Eurographics Symposium on Virtual Environments (2006)
Roger Hubbold and Ming Lin (Editors)

Wearable Mixed Reality System In Less Than 1 Pound

Achille Peternier,[1] Frédéric Vexo[1] and Daniel Thalmann[1]

[1]Virtual Reality Laboratory (VRLab), École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland

Abstract

We have designed a wearable Mixed Reality (MR) framework which allows to real-time render game-like 3D scenes on see-through head-mounted displays (see through HMDs) and to localize the user position within a known internet wireless area. Our equipment weights less than 1 Pound (0.45 Kilos). The information visualized on the mobile device could be sent on-demand from a remote server and realtime rendered onboard. We present our PDA-based platform as a valid alternative to use in wearable MR contexts under less mobility and encumbering constraints: our approach eliminates the typical backpack with a laptop, a GPS antenna and a heavy HMD usually required in this cases. A discussion about our results and user experiences with our approach using a handheld for 3D rendering is presented as well.

1. Introduction

The goal of wearable Mixed Reality is to give more information to users by mixing it with the real world in the less invasive way. Users need to move freely and comfortably when wear such systems, in order to improve their experience without compromising their mobility. Nevertheless, mobile Mixed Reality is often limited by the lack of user comfort and mobility: wearing several pounds of shock-sensible electronic devices reduces the user's capability to interact with the surroundings and to fully benefit of the freedom this framework should provide. For example, the solution proposed by Reitmayr and Schmalstieg in [RS03] shows a typical scenario where such drawbacks can not be avoided.

Usually, wearable Mixed or Augmented Reality applications require high computational capabilities, directly translated into an encumbering and fragile set of devices. A typical complete wearable framework is composed by a backpack containing a compact computer with an internet wireless interface, a positioning system, an orientation tracker, a see-through head-mounted display and (sometimes) a camera. Such kind of platform has been largely used in projects like MARS [HFT*99] or ARQuake [TCD*02]. Those systems offer a full Augmented or Mixed Reality support but reduce considerably the user's mobility because of their weight and size: it is often almost impossible to run, jump, stride, crouch or even to sit when wearing them. Users need also few minutes to put on or remove the whole system. Additionally, a second person is required to help him/her installing the framework for the first time. Gleue and Daehne pointed the encumbering, even if limited, of their platform and the need of a skilled technician for the maintenance of their system [GD01]. Schnaedelbachin and al. in [SKF*02] even preferred to use a tripod-mounted Augmented Reality setup instead of a backpack. Similar platforms cost also thousands of dollars per unit and are difficult to shield against accidentally shocks or tumbles in action-based contexts.

In this paper we present our wearable, low encumbering, Mixed Reality framework. We will show how to build a lightweight, hands free, Mixed Reality platform featuring a semitransparent display allowing to superimpose over the real world 3D high-quality scenes rendered onboard, with localization capabilities, voice connectivity and dynamic information retrieval through a Wi-Fi connection. We will test our system in different contexts to illustrate the benefits of our light-weight framework over the typical platforms described above. We will conclude with a discussion about the advantages and drawbacks of our solution and the perspectives the next generation of mobile devices will offer in the domain of wearable Augmented and Mixed Reality.

2. Related works

In this section we explore the current state of the art of the technologies and solutions related to the topic of wearable Mixed Reality. Since the evolution in this domain is strongly hardware dependent, we split this part in two subsections: the first one as an overview of the current wearable and mobile Mixed or Augmented reality applications, the second one as a survey of the devices and software currently available to build a complete low-encumbering framework. Because of the extremely specific needs of our implementation, the state-of-the-art of this second part will be particularly detailed.

2.1. Wearable Augmented and Mixed Reality frameworks

A complete wearable Augmented Reality framework is used in the project MARS (Mobile Augmented Reality System), described by Höllerer and al. in [HFT*99]. This platform consists of a backpack equipped with a computer powered by a 3D graphic acceleration card (usually a notebook), a GPS system, a see-through head-worn display, a head-tracker and a wireless network interface. This system includes also a compact stylus-operated handheld device used as remote controller and interface to the laptop worn in the backpack. Some batteries need also to be packed within the system. Although the completeness of their solution, the size and weight of the whole system are noteworthy.

A similar system has also been used for ARQuake by Thomas and al. in [TCD*02], an Augmented Reality first-person shooting game. They also used a Differential GPS interface to increase the localization accuracy combined with fiducial markers to transform an area of their campus in an interactive game map. Although they obtained interesting results, the use of fiducial markers and DGPS antennas is a unpractical solution when extremely accurate positioning is not required: Differential GPS emitters are expensive and not widely available, image-based tracking requires to be formerly placed and configured, a camera support and works efficiently only under regularly enlighten environments. Finally, the computer-vision algorithms involved in image-based tracking systems can be computationally expensive for a mobile device. Wagner and Schmalstieg reported in [WS03] the difficulties and limitations imposed by image-based trackers when used with a PDA.

In [VKT*02], Vlahakis and al. used again a similar back-worn Augmented Reality platform to mix artificial ancient buildings with reality in an outdoor archaeological site. Nevertheless, they also implemented a handheld device alternative to their encumbering classic framework. They used a PDA with a position tracker (but not orientation) to check when the user entered within a perimeter around an interest point. Some information (like pre-rendered Augmented Reality images or videos) were then sent to the mobile device through a wireless internet connection. This approach was more like an electronic substitute to a paper-guide than a really alternate to a full AR wearable system. Furthermore, the PDA was just receiving pre-computed information over a Wi-Fi connection, without any kind of realtime 3D rendering done onboard, forcing the user to passively digest the content he/she received.

Figure 1: *Required equipment for a typical wearable Augmented Reality framework: a notebook, a stereographic see-through HMD, a GPS antenna, some battery packs.*

Some considerations about using a PDA as a mobile Augmented Reality device with high-quality 3D scenes have also been pointed out in [PWH*04]. Pasman and al. developed a client/server architecture using the PDA as an acquisition and visualization device, forwarding all the computational intensive tasks to a remote server over a wireless connection. Once the information elaborated, the final images were sent back the PDA. In their approach, the PDA was just used like a mobile terminal. In our solution, every 3D rendering or computational expensive task is performed directly on the PDA, which is the core of our platform.

A first utilization of PDAs as a truly, entirely on-board, interactive Augmented Reality context is showed in [WPLS05]. Wagner et al. created a game scenario using a wooden toy train model filled with fiducial markers and added virtual trains through Augmented Reality. They built their platform on the top of a camera-equipped PDA, mixing reality with synthesized images directly on the handheld embedded display. This way led to what they defined a “magic lens” effect: the user sees the world enriched with virtual images by moving the PDA in front of his/her eyes in a real environment, like through a magic glass. This approach reduces considerably the amount of weight to be worn by the user but forces him/her to keep the PDA in one hand and to continually move the handheld around to gather information from the mixed world. Moreover, the user sees the world indirectly through the PDA display, altering his/her perception of the surroundings: it is extremely uncomfortable to walk or run without looking around, thus loosing the Augmented Reality effect. Besides, they used an image-based system to track the camera orientation and position: this approach works finely only in a good and constantly enlighten space formerly supplied with fiducial markers, like the table they used for the train model, with 11 markers in a 5x5 feet (1.5x1.5 meters) surface.

2.2. Mobile Mixed Reality hardware

We have deepened our research to find the best hardware background allowing at the same time good 3D rendering capabilities, geo-localization features and Mixed Reality imagery in the smallest size and weight possible, hands-free, without sacrificing comfort and mobility freedom.

2.2.1. Core device

The core of a Mixed or Augmented Reality platform is a computing machine charged with the information retrieval and management, image synthesis, and acting as a crossroad between the different devices worn by the user (displays, trackers) and the environment (GPS, internet connections, etc). Since many years, the best candidate to perform this task has been a laptop PC. In fact, since few years, notebooks come with an excellent computational power, onboard Wi-Fi and Bluetooth interfaces and a true embedded 3D graphic accelerator which dramatically increases the rendering capabilities of such machines. The choice of a laptop PC has been so far the best compromise.

Unfortunately, in our context we have to deal with the weight and size restrictions the mobility and wearability constraints impose. Firstly, notebooks are conceived to be portable but not mobile: their hard-disks work badly when stressed by continued vibrations and shocks and may also be damaged. Secondly, they become rapidly hot (especially when 3D acceleration is used) and their fans need openings to efficiently compensate the heating. We tested this case with a low-end assembled laptop (2Ghz Intel Pentium Centrino with an ATI Radeon Mobility 9700 graphic card) and a high-end model (IBM Thinkpad T43P with a FireGL 3200 3D accelerator): after about 30 minutes of intensive 3D rendering, both computers showed temperature above the 104^oF (40^oC). This fact leads to some problem when the notebook is closed into a backpack or near the user's body. Finally, they are heavy and cumbersome: current generation average weight is above 4.5 pounds (2 kilos).

Nevertheless, some compact notebook model exists and features attracting specifications under reasonable encumbering constraints, but at the price of reduced 3D accelerated capabilities in regard to the full-sized laptop. For example, Dell's Latitude X1 (http://www.dell.com), weighting only 2.6 pounds (1.2 kilos) and measuring 11.4x7.9x1 inches (29x20x2.5 cm), or the tiny Oqo 01 (http://www.oqo.com) are interesting machines which should be used at least as a maintenance update to typical backpack-based wearable frameworks. Despite of their reduced size and weight, they still suffer the same heating and fragility problems the full-sized notebooks showed. PDAs seem to solve some of those problems.

Personal Digital Assistants (PDA or palmtops), originally conceived as evolution of handheld agendas, have greatly grown in versatility over the years and they are more and more getting similar to a compact PC. In fact, the current generation features a powerful RISC microprocessor (today with a clock rate between 400 and 700MHz) with a Wi-Fi adapter, 640x480 VGA embedded display, audio card and a good set of expansion tools, like compact cameras, GPS and GSM adapters, micro hard-disks, USB/Bluetooth interfaces and external monitors. Mobile by definition, PDAs are an excellent compromise between the computational power offered by notebooks and a truly wearable and lightweight device: they are less sensible to shocks and do not need special cooling cares. Furthermore, they are cheaper than a laptop PC. Some of them are also equipped with a 2D and, recently, a 3D acceleration chip, like Dell's Axim x50v featuring PowerVR's MBX Lite graphic processor (http://www.powervr.com). Thanks to the heavy improvements on the software side, PDAs are today reasonably easy to program and use in new domains far from the simple schedule or address book they were originally conceived for.

Handheld gaming devices and mobile phones need also to be mentioned: modern consoles like Sony's PlayStation Portable or Nintendo's GameBoy DualScreen (http://www.nintendo.com) offer great 3D rendering and computational power in compact sizes and weight. Unfortunately, these platforms are closed systems extremely entertainment-oriented and difficulty usable out of their original context. Moreover, the game industry does not seem to be interested in deepen their hardware on the Augmented or Mixed Reality way, thus the lack of non expensive gaming devices natively suited for this purpose. A few exceptions exist, like Sony's Eyetoy, or have existed, like Nintendo's Virtual Boy in 1995, but are far from being a worldwide top-selling product or standard.

Mobile phone constructors are also implementing in their products PDA-like functionalities, on models like Motorola's A1000 (http://www.motorola.com) or Sony Ericsson's P910i (http://www.sonyericsson.com), as well as gaming features, like on Nokia's NGage (http://www.nokia.com). Their imagery and computational capabilities are also increasing thanks to a new generation of 3D acceleration chips conceived explicitly for mobile phones (like NVidia's GoForce (http://www.nvidia.com) and ATI's Imageon (http://www.ati.com)). The same considerations about PDAs and handheld gaming devices are applicable to this category as well. It may also worth to cite some futuristic technologies (http://www.cs.nps.navy.mil/people/faculty/capps/4473/projects/fiambolis) foreseeing mobile phones with embedded see-through retinal displays: this approach would eventually eliminate the need of an external head-mounted display to superimpose synthesized images with reality, further improving the mobile aspect of a wearable Mixed Reality platform. In every case, we preferred the PDA approach over gaming and phoning devices because of the slightly faster and more versatile hardware architecture and optional accessories.

2.2.2. Head-mounted displays

Both notebooks and PDAs do not own semitransparent displays to mix synthesized images with reality behind the screen: in order to achieve this effect, an external specific device needs to be connected. In a wearable Mixed Reality context, the best way to do that is using a see-through head-mounted display (see-through HMD). Two different categories of see-through head-mounted displays exist: video see-through and optical see-through.

Video see-through head-mounted displays have closed, non transparent monitors and need an external camera to bring reality into the screens. Papagiannakis and al. already experienced this approach in [PSP*04]: users felt very uncomfortable with those devices, their spatial perception was completely distorted and they preferred to not move around, thus exploring the scene standing at a fixed point by simply heading around.

Optical see-through head-mounted display dispose of semi-transparent screens which allow the user to see the real world behind the lenses, like if he/she was wearing standard glasses. The user still perceives the reality directly through his/her own eyes.

There also exist monocular displays which further improve user comfort and spatial perception by keeping one user's eye completely free, which are lighter and more wearable, particulary important when walking or running around. Interesting models are Micro Optical's SV-6 PC Viewer (http://www.microopticalcorp.com) or Icuiti's M-920CF (http://www.icuiti.com), explicitly conceived to work with a PDA. Unfortunately, these models do not belong to the optical see-through category. In fact, optical see-through head-mounted displays are best suited for our scenario because they don't need to capture external images to bring them inside the HMD and improve the user natural spatial perception. In our opinion, Shimadzu's Data Glass2/A (http://www.shimadzu.com) and Liteye's Liteye-500 (http://www.liteye.com) are actually two good candidates for wearable Mixed Reality platforms: both are lightweight, high resolution, monocular optical see-through displays supplied by a powered USB port. Another interesting optical see-through HMD is an Olympus prototype (http://www.olympus.com): this next-generation model weights only 27 grams and cans simply be mounted on a pair of glasses, ideal for long time sessions.

2.2.3. Geo-localization devices and techniques

In our context-based Mixed Reality application, keeping track of the user's position is a necessary task because we used this information as the variable specifying the context itself. Several solutions for mobile scenarios exist and work efficiently in different contexts, depending on the needs and constraints. The Global Positioning System (GPS) is a worldwide standard based on satellite signals and terrestrial receivers. GPS accuracy largely depends on both the quality of the signals received and the receiver. This accuracy can be further improved by using a Differential GPS (DGPS). Differential GPS adds terrestrial signal emitters to use with signals sent from satellites: by knowing their spatial coordinates, the accuracy can theoretically raise up to some inches (centimeters). Unfortunately, Differential GPS works only on DGPS covered area, when standard GPS works worldwide. Simcock and al. used both systems to create a location-based tourist guide on PDA in a real context: their considerations about GPS and DGPS results can be found in [SHT03]. More recently, Burgit and Chittaro also developed a similar GPS-based PDA tourist guide in [BC05], adding onboard rendered 3D VRML models as well. Contrarily to Simcock and al., they experienced some accuracy problem with a simple (non differential) GPS.

Due to the growing request, an impressive amount of GPS devices for notebooks and PDAs has been developed and is continuously updated with new products. Actually, it is straightforward to find a low-consumption, compact GPS unit weighting less than 100 grams. Although, GPS suffers one major drawback: it works only outdoor.

Alternative positioning techniques have also been created using non-localization specific devices, like GSM networks and Wi-Fi cards with access points: Howard and al. in [HSS03] and Cheng and al. in [CCLK05] confirmed the feasibility of this technique. Both technologies use the quality of signal received from different access points (APs) to estimate the position of devices within a known area. This approach works very well in a high-density Wi-Fi covered zone, either outdoor or indoor, but requires precise spatial coordinates of every available access points and some calibrations before being used. Nova and al. presented in [NGD05] their considerations about Wi-Fi location-awareness using our campus as test area.

The main advantage of using an already existing Wi-Fi infrastructure for this purpose is that specific GPS hardware is no longer required, reducing the weight and the price of a wearable localization-capable framework. Moreover, both technologies (GPS and Wi-Fi based positioning) can be used together to track the user when far from a Wi-Fied area or when inside of buildings, and thus out of the scope of GPS satellites.

After knowing the current availability and state of the art of those technologies, we decided to adopt a PDA-based core, a monocular optical see-through head-worn display and use Wi-Fi geo-localization as positioning system.

3. System architecture

In this section we present our wearable Mixed Reality framework: firstly, we show a conceptual overview of the entire system architecture; secondly, we deepen into more technical details about every aspect of our platform.

Figure 2: *Our lightweight platform: Toshiba e800 PDA, Liteye-500, headphones.*

3.1. Overview

Our platform is based on three elements: a PDA, a head-worn optical see-through display and a geo-localization Wi-Fi based system. This platform is worn by a mobile user and is capable to display on the semi-transparent monocular head-mounted monitor 3D onboard rendered scenes made by several thousands of triangles fully textured and dynamically enlighten. The user sees the reality enriched by the information visualized on the HMD. The onboard Wi-Fi adapter is used to gather information about internet access points and quality of signal: we compare this information with a previously created database containing the spatial coordinates of every access point in our campus. This comparison let us identify the user's position. This position is then used as the context determining the feedback our application will provide. This system is completely autonomous and does not require a remote server or assistance. If the PDA is not hold in a pocket, the user can rotate the displayed scene by using the PDA's pad with a finger.

We also developed an improved framework, based on a client-server architecture and using a voice-over-internet service. In this setup, the mobile client wears headphones and holds the PDA on the forearm. The information relative to the Wi-Fi localization is sent to the server, which keeps track of the user's displacements. Additionally, a human operator works on the server and is vocally connected with the client: this allows both the mobile user to retrieve some specific information on-demand and the operator to send customized feedback when needed.

3.2. Implementation details

We chose the PDA as the core device of our mobile platform. PDAs are now widely available and offer a low-cost and less encumbering alternative to laptop PCs, still keeping a reasonable computational power, autonomy and shock resistance. We used two different models in our system: a Toshiba e800 or a Dell Axim x50v, both running under Microsoft Windows PocketPC 2003.

The e800 runs on a 400MHz Intel XScale PXA263 processor and includes Secure Digital and CompactFlash slots, as well as integrated Wi-Fi or Bluetooth capabilities. It owns an ample memory of 128MB of RAM, 32MB of ROM and a separate 32MB of flash memory for data backup. An integrated VGA display supports 640x480 VGA resolution with 65.536 colors. Weight: 0.42 pounds (192 grams).

The Axim features a 624MHz Intel XScale PXA270 processor, Secure Digital and CompactFlash slots, integrated Bluetooth and Wi-Fi, 64MB of RAM, 128 MB of ROM, a 640x480 VGA 65.536 colors display. Weight: 0.39 pounds (175 grams). One of the most interesting advantages of the Axim over the e800 is that the x50v is the first PDA equipped with a 3D accelerated graphic board (Intel 2700G) with 16 MB of video memory.

Both PDAs required an optional VGA adapter in order to output a stream to an external monitor. The e800 adapter (Toshiba Presentation Pack for PocketPC e800/e805) allows the handheld to output video at 640x480 or 800x600 VGA, 60Hz. This adapter includes also a useful powered USB port. The Axim x50v required another adapter (Dell Presentation Bundle), allowing video at 640x480, 800x600 and 1024x768, 60Hz. Adapters are in both cases fixed at the bottom of the PDA, freeing the two expansion slots on the top for more optional devices (like a GPS receiver, a tiny hard-disk or a camera).

On the semi-transparent monitors side, we adopted two different head-mounted see-through displays: one, more recent and expensive but light and compact, to show the ideal approach, another, older and widely distributed, low-cost, to illustrate a common case.

We used Liteye's Liteye-500 as the best one. Liteye-500 is an optical monocular see-through switchable head-mounted display (switchable because of a little sliding panel converting him to a non-transparent HMD) capable of 800x600

SVGA resolution. It doesn't require external adapters nor boxes and drains power directly from a powered USB port. It comes with a light-weighting, robust, headset. We had to add a small external 5V battery to our system when this HMD was used with the Axim because of the x50v lack of a powered USB port. Finally, the Liteye-500 is waterproof.

The second one is a I-O Displays i-Glasses!. This model is a binocular see-through HMD, with up to 640x480 VGA pixels resolution, stereovision and includes headphones and a head-tracker. Unfortunately, this model requires an external control box and a battery-pack (to avoid the use of the AC transformer), is more encumbering, heavy and fragile than the Liteye. Nevertheless, even if this device is by far less suited for our purposes, it is extremely cheap and widespread, thus an ideal comparative model.

On the software side, we entirely developed our code in C/C++. The 3D graphic engine used is a port to PocketPC of our own renderer: MVisio 3D Engine (see [PTV06]).

Our software is a full featured 3D engine allowing to easily display and manage complex textured scenes with dynamic lighting, animate characters through a skeletal-animation system, handling the scene-graph, layering a 2D GUI, etc. MVisio on PC uses OpenGL (http://www.opengl.org) as API to the underlying graphic hardware. Unfortunately, OpenGL does not exist for PocketPC, so we had to switch to something else on the Windows CE platform. Actually, there are several complete 3D graphic suites available for this platform, like the Diesel-Engine used in [GVT04] by Gutierrez and al. to display virtual humans or Klimt, used by Wagner and al. in [WPLS05] for their Augmented Reality invisible train. We decided to use our own engine because of the large amount of models and tools we already developed for the PC platform, because we were capable to rapidly modify the parts of code we needed without starting to learn a third-part 3D library from the beginning, and because we wanted to develop a full portable engine, allowing seamless integration of both PC and PDA platforms with exactly the same interface. Moreover, we wanted to benefit from the 3D accelerated power offered by the x50v, which required the use of specific software libraries. We chose to use OpenGL ES (Embedded System) (http://www.khronos.org) because of its similarities with the standard OpenGL and the availability of different software and hardware accelerated implementations. In fact, OpenGL ES was also the only valid low-level 3D rendering API currently running on PocketPC: only recently, with the imminent release of Windows PocketPC 2005, Microsoft has announced a mobile version of Direct3D. Among the various OpenGL ES libraries, we chose a generic software implementation of Hybrid (http://www.hybrid.fi), formerly referred as Gerbera OpenGL ES, and an hardware accelerated x50v-specific implementation released by PowerVR. We have to precise that different versions of OpenGL ES exist, formerly 1.0, 1.1, 1.2 and 2.0: these versions work like the OpenGL ones, progressively adding features both directly through the API and by driver-specific extensions. Furthermore, every version can use a common profile, allowing floating and fixed point computations, or a common-lite profile, including fixed point maths only. MVisio for PC requires at least an OpenGL 1.1 version to run and uses floating point maths: PowerVR's hardware accelerated library exposes only an OpenGL ES 1.0 fixed point interface. OpenGL ES 1.0 specifications have been conceived to improve graphic performances on non 3D-accelerated PDAs, thus implementing only a small subset of the whole standard OpenGL interface. This forced us to develop an OpenGL ES 1.1 to OpenGL ES 1.0 wrapper, in order to supply the missing functions like the current matrix retrieval, largely used in our engine. We finally obtained two different versions of MVisio for PDA: one running in software mode, compatible with every PocketPC we tested, and a second one, hardware accelerated, working only on the Axim x50v. We want to underline that the source code of the applications running MVisio on PC can be compiled without modifications for PDA, thanks to a big effort made to keep the same interface. We are observing since few years a kind of convergence in matter of 3D features on electronic devices: our approach benefits from this trend. Our testing devices were capable to render the same scenes, with the same quality settings, independently from the implementation of OpenGL ES they were using. Of course, the Axim x50v overtook every other PDA in matter of 3D rendering speed, thanks to its MBX-lite graphic accelerator: game-like scenes with more than 10.000 triangles and several megabytes of textures can be displayed in 640x480 VGA resolution with texturing filters activated at 25-35 frames per second. The same scene rendered in software-mode runs ten times slower. We obtained an acceptably smooth speed in software-mode by deactivating texture filtering and using a 320x240 QVGA resolution: in this case, we raised up to 6-7 frames per second with the same scene.

We managed two different solutions to display images rendered on the PDA to an external monitor: on the Toshiba e800, we activated a mirroring software included with the Presentation Pack, on the Axim x50v, we developed a custom software blitter by using the Intel's 2700G SDK. The rendering speed results showed above refer to a rendering done using the PDA embedded display as output screen: sending images to an external device is more difficult and time-expensive. In fact, PDAs can not render directly to an external monitor, so images have to be stored in memory and then swapped to the VGA output. In the e800 case, we used a monitor mirroring utility shipped with the adapter. We used this software like a black-box, because we didn't find any kind of support or documentation both from the producer and over the web. With the Axim x50v, we created a customized swapping procedure using the 2700G source development kit. Unfortunately, the Axim x50v showed a

Figure 3: *Test scene rendered in realtime on the e800 and x50v embedded screen and on the head-worn display.*

frustrating drawback: enabling an external display disables onboard graphic hardware acceleration, so we had to use the software-mode OpenGL ES implementation in this case, too, thus degrading performances. Moreover, OpenGL ES specifications do not allow to directly access the draw-buffer: we had to call the slow glReadPixels function once per frame, in order to get a copy of the rendered image. glReadPixels under OpenGL ES allows only to retrieve images in 32 bit RGBA format, wasting unnecessary resources and requiring a second conversion to a 16 bit format working with the VGA external adapter. Finally, Liteye-500 accepted only 800x600-sized streams, requiring an additional bitmap scale. We extremely optimized the code we used for this purpose: nevertheless, the biggest bottleneck was the glReadPixels which is unavoidable. All these limitations halved our framerate when connected to the see-through head-mounted display.

We used the Network Driver Interface Specification (NDIS) [Spe] to access low-level Wi-Fi adapter information like access point availability and quality of signal. Values returned may vary considerably over different Wi-Fi adapters and models, so we applied a simple calibration procedure in order to normalize this information in a range between 0 (extremely low-signal) and 1 (excellent reception). We normalized these values by storing the maximal and minimal intensity ever returned by the driver and by using this information to normalize the data between 0 and 1. We noticed that differences between Wi-Fi adapters weren't only limited to the range of values returned but also to their linearity: some adapters were more linear in the decay of the signal, other more exponential. We did'nt care about this detail in our tests because extreme precision was not required, nevertheless, taking in the account some kind of linearization of the decay curve over different hardware would improve accuracy and localization consistency over heterogeneous devices. Thanks to the extremely high amount of access points spread all over our campus, we had an average number of simultaneously scanned APs between 6 and 8. To identify the user's position, we created first a 3D array of cells (a kind of cube-tree) to discretize and model the area covered by our system, then we approximatively putted the MAC number of the different access points into the boxes corresponding to their physical position in our campus. After that, we walked around and stored in a database a capture of the access point list seen from a specific position with the respective quality of signal: this way, we filled a part of the empty cells in the 3D array. The remaining empty entries were completed by linearly interpolating values from the neighboring cells. To identify the position of the user, we simply checked his/her current list of visible access points with their respective signal intensity to find the nearest entry in the database. Such system offered us a reasonably accuracy (up to 3-5 meters) that we used to determine the area within a user was at a specific moment.

In the improved framework, we used Skype (http://www.skype.com) to implement a voice over internet connection between the user and the server operator. We simply launched the application in the background. A future improvement would use their open API to implement the same features directly into our client software. Custom data communications between client and server were managed by our own library developed on the top of Berkeley sockets. This library manages connections, messages and file synchronization on remote machines over a TCP/IP protocol.

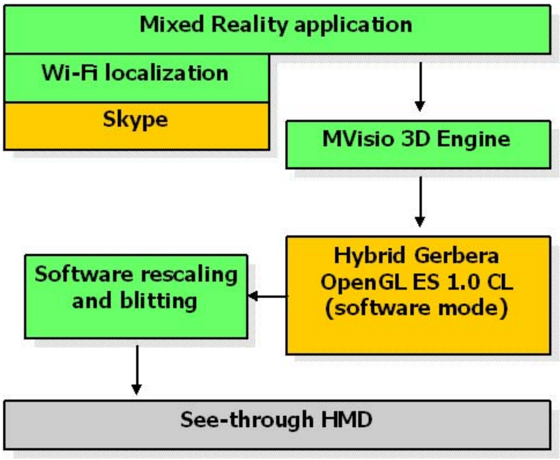

Figure 4: *Architecture overview: improved framework with software rendering. Dark boxes are third-part products.*

4. Tests and results

In this section we describe the tests we did and the results we gathered to check the efficiency of our platform. Our aim is to show how wearable, low-encumbering Mixed Reality is possible in new scenarios where formerly frameworks were useless. We tested the effective wearability and encumbering of our system *in vitro*, without creating a complete case of study but by examining every aspect separately. We asked to a group of colleagues to wear our platform and perform specific tasks, giving us their feedback and opinions.

4.1. Tests

We tested our platform in different configurations and contexts.

The first case was a simple comfort test. We asked the users to wear our framework first with the i-Glasses! then with the Liteye-500 head-mounted display. We didn't use geo-localization nor context-based feedback during this test: we putted simply a rotating human model on the display. We asked them to walk out of our building and come back.

We used a gaming-scenario as the second test: we simulated a treasure hunt by giving some feedback when a user entered within a specific access point range, like if he/she was looking for hidden treasure chests. This was just a pretext to force users to wear our framework for a long time and test its comfort when worn for more than 30 minutes.

In the third test we used the improved framework with headphones and the PDA held on the forearm. The mobile client was connected to a remote server and the user with a remote human operator. The user could ask for a 3D scene stored in the server database and receive it over internet directly on his/her PDA. By moving a finger on the PDA's touch-pad, the user could move and rotate the rendered scene displayed on the HMD.

In the last test we used our framework in contexts where encumbering constraints were critical: we asked them to run, jump and ride a bicycle while wearing our platform.

4.2. User opinions

Our platform worked efficiently during the tests. Here a list of comments reported by our volunteers:

- The i-Glasses! was completely rejected over the Liteye, more because of the reduced field of view showed behind the lenses than its weight. One user felt also embarrassed wearing the encumbering i-Glasses! in front of other persons.
- Even if the x50v was running slightly faster than the e800, most users liked the absence of the extern battery pack needed to power the Liteye.
- Users felt comfortable even after long time sessions. They compared our platform to a walkman.
- We were always displaying something on the HMD. During displacements between access points in the treasure hunt context, users preferred to have the choice to activate and deactivate the monocular screen.
- Everybody appreciated the voice interface with the remote server. They felt very natural to retrieve information by simply asking for it.
- Few users complained about the slow framerate, but only when they tried to rotate complex scenes using the touch-pad.
- Some slowdown occurred during information exchanging when mobile clients were far from the connected access point and the quality of signal low. Those limitations are due to the current implementation of the Wi-Fi protocol, which keeps the connected access point as long as possible, even if better APs are available.
- Few users refused to test our system when riding a bicycle. Moreover, they complained about the danger of wearing headphones while riding a bicycle. Nonetheless, users who tried that felt surprised by the low-impact our platform had on their actions.

4.3. Results and discussion

Our system achieved its goal: our framework showed to be usable between 2 and 4 hours, depending on the configuration and the distance from the active access point. The scenes were rendered with an average speed between 2 and 10 images per second, depending on the PDA used and the model loaded. Our platform proved to be extremely user-friendly and comfortable: we just needed a couple of minutes to teach to our users how to use it and nobody complained about its weight or encumbering.

On the other hand, some constraints need to be signaled. We faced several difficulties building our framework: the worse problem we encountered was the lack of documentation about how to access low-level PDA's accessories like the VGA adapter or the graphic chip. Often such kind of information is delivered by producers only to business partners: apparently, academic institutes do not seem to belong to this category. This lack forced us to recur to alternate and slower paths to find a solution.

Another problem was related to some minor incompatibilities between almost every element we used, incompatibilities which, once cumulated, caused a chain of slowdowns, like the time-expensive blitting procedure and the 3D acceleration disabled on external display. These problems could be avoided by simply offering a more open-architecture and documentation, which is *de facto* not the case on the PocketPC platform: in confirmation of this point, these drawbacks disappeared when we used the same architecture but without an external screen, by simply displaying the images on the PDA embedded monitor.

The Liteye showed also an hardware incompatibility with the VGA stream generated by the x50v: we needed to apply a second customized adapter to the one bought by Dell to modify the output into one correctly recognized by the HMD.

5. Conclusion and perspectives

We have tested if mobile and wearable Mixed Reality is today possible with less encumbering and lighter frameworks than in the past. We found that this is possible, but some bottlenecks reduce the performance of this approach. We pointed out that the actual limitations to our solution were due to avoidable slowdowns, either at hardware and software level, which should disappear with the next generations of PDA, allowing high quality 3D accelerated rendering directly on external monitors, dramatically improving performances and reducing the gap between handheld devices and notebooks. The technology is here: we just lack some connections. We are actually planning to change core and adopt a customized wearable platform, like the Texas Instruments' OMAP 2420, Linux based, entirely documented and opened, in order to avoid the limitations and constraints caused by under-documented components or avoidable software incompatibilities.

The need of a remote server operator, which can appear expensive in some contexts, could be avoided by using a voice-recognizer instead of a human: when used with a reduced number of words and a specific vocabulary, voice-recognizers can be extremely efficient.

In its current state, our solution could be directly used in other scenarios. For example, aircraft maintenance requires the technicians to have a rapid and efficient access to a huge information database containing the documentation related to every piece and annexe handling operations. Through our platform, this information could be accessed on-demand and uploaded from a central database to a mobile client. A 3D representation of the piece and a brief animation showing the manipulations could then be displayed on the HMD from different points of view, thanks to the onboard rendering capabilities of our platform.

With some expedient, our platform could also be adapted to work underwater, as a modern support for divers: with some customizations to a compact HMD (like the prototype presented by Olympus and cited before), images could be displayed on the mask glasses, showing the oxygen availability and the current depth, with blinking alert messages when values fall under security parameters. Many underwater accidents occur because divers forget to regularly check this information on they forearm-worn computers. In this case, unfortunately, the Wi-Fi based positioning system and on-demand information retrieval cannot be applied, because current wireless technology does not work at all underwater.

Acknowledgement

Many thanks to Sylvain Cardin for developing the VGA adapter to convert the Axim x50v output to a format correctly recognized by the Liteye-500.

References

[BC05] BURIGAT S., CHITTARO L.: Location-aware visualization of vrml models in gps-based mobile guides. In *Web3D '05: Proceedings of the tenth international conference on 3D Web technology* (New York, NY, USA, 2005), ACM Press, pp. 57–64.

[CCLK05] CHENG Y., CHAWATHE Y., LAMARCA A., KRUMM J.: Accuracy characterization for metropolitan-scale wi-fi localization. In *Third International Conference on Mobile Systems, Applications and Services (MobiSys 2005)* (2005).

[GD01] GLEUE T., DAEHNE P.: Design and implementation of a mobile device for outdoor augmented reality in the archeoguide project. In *VAST '01: Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage* (New York, NY, USA, 2001), ACM Press, pp. 161–168.

[GVT04] GUTIERREZ M., VEXO F., THALMANN D.: The mobile animator: Interactive character animation in collaborative virtual environments. In *IEEE Virtual Reality Conference* (2004), pp. 125–132.

[HFT*99] HÖLLERER T., FEINER S., TERAUCHI T., RASHID G., HALLAWAY D.: Exploring mars: Developing indoor and outdoor user interfaces to a mobile augmented reality system. In *Computers and Graphics, 23(6), Elsevier Publishers* (1999), pp. 779–785.

[HSS03] HOWARD A., SIDDIQI S., SUKHATME G. S.: An experimental study of localization using wireless ethernet. In *Proceedings of the 4th International Conference on Field and Service Robotics, Lake Yamanaka, Japan* (2003).

[NGD05] NOVA N., GIRARDIN F., DILLENBOURG P.: Location is not enough! an empirical study of location-awareness in mobile collaboration. In *IEEE International Workshop on Wireless and Mobile Technologies in Education, Tokushima, Japan* (2005).

[PSP*04] PAPAGIANNAKIS G., SCHERTENLEIB S., PONDER M., AREVALO M., MAGNENAT-THALMANN N., THALMANN D.: Real-time virtual humans in ar sites. In *In Proc. IEE Visual Media Production (CVMP)* (2004), pp. 273–276.

[PTV06] PETERNIER A., THALMANN D., VEXO F.: Mental vision: a computer graphics teaching platform. In *Edutainment 2006, Hangzhou, China, Lecture Notes in Computer Science (to appear)* (2006), Springer-Verlag.

[PWH*04] PASMAN W., WOODWARD C.,

HAKKARAINEN M., HONKAMAA P., HYVAKKA J.: Augmented reality with large 3d models on a pda: implementation, performance and use experiences. In *VRCAI '04: Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry* (New York, NY, USA, 2004), ACM Press, pp. 344–351.

[RS03] REITMAYR G., SCHMALSTIEG D.: Location based applications for mobile augmented reality. In *CRPITS '03: Proceedings of the Fourth Australian user interface conference on User interfaces 2003* (Darlinghurst, Australia, Australia, 2003), Australian Computer Society, Inc., pp. 65–73.

[SHT03] SIMCOCK T., HILLENBRAND S. P., THOMAS B. H.: Developing a location based tourist guide application. In *CRPITS '03: Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003* (Darlinghurst, Australia, Australia, 2003), Australian Computer Society, Inc., pp. 177–183.

[SKF*02] SCHNAEDELBACH H., KOLEVA B., FLINTHAM M., FRASER M., IZADI S., CHANDLER P., FOSTER M., BENFORD S., GREENHALGH C., RODDEN T.: The augurscope: a mixed reality interface for outdoors. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2002), ACM Press, pp. 9–16.

[Spe] SPECIFICATION N. D. I.: http://www.ndis.com.

[TCD*02] THOMAS B., CLOSE B., DONOGHUE J., SQUIRES J., BONDI P. D., PIEKARSKI W.: First person indoor/outdoor augmented reality application: Arquake. *Personal Ubiquitous Comput. 6*, 1 (2002), 75–86.

[VKT*02] VLAHAKIS V., KARIGIANNIS J., TSOTROS M., IOANNIDIS N., STRICKER D.: Personalized augmented reality touring of archaeological sites with wearable and mobile computers. In *Sixth International Symposium on Wearable Computers* (2002), pp. 15–22.

[WPLS05] WAGNER D., PINTARIC T., LEDERMANN F., SCHMALSTIEG D.: Towards massively multi-user augmented reality on handheld devices. In *Third International Conference on Pervasive Computing* (2005).

[WS03] WAGNER D., SCHMALSTIEG D.: Artoolkit on the pocketpc platform. In *IEEE Augmented Reality Toolkit Workshop* (2003), pp. 14–15.