



**HAL**  
open science

## Distributed Mixed Reality for remote underwater telerobotics exploration

Mehdi Chouiten, Christophe Domingues, Jean-Yves Didier, Samir Otmane,  
Malik Mallem

► **To cite this version:**

Mehdi Chouiten, Christophe Domingues, Jean-Yves Didier, Samir Otmane, Malik Mallem. Distributed Mixed Reality for remote underwater telerobotics exploration. 14th Virtual Reality International Conference (VRIC 2012), Mar 2012, Laval, France. (elec. proc), 10.1145/2331714.2331716. hal-00675319

**HAL Id: hal-00675319**

**<https://hal.science/hal-00675319v1>**

Submitted on 29 Feb 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Distributed Mixed Reality for remote underwater telerobotics exploration

Mehdi Chouiten, Christophe Domingues  
Jean-Yves Didier, Samir Otmane, Malik Mallem  
RATC Team, IBISC Laboratory  
University of Evry, France

mehdi.chouiten@ibisc.fr - christophe.domingues@ibisc.fr

## ABSTRACT

The visionary 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 eventually involved in finding alternative solutions for our vital terrestrial problems – pollution, climate changes, and destruction of biodiversity and exhaustion of Earth resources. The introduction of Mixed Reality and Internet in aquatic activities constitutes a technological rupture 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. 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 ROV video camera. ROVs are presently operated through an umbilical cable by a distant operator. 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 and to ease and speed-up decision taking in such environment.

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, a lot of commercial operational tasks require to dive (ex. underwater inspection tasks). Industry has already tried to ease these tasks in a variety of technological and functional ways, providing the diver with special suits, detailed seabed map.

Even if these improvements helped the divers there still were in-situ communication problems, no ability to automate the tasks and physiological effects on the divers. Effects of working in such a high pressure, viscous and weightless environment are very numerous. 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 Studierstube [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.

### 4. ROV Teleoperation System

#### A. System architecture

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

- ROV site (usually in the sea / pool);
- ARCS main 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).

The figure 1 illustrates the relation and data streams between the five sites. Except the ROV site, all the other sites may be regrouped depending on the application scenario.

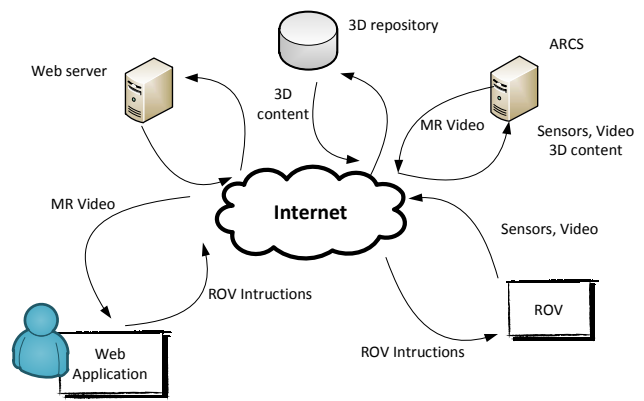


Figure 1. Data streams between the five sites

Basically, the system works as follows. The user controls the robot via the web user interface. The robot equipped with a set of sensors (including two cameras, but only one can be used) 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 2) of the Web Application is divided into three parts. The internal architecture involves as well: a webpage 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 is 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 by 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.

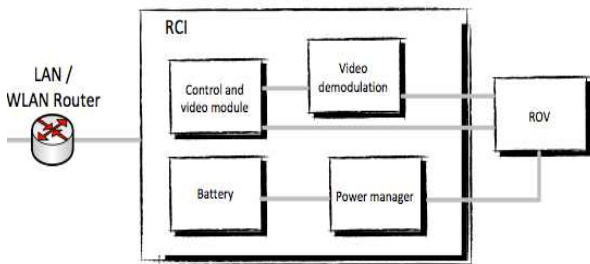


Figure 2. Robot and Computer Interface – Electronic design

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). The web application will communicate with ARCS through the network (eventually internet). ARCS will allow mixing live video from ROV with 3D content. The 3D content is stored in a MySQL database.

## B. Web application user interface

The Web Application is written with 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...) in the web page.

The web application is divided in six parts:

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

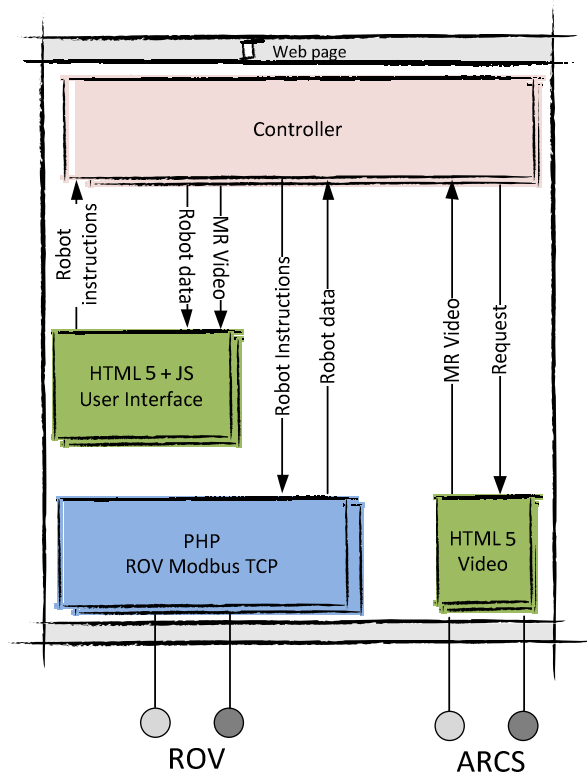


Figure 3. Software architecture of the web application – Data stream between modules

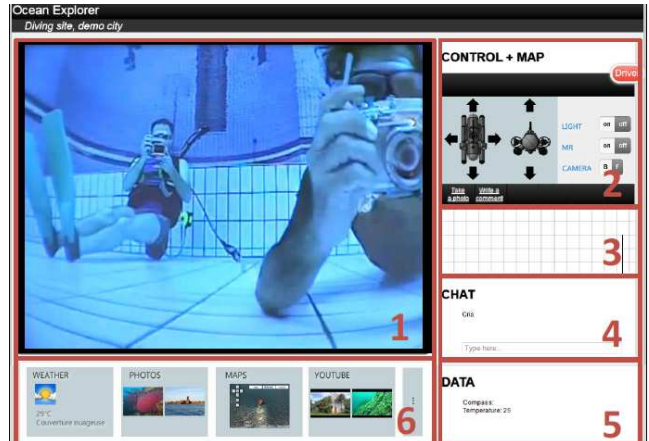


Figure 4. Web application – Graphical User Interface

### C. ARCS main application

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 recalculates 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 content taken from a 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 are interested in different textual information.

Figure 5 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 webservice.

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

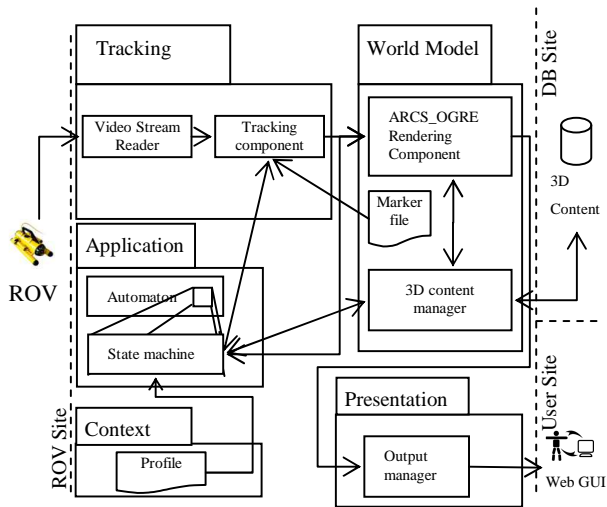


Figure 5. ARCS main application architecture

### D. 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 capable of traversing any firewall or proxy server that lets through standard HTTP traffic.

### 5. Scenarios

The application being designed to be distributed and relatively generic, different scenario 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 is avoided by multicasting the output to users.

### 6. Preliminary Tests

#### 6.1 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 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 6 and 7). 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).

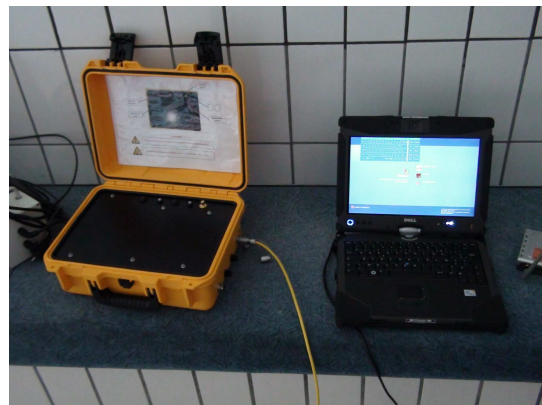


Figure 6. On the left: ROV box – On the middle: the computer used as a client and server – On the right: the Wifi router used for the test



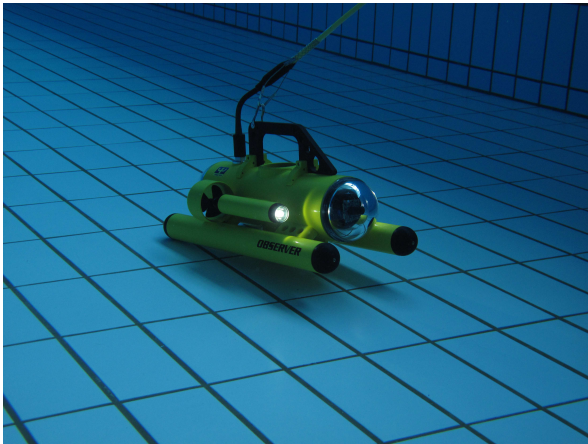


Figure 7. The ROV in the UCPA Aqua 92 pit at 5 m depth

## 6.2 Mixed Reality testing with ARCS

Real-time scene augmentation has been performed following three custom scenarios. First is contextual scene augmenting (figure 8), 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.). Second test is based on marker tracking (ex. Display information on a marked zone or object such as a pipe or a wreck). In our example (figure 9), those markers are set on buoys. 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).



Figure 8. Markerless contextual augmentation

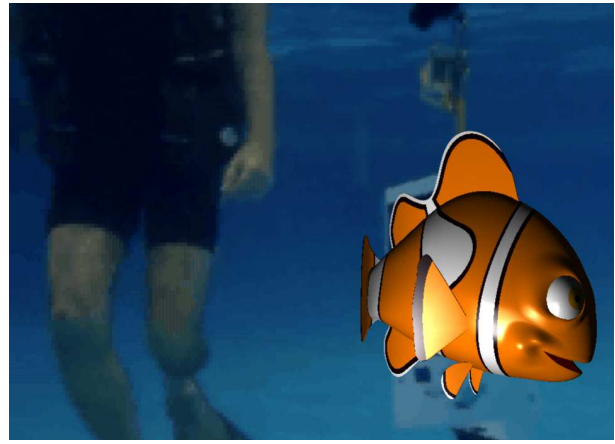


Figure 9. The marker based approach. The marker is used to display entertainment content in the UCPA aqua 92 pool

## 7. Conclusion and Future work

We have specified and built a customizable software and network architecture in order to enable enriched teleoperation with a ROV involving Mixed Reality content. 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 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 decentralized hybrid architecture [15], 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 distant remote ROV.

## 8. Acknowledgements

This project (Digital Ocean - FP7 262160) is a collaborative project funded through European Community's Framework Program FP7.

## 9. 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. Otmame, 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.