Dynamic 6DOF Metrology for Evaluating a Visual Servoing System

Tommy Chang, Tsai Hong,
Michael Shneier
Intelligent Systems Division, NIST
Gaithersburg, MD 20899-8230
{hongt, tchang, shneier}@cme.nist.gov

German Holguin, Johnny Park
Robot Vision Laboratory
Purdue University
West Lafayette, IN 47907
{gholguin,jpark}@purdue.edu

Roger D. Eastman
Dept. of Computer Science
Loyola College in Maryland
Baltimore, Maryland 21210
reastman@loyola.edu

ABSTRACT

In this paper we demonstrate the use of a dynamic, six-degree-of-freedom (6DOF) laser tracker to empirically evaluate the performance of a real-time visual servoing implementation, with the objective of establishing a general method for evaluating real-time 6DOF dimensional measurements. The laser tracker provides highly accurate ground truth reference measurements of position and orientation of an object under motion, and can be used as an objective standard for calibration and evaluation of visual servoing and robot control algorithms. The real-time visual servoing implementation used in this study was developed at the Purdue Robot Vision Lab with a subsumptive, hierarchical, and distributed vision-based architecture. Data were taken simultaneously from the laser tracker and visual servoing implementation, enabling comparison of the data streams.

Keywords

computer vision, laser tracker, dynamic 6DOF metrology, performance evaluation.

1. INTRODUCTION

Real-time three-dimensional vision has been rapidly advancing over the past twenty years, leading to a number of successful laboratory demonstrations, including real-time visual servoing [7,16,17]. However, the advances have frequently not yet made the transition to commercial products, due in part to a lack of objective methods for empirical performance evaluation. To ensure a new algorithm for optical flow, stereo, visual servoing, laser Simultaneous Localization And Mapping SLAM, or other dynamic visual 3D task is valid, it would be very helpful to have a reference standard sensor system (ground truth) along with appropriate metrics for the comparison of test systems with the reference system. Standards and test procedures for dimensional metrology are well-established and highly accurate for static measurements, with coordinate measuring machines and laser trackers giving position measurements to microns. However, the theory, technology, and test procedures are not well established for dynamic dimensional measurements in uncontrolled environments.

In this paper we demonstrate the use of a dynamic, six-degree-of-freedom (6DOF) laser tracker to empirically evaluate the performance of a real-time visual servoing implementation, with the objective of establishing a general method for evaluating real-time 6DOF dimensional measurements of an object or assembly component under moderately constrained motion. The proposed evaluation procedure collects data simultaneously from the laser tracker and the visual servoing system under test, so the two data streams can be compared. Laser trackers produce highly accurate position and orientation data at a high data rate and for the purposes of this study will be considered ground truth.

The questions addressed in this work primarily focus on how to collect and compare the data streams. Issues here include synchronizing the data streams, so individual data points are taken at the same time; external calibration of the two sensors, so individual data points can be compared in the same coordinate system; and comparison metrics, so individual data points can be compared between the two sensors to determine how close the system under test comes to the ground truth. It would be useful to have comparison metrics that are robust to errors in synchronization and calibration, or even not dependent on calibration, to ensure accurate comparisons in the field.

The real-time visual servoing implementation used in this study was developed at the Purdue Robot Vision Lab using a subsumptive, hierarchical, and distributed vision-based architecture for smart robotics [3,6,16,17]. This is a robust, advanced dynamic visual servoing implementation with a high level of fault tolerance to non-cooperative conditions such as severe occlusions and sudden illumination changes. The Purdue system combines a ceiling mounted camera with a trinocular system mounted on the robot end-effector, and uses position based visual servoing (PBVS). The work in this paper is aimed at the evaluation of sensors for PBVS, in which the servoing system senses the position and orientation of the part in 3D coordinates, as opposed to image based visual servoing (IBVS), in which the servoing system senses the position and orientation of the part in 2D image coordinates.

2. PREVIOUS WORK

The general need for empirical evaluation has been well-recognized in the computer vision literature. Here we review articles on empirical evaluation of sensor-only 6DOF static and dynamic pose estimation, as well as articles on evaluation of

PerMIS'08, August 19-21, 2008, Gaithersburg, MD, USA.
Copyright 2008 ACM 978-1-60558-293-1…$5.00.

visual servoing algorithms that may combine sensor evaluation with robot control issues. In reviewing the literature, there are a number of questions that can be asked: what is under evaluation (pose estimation only, pose calculated for visual seroving, or other tasks), how measurements are taken from the system under test, how ground truth or other reference measurements are taken, what metrics are computed from the measurements, and how the metrics are interpreted to give summary judgments on system performance. We also have looked at the conditions of each test to consider the objects, motions, occlusions and environment.

For pose estimation, most evaluation papers consider static pose only [9,12] and not dynamic 6DOF sensor measurements of a part under motion. References [12,13] use Monte Carlo simulation for the evaluation of pose algorithm accuracy under noise and object orientations. In those articles, results are given for pose estimation for a complex industrial part and the error from unnamed ground truth is plotted as position or orientation error vs. the rotation of the object. The key result is to note the error as a function of part rotation varies considerably, spiking at ambiguous orientations of the object. Two papers that do consider dynamic pose are [10,15]. In [10] benchmarked video sequences are used for tests of a model-based algorithm with four parameter variations to analyze the relative contributions of subcomponents such as the edge detection operator or search technique. The results are given as deviations from the results of the one parameter set that successfully maintained track through the video sequences, but the nature and quality of this retrospective ground truth is not described in the article. In [15] three tracking approaches for 6DOF pose estimation and grasping of hand-held objects are evaluated using ground truth from an unnamed infrared marker tracking system good to 1.5 meters in position but with no rotation accuracy or measurements per second cited. The three approaches run at 8 Hz to 25 Hz. The article gives results in graphs that compare ground truth position and orientation data to robot end-effector position and tracked position, but no quantitative or summary statistics are given for the graphed data.

For visual servoing, many papers that present a new approach include an empirical evaluation, but since the paper emphasizes the development of the new approach, the evaluation section can be brief. An exception is [8] which uses sensitivity analysis and simulation to compute the contribution of image measurement errors to the calculated pose and control trajectory for PBVS and hybrid visual servoing.

The metrics used to evaluate pose estimation and visual servoing systems vary. Typical is the mean and standard deviation of a measure of error in world coordinates, including individual differences for each coordinate, a norm for position and orientation separately, or rarely a combined norm for all 6 degrees of freedom. The orientation can be compared in roll-pitch-yaw or as quaternions. In experiments without ground truth in world coordinates, or for IBVS in which pose in world coordinates is not computed, errors are computed in the image domain. [5] uses the reprojection error in the image domain. In some visual servoing evaluations, the metric is the number of cases successfully completed during the experiments.

In physical experiments in the evaluation of pose estimation or visual servoing, a mechanism must be used to generate motion, frequently a robot arm [2,4,15]. [15] uses an arm to move a camera towards a car battery through a known trajectory linear in both translation and angle, and repeats the motion 80 times to judge repeatability of the tracking algorithm.

3. PURDUE LINE TRACKING SYSTEM

Using robots to perform industrial assembly tasks is not new. In fact, robots have been used successfully in such applications over the past few decades. However, one common constraint still present in most of these applications is that the objects to be manipulated by the robot must be stationary in a known position, or moving along a well-known path with a very small amount of uncertainty. Consequently, industrial tasks such as painting, palletizing, welding, or decking have traditionally been set up in stationary locations in the assembly line where the involved parts have to come to a complete stop.

In order to eliminate inflexible, expensive stationary stations in the assembly line, robots must be able to perform their tasks on moving targets. The Purdue Robot Vision Lab addresses this goal using a subsumptive, hierarchical, and distributed vision-based architecture for smart robotics [3]. The system consists of multiple real-time control modules running in parallel, where each module is controlled by a different tracking method, with unique capabilities with respect to accuracy, computational efficiency, sensitivity to varying conditions, etc. By taking the most reliable input from all the modules, the system is able to achieve a high level of fault tolerance and robustness to non-cooperative conditions such as severe occlusions and sudden illumination changes.

In this architecture, each control module can be in a different hierarchy level. The more accurate the module is, the higher its hierarchy level. Modules from higher hierarchies can subsume functionality of modules in lower levels. Each module can run independently in a different computer over a network. Any number of modules is allowed in any given level of hierarchy, providing redundancy and fault tolerance.

Each module is composed of two main parallel processes. The first process is the visual-tracking loop that estimates the 6DOF 3D pose of the target object. This loop can be further broken down into three main parallel threads: data acquisition, data processing, and message exchange. The second process is the visual-servoing loop, which generates commands to the robot to move its end-effector to a desired position. This second loop is composed of two main threads: the message exchange and the control law calculation.

Even though every control module is able to run at its own sample rate and accuracy level in our architecture, only one module will be able to pass its generated command to the robot controller. The arbitrator listens to all the control modules and decides which module input to use to control the robot, based on module availability, estimation reliability, and module hierarchy.

Finally, the robot controller is implemented at the top of the robot controller interface. Therefore, the control modules in our system fall into the position-based look-and-move category [7]. This means that the robot controller is receiving commands directly in the Cartesian space, which makes the control plant much simpler compared to the image-based servoing category [3]. However, the position-based servoing requires camera calibration, robot calibration, and hand-eye calibration [6].

3.1 Peg and Hole Experiment

In order to demonstrate the capabilities of this architecture, we set up a variation of the traditional peg-and-hole experiment (see Figure 1). In our experiment, we attached a hollow wooden cylinder to an engine cover part, and attached a peg to the end-effector of a 6DOF robot manipulator. The engine cover is loosely attached to a linear slide, so the main motion of the engine cover is provided by the motion of the linear slide. In addition, we also attached strings to the engine cover, which a human can manipulate to generate random motions of the engine cover during the experiment. The goal of this experiment is to command the robot to insert the peg into the cylinder on the engine cover while the engine cover is in motion, resembling the automation needed to perform tasks such as glass decking or wheel decking on-the-fly [16].

Our system consists of three control modules, a system arbitrator, and a robot controller interface. The three control modules are: coarse control, model-based fine control, and stereo-based fine control.

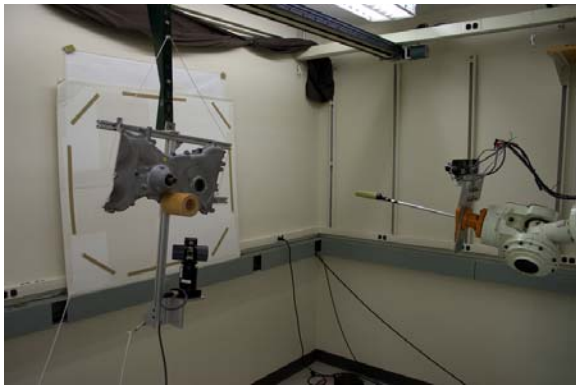

Figure 1. Peg and Hole experiment

3.2 Coarse Control

The coarse control module resides at the lowest level in the system hierarchy, meaning that it will work only when the other two modules become unavailable or unreliable. Its main purpose is to provide an initialization point for the controllers with higher hierarchy. That is, the coarse control is intended to track the target and command the robot to an approximate location in front of it. This module only requires the level of accuracy that would place the end-effector in front of the target such that cameras for the other controllers can view the target.

Coarse control visual-tracking uses a camera mounted on the ceiling with a view of the entire workspace. When the target object enters the workspace, the bar holding the target is detected and the 3D translational position of the bar in the coordinates of the robot is estimated. No rotation information is needed here. This estimated pose is passed to the visual-servoing loop that in this case would simply command the robot to move its end-effector near the target.

3.3 Stereo-based Fine Control

This is the control module that resides at the highest hierarchy level in the system, meaning that while tracking the target it subsumes the other two modules. It uses a stereo pair of cameras located on the robot's end-effector.

The visual-tracking loop in this module uses a blob analysis algorithm to detect, in both cameras, three prominent coplanar features on the engine cover. Then, using the calibration information, the 3D coordinates of those features are reconstructed, and the 6DOF 3D pose of the target is estimated in the robot coordinate system [17]. Based on the estimated pose, the visual-servoing loop then performs the peg-and-hole motion using a Proportional-Integral-derivative (PID) control law.

Since the 3D pose estimation relies on the visibility of three specific features on the engine cover, this module will only work if all three features are detected in both cameras.

Using an efficient blob analysis algorithm makes this module run very fast. It takes an average of 8.2 ms to estimate the 3D pose of the target, which is more than sufficient to process stereo images at 30 frames per second. However, this approach requires some thresholds to extract the blobs, making this module sensitive to sudden illumination changes.

3.4 Model-Based Fine Control

The main purpose of this module is to provide redundancy to the stereo-based fine controller. It uses a monocular vision system and a known wire-frame model of the target [16].

The visual-tracking loop of this module first projects the model into the input scene with respect to the initial pose that is given by the coarse control module or the stereo-based control module. Then, it sequentially matches the straight lines of the wire-frame model to the detected edges in the scene for an updated calculation of the pose of the target. For robust pose estimation, it uses a backtracking scheme for correspondence search [16].

Since this module uses extracted edges in the image, as opposed to applying thresholding to extract blobs in the stereo-based control module, it is less sensitive to illumination changes. Also, since the number of model features used in this module is typically much larger than 3 (the bare minimum number of features required to estimate the pose of an object), it is naturally robust to partial occlusions. However, it is slower than the Stereo-Based controller. It takes an average of 48 ms to estimate the 3D pose of the target, where 80 % of the computation time is taken by the edge detection phase.

4. EVALUATION EXPERIMENTS

In the experiments, the visual tracking system was set up to perform the peg and hole task using an engine part cover as the target. The engine part was either stationary or moved by an overhead linear rail at a constant velocity. The part was suspended from the linear rail to allow the experimenters to move the part back and forth by an attached string.

During each pass of the experiment, the position of the engine part was measured simultaneously by the Purdue visual tracking system and the NIST laser tracker. Data acquisition from each system was triggered by a common hardware signal, and the data time stamped and streamed to storage for off-line comparison.

The key elements of the data collection system were the laser tracker, the simultaneous data collection, and the establishment of common coordinate systems through calibration.

4.1 Laser Tracker

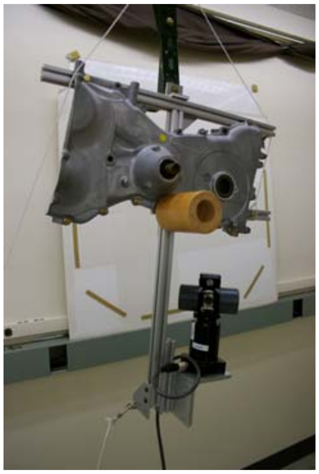

Figure 2. SmartTRACK sensor below engine part

The laser tracker system used consisted of two major components, a base unit and an active target, both made by Automated Precision Inc[1]. (API). The active target is a SmartTRACK Sensor (STS) capable of determining its orientation in 3D space. Weighing 1.4 kg, the STS has an angular resolution specification of ±3 arc-sec (or ±0.000833… degrees). The complete manufacturer's specification can be found in [18]. The base unit is the Tracker3^{TM} Laser Tracking System (T3) which tracks the 3D position of the STS. The T3 system has ± 10 ppm absolute accuracy (e.g., ± 50 µm at 5 m). In addition, the base unit can be used with a passive target (a spherically mounted reflector (SMR) retro-reflective mirror) instead of the active target for single point measurements.

Together, the T3 and STS provide an accurate but limited 6DOF pose estimation system. The STS and T3 devices can take measurements independently at their own rate or they can be connected to a common external trigger.

4.2 Synchronization Issues

Both the T3/STS system and the Purdue visual-tracking system allow an external signal to trigger their data acquisition. Although the STS/T3 system is capable of handling a trigger signal up to 150 Hz, the visual-tracking system requires a 30 Hz data stream. We use a single 30 Hz trigger signal shared by both systems.

Although a shared trigger signal provides a solution to simultaneous measurements, the trigger itself needs to be reliable, repeatable, and controllable. We use a digital function generator that can be programmed to produce a clean squarewave signal. The digital function generator also allows us to start and stop the signal with a push of a button. The resulting signal is deterministic and free of the button-bouncing effect typically associated with inexpensive analog function generators.

The data collection programs maintain their own sequence number, which is increased each time a trigger signal is received. In addition, the program time-stamps the sequence number with a microsecond timer from the computer clock. Both systems synchronize their computer clock with a NTP (Network Time Protocol) server every 10 s throughout the entire data collection.

For data to be a matched pair, we require them to have the same sequence number. In addition, we check the difference between the corresponding timestamps and verify that the difference is small (i.e., the difference should never be greater than the period of the trigger signal shared by the two systems).

4.3 External Calibration Issues

To evaluate data collected during the experiments, we need to define metrics to compare the 6DOF pose data collected by the API T3/STS laser tracker and the Purdue line tracking system. There are eight coordinate systems involved, defined in Figure 2. We want to establish relationships between data collected in the coordinate system T of the laser tracker and one more coordinate systems of the Purdue system. We use the notation ${}_X H_Y$ to denote the homogenous transformation from coordinate system Y to X, so ${}_T H_W$ is a transformation from W coordinates to T coordinates. The 6DOF pose of the object O in the T3 coordinate system would be represented by ${}_T H_O$.

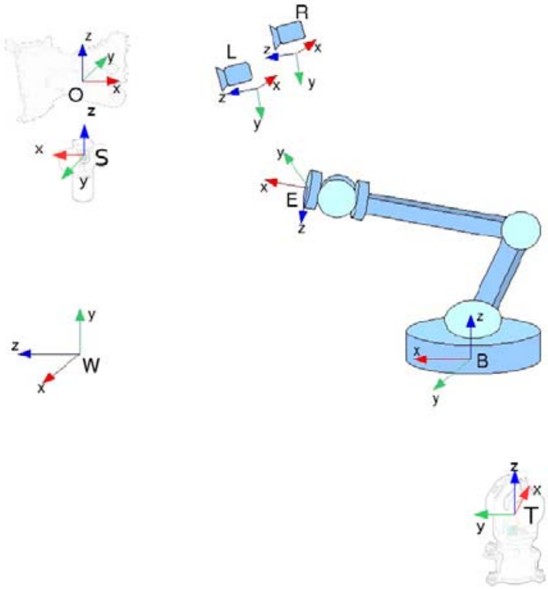

Figure 2. The coordinate systems of the Purdue Line Tracker and the STS/T3 Tracker System

[1] Certain commercial equipment, instruments, or materials are identified in this paper in order to adequately specify the experimental procedure. Such identification does not imply recommendation or endorsement by NIST, nor does it imply that the materials or equipment identified are necessarily best for the purpose.

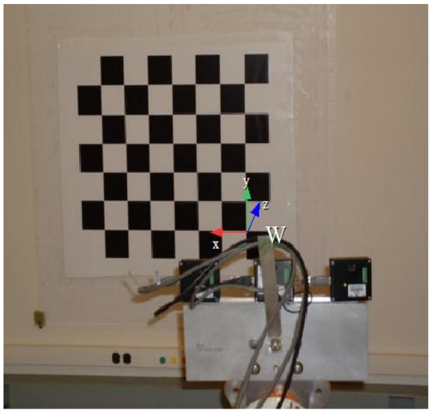

Figure 3. The world coordinate system

External calibration provides an estimate of the homogeneous transformation matrix (H) between the world W, and the robot base B, where W is defined by the calibration pattern behind the target as shown in Figure 3.

If we can successfully use the laser tracker coordinates T to establish the transformation ${}_T H_W$ or ${}_W H_T$ between the world (W) and T3, we can relate the coordinate T to the robot base B coordinate as in Equation 1.

$${}_T H_B = {}_T H_W \times {}_W H_B \qquad \text{Eq. 1}$$

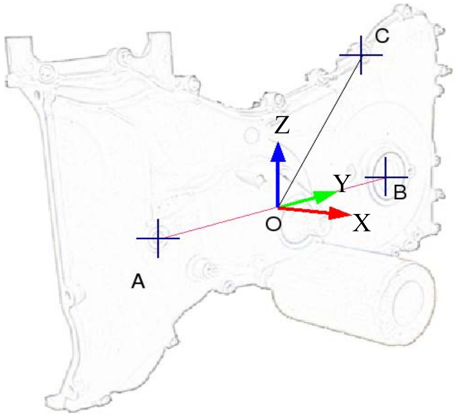

Figure 4. The object coordinate system (Y=OB, Z=up)

We don't have direct access to the origin of the engine cover O coordinates. Instead we use the Cartesian positions of three features (A,B,C), to reconstruct the coordinate frame for O with respect to coordinate T (see Figure 4).

We use the T3 laser tracker with an SMR to measure A, B and C in T coordinates. We then compute the origin O as the midpoint between A and B, and use the position of C to complete the object reference frames. Hence we construct ${}_T H_O$ or ${}_O H_T$. Similarly, we construct ${}_T H_W$ by measuring the origin of W.

Given ${}_S H_T$ from T3/STS system, we obtain ${}_S H_O$ from Equation 2.

$${}_S H_O = {}_S H_T \times {}_T H_O \qquad \text{Eq. 2}$$

Using Equation 1 and Equation 2, we establish transformations between the T3 base system T and the object system O in both the T3 and Purdue system, and compare those two transformations. The ground truth transformations provided by the T3/STS are on the left hand side, and the measurements provided by Purdue system and the hand-eye calibration are on the right hand side:

$${}_W H_T \times {}_T H_S \times {}_S H_O = {}_W H_B \times {}_B H_O \qquad \text{Eq. 3}$$

This gives the relationship ${}_W H_O(laser) = {}_W H_O(visual)$ and the inverse between the two systems in a common coordinate system, where visual stands for the Purdue tracking system.

Now we relate the two estimations of the inverse ${}_O H_W$ in the world coordinate system. In Equation 3, the calibration error in ${}_W H_T$, ${}_S H_O$ and ${}_W H_B$ can't be eliminated, so the absolute measurement approach was not used in this paper.

Since the Purdue system is a tracking system, the differential pose with respect to time is computed every cycle. It is reasonable to compare the differential pose between ground truth and the Purdue system using the world coordinates. We define ${}_{Oi} H_W(P)$ as the pose of the engine part in the world coordinate at time i for the Purdue Tracking system ${}_{Oi} H_W(G)$ as the pose of the engine part in world coordinate at time i for the ground truth system (laser tracker).

We can then obtain the following relationships for each system between time i and time i+1:

$${}_{Oi} H_W(P) \times {}_W H_{Oi+1}(P) = {}_{Oi} H_{Oi+1}(P)$$

$${}_{Oi} H_W(G) \times {}_W H_{Oi+1}(G) = {}_{Oi} H_{Oi+1}(G)$$

As one can see from the above derivation, the differential measurement will not depend on coordinate system. This approach produces the same numerical result for ${}_{Oi} H_{Oi+1}(G)$ and ${}_{Oi} H_{Oi+1}(P)$ in any coordinate system.

5. EXPERIMENTAL RESULTS

We present here preliminary results from a series of experiments conducted at Purdue in April, 2008. The protocol was to run three sets of experiments, one with the target stationary, one with the target moving with a simple linear velocity, and one with the target moving with a linear velocity but randomly displaced manually by the experimenters. The results are given below for the three sets.

The experiment had two complications. One is that the calibration of the laser tracker and Purdue system proved difficult and we were not able to establish a full set of coordinate transformations between laser tracker and visual system data. We did establish enough to compare relative poses, which are reported. Also, the

Purdue system is both a visual tracker and a robot control system. The data may at times confound the effects of both subsystems, as the robot introduces a physical motion. We only wished to evaluate the performance of the visual tracker.

5.1 Stationary Tests

The stationary tests allowed us to evaluate the basic performance of both systems and assure that the laser tracker was performing to specifications after shipping. The target was placed in four positions and data collected for 15 to 30 seconds for each. The results showed that both systems performed within specifications.

Table 1 lists the standard deviations of the stationary data set measured by the STS/T3 system. The results are consistent with the specifications [7]. The laser tracker stayed in a fixed position so the target distance varied.

Table 1. STS/T3 system: repeatability for stationary data

	Sample Size	T3/STS mean distance (mm)	2 std (mm)
Position 1	466	3550.054	0.006
Position 2	1157	3781.466	0.006
Position 3	1050	3882.787	0.005
Position 4	1018	4002.035	0.008

Table 2 lists the standard deviations of the stationary data set measured by the Purdue system. The Purdue system moves the robot end-effector near the object, so the distance remains relatively constant. The results show a consistent value near 0.6 mm at a range of approximately 2.6 m.

Table 2. Purdue system: repeatability for stationary data

	Sample Size	T3/STS mean distance (mm)	2 std (mm)
Position 1	466	2670.582	0.629
Position 2	1157	2625.820	0.582
Position 3	1050	2625.036	0.560
Position 4	1018	2636.701	0.561

For use as a reference system, a metrology technology should have an accuracy at least one order of magnitude greater than the system under test. In this case, the laser tracker is two orders of magnitude more accurate than the Purdue vision system.

5.2 Linear Motion Tests

In the linear motion tests, the target was moved about 1.5 meters left to right and tracked by both the laser tracker and the Purdue system. For each trial, the motion was repeated 30 times as the target moved, and then was quickly moved back to the start position. The backward sweep was ignored in the data analysis as the Purdue system only tracked during the forward motion.

The differential motion as measured by both systems was used to determine the consistency between laser tracker and Purdue system. In effect, the comparison is being made on the measured speed of the target in each separate coordinate (X,Y,Z, roll, pitch, yaw).

The data below are from pass 6 of the first trial, and are typical of the linear runs. The sample size is 453, with 33 ms between data points. In the graphs below the horizontal axis is frame number, and the vertical axis is the difference between the laser tracker motion change (delta) in each coordinate and the Purdue system delta. Since the laser tracker was accepted as ground truth, the difference is defined as the error in the Purdue system.

In Figure 5 the errors can be seen to be consistent and relatively independent of position along the path, although that is yet to be evaluated statistically. The error does vary by coordinate – different coordinates proved to be more sensitive to the left-to-right motion. In object coordinates the target is moving in the Y-Z plane, primarily in the Y direction, and the Z coordinate has slightly less error. Similarly, the roll angle is around the X axis, and proved an order of magnitude reduced in error over the other rotations. Table 3 quantifies the error values, while Figure 6 gives a histogram of the error values to show roughly symmetric, zero centered error distributions. For coordinate Y there appears to be a secondary peak in positive error.

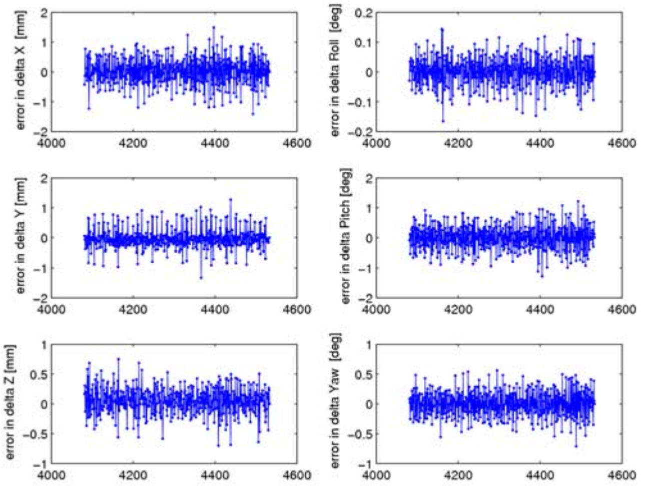

Figure 5. Pass 6 – Coordinate Errors in Delta vs. Frame Number

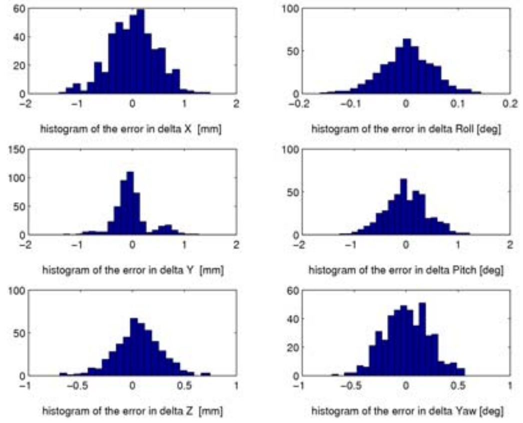

Figure 6. Pass 6 – Coordinate Errors in Delta (histogram)

Table 3. Pass 6 – Coordinate Error Statistics (n=453)

	Mean	Std	Max	Min
X in mm	0.01614	0.45376	1.49003	-1.41190
Y in mm	-0.03513	0.33355	1.27399	-1.32852
Z in mm	0.05206	0.21991	0.74652	-0.69851
distance	-0.04305	0.33273	1.21320	-1.50436
Roll in deg	0.00168	0.04915	0.14368	-0.16532
Pitch in deg	0.00088	0.40811	1.21672	-1.27938
Yaw in deg	0.00184	0.21589	0.56194	-0.71260
Total angle	-0.04798	0.05195	0.03383	-0.28635

5.3 Shaking Motion Tests

In the shaking motion tests, the basic target motion and repetitions were identical to those in the moving target test, but the experimenters could pull a string to swing the target with an impulse motion. This was done through a series of different motions – first with no extra motion to match the linear case, then with the impulse motion varying by amplitude and frequency.

The data below are from pass 3 of the first shaking motion trial. The sample size is 455, with 33 ms between data points. In the graphs below, the horizontal axis is frame number, and the vertical axis the difference between the laser tracker delta in each coordinate and the Purdue system delta.

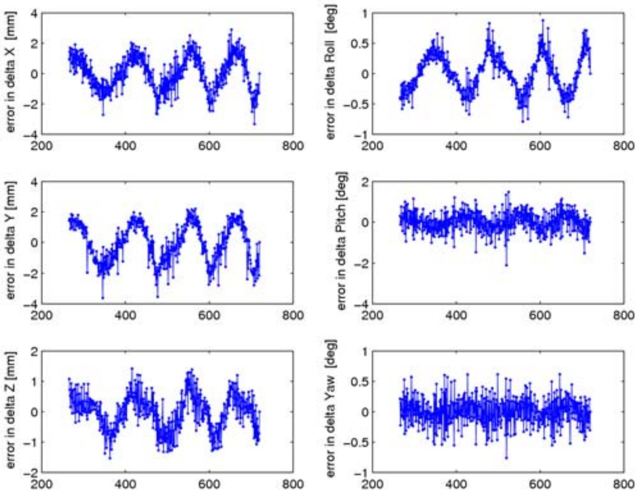

Figure 8. Pass 3 – Coordinate Errors in Delta vs. Frame Number

The graphs show four impulse motions as the target was pulled back four times, relatively smoothly and consistently. The error first goes positive as the speed of the target slows down and the Purdue system undershoots the speed, and then negative as the Purdue system overshoots the speed. The graph scales have changed from the linear motion case, as the error range has approximately doubled. The Z axis remains the one with lowest error, while the roll angle error is greater compared to the linear motion tests as the impulse motion rotated the target around the X-axis.

6. CONCLUSIONS

In this paper we have demonstrated the use of a precision laser tracker to evaluate a state-of-the-art visual servoing perception system, with the objective of establishing general techniques for evaluating 6DOF sensors. The demonstration involved a synchronized data collection system that used a hardware trigger to collect at a 30 Hz rate, while the laser tracker has the ability to collect at up to 150 Hz. The laser tracker was verified accurate enough for the approximate 1 mm range of error that needed to be measured.

Future work will involve a more detailed analysis of the data and the establishment of better calibration techniques and metrics to insure consistent comparisons between laser tracker and sensor data streams.

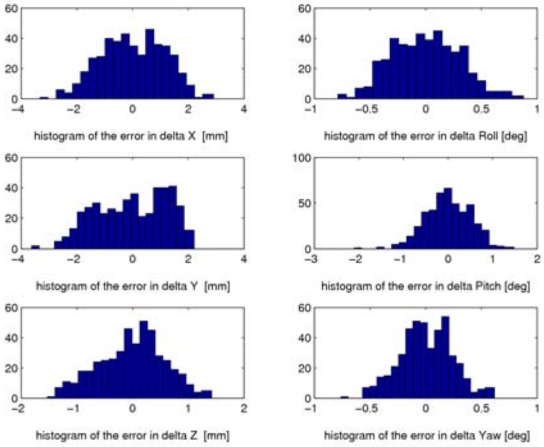

Figure 9. Run 6 – Coordinate Errors in Delta (histogram)

Table 4. Run 6 – Coordinate Error Statistics (n=453)

	Mean	Std	Max	Min
X in mm	0.00219	1.13255	2.90429	-3.32659
Y in mm	-0.04633	1.26350	2.21070	-3.62808
Z in mm	-0.01445	0.58950	1.42051	-1.53047
distance	0.90879	0.95975	3.77670	-1.84237
Roll in deg	0.00362	0.30460	0.87745	-0.78979
Pitch in deg	-0.00329	0.50645	1.48264	-2.10615
Yaw in deg	0.00103	0.23520	0.61750	-0.76112
Total angle	-0.42713	0.30752	0.15745	-1.51979

7. ACKNOWEDGMENT

We thank Elena Messina and Hui-Min Huang for the fruitful discussions. We thank Nick Scott, Alan Lytle, Chris Blackburn, Daniel Sawyer, Steve Phillips from NIST, Avi Kak from Purdue University, Jane Shi from General Motors, and Kam Lau, Alan Cui, Zaifeng Chen from API for their collaborations.

8. REFERENCES

[1] Chai, X., Shan, S., Qing, L. and Gao, W. 2006 Pose Estimation Based on Gaussian Error Models, International Conference on Biometrics, 2006: 136-143

[2] Deguchi, K., 2000 A Direct Interpretation of Dynamic Images with Camera and Object Motions for Vision Guided Robot Control. International Journal of Computer Vision 37(1): 7-20 (2000).

[3] DeSouza, G. N. and Kak, A. C., 2004, A Subsumptive, Hierarchical, and Distributed Vision-Based Architecture for Smart Robotics, IEEE Transactions on Systems, Man, and Cybernetics -- Part B: Cybernetics, 34(5): 1988-2002.

[4] Espiau, B., 1993 Effect of Camera Calibration Errors on Visual Servoing in Robotics. Experimental Robotics III, The 3rd International Symposium, Kyoto, Japan, October, 1993: 182-192.

[5] Fiore, P. D, 2001 Efficient Linear Solution of Exterior Orientation, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23(12): 140-148, 2001

[6] Hirsh, R., DeSouza, G. N. and Kak, A. C., 2001 An Iterative Approach to the Hand-Eye and Base-World Calibration Problem, Proceedings of the IEEE International Conference on Robotics and Automation, Seoul, May 2001

[7] Hutchinson, S.A., Hager, G.D., and Corke, P.I, 1996 A tutorial on visual servo control. IEEE Transactions on Robotics and Automation, 12(5):651--670, October 1996.

[8] Kyrki, V., Kragic, D., and Christensen, H. I., 2006 Measurement errors in visual servoing. Robotics and Autonomous Systems 54(10): 815-827 (2006)

[9] Madsen, C.B, A comparative study of the robustness of two pose estimation techniques. In H.I. Christensen, W. Förstner, and C.B. Madsen, editors, Proceedings: ECVnet Workshop on Performance Characteristics of Vision Algorithms, held in conjunction with Fourth European Conference on Computer Vision, Cambridge, England, pp. 181 - 200, April 1996.

[10] Preisig, P.. and Kragic, D., 2006 Robust Statistics for 3D Object Tracking. Proceedings of the Int. Conference on Robotics and Automation, Orlando, Florida, 2006.

[11] Chai, X., Shan, S., Qing, L. and Gao, W. 2006 Pose Estimation based on Gaussian Error Models, Proceeding of International Conference on Biometrics (ICB2006), Lecture Notes on Computer Sciences (LNCS3832), D.Zhang, A.Jain (Eds.), pp136-143, Hong Kong.

[12] Reid, G. J., Tang J., and Zhi, L., 2003 A complete symbolic-numeric linear method for camera pose determination, International Symposium on Symbolic and Algebraic Computation, Philadelphia, Pennsylvania 2003: 215-223.

[13] Rivera-Ríos, A. H., Shih F. L, and Marefat, M. M. 2005 Stereo Camera Pose Determination with Error Reduction and Tolerance Satisfaction for Dimensional Measurements, Proceedings of the Int. Conference on Robotics and Automation, Barcelona, Spain, 2005: 423-428.

[14] Sepp, W., Fuchs, S. and Gerd Hirzinger. 2006 Hierarchical Featureless Tracking for Position-Based 6-DoF Visual Servoing, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2006, Beijing (China), 2006

[15] Tonko, M. and Nagel, H. 2000 Model-Based Stereo-Tracking of Non-Polyhedral Objects for Automatic Disassembly Experiments, International Journal of Computer Vision 37(1): 99-118 (2000).

[16] Yoon, Y., DeSouza, G. N. and Kak, A. C., 2003 Real-time Tracking and Pose Estimation for Industrial Objects using Geometric Features, in Proceedings of the Int. Conference in Robotics and Automation, Taiwan, 2003

[17] Yoon, Y., Park, J. and Kak, A.C. 2006 A Heterogeneous Distributed Visual Servoing System for Real-time Robotic Assembly Applications, in Proceedings of the Int. Conference on Robotics and Automation, Orlando, Florida, 2006.

[18] SMARTTRACK SENSOR, Automated Precision Inc., http://www.apisensor.com/PDF/SmartTrackeuDE.pdf, 2007.