

USER INTERFACE ORIENTED TO THE SPECIFICATION OF UNDERWATER ROBOTIC INTERVENTIONS

J.C. García^{1,2}, M. Prats^{1,3}, R. Marin^{1,4} and P.J. Sanz^{1,5}

Received 20 December 2010; in revised form 30 December 2010; accepted 11 June 2011

ABSTRACT

This paper presents our recent work in a graphical user interface (GUI) oriented to specify a robot intervention mission in underwater environments. The complete mission is divided in two phases: the survey and the intervention. The GUI helps the user to set the survey area and identify the target by using images compiled by the I-AUV (Autonomous Underwater Vehicle for Intervention) in the survey phase. Some GUI implementation details and their facilities, which assist assisting the user in the required specification of underwater intervention missions will be addressed. Finally a short description of the simulator, currently under development, will be described. Furthermore, preliminary simulation results will be shown, demonstrating the viability of the survey mission specified by the user through an available I-AUV.

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

INTRODUCTION

Nowadays a relevant number of field operations with unmanned underwater vehicles (UUVs) in applications such as marine rescue, marine science and the offshore industries, to mention some but a few, need intervention capabilities in order to undertake the desired task. In the offshore industry, UUVs have to dock to an under-

¹ Universitat Jaume I, Av. de Vicent Sos Baynat, s/n, 12071 Castellón, Spain. ² Personal investigador, Email: garciaju@uji.es. Tel. 964728294. ³ Contratado doctor, Email: mprats@uji.es. Tel. 964387048. ⁴ Profesor asociado, Email: rmarin@uji.es. Tel. 964728256. ⁵ Profesor titular, Email: sanzpj@uji.es. Tel. 964728285.

water panel to manipulate valves with their robotic arms. It is of interest, for marine scientists, to further develop the capability of accurate deploy and recover specialized instrumentation. In the context of the permanent observatories currently under design and development, the intervention capability for maintenance operations will be of vital importance. Interventions in marine rescue, are needed for instance to tighter ropes to wrecks for towing or recovery.

Currently, most of the intervention operations are being undertaken by manned submersibles endowed with robotic arms or by Remotely Operated Vehicles (ROVs). Manned submersibles have the advantage of placing the operator in the field of operation with direct view to the object being manipulated. Their drawbacks are the reduced time for operation (e.g. a few hours), the human presence in a dangerous and hostile environment, and a very high cost associated with the need of an expensive oceanographic vessel to be operated. Work class ROVs, are probably the more standard technology for deep intervention. They can be remotely operated for days without problems. Nevertheless, they still need an expensive oceanographic vessel with a heavy crane and automatic Tether Management System (TMS) and a Dynamic Position system (DP). It is also worth mentioning the cognitive fatigue of the operator who has to take care of the umbilical and the ROV while cooperating with the operator of the robotic arms. For all these reasons, very recently some researchers have started to think about the natural evolution of the intervention ROV, the Intervention Autonomous Underwater Vehicle (I-AUV). Without the need for the TMS and the DP, light I-AUVs could theoretically be operated from cheap vessels of opportunity reducing considerably the cost. With the fast development of batteries technology, and the operator being removed from the control loop, we can even think about intervention operations lasting for several days, where the ship is only needed the first and the last day for launching and recovering operations (Trident, 2010) But this fascinating scenario, where I-AUVs are launched to do the work autonomously before recovery, comes at the cost of endowing the robot with the intelligence needed to keep the operator out of the control loop. Although standard AUVs are also operated without human intervention, they are constrained to survey operations, commonly flying at a safe altitude with respect to the ocean bottom while logging data. I- AUVs must be operated in the close proximity of the seabed or artificial structures. Therefore, they have to be able to identify the objects of interest (i.e. for manipulation purpose) and the intervention tasks to be undertaken, while safely moving within a cluttered area of work.

Aim of the project

This work is part of two coordinated projects, RAUVI (Spanish national project) and TRIDENT (European Commission FP7). The main goal of these projects is to develop and improve the necessary technologies that will allow an I-AUV to autonomously perform an intervention mission in underwater environments.



In the RAUVI/ TRIDENT projects an intervention mission is divided in two phases: the survey and the intervention (see Figure 1). The GUI presented in this paper plays a significant role in these two steps. It is in charge of assisting the user in all the tasks related with the specification of the mission. The main advantages of the GUI are its friendly use and the low requirements of user expertise.

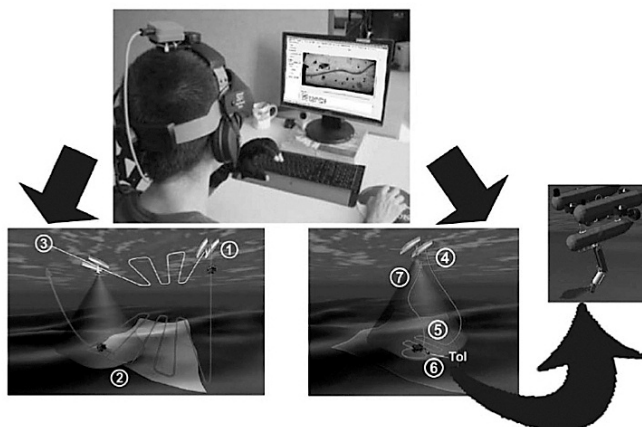


Figure 1: The different phases of the RAUVI/ TRIDENT projects and the role of the GUI.

In the first phase, the user specifies the survey area. The robot deploys (1), performs the survey task (2) and finally, the robot surfaces (3). In the second phase, after acquiring the images taken on the survey, the user can identify the Target of Interest (ToI) and select what kind of intervention should be performed. The robot deploys again (4), searches the ToI (5), performs the intervention (6) and docks again into the vessel platform (7).

State-of-the-Art

The I-AUV concept appears as the evolution of the AUV and the need to perform some kind of task along the mission. Most of the pioneering works relied on numerical simulations of the coupled dynamics of both systems, the coordinated control of the vehicle and the manipulator system. Some of these first robots were the ODIN AUV at University of Hawaii (Choi et al., 1993), the OTTER AUV (MBARI) (Wang et al., 1996) and the VORTEX/PA10 robot within the UNION European project (Rigaud et al., 1998).

During the mid 90s, AMADEUS EU project (Bono et al., 1998) represented a step forward in the field of dexterous underwater manipulation. It included within its objectives the realization of a set-up composed of 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 (Evans et al., 2001a). 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, without the need of a heavy umbilical. The ALIVE (EU) (Evans et al., 2001b) project 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, detection of the panel using an imaging sonar and finally, approach 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.

Probably the most advanced I-AUV is the one developed on the SAUVIM project (USA) (Yuh et al., 1998)(Marani, 2009), which is an AUV with a 7 DOF electrically driven arm (ANSALDO), the same used in the AMADEUS EU project, which is intended to recover objects from the seafloor using dexterous manipulation. One of the most noticeable aspects is the increasing amount of information exchanged between the system and the human supervisor. The user makes use of the Sauvim Programming Language (SPL) for writing in a console high-level information (e.g. "unplug the connector") for a particular mission.

Structure of the paper

This paper is structured as follows: Section 2 presents the graphical user interface, and describes in detail the facilities included in order to specify an underwater intervention mission. Section 3 briefly describes the integration of the GUI with the rest of the software architecture. Section 4 shows the use of the GUI with a simulated survey and intervention task. Finally, Section 5 concludes the paper.

THE GRAPHICAL USER INTERFACE

Usually, Remotely Operated Vehicles (ROVs) are controlled within master/slave architectures by pilots, who have to be experts in the particular task and the specific robot (see Figure 2 for an example). The main drawback in this kind of systems, apart from the necessary degree of expertise among the pilots, concerns the cognitive fatigue inherent to master/slave control architectures (Sheridan, 1992).

In order to improve this situation, our main goal is to develop a new GUI that does not require expert users with a high level of technical knowledge for its use. A description of the proposed GUI is given in the following.

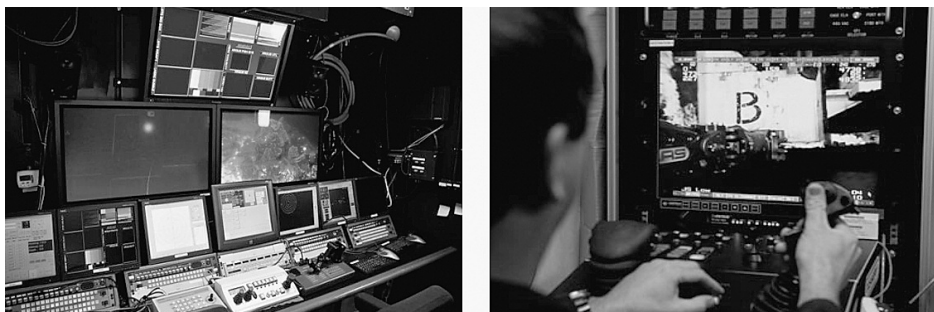


Figure 2: usual ROV resources and ROV pilot GUI.

Input characterization

The initial input compiled in the survey mission will always be different to the final conditions arising during the intervention mission, due to the fact that the underwater scenario provides a hostile and very changing environment. In this way, an intuitive GUI is

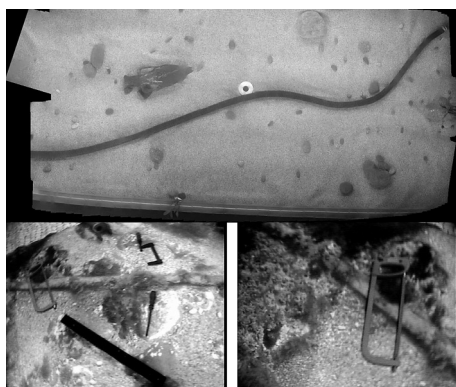


Figure 3: real mosaics, which have been taken by different AUVs.

very important to help the user specify the survey and the final intervention missions. Some of the images used along the GUI development can be observed in Figure 3.

The University of Girona AUV obtained the Figure 3-above in a real scenario, while the University of the Balearic Islands AUV obtained the Figure 3-below using real tools on a photo-mosaic. As we can observe, the quality of the images varies significantly, depending on the scenario and the robot used. So, the GUI should have different tools that allow the user to perform the mission successfully.

GUI features

When the user runs the GUI, it gets a clear interface where there are only few buttons enabled. These buttons are the “Predefined Intervention Task”, “Open an image” or “Connect to the simulator”.

Although Figure 4 shows different types of interventions, currently only the “Survey” and the “Object Recovery” task are available. We expect to implement the rest of the tasks in the future.

Depending on the user selection, a panel with task options is enabled. In this panel, the user can find different parameters that allow modification of the default behaviour. This panel can be observed in Figure 5.



Defining a “Survey” task

GUI offers the user two possibilities: use GoogleEarth or load a map. The use of GoogleEarth provides the user with some advantages: the maps are always up-to-date by Google, so there is no work for the user; it is a free service; the user can customize the selected view with more detailed data or add images into the map; the user can store information and maps for using it in offline mode. On the other hand, there are few drawbacks: an Internet connection is needed and it does not work properly if the area of interest is too small. We are still trying to get all the benefits of this feature, which is under development (fig. 7).

If the user has a custom map, it can also be opened into the GUI. For specifying the survey area, the user selects the initial and final point of the survey. Then, in the Task Options panel, the user selects what type of path the robot has to follow: either a one-way trip, return trip or sinusoidal trip. Depending on this selection, the user indicates some specific parameters such as the amplitude of the sinusoidal path, the altitude and depth, and so on. Forthcoming developments, a more realistic path planner will be included (Garau et al., 2009).

In order to finish the mission specification, the user can export all the data to a XML file or simulate the mission in our simulator, which will be detailed in Section 4.



```
<?xml version="1.0" encoding="UTF-8"?>
<Root>
  <Image>
    <ImagePath>E:\Maps\map1.png</ImagePath>
    <ImageWidth>791</ImageWidth>
    <ImageHeight>445</ImageHeight>
  </Image>
  <Intervention>
    <Type>Survey</Type>
  </Intervention>
  <StartPoint>
    <xPoint>324</xPoint>
    <yPoint>126</yPoint>
  </StartPoint>
  <WaypointsList>
    <Waypoint1>
      <xPoint>376</xPoint>
      <yPoint>188,4</yPoint>
    </Waypoint1>
  </WaypointsList>
  ...
  <UserOptions>
    <GpsOnTop>GpsOnTop</GpsOnTop>
    <PixelRatio>0,91</PixelRatio>
    <WaypointDelay>60</WaypointDelay>
    <Unit>Meter</Unit>
  </UserOptions>
  ...
</Root>
```

Figure 6: XML file example.

Defining an “Object Recovery” task

We assume that once the robot has finished the first phase (the survey), the robot has got a mosaic of the seabed. The user can load this mosaic into the GUI and navigate through it until finding the Target of Interest (ToI) (fig. 8).

The mosaics can vary a lot depending on multiple parameters. Algorithms that work well with one mosaic, may fail when applied to other mosaics due to different visibility conditions, different backgrounds, etc. For this reason, the GUI offers the

user three different types of use, where the system autonomy increases as the user responsibility decreases. Further information can be found in (Garcia et al., 2010a).

Mode 1: when the user clicks on the ToI, the GUI gets the greyscale colour of that point and uses this value as starting threshold value to segment the scene. Then, the algorithm processes the image identifying all the blobs detected and the GUI shows the blobs that fit within a size range set by the user. If the ToI is smaller than the size threshold, it is not shown. In this case the user can modify the blob filtering parameters until the ToI is found. This is the lowest-level interaction mode for the user.

Mode 2: the user selects one point of the ToI and the GUI applies a colour connectivity algorithm to get all the colour pixels with a similar RGB colour value. Then, the GUI will show a blob containing that area.

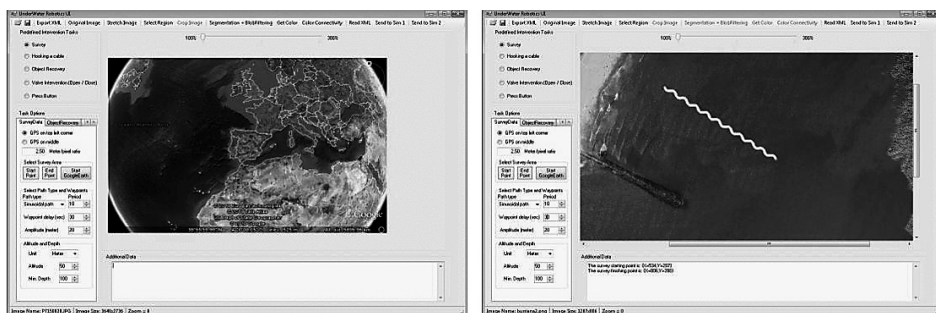


Figure 7: different ways to specify a “Survey” mission: using GoogleEarth or using a map.

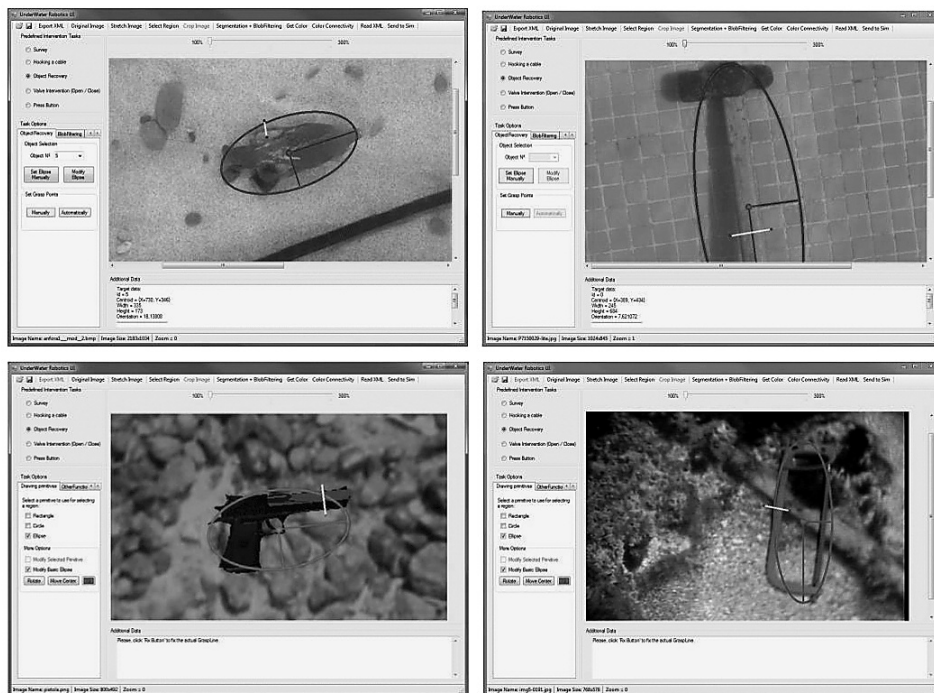


Figure 8: Target of Interest identified, selected and grasping points defined manually.

Once the user has identified the target, the user selects its identifier number in the Task Options panel. Then, automatically, the GUI prints an ellipse over the ToI. This ellipse follows the *best fitting ellipse principle*, which gets the object length and width, as well as object centroid and orientation.

Finally, the last user interaction mode:

Mode 3: the user draws the ellipse directly on the ToI. This is the lowest system autonomy mode, so the user has all the responsibility.

Regardless of the mode, the user always gets an ellipse on the ToI. This moment is called “Target Selection”. If the ellipse does not correctly match with the object centroid and orientation, the user can manually adjust them using specific methods.

We are currently focusing our research on an “Object Recovery” intervention mission. In order to select the grasping points, the user has two possibilities:

- Drawing a line, where the extreme points represent the grasping points (Figure 9).
- Running a function. This function searches the first point placed out of the blob, starting on the centroid and following the ellipse minor axis in both direc-

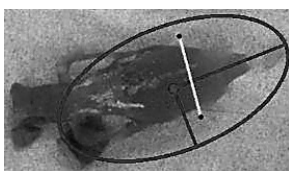


Figure 9: grasping points defined manually by the user.

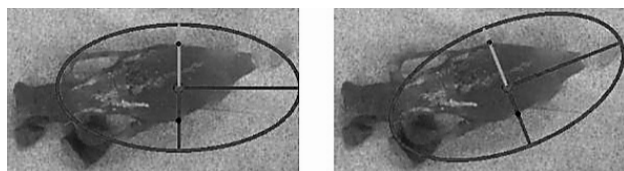


Figure 10: grasping points defined automatically.

user can export the intervention mission in a XML file and the specification process finishes.

ARCHITECTURE INTEGRATION

The project architecture is distributed and has four main nodes: the GUI, the Mission Control System (MCS), the Architecture Abstraction Layer (AAL) and the real robot and simulator architectures.

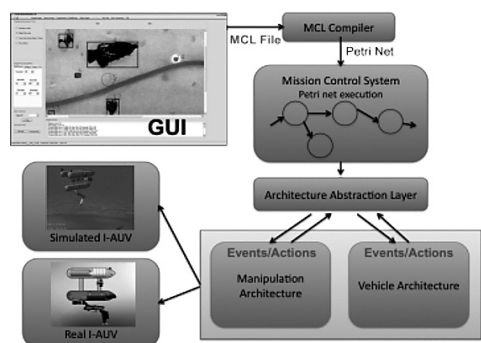


Figure 11: general diagram of the software architecture.

tions. When the function detects this point, it stops and marks the point as grasping point (see Figure 10).

After marking the grasping points, the

GUI XML file into the Mission Control Language, creates a Petri net for the mission, and uses the AAL for sending actions and events to the architectures. At the lowest level, the control signals can be sent to either the real robot or a simulated one. This allows using the same GUI and control algorithms independently of the low-level layer. Further information about this can be found in (Palomeras et al., 2010).

A MISSION SIMULATION

In order to validate the GUI and the control algorithms, a simulated underwater environment has been developed. It has been implemented in C++ on top of the OpenSceneGraph libraries (OpenSceneGraph, 2010) and osgOcean nodekit for OpenSceneGraph (osgOcean, 2010). This allows visualizing underwater effects like silt, light attenuation, water distortion, etc. More specifically, the simulation environment includes:

- The I-AUV 3D model, including both the vehicle and the arm. Arm kinematics have also been implemented, thus allowing the movement of the arm joints.
- A virtual camera attached to the bottom of the vehicle and looking downwards. Another virtual camera has been attached to the wrist of the arm. These cameras capture images of the seabed in real-time.

- A lamp, placed on the bottom part of the vehicle and pointing towards the floor.
- A seabed model including a texture.
- Different object models that can be placed on the seabed.

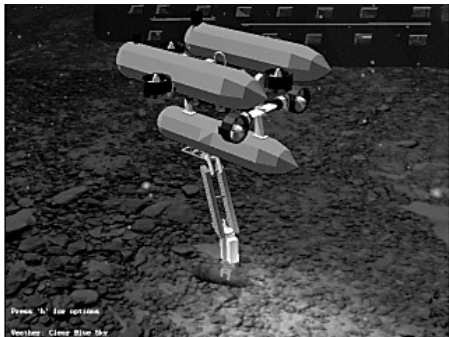


Figure 12: a screenshot of our developed underwater simulator, where an I-AUV grasps an amphora according to the user specification on the GUI.

The simulator has been implemented as a network server, where other software modules can be connected in order to update the pose of the vehicle-arm, or to receive input from sensors like joint odometry or images from the virtual cameras. This allows using it as a mere visualizer, where the actual control algorithms are not part of the simulator, but external to it, and can be interfaced to the real robot without modification. This leads to more realistic simulations. In fact, the simulator is external to the control architecture and interfaced to it

in the same way the real robot is interfaced (see Figure 11). This enables the user to switch easily between the real robot and the simulated one without modifying the rest of modules.

Control algorithms in the software architecture take as input the XML file generated by the GUI, and provide control signals to the vehicle and arm (either real or simulated) in order to perform survey and intervention.

In the particular case of survey simulation, the vehicle path, specified in pixel units in the GUI, must be transformed into 3D coordinates given with respect to the origin frame of the simulated environment. Given an image waypoint in pixel units:

$$p_i = (p_x, p_y) \quad (1)$$

its 3D homogeneous coordinates with respect to frame I (see Figure 13), are given by:

$${}^I P_i = \left(\frac{Z \cdot p_x}{f_x}, \frac{Z \cdot p_y}{f_y}, 0, 1 \right) \quad (2)$$

where Z/f_x and Z/f_y are the meter/pixel ratio of the image on the horizontal and vertical directions. Having the frame I expressed with respect to a global positioning reference system, G, by the homogeneous transformation matrix ${}^G M_I$, and a local reference system M used for the survey (in simulation M corresponds to the origin of the



simulated world), also given with respect to G by ${}^G M_M$, the point p_i can be transformed to the local reference frame as:

$${}^M P_i = \left({}^G M_M \right)^{(-1)} \cdot {}^G M_I \cdot {}^I P_i \quad (3)$$

In the particular case of the sinusoidal path used in the GUI for survey specification, the curve is sampled at regular intervals and each image waypoint is trans-

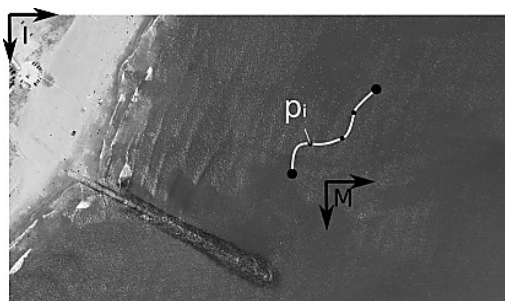


Figure 13: all the different reference systems in a “Survey” mission.

formed into the 3D frame where the vehicle is localized (M) as mentioned above. These 3D points are sent to the vehicle navigation module that performs the vehicle control. After doing the survey, the ToI is detected and a grasp is specified using the GUI. Figure 12 shows a grasp performed on the simulated environment, taking as input the user specification on the GUI.

CONCLUSION

We have developed a user-friendly GUI for autonomous underwater intervention missions specification. The GUI is used for indicating a survey area, identifying the target object and specifying a suitable intervention task, in the context of the Spanish and European research projects RAUVI and TRIDENT.

We have focused our GUI on the user requirements. Our approach differs from other solutions in the little need of user experience. In fact, the user does not need to learn a specific language to manage the GUI and does not require a long previous experience in the field.

Future work

Future work will be mainly focused on enhancing the human-robot interaction, using different types of Virtual and Augmented Reality devices and techniques. Currently, our work is focused on nVidia 3D, which allows the user to have a 3D experience with low hardware requirements and costs.

Furthermore, there is still more work to do related with GUI features, such as adding more functions to the GoogleEarth integration or adding new types of intervention (i.e. hooking a cable, pressing a button...).

ACKNOWLEDGEMENT

This research was partly supported by the European Commission Seventh Framework Programme FP7/2007- 2013 under grant agreement 248497 (TRIDENT Project), by Spanish Ministry of Research and Innovation DPI2008-06548- C03 (RAUVI Project), and by Fundació Caixa Castelló-Bancaixa P1-1B2009-50.

REFERENCES

- Bono, R., Veruggio, G., Virgili, P., (1998) Evolution of Design and Implementation of an Ergonomic Human Computer Interface through the different phases of the AMADEUS Project, *Proceedings of OCEANS-1998*, 28-Sept. – 1-Oct., Nice, France, pp. 718-722.
- Choi, S.K. and Yuh, J. (1993) Design of Advanced Underwater Robotic Vehicle and Graphic Workstation, *Proceedings of IEEE Conference on Robotics and Automation*, 2-6 May, Atlanta, USA, pp. 99 – 105.
- De Novi, G., Melchiorri, C., García, J. C., Sanz, P. J., Ridao, P., Oliver, G., (2009) A New Approach for a Reconfigurable Autonomous Underwater Vehicle for Intervention, *Proceedings of IEEE SysCon 2009 —3rd Annual IEEE International Systems Conference*, 23–26 March, Vancouver, Canada, pp. 23-26.
- Evans, J.C.; Keller, K.M.; Smith, J.S.; Marty, P.; Rigaud, O.V., (2001) Docking techniques and evaluation trials of the SWIMMER AUV: an autonomous deployment, AUV for work-class ROVs. *Proceedings of OCEANS-2001; MTS/IEEE Conference and Exhibition*, 5-8 Nov., Honolulu, USA, pp. 520-528.
- Evans, J., Redmond, P., Plakas, C., Hamilton, K and Lane, D, (2003) Autonomous docking for Intervention-AUVs using sonar and videobased real-time 3D pose estimation, *Proceedings of OCEANS-2003*, , 22-26 Sept., San Diego, California, USA, pp. 2201-2210.
- Garau B., Bonet, M., Alvarez, A., Ruiz, S., Pascual, A. (2009) Path Planning for Autonomous Underwater Vehicles in Realistic Oceanic Current Fields: Application to Gliders in the Western Mediterranean Sea, *Journal of Maritime Research*, Volume 6, Number 2. 2009. ISSN: 1697-4840
- García, J. C., Fernández, J. J., Sanz, P. J., Marin, R. (2010) Increasing Autonomy within Underwater Intervention Scenarios: The User Interface Approach. *Proceedings of IEEE International Systems Conference –4th Annual IEEE International Systems Conference*, 5-8 April, San Diego, CA, USA, pp. 71-75.
- Garcia, J.C., Prats, M., Sanz, P.J., Marin, R. (2010) Exploring Multimodal Interfaces For Underwater Intervention Systems, *Proceedings of the ICRA 2010 Workshop on Multimodal Human - Robot Interfaces*, 3-8 May, Anchorage, Alaska, pp. 37-41.
- Marani, G., (2009) Advances in Autonomous Underwater Intervention for AUVs, Work Shops and Tutorials, *Proceedings of the IEEE International Conference on Robotics and Automation*, 12-17 May, Kobe, Japan.



- OpenSceneGraph, 2010. OpenSceneGraph is an open source high performance 3D graphics toolkit. Available from: <http://www.openscenegraph.org>
- osgOcean, 2010. OpenSceneGraph is an ocean rendering nodekit for OpenSceneGraph. Available from: <http://code.google.com/p/osgocean>
- Palomeras, N., García, J. C., Prats, M., Fernández, J. J., Sanz, P. J., Ridao, P., A Distributed Architecture for Enabling Autonomous Underwater Intervention Missions, *Proceedings of the ICRA 2010 Workshop on Multimodal Human - Robot Interfaces*, 3-8 May, Anchorage, Alaska, pp. 159-164.
- 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, *IEEE Robotics & Automation Magazine* 5 (1) 25-35, Mar 1998.
- Sheridan, T.B. (1992) *Telerobotics, Automation and Human Supervisory Control*. Massachusetts: MIT Press.
- Trident FP7 European Project, 2010. Available from: <http://www.irs.uji.es/trident>
- Wang, H. H., Rock, S. M., Lee, M. J., (1996) OTTER: The Design and Development of an Intelligent Underwater Robot. *Autonomous Robots* 3 (2-3), 297-320.
- Yuh, J.; Choi, S.K.; Ikehara, C.; Kim, G.H.; McMurty, G.; Ghasemi-Nejhad, M.; Sarkar, N.; Sugi-hara, K., (1998) Design of a semi-autonomous underwater vehicle for intervention missions (SAUVIM), *Proceedings of the 1998 International Symposium on Underwater Technology*, 15-17 Apr., Tokyo, Japan, pp. 63-68.

INTERFAZ GRÁFICA DE USUARIO ORIENTADA A LA ESPECIFICACIÓN DE UNA MISIÓN DE INTERVENCIÓN PARA UN ROBOT EN ENTORNOS SUBMARINOS

RESUMEN

El presente artículo muestra nuestro reciente trabajo de una interfaz gráfica de usuario orientada a la especificación de una misión de intervención para un robot en entornos submarinos. La misión se divide en dos fases: el reconocimiento y la intervención. La interfaz asiste al usuario estableciendo el área de reconocimiento e identificando el objetivo, usando las imágenes compiladas por el I-AUV (Vehículo Autónomo Submarino para Intervención) en la fase de reconocimiento. A continuación, se detallarán algunos detalles de implementación de la interfaz, así como sus facilidades para la asistencia al usuario en la especificación de las tareas de misiones de intervención submarinas. Finalmente, se describirá brevemente el simulador bajo desarrollo.

METODOLOGÍA

Actualmente, los sistemas de intervención submarina más comunes son los sistemas Vehículo Operado Remotamente (ROV), que mediante una arquitectura maestro/esclavo se operan por un piloto de ROV altamente cualificado. Estos sistemas requieren de una gran cantidad de recursos, desde grandes buques oceanográficos, cables umbilicales, grandes salas para el control de los mismos, etc. Además, toda la responsabilidad de la intervención recae sobre el usuario, que debe ser un experto tanto en el robot como en el tipo de intervención a realizar. Desde los proyectos de investigación español y europeo RAUVI y TRIDENT, se pretende desarrollar y mejorar las tecnologías necesarias que haciendo uso de un Vehículo Autónomo Submarino para Intervención (I-AUV) se consigan realizar satisfactoriamente tareas de intervención.

Uno de los nodos dentro de la arquitectura de ambos proyectos es la interfaz gráfica de usuario (GUI), mediante la cual, se podrán establecer cada una de las fases en las que consta una intervención: la fase de reconocimiento y la fase de intervención. Para la fase de reconocimiento, el usuario tiene a su disposición dos métodos distintos, que son el uso de GoogleEarth, que aún está en fase de desarrollo, o bien cargar un mapa. Una vez se ha cargado el mapa dentro de la aplicación, el usuario tan solo tiene que especificar los puntos iniciales y finales del área a reconocer, así como ciertos parámetros necesarios (tipo de recorrido, profundidad, altitud). Para la fase de intervención, y tras cargar el mosaico de fotos realizado por el robot en la



anterior fase, el usuario podrá buscar, identificar y seleccionar su objeto de interés. Dado que se prevén diversos tipos de intervención, en la actualidad nos hemos centrado en el tipo *recuperación de un objeto*. Para poder identificar el objeto de interés, el usuario dispone de 3 modos de interacción distintos, dependiendo de la complejidad de las imágenes. Así pues, a parte especificar el objeto de interés, el usuario deberá especificar los puntos de agarre de dicho objeto, para lo cual, dispone de dos funciones: especificarlos manualmente o usar una función que calcula una aproximación a los puntos de agarre óptimos.

Finalmente, se ha integrado la interfaz en la arquitectura del proyecto mediante el uso de un archivo XML donde se guardan los datos más importantes de cada fase de la intervención. Además, existe la posibilidad de simular la fase de reconocimiento en el simulador que estamos desarrollando, usando una conexión vía sockets.

CONCLUSIONES

Se ha desarrollado una interfaz de usuario de fácil uso para la especificación de misiones de intervención submarinas autónomas. La interfaz se usa para establecer el área de reconocimiento, identificación del objeto de interés y la especificación de la tarea final de intervención, en el contexto de los proyectos de investigación español y europeo RAUVI y TRIDENT.

Durante el desarrollo, nos hemos centrado en los requisitos del usuario. Nuestro enfoque difiere de otras soluciones, en la poca necesidad de experiencia por parte del usuario. De hecho, el usuario no necesita aprender un lenguaje específico para usar la interfaz y no requiere una experiencia previa en dicho campo.

