Increasing Autonomy within Underwater Intervention Scenarios: The User Interface Approach

J. C. García, J. J. Fernández, P. J. Sanz, R. Marín
UJI; Castellon, Spain

Abstract— **The present work represents working progress for designing a Graphical User Interface (GUI) within an ongoing research project named RAUVI (e.g. Reconfigurable AUV for Intervention Missions). This GUI should help the user to identify the target using images compiled by the I-AUV through a previous survey stage. After that, the user is able to specify the most suitable intervention task selected among a set of predefined ones. Thus, a very intuitive and *user-friendly* interface has been designed, enabling a non qualified user to succeed in the specification of an intervention mission. Furthermore, some implementation details and their performance about different facilities integrated within this GUI to assist the user in the required specification of underwater intervention missions will be addressed.**

Keywords— Graphical User Interface, Autonomous Underwater Vehicle for Intervention (I-AUV), Autonomous Underwater Vehicle (AUV), underwater task specification, user assistance, object recovery.

I. THE AIM OF THE PROJECT

Traditionally, underwater intervention tasks have been made with "manned submersibles" and "Remotely Operated Vehicles" (ROVs), both equipped with one or more robotic arms to carry out the interventions. In both cases, the control of the robotic arms presents a master/slave configuration, in which the user must continuously control them during the intervention. The main drawbacks of this methodology are fatigue of the user during long-term interventions, the high skills required by the users in order to control the arm and the need of permanent user-arm communication for suitable control.

To overcome these drawbacks ROVs systems have been designed as the evolution of "manned submersibles", removing the user inside the vehicle. Besides this, recently appeared the ROVs evolution, called "Intervention Autonomous Underwater Vehicles" (I-AUVs), with the aim to remove the user from the intervention control loop and resolve the communication constraint. With I-AUVs, researchers face the challenge of increasing the autonomy of them with skills to take part in unstructured underwater scenarios without user intervention. Actually, this problem is not solved yet.

The main goal of the RAUVI project [1] is to develop and improve the necessary technologies that will allow us to autonomously perform an intervention mission in underwater environments. The GUI module will play an important role and will make possible a friendly system's integration. A Graphical User Interface will be implemented and linked with the AUV-Manipulator control architecture, in order to (1) specify survey missions, (2) evaluate and display the data logged during the survey task, and (3) specify intervention missions.

In the context of this proposal, a generic intervention mission will be carried out in two stages (using the GUI), each one of them composed of several phases (see "Fig.1"):

1. Survey Stage
   - 1.1. Phase I: Launching
   - 1.2. Phase II: Survey
   - 1.3. Phase III: Recovery
2. Intervention Stage
   - 2.1. Phase IV: Intervention Specification (user)
   - 2.2. Phase V: Launching
   - 2.3. Phase VI: Approaching
   - 2.4. Phase VII: Intervention
   - 2.5. Phase VIII: Recovery

As it can be observed in "Fig.1", the proposed GUI design enables the user to specify an intervention task, among a set of predefined subtasks. This subtask will be undertaken with a particular target object selected by the user, by means of the map previously build. Hence the intervention task is seen as a semiautomatic process where the target is manually selected but then it is automatically recognized and manipulated by the robot in a complete autonomous way.

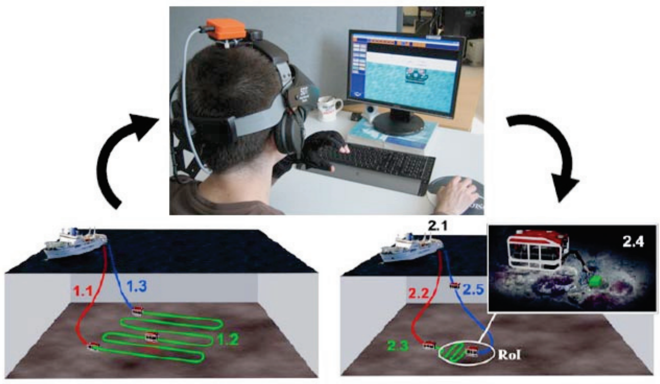

Figure 1. The envisioned I-AUV system to develop in the RAUVI project.

II. STATE-OF-THE-ART

First works concerning I-AUVs were published in the former 90's. They addressed the coordinated control of the vehicle-manipulator system. Most of these pioneering works relied on numerical simulations of the coupled dynamics of both systems. First attempts to achieve an AUV endowed with a manipulator entailed to the development of the ODIN AUV [2] (University of Hawaii), the OTTER AUV (MBARI) [3] and the VORTEX/PA10 robot within the UNION European project [4]. Although these vehicles represented a step forward in I-AUV technology they were mainly used as research testbeds, always working in water tank conditions.

During the mid 90s, AMADEUS EU project [5] supposed a step forward in the field of dexterous underwater manipulation. It included within its objectives the realization of a set-up composed by two 7-DOF ANSALDO manipulators to be used in cooperative mode. After this period, researchers proposed new concepts to avoid the complexity of the coupled motion of the vehicle-manipulator system in order to achieve true field operations in open sea conditions. In 2001, Cybernetix tested its hybrid AUV/ROV prototype with the SWIMMER project [6]. In this case, an autonomous shuttle (an AUV) carrying a ROV, is launched from a support vessel to autonomously navigate and dock into an underwater docking station in an offshore infrastructure. The docking station provides a connection to the AUV and from it to the ROV. This allows carrying out a standard ROV operation without the need of a heavy umbilical. After SWIMMER, two more projects were launched, ALIVE (EU) [7] and SAUVIM (USA) [8]. ALIVE is a 4 DOF I- AUV with a 7 DOF manipulator which has shown its capability of autonomous navigation towards a position nearby an underwater intervention panel, detect the panel using an imaging sonar and finally, approximating and docking to the panel with the help of a vision system and two hydraulic grabs. Once the AUV is grabbed to the panel, and assuming the panel is known, the manipulation is a simple task. ALIVE's project was complemented with the European Research and Training (RTN) network FREESUB devoted to the fundamental research on areas like the Navigation, Guidance, Control, Tele-Manipulation and Docking needed to further develop the I-AUVs.

Nowadays, maybe the most advanced I-AUV may be is the SAUVIM project, which is an AUV with a 7 DOF electrical driven arm (ANSALDO), the same used in the AMADEUS EU project, which is intended to recover objects from the seafloor using dexterous manipulation. The evolution of SAUVIM [9] introduces a different concept in order to release the necessity of the constant presence of the communication link. This minimizes human involvement in the control of the underwater vehicle and its manipulation tasks. In this proposal approach, the most noticeable aspect is the increasing amount of information exchanged between the system and the human supervisor. The user will use the Sauvim Programming Language (SPL) to write in a console a higher-level information (e.g. ''unplug the connector'') for a particular mission. This SPL will transfer to the system, the responsibility of all the proper control issues and interaction with the environment for proper executing of the task.

III. The User Interface

In order to improve the RAUVI's performance in general and, respecting the intervention capabilities in particular, the design of a suitable user interface represents, without doubt, a critical factor. In fact, the RAUVI project has focused on the implementation of an efficient GUI, to enhance the way the system can be controlled when specifying a robot task. If we consider the possibility that the user could be non-high qualified, a very intuitive and *user-friendly* GUI must be developed. Additionally, the GUI offers to the user the possibility to modify the input (e.g. an image), applying some of the most common image processing algorithms (e.g. *erode, dilate, Canny, Hough*). A brief description of the GUI will be given in the following.

a) The target identification and selection.

Once the robot finishes the Survey stage, it will download all the captured images into a server. Due to the differences between the scenarios where the robot could work, is very difficult to define a general description of the images. This fact increases the overall difficulty, because in each scenario a specific identification algorithm will be necessary.

Furthermore, the generic segmentation process is an unsolved problem. Environmental conditions for underwater robotics are usually very hostile, including poor visibility. Taking these constraints into account, we could consider that a fully automatic segmentation process is out of the scope of this work.

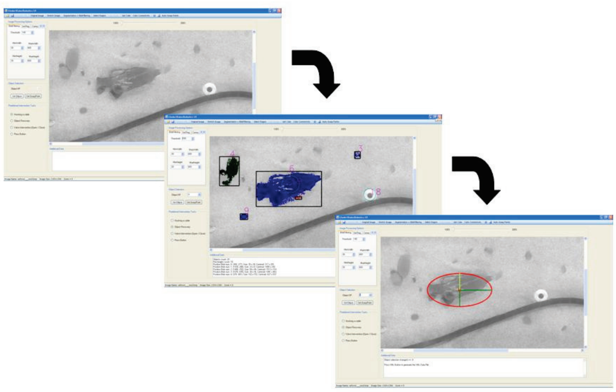

Figure 2. a) Initial input. b) Objects identification. c) Target identification & selection.

Bearing this in mind, the first step in the target identification is the image segmentation. Despite the high number of segmentation algorithms available, we decided to use the binary threshold. This method is very simple, but is still quiet efficient in most scenarios (see "Fig. 2a").

Once the scenario is segmented, blob detection and connected components labelling algorithms is applied (see "Fig. 2b"). The user can filter the detected blobs, using the blob's length and width. Thus will reduce the number of blobs showed in the image. These values can be changed by the user in the first tab "of the Image Processing Options".

Finally, the user selects the target identifier from a contextual list, which contains all the detected and filtered blobs. In this moment, an ellipse will be drawn on the target (see “Fig. 2c”).

b) The intervention specification.

When the target has been identified, a suitable intervention must be specified by the user. This intervention could be retrieved from a knowledge database, which contains the different available interventions (e.g. grasping an object, press a button from a panel, hooking a cable or take a ground sample). Some of these possible interventions are shown in “Fig. 3”.

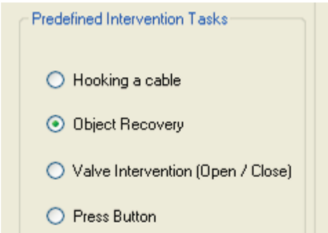

Figure 3. Predefined Intervention Tasks.

Currently the implementation is focused on the “Object Recovery” intervention task, which is actually the one that will be described in this paper. For that, we must calculate the grasping points for a given object, in order to perform the robot task in a reliable way.

According to [12], we can identify the object using its geometric properties. For global shape characterization, the system uses the centroid, its orientation, and the inertia axes (see “Fig. 4”). These features are calculated using the boundary box and the best fitting ellipse, which are drawn on the object (see “Fig. 5b-c”). The reason of using a best fitting ellipse is because the ellipse captures the size and shape of the object.

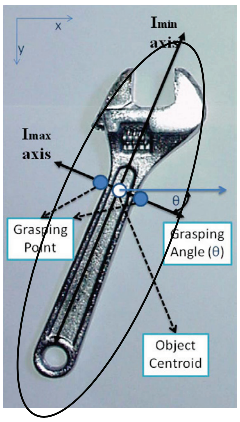

Figure 4. Grasping Points Theory.

Following the grasping determination in [12], we need to choose grasping points that (1) maximize the surface of contact between fingers and object, (2) the grasping line should lie inside the friction cones, and (3) the grasping line is close to the object centroid to minimize the gravity effect. So, the best points, which satisfy these conditions, are represented by the intersection points between the ellipse minimal axis (or Inertia Maximal Axis, I_{max} *axis*) and the object edge. All these concepts are shown in “Fig. 4”.

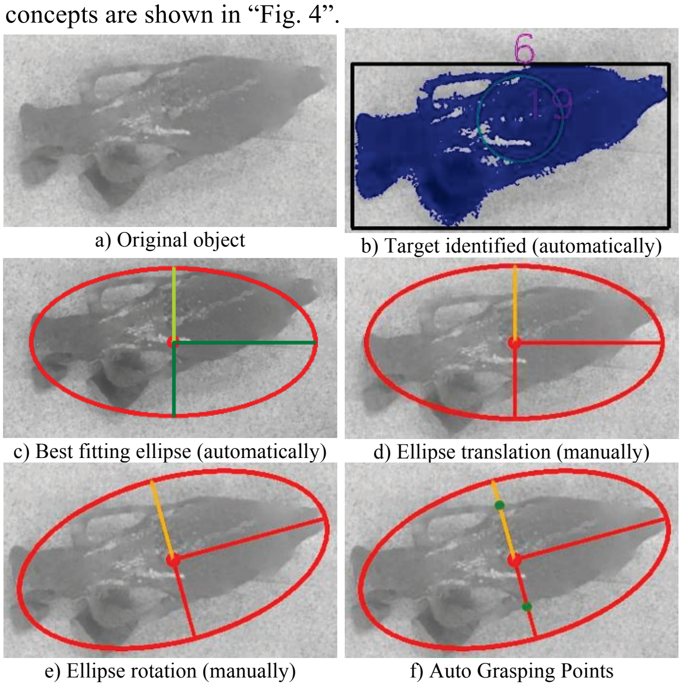

Figure 5. Target identification and auto grasping points selection.

Now, focusing on the GUI, the user can identify and select the target using the mouse, only with a few clicks. An example of this process can be seen on “Figure 5”. Once the object is detected by the program (see “Fig. 5b”), the user has to select the object identifier from a contextual list. Then, automatically the best fitting ellipse possible will appear drawn on the object (see “Fig. 5c”). If this ellipse does not fit correctly, the user must modify it, using the translation and rotation button (see “Fig. 5d-e”). Finally, the user can draw by itself the grasping points or can find the grasping points automatically using the *AutoGraspingPoints* button. When the action is performed, the cross point between the ellipse minimal axis and the object edge is marked and stored as grasping point (see “Fig. 5f”).

c) Three interaction modes to use the GUI.

The user has three different ways to use the GUI, depending on the image quality and the way the user wants to be involved in the final decision. We have noticed that a user action is always required in order to select the target, specify the task and validate it (using a simulator). A general description of the GUI functionality is shown in “Fig. 6”.

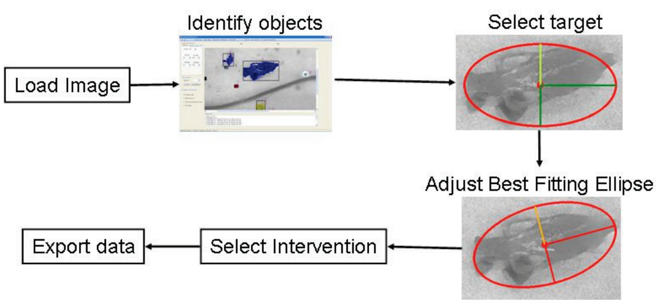

Figure 6. general GUI algorithm.

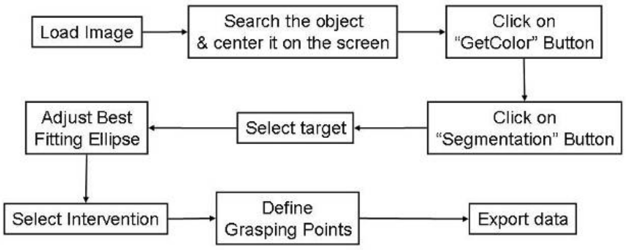

Figure 7. User interface, in Mode-1.

In Mode-1 (see "Fig. 7"), the user clicks on the interested target to get its colour in greyscale, which will be used as starting threshold value to segment the scene. Then, blob detection and size filtering process will be applied on the image, just clicking on the button *Segmentation + BlobFiltering*. The blob size filtering thresholds must be specified by the user using the left panel.

Once the process has finished, the blobs are identified and the user has selected the target. The best-fitting ellipse (see Sub-section b below) will be drawn on the target and relative information relating to the object will be displayed in the *Additional Info* panel. If this ellipse does not fit perfectly, the user can correct it accordingly. Then, the user can manually define the grasping points or, if suitable, an automatic process is carried out. Finally, the user selects the action to perform and export all the data in a XML file. This file contains the image source path (to be used by the simulator), the target properties (width, length and centroid position), the grasping points' properties, and the task that should be done by the robot.

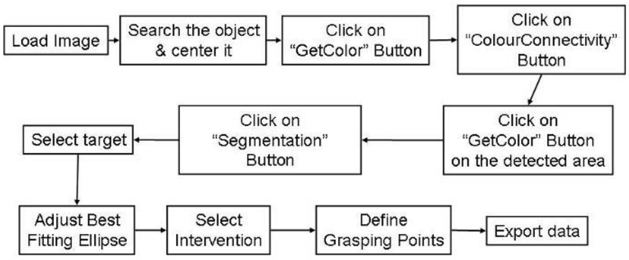

Figure 8. User interface, in Mode-2.

In Mode-2 (see "Fig. 8"), the user will use a colour connectivity algorithm to get all the colour pixels (the value in RGB) which are close to the point selected by the user. This is made just by clicking on *GetColour* button, to select the initial point, and then the *ColourConnectivity* button. A real example of this mode is shown in "Fig. 10".

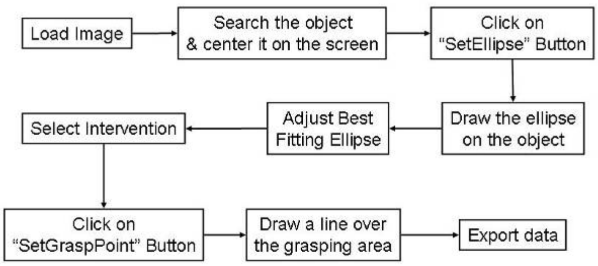

Figure 9. User interface, in Mode-3.

Finally, in Mode-3 (see "Fig. 9"), the user will draw manually on the image an ellipse which fits on the target. In order to do so, the object should be centred in the screen. Then, the user clicks on the *SetEllipse* button and draws it. If this ellipse does not fit perfectly, the user can modify it, using the rotation and translation buttons. Then, the user finishes the intervention specification like in the other modes.

Depending on the image properties (usually the contrast), the button *Segmentation + BlobFiltering* could not result so effective and could not get all the interesting blobs. In this situation, the user can crop the image around a specific area to reduce the differences in contrast. Then, any mode-of-use can be applied.

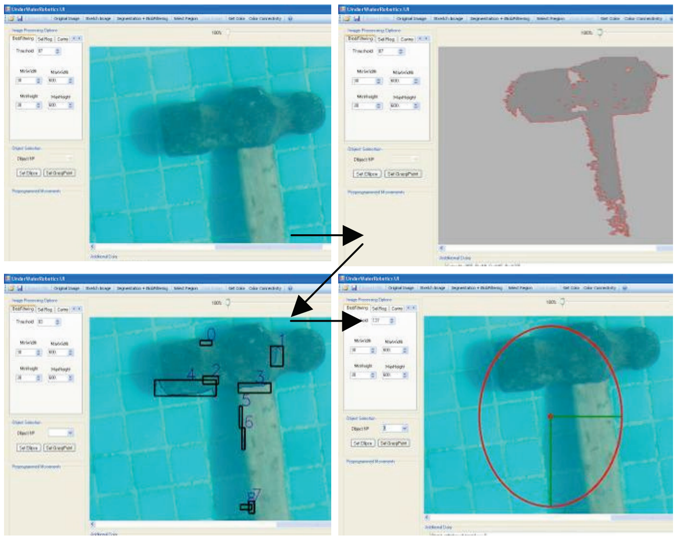

Figure 10. Example of Mode-2.
a) Initial input. b) *ColourConnectivity* result.
c) Segmentation result. d) Target selection.

d) Implementation details.

The GUI has been developed using C#, and the image process has implemented with the OpenCV [10] wrapper (EmguCV [11]) and the AForge.NET library. With this technologies combination, the system works efficiently in a Windows-based PC, as well as in other operative systems, if the GUI is correctly compiled.

IV. Conclusions and Future Lines

This work focuses on the design of a *user-friendly* GUI. This GUI has been developed to work in underwater intervention scenarios, and has been focused to identify the target and specify the suitable intervention task. This GUI does not need a high-qualified user and it is not robot dependant.

This research opens several future lines of work. Currently, a connection between the GUI and the simulator [1] is under development. This connection is made with a XML file, although the architecture approach that is being studied [13] suggests new possible integration directions. With this new connection type, the GUI will work and interconnect directly within other parts of the system, without a user intervention. This fact will improve the GUI usability. Some improvements such as using a Head-Mounted Display with a gyroscope will involve the user into a more realistic scenario. Future steps can concentrate on using neural networks or machine learning for enhancing the autonomy of the system and making it more reliable to uncertainties.

ACKNOWLEDGMENTS

This research was partly supported by the European Commission's Seventh Framework Programme FP7/2007-2013 under grant agreement 248497(TRIDENT Project), by Ministerio de Ciencia e Innovación (DPI2008-06548-C03-01), and by Fundació Caixa Castelló-Bancaixa (P1-1B2009-50).

REFERENCES

[1] G. De Novi, C. Melchiorri, J. C. García, P. J. Sanz, P. Ridao, G. Oliver, "A New Approach for a Reconfigurable Autonomous Underwater Vehicle for Intervention", *IEEE SysCon 2009 —3rd Annual IEEE International Systems Conference, 2009*, Vancouver, Canada, March 23–26, 2009.

[2] Choi. S.K. and Yuh. J.. "Design of Advanced Underwater Robotic Vehicle and Graphic Workstation". *1993 IEEE Conference on Robotics and Automation*.

[3] H. H. Wang, S. M. Rock, and M. J. Lee. OTTER: "The Design and Development of an Intelligent Underwater Robot". *Autonomous Robots*, 3(2-3):297-320, June-July 1996.

[4] Rigaud, V.; Coste-Maniere, E.; Aldon, M.J.; Probert, P.; Perrier, M.; Rives, P.; Simon, D.; Lang, D.; Kiener, J.; Casal, A.; Amar, J.; Dauchez, P.; Chantler, M., "UNION: underwater intelligent operation and navigation," *Robotics & Automation Magazine*, IEEE , vol.5, no.1, pp.25-35, Mar 1998.

[5] R. Bono, G.Veruggio, P.Virgili, "Evolution of Design and Irnplementation of an Ergonomic Human Cornputer Interface through the diflerent phases of the AMADEUS Project, *Oceans'98 Conference*, Nice, France, September 28-October 1, 1998.

[6] Evans, J.C.; Keller, K.M.; Smith, J.S.; Marty, P.; Rigaud, O.V., "Docking techniques and evaluation trials of the SWIMMER AUV: an autonomous deployment," AUV for work-class ROVs. *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, vol.1, no., pp.520-528 vol.1, 2001.

[7] Evans J., Redmond, P., Plakas, C., Hamilton, K and Lane, D, "Autonomous docking for Intervention-AUVs using sonar and video-based real-time 3D pose estimation", *OCEANS 2003. Proceedings* , vol.4, no., pp. 2201-2210 Vol.4, 22-26 Sept. 2003.

[8] Yuh, J.; Choi, S.K.; Ikehara, C.; Kim, G.H.; McMurty, G.; Ghasemi-Nejhad, M.; Sarkar, N.; Sugihara, K., "Design of a semi-autonomous underwater vehicle for intervention missions (SAUVIM)", *Underwater Technology, 1998. Proceedings of the 1998 International Symposium on* , vol., no., pp.63-68, 15-17 Apr 1998.

[9] Giacomo Marani, "Advances in Autonomous Underwater Intervention for AUVs", Work Shops and Tutorials, *2009 IEEE International Conference on Robotics and Automation* 12-17 May Kobe, Japan.

[10] Bradski, G., Kaehler, A., *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly, Sept. 2008

[11] http://www.emgu.com/wiki/index.php/Main_Page

[12] Sanz, P. J., Requena, A., Iñesta, J.M., del Pobil, A.P., "*A Grasping the Not-so-Obvious"*, IEEE Robotics & Automation Magazine, 2005.

[13] Palomeras, N., García, J. C., Prats, M., Fernández, J. J., Sanz, P. J., Ridao, P., "*A Distributed Architecture for Enabling Autonomous Underwater Intervention Missions"*, Unpublished.