INTERACTIVE MULTIMEDIA FOR ENGINEERING TELE-OPERATION

Benjamin Jailly(1), Marius Preda(2), Christophe Gravier(1), Jacques Fayolle(1)

(1) Université de Lyon, F-42023, Saint-Étienne, France;
Université de Saint-Étienne, Jean Monnet, F-42000, Saint-Étienne, France;
Télécom Saint-Étienne, école associée de l’Institut Télécom, F-42000, Saint-Étienne, France;
Laboratoire Télécom Claude Chappe (LT2C), F-42000, Saint-Étienne, France.
(2) ARTEMIS, Télécom SudParis, Institut Télécom
{firstname.name}@telecom-st-etienne.fr
{firstname.name}@it-sudparis.eu

ABSTRACT

Online Engineering allows users to perform tele-operation over the Internet. It is used in collaboratories, distance learning curriculum, and remote maintenance processes. Tele-operation over the Internet is however restrained by the development time and cost of ad hoc solutions. These solutions a) are hardly reusable, b) present a low fidelity of the human computer interface, and c) barely enable pervasive accesses (standalone clients or Web forms). We present in this paper a novel approach for building tele-operation Human Computer Interfaces based on interactive multimedia. The end-user commands the remote device using interactive elements in the multimedia interface. The feedback information is seen as a combination of natural video content, produced by an IP camera capturing the instrument and synthetic graphics corresponding to elements of the device input interface (control panel). The proposed architecture fosters the reuses and gives a high level of interoperability between command terminals, since the interface can be displayed in any terminal able to lay out multimedia content.

Index Terms— Remote Laboratories, Interactive Multimedia, Human Computer Interface, MPEG-4 BIFS, Computer Vision

1. INTRODUCTION

Remote laboratories give the ability to perform remotely operations on real devices over the Internet. On the equipment side, the infrastructure has to perform the control of the device. On the user side, the infrastructure needs to provide a Human Computer Interface (HCI later in the paper) that allows interactions with the remote devices and ensures the feedback information from the device. Remote control of devices is nowadays employed in different fields, but especially in:

• Learning institutes mainly for distance learning curriculum,
• Industries for maintenance operations, production controls, etc.,
• Research institutes to manage experimentations with researchers from various geographic locations (collaboratories).

Several studies have been conducted in this field, but few of the proposed solutions in the literature grant for reusable developments [1]-[2]. The hard taste of HCI construction is a story deadlock for remote laboratories development due to resources costs. We can also notice the lack of fidelity of the HCI’s representation of the device. Usually, a web form is used to post experiment parameters, which obviously do not foster the “sense of being there” [3].

We propose in this paper a novel approach to build generic HCIs of devices for remote laboratories based on interactive multimedia. Interactive elements in the multimedia allow triggering some of the remote device functionalities and the feedbacks are displayed in the same multimedia presentation as a natural video. Our system is also capable of identifying one device among a corpus of available instruments, leading in automatic adaptation of the interface to the corresponding instrument.

The paper is organized as follows. The second section presents the architecture of the proposed system, the third section shows how to build interactive multimedia for remote control of devices and the fourth one shows how to automatically identify the instruments. We conclude the paper and indicate future developments on the fifth section.

2. GLOBAL APPROACH

2.1. Using interactive multimedia as a command interface
Some solutions in the literature use IP camera in order to obtain a video feedback of the results of the bench test such as reported in [4] and [5]. Others like [6] and [7] display results in the Web application. We can then identify two global approaches for the user to interact with the remote device. The two approaches are illustrated in Figure 1:

a) Exclusive applicative interaction: commands are relayed to the remote device through the middleware via a web application and results are exposed in the same application.

b) Mixed interaction: commands are relayed to the remote device through the middleware via a web application and results are exposed by a video feedback

The usage of multimedia as a control and monitor interface can possibly lead to a paradigm shift in online engineering.

- It improves the representation of distant devices, capturing parts of the device itself.
- It decreases development costs since the HCI does not need to be programmed any more.
- It improves pervasive situations and interoperability between different types of devices. Indeed, every kind of terminal able to display multimedia content (smartphones, IPTV, tablets, PCs, etc.) can be used as a command terminal.
- A loose coupling between instruments and middleware is obtained because all software developments are device dependent.

### 2.2. General architecture

Figure 3 illustrates the global architecture of our system.

**Figure 3. Global architecture**

The IP camera delivers encoded video. The processing block (PB), presented in details in Section 4, identifies the instrument present in the video stream by comparisons with available devices in the database and computes the position of this instrument in the images. Instruments storage in the database (DB) is based on the instrument’s image features. PB also adapts the video by cropping the region of interest, which in general corresponds to the device’s screen. Let us note that the image processing operations in the PB are effectuated in the non-compressed domain, therefore a video decoder and encoder library based on FFmpeg [30] is integrated. Once the instrument is recognized, the corresponding graphical interface is streamed by using the broadcaster block (BB).

By interacting with the graphical interface, the end-user sends requests to the command interpreter (CI). If necessary, the CI can send an update signal to the multimedia broadcaster BB. Such situation occurs when adding graphic elements such as textual information, changing the color of buttons when pushed, etc. Then it relays the commands to the remote device.

### 3. COMPOSITION OF THE MULTIMEDIA INTERFACE

In this paper, we propose a novel approach for interacting with remote devices. This approach is illustrated in Figure 2. The usual web application is no longer required. The commands are relayed to the remote device through the middleware via an interactive multimedia graphical interface. Feedbacks and results are exposed in the same multimedia interface. An interactive multimedia is a multimedia that gives interaction possibilities to the multimedia’s consumer.

**Figure 1. a) Exclusive applicative interaction**

**Figure 1. b) Mixed interaction**

**Figure 2. Global mechanism of the proposed system**

**Figure 3. Global architecture**
In this section, we will present how to construct a multimedia HCI needed to control the remote devices.

3.1 Scene content

Figure 4 presents an instrument in its typical environment. In order to construct the multimedia HCI of the instrument, the different kinds of existing regions in the scene need to be identified. We encountered 3 types of regions for our use case:

- The video regions where objects are likely to change during time, corresponding to the feedback from the instrument (the display of the instrument in Figure 4 for instance),
- The interactive regions enabling functionalities of the instrument, (the buttons in Figure 4 for instance),
- The background (static unchanged pixels elements over the session).

Based on advantages exposed in a previous study on multimedia descriptor languages [9], we decided to use MPEG-4 BIFS, Binary Format for Scene, to describe and compose the HCI of the instrument.

3.2 Multimedia composition with MPEG-4 BIFS

MPEG-4 BIFS, published in Part 11 of the MPEG-4 standard, is a multimedia description format that specifies multimedia objects of different natures in space and in time, from natural videos, textures, audio objects to 2D or 3D synthetic objects. Using MPEG-4 BIFS, one can represent a scene as a hierarchical dynamic tree, where leaves are audiovisual objects, as illustrated in Figure 5. When designing complex behaviors, branches of this tree can be created, modified or deleted as demonstrated by Tran et al. [10].

MPEG-4 proposes as well a textual format based on XML for representing the scene, called XMT, eXtented MPEG-4 Textual Format. This textual form facilitates editing the scene description, via a basic XML parser and a dedicated compression schema is offering efficient representation.

Since MPEG-4 BIFS allows describing scenes with any kind of media, the video region of our HCI can thus be encoded with MPEG-4 AVC (Advanced Video Coding), the latter offering very powerful video compression [11]. Section 4 of this paper presents how we construct the video part of the scene. The background of the scene is unchanging with time; therefore we can represent it with a simple image, such as a JPEG image.

In order to imitate inputs of the instrument (buttons, knobs, etc.), we use MPEG-4 BIFS’ interactive elements named Sensors. Because BIFS can embed ECMAScript, we are able to link Sensors with AJAX requests, Asynchronous JavaScript and XML, and then send a specific command to the device via the middleware.

MPEG-4 BIFS also provides a streaming environment, that is to say allowing the scene descriptor to be streamed as any other media, in opposition to the other scene description formats where the complete scene has to be downloaded before the scene can be viewed [12]. This property gives the ability to stream a BIFS scene over the Internet, via UDP over RTP protocol. One can then just connect to an URL via a MPEG-4 player and manipulate the remote device.

An authoring tool is provided with our system to help users designing new interfaces. The user has to provide an image corresponding to the background. He also has to specify interactive elements positions, sizes and natures (knobs, buttons, etc.). For each interactive element, the user has to choose which Interchangeable Virtual Instrument (IVI) [13] function stored in the database is linked to it. The IVI standards define driver architecture, set of instruments class and software components for instrument interchangeability. Finally he provides the position and size of the interactive elements.
of the feedback, i.e. the video captured by the camera. The system is building the interactive multimedia file based on these user’s settings.

4. AUTOMATIQUE INSTRUMENT RECOGNITION

In this section, we will present how we can reconstruct the instrument feedback, seen as a natural video for the end-user and integrated in the multimedia interface.

4.1 Situation scenario

To clarify why we need to recognize the instrument and calculate its position in the video stream, let us consider the following scenario.

Bob is in maintenance situation of a device in the USA. He is unfortunately not competent enough to realize one maintenance operation. He calls Alice, located in England, who is able to effectively do the operation. He streams the video of the instrument to our system with his smartphone. The system identifies the instrument and delivers the proper interface to Alice. Since he is shaking a bit with his hand, the video is not stable, so the system keeps calculating the instrument positions in order to crop the good region of the stream to broadcast it to Alice. Alice can then realize the maintenance operation with a stable video feedback from the instrument by clicking buttons in the multimedia interface. Recent advancements in object identification techniques conducted us to analyze the pertinence of using local features techniques in the case of instruments.

4.2 Extraction of local features

Tuyltelaars et al. define a local feature as “an image pattern which differs from its immediate neighborhood. It is usually associated with a change of an image property or several properties simultaneously, although it is not necessarily localized exactly on this change [14]”. We call extractor or detector the method that allows detecting local feature in an image and descriptor the data measures centered on the local features.

Most of features extractions rely on contour methods (shape intersections, point of high bending, etc.) or on image intensity based methods (“Hessian-based”). Among the famous descriptors, we can cite Scale Invariant Feature Transform (SIFT) [15], the Difference of Gaussian (DoG) descriptor [16], or the Histogram of Gaussian (HoG) descriptor [17], etc.

In 2008, Bay et al. [18] present the Speeded-Up Robust Features (SURF) descriptor. Tuytelaars et al. categorize this descriptor as efficient, because its implementation privileges the execution speed, SURF being 5 times faster than DoG [14]. The SURF descriptor is scale and rotation invariant and is presented in the form of a 128-dimensional vector.

All the devices from the database have associated SURF descriptors. When the camera captures a device, the descriptors are extracted and compared with those in database. Due to the computation time and the real-time constraint, the SURF descriptor is computed for one image out of two. The comparison between descriptors is based on a simple Euclidian distance. The identified device is the one presenting the lowest Euclidian distance.

4.3 Device’s pose estimation in the video stream

The tracking of rigid objects consists in following their position and orientation within a sequence of images. Several tracking techniques were proposed for Augmented Reality, which has to answer the problem of virtual/time alignment. We can regroup those techniques in two main fields: tracking with marker and marker-less tracking.

The tracking with marker consists in positioning a target object in a scene and detect it every image. This target object should ideally be detectable with the minimal image processing in order to increase the system efficiency. ARToolKit [19] is a famous example of free C library for building augmented reality applications.

Tracking methods with markers are robust and efficient, but they suffer of an essential drawback: the marker itself. Using marker would entail in a hard calibration phase when designing a new remote instrument, since it would mean knowing the relative positions of interactive element from the marker position when building the interactive multimedia content. Because the device is not tagged in advance, we decided to not use such tracking method.

There are several marker-less augmented reality systems reported in the literature. We can mention the BazAR algorithm [20]-[21], published for the first time in 2006. This algorithm is based on randomized trees as a classification technique of wide baseline matching of feature points. OnlineBoosting [22] is another marker-less method. The extraction of feature points is based on a Haar filter response. PTAM [23] is also presented in 2006 and is based on the Simultaneous Localization and Mapping (SLAM) algorithm.

In our work, since SURF is used to recognize the device in the frames, we also use SURF to calculate the pose estimation. The local feature points detected are matched with the local feature points of the reference image stored in the database using the Fast Library for Approximate Nearest Neighbor [24]. It is then possible to estimate the position of the instrument in the video stream thanks to the Zhang algorithm [25]-[26]. Knowing the position of the device in the video stream allows us to crop the display of the instrument, encode and stream it to the final user with the whole interactive multimedia content. Figure 6 is a capture of the interactive multimedia HCI streamed to the client in a
5. CONCLUSIONS AND FUTURE WORKS

In this paper we presented a novel approach to build interactive multimedia for controlling devices in remote laboratories. Our proposed system uses MPEG-4 BIFS to build the interface and local invariant descriptors to identify and compute the position and orientation of the instrument present in front of the camera.

This generic approach has the advantage of decreasing the time needed for setting up the system. It increases pervasive computing situations because every terminal able to display multimedia content can be used as terminal control. And finally it enables a very low coupling between the middleware and the HCI, since all software developments are device independent.

The system we developed is able to identify the device present in the video stream, using SURF descriptors. It can also calculate the position and orientation of the instrument in the video stream in order to deliver a stable cropped video to the end user.

The future works will consist firstly in improving the authoring tool responsible for editing the interactive multimedia with a graphical user interface. Secondly, our efforts will consist in coupling the presented architecture with the OCELOT framework, Open Collaborative Environment for the Leverage Of insTrumentation [29]. We will then focus in bringing collaborative functions and group awareness in our architecture and finally run user tests within a practical work in our university.

6. REFERENCES

[9] C. Concolato and J.-C. Dufourd, “Comparison of MPEG-4 BIFS and some other multimedia description languages”, Workshop and Exhibition on MPEG-4, WEPM 2002, San Jose, California, USA


