# Visual Control of an AUV for Multi-Robot Intervention Tasks

Emilio Garcia-Fidalgo, Alberto Ortiz, Miquel Massot-Campos

Systems, Robotics and Vision Group
Department of Mathematics and Computer Science
University of the Balearic Islands, 07122 Palma de Mallorca, Spain
Email: {emilio.garcia, alberto.ortiz, miquel.massot}@uib.es

## I. Introduction

In the last decades, robots have been used to explore areas hard to reach for humans. Underwater environments fall into this category, since their operating conditions make even simpler operations risky to be carried out by divers, especially when they have to be performed at high depth. A possible approach to overcome this problem is to use a Remotely Operated Vehicle (ROV), although this easily becomes a difficult and expensive solution because it usually requires a sophisticated support infrastructure and specialized staff. In this regard, the project MERBOTS proposes a new robot-based methodology to make intervention tasks safer, simpler and at a lower cost. The proposed system involves two vehicles, being one of them a Hybrid ROV (H-ROV) equipped with an arm and a manipulator, as well as the necessary perception devices, which altogether implement the supervised intervention task, while the other is an Autonomous Underwater Vehicle (AUV) endowed with cameras to provide alternative points of view of the target for the operator in charge of the H-ROV, enabling thus a more robust and reliable operation. In this paper, we focus on the target detection and tracking tasks to be performed to provide this secondary view, which are part of the goals addressed by the SUPERION subproject.

To this end, the object to manipulate must be ensured to appear continuously in the field of view of the camera placed in the AUV. This naturally leads to the implementation of a visual servoing task, whose input is the image stream coming from the AUV camera and its outputs are the velocity commands to be sent to the vehicle controller so as to keep the target in the field of view at all times. Due to its well-known robustness and simpler implementation, in this application, we choose an Image-Based Visual Servo-control (IBVS) approach [1]. From a global point of view, the solution comprises two interacting processes, target detection and tracking, which provide input to the visual servo control strategy. The following sections describe briefly all three aspects of our solution: target detection, target tracking and visual servo-control.

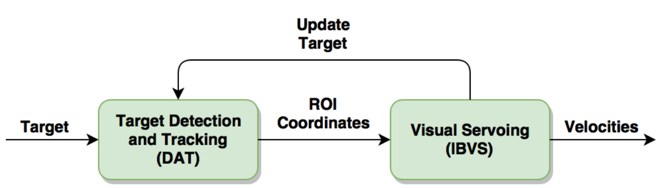


Fig. 1. Outline of the visual control approach. The DAT module is in charge of detecting the target in the current image. The detected position is then used by the IBVS module to generate the AUV control commands.

## II. Visual Control Approach

Figure 1 outlines our approach. As can be seen, the system comprises a module to detect and track the object in the image (DAT) and a module to generate the corresponding control velocities for the vehicle (IBVS). Initially, the target is selected in the current image by defining a Region of Interest (ROI). The DAT module computes then a set of SIFT keypoints [2] as the target model. This model is used to search and track the target in the image stream. The coordinates of the ROI where the target has been found are accordingly updated and sent to the IBVS module, which generates the necessary control commands that are to make the target get centred in the image. Both modules, DAT and IBVS, are detailed next.

### A. Target Detection and Tracking

As shown in Fig. 2, our strategy to estimate the position of the target in the image plane is based on two different stages, *detection* and *tracking*, which interact with one another. The *detection* stage is computationally expensive but robust to appearance changes. Conversely, the *tracking* stage is a more efficient process, but tends to lose the target from time to time. Taking into account these considerations, our strategy employs the tracking stage as much as possible and the detection stage is only used when the tracking system needs to be retrained. The system starts executing the detection stage of the DAT module. If the target is found in the current image, the corresponding bounding box is set as the ROI and used to initialize the tracking process. This stage keeps estimating the position of the target until it considers that it has lost track of it. In such a case, the detection process activates again and operates until the target is relocated.

The detection stage begins computing a set of SIFT keypoints in the current image. A collection of putative matches are found between the current image SIFT features and the target model, also consisting of a set of SIFT features. For

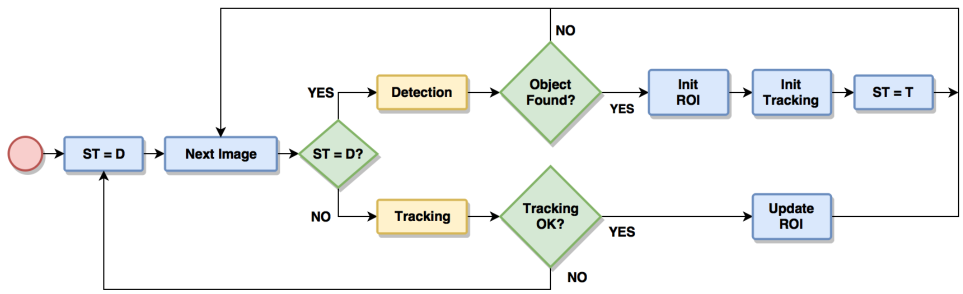


Fig. 2. Target detection and tracking. As can be seen, the strategy is based on the interaction between the detection and tracking stages. ST (*status*) flags the current operation mode.

efficiency reasons, this task is implemented using a set of randomized kd-trees and applying the nearest neighbour distance ratio test [2] to discard incorrect matches. The surviving matches are then employed to compute a homography between both descriptors. After that, if the resulting number of inliers is high enough, we consider that the target has been found and the resulting homography is used to estimate the coordinates of the target ROI corners in the current image. The minimal upright bounding box is calculated using these coordinates, and the corresponding corners used as input to the IBVS module.

For the tracking process, we have considered two well-known visual tracking algorithms, Struck [3] and KCF [4], which have correspondingly been adapted to our purposes, so that the system can make use of any of them. Nonetheless, we have empirically noted that KCF performs better in computational terms. In any case, during tracking, we compute the distance between global PHOG descriptors [5] for the target and the current ROI to determine whether the target has been lost. The detection stage becomes active again if this distance is higher than a threshold.

### B. Image-Based Visual Servoing

IBVS control operates in terms of image positions. In one of the many possible approaches, the goal is to make a set of image points (features) $s$ attain a set of desired positions $s^*$, which implicitly moves the involved platform. To this end, IBVS defines a model that relates the camera velocities $\xi_c(t)$ to the velocities of the selected features over the image plane $\dot{s}(t) = [\dot{s}_{1,x}(t), \dot{s}_{1,y}(t), \ldots, \dot{s}_{n,x}(t), \dot{s}_{n,y}(t)]^T$ through the so-called *interaction matrix* $L$ [6]. In our case, we conveniently include the transformation from robot to camera ${}^cT_r$, to obtain velocity commands in the robot frame ($\xi_r$):

$$\dot{s}(t) = L\,\xi_c(t) = L\,({}^cT_r\,\xi_r(t)) = L'\,\xi_r(t) \qquad (1)$$

Robot motion needed to move the image features to the desired image positions is then derived from (1) in the form of (2):

$$\xi_r = (L')^+\,\dot{s}(t) \qquad (2)$$

where $(L')^+$ is the pseudoinverse of $L'$ resulting from a least squares framework. For our application, the corners of the ROI detected by the DAT module are used as the features $s$, while, to set $s^*$, those corners are required to get centred in the image.

In general terms, IBVS is designed to make the current feature positions $s(t)$ coincide with the set of desired positions $s^*$, i.e. minimize the corresponding error function $e(t) = s(t) - s^*$. In our approach, we adopt a PID-like control scheme to this end, so that the final control law results to be:

$$\xi_r(t) = -(L')^+\left(\lambda_\mathrm{p}\,e(t) + \lambda_\mathrm{i}\int_0^t e(\tau)\,d\tau + \lambda_\mathrm{d}\,\frac{d\,e(t)}{dt}\right) \qquad (3)$$

being $\lambda_\mathrm{p}$, $\lambda_\mathrm{i}$ and $\lambda_\mathrm{d}$ the, respectively, proportional, integral and derivative gains of the controller. This control scheme is replicated for each degree of freedom (d.o.f) of the AUV, adopting an uncoupled control solution, so that different gain values result for each d.o.f.

As previously said, in this work, we make use of the ROI corners as image features, which have to be properly tracked to correctly compute the error function $e(t)$ required by (3). Additionally, the appearance of the target is updated during the intervention to improve the performance of the tracking module; the update takes place whenever the norm of $e(t)$ is low enough (see Fig. 1).

## III. Experimental Results

For validation purposes, a first series of field trials in a water tank at the Research Center in Underwater Robotics (CIRS, UdG) and at sea (Sant Feliu de Guíxols, Girona) have been recently performed by the MERBOTS consortium. Those experiments involved the Sparus II platform [8] as the AUV, fitted with a lateral thruster for sway motion. Videos for these experiments are available at *http://srv.uib.es/automar17*.

## References

[1] F. Chaumette and S. Hutchinson, "Visual Servo Control. Part I: Basic Approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, 2006.
[2] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[3] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr, "Struck: Structured Output Tracking with Kernels," *IEEE T. Pattern Analysis*, vol. 38, no. 10, pp. 2096–2109, 2016.
[4] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-Speed Tracking with Kernelized Correlation Filters," *IEEE T. Pattern Analysis*, vol. 37, no. 3, pp. 583–596, 2015.
[5] A. Bosch, A. Zisserman, and X. Munoz, "Representing Shape with a Spatial Pyramid Kernel," *IET Image Process.*, vol. 5, no. 2, pp. 401–408, 2007.
[6] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer, 2011.
[7] D. Ribas, N. Palomeras, P. Ridao, M. Carreras, and A. Mallios, "Girona 500 auv: From survey to intervention," *IEEE/ASME Transactions on Mechatronics*, vol. 17, no. 1, pp. 46–53, Feb 2012.
[8] M. Carreras, C. Candela, D. Ribas, A. Mallios, L. Magí, E. Vidal, N. Palomeras, and P. Ridao, "Sparus II, Design of a Lightweight Hovering AUV," in *MARTECH*, vol. 911, 2013, pp. 163–164.