A 2-D visual servoing for underwater vehicle station keeping

J.-F. Lots, D. M. Lane, E. Trucco *
F. Chaumette †

Abstract

This paper introduces a 2-D visual servoing technique for the station keeping of an unmanned underwater vehicle (UUV) with respect to planar targets on the sea bed. The underwater vehicle is subject to sea current disturbances which make it drift from its desired position. Feature points from unmarked objects are extracted and tracked with a sparse feature tracker developed in-house [1]. The scene depth is estimated from a planar homography. To validate our approach, we emulate the dynamics of the surge and sway degrees-of-freedom (d.o.f) of an UUV model with a planar Cartesian robot in our water test tank. Successful station keeping experiments obtained with a range of sea current disturbances are presented.

1 Introduction

So far, in the underwater robotics field few attempts have been made to use vision sensors for control [2, 3, 4, 5, 6]. However, vision shows some interesting features compared to classical positioning sensors to perform station keeping tasks. For example, magnetic compasses suffer from a slow update rate and cannot be used in the vicinity of man-made metallic structures. More importantly, with the exception of depth sensors which are both accurate and fast, on-board translational motion sensors (for surge and sway) are integrating sensors (i.e. accelerometers, Doppler velocity logs) hence subject to drift, and therefore unsuitable for station keeping.

A standard camera, however, is not subject to magnetic influences, and has an update rate of 25 Hz. It can also be used as a local absolute positioning sensor: indeed, comparing a current image to the first one yields a position measurement whose drift can be controlled.

Despite its short range (typically 3-10 metres) and the need for heavy computing power, visual control (also called visual servoing) allows very diverse tasks such as for example station keeping [2,4] or pipe-following [3] to be carried out. Indeed, in [3], Rives and Borelly used the task function approach [7] to perform pipe-following with the Ifremer Vortex ROV in their swimming pool. Marks et al. [2] used a stereo camera and Laplacian of Gaussian filtering with dedicated hardware to solve the station keeping problem. Negahdaripour et al. [4] recovered 3-D motion from an optical flow based approach.

In a previous paper [6], we tackled the station keeping problem with a modification of the 2 1/2 D visual servoing method from Malis and Chaumette [8]. We proved in simulation its adequacy to underwater vehicle station keeping for
ified in a test tank with a planar motion mechanism [9]. For the purpose of this paper, we concerned ourselves with the dynamics of its surge and sway axis. Let the body-fixed vehicle’s velocity vector be defined as \( \nu = [u, v]^T \), and the Earth-fixed velocity vector as \( \hat{\nu} = [\dot{x}, \dot{y}]^T \). We assume that the vehicle’s heading angle \( \psi \) can remain constant and close to zero (using an independent control loop for example [6]).

In addition, since ANGUS was designed to be stable in roll \( \theta \) and pitch \( \phi \), these angles remain also very small. As a result, the \( 2 \times 2 \) Jacobian matrix relating the body-fixed velocity vector \( \nu \) to the Earth-fixed velocity vector \( \hat{\nu} \) can be assumed to be the \( 2 \times 2 \) identity matrix. In the following, we will therefore no longer make the distinction between Earth-fixed position and velocity, and body-fixed ones. The motion equations for the ROV can thus be written as:

\[
\begin{align*}
M_{11} \ddot{u} &= B_{11}(u - u_c)(|u - u_c| + D_u) + a_1\beta + a_2|\beta| \\
M_{22} \ddot{v} &= B_{22}(v - v_c)(|v - v_c| + D_v) + a_3\gamma
\end{align*}
\]

where \( M_{ii} (i = 1, 2) \) are the mass matrix coefficients, the \( B_{ii} (i = 1, 2) \), \( D_u \) and \( D_v \) the hydrodynamic drag coefficients, \( u_c \) and \( v_c \) are the velocity of the sea current expressed in the body-fixed frame. The normalised control input of the two back thrusters is \( \beta \in [-1, 1] \), \( \gamma \in [-1, 1] \) is the normalised control input of the sway thrusters, and \( a_i \) \( (i = 1, 2, 3) \) are the thrusters’ efficiency coefficients. Note that the back thrusters are less efficient when operated in reverse.

The numerical values of the parameters are gathered in Table 1.

To assess the performances of the proposed visual servoing technique, we compared it to a "ground truth" PID controller assuming perfect position and velocity measurements at the same sampling rate as the visual servoing experiments, i.e. \( T \approx 200 \text{ ms} \). We have also included the time delay of one sampling period caused by the visual processing. The PID control law is:

\[
\begin{align*}
K_p &= \text{diag}(1.0, 4.0) \\
K_d &= \text{diag}(0.4, 0.4) \\
K_i &= \text{diag}(0.011, 0.02)
\end{align*}
\]

Since surge and sway are decoupled, we tuned the parameters of each axis independently. We tried to obtain a time response similar to a second order critically damped system.

In Figure 1, one can note the slow dynamic response of the ROV subject to a sea current velocity step input disturbance \( \nu_{c} = (-0.2, -0.5) \text{ m/s} \). The settling times, corresponding to a positioning error of less than one centimeter, of the surge and sway axis are \( t_{\text{surge}} = 80 \text{ s} \) and \( t_{\text{sway}} = 95 \text{ s} \). Figure 2 shows the corresponding thrusters’ values of the 2 d.o.f. model of ANGUS subject to a sea current disturbance \( \nu_{c} = (-0.2, -0.5) \text{ m/s} \) with a PID controller.
values. These numbers will serve as a comparison basis for the visual servoing experiments.

In the following sections, we present an image-based visual servoing approach to solve the station keeping problem. It is a simple 2-D visual servoing scheme [10] using the centre of gravity of the set of extracted features in the images as the visual feature. Since the camera is rigidly mounted onto the robot, the frame transformation between the robot frame and the camera frame is therefore constant. To simplify the expression of the following equations, we assumed that the camera reference frame and the robot frame coincided. In other words, controlling the robot is equivalent to controlling the camera whose motion dynamics are those of an underwater vehicle.

<table>
<thead>
<tr>
<th>$M_{11}$</th>
<th>$M_{22}$</th>
<th>$B_{11}$</th>
<th>$B_{22}$</th>
<th>$D_u$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1800</td>
<td>2200</td>
<td>-350</td>
<td>-680</td>
<td>0.6</td>
</tr>
</tbody>
</table>

2.3 2-D visual servoing

A visual servoing positioning task can be expressed as the regulation to zero of a task function $e(s, t)$, where $s$ is a visual feature (in our case, points in the image), and $t$ is the time [7]. If $T_c$ is the camera velocity screw (6-vector), and $L(s, t)$ is an appropriate matrix, called interaction matrix, then the camera velocity screw is related to the task function $e$ by:

$$
\dot{e} = L \cdot T_c
$$

In the case of 2-D visual servoing, if we consider a feature point $s_i = [x_i, y_i]^T$ expressed in the image ($Z_i$ being the depth of the corresponding 3-D point projected on the image point $s_i$), $L$ is made up by stacking the following lines (two lines for each feature point):
where \( K_p, K_d, \) and \( K_i \) are 2 diagonal matrices with the proportional, derivative and integration control gains of the PID, and \( Z_g \) is the depth of the centre of gravity in the current image, estimated from eq. (3) (see section 2.2).

### 3 Results

#### 3.1 Experimental setup

To experimentally validate our method, we used a planar Cartesian robot in our 4 m \( \times \) 3 m \( \times \) 2 m test tank. This robot was servoed by a DMC-1360 motion control card from Galil Motion Control Inc. hosted on a VME crate. The VME crate was composed of a Unix Host (Motorola MVME-167 board) and three real-time target boards (Motorola MVME-162-22) running Motorola PSOS real-time operating system. We used one of these MVME-162-22 boards to drive the motion control card and ran the dynamic model of ANGUS.

An underwater black and white camera was rigidly mounted on the Cartesian robot. The camera was pointing downward about one metre from the bottom of the tank, imaging a surface of about one metre squared. Its analogue signal was digitised by a Brooktree B254 frame grabber hosted on a AD164R parallel processing board from Alpha Data Ltd. The latter was itself hosted by a Linux PC communicating via TCP/IP to the MVME-162-22 board. The visual servoing algorithms (feature extraction and tracking included) ran on the AD164R; the new thrusters values were sent via TCP/IP to the real-time target board. With this setup, the servoing ran at 5 Hz.

#### 3.2 Visual station keeping experiments

Each station keeping experiments followed the same procedure. The robot was initially immobile and the desired set of \( \text{diag}(0.01, 0.01) \). The estimated distance to the target \( Z^* \) was roughly measured and set to 1.1 m.

The final positioning errors of the robot are collected in table 2. In each case, the station keeping was successfully performed to within a good positioning accuracy (the maximum error observed was of a few centimeters).

<table>
<thead>
<tr>
<th>Sea current velocity ((u_c, v_c)) [m/s]</th>
<th>(\Delta X) [mm]</th>
<th>(\Delta Y) [mm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>((-0.1, -0.1))</td>
<td>+0.95</td>
<td>+5.60</td>
</tr>
<tr>
<td>((-0.2, -0.1))</td>
<td>+0.00</td>
<td>+8.80</td>
</tr>
<tr>
<td>((-0.3, -0.1))</td>
<td>-0.70</td>
<td>+7.80</td>
</tr>
<tr>
<td>((-0.4, -0.1))</td>
<td>-1.20</td>
<td>+6.65</td>
</tr>
<tr>
<td>((-0.5, -0.1))</td>
<td>-10.30</td>
<td>+3.65</td>
</tr>
<tr>
<td>((-0.1, -0.2))</td>
<td>+2.95</td>
<td>-2.95</td>
</tr>
<tr>
<td>((-0.2, -0.2))</td>
<td>-8.65</td>
<td>+0.30</td>
</tr>
<tr>
<td>((-0.3, -0.2))</td>
<td>-1.70</td>
<td>-1.25</td>
</tr>
<tr>
<td>((-0.4, -0.2))</td>
<td>-4.90</td>
<td>-2.25</td>
</tr>
<tr>
<td>((-0.5, -0.2))</td>
<td>-33.65</td>
<td>-14.55</td>
</tr>
</tbody>
</table>

Table 2: Final positioning errors of the robot subject to a range of sea current disturbances.

Figure 3 shows the positioning error of the robot during a station keeping experiment with sea current velocity \((u_c, v_c) = (-0.5, -0.2)\) m/s. The robot successfully stabilizes itself within 1 cm accuracy with a settling time of 130 s in surge and 290 s for sway. These times are greater than for the ideal controller of section 2.1. This is due to the added dynamics of the Cartesian robot onto ANGUS', which is more pronounced at low speeds. However, the trajectory followed by the robot is similar to those obtained with our ideal controller.

The thrusters’ values time history is shown in figure 4. Here again, we note a behaviour similar to the one of the ANGUS' controller.
Figure 3: Visual station keeping experiment with sea current \((u_c, v_c) = (-0.5, -0.1)\) m/s: Cartesian robot position.

Figure 4: Visual station keeping experiment with sea current \((u_c, v_c) = (-0.5, -0.1)\) m/s: thrusters applied to the robot model.

Figure 5: Desired image (left), and final image with pixel tracks (right) obtained with the visual station keeping experiment: sea current \((u_c, v_c) = (-0.5, -0.1)\) m/s and \(v_c = -0.2\) m/s being the maximum admissible values.

The limitations in sea current velocity magnitude are due to several factors. One is the limited thrusters' power available for ANGUS which sets a maximum sea current velocity that the robot can counteract. The slow dynamics of the vehicle are also an issue. Indeed, too strong a sea current will cause the robot to drift so far away that no feature from the desired set would remain in the image. A solution to that problem would be to re-extract features during the servoing. However, in that case, the robot would drift unless the initial features are maintained in memory and recognised by data association.

A more serious limitation in our implementation is the 200 ms time delay incurred by the slow TCP/IP communication and the feature tracker running time (150 ms on 512 \(\times\) 512 images) which decrease the control stability margins. Tests in simulation without time delay clearly demonstrated this downside effect.

Lastly, some of the performances limitations are caused by the feature tracking. The feature tracker relies on the assumption of small pixel motions in the image. Sampling the video signal at 5 Hz clearly reduces the working conditions. We carried out some tests to assess quantitatively the maximum admissible pixel displacement between two frames and it proved to be 5 pixels. However, 512 \(\times\) 512 pixels images represents an important amount of data for tracking purposes, and explain the relatively slow tracking rate. Image sub-sampling would certainly increase the tracking speed to the cost of a reduced positioning accuracy. Abrupt changes of motion direction and changes in lighting, may also cause the tracker to lose features. Further experiments will be needed to assess quantitatively the effect of illumination changes on the tracking behaviour. However, we found that, with sufficient and constant illumination (4
lux), tracked features were successfully positioned with a repeatability of 0.5 pixels standard deviation.

5 Conclusion and Future work

This work showed the validity of 2-D visual servoing for underwater vehicle station keeping. This approach was robust to various sea current disturbances and allows good positioning precisions. The use of a linear PID controller permitted slow, but very stable hover capabilities. Besides, we were able to perform these experiments on a number of unmarked planar targets without any changes in the feature tracking parameters, which demonstrated the robustness of the visual processing part.

Future work will concentrate on transferring this visual servoing algorithm on one of our small ROVs, namely RAUVER built in-house. In addition, more characterisation will be done under various conditions of visibility and lighting since these are major issues underwater. Nonlinear controllers, making use of the vehicle’s dynamics would also improve the disturbance rejection, and will be looked into.

6 Acknowledgements

This work was partially funded by EU-MAST project MAS3-CT97-0083 ARAMIS ("Advanced ROV Package for Automatic Investigation of Sediments").

References


