Proceedings of the IEEE ICRA 2010
Workshop on Multimodal Human-Robot Interfaces
Anchorage, Alaska, May, 2010

Exploring Multimodal Interfaces For Underwater Intervention Systems

J. C. Garcia, M. Prats, P. J. Sanz, *Member, IEEE,* R. Marin, *Member, IEEE, and* O. Belmonte

Abstract— **Graphical User Interfaces play a very important role in the context of Underwater Intervention Systems. Classical solutions, specially concerning Remotely Operated Vehicles, frequently require users with an advanced technical level for controlling the system. In addition, continuous human feedback in the robot control loop is normally needed, thus generating a significant stress and fatigue to the pilot.**

This paper shows work in progress towards a new multimodal user interface within the context of autonomous underwater robot intervention systems. We aim at providing an intuitive user interface that can greatly improve the non-expert user's performance and reduce the fatigue that operators normally experiment with classical solutions. For this, we widely adopt advanced interaction systems such as haptic devices, projectors, Head-Mounted Display and more.

Keywords— **Graphical User Interface (GUI), Autonomous Underwater Vehicle for Intervention (I-AUV), multimodal interface, simulator.**

I. Introduction

CURRENTLY Remotely Operated Vehicles (ROVs) are commercially available to develop all kind of intervention missions. These systems are underwater robots tethered to a mother ship and controlled from onboard that ship. Here the control is assumed by an expert user, called the ROV pilot, by means of a special Graphical User Interface (GUI) with specific interaction devices like a joystick, etc. The main drawback in this kind of systems, apart from the necessary expertise degree of pilots, concerns the cognitive fatigue inherent to master-slave control architectures [1].

On the other hand, the best underwater robotics labs around the world are recently working for the next technology step, trying to reach new levels of autonomy far beyond those present in current ROVs. These technologies have lead to Autonomous Underwater Vehicles for Intervention (I-AUVs), which represent a new concept of undersea robots that are not tethered to a mother ship. In fact, the history about I-AUVs is very recent, and only a few laboratories around the world are currently trying to develop this kind of systems [2].

One of the most well-known research projects devoted to develop an I-AUV is SAUVIM [3]. Along its life, this project has implemented a GUI combining all kind of sensor data inside a common simulation environment. Their GUI uses its own programming language and allows for high level interaction of the user and the underwater robot in text mode. In addition, virtual reality (VR) is available within the GUI, thus showing the evolution of the complete system along the intervention mission, and assisting the user in the high-level control. This very complete interface has shown to be very suitable for users with an advanced previous expertise, but might be too complex for a new user without technical knowledge.

Our research group is working on this kind of underwater intervention systems in general, and more concretely in specific multimodal interfaces that allow an intuitive use by non-expert users. In fact, because of the impossibility to implement a complete I-AUV autonomy level with available technology, we design a two steps strategy [4], guaranteeing the "intelligence" in the system performance including the user in the control loop when strictly necessary, but not in a continuous way like in ROV's. Thus, in a first step, our I-AUV is programmed at the surface, and then navigates through the underwater Region of Interest (RoI) and collects data under the control of their own internal computer system. After ending this first step, the I-AUV returns to the surface (or to an underwater docking station) where its data can be retrieved. A 3D image mosaic is constructed, and by using a specific GUI, including virtual and augmented reality, a non-expert user is able to identify the target object and to select the suitable intervention task to carry out during the second step. Then, during this second step, the I-AUV navigates again to the RoI and runs the target localization and the intervention modules onboard. Our I-AUV system concept, currently under construction in Spain (i.e. RAUVI's Spanish Coordinated Project), can be observed in Figure 1, where the vehicle, developed in the University of Girona (Spain) and the arm, under responsibility of Univerisity Jaume I (Spain), that is an adaptation of the "arm 5E" from CSIP Company (UK) must be assembled in the next months. Moreover, it is noticeable that just now we are starting out the coordination of a European Project named TRIDENT within the same context but with a bit more challenging long term objectives.

Thus, this paper shows our ongoing research on

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), and by Fundació Caixa Castelló-Bancaixa (P1-1B2009-50).

J.C. García, M. Prats, P.J. Sanz and R. Marin are with the Department of Computer Science & Engineering, at the Universitat Jaume I, Spain ([garciaju,mprats,sanzp,rmarin]@uji.es).

O. Belmonte is with the Department of Computer Languages & Systems, at the Universitat Jaume I, Spain (Oscar.Belmonte@uji.es).

multimodal user interfaces for enabling the aforementioned kind of underwater intervention missions, initially focused on recovery object tasks. We aim to provide an intuitive user friendly interface improving the non-expert user's performance and reducing the inherent fatigue within traditional ROV interaction ways. Section II describes our recent efforts for building such an interface, including our ongoing work on immersive underwater simulation, facilities for target identification and task specification, and recent progress in grasp simulation. Section III clarifies the main drawbacks and advantages of our solutions when compared with the state of the art technologies, and also discusses the results obtained so far and the long list of challenges that need to be addressed. Finally, Section IV concludes this paper.

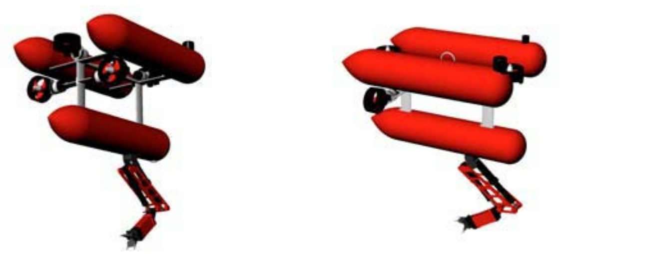


Fig. 1. The I-AUV envisioned concept currently under construction within the RAUVI's Spanish Coordinated Project.

II. TOWARDS A NEW MULTIMODAL INTERFACE

The whole mission specification system is composed of three modules: a GUI for object identification and task specification, a grasp simulation and specification environment, and the I-AUV Simulator. After target identification and specifying the intervention task, all the information is displayed into another 3D environment where the task can be simulated and the human operator can either approve it or specify another strategy by means of some facilities addressed within the interface. Finally, another environment is used for simulating and supervising the overall intervention mission. The ongoing work on these three modules is detailed in the following.

A. GUI for target identification and task specification.

Two main tasks must be solved in the underwater intervention context: the target identification and the specification of the suitable intervention to carry out over the target. Initially, a GUI is used for specifying the task to perform. Once the desired task has been selected, the GUI provides facilities for detecting interesting objects and identifying the target.

We are currently trying to expand the facilities available through the GUI for enabling a more intuitive level of interaction. In this way, the developed GUI (Figure 2) tends to be *user-friendly* with few requirements from the user side. Some examples of the intervention tasks to specify could be hooking a cable, pressing a button, etc. Currently we are focused on a specific task related with object recovery, where a suitable grasp has to be performed in order to manipulate in a reliable manner the target object.

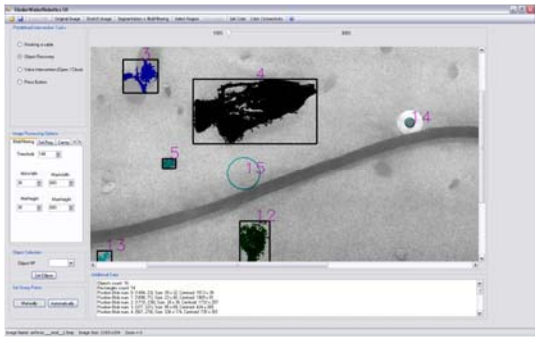


Fig. 2. An example of GUI screenshot: the object detection process

Looking for easy-to-use interaction ways, the GUI assists the user adapting its interface depending on the task to perform. Once the user has loaded the input image (i.e. first step in the process) and selected the intervention task, the user identifies the object and selects the target. For that, the GUI provides methods for object characterization and also for assisting in the grasping determination problem. The planned grasp will be later used in the grasping simulator and finally, in the real system. The general process can be observed in Figure 3. Due to the pour visibility conditions in the underwater environment and so, in the input image, the user could have problems to identify correctly the target. Low-level details about the different interaction ways currently available within thee GUI under development can be found elsewhere [5].

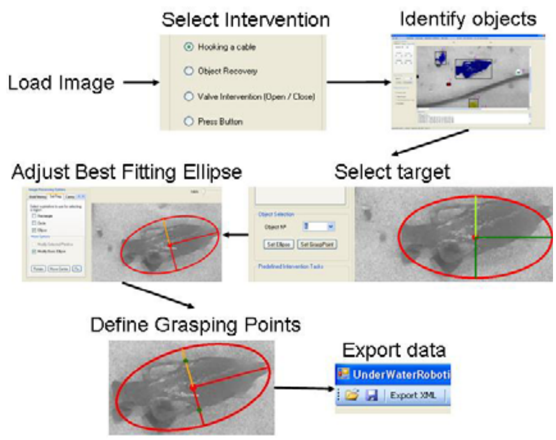


Fig. 3. Main steps through the GUI under development during the object characterization process.

The underwater scenario provides a hostile and very changing environment, including poor visibility conditions, streams and so on. So, the initial input compiled during the survey mission will be always different to the final conditions arising during the intervention mission. Thus, a predictive interface ensuring realistic task simulation is more than convenient before the robot be able to carry out the intervention defined by the user in the GUI.

B. *Grasp simulation and specification.*

Our most recent work is focused on an intuitive grasp simulation and supervision system that allows the user to visually check and validate the candidate grasps or to intuitively refine them in case they are not suitable. The grasping simulator will get data from the XML file generated by the previous object detection and task specification GUI. This data will include candidate grasping points and other target object properties that will be displayed in the simulator following augmented reality techniques (e.g. grip opening, joint angles, planned contact points, etc.).

The user's hand will be covered by a data glove with a tracking system that will allow replicating the human hand motion in the simulated environment. This will be used for specifying the required elements of a grasp (e.g. the hand configuration, grip opening, etc.), and also for indicating predefined actions through specific gestures (see Figure 4).

Fig. 4. Detail of the P5 data glove during a simple test: "grasp a virtual cube".

Our research team has a long experience in robotic grasping using the knowledge-based approach [6]. This approach defines a set of hand preshapes, also called hand postures or prehensile patterns, which are hand configurations that are useful for a grasp on a particular shape and for a given task. Several hand preshapes taxonomies have been developed in robotics, being the one proposed by Cutkosky [7] the most widely accepted. Since the publication of the Cutkosky's taxonomy, several researchers in the robotics community have adopted the grasp preshapes as a method for efficient and practical grasp planning in contrast to contact-based techniques.

One of our recent contributions in the field of robotic grasping is the concept of ideal hand task-oriented hand preshapes [8], which are a set of hand preshapes defined for an ideal hand and extended with task-oriented features. The ideal hand is an imaginary hand able to perform all the human hand movements. Our approach is to plan or define grasps by means of ideal preshapes, and then define hand adaptors as a method for the instantiation of the ideal preshapes on real robotic hands. The main advantage of this approach is that the same grasp specification can be used for different hands, just by defining a suitable mapping between the ideal hand and the real one. This concept is illustrated in Figure 6, which shows three different ideal preshapes and their mapping to a robotic Barrett Hand.

We plan to adopt this approach for the grasp specification and execution in the context of our grasp simulator. The human operator will specify a grasp using its own hand covered with a data glove. The finger joint angles captured by the data glove tracking system will be passed to a standard classifier (e.g. like in [9]) that will select the ideal hand preshape that best suites the human hand posture. The grasp will be specified by the ideal hand preshape and the part of the object where it is applied. For its execution by a robotic hand, the corresponding hand adaptor will transform the ideal preshape into a real posture depending on the robotic hand. The grasp will be finally simulated with the real robotic system as shown in Figure 5.

1) *Low level details for the grasp simulator.*

In order to develop the grasping simulation, some of the most common and used game and physics engine software, have been explored. A *game engine* is a software system designed for the creation and development of video games. The core functionality typically provided by a game engine includes a rendering engine for 2D/3D graphics, a physics engine or collision detection and response, and so on. On the other hand, a *physics engine* is used to model the behaviors of objects in space, using variables such as mass, velocity, friction, and wind resistance. It can simulate and predict effects under different conditions that would approximate what happens in real life or in a fantasy world. They are also used to create dynamic simulations without having to know anything about physics.

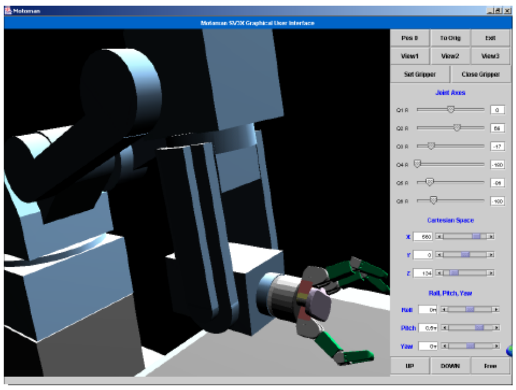
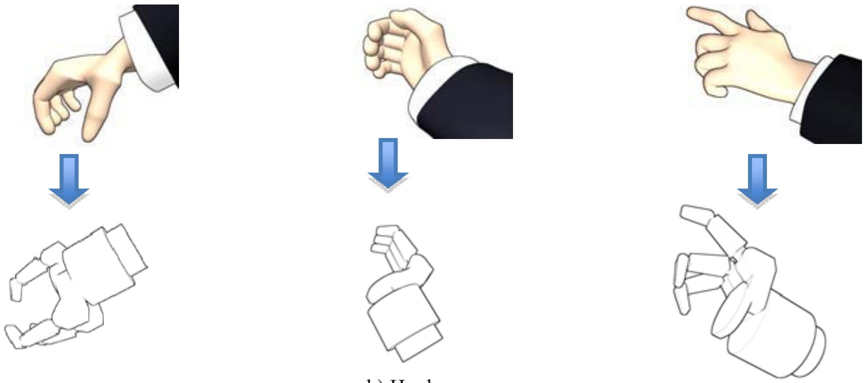


Fig. 5. GUI integrating the Barrett Hand 3D-model simulator

Despite both software platforms seems to be similar, a very important difference exists between them. The *physics engine* uses the Physics Processing Unit (PPU), which is a dedicated microprocessor designed to handle the calculations of physics, (e.g. rigid and soft body dynamics, collision detection or fracturing of objects). Using this dedicated microprocessor the CPU is off-loaded of high time-consuming tasks.



a) Cylindrical b) Hook c) One finger

Fig. 6. Three different ideal preshapes and their mapping to a Barrett Hand

In this way, the software compared is the jMonkeyEngine [10] (JAVA game engine) and PhysX [11] (physic engine). jME is a high performance scene graph based graphics API and is completely open source under the BSD license. A complete feature list can be found in [12]. On the other hand, PhysX is a proprietary solution of NVIDIA, but its binary SDK distribution is free under the End User License Agreement (EULA). A complete feature list can be found in [13].

The main difference between both engines is the platform compatibility and PC performance. Whereas jME is available for PCs (Windows, Linux and MacOS), PhysX is available for PCs (Windows and Linux) and all the actual videogames platforms (PS3, Xbox360, Wii). This justifies the number of more than 150 title games using PhysX technology. In PC performance terms, the use of a NVIDIA graphic card compatible with PhysX increases the general PC performance. Of course, with a SLI [14] schema with one dedicated graphic card, PhysX would deliver up to twice the PC performance (in frames per second). We should notice that PCs with an ATI graphic card would not get all the advantages of this technology, due to PhysX is a proprietary solution of NVIDIA, although they could still run the program.

Thus, in our first approach developing the grasping simulator, we are considering the NVIDIA physics engine. Besides the advantages explained before, we will try to take profit of the latest NVIDIA graphic card features, even using its 3D Vision technology [15]. This technology enables 3D vision over every single application, and only needs a *3dReady LCD monitor* and a NVIDA GeForce 3D Vision glasses.

C. I-AUV Simulator.

Previous research in this context has been developed in our Laboratory since 2008, starting with the cooperation with the University of Bologna, Italy, in order to implement a complete simulator [4]. This simulator includes a complete I-AUV 3D model, and emulates the physics of both the underwater environment and the robotic system. Currently, we are improving the user interaction capabilities by using a *Head Mounted Display* with an accelerometer, enabling to control the virtual cameras by means of the human head's movements. Further development could also include data gloves for gesture recognition, as can be observed in Figure 7.

Fig. 7. The initial simulator under development.

On the other hand, another I-AUV simulator is being developed at our laboratory, as observed in Figure 8. Its main features are the distributed and collaborative properties, as well as the use of advanced Virtual Reality (VR) devices. Low-level details can be found elsewhere [16]. This simulator uses a distributed and collaborative system, which enables to combine remote data coming from different PCs that can be placed in different locations. Thus, different users can work in cooperation with this kind of interface achieving simultaneously task specification missions/simulations that can be observed by different users in real time.

Fig. 8. The I-AUV is teleoperated by the user by means of special VR hardware, including immersive 3D vision and a virtual joystick controlled with data gloves.

In particular, this kind of cooperative interface opens new capabilities for personal training, enabling the possibility of sharing the VR interface among several remote experts and non expert's users. In this way, researchers on different disciplines can focus on the simulation aspects that are most interesting for their research, either if they are not physically present in the ship.

However, this cooperative VR interface has a serious drawback: the high costs underlying the specific hardware resources included in such a system.

III. Discussion

After exploring different possibilities of interfaces including all kind of VR devices, simulators and the potential of cooperative work, it is clear that significant benefits can be achieved. Probably one of the main advantages is what concerns the user training. In fact, the interaction by means of more intuitive and user friendly interfaces would allow reducing the pilot training period. In particular, the use of the developed VR technology, including distributed, collaborative and multimodal components, allows the user to interact in a very realistic way with the intervention scenario, promoting prediction actions. In addition, it allows appreciating the nature of the problems in case the simulation of the mission plan fails.

The most important difference between our approach and other existing solutions is that we put a special emphasis on the use of advanced technologies and functionalities making easier the human robot interaction for non-expert users.

For instance, the SAUVIM's GUI integrates several modules into one single interface, so the overall user interface provides a very powerful and flexible solution for monitoring the state of the robot during the mission, and provides advanced mechanisms for the low-level control. However, the interface has been designed for expert users that require an advanced technical background, including very specific and intensive training periods.

In contrast, our GUI is being developed focusing basically on the user experience. In fact, the GUI is divided in three different applications: the object identification & task specification GUI, the grasping simulator and the general I-AUV simulator. All of them make use of advanced devices for human-computer interaction (e.g. data gloves, Head-mounted Displays, etc.) and enabling an immersive 3D environment where interaction is more satisfactory for the end-user.

However, this project is still in a preliminary stage and needs further research for a complete validation. So, in the work developed so far, we have analyzed several human-computer interaction devices that could potentially improve the way humans currently interact with underwater robotic systems. We have explored and implemented different possibilities that have to be carefully analyzed, having into account the end-user's requirements and preferences, before its final implementation. Therefore, future lines will mainly focus on a thorough analysis of the different options and the selection and complete implementation of the most suitable solution.

IV. Conclusions and Future lines

This work has presented the first steps towards the development of a *user-friendly* GUI for autonomous underwater intervention missions. We are considering an interface composed of three different applications for object detection and task specification, task simulation, and for the overall supervision of the mission. We claim that the use of new graphics technology and VR devices can greatly increase the overall immersive sensation of the user in the virtual world, thus facilitating its interaction with the robotic system even with little technical knowledge. Therefore, our explored solutions combine different interaction devices such as data gloves for the grasp specification and Head-mounted Displays for immersive visualization.

Our long-term objective is to reach new levels of human-robot interaction in the context of autonomous underwater intervention missions, thus improving the user's satisfaction and performance while using the system.

References

[1] Sheridan, T.B. "Telerobotics, Automation and Human Supervisory Control". MIT Press. 1992.

[2] Yuh. "Design and Control of Autonomous Underwater Robots: A Survey". In Int'l J. of Autonomous Robots 8, 7–24. 2000.

[3] 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.

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

[5] García, J. C., Fernández, J. J., Sanz, P. J., Marin, R., "Increasing Autonomy within Underwater Intervention Scenarios: The User Interface Approach". In Proc. of *IEEE International Systems Conference —4th Annual IEEE International Systems Conference, 2010,* San Diego, CA, USA, April 5–8, 2010. Accepted, pending of publishing.

[6] Stansfield, S.A. "Robotic Grasping of Unknown Objects: A Knowledge-based Approach". International Journal of Robotics Research, 10(4), 314–326. 1991

[7] Cutkosky, M., & Wright, P. "Modeling manufacturing grips and correlations with the design of robotic hands". Pages 1533–1539 of: IEEE International Conference on Robotics and Automation, vol. 3. 1986.

[8] M. Prats, P.J. Sanz and A.P. del Pobil. "A Framework for Compliant Physical Interaction: the grasp meets the task". Autonomous Robots, 28(1), pp. 89-111, 2010.

[9] S. Ekvall and D. Kragic, "Grasp Recognition for Programming by Demonstration". In IEEE Intl. Conf. On Robotics and Automation (ICRA), pp. 748-753, Barcelona, Spain, 2005.

[10] http://www.jmonkeyengine.com

[11] http://www.nvidia.com/object/physx_new.html

[12] http://www.jmonkeyengine.com/wiki/doku.php/complete_features_list

[13] http://developer.nvidia.com/object/physx_features.html

[14] http://www.slizone.com/page/slizone_learn.html

[15] http://www.nvidia.es/object/GeForce_3D_Vision_Main_es.html

[16] O. Belmonte, M. Castañeda, D. Fernández, J. Gil, S. Aguado, E. Varella, M. Nuñez, J. Segarra. In Int. Journal of Future Generation Computer Systems 26, 308-317. 2010.