Automatic Visual Station Keeping of an Underwater Robot

Richard L. Marks * Howard H. Wang † Michael J. Lee ‡ Stephen M. Rock §

Abstract **This paper presents a method for drift-free station keeping of an underwater robot using computer vision. The sensing problem is simplified by assuming an active control system can be used to keep positional errors small. Robot position is obtained by tracking texture features using image filtering and correlation. Errors in four degrees of freedom (translation and yaw) are determined in real time and are fed into a robot control system to accomplish the task of station keeping. Experimental results demonstrating sensing quality and robot station keeping are presented.**

I. INTRODUCTION

The ability to hold station is critical for an operational underwater robot. Staying fixed at a given position and orientation is a required behavior for successful operation of many mobile systems (vehicles, robots, etc). This ability is often taken for granted in land-based robotics because of the ease of implementation. However, an underwater robot must deal with unknown currents and forces (e.g., from a tether) which can cause its position to change undesirably. In addition, for typical systems the translation and heading degrees of freedom have no inherent restoring forces to counter these disturbances automatically. Thus, an underwater robot must sense and actively control its position and orientation in order to keep from drifting. The control performance that is achievable relies heavily upon the quality of the sensing.

II. BACKGROUND

Underwater position and orientation sensing are subjects that have been given much attention. The orientation of an underwater robot is typically easier to determine than robot position. The pitch and roll angles can be accurately obtained by sensing the direction of gravity, whether it be with a pendulum, accelerometers, etc. Heading (or yaw) is typically sensed using some type of compass. However, the quality of heading sensors is greatly affected by the local magnetic properties of the surroundings. In areas such as the Antarctic, magnetic heading sensors are virtually unusable [2].

*Doctoral Candidate, rlmarks@sun-valley.Stanford.Edu

†Doctoral Candidate, lazarus@sun-valley.Stanford.Edu

‡Senior Research Engineer, Monterey Bay Aquarium Research Institute, lemi@hp850.Mbari.Org

§Associate Professor, Department of Aeronautics and Astronautics, rock@sun-valley.Stanford.Edu

A variety of position sensing systems exist. Planar motion determination is often accomplished using dead reckoning, accelerometers, and Doppler sonars. However, the positions obtained from these systems are susceptible to integration drift, which is not an acceptable attribute for a station keeping sensor. Depth (distance to the surface) can be determined by measuring static pressure. Transponder systems provide drift-free position measurements once they have been put in place. However, these measurements are not directly related to the local ocean floor environment in which the robot is operating. Point-ranging sonars are often used to measure altitude (distance from the bottom). The use of sonar for position station keeping is the focus of current research [4].

Computer vision is another sensing approach that can be used to measure position and orientation of mobile systems [20, 1, 3]. The use of (optical) computer vision for underwater position and orientation sensing was studied extensively in [15, 13, 12]. Fox and Negahdaripour specifically addressed the problem of near-bottom optical station keeping [5, 14, 11, 10]. Their most recent approach involved extending standard *optical flow* theory to handle difficulties introduced by the underwater environment. They proposed that consecutive-frame optical flow measurements could be used to determine camera motion and therefore be used to compensate for an underwater vehicle's drift from station. However, such an approach does not overcome the problem of slow station drift due to buildup of measurement errors over time. To address this problem, we have developed an approach to accomplish true drift-free station keeping using computer vision.

III. APPROACH

To avoid drift, our approach determines camera motion by comparing the current live image with the initial image obtained when the station keeping command was given. In general this cannot be used to determine motion because the live image changes significantly compared to the stored image. However, for the specific task of station keeping, there is a control system that actively attempts to keep the robot stationary. Thus, for static scenes and limited-magnitude disturbances, the live image and initial image will change only by small amounts. Based upon this, we have developed a texture tracking strategy to determine robot position and orientation error.

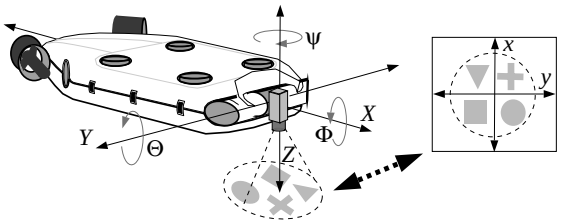

Figure 1: **Robot coordinates and image coordinates. The gray shapes show camera orientation.**

A. Displacement equations

Figure 1 shows an underwater robot with a downward-pointing camera. Let $\mathbf{q} = \begin{pmatrix} q_x & q_y & q_z & q_\Phi & q_\Theta & q_\Psi \end{pmatrix}^T$ represent the robot state at the time the station keeping command is given. At some arbitrary time later the robot's state will have changed by an amount $\delta\mathbf{q} = \begin{pmatrix} \delta q_x & \delta q_y & \delta q_z & \delta q_\Phi & \delta q_\Theta & \delta q_\Psi \end{pmatrix}^T$ as measured in the body-fixed coordinates corresponding to $\mathbf{q}$. Consider an inertially-fixed point whose position is $\mathbf{P} = \begin{pmatrix} X & Y & Z \end{pmatrix}^T$ as measured in the body-fixed coordinates corresponding $\mathbf{q}$. Now consider the position $\mathbf{P}' = \begin{pmatrix} X' & Y' & Z' \end{pmatrix}^T$ of the same point as measured in the body-fixed coordinates corresponding to $\mathbf{q}+\delta\mathbf{q}$ (Figure 2).

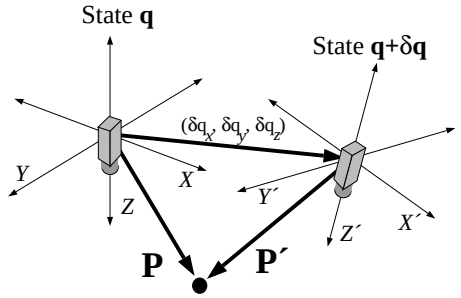

Figure 2: **Exaggerated robot state change.**

$\mathbf{P}'$ is related to $\mathbf{P}$ by:

$$\mathbf{P}' = \mathbf{R}^T(\mathbf{P} - \mathbf{t}) \tag{1}$$

where for $\delta q_\Phi, \delta q_\Theta, \delta q_\Psi \ll 1$:

$$\mathbf{t} = \begin{bmatrix} \delta q_x \\ \delta q_y \\ \delta q_z \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 1 & -\delta q_\Psi & \delta q_\Theta \\ \delta q_\Psi & 1 & -\delta q_\Phi \\ -\delta q_\Theta & \delta q_\Phi & 1 \end{bmatrix} \tag{2}$$

The assumption of small angles can be made because the robot control system will actively attempt to regulate the state to the desired state $\mathbf{q}$. Equation (1) can be expanded and rewritten as:

$$\mathbf{P}' = \begin{bmatrix} X - \delta q_x + \delta q_\Psi Y - \delta q_\Theta Z \\ Y - \delta q_y - \delta q_\Psi X + \delta q_\Phi Z \\ Z - \delta q_z + \delta q_\Theta X - \delta q_\Phi Y \end{bmatrix} \tag{3}$$

Introducing the transformations to image coordinates:[1]

$$\mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \end{bmatrix}, \quad \mathbf{p}' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \frac{X'}{Z'} \\ \frac{Y'}{Z'} \end{bmatrix} \tag{4}$$

Next we can substitute (4) and rearrange (3) to obtain:

$$\mathbf{p}' = \begin{bmatrix} \frac{x - \delta q_x/Z + \delta q_\Psi y - \delta q_\Theta}{1 - \delta q_z/Z + \delta q_\Theta x - \delta q_\Phi y} \\ \frac{y - \delta q_y/Z - \delta q_\Psi x + \delta q_\Phi}{1 - \delta q_z/Z + \delta q_\Theta x - \delta q_\Phi y} \end{bmatrix} \tag{5}$$

Assuming $\frac{\delta q_x}{Z}, \frac{\delta q_y}{Z}, \frac{\delta q_z}{Z} \ll 1$, we apply the Taylor series expansion $\frac{1}{1+\epsilon} \approx 1 - \epsilon$ to (5). Again, this assumption can be made because the robot will be under control during the station keeping task. Dropping higher order terms, we obtain the result:

$$\mathbf{p}' - \mathbf{p} = \frac{1}{Z}\begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix}\begin{bmatrix} \delta q_x \\ \delta q_y \\ \delta q_z \end{bmatrix} + \begin{bmatrix} xy & -1-x^2 & y \\ 1+y^2 & -xy & -x \end{bmatrix}\begin{bmatrix} \delta q_\Phi \\ \delta q_\Theta \\ \delta q_\Psi \end{bmatrix} \tag{6}$$

Note that this result can also be directly obtained from the equation for the optical flow Jacobian as presented in [9] by observing that the Jacobian matrix changes smoothly for small changes in the state. The derivation shown here, however, provides useful understanding about how the small state change assumption simplifies the general solution.

Equation (6) relates the induced image displacement of a point $\mathbf{P}$ viewed by the camera to the change in robot state $\delta\mathbf{q}$ and the range Z of the point from the camera. The image displacements of multiple points $\mathbf{P}_1 \ldots \mathbf{P}_n$ can be measured to increase the number of equations to $2n$; however, for each point $\mathbf{P}_i$, a new unknown Z_i is introduced. Before presenting a solution to this system of equations for the robot state, we present an approach for measuring image displacements $\mathbf{p}'_i - \mathbf{p}_i$.

B. Feature tracking

Many approaches to image feature tracking have been developed [19, 21]. As pointed out in [10], however, such feature tracking approaches cannot typically be used underwater due to the unique challenges imposed by the marine environment. Lack of necessary features, nonuniform lighting, low contrast, and marine snow can make feature tracking difficult and unreliable. To address these problems, our feature tracking approach uses the image filtering and correlation techniques proposed by Nishihara in [17] for stereo disparity calculation. Filtering is used to make image textures more pronounced,

[1] Note that these image coordinates are nonstandard: we have chosen to follow the robot/vehicle standard coordinate notation as shown in Figure 1.

and correlation is used to establish feature correspondence between images. Feature correspondences between the initial image and the live camera image determine the feature displacements $\mathbf{p}'_i - \mathbf{p}_i$.

The intensities I of an image are first filtered using the signum of Laplacian of Gaussian operation [17]:

$$J(x,y) = \mathbf{sgn}[\nabla^2 * \frac{e^{\frac{-x^2-y^2}{2\sigma^2}}}{2\pi\sigma^2} * I(x,y)] \qquad (7)$$

This bandpasses the spatial intensity frequencies and highlights zero-crossings of the spatial intensity gradient as a change in the binary value of the filtered output (see Figure 3). The output images are less affected by nonuniform lighting, ambient lighting changes, and image noise [6].

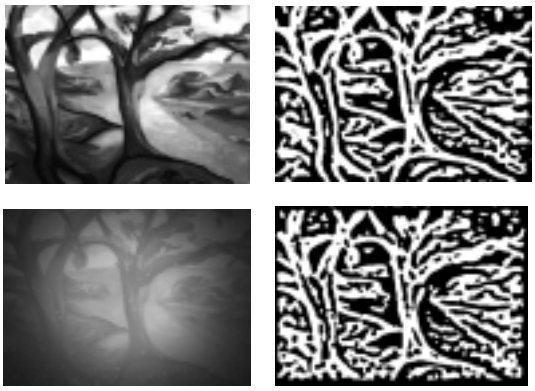

Figure 3: **Although the input images have significantly different lighting, contrast, and noise, the filtered images are quite similar.**

After the images are filtered, correlation is used to measure changes in the image. The zero-mean nature of the Laplacian of Gaussian and the binarization of the signum function effectively make correlation a simple XOR function. Multiple pixel correlations must be used to get a statistically-valid sample. The normalized correlation C between location (x,y) in the filtered initial image J and location (x',y') in the filtered live camera image J' is:

$$C = -wh + 2\sum_{j=-\frac{h}{2}}^{\frac{h}{2}} \sum_{i=-\frac{w}{2}}^{\frac{w}{2}} \mathrm{xor}[J(x+i,y+j), J'(x'+i,y'+j)] \qquad (8)$$

The features that are tracked are actually $w \times h$ rectangular pixel regions of the filtered initial image centered at $\mathbf{p}_i = (x_i, y_i)$. By choosing an appropriate region size and Laplacian of Gaussian filter width σ, features will contain sufficient texture for meaningful correlation. Techniques for monitoring the texture content of features have been explored but are beyond the scope of this paper.

The actual tracking of a feature located at (x,y) in J consists of finding (x',y') in J' that maximizes Equation (8). To minimize the number of correlations that must be performed, the search area is centered about the location of correspondence found on the previous iteration. The size of the area that must be searched is dependent on the robot velocity and image processing rate. Because we use special-purpose hardware to process the images at 30 Hz, the required search area is small (20×20 pixels) for typical robot motions. This feature tracking approach has been utilized for real-time video mosaicking of the ocean floor [8].

It should be noted that camera motion in the Z and Ψ directions can cause features to warp (scale and rotate). The effect of this warping on correlation is a direct function of the motion magnitude. The severity of this effect is a function of the ratio of the width of the Laplacian of Gaussian to the feature region size. For small motions and proper selection of this ratio, the effects of feature warping on correlation can be ignored for the task of station keeping.

C. Solving for the robot state

It would appear that by tracking a large number of features, the robot state could be determined in a straight-forward manner. Optical flow researchers have shown that this is not the case due to the inherent non-observability and poor conditioning of the problem. Feature displacements cannot be used to determine the absolute ranges Z_i without other knowledge of the change in robot state or a model of the scene structure. In addition, differentiating between changes in δq_x and δq_Θ and between δq_y and δq_Φ is poorly conditioned, although recently proposed solutions show promise [18]. Our solution to these problems is to focus again on the characteristics specific to underwater robot station keeping.

As pointed out previously, pitch (Θ), roll (Φ), and altitude (Z) can be sensed using other means. For most underwater robots, pitch and roll angles are small and can be assumed to be zero due to designed stability from a large separation between the center of buoyancy and center of gravity. In other cases, the direction of gravity can be sensed to accurately determine pitch and roll. We will therefore assume that δq_Φ and δq_Θ are known quantities.[2] Rewriting Equation (6):

$$\Delta\mathbf{p}_i - \begin{bmatrix} x_i y_i & -1-x_i^2 \\ 1+y_i^2 & -x_i y_i \end{bmatrix} \begin{bmatrix} \delta q_{\Phi_0} \\ \delta q_{\Theta_0} \end{bmatrix} = \begin{bmatrix} -1 & 0 & x_i & y_i \\ 0 & -1 & y_i & -x_i \end{bmatrix} \begin{bmatrix} \delta q_x/Z_i \\ \delta q_y/Z_i \\ \delta q_z/Z_i \\ \delta q_\Psi \end{bmatrix} \qquad (9)$$

[2]The error in pitch and roll measurements will be directly reflected in the determination of δq_x and δq_y. For large errors, a vertically-stable robot may exhibit a limit-cycling behavior with x and y periods roughly equal to the pitch and roll natural frequencies.

From Equation (9), we see that it is impossible to determine the absolute ranges (altitudes) Z_i solely from feature displacements regardless of the number of points used. Therefore, it is also impossible to determine absolute robot translation. It can be shown that given the absolute range of a single feature, the displacements of multiple features can be used to determine the robot state. However, because of measurement noise, this problem is typically inconsistent and involves a residue minimization (least-squares fit). Although this can be accomplished, the nonlinear relation between measurements and unknowns makes the problem computationally expensive.

We have chosen to simplify the problem by assuming that the absolute ranges of tracked features can be measured by other means. Our system uses stereo vision to determine these ranges. Note that feature ranges only need to be computed for the *initial* image; thus, the ranges are calculated once at the beginning of the station keeping task. The stereo vision system uses the same filtering and correlation techniques as the feature tracking presented above and is addressed in [7, 6, 17].

Assuming the feature ranges are known, two features must be tracked to uniquely determine the robot state, although more features can be tracked to minimize the effects of noise and measurement error. However, because the problem is well-conditioned, we have chosen to keep the solution simple and track only two features.

To maximize the allowable translational range, we chose one of the tracked features to be in the center of the field of view, or $\mathbf{p}_0 = [0 \quad 0]^T$. This particular feature also has the property that its x and y displacements are solely a function of unknowns δq_x and δq_y, which we can therefore compute:

$$\begin{bmatrix} \delta q_x \\ \delta q_y \end{bmatrix} = Z_0 \begin{bmatrix} -\Delta x_0 - \delta q_{\Theta_0} \\ -\Delta y_0 + \delta q_{\Phi_0} \end{bmatrix} \tag{10}$$

Using this, the remaining states δq_z and δq_Ψ can be determined from the displacements of a second feature. The second feature should be chosen as distant from the image center as possible[3] to maximize the effects that rotation and altitude change will have upon its displacements. For our derivation, we have chosen a feature $\mathbf{p}_1 = [0 \; d]^T$ in order to decouple the effects and simplify the results. Solving, we obtain:

$$\begin{bmatrix} \delta q_z \\ \delta q_\Psi \end{bmatrix} = \frac{1}{d} \begin{bmatrix} \delta q_y + Z_1(\Delta y_1 - \delta q_{\Phi_0} - d^2 \delta q_{\Phi_0}) \\ \delta q_x / Z_1 + (\Delta x_1 + \delta q_{\Theta_0}) \end{bmatrix} \tag{11}$$

Even without a stereo measurement system, in many cases it is possible to use this approach for station keeping. Many underwater robots can only measure one absolute range using, for instance, a point-ranging sonar. If $\frac{|Z_1 - Z_0|}{Z_0} \ll 1$ and Z_0 is assumed to be known, the range Z_1 can be approximated by Z_0. This approximation holds when the ocean floor is fairly flat and perpendicular to the camera. If actually $Z_1 = Z_0 + \epsilon$, then the error $\delta\tilde{\mathbf{q}} = \delta\mathbf{q}_{act} - \delta\mathbf{q}_{calc}$ in the calculated states will be:

$$\begin{bmatrix} \delta\tilde{q}_x \\ \delta\tilde{q}_y \\ \delta\tilde{q}_z \\ \delta\tilde{q}_\Psi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{\epsilon}{Z_0+\epsilon}\left(\delta q_z - \frac{\delta q_y}{d}\right) \\ \frac{\epsilon}{Z_0+\epsilon}\left(-\frac{\delta q_x}{Z_0 d}\right) \end{bmatrix} \tag{12}$$

The errors incurred by this assumption are proportional to the ratio of the difference in the feature ranges ϵ to the actual range Z_1. The errors also depend upon the translational state offsets.

IV. POSITION/ORIENTATION SENSING RESULTS

We have conducted several tests to verify the correctness of our visual sensing approach to station keeping. Our first set of tests demonstrate how changes in robot state cause detectable feature displacements. Figure 4 shows results for the case $\delta q_{\Phi_0} = \delta q_{\Theta_0} = 0$ and $Z_0 = Z_1 = Z$. For this case, the state offset solution simplifies to:

$$\begin{bmatrix} \delta q_x \\ \delta q_y \\ \delta q_z \\ \delta q_\Psi \end{bmatrix} = \begin{bmatrix} -Z\Delta x_0 \\ -Z\Delta y_0 \\ \frac{Z}{d}(\Delta y_1 - \Delta y_0) \\ \frac{1}{d}(\Delta x_1 - \Delta x_0) \end{bmatrix} \tag{13}$$

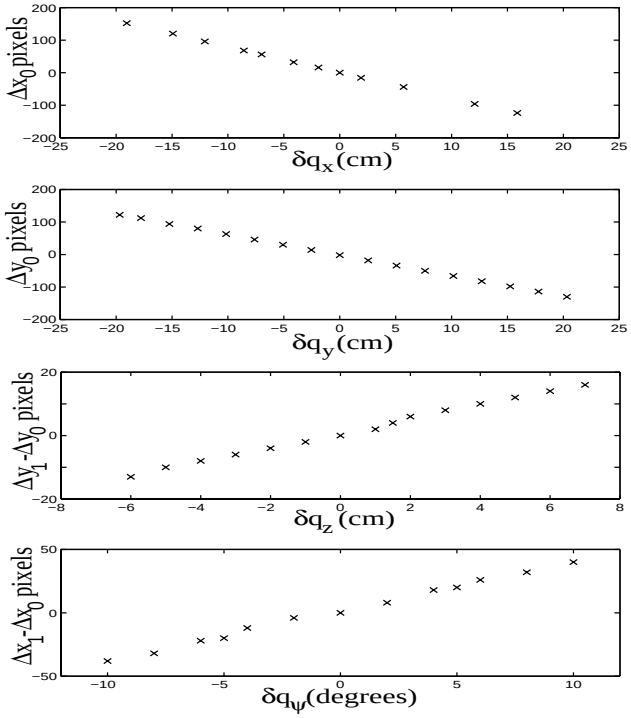

Figure 4: **Vision system measurements.**

[3] The second feature can be chosen based upon the displacement of the first feature to ensure that it has not been displaced out of the field of view.

For the results shown in Figure 4, $Z = 79$ cm, $d = 180$ pixels, $1x$ pixel $= 0.00160$ radians, and $1y$ pixel $= 0.00201$ radians (in air). Measurements were made by moving the camera and recording camera displacement along with vision system output. For our particular hardware, the feature displacement measurement resolution is two pixels in x and one pixel in y. Thus, at an altitude Z, the possible resolution in the calculations for δq_x is $0.0032Z$, for δq_y is $0.00201Z$, for δq_z is $0.0055Z$, and for δq_Ψ is 0.485 degrees. Note that the resolutions for translational motion scale with the range, but the heading resolution is absolute.

Motions of up to $\pm 0.2Z$ were conducted for δq_x and δq_y, motions of up to $\pm 0.1Z$ were conducted for δq_z, and motions of up to ± 10 degrees were conducted for δq_Ψ. The range of motion for δq_z and δq_Ψ is limited by the feature warping previously mentioned.

The effect of the quantized resolution is clearly visible in the data. However, the plots successfully show the linear relationship between changes in robot state and feature displacements.

V. STATION KEEPING EXPERIMENTS

The *OTTER* robot (Oceanographic Technology Testbed for Engineering Research) was used for experimental testing of our visual station keeping approach. *OTTER* is roughly 2 meters long, 1 meter wide, and has a dry mass of 145 kg. It is made up of three pressure housings surrounded by 8 ducted thrusters and covered by a fiberglass shell. One housing holds two independent VME card cages with 68040 single board computers for control and sensor processing. The other two hold NiCad batteries which provide approximately 750 W·hrs of power. Currently, a tether is used to trickle charge the batteries and provide serial communications at 38.4 kbaud. The sensor suite includes custom real-time vision processing boards, pitch and roll gravity sensors, a small inertial measurement unit with 3 accelerometers and 3 rate gyros, a flux-gate compass, a pressure depth sensor, and water leak detectors. Two black/white CCD video cameras are mounted as a stereo pair on a custom pan/tilt unit. Main propulsion is provided by two 2 hp brushless DC variable reluctance motors. Six 1/2 hp VR motors are used for lateral and vertical motions as well as attitude control. The real-time vision hardware that performs the digital filtering and correlation is produced by Teleos Research [16].

Testing was performed at the Naval Postgraduate School's test tank in Monterey, CA. The 20 by 20 by 7 feet tank was filled with 6 feet of fresh water. It is worth noting that because the tank is made of steel plates, *OTTER*'s fluxgate compass gave nonlinear heading output highly dependent on position within the tank.

Figure 5: **The OTTER robot.**

The station keeping tests were conducted by centering *OTTER* in the tank and selecting the *Hold Station* task. For the results in this paper, all automatic control testing was done with the robot positively buoyant, floating on the surface. Thus, only disturbances in X, Y, and Ψ required active control. Disturbances were generated by pushing *OTTER* with a pole in order to displace it from station.

Errors in the robot state were calculated from two image feature displacements. The measurements from *OTTER*'s pitch and roll sensors were used for δq_{Φ_0} and δq_{Θ_0} in Equations (10) and (11). Simple digital lead control laws were used to calculate robot thrust in order to zero the state errors.

Figure 6 shows calculated state errors and control efforts for several disturbances. Three disturbances were applied to displace the robot independently in the X, Y, and Ψ directions. The plots of commanded control effort demonstrate how the robot actively countered the disturbances in order to hold station. The calculated state error plots show *OTTER*'s time response in each degree of freedom.

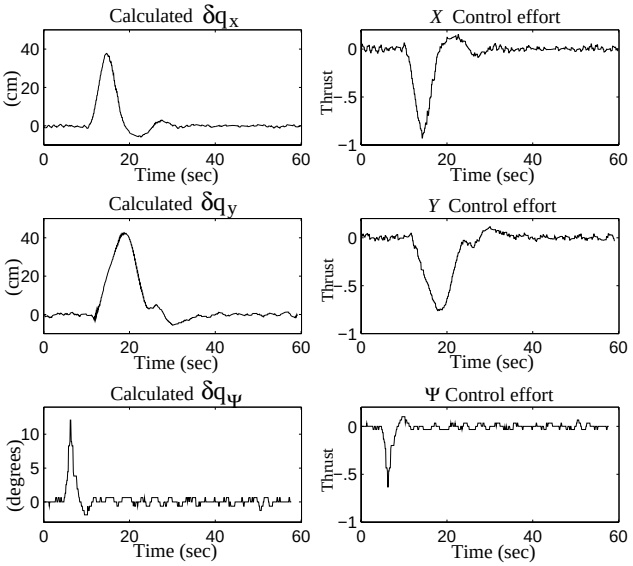

Figure 6: **Station keeping measurements.**

Unfortunately, no reliable external measurements of robot state were available as a reference for comparison. Future experiments will use the SHARPS transponder system to obtain such a reference.

OTTER was able to hold station without drift indefinitely using our station keeping approach. Without active station keeping, the pull of the tether would consistently drag *OTTER* to one side of the tank in a matter of seconds. Thus, we have successfully demonstrated an approach for holding an underwater robot's station relative to the local environment, despite the presence of disturbing forces.

VI. CONCLUSIONS

We have presented an approach for *drift-free* station keeping of an underwater robot. The approach uses visual sensing to detect changes in the four degrees of freedom (translation and heading) that are inherently prone to drift in typical underwater robots. Changes in the robot state are determined from relative image feature displacement measurements.

Station drift is avoided by makingd feature displacement measurements with respect to a fixed initial image. This sensing approach is made viable when incorporated with an active control system that regulates the robot state to zero the station error. The control system keeps the errors in station small and thereby ensures that the difference between the live camera image and the fixed initial image is small enough so feature tracking can be performed.

We have *experimentally* implemented this approach successfully to achieve drift-free station keeping of the underwater robot *OTTER*. We have presented results showing the outputs of the sensing system for changes in robot translation and rotation, as well as results from actual tank tests demonstrating successful robot station keeping. Future tests are planned for station keeping in a genuine marine environment.

ACKNOWLEDGEMENTS

This paper was supported in part by the National Science Foundation under Grant BCS-9306252 to the Naval Postgraduate School, Monterey, CA. The authors gratefully acknowledge the Naval Postgraduate School for the use of their test tank facility.

REFERENCES

[1] M. Barth, H. Ishiguro, and S. Tsuji. Computationally inexpensive egomotion determination for a mobile robot using an active camera. In *Proc. IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 2792–2797, 1991.

[2] J. Bellingham. AUV at MIT. In *Proc. of IARP Second Workshop on Mobile Robots for Subsea Environments*, 1994.

[3] F. Chenavier and J.L. Crowley. Position estimation for a mobile robot using vision and odometry. In *Proc. IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 2588–2593, 1992.

[4] A.J. Healey et. al. Tactical/execution level coordination for hover control of the NPS AUV II using onboard sonar servoing. In *Proceedings of IEEE Symposium on Autonomous Underwater Vehicle Technology*, 1994.

[5] J.S. Fox. Stationkeeping using optical ranging of natural features. In *Proc. IEEE Oceans*, pages 1027–1031, 1989.

[6] R. Marks, M. Lee, and S. Rock. Visual sensing for control of an underwater robotic vehicle. In *Proceedings of IARP Second Workshop on Mobile Robots for Subsea Environments*, 1994.

[7] R. Marks, S. Rock, and M. Lee. Automatic object tracking for an unmanned underwater vehicle using real-time image filtering and correlation. In *Proceedings of IEEE Systems, Man, and Cybernetics*, 1993.

[8] R. Marks, S. Rock, and M. Lee. Real-time video mosaicking of the ocean floor. In *Proceedings of IEEE Symposium on Autonomous Underwater Vehicle Technology*, 1994.

[9] L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *Int. Journal of Computer Vision*, 3:209–236, 1989.

[10] S. Negahdaripour and J. Fox. Undersea optical stationkeeping. Improved methods. *Journal of Robotic Systems*, 8(3):319–338, 1991.

[11] S. Negahdaripour, A. Shokrollahi, J. Fox, and S. Arora. Improved methods for undersea optical stationkeeping. In *IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 2752–2758, 1991.

[12] S. Negahdaripour, A. Shokrollahi, and C. Yu. Optical sensing for undersea robotic vehicles. In *Second Int. Conf. on Intelligent Autonomous Systems*, volume 7, 1991.

[13] S. Negahdaripour, A. Shokrollahi, and C.H. Yu. Passive vision sensing techniques for autonomous undersea vehicles. In *Proc. of the Second Int. Conf. on Intelligent Autonomous Systems*, 1989.

[14] S. Negahdaripour and C. Yu. Passive optical sensing for near-bottom stationkeeping. In *Proc. IEEE Oceans*, pages 82–87, 1990.

[15] S. Negahdaripour, C.H. Yu, and A. Shokrollahi. Recovering shape and motion from undersea images. *IEEE Journal of Oceanic Engineering*, 15(3):189–198, 1990.

[16] H.K. Nishihara. Prism: a practical realtime imaging stereo matcher. In *SPIE*, pages 134–142, 1983.

[17] H.K. Nishihara. Practical realtime imaging stereo matcher. *Optical Engineering*, 23(5):536–545, 1984. Also in *Readings in Computer Vision: issues, problems, principles, and paradigms.*

[18] J. Shi and C. Tomasi. Direction of heading from image deformations. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1994.

[19] J. Shi and C. Tomasi. Good features to track. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1994.

[20] C.E. Thorpe. *FIDO: vision and navigation for a robot rover.* PhD thesis, CMU, 1984.

[21] Q. Zheng and R. Chellappa. Automatic feature point extraction and tracking in image sequences for unknown camera motion. In *Proc. IEEE 4th Int. Conf. on Computer Vision*, pages 335–339, 1993.

REFERENCES