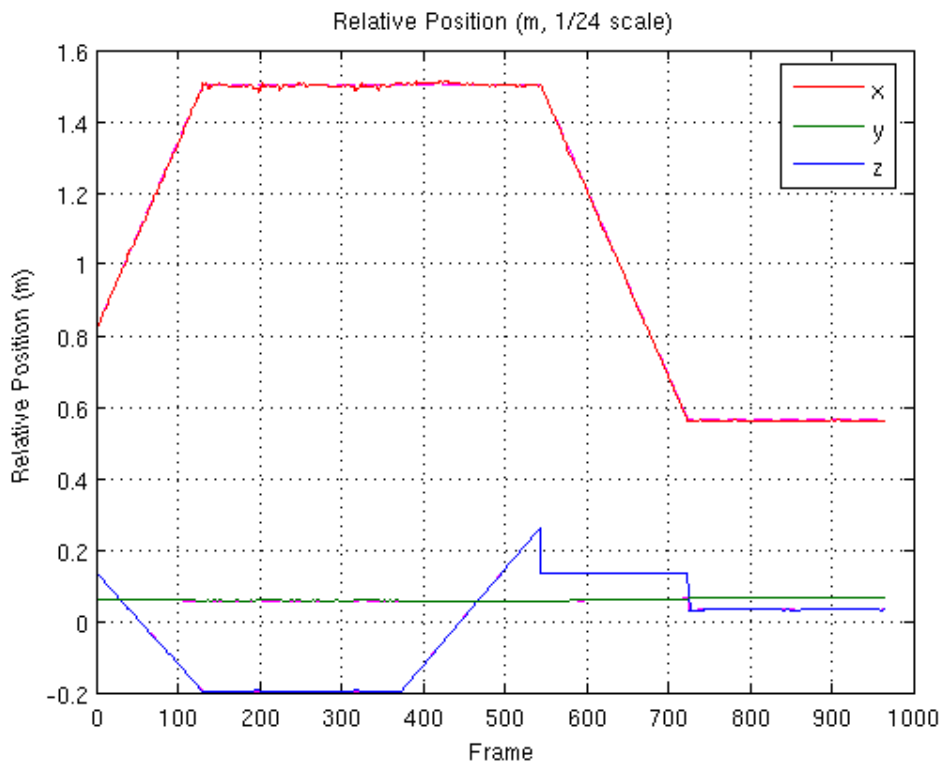
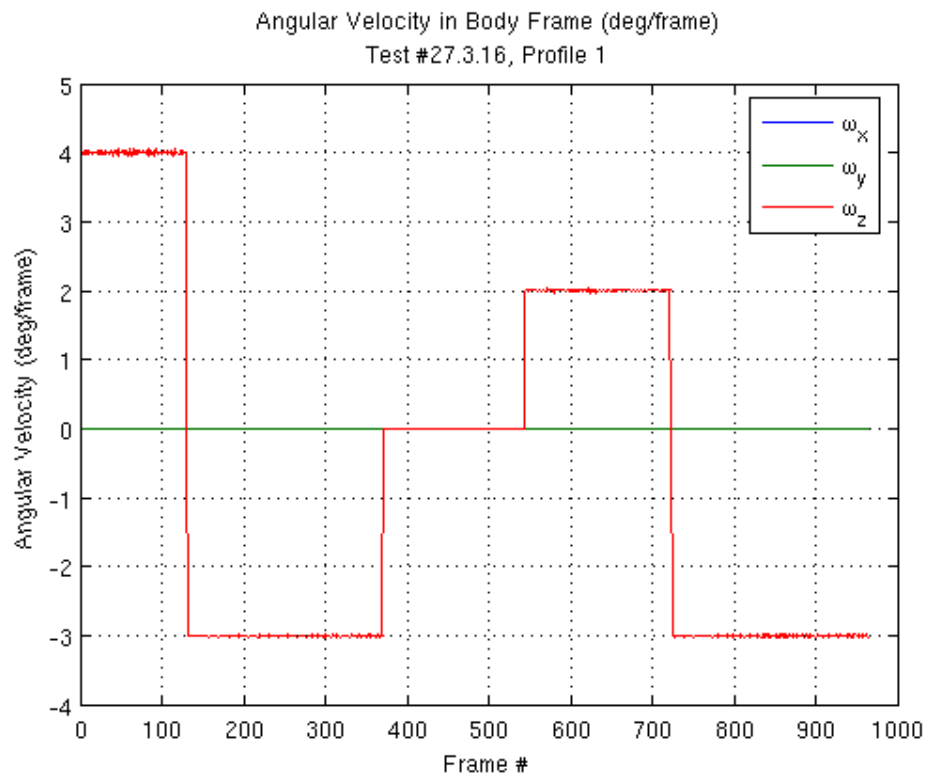




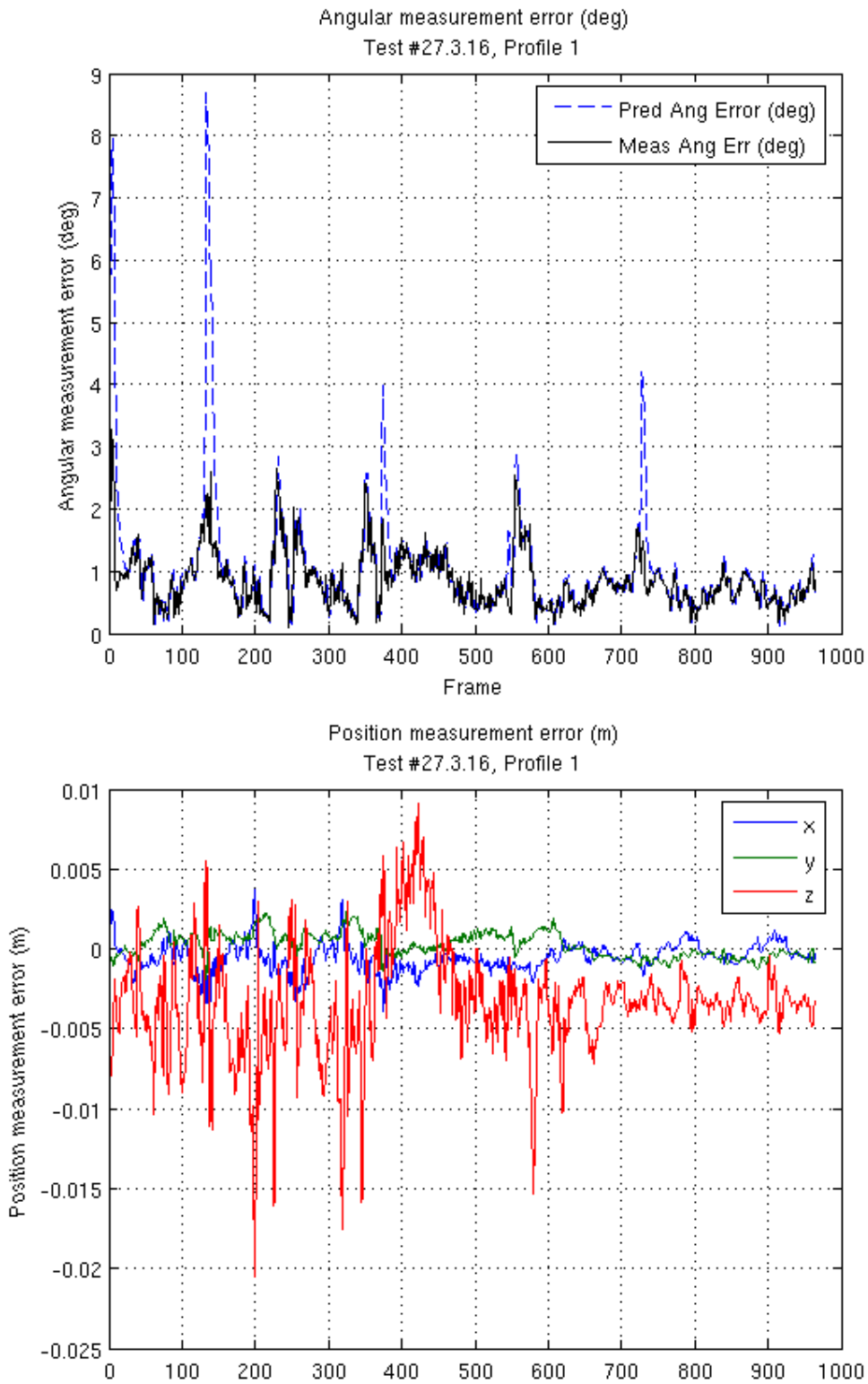
**Figure 14.** Attitude Measurement Error w.r.t. Initial Error, TestID 60.3.3



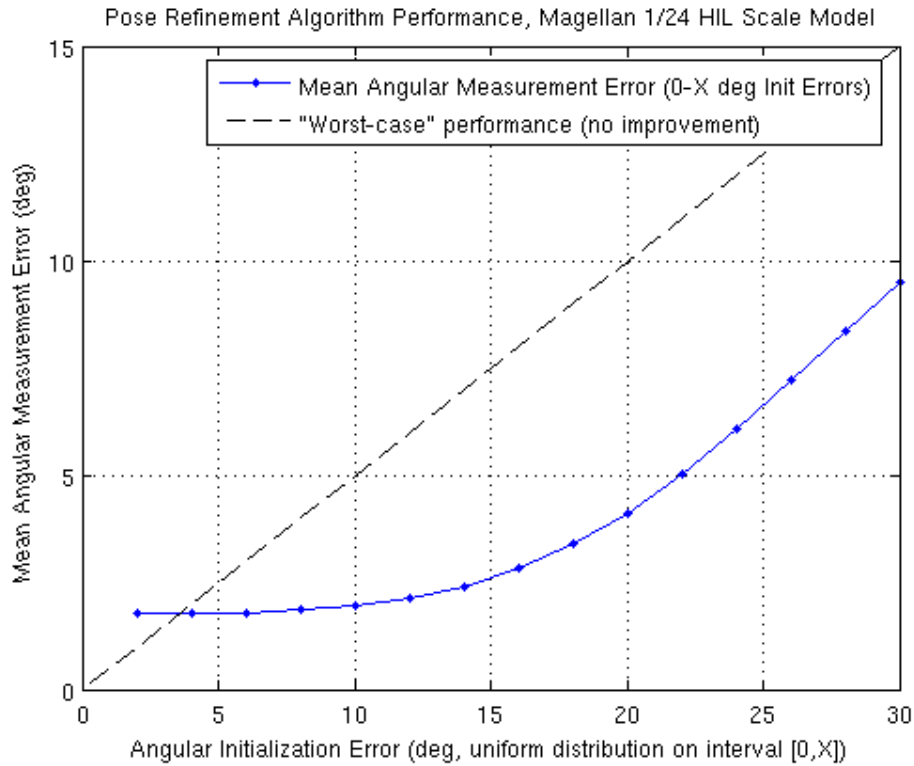
**Figure 15.** Tracking Performance, TestID 27.3.16



**Figure 16.** Tracked Motion Profile, TestID 27.3.16



**Figure 17.** Measurement/Tracking Errors, TestID 27.3.16



**Figure 18.** Model-Based Pose Refinement Algorithm Performance, Magellan Model

ter than that obtained with the real testbed imagery due to the lack of mounting imprecisions, fewer lighting and defocus/blurring artifacts, etc.

The tracking algorithms had a more difficult time with the Soyuz model. The physical model is about a third the size of the Magellan model (1/72 scale compared to 1/24 scale), and was painted a dark color which was difficult to distinguish from the black background. In addition, the solar panels were much more difficult to distinguish at times due to their highly specular reflectivity, and they would periodically “disappear” when they moved in line with the body of the model. Furthermore, there were no other significant body shape attributes that could provide a good indication of z-axis rotation. However, in a number of the test cases, the tracking filter was still able to guide the estimate through these spurious conditions.

*Model-Based Pose Refinement Algorithm Parameters*—Specification of appropriate Model-Based Pose Estimation algorithm parameters such as search length, sample point separation criteria, and maximum number of IRLS iterations is critical to the operation of the tracker. Selection of good search length and sample point separation criteria involves tradeoffs between known statistics of pose estimate initialization accuracy, estimated object size and range, and expected object edge visibility statistics.

If an initial pose estimate (either fed back from previous iterations or provided by a “bootstrap” initialization routine) is

statistically known to some “good” accuracy, a smaller search length both reduces search time and also the likelihood of outliers due to incorrect matching of sample points with distant strong edges found in the preprocessed camera images.

As the object moves further away from the cameras or the sensor image size is decreased, the likelihood of these incorrect associations increases because apparent edges draw closer to each other in the camera images. In addition, low sample point density in this case may not allow a sufficient number of associations to be made to accurately refine the pose.

Conversely, when the initial pose estimate accuracy is very coarse or the object is very close to the camera sensor, larger search lengths may be needed to ensure that the correct object edges are found. High sample point density in this case may use significant processor resources without a corresponding improvement in pose measurement accuracy.

Finite pixel sizes lead to a lower bound on useful search length and sample point separation, causing tracking failure for extremely small/distant objects or very low-resolution images even without additional camera image noise.

One possible approach to intelligently handling these tradeoffs is to continually adjust the search length and sample point density according to image size, object dimensions, initial range estimate, and filter covariance/convergence metrics.

This would improve flexibility by allowing optimal performance of the tracker over a wide range of operating conditions, and reduce the number of trial-and-error iterations for manually selecting the appropriate parameters.

Optimally accounting for initial pose estimate accuracy requires an estimate of how “good” the initial pose is. This may be obtained from filter convergence metrics or operating assumptions for the module providing the initial pose. For example, one would likely choose a larger search length to start off a tracker and reduce the search length as the tracker covariance decreases.

Optimal selection of these test parameters was not done for the tests detailed herein, but with a widely-varying class of objects to track, it would be a useful feature to improve algorithm robustness against these factors.

## 6. CONCLUSIONS

The VISARD suite of algorithms includes pose estimation and tracking algorithms, as well as tools for synthetic stereo view synthesis and a Hardware-In-the-Loop Testbed. The algorithms include both model-free and model-based approaches to vision-based RSO pose estimation and tracking. The algorithms described in this paper fall into the latter class.

Substantial testing was performed on the algorithms using both synthesized and real stereo camera imagery, using a number of different model types. For the tested scenarios, models, and testbed configurations, test results indicated that in most cases, the Model-Based Tracker pose estimation algorithm can handle initial pose errors with the following components:

- Attitude Errors: up to about 20 degrees
- Transverse Errors: up to about 2% of range (depending on model, sensor resolution, etc.)
- Range Errors : more than 10% of range

For the tests using the Magellan 1/24 scale model, the root mean square error (RMSE) of the model-based pose refinement relative position, given uniformly-distributed initial attitude errors of up to 30 degrees, was approximately [5.16 2.92 30.0] mm in the camera [X,Y,Z] coordinate frame, or 5.93 mm transverse error and 30.0 mm range error respectively. At full scale, this corresponds to an effective RMSE of 14.2 cm (transverse) and 72.0 cm (range).

The Model-Based Tracker pose refinement algorithm implemented in the VISARD toolset has a number of strengths compared to active-sensor or traditional feature-based visual tracking approaches. Benefits include:

- The ability to use any number of cameras
- Use is not restricted to a parallax stereo camera configuration

- Full 6DOF pose measurements are available as outputs of the algorithm
- The algorithm uses passive sensors and does not require retrofitting of targets with fiducial markings
- It can use low-cost, commercial off-the-shelf cameras
- The algorithm can be implemented to run in real-time

Inherent limitations of the algorithm include

- It requires an initial pose estimate
- A model of the tracked object is required
- Accuracy may be limited due to low image resolution and large distances (as with all image-based approaches)

### *Recommendations for Future Work*

The model-based tracking algorithms developed during this investigation have a wide range of possible applications, from autonomous space rendezvous to close-formation flight for mid-air refueling tasks. We feel that the set of tools in the VISARD framework is a good basis for developing vision-based autonomy applications, but there are several enhancements necessary for a fully autonomous pose estimation and tracking system.

Testing results showed that the VISARD pose tracking algorithm performance is highly dependent on the initial pose estimate accuracy. However, pose estimation initialization is still somewhat of an unanswered question. Several methods of obtaining initial pose estimates were investigated during the course of this investigation, some of which seem promising.

Finally, performance of the model-based pose refinement and tracking algorithm would greatly benefit from a method of online selection/optimization of algorithm parameters, as would integration of known fiducial markings into the pose refinement scheme. These items should be addressed as topics for further study in order to fully utilize the potential of these algorithms.

## 7. ACKNOWLEDGEMENTS

The research was supported in part by a U.S. AFRL contract # F29601-02-C-0020.

## REFERENCES

- [1] K. S. Arun, T. S. Huang, and S. D. Blostein, “Least-Squares Fitting of Two 3-D Point Sets,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 5, pp. 698-700, Sep. 1987.
- [2] M.J. Black and A.D. Jepson, “EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation”, *International Journal of Computer Vision*, vol.26 no.1, pp.63-84, January 1998.
- [3] C. H. Chien, and K. Baker, “Pose estimation for servic-



ing of orbital replacement units in a cluttered environment”, *IEEE International Conference on Robotics and Automation* April 2004, pp. 5141-5146.

- [4] P. David, D.F. DeMenthon, R. Duraiswami, and H. Samet, “Softposit: Simultaneous pose and correspondence determination”, *7th ECCV*, vol. III, pp. 698-703, Copenhagen, Denmark, May 2002.
- [5] T. Drummond and R. Cipolla, “Real-Time Visual Tracking of Complex Structures”, *PAMI*(24), No. 7, July 2002, pp. 932-946
- [6] D. B. Gennery, “Visual Tracking of Known Three-Dimensional Objects”, *IJCV*(7), No. 3, April 1992, pp. 243-270
- [7] BKP Horn, “Closed-form solution of absolute orientation using unit quaternions”, *Journal of the Optical Society of America, Series A*, vol. 4, no. 4, pp. 629-642, April 1987.
- [8] B. Isard, “Condensation: Conditional Density Propagation for Visual Tracking”, *IJCV* 29(1), 1998
- [9] F. Jurie and M. Dhome, “Real time robust template matching”, *British Machine Vision Conference*, pp. 123-131, 2002.
- [10] J. Krumm, “Eigenfeatures for Planar Pose Measurement of Partially Occluded Objects”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1996)*, pp. 55-60, 1996.
- [11] X. Lan, D.P. Huttenlocher. “A Unified Spatio-Temporal Articulated Model for Tracking”, *IEEE CVPR*, vol. 01, no. 1, pp. 722-729, 2004.
- [12] V. Lepetit, J. Pilet and P. Fua. “Point Matching as a Classification Problem for Fast and Robust Object Pose Estimation”, *CVPR*, vol. 02, no. 2, pp. 244-250, 2004.
- [13] I. Matthews and S. Baker, “Active Appearance Models Revisited”, *International Journal of Computer Vision*, Vol. 60, No. 2, November, 2004, pp. 135-164.
- [14] P. Mittraipyanuruk, G.N. DeSouza and A.C. Kak, “Calculating the 3D-Pose of Rigid Objects Using Active Appearance Models”, *Proceedings of the Int. Conference in Robotics and Automation*, 2004.
- [15] L. Morency and R. Gupta, “Robust Real-time Egomotion from Stereo Images”, *Proceedings International Conference on Image Processing*, 2003
- [16] K. Ohba and K. Ikeuchi, “Detectability, Uniqueness, and Reliability of Eigen-Windows for Robust Recognition of Partially Occluded Objects”, *IEEE Pattern Analysis and Machine Intelligence*, Vol.19, No.9, pp.1043-1048, 1997.
- [17] N. Ohta and K. Kanatani, “Optimal estimation of three-dimensional rotation and reliability evaluation”, B. Neumann and H. Burkhardt, editors, *Computer Vision - ECCV '98*, Vol. 1, pp. 175-187. Springer, June 1998.
- [18] D. Sinclair, A. Blake, and D. Murray, “Robust esti-

mation of egomotion from normal flow”, *International Journal of Computer Vision*, 13: 57–69, 1994

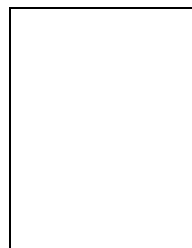
- [19] C. Tomasi and T.Kanade, “Shape and motion from image streams under orthography: a factorization method”, *International Journal of Computer Vision*, 9(2):137-154, November 1992
- [20] RealSpace Models, <http://www.realspacemodels.com>, Online; accessed August 2003.



**Jed Kelsey** is a Research Engineer at Scientific Systems Company Inc. (SSCI) in Woburn, MA. His research interests include autonomous systems, computer vision, and image-based navigation. Mr. Kelsey received an M.S. degree in Electrical Engineering from Brigham Young University.



**Jeffrey Byrne** is working in the areas of Computer Vision, Image Analysis, Data Fusion and Automatic Target Recognition. He is the principal investigator for the Visual Threat Awareness (VISTA) and Image-Aided Navigation projects, and has contributed to projects involving perceptual organization for ATR, image processing for mine detection and multi-target tracking. Prior to joining SSCI, he has worked as a consultant to Carnegie Mellon University on the Micro Vision Engine hardware platform for micro air vehicles (MAV), and as a research staff member in the CMU Robotics Institute working on image segmentation and classification, high-speed vision based inspection, real time vision for helicopter perception and stereo vision based autonomous mobile robot navigation. He also worked as a founding engineering manager in an embedded hardware start-up company in Silicon Valley. He received an M.S. in Electrical and Computer Engineering from Carnegie Mellon University.



**Martin Cosgrove** is a Research Engineer at Scientific Systems Company Inc. (SSCI) in Woburn, MA. He is working in the areas of Computer Vision, Image Analysis, Data Fusion and Automatic Target Recognition. His background and interests are robotic system design, hardware design, software design, and vision systems. Mr. Cosgrove was a Draper Fellow at C.S. Draper Laboratory working on system design, hardware design, and low-level software implementation for Land and Air autonomous vehicles. He also

*worked as a principal software engineer in several startup companies working on video game software, embedded hardware, and cable box firmware. He received an M.S. degree from Carnegie Mellon University.*



**Dr. Sanjeev Seereeram** is the Manager of Technology Commercialization at Scientific Systems Company, Inc. (Woburn, MA). He has supervised and performed advanced research and technology development in Autonomous and Intelligent Systems, Advanced Controls and Estimation, and Information Technologies, with applications in Unmanned Aerial Vehicles, Autonomous Space Systems, Robotics, Aerospace Controls, and Enterprise Decision Support systems. Dr. Seereeram holds a M.Sc. in Electrical Engineering (Brown University) and a Ph.D. in Computer & Systems Engineering (RPI).



**Dr. Raman K. Mehra** is President and Founder of Scientific Systems Company, Inc. and responsible for overall technical & financial management of advanced research and development projects. He has participated in the development of innovative applications of bifurcation analysis and catastrophe theory for nonlinear systems; global stability of aircrafts, power systems and jet engines. Dr. Mehra also developed new approaches for multivariable digital control & applied methods of State Space forecasting to business, energy, socioeconomic and hydrological time series.