Web-based Augmented Reality applied to upper limb simulation

Lucas Abrahão¹, Jean-Baptiste Gagnepain¹,², Kenedy Nogueira¹, Edgard Lamounier¹, Alexandre Cardoso¹

¹Universidade Federal de Uberlândia
²INSA Rouen, France

Abstract

Research in Virtual and Augmented Reality applied to biomedicine has been deeply investigated in recent years. The application to upper limb prosthesis is a wide field. Traditionally virtual upper limb prosthesis is a desktop application, used for training in rehabilitation centers.

Thus, this article reports on the use of Augmented Reality (AR) in a web-application. The objective here is to create a home based training environment.

1 Introduction

The purpose of this research project is to support AR simulation of upper limb prosthesis, through the internet.

Virtual Reality is a field of computer science that works with 3D modelling in an environment build by computers. Augmented Reality has proved to be an efficient tool for Health and Medicine desktop applications [7, 4]. For this project, the real environment is provided by a webcam, and the objective is to allow the user to interact with a virtual arm prosthesis by the intermediate of different signals.

From a biomedical point of view, a virtual prosthesis has advantages and disadvantages compared to real prosthesis. But the goals are different: the virtual prosthesis does not replace an arm in a daily way. It enables the patient to train with a prosthesis, and it is important for his rehabilitation. Thus, virtual prosthesis has a first advantage: the small amount of time needed to manufacture it. Indeed, a real prosthesis must be customized for the patient, according to his age, weight, size, and other important factors. It is not possible therefore to use this type of prosthesis during a time interval in which the Augmented Reality prosthesis is available. A second advantage exists for patients who were victims of an accident. During the recovery period it is impossible to apply weight on the injury, or expose it to friction. That is why an Augmented Reality prosthesis is useful, because contact with the patient is made only through the markers and electrodes.

On the other hand, using an Augmented Reality arm prosthesis is not easy. First of all, you need access to the application. This is why one of the objectives of this research is to create a web application usable on-line, and thus increase the application’s portability and mobility. Second of all, it is the difficulty involved in the use of such applications, and it is often required to spend an adaptation period before using correctly the application, and this might repel the patient. But this difficulty is mostly caused by the signal processing tools that were used to control the prosthesis.

2 Related work

Most of the previous research that used Virtual and Augmented Reality focused on the control with electromyographic signals (EMG) of upper-limb prostheses, in a desktop architecture [7, 4, 1]. The technical university of Cluj-Napoca presented a paper [1] about classification technique of surface electromyographic signals, based on neural networks. The system presented in [4] works with a hand prosthesis controlled by EMG signals, and presents a training and practising process in order to use the prosthesis more easily. The article presented at [7] focused on a mechanism intended to help the patient during the training period, since it can be long and tiring. It results in a very easy-to-use system that can greatly reduce the period of the training stages. The Lund Institute of Technology, presented this article [6] about real time control of a virtual hand. It shows tests executed on healthy subjects with a virtual hand, in order to improve the accuracy and to minimize the delay on a set of ten movements.

The work in [2] introduces Augmented Reality for upper limb prosthesis simulation. To allow a better immersion of the patient, the author made an application where EMG signals were captured, classified in real time processing, and then Augmented Reality helps the patient to visualize the prosthesis.
3 The proposed system

3.1 Methodology

The application was based on the work of Kenedy Nogueira’s master thesis [3]. But it needed to work on internet in order to increase mobility of the application, so a greater number of people can use it. The Flartoolkit technology allows us to implement Augmented Reality in a Flash application with Actionscript 3 and XML languages.

With the help of Flex Builder 3’s profiler, a canvas object was added and linked to the webcam. Then the animation of the arm was implemented. Four movements: pronation, supination, flexion and extension was designed in Augmented Reality. Usually, 3D objects are created in VRML (Virtual Reality Modelling Language). This is proposed since the objects files are not heavy, and it is very useful on internet. Figure 1 shows the VRML version of the Virtual Reality arms. However Flartookit only accepts the Collada format which is an open standard digital asset schema for interactive 3D applications.

In order to allow the user to interact with the system, the CheckBox tool was used. The CheckBox tool is adapted, since this component allows you to click on one of the buttons, and the appropriate event will start. One can distinguish four buttons of CheckBox type on this application: one for each of the movements. At this point, the user was able to choose which movement he wanted the Augmented Reality environment to show him, and thus switch the graphical component.

To simulate the arm movement an algorithm has been developed, which use the EMG signals from a file and send them to the AR web environment.

3.2 The classification algorithm

An Artificial Neural Network (ANN) is a mathematical model, whose representation is an oriented graph. The nodes represents the artificial neurons, and the edges represents the connexions between those neurons.

The ANN treatment is an adaptation of the master thesis [3]. For the limits cases: the 25 groups of characteristics are randomly send to the ANN. After the process is finished, the same signals were sent to the network so they can be recognized. We can see that the ANN was able to learn all the signals of all the groups of motions, trained with the variation of each limits, and the analysis of models of contraction (dynamic or static).

After the computation of the coefficients, the ANN receive and classify the motions.

An example of architecture of ANN with Multiple Layer Perceptron (MLP) is shown on Figure 2. Notice that in this case, the network has a layer of neurones in input who receive the coefficients AR, an internal layer, and an output layer who classify the motions.

Here is the algorithm used to sum up the data collected from the signal into four points, and then choose a motion characteristic of those points.

1- Initialize all the coefficients of the filter \( a_m(n) \) to 0.

2- Repeat all the points 3 to 5 for each \( N \) of \( y(n), \ n = 0, \ldots, N - 1 \).

3- Compute the estimated value of the signal \( y(n) \):

\[
\hat{y}(n) = \sum_{m=1}^{M} a_m(n)y(n - m)
\]

4- Compute the estimation error \( e(n) \):

\[
e(n) = y(n) - \hat{y}(n)
\]

5- Actualize the coefficients to the AR model:

\[
a_m(n + 1) = a_m(n) - 2\mu e(n)y(n - m)
\]
Where $\mu$ with $0 < \mu < 1$, is a convergence constant of the filter.

Figure 3 shows how two different motions: flexion and extension has signals totally different, which allows us to identify a motion with the help of the associated EMG signal captured.

### 3.3 Flartoolkit

Flartoolkit is an open source library for Augmented Reality in Flash. With the help of a marker as the one shown in Figure 4, and a webcam, the virtual object can appear in the environment. A particularly interesting point is that the object adapts itself to the inclination plan, and the distance of the marker. This is the key of Augmented Reality, since it makes us believe that the object behave as a real object. For example if you bring the marker closer from the camera, the object will enlarge, the same way an object enlarge when it is closer from someone. This library was based on Artoolkit, and implemented by a Japanese team[5]. It is now a reference for the use of Augmented Reality in a flash application.

### 3.4 The architecture of Flex and PHP applications

Flex applications can be characterised as a SOA architecture model, where Flex is used to create the client and to be connected to data using services.

To understand this architecture, consider how browsers and web applications are delivered. When the browser makes a request, the server uses a combination of static content (HTML/CSS/JS code) and scripts (these scripts may query a database or call other scripts, but in the end they output HTML/CSS/JS) to prepare a page. This page gets loaded and rendered by the browser. A key element here is that, usually this page (or response) has the presentation markup and the data into the same message.

When a new state of the application is to be presented, the browser makes a new request and the server prepares the page. The client “just” renders the data.

Flex applications works differently. The server sends the compiled Flex application (the SWF file) that runs inside
the browser using the Flash Player plug-in. Usually, this SWF file holds only the client-side business logic. If data are needed (from a database for example) the Flex application makes a request for those data. The server sends only the data (this can be in XML, JSON, AMF3 format), and the client knows how to represent this data visually. What happens here is a service-oriented architecture: the Flex application is the client that can consume data services from the server. The application can change state without refreshing the page or reloading the SWF file in the browser. The application is a client that can do more than “just” render data. Thus using Flex and Flash Player it is possible to create almost anything that makes sense to deploy on the web, from games to applications, to widgets that are integrated within “classic” web applications, and more (see Figure 5).

Rendered in a nutshell, Flex is a graphical interface that will interact with the user and the PHP language will do the task of communicating with the server.

This communication is required because data is sent from a personal computer (user) to a remote server. When this data is saved, the program will access this data and compile it, thus providing the results to the user.

When one sends a file to the server, it is not important to define what the user is sending or if he has permission to access certain directory. It can be defined within the php code which is the folder and name in the data to be saved, without any direct contact with the user. This code is not shown when the application is running and yes, working invisibly, based on user interaction. Thus, there is an increase in application security and prevents of future complications.

3.5 Case study

Figure 6. Screen shot of the application

Figure 6 shows a screen shot of the application. It has, on the left side, two buttons which allow to download the signal associated to a specific motion. The signal is loaded in the text area beneath the buttons. It then pass in the algorithm seen in 3.2, so the motion can be selected. On the right part of the interface the virtual environment is shown. It is composed by the scene, the webcam can record and the virtual object on the marker. Figure 7 shows the system architecture. First, the electrode gets the signal, that the electromyograph will analyse and send to the computer. Then, the host runs the application on a server, on the internet, which search in the database for the correct information. It is then send back to the Flash application on the host’s computer, and the interaction with the user can be visualize with the screen.

4 Conclusions and future work

Although, to move from a desktop architecture to a web one is not an easy task, the system proposed here has demonstrated that such migration applied to virtual upper limb simulation is possible.

Most of the authors future work will concentrate in using more computer techniques and algorithm to improve control of the virtual prosthesis and to simulate immersion AR.
References


