Low-cost and home-made immersive systems

Sébastien Kuntz [1], Ján Cíger [2]

1 VRGeeks, Paris, France
2 VRGeeks, Paris, France

Abstract—**A lot of professionals or hobbyists at home would like to create their own immersive virtual reality systems for cheap and taking little space. We offer two examples of such "home-made" systems using the cheapest hardware possible while maintaining a good level of immersion: the first system is based on a projector (VRKit-Wall) and costs ≈ 1000€, while the second system is based on a head-mounted display (VRKit-HMD) and costs between 600€ and 1000€. We also propose a standardization of those systems in order to enable simple application sharing. Finally, we describe a method to calibrate the stereoscopy of a NVIDIA 3D Vision system.**

Index Terms—**Input/output, low cost VR, immersion, interaction.**

I. Introduction

With the new affordable interaction devices, 3D TVs, 3D projectors and head mounted displays (HMDs), it is now possible to create a truly immersive VR system of surprising quality for a very affordable price, especially when compared to the commercial systems.

We have created proofs-of-concept for the two most common types of VR systems. These are based on the cheapest commercially available parts that we could find, with the goal of getting a decent immersion and potentially a real sense of presence.

The first step to create such systems is to determine the minimum functionality they should provide in order to be immersive. As to our knowledge, there is no accepted norm defining a minimum immersion leading to a feeling of presence. We have decided that our systems should be driven by the abilities they should provide, and not by the specific hardware available on hand. By abilities we mean the set of actions of the user in the real world that will result in actions/perceptions in the virtual one. For example, the ability to move the hand or to rotate the hand will only be meaningful if the hand is correctly tracked and this information results in a perceptible reaction. This maximizes compatibility of applications using our "standard": if a system has similar abilities, it should be able to provide a similar experience.

Based on our experience we have settled on the following:

- For an immersive wall we need to track the head's *position*. The head's orientation can be omitted if we assume that the user keeps their head facing the screen.
- The *orientation* is essential for an HMD, but we also believe that we should track the head's *position* to obtain a more natural viewpoint control. Furthermore, the movement parallax provides an important cue for the perception of depth.
- Tracking of at least one hand (both position and orientation) seems important in order to be able to interact with the virtual environment in both systems.
- Having a few buttons and a joystick to simplify the interaction is interesting too (action selection, navigation in the environment, etc).
- For the immersive wall, a stereoscopic screen two meters wide is a good minimum because it provides field of view large enough to cover the peripheral vision.

With these criteria defined, we had to find the right components and assemble them to fulfill the objectives. It is assumed that everybody interested in this type of setup already owns a computer equipped with a decent 3D graphics card, so we don't count that into the total budget. Counting only the extra parts that are not usually on hand, we have arrived to a total budget in the range of 600–1000€ for the HMD-based system and 1000€ for the projector based one.

From there we can wonder why commercial systems can cost at least ten to twenty times more. In fact, their cost is justified by:

- Design, setup and maintenance fees,
- Ergonomics: reconfigurable and/or remote controlled,
- A better final quality meeting industrial requirements (colorimetry, calibration, precision, robustness, support, safety norms)

On the other hand, a low-cost system has several distinctive advantages:

- One builds their own immersive system and thus has a complete control over the budget and the system properties,
- The system can evolve based on the needs and available budget,
- It is possible to experience VR outside of a specialized laboratory, making it more accessible to a wider audience (hobbyists, educators, therapists, etc).

Good example of such need is physical rehabilitation, where low cost devices allow therapy approaches which wouldn't be accessible otherwise. Such examples are provided in [1], [4], [8].

Even if the laboratory, company or school already has one or multiple professional VR systems, building a secondary low cost one still has its own advantages—the system is always available and the maintenance/running costs are much lower, thus it can be used for tasks where the commercial system is prohibitively expensive (such as prolonged application development using many projectors). Low cost systems can also be useful for schools, for teaching students how to build a VR system, or simply to have equipment to teach VR application development.

The disadvantage of such low cost, do-it-yourself solution is that it often costs significant time and specialized skills (programming, electronics, computer vision experience, ...) to deploy and make work. Whether or not is this an acceptable trade-off needs to be carefully evaluated on a case by case basis.

The two low-cost systems presented here are "classical" VR setups that can be used for many applications. Moreover, there are many different combinations of hardware possible, leading to the same/similar result. We think that it would be interesting to offer some guidelines on how to keep some compatibility between all the potential systems and the software running on them, in order that people could share their applications easily.

II. RELATED WORK

Considering how much does an average commercial head mounted display or tracker cost, it is no surprise that attempts to design and build "low cost" solutions flourish in the VR community. However, when trying to compare our proposed solution with the body of existing work, two problems were discovered:

- There are wildly different ideas about what is a "low cost" solution. Considering that the cost of e.g. decent commercial HMDs starts at 20000€ (such as Kaiser Electro-Optics ProView series), even a $1700 HMD (eMagin Z800) is a "low cost" solution. However, that doesn't mean it is an affordable one.
- Many of the truly cheap, but still usable designs are created by enthusiasts and hobbyists. Unfortunately, they are not very likely to publish their work in scientific venues, so it may be hidden away at obscure web sites.

In order to narrow down the scope of the relevant works to ideas truly affordable even for hobbyists, the cost limit was set to be around 1000€. This limit is arbitrary, however it is approximately the price of a projector or a basic consumer HMD that many low cost users already posses. The second issue leads to the compromise that some works mentioned in this section are websites, personal blogs and similar resources of varied quality.

The list of works in this section is by no means comprehensive the works selected were considered as especially interesting or relevant from the point of view of someone looking at constructing a low cost VR system.

When speaking about low cost VR, one cannot avoid mentioning the work of Randy Pausch [9] from 1991. He describes how to build a very usable HMD-based system with a Nintendo PowerGlove for interaction and a Polhemus Isotrak magnetic tracker for about $5000. Considering when was this work done, it was a truly low cost system that was actually better than some commercially available offerings available at the time. A hugely popular device for low cost VR systems is the Nintendo Wii controller—the Wiimote. J. C. Lee's work [6] showed several possible ways of how this controller could be used and made the Wiimote popular as a very cheap tracking solution. The work of Schou and Gardner [12] describes how to use a Wiimote to work with multiple sensor bars, allowing its utilization in a CAVE-like surround setting. Schmeder [11] takes a different approach, replacing the Wiimote camera sensor with a magnetic compass and using the Gametrak device (see the description in the section 3) to build a fully 6DOF tracked (orientation + position) controller. Chow [2] describes a method how to use a Wiimote together with a head mounted display, working around the problem that the controller has to "see" the sensor bar in order to be able to determine its own yaw—a condition that is difficult to fulfill when wearing an HMD. Another related device is the Nintendo Balance board. De Haan et al. [3] describe how to use this peripheral as a cheap input device for a VR system.

Webcams are another popular choice for tracking, because they are cheap, widely available and allow wireless tracking. Kato's AR-Toolkit [5] and its derivatives have been used as low cost 6DOF trackers for VR. Another interesting project is FreeTrack [14], this software allows tracking of a constellation of several markers (LEDs) using a single webcam and estimates both the orientation and position, typically of the user's head, similarly to the Natural Point's TrackIR device. Finally, the Microsoft Kinect device can be also considered in this category (it is a specialized camera), with the FAAST project [13] by Suma et al. making the device readily accessible as a motion tracker without having to deal with the Kinect-specific APIs.

However, low cost VR projects do not focus on motion tracking only. OpenEyes, described by Li et al. [7], is a project building a low cost eye tracking device. Ryves [10] took on the challenge to produce an inexpensive shutter glasses controller that can be used without expensive graphic cards and with the very cheap, "dumb" glasses (such as the old Elsa 3D Revelator).

III. RESOURCES

In this section we aim to provide an overview of the available off-the-shelf hardware that can be re-purposed for the needs of our two VR kits.

III.1. Interaction devices

The biggest challenge is tracking the head and the hand(s)

of the user. The high-end systems can use optical or magnetic tracking with costs ranging from a few thousand euros (such as Natural Point Optitrack) to a few tenths of thousands euros or more (such as A.R.T, Polhemus systems).

There are several cheap alternatives, sometimes with a bit of development required:

- Webcam(s) with markers/props can provide 3D tracking (6DOF) and simulate buttons.
- Wiimote can provide absolute orientation and position changes over short time plus buttons (limited 6DOF, integration errors make it unusable for position tracking over larger distances)
- NaturalPoint TrackIR can provide absolute orientation and position (3-6DOF, depending on the tracked prop) of an object within a small workspace
- Razer Hydra is capable of tracking two joysticks (6DOF) and provides several buttons
- Playstation Move has inertial sensors that can provide absolute orientation, and the position can be tracked optically using an additional webcam (6DOF); unfortunately the device is difficult to use with a PC due to non-standard protocols and lacking driver support
- Microsoft Kinect can track the whole body facing the camera, providing joint positions of up to 4 users (3DOF + limited orientation estimation for some joints)
- Smartphones (such as iPhone) can provide absolute orientation tracking using their on-board sensors.

Fig. 1. Gametrak device, base station with the two sensors, on the left the tether is visible.

Choosing among these is a trade-off between robustness, workspace size, latency, simplicity of connecting them to the application and availability. A more complete table of the different devices that we have evaluated can be found on the VR Geeks website [15].

III.1.1) Position tracking

For tracking the position only, without orientation, the cheapest solution is also our preferred one: the Gametrak by In2Games, fig. 1.

This device was originally conceived and sold as an accessory for golf and boxing games for Playstation 2. Unfortunately, its production is discontinued, but it is still easily available from eBay for as low as 25€. It has a low latency, large workspace size (3 meters radius from the base station!) and a reasonable resolution (specified by manufacturer as ≈ 1mm, more like ≈ 1cm in reality due to mechanical plays in the gearing).

The Gametrak appears to the host machine as a pair of analog joysticks with 3 axes each and a single button. The joysticks provide two angles and a length of a tether's end (distance) from the base station, permitting calculation of a 3D position of each hand relative to the base using polar coordinates. We want to track one hand and the head position, so one tether will be attached to the HMD or baseball cap worn by the user.

III.1.2) Orientation tracking

When using the immersive wall we do not always need the orientation of the head. Most of the time, only the position is really required, and the stereoscopy will be correct as long as the user keeps his head horizontal and facing the screen, which is a reasonable simplification.

On the other hand, the orientation of the head is required when using an HMD. Also the hand orientation is required for better interaction in both systems.

Using a Wiimote, even with the Wii Motion Plus gyroscope accessory, isn't satisfactory for the head tracking without additional resources because of the limited yaw capability. The Wiimote has to either have the "sensor bar" in its narrow field of view or it has to rely solely on the gyroscopes in the Motion Plus. This is not practical for a head mounted display, because the user can be looking away from the "sensor bar" for extended periods of time without realizing it, leading to the accumulation of integration errors and inevitable drift. Possible solutions addressing this issue are described in [2], [12].

Fig. 2. PNI SpacePoint Fusion

We have taken another approach and have evaluated a different inertial sensor: the SpacePoint Fusion by PNI which was available[1] for about 100€, fig. 2. This device was originally intended as a technology demonstrator/reference design for the sensors developed by the PNI Corporation. It is very easy to use, reporting either the raw data from the accelerometer, gyroscope and magnetic compass or a fused orientation value as a quaternion. The on-board Kalman filter is very good, this sensor has proved to be very robust at tracking the orientation of a head or hand, with very little drift. There are two buttons as well that can be used for selecting objects on the screen.

III.1.3)   Position and orientation tracking

Fig. 3. Razer Hydra magnetic tracker

The newly released Razer Hydra (fig. 3) offers two 3D magnetic trackers with declared resolution of 1mm and 1∘ when closer than 0.5m from the base station.

This tracker works extremely well in good conditions. The

downside is that the tracking becomes unstable when too far (>50cm) from the base, limiting the available workspace. The device is also subject to the interference (distortion of the magnetic field) due to the presence of metals nearby, problem common with other magnetic trackers as well.

III.1.4)   Joysticks and buttons

We have used the joystick of the Razer Hydra, the Wiimote Nunchuk or a regular gamepad to provide us with the buttons and the two-axis joystick.

III.2.   Displays

A popular low-cost choice for image generation is the NVIDIA 3D Vision system. This system enables stereoscopy on a simple GeForce consumer graphics card, with some constraints (only full screen Direct3D applications are supported, limited control over stereo settings, only NVIDIA-certified devices are compatible), where it was only possible on the professional Quadro/FireGL graphic card series before. The 3D Vision setup consists of a pair of LCD shutter glasses, an USB emitter and a plugin for the regular NVIDIA driver. This driver automatically adapts the rendering of any Direct3D application into a stereoscopic one and synchronizes the glasses, the display and the application. The list of compatible display devices includes televisions with 3D support, 3D monitors and projectors.

III.2.1)   Projection

Back projection is the preferred option if there is enough space and budget available. It avoids the shadows of the user on the screen.

If this option is not available, front-projection is possible. In this situation a short throw projector is a very interesting option for two reasons:

- If the room is small one can still obtain a large picture,
- The closer to the wall the projector is, the less chance there is for the user to cast a shadow.

We have been using the Viewsonic PJD6381 projector which is compatible with the NVIDIA 3D Vision system. It has a 1024x768@120Hz resolution and can project a 2.2m wide image with a distance from the screen of only 1.5m.

As we are simply projecting on a white wall and not on a mounted screen, the system takes absolutely no space on the floor and fits nicely even in a very small room.

Another really interesting option that has become available recently are the new ultrashort throw projectors such as those manufactured by Sanyo (Sanyo PDG-DWL2500) or Canon (such as Canon LV 8235). Even though they are a bit more expensive, they offer better resolution and are easy to setup. Placed on the ground, adjacent to the wall, they can project an image of several meters across.

---

[1]It seems that at this time (May 2012) the SpacePoint is available only for large volume orders from PNI. A possible replacement could be the Razor AHRS project: https://dev.qu.tu-berlin.de/projects/sf-razor-9dof-ahrs

III.2.2) Head-mounted display

The Vuzix VR920 HMD offers a reasonable quality for a reasonable price: 400€. It has two 640x480 LCD displays providing about 22° horizontal FOV and a 3DOF more or less reliable orientation tracker. The new Vuzix VR1200 offers two 825x840 LCD displays, however the price is higher.

The Sony HMZ-T1 HMD offers the best quality/price ratio as of today: 720p stereoscopic OLED display with a 45° horizontal FOV, for about $800. However, it does not come with a built-in tracker.

IV.   VRKIT-WALL

Let us recapitulate the requirements for the VRKit-Wall system:

- Head tracking (position),
- Hand tracking (position+orientation),
- A two-axis joystick + 4 buttons,
- Large stereoscopic display

By combining the various devices examined in the section III we obtain the result shown in the fig. 4.

This installation uses the Gametrak device with one tether attached to the baseball cap worn by the user and the other is used to track the user's hand. The user interacts with the application using a Wiimote and Nunchuk, the stereoscopy is provided by the NVIDIA 3D Vision system.

Fig. 4. VRKit-Wall setup
Left: Ceilling-mounted projector, front projection
Right: User wearing a cap tracked using the Gametrak device and interacting using a Wiimote with Nunchuk

IV.1.1) Calibration

As mentioned before, the NVIDIA 3D Vision driver doesn't allow setting of the interocular distance, nor setting of the distance to the screen or the screen size, but it has some keyboard shortcuts to incrementally modify those parameters. Therefore an empirical calibration method has been created.

IV.1.2) Screen distance

To determine the correct distance from the user's eyes to the plane of projection, let us put the user's head at a known distance from the screen. Then we create a 3D environment where an object is exactly at the same distance from the virtual camera, and therefore should be exactly on the focal plane (with zero disparity).

By modifying the screen distance setting with the keyboard shortcuts (named "Convergence" in the NVIDIA 3D Vision configuration) until the two pictures overlap we can configure
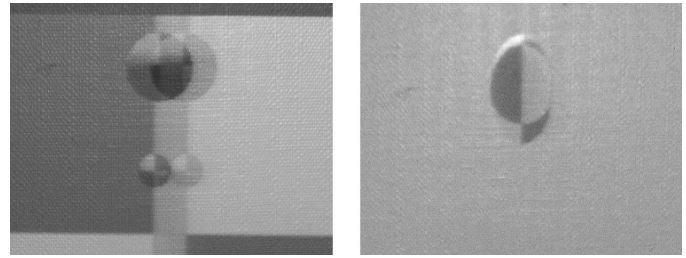
this parameter—see fig. 5.

Fig. 5. Incorrectly (left) and correctly (right) calibrated screen

IV.1.3) Interocular distance

In order to adjust the interocular distance we can place a virtual object behind the focal plane, at the same distance from the focal plane as the user (fig. 6).

By measuring the disparity value of the object (the distance between the same point on the right and left pictures), we can estimate the interocular distance of the 3D Vision system. Using similar triangles we can find that the interocular distance is twice the disparity of our object.
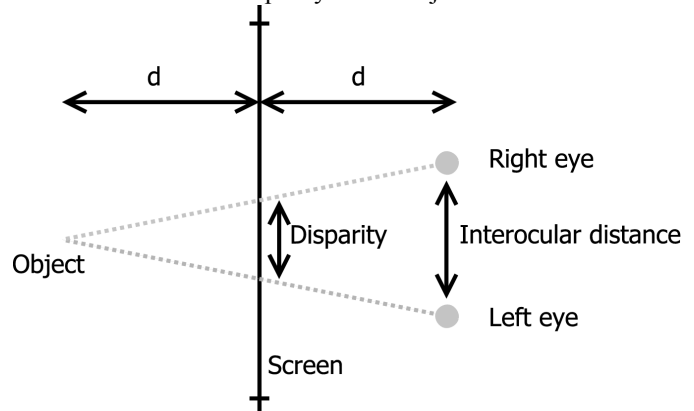

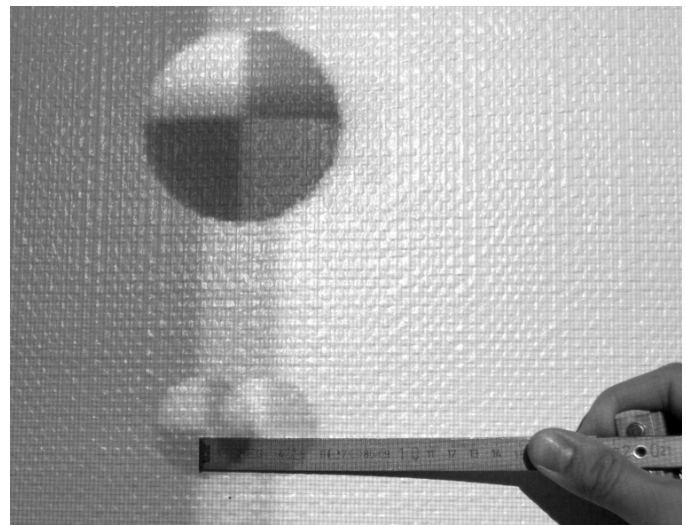Fig. 6. Calculation of the interocular distance

Fig. 7. Measuring the disparity and interocular distance

The interocular distance is modified by the "Depth amount" parameter of the 3D Vision driver. It has to be modified until the disparity measured is half of the required interocular distance (fig. 7).

IV.2. Discussion

We believe that we have managed to create a home-made immersive VR system. The users feel present in the applications, they can experience natural reaching for virtual objects, use natural head movements to explore the environment, exhibit reflexive reactions upon collisions with virtual objects, have a good sense of motion and even fear of heights.

The total budget of this system is approximately 1000€:

- Viewsonic projector: 800€
- NVIDIA 3D Vision kit: 150€
- Wiimote + Nunchuk: 50€
- Gametrak: 25€

The system as presented has several limitations. First, because of the lack of orientation tracking of the head and the limitations of the 3D Vision driver, the system cannot create a correct image if the user turns their head. As we do not know the orientation of the head we cannot know precisely where the user's eyes are looking. Moreover, the 3D Vision driver will only generate image assuming the user's head is horizontal and facing the screen. A proper solution would require tracking the orientation and using an oblique projection in the application (the optical axis not being perpendicular to the screen any more).

Another issue is that the screen distance value is static – this parameter is fixed by a keyboard shortcut and does not follow the user's distance from the screen as it should. Using the NVIDIA API would allow to fix that, however this is possible only for Direct3D applications developed specifically for the 3D Vision API or ones where the source code is available and can be modified.

Furthermore, due to the front projection being used, if the user comes too close to the screen, they will cast a shadow. This can be limited by using short throw projectors, mounting the projector on a ceiling or, ideally, changing to back projection.

Finally, there are too many cables.

V. VRKIT-HMD

Let us summarize the requirements of the VRKit-HMD system:

- Head tracking (position + orientation)
- Hand tracking (position + orientation)
- A two-axis joystick + 4 buttons)
- A head-mounted display (HMD)

The Vuzix VR920 is not 3D Vision compatible even though it's stereo capable, using either the frame sequential stereo or a Vuzix proprietary sync protocol over USB. We have decided that for the time being the motion parallax alone would be enough to obtain a good depth perception.

Fig. 8. VRKit-HMD with the Sony HMD

The Sony HMZ-T1 is not 3D Vision compatible neither, but it accepts a side-by-side stereoscopic picture and will display it in stereo, which is much easier to provide using a standard graphic card than a frame sequential signal.
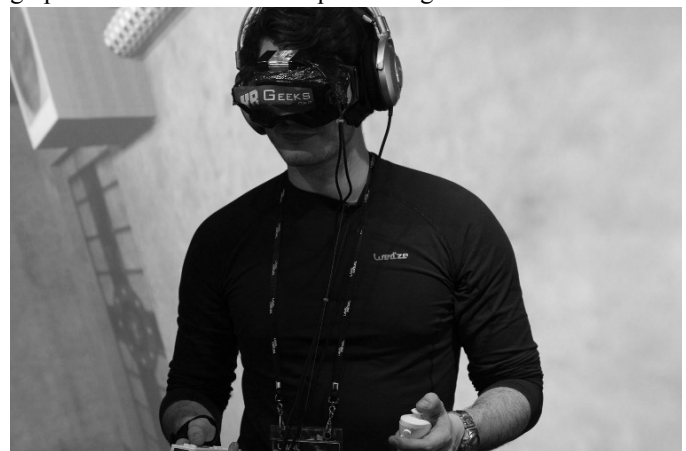

Fig. 9. VRKit-HMD with the Vuzix VR920, Gametrak and Wiimote for interaction

Figure 8 illustrates the setup of the VRKit-HMD. The HMD in use is the Vuzix VR920 one, tracked using the Gametrak device for position (using one of the tethers) and the SpacePoint Fusion (mounted on the top of the HMD) for orientation because the tracker inside the Vuzix HMD was unreliable. The user is using a Wiimote with Nunchuk for interaction and navigation.

Figure 9 shows the second variant of the HMD kit using the Sony HMZ-T1 display and Razer Hydra for tracking the hands. The head orientation is tracked using the SpacePoint Fusion device.

V.1. Discussion

We have created a transportable, immersive system; the entire setup (laptop included) fits into a 30 liter backpack. It

can be unpacked and ready to use in less than 10 minutes.

The total budget is approximately 600€ with the Vuzix VR920, or 1000€ with the Sony HMZ-T1, as can be seen in the table 1.

Table 1: VRKit-HMD budget

Variant A		Variant B	
Vuzix VR920	400 €	Sony HMZ-T1	800 €
SpacePoint Fusion	100 €	SpacePoint Fusion	100 €
Wiimote + Nunchuk	50 €	Razer Hydra	130 €
Gametrak	25 €	Gametrak	25 €
TOTAL:	577 €	**TOTAL:**	1 055 €

Because of the cables to the HMD, the inertial sensor and the Gametrak, the workspace is fairly limited but remains comfortable.

Care has to be taken to not tangle up the tethers from the Gametrak during use as that would impede the proper function of the device.

Originally, the Vuzix HMD is not fully immersive because it does not completely occlude the real world. There is a "Light-Shield" accessory available from Vuzix for this purpose, however we have opted for another approach, addressing the wearing comfort of the device at the same time. We have disassembled the HMD and placed it into a pair of ski goggles (fig. 8, 10) and covered it by tape, so that the user is unable to see anything else than the screens. This has improved the immersion dramatically.

Fig. 10: The modified Vuzix HMD, without the tape covering used to block the ambient light

Alternative techniques for position tracking need to be investigated, in particular optical tracking would remove the cables from the Gametrak and the inertial sensor. Unfortunately the widely available Kinect device is not really suitable for this task, because it has only very limited ability to track orientation. Custom software would need to be written to permit marker tracking.

Putting the laptop in a backpack would also allow to hide some wires and increase the mobility of the user.

VI. Standardization

The objective of our VR kits is that anyone can build their own VR system and use the VR applications developed by others. The amount of possible combinations of different hardware to achieve similar results is vast, so we have thought it would be interesting to offer some standardization so that those systems remain compatible in terms of hardware capabilities and software interfacing.

The hardware requirements were already outlined in the previous sections, what remains is the software part.

We have decided that applications should use VRPN, an industry-standard, free, open-source middleware that supports many common input and output devices used in the field. The VRKit-compatible application should support three different generic VRPN devices: a head tracker, a hand tracker, and a joystick, matching the hardware capabilities.

VRPN uses a client/server architecture, thus the client application should use devices with specific names matching the names in the VRPN server. This allows to setup a VRPN server only once for your particular system and run different compatible applications without reconfiguration. We are proposing the following names:

- The VRPN head tracker "HeadTracker@host",
- The VRPN hand tracker "HandTracker@host",
- The device offering two axes and four buttons should be named "Joystick@host"

Host denotes the network address/hostname of the computer where the VRPN server(s) run(s). If this convention is followed, different applications using these names will be able to run on any VRkit-compliant setup, regardless of what are the actual devices used.

The standard is currently only taking into account tracking devices, and not screen size/resolution/positioning. The user should be able to customize these settings to accommodate different screens/HMDs, either by modifying the source code or by the provided configuration options.

To guarantee a certain minimal level of immersion, this standard could impose stricter rules.

- We could impose a minimum tracking volume, a maximum acceptable latency and minimum update frequency and accuracy.
- We could also impose a minimum screen size and a minimum resolution, as well as require a complete real world occlusion.

At this time we have chosen not to do that, because demanding too strict standard could put the kits out of the reach of our target group, it is always a trade-off between the hardware capabilities and associated costs. This decision and the proposed specifications are, of course, open to debate and critique, authors welcome any suggestions.

VII. Applications

Thanks to the VRKits, we have been able to create several applications already. Several of them were developed during the Global Game Jam 2011 and 2012. We were able to bring the virtual reality to people who wouldn't otherwise have had access to it: game designers, 3D artists and sound designers,

with good success.

Some game examples that were created by the association members for the VRKits mainly during the Game Jam events:

- *InSnaketion*: a simple shooter in a futuristic environment (fig. 11),
- *Sadistic*, escape from a cube-shaped puzzle by manipulating it like a Rubik's cube while avoiding traps (fig. 12),
- *VR Escape*, escape a dark laboratory by interacting with the environment (fig. 13).

Fig. 11: InSnaketion, shooter game

InSnaketion is a simple shooter game. The goal was to use an existing and well-known game genre and adapt it to VR. Being immersed inside the game was already an interesting addition, but we really wanted to take advantage of the capabilities of our system.

Players who use an HMD for the first time have a strong tendency to look straight ahead, like they're in front of a TV, and to not move the head. We have forced them to look around by adding monsters that appear on the sides.

Moreover, typical PC shooters aim where you look at. In reality one can look and shoot/move in different directions. This is what we did in InSnaketion also, it is possible to look for example to the left while shooting enemies overhead.
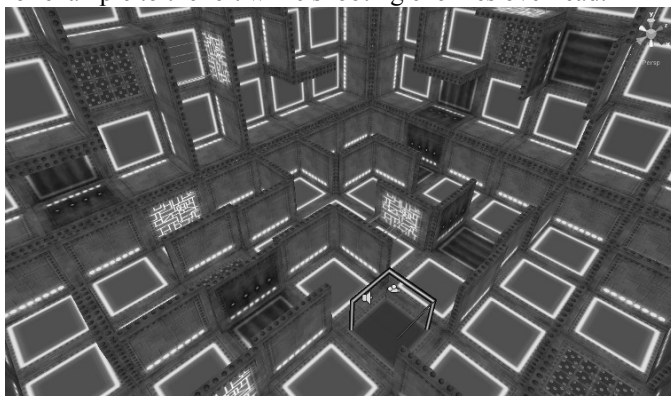
Fig. 12: Sadistic, puzzle game requiring environment manipulation

The game was created in 48 hours during the Global Game Jam 2012 in Paris by Judith Guez, Camille Gazeau, Jean-François Jégo, Guillaume Bertinet, Sylvestre Bauhain, Nicolas Conil and Sébastien Kuntz.

Sadistic is a maze full of traps that the user has to escape. The trick is that the maze is in three dimensions: the player is inside of a cube and each face of the cube is a maze. It is possible to move from one face to another by pressing a button that will rotate a ring of the cube, exactly like Rubik's cube.

The game takes advantage of the head tracking so the user can look around to orient themselves in the cube and to intuitively search for the exit. It was demonstrated at Laval Virtual 2011, and created by Tahir Vico, who invented the gameplay and designed the game, helped by Takara Orca, Thierry Desseaux, Eric Beets, Nicolas Conil and Sébastien Kuntz.

Fig. 13: VR Escape

VR Escape is our most audacious attempt at taking full advantage of VR. The user has to move their head, has to crouch and look around to understand how to escape. For example, the player has to take a magnetic key and insert it into a key reader so that the first door will open. Then they have to obtain access to a secret code to open the second door. They will need to open a drawer, take a CD from i, and insert it into a computer which will display the secret code.

The 3D graphics are really dark and scary; a lot of emphasis was put on the sound experience which, we believe, greatly increases the immersion.

The game was created in 48 hours during the Global Game Jam 2011 in Paris by Kevin Passageon, Thomas Chalan, Johan Lajili, Nicolas Bourdon, Romain Lecomte, Nicolas Conil and Sebastien Kuntz.

The gameplay videos are available from the VR Geeks website [16].

VIII. COMMUNITY

In 2011 we have established a community of people passionate about virtual reality, the "VR Geeks". The community started as a group of French virtual reality enthusiasts and has rapidly grown to the current state where there are many international members, including professionals,

academics, students and hobbyists interested in the practical aspects of VR.

The Parisian chapter meets regularly for informal discussions, organize workshops or visit VR centers. The association has also presence at the annual LAVAL Virtual trade show. The kits were presented at the AFRV (French Association of Virtual Reality) and Web3D conferences, as well as the Intel Geek's So In event.

Everyone is welcome to join the mailing lists and/or create local chapters. All relevant information about both the community and VRKits is available at the association's website: http://www.vrgeeks.org/.

IX. CONCLUSIONS

We have presented two low-cost home-made immersive VR systems:

- VRKit-Wall: a projector-based stereoscopic VR wall with a 2.2m wide image for 1000€
- VRKit-HMD: an HMD-based VR system starting at 600€

We have demonstrated that we can create a decent low-cost immersive VR systems using off-the-shelf hardware, making the technology accessible even to audiences where VR was not considered before.

We have also introduced a hardware and software standard based on minimum hardware capabilities and on VRPN, aiding the compatibility across different systems and enabling the application exchange.

Another contribution presented is the method to calibrate the low-cost NVidia 3D Vision stereoscopic system, working around the built-in driver deficiencies.

ACKNOWLEDGEMENT

REFERENCES

[1] S. Attygalle, M. Duff, T. Rikakis, and J. He. Low-cost, at-home assessment system with Wii remote based motion capture. In *Virtual Rehabilitation*, 2008, pages 168–174. IEEE, Aug. 2008.

[2] Y. Chow. The Wii Remote as an input device for 3D interaction in immersive head-mounted display virtual reality. In *Proc. IADIS International Conference Gaming 2008: Design for Engaging Experience and Social Interaction*, page 8592, 2008.

[3] G. de Haan, E. J. Griffith, and F. H. Post. Using the Wii Balance Board as a low-cost VR interaction device. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, VRST '08, page 289290, New York, NY, USA, 2008. ACM.

[4] J. E. Deutsch, M. Borbely, J. Filler, K. Huhn, and P. Guarrera-Bowlby. Use of a Low-Cost, commercially available gaming console (Wii) for rehabilitation of an adolescent with cerebral palsy. *Physical Therapy*, 88(10):1196–1207, Oct. 2008.

[5] H. Kato. ARToolKit: library for Vision-Based augmented reality. *IEIC Technical Report (Institute of Electronics, Information and Communication Engineers)*, 101(652(PRMU2001 222-232)):79–86, 2002.

[6] J. C. Lee. Hacking the Nintendo Wii Remote. *IEEE Pervasive Computing*, 7(3):39–45, Sept. 2008.

[7] D. Li, J. Babcock, and D. J. Parkhurst. openEyes: a low-cost headmounted eye-tracking solution. In *Proceedings of the 2006 symposium on Eye tracking research & applications*, ETRA '06, page 95100, New York, NY, USA, 2006. ACM.

[8] K. Morrow, C. Docan, G. Burdea, and A. Merians. Low-cost virtual rehabilitation of the hand for patients Post-Stroke. In *2006 International Workshop on Virtual Rehabilitation*, pages 6–10. IEEE, 2006.

[9] R. Pausch. Virtual reality on five dollars a day. *In Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, CHI '91, page 265270, New York, NY, USA, 1991. ACM.

[10] B. Ryves. LCD shutter glasses adaptor. http://benryves.com/products/lcdglasses, Feb. 2012.

[11] A. Schmeder. Hacking a Wii remote for full 3D absolute position and orientation sensing | CNMAT. http://cnmat.berkeley.edu/node/7649, May 2009.

[12] T. Schou and H. J. Gardner. A surround interface using the wii controller with multiple sensor bars. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, VRST '08, page 287288, New York, NY, USA, 2008. ACM.

[13] E. A. Suma, B. Lange, A. Rizzo, D. M. Krum, and M. Bolas. FAAST: the flexible action and articulated skeleton toolkit. In *2011 IEEE Virtual Reality Conference (VR)*, pages 247–248. IEEE, Mar. 2011.

[14] The Freetrack Team. FreeTrack optical head tracking software. http://www.free-track.net/english/, Feb. 2012.

[15] VR Geeks Association. Device classification. http://www.vrgeeks.org/wiki/device-classification, Jan. 2012.

[16] VR Geeks Association. VR experiences. http://www.vrgeeks.org/vrexperiences, Jan. 2012.

Sébastien Kuntz is a software engineer living in Paris, France. He started his career at the French railways by creating two highly interactive VR training simulators, which used 3D tracking of the user and a treadmill.

He then joined Virtools / Dassault Systèmes as VR lead engineer.

He is the founder and president of the VR software company "i'm in VR" located in Paris, France.

Mr. Kuntz is a member of the board of the French VR association (AFRV). He is the founder and president of the VRGeeks association. He was the co-organizer of the first 3D user interaction contest at 3DUI 2010.

He earned two Laval Virtual awards for his work at the French railways and for MiddleVR, the flagship product of "i'm in VR".

Ján Cíger received the magister degree in computer science from Comenius University, Bratislava, Slovak Republic in 1999; and the PhD in virtual reality from the École Polytechnique Fédérale de Lausanne, Switzerland in 2005.

He worked as a software engineer with West-Ost-Connection from 1999 to 2001, then joined the Virtual Reality Laboratory at EPFL, Lausanne as a research assistant from 2001 until 2005 to pursue his doctorate.

He was an assistant professor in the Medialogy department of the Aalborg University, Denmark from 2005 until January 2010. His is currently employed by the French startup Reviatech SAS as a researcher, his research interests include computer graphics, virtual reality, human-computer interaction and artificial intelligence.

Mr. Cíger is a member of the French chapters of the VRGeeks and A3 (EPFL alumni) associations.