Distributed Mixed reality for diving and underwater tasks using Remotely Operated Vehicles

Mehdi Chouiten[1],Christophe Domingues[2],Jean-Yves Didier[3],Samir Otmane[4] and Malik Mallem[5]

[1]IRA2 Team,IBISC Laboratory,University of Evry,France Wassa, Boulogne Billancourt, France
[2,3,4,5]IRA2 Team, IBISC Laboratory,University of Evry, France

ABSTRACT

Taking advantage of state of the art underwater vehicles and current networking capabilities, the visionary double objective of this work is to "open to people connected to the Internet, an access to ocean depths anytime, anywhere." Today, these people can just perceive the changing surface of the sea from the shores, but ignore almost everything on what is hidden. If they could explore seabed and become knowledgeable, they would get involved in finding alternative solutions for our vital terrestrial problems – pollution, climate changes, destruction of biodiversity and exhaustion of Earth resources. The second objective is to assist professionals of underwater world in performing their tasks by augmenting the perception of the scene and offering automated actions such as wildlife monitoring and counting. The introduction of Mixed Reality and Internet in aquatic activities constitutes a technological breakthrough when compared with the status of existing related technologies. Through Internet, anyone, anywhere, at any moment will be naturally able to dive in real-time using a Remote Operated Vehicle (ROV) in the most remarkable sites around the world. The heart of this work is focused on Mixed Reality. The main challenge is to reach real time display of digital video stream to web users, by mixing 3D entities (objects or pre-processed underwater terrain surfaces), with 2D videos of live images collected in real time by a teleoperated ROV.

Categories and Subject Descriptors

D.2.11 [**SOFTWARE ENGINEERING**]: Software architectures—Domain specific architectures. H.5.1 [**INFORMATION INTERFACES AND PRESENTATION**]: Multimedia Information Systems—Artificial, augmented, and virtual realities. C.2.1 [**NETWORK ARCHITECTURE AND DESIGN**]: Distributed networks

General Terms

Design, Experimentation

Keywords

Augmented Reality, Mixed Reality, Distributed Architecture, Underwater, Telerobotics.

1.INTRODUCTION

Oceans represent 70% of earth surface, while those who have some knowledge of oceans and seas depths (divers and marine scientists) represent less than 0.5% of world's population. This is not only due to the lack of knowledge but more to distance issues for people living in outback areas and to the cost of safe diving experience. On the other hand, even though underwater equipment has been greatly improved, marine scientists and divers still have to perform a lot of manual repetitive tasks that could be automated. We address this issue in our work by providing an assisted solution to marine wildlife monitoring and by setting a framework able to support extension to other applications. Virtual diving in real time through web teleoperation of a ROV and Mixed Reality is a new, innovative way to discover the undersea world on-line, complementing or replacing scuba diving, giving access to knowledge and discovery of seabed.

The challenge is to mix 3D pre-processed underwater terrain surfaces of distant sites with the video stream provided by the ROV. ROVs are presently operated by a distant operator through an umbilical cable. Nevertheless it appears that technologies and experience already acquired in the field of teleoperation via internet of robots, in general, and more specifically, in the field of operation of underwater robots is now mature enough to seriously consider the development of ROVs teleoperation by Internet ([1], [2], [3] and [4]), which may constitute a technical and technological breakthrough.

In addition to teleoperation as it is meant classically, the main objective of this work is to enrich the user's experience with reliable, real time generated, graphical and textual entities helping to understand the situation in a relatively unfamiliar environment, to ease and speed-up decision taking in such environment and to automate tasks such as marine animals detection and counting in real-time and allowing to create a log for further statistical studies over time.

As an additional feature, the whole architecture is designed to be distributable on several sites. This makes the application more flexible and allows the involvement of multiple users.

2.ROV Teleoperation

In addition to exploration applications, a lot of commercial operational tasks require to dive (ex. underwater structure and boat hull inspection tasks, wildlife monitoring). Industry has already tried to ease these tasks using a variety of technological and functional ways, providing the diver with special suits, detailed seabed map and specific hardware tools.

Even if these improvements helped the divers, there still were in-situ communication problems, no ability to automate the tasks and numerous physiological effects on the divers. Health specialists have studied effects of working in such a high pressure, viscous and weightless environment. Beside the disorientation and affection of tactile-kinesthetic and vestibular systems, some serious medical issues may appear.

We can quote Nitrogen Narcosis, Pulmonary Oxygen Toxicity, decompression Sickness (DCS), Arterial Gas Embolism (AGE), Hypothermia, Barotraumas, etc. Those are discussed in [5], [6] and [7]. The use of ROV (Remote Operated Vehicle) allows avoiding the largest part of these issues (especially on exploration [8] and inspection tasks [3] and [4]). In addition, the ability to operate distantly in a human friendly environment with all necessary information from a set of sensors also improves the speed and the quality of decisions.

3.Augmented Reality Component System

Practically, AR applications which objective is assisting the users often share several common components used in different ways. Due to heterogeneous input devices providing data at different rates and due to different processing algorithms requiring different computation times, research efforts were initiated to offer common frameworks aiming at offering flexible, reusable and customizable components to build AR applications. These efforts have allowed the emergence of component based AR dedicated frameworks.

Amongst the most remarkable ones, we cite Studiers tube [9]. It is based on the concept of distributed scene graphs. Each user has a local scene graph which modifications are propagated through the network to maintain the graph consistency.

One of the most accomplished state of the art frameworks is DWARF [10] (Distributed Wearable Augmented Reality Framework). It is based on interdependent CORBA (Common Object Request Broker Architecture) services. The architecture is decentralized and several applications proving efficiency of the framework have been developed. Some other frameworks are based on the same main concepts such as AMIRE [12] (Authoring Mixed Reality) and MORGAN [11].

The framework on which our application is built is called ARCS [14] (Augmented Reality Component System). ARCS is a component-based framework dedicated to AR. Its components, as classical components [15], can be configured and composed with other components. ARCS uses the signal/slot paradigm (borrowed from user interface libraries) to connect components to each other in order to make them communicate.

Tinmith[13], is a library written to develop mobile AR systems. The communication between modules is made possible by client-server style architecture. A module providing data is the server that listens to clients that request a subscription. When data on the server is changed, the new values are sent to all clients that have registered their interest to this message. The clients can then use the new data to perform the task of the module (e.g. refresh the display). The system is asynchronous and data driven. If there is no new data, no new message will be generated and no action performed by any software module.

In ARCS, every application is described as a set of threads. Basically, a finite state machine,which states represent a specific configuration of the application's data flow controls each thread. Such a configuration is called a *sheet* and contains configuration values for components as well as a list of signal/slot connections. Each change of state in the state machine results in a change of global configuration of components and hence reconfigures connection between components, that is to say the dataflow.

A simple example would be an application with one automaton (one thread), with a given number of *sheets* (eg. each *sheet* representing a given scenario). Here, the state machine's role would be to switch from a scenario to another.

From the technical point of view, ARCS is written in C++ and is based on Qt Library, which already implements the signal/slot concept. ARCS supports XML and JavaScript as scripting languages to describe applications behavior.

The framework also supports distributed components via a specific middleware and webservices in standard formats (XML, JSON). The performance of distributed applications based on ARCS has been assessed with a methodology dedicated to AR distributed systems [19].

4.ROV Teleoperation System

4.1.System architecture

The system is constituted of five main types of sites (network nodes) that are:

- ROV site (usually in the sea / pool);
- ARCS module (Mixed Reality application site);
- 3D repository site where are stored 3D content (e.g: fauna and flora models);
- Web server which hosts the web application;
- User site (can be several users on different sites).

Figure 1 illustrates the relationships and data streams exchanged between the five sites. Except the ROV site, all the other sites may be regrouped depending on the application scenario. The architecture has been built this way to offer more flexibility.

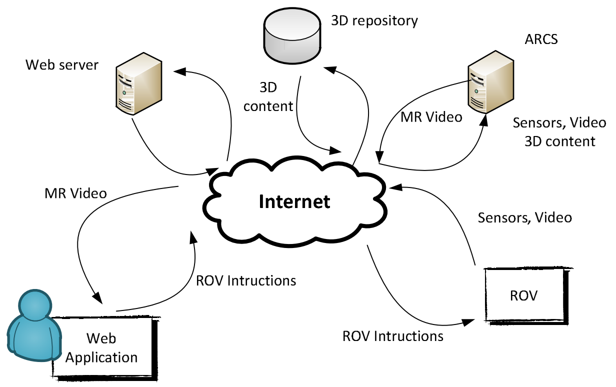

Figure 1. Data streams exchanged between the five sites

Basically, the system works as follows. The user controls the robot via the web user interface. The web GUI allows high level commands that are composed of several instructions transmitted to the ROV. The robot equipped with a set of sensors (including two cameras, but only one can is used in our case) sends a continuous data stream to the ARCS main application, which creates the augmented scene and sends out the final output data (video stream and other information) that is then available to the user who can build a custom view of the scene.

The internal architecture (see figure 3) of the Web Application is divided into three parts. The internal architecture involves as well: a web page written with HTML5, CSS 3 and JavaScript (JS), a PHP script in order to communicate with the ROV using Modbus TCP protocol and a PHP controller. Modbus is a serial communications protocol published in 1979. The controller is designed to manage all data streams between modules:

- *Robot instructions*, which are high level commands, created by users by clicking on the interfaces buttons (e.g. go forward, turn left, etc.). The PHP Modbus script will convert those high level commands to Modbus commands (e.g.: ROV Address + Read/Write code + First memory registry code + Number of bytes + Data to send + CRC16);

- *Robot data, which* are sent by the ROV (e.g. sensors status, error, etc.).

A specific module called Robot Computer Interface (RCI) permits to link the ROV to the network (see figure 3). It is composed of four components:

- Power manager;
- Battery;
- Control and video module: convert LAN/WLAN data into RS485 data;
- Video demodulation: convert PAL/NTSC video signal into a numeric video signal.

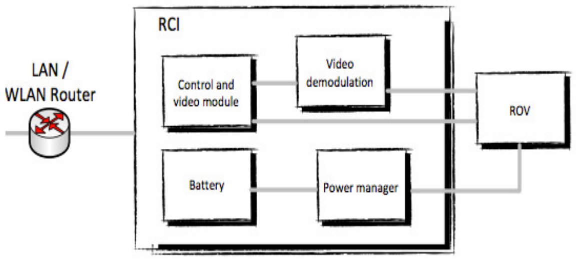

Figure 2. Robot and Computer Interface – Electronic design

The web application can also receive data from ARCS. This is a specific data stream which contains data computed by ARCS application in order to allow a local augmentation if many users are viewing the live video (e.g. add sensors information on the video, build a custom view for a custom user...). The web application will communicate with ARCS through a network (possibly internet). ARCS will allow to mix live video from ROV with 3D contents. The 3D contents is stored in a MySQL database. The ARCS application can also perform most of the known computer vision state of the art algorithms (more than 150 reusable components are available).

4.2.Web application user interface

The Web Application is written in PHP, HTML5, JavaScript and CSS3. The user interface (see figure 4 below) is a human machine interface in order to send commands to the ROV (teleoperation), supervise sensors data from ROV (virtual dashboard) or enable a user to interact with other features (chat with other divers or get more information via web services, such as weather, maps, geo-localized pictures, historical information about the area…) in the web page.

The web view is divided in six parts:

Mixed Reality live video from ARCS application;
Navigation panel to control the ROV (Moves, ROV lights, front or back cameras and Mixed Reality On/Off switches);
HTML5 2D canvas to plot a virtual trajectory;
A chat to communicate with other users;
Data panel which displays data from ROV sensors;
Web services panel which displays or shares data from diving site.

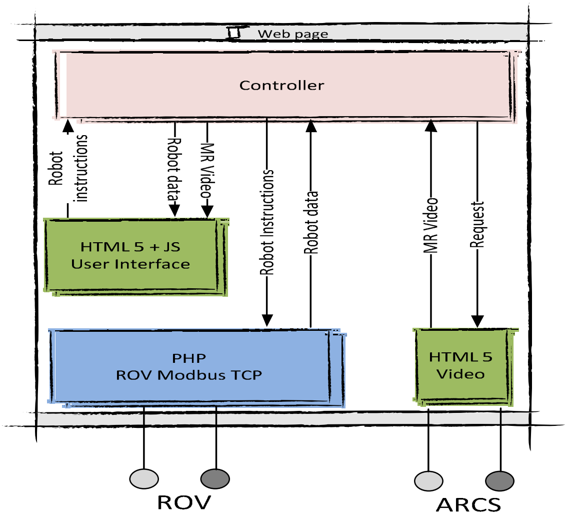

Figure 3. Software architecture of the web application – Data stream between modules

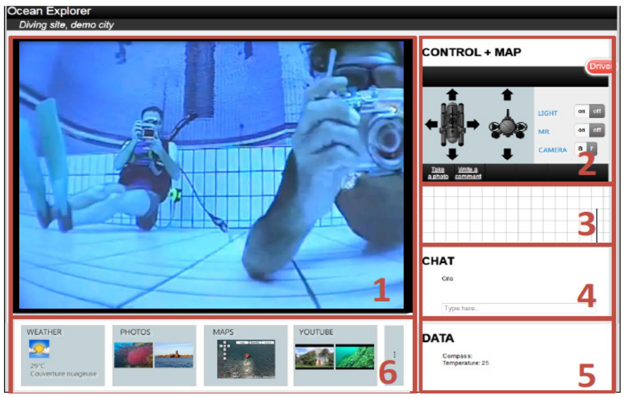

Figure 4. Web application – Graphical User Interface

4.3.Multimodal application user interface

The user interface of the multimodal application also permits to send commands to the ROV (teleoperation), supervise sensors data from ROV (virtual dashboard) using natural gestures. Users control a virtual ROV using the markers placed on the flystick device (Figure 5). Those markers are tracked with the infrared cameras placed on both sides of the screen.

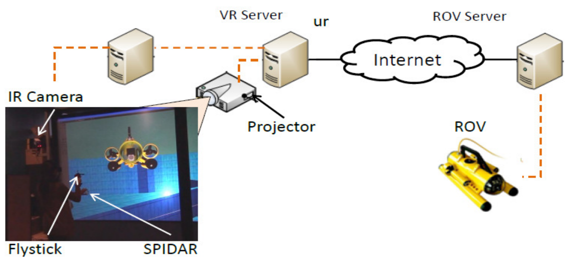

Figure 5. Hardware solution for multimodal application

This system tracks the user’s positions, so we can reproduce it on the ROV. Hence, by manipulating a virtual ROV, the operator is controlling the real one. We also use the metaphor of pulling a rope. Indeed, during his/her movements, the user must hold the trigger of the fly stick pressed. When he/she releases it, the ROV stays in its last position and the user is free to move. At the next displacement, the ROV starts from its last position and not from the user’s current position. Therefore we do not apply directly the user's position to the one of the ROV, but we add the movement vector performed by the user to the last position of the ROV. Users can manage the robot’s features (camera switch, activate lights) or to show instructions for effective usage. Mixed Reality (overlay both virtual and real worlds) is used here to help users to manipulate the distant ROV. Indeed, assembling the different issues due to human, environment and teleoperation factors, we observed some technical constraints (Loss of ROV maneuverability, transmission delay/stop, etc.) affecting the application’s usage (navigation precision, safety of the robot and spatial awareness for the diver).

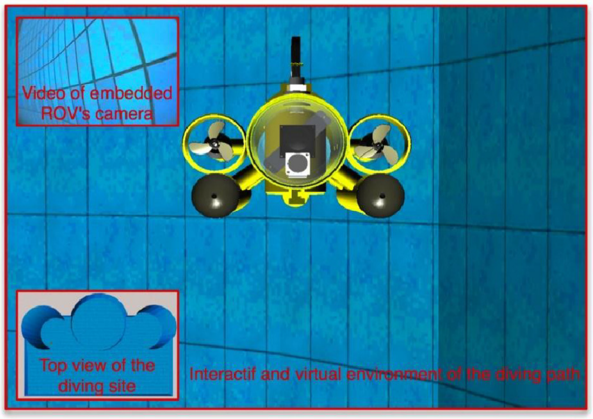
Video of embedded ROV's camera
Top view of the diving site
Interactif and virtual environment of the diving path

Figure 6. Multimodal application – Graphical User Interface

5.ARCS-based module

Built following the standard ARCS application pattern which itself is based on the generic Augmented Reality application pattern introduced by Asa McWilliams [17], the application is thought to support multiple distant users.

The architecture of the middleware that has been designed for ARCS can be found in [16].

The model of the application is data driven. Video stream coming from the ROV implies that for each new frame processed by the video stream reader, the tracking component computes visible 3D entities and their position (3D registration). ARCS offers a framework to support several 3D rendering engines and already supports OGRE 3D and OpenInventor. Depending on the chosen rendering engine, the corresponding component creates then the final representation blending the raw input with 3D contents taken from a potentially distant database. This final output is then converted to a video stream, which is sent to the users. It is important to notice that the output manager supports different kind of data that can be sent separately to users who will chose different display modes and who may be interested in different textual information.

Figure 7 illustrates an example with OGRE as a rendering engine and video stream as the unique input. We also assume in this example that the user site hosts the webserver.

The state machine is a specific component of the general application (written in XML or built via the application designer that generates the XML description), it manages the connections between the other components and is initialised by the profile (a configuration file containing given values to application constants).

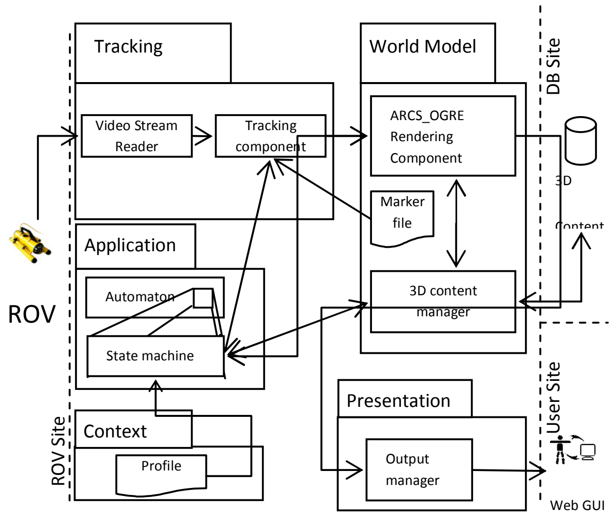

Figure 5. ARCS main application architecture (View based on McWilliams generic architecture)

As stated in the introduction, the architecture of the whole system is meant to be distributed (figure 1). In addition, the ARCS module itself can be built as a distributed application constituted of native ARCS components, exogenous components interfaced with ARCS, and even external webservices (figure 8).

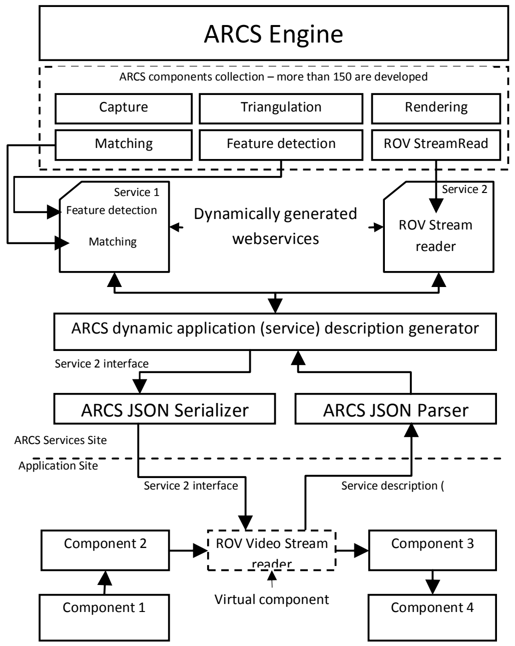

Figure 8. Example of an application using ARCS components as services (ARCS applications can be accessed as services and can also use external webservices)

6.Mixed Reality video streaming

In order to transmit the MR video over the Internet the ARCS output manager needs to digitalize the created MR video (ROV video + 3D content). For encoding and delivering, we have chosen Ogg format. Ogg is an open standard, open source–friendly, and unencumbered by any known patents. Ogg is based on three technologies: a container (Ogg), a video codec (Theora) and an audio codec (Vorbis). The output manager will stream over the Internet using the HTTP protocol on 8080 port. This protocol is chosen to avoid firewall issues and proxy servers filtering that usually don't block the standard HTTP traffic necessary for people using the web.

7.Scenarios

The application being designed to be distributed and relatively generic, different scenarios have been identified to explore the feasibility of the global solution. ARCS has already been assessed and proved to be real time capable for state of the art AR applications. However, applications involving network communications depend not only on the used framework but also on their consumption of network resources.

The simplest scenario is to have a ROV that communicates with the operator site on which all the other sites are regrouped. Here we have a unique user, and no intricate network communication. Performance issues rely only on the operator machines capabilities depending on the kind of application.

In the general case, each site is on a different machine and there may be several users. The only point where serious network performance issues emerge is when having multiple user terminals. This issue may be avoided by multicasting the output to users.

8.Applications and tests

8.1.Set-up of the teleoperation system

Tests have been performed on November 2011 at UCPA Aqua 92 diving pit at Villeneuve-la-Garenne. Those tests were performed to detect potential waterproof sealing problems in the ROV and also the WWW demonstrator with the use of ARCS. The ROV is wired (umbilical cable) to its own box (see figures 9 and 10). This box permits to charge the ROV batteries and connect it to LAN/WLAN network (for control and video). For those tests, we have set-up a local network using a router. A computer was connected to this network. This computer (x86 based-Windows XP)was running ARCS and the web application with the use of a local Apache server(see figure 1).

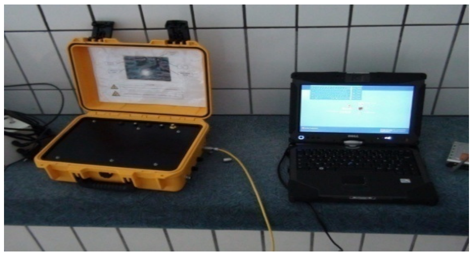

Figure 9. On the left: ROV box – On the middle: the computer used as a client and server – On the right: the Wifi router used for the test

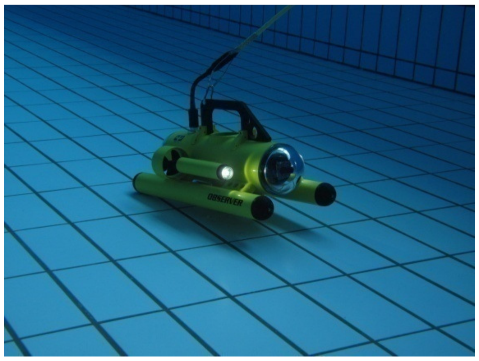

Figure 10. The ROV in the UCPA Aqua 92 pit at 5 m depth

8.2.Mixed Reality with ARCS

Real-time scene augmentation has been performed following three custom scenarios. First is contextual scene augmenting (figure 11), that only consists in adding virtual entities for immersive and decorative goals (ex. Adding specific species of fish depending on the environment or adding virtual funny fish in a swimming pool, etc.).

The second test is based on marker tracking (ex. Display information on a marked zone or object such as a pipe or a wreck).

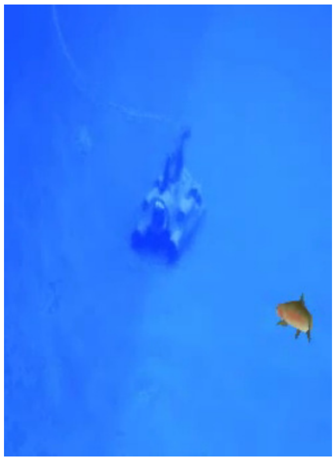

Figure 11. Markerless contextual augmentation

In our example (figure 12), those markers are set on buoys. Applications such as collaborative treasure hunting game and virtual fish encyclopedia have been developed (Figure 13). This virtual encyclopedia provides 3D models of fish, and information from a database such as name, natural environment and zones of the world where it can be found. Other information such as maximum size, maximum depth, etc. are available in the database and can also be displayed. One can easily build new applications with very little effort of development since ARCS offers great reusability of components and parametrisation ability. The third and last one is based on natural features tracking (ex. Automatic fish classification). A preliminary test has been made with the SURF [18] (Speeded-Up Robust Features) component of ARCS (which is based on OpenSURF library), but a more complex application is being built based on several descriptors.

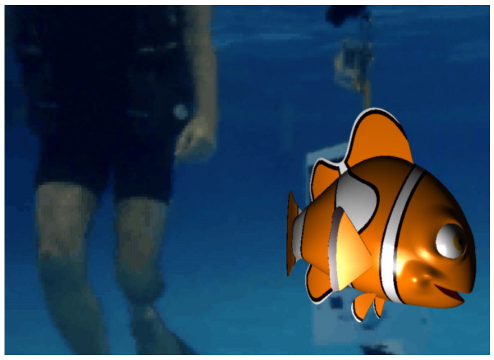

Figure 12. Marker based approach. It’s used to display animated entertainment content in the UCPA aqua 92 pool

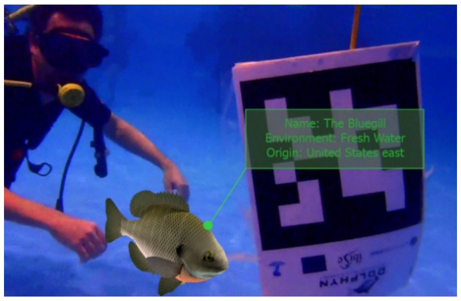

Figure 13. Marker-based pedagogical purpose application

8.3.ROV Localization

The location of the ROV's camera position and orientation in real time can be done using several methods. These methods are based on the knowledge of the intrinsic camera parameters, as we suppose that the extrinsic parameters that constitute the pose of the camera may change over time. The principle of these methods is to use markers (or coded targets) in the real world to calculate the camera’s position and orientation. In this case, instead of trying to directly recognise the objects in the real world, we can only recognise those different markers placed in specific locations in the diving site (see Figure 14). These markers are recognised by the ROV's camera while exploring the diving path, using algorithms of the MR module.

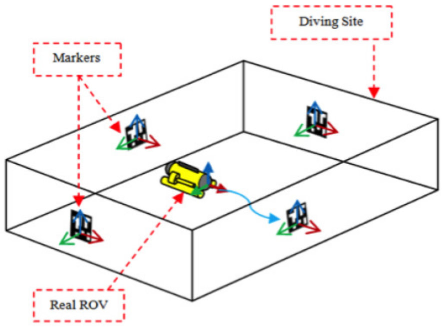

Figure 14. Localization system using 2D markers

8.4.Marine wildlife monitoring

One other application, proving the abilities of our architecture to support state of the art computer vision algorithms is the Tethys Project. Its objective is to detect marine animals and specifically fish in real time. The application is composed of distinct components allowing different uses.

After acquiring the video stream from a stream reader (a result similar to what is shown in figure 15), one of the most important components of the application is the fish detector. It is based on regions of interest detection algorithms taking in consideration colors, texture and shape of the region of interest. A basic non refined blob detection is shown in Figure 16.

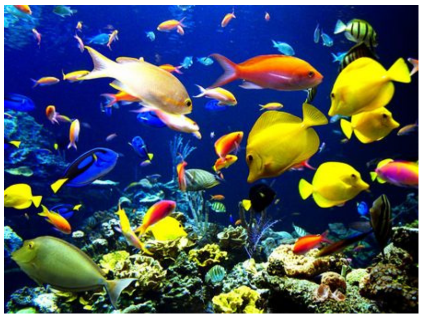

Figure 15. Marine wildlife application: Test image

The second step is to refine the results with morphology algorithms (erode or dilate and then smooth) and eliminate irrelevant detected regions. The user can also define specific characteristics of the fish to detect (color, size…) as shown in figure 13. Then, for classification and identification purposes, each region of interest is described with a features vector including statistical moments of the shape, color information extracted from the color histogram and texture descriptors.

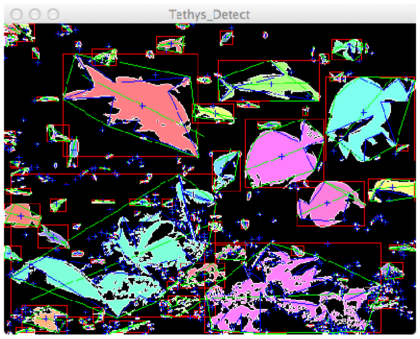

Figure 16. Marine wildlife application: Detected regions of interest

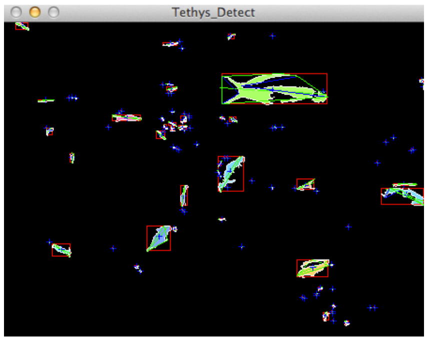

Figure 17. Marine wildlife application: example of specific detection (reddish colored fish)

The shape descriptors are usual blob descriptors (centroid, area, elongation, and statistical moments). Color descriptors are also based on statistical moments. For texture descriptors, several options have been studied. Haar-like features being very sensitive to rotation, we decided to combine statistical properties with spectral and structural texture analysis methods.
The feature vector is then matched with the features vectors in the known fish database (the application has been tested only with 3 different species at the moment). A score is then computed and if it satisfies a minimal threshold, the detected fish is identified as a member of a specific species.

This application is still in development and it still needs implementation of some of the feature descriptors and validation with a dataset of decent size.

The final goal is to identify fish and augment the scene with in real time with information about the identified fish using any distant tablet or even smartphone with access to Internet. Within this project, a special device has also been developed to visualise augmented reality contents underwater [20].

9.Conclusion and Future work

We have specified and built a customizable software and network architecture in order to enable enriched teleoperation of a ROV involving a wide variety of Mixed Reality content and computer vision applications. Augmented scene is computed on a distant site using any relevant ARCS-based application. This makes the system more flexible and allows the involvement of multiple users. We have set-up three scenarios. The first was to augment the distant scene by adding 3D content for decorative purpose. The second scenario is a marker based approach. And finally, a complex scenario based on natural features (fish classification). The three previously described applications are only to illustrate a proof of concept. The possible applications are very numerous. There are way more scenarios to develop. As an example, one useful and interesting application is the 3D reconstruction that we are working on. We already explored this research area using a monocular SLAM (Simultaneous Localization And Mapping) approach providing valuable feedback indoors. Taking advantage on the ARCS decentralised hybrid architecture [14], a collaborative multi-robot and multi-sensor (including sonar-based approach) will be explored for real-time 3D reconstruction of seabed and submarine natural environments. We are also investigating the use of VR devices as optical tracking and haptic devices to control more easily and feel the remote ROV.

10.Acknowledgements

Some parts of the work described in this article are made for the Digital Ocean Project (FP7 262160) which is a collaborative project funded through European Community's Framework Program FP7.

References

[1] Maza M., Baselga S., Ortiz J., “Vehicle Teleoperation with a Multisensory Driving Interface”, in Climbing and Walking Robots Journal, Springer, p. 437-445, 2005.
[2] Roston J., Bradley C., Cooperstock JR, “Underwater window: high definition video on VENUS and NEPTUNE”, in IEEE OCEANS 2007, p. 1-8, 2007.
[3] Sattar J., Dudek G., “Underwater Human-Robot Interaction via Biological Motion Identification”, in Robotics: Science and Systems V, June-July 2009, Seattle, WA, USA.
[4] Jenkyns R., “NEPTUNE Canada: Data integrity from the seafloor to your (Virtual) Door”, in IEEE OCEANS 2010, p. 1-7, 2010.
[5] James T. Joiner, NOAA Diving Manual: Diving for Science and Technology, 4th ed., February 2001.
[6] Stanley, J.V. Scott, C., “The effects of the underwater environment on perception, cognition and memory,” in vol. 3, pp. 1539-1548 [OCEANS '95. MTS/IEEE, Challenges of Our Changing Global Environment Conference Proceedings].
[7] Morales-Garcia, R., Keitler, P., Maier, P., Klinker, G.: An Underwater Augmented Reality System for Commercial Diving Operations. OCEANS 2009 MTS/IEEE, Conference Proceedings (2009)
[8] Bruzzone G., Bono R., Caccia M., Coletta P., Veruggio G., “Internet-based teleoperation of the Romeo rov in the arctic region”, Manoeuvring and control of marine craft 2003 (MCMC 2003): a proceedings volume from the 6th IFAC Conference, Elsevier Science Ltd, 2004.
[9] A. Fuhrmann, G. Hesina, Z. Szalavari, L. M. Encarnacao, M. Gervautz, and W. Purgathofer. “The studierstube augmented reality project”, in Presence: Teleoperators and Virtual Environments, volume 11, Feb 2002.
[10] M. Bauer, B. Bruegge, G. Klinker, A. MacWilliams, T. Reicher, S. Riss, C. Sandor, and M. Wagner. “Design of a component-based augmented reality framework.”, in Proceedings of the International Symposium on Augmented Reality (ISAR), Oct. 2001.
[11]J. Ohlenburg, W. Broll, and A.-K. Braun. Morgan: “A framework for realizing interactive real-time AR and VR applications”, in IEEE VR, Mar. 2008.
[12] R. Dörner, C. Geiger, M. Haller, and V. Paelke, "Authoring mixed reality - a component and framework-based approach", in Proc. IWEC, 2002, pp.405-413.
[13] W. Piekarski and B. H. Thomas. “Tinmith-evo5 - an architecture for supporting mobile augmented reality environments.”, in International Symposium on Augmented Reality, Oct. 2001.
[14] J. Didier, S. Otmane, and M. Mallem. “A component model for augmented/mixed reality applications with reconfigurable data-flow.”, in 8th International Conference on Virtual Reality (VRIC 2006), pages 243–252, Laval (France), April 26-28 2006.
[15] C. Szyperski, Component Software - Beyond Object-Oriented Programming, second edn, Addison-Wesley, Harlow, England, 2002.
[16] M. Chouiten, J. Didier, and M. Mallem. “Component-based middleware for distributed augmented reality applications.”, in Proceedings of the 5th International Conference on Communication System Software and Middleware (COMSWARE '11). ACM, New York, NY, USA
[17] A. Macwilliams, T. Reicher, G. Klinker, B. Bruegge. Design Patterns for Augmented Reality Systems. In Proc. International Workshop Exploring the Design and Engineering of Mixed Reality Systems - MIXER 2004
[18] H. Bay, A. Ess, T. Tuytelaars, and L.J.V. Gool, “Speeded-Up Robust Features (SURF)”, presented at Computer Vision and Image Understanding, 2008, pp.346-359.
[19] M. Chouiten, J-Y. Didier, M. Mallem: Distributed Augmented Reality Systems: How Much Performance is Enough? ICME Workshops 2012: 337-342.
[20] Christophe Domingues, Samir Otmane, Alain Dinis: A new device for a virtual or augmented underwater diving. 3DUI 2012 : 141-142