Virtual Reality system on embedded platforms

Tarek FRIKHA, Nader BEN AMOR, Kais LOUKIL, Mohamed ABID
CES-Laboratory
Sfax University, National Engineering School of Sfax
Sfax TUNISIA
tarek.frikha@gmail.com

***Abstract*: Virtual reality applications grow up nowadays. These applications are not only developed for the PCs but also for embedded systems such as game console, Smartphone, touchpad… The limited touchpad resources and network communication between touchpad need an adaptation to environment noises. To solve this type of problem, we'll test a 3D application adaptation on FPGA platforms. These platforms will be simulated to touchpad. This application is assimilated to a virtual reality application. In this paper we describe the OS implementation results and the adaptation of the 3D application data to the network transfer [1].**

I. INTRODUCTION

The virtual reality (VR) is a technical and scientific field using computer sciences and interaction devices to simulate the 3D features behavior in a virtual environment. The real-interaction between the user pseudo immersion and the environment is done through sensor motor engines. [3] This very important field permits to simulate real applications to obtain results that are similar to reality [2].

VR concerns many applications domains. We use VR not only in industry such as Simulation training (aerospace, medicine, vehicules driving), medical applications (phobias treatement, surgery simulation), nuclear application (dismantling in contaminated environments), video games, tele-immersion, scientific visualization, meteorology, astrophysics, basic research, architectural-urban planning, domotics, cultural heritage conversation, but also in artistic domains like virtual museum and sites visits, damaged objects reconstructions, works sound creating, virtual sculptures. [4]

The virtual reality applications are very greedy in resources needs. We need very performing PCs to have a good VR application. The aim of having moving virtual reality applications with autonomy make the VR embedded application conception a real hard challenge. This challenge is harder if we use two embedded systems which are linked by a server.

In this paper, we present a partial design of a prototype of an embedded VR system based on two embedded transceivers connected using a peer to peer link. The two transceivers are implemented in a two Xilinx ML 507 FPGA platform

The first transceiver permit to compute the virtual scene. It send the scene computation data to the second transceiver that must reconstitute in real time the virtual scene at the.

The design of such system has many challenges: design of an optimum architecture for such complex application, and the second major constrain is the adaptation to the wireless link perturbation.

The paper is organized as follow. Section II gives our work major features and compares it with related works. Section III presents the embedded multimedia system. Section IV is reserved to present the application design and the adapted 3D application. In section V, we talk about the obtained results. Finally, section VI concludes the paper with a brief outlook on future works.

II. STATE OF ART

In this section, we detail firstly related work in the domain of VR system. Secondly, we detail adaptation in the wireless network

1. Virtual reality

Virtual reality, which is a PC simulation of a visual environment, is a revolution of the perceived world. This field which was reserved only for army application is more generic now. In the precedent decades, the US army used virtual reality to simulate war scenes. These applications are generated in different countries to simulate different war situations. The aim of this application is to minimize the exercises cost. The soldier use virtual guns and is exposed to different ambuscades. The different scenarios can be created and soldier can do it in one place without any need to change the test environment. Everything is done automatically. [5]

The virtual reality applications are executed and simulated with much performed systems that support the huge calculations complexity.

The VR world touches the defense and other fields. Now the VR can be used not only in professional field but also in personnel one

In industry, it was used in the conception plan of architectural project such France's Nice Metro project which was simulated before the beginning of the project [6]. Many other industrial examples can be cited such as automotive applications. [4]. The whole production chain is created virtually and the technicians are formed to understand how to do their job properly. They learn how to find different components and how to add or retire them. It's also used in reeducation and health domain. The simulation of the chirurgical operation helps doctors to the application of new medical theory. Now a dentist can with 3D glasses and special tools execute all the teeth operations with a retro effect and different complications similar to reel operations.

VR can be used in other applications such as the video games in which the user becomes an entire part of the game. He's projected in an immersion world and he's reactive with the game environment. Many wars, sports, driving and RPG games are used the VR to make their game real.[7]

2. Peer to peer transfer¨

A peer to peer computer network represents any network that doesn't have fixed clients and servers but a number of autonomous peer nodes functioning as clients and servers. Peer nodes may differ depending on network bandwidth and storage quality.

Peer to peer (P2P) applications have attracted attention not only in academia but also in industry. Sharing application such as Napster and now Bitorrent, Skype, ooVoo … have a great success. The use of peer to peer touches not only sharing data and applications but also in P2P network for multimedia streaming and conferencing. The P2P internet trafic represents 70% of the internet trafic. [8]

The need to have a network adaptation without having to use a network server is not frequent. The network adaptation touching the QoS. In [9], the network adaptation is used to have a good video quality in spite of network perturbations.

III. EMBEDDED MULTIMEDIA SYSTEM DESIGN

The application demonstrator is represented in the figure 1. It contains two FPGA which communicate by a P2P transmission. They transmit data with 3D objects.

The emitter is embedded and permanent with the person. It can be a cask, a captor or any other embedded system. It can be also an IPAD or a tablet.

The receiver has to construct the 3D scene shown by the user. This scene has to be similar to reality with the added objects. The user needs to feel that he belongs to this environment. The real time constraints are very important and make the system very complex not only in the architectural part but also in software application.

The figure 1 represents an application demonstrator example.

Figure 1: Application demonstrator

The complexity of the applications narcissists the use of an operating system.

Operating systems provide a link between hardware and applications and perform a variety of tasks: from the abstraction of certain features of the equipment to the management of file system, and control of several concurrent processes to ensure that they do not interfere each other.

The virtual reality application sent from the first embedded system to another one need, due to its complexity, to use an OS with a graphical interface.

Often, any complex application requires the use of an embedded operating system not only for common services it provides but also for the possibilities of testing and trace it offers. Thus, some operating systems guarantee low latency (for real-time applications), control of multiple applications running simultaneously (multitasking), allow simultaneous access by multiple users in the system (multi-user) and allow multiple instances of an application to run parallel (multithreading). Furthermore any embedded operating system combines processes designed in order to detain the three constraints below:

- Available system memory
- Available processor speed
- The need to limit the energy dissipation when running the system.

We'll embed on the FPGA an embedded Linux application and graphical library. Linux is a free OS and its versions are optimized and adapted with Xilinx platforms

a) Embedded Linux implementation:

It's an adaptation of the Linux kernel to an embedded system. Because of many economic and technical advantages, we are seeing strong growth in the adoption of Linux for embedded systems [9]. This trend has invaded almost all markets and all technologies. Linux has been adopted for embedded systems in the switched phone network (PSTN) worldwide, the global data networks, mobile phones, and equipment for providing services to these networks [10]. Linux has been successful in automotive applications, consumer products such as games and PDAs, printers, switches and routers business, and many other products. The rate of adoption of embedded Linux grows up, with no end in sight. Embedded Linux is continuing to overwhelm the market for many reasons:

- Linux become a mature and SE high performance, stable alternative to embedded operating systems and traditional owners.
- Linux supports a wide variety of applications and network protocols.
- Linux is scalable, can be adapted to the products of small appliances for the public to larger industrial systems.
- Linux can be deployed without the payments required by the embedded operating systems and traditional owners.
- Linux has attracted a significant number of active developers, which allows rapid support of new hardware architectures, platforms and devices.
- An increasing number of suppliers of hardware and software, including almost all software vendors now support Linux.

For these reasons, we find an accelerated adoption rate of Linux in many common household items, ranging from high definition television to cell phones and depending on the capabilities of the system, there are some or all core features:

- Fewer services available.
- Less memory required (<8 MB).
- Boot from a ROM (FLASH).
- No keyboard or mouse required.
- Special software needed to control system devices (LCD, flash disk, IR remote ...).

The Linux is ported on the emitter platform.

b) Graphic library integration:

The degree of user interaction varies greatly from one system to another one. Some systems, such as PDAs, provide opportunities for complex interaction. Others, such as network equipment, are configurable through Web interfaces with a limited degree of freedom. Other one, such as industrial control systems use only few LEDs and few buttons for user interaction. Some other systems have no man/machine interface. However, using a graphical interface has the advantage of being particularly accessible to users unfamiliar with the environment, and thus learning is easier and faster.

To make easier the emitter use, it's important to use a graphical interface that's help the user for configuration.

Many Linux libraries can be cited such as:

- X11 graphical interface: X Window system, it's the natural defaults UNIX and consequently Linux one.
- XFree86 package: provided in most standard Linux distributions. The heavy execution and the memory footprint of the XFree86 packages make it incompatible with the reduced architectures.
- GTK+/Embedded library: Software library permitting to have a graphical interface. The GRK+ compilation is more complex than the Qt one because of the necessity of library annexes.
- QT-Embedded library: written in C++ is a multi-platform one. Is compatible not only for Linux but also MacOS and Windows with the some functionality. This library is characterized by its fiability. [11]

IV. Application design and 3D application architecture:

To remedy to the real time need of data transmission [12] we'll have to minimize the 3D application time execution. The 3D application is a very important part of VR. In this part, we'll describe the architecture application design, the used application and the 3D adaptation.

A. Application design:

The used application consists on transmitting the compressed application through the data from the first platform to the second one using a peer to peer network.

The major problem is the complexity of the application that contains a video MJPEG compression and decompression modules. To these modules we'll add a 3D application to adopt. The 3D part is described in the B part of this chapter. The complexity of the application needs an OS to manage different tasks.

a. Application choice (MJPEG) :

The MJPEG algorithm consists on compressing images to the JPEG format then we'll make these images into a video format. The first step of our system consists on taking snapshots from a webcam. These obtained pictures will be compressed by a JPEG image algorithm to obtain JPEG compressed images. To ease the Xilinx ML 507 platform memories use and manage the number of memory access, we need an OS.

b. Graphical interface and OS choose :

The embedded linux used architecture is composed by 5 important parts [13]:

- ✓ Hardware component driver: manage all the FPGA platform drivers used in the application.
- ✓ Memory manager: save data on the embedded platform memory.
- ✓ File system manager: manage the system files.
- ✓ Task scheduler: manage different tasks and make the application easier to use. In this case the task scheduler organizes the different steps to obtain of the MJPEG final video.
- ✓ Interprocess communication synchronization: in case of use of multi processor application this OS part do the interprocess communication and synchronization.

The figure 2 describes the embedded OS architecture.

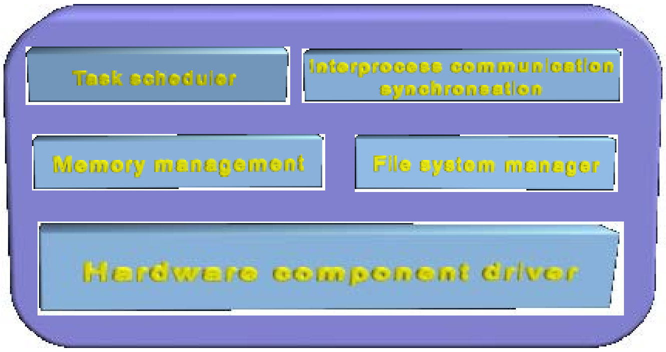

Figure 2: Embedded OS architecture

As described in III. b), we choose the QT embedded library. The QT library permits to have many needed widgets permitting to create graphical and communication interfaces between the user and to coder. QT is an embedded open source library. the QT advantage is that QT is writing images that are communicating with the accelerated graphic device via frame buffer. The figure 3 is describing the embedded QT architecture.

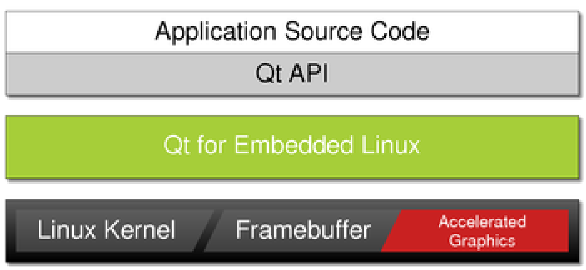

Figure 3: Embedded QT architecture

c. Transmission techniques

The environmental noises, the problems due to network transmission are influent for the good transfer of data. To obtain a good quality of service (QoS) and respond to the real time constraint of our system we have adapt our data not only the MJPEG algorithm but also the 3D application.

In the part B, we'll talk about the 3D chosen application.

B. *3D application:*

The 3D is now one of the most important image processing fields. The 3D applications which are the kernel of the VR is an important part in our application. [14]

The 3D application code has an asc file as input. This input which contains the polygons (triangles) coordinates and color data is attached to the two 3D object shaders. The 3D matrix result blocs output are sent to be displayed on the screen.

The 3D application study divides the function in two important parts:

- Geometric shader
- Vertex shader

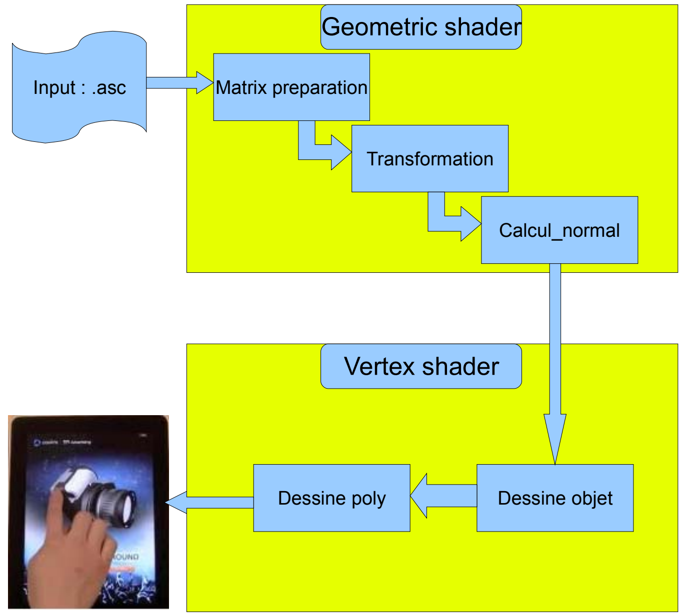

Figure 4: 3D application design

- Geometric shader

The geometric shader is the moving 3D part. It permits to the 3D object to make different rotations and translations. The geometric shader bloc is composed by 3 parts:

✓ Matrix preparation:

It's composed by many functions like matrix multiplications, perspective calculation, scaling, rotations and translations matrix, matrix multiplications and color palette calculations. This bloc is calculated one time and saves in a matrix to be used by the other ones.

✓ Transformation:

The transformation bloc, with normal one, are the most important bloc in the shader, they are called as much as the number of the polygons summits. The transformation bloc consists on multiplying the summit coordinates (abscise, ordinate, depth, color) by the value of the matrix preparation.

✓ Calcul_normal:

The last and not the least bloc of the geometric shader is the normal calculation one which determinates the surface normal vector coordinates. This is the last geometrical vertex calculation part.

The 2 last parts are the most calculating important part. The important number of call of each function makes their calculating time consumption important.

- Vertex shader:

The Vertex shader permits to compute each pixel color value to save it on on memory and to make it ready to the VGA controller. The figure 4 describes the Vertex shader functions used to prepare each pixel color value.

The vertex shader functions based on the open GL ES principle contains dessine_object function. This function includes dessin_poly function which permits to draw the each polygon. Each pixel color is obtained after that by calculating the barycentre function. This function is the base of Open GL ES 3D application implementation.

Due to the 3D application complexity we need to accelerate the application architecture. These architectural modifications are described in the next part.

V. Obtained results:

A. Os configuration and graphical interface:

On the ML 507 platform we can have a Power PC (PPC) or a microblaze (µB) soft core in which we can embed the OS. We choose the PPC because of his high frequency that make the execution faster and which is adapted to the OS embed.

After the OS installation, we create the graphic interface with QT to make easier the MJPEG video compressor.

In this paper we'll speak only about the MJPEG. The graphical obtained interface after the implementation is represented by the figure 5.

Figure 5: Graphical MJPEG QT interface

B. 3D application:

The 3D application adaptation, as cited previously, consists on how to transfer the object execution.

In fact, the 3D object data transfer depends on the bandwidth. If the network is ok, we can send the whole object packages. If there is a network problem and we can send the whole packages to the destination, we can subdivide the application into hardware and software blocks. The number of hardware blocks depends on the different network and real time constraints. The better is the network the smaller is the hardware execution. The more the QoS is needed the more is the accelerator number. This accelerator depends on the object type. If the object is much textured, the number of vertex shader increase, else if the object has many movement degrees the vertex shaders decrease and they are replaced by geometrical one.

To remedy to all this constraints, we can use a partial reconfigured architecture. One of the important Xilinx platform characteristics [15] is their ability to be dynamically reconfigured. We can allocate a zone to have a bitstream of the vertex shader if there is a textured 3D object and if there is the second 3D type, we download the geometrical shader bitsream. [16]

The real time constraints are very important and we have to respect it because in the VR the user has to feel that the added object is include in his environment.

VI. Conclusion / Perspectives:

We have presented our concept of flexible architecture for VR environement and presented the proposed hardware accelerators dedicated to pixel shading and geometrical one. We talk also about the partial reconfiguration which is more presented in the paper [17]. The future works will conclude these different approaches and will replace the MJPEG by the H264 algorithm. This second one is more complex than the MJPEG but presents a better video quality.

References

[1] P. Ngoc, G. Lafruit, J-Y. Mignolet , G. Deconinck, and R. Lauwereins "QOS aware HW/SW partitioning on run-time reconfigurable multimedia platforms" Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms, ERSA'04, June 21-24, 2004, Las Vegas, Nevada, USA. CSREA Press 2004, ISBN 1-932415-42-4

[2] Roudavski, S. (2010). Virtual Environments as Techno-Social Performances: Virtual West Cambridge Case-Study, in CAADRIA2010: New Frontiers, the 15th International Conference on Computer Aided Architectural Design Research in Asia, ed. by Bharat Dave, Andrew I-kang Li, Ning Gu and Hyoung-June Park, pp. 477-486

[3] S. R. Ellis. What are virtual environments ? In Proceedings of IEEE Computer Graphics et Applications, pages 17–22. IEEE, 1994

[4] B.Arnaldi, P.Bourdot, J.M.Burkhardt, "le traité de la réalité virtuelle: Volume 4, Les applications de la réalité virtuelle.

[5] M. Hachet. Interaction avec des environnements virtuels affichés aumoyen d'interfaces de visualisation collective. PhD thesis, Université Bordeaux I, 2003.

[6] http://www.enodo.fr/content/Home-page/home_pageFR.php

[7] P. Fuchs, B. Arnaldi, and J. Tisseau. La réalité virtuelle et ses applications,chapter 1, pages 3–52. Les Presses de l'Ecole des Mines de Paris, 2003.

[8] Karl Aberer, Zoran Despotovic. "Managing Trust in a Peer-2-Peer Information System. Technical report. Swiss Federal Institute of Technologt (EDFL). Lausanne 2001

[9] SASS, Ron et Andrew G. SCHMIDT. "Embedded Systems Design with Platform FPGAs Principles and Practices ", Morgan Kaufmann; 1ère edition (2010)

[10] GIFFORD Jim et Ryan OLIVER, "Cross-Compiled Linux From Scratch ", 1ere édition, (2009)

[11] MARQUET, Kevin, Thèse « Gestion de mémoire à objets pour systèmes embarqués » , Lille, (2007)

[12] Qt Reference Documentation, http://doc.qt.nokia.com

[13] M.Ben Said, N. Ben Amor, M. Abid, F. Ben Taher, J.P.Diguet "A bi-constraints adaptation technique for embedded multimedia systems", ICMCS 2010

[14] R.Ben Yemna e d of study memory in computer sciences engineering, ENIS 2011

[15] Kais Loukil, "Approche de gestion de performances/contraintes pour les systèmes embarqués temps réel"

[16] www.xilinx.com

[17] T.FRIKHA, N.Ben Amar, I. Ben Hlima, K.Loukil, M.Abid, J.P.Diguet « Self adaptative augmented reality systems on FPGA », ICMCS, tangier 2012