TECHNICAL REPORT ON THE DEVELOPMENT OF VIDEO GLASSES FOR EYE PROTECTION

Eric Mathieu, Jean-Philippe Farrugia, Aurélien Grivollat

TECHNICAL REPORT ON THE DEVELOPMENT OF VIDEO GLASSES FOR EYE PROTECTION

Authors: Eric MATHIEU[1], Jean-Philippe FARRUGIA[2], Aurélien GRIVOLLAT[2]

1. ABSTRACT

There is a lack of eye protection for high energy lasers or those having too many wavelengths. Therefore, we have developed a video see-through Head Mounted Display (HMD) to have an indirect vision.
The use of video display allows us to add Augmented Reality and help the user in his task.
The HMD is composed of off-the-shelf elements. The displays are general market glasses (Sony, Oculus) whereas cameras are industrial highly configurable cameras (IDS - ueye).
This paper is a technical description of the device assembly, programming and testing. We highlight the current constraints such as time lag, picture resolution, field of view, weight and autonomy.

Keywords: HMD, laser safety, eye safety, Augmented Reality, image remap, video glasses, Personal Protective Equipment (PPE), video see-through

[1] CETHIL : Centre d’Energétique et de Thermique de Lyon, 9 rue de la physique - 69621 Villeurbanne cedex, France
[2] LIRIS : Laboratoire d'InfoRmatique en Image et Systèmes d'information, Bâtiment Nautibus, 8 boulevard Niels Bohr - 69622 Villeurbanne Cedex, France

2. INTRODUCTION

When doing an eye-hazardous activity such as soldering or using high energy lasers, workers currently have the possibility to wear filters glasses. Two types of filters exist: Intensity filters reduce the whole light spectrum to an acceptable level, and wavelength selective filters let part of the visible spectrum reach the eyes (and stop a narrow bandwidth laser for example).

Figure 1: Example of a laser protection mask using filter

The filters cannot protect from (i) too high energy lights (case of high energy lasers) and (ii) too many visible wavelengths at high energy (multiple lasers or white lasers). Consequently, they result in color blindness.

To avoid these inconvenient, we developed a new closed protective helmet with indirect viewing through stereo cameras. This has multiple advantages: (i) full protection as the light cannot reach the eyes, (ii) possibility to add additional information via augmented reality in the display to help the user, (iii) full color vision.
To preserve 3D perception, we used a stereo system with 2 cameras and 2 displays in front of each eye.

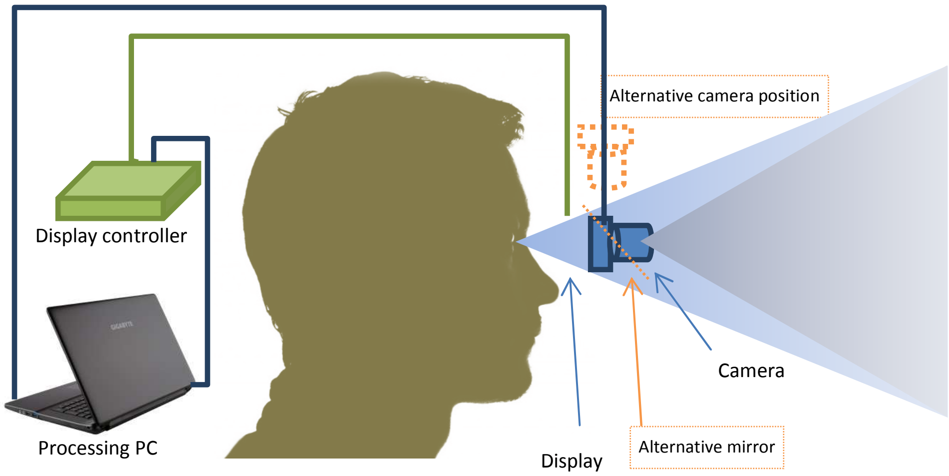


Figure 2: System Layout

Figure 3: Augmented Reality application example on a dye laser.
Visualization of invisible beam, delimitation of dangerous area, display of laser power during adjustments...

This project has been initiated in 2012 in the CETHIL lab which is using high energy lasers. Since 2013, the LIRIS lab is also involved to bring computer vision and computer graphics expertise into the project.
Until 2015, the project's goal was to demonstrate the relevance of the concept and its feasibility.

This technical report will be structured as followed. After this brief introduction, the hardware and software structure of the system will be presented through sections 3 and 4. Section 5 will present user tests in working conditions, with various configurations. Section 6 will conclude with future developments and perspectives.

3. HARDWARE CHOICE

In this section, we will present the hardware selection process. Knowing the constrains, we selected the best suited hardware available on the public market (military market was not an option).

CONSTRAINS

We identified the following constrains for the hardware selection:

- High refreshing rate for visual comfort (at least 30 fps for each eye). By experience, 50 fps would be even more comfortable. Advanced virtual reality systems such as the Oculus HMD recommend 75fps [1].
- Minimum lag between picture recording and display. Less than 20ms is recommended by the oculus HMD [2], although, it becomes sensible to the user (and give sickness) above 48ms [3].
- Highly sensitive cameras for dark environments.
- Good angle resolution. To meet human vision, we should display with a resolution of 1 arcminute (approximately 0.02°) [4].
- Large field of view, especially when moving, in order not to hit objects around the user. Ideal field of view would be 175° (as the human eyes [5]).
- Light weight. At least for the part that will be worn on the head.

HARDWARE CHOICE

For ease of development and maintenance, the software part of the project runs on a laptop with USB connected cameras and different off-the-shelf video displays. The code itself could run on a smartphone or tablet, but these devices actually lack a high speed USB connection for the cameras.

For the first prototype, we used a VUZIX Wrap 920 AR mask already equipped with 2 cameras (640*480px) and 2 displays (640*480px, horizontal FOV of 30°).

Figure 4: the Vuzix Wrap 920 AR glasses

Figure 5: the Vuzix camera and displays mounted on a protective mask

This prototype has demonstrated the potential of the technology but had the following strong limitations for our usage:

- Cameras with rolling shutter (webcams): not compatible with pulsed lasers
- Low resolution
- Narrow Field of View

Figure 6: laser saturation example with rolling shutter cameras

CAMERAS

Our next choice was driven by the need of high speed port (USB3) for high resolution cameras with high frame rate (>50fps). We chose the IDS UI-3251LE

Figure 7: IDS UI3251 camera

- CMOS sensor of 1/1.8" diagonal
- USB3 interface
- 1600*1200px
- Rolling or global shutter modes
- 60 fps max
- M12 optics mounting (for easy optics change)
- Dimensions H/W/L : 36,0 mm x 36,0 mm x 20,2 mm
- Weight of 12g
- Manual control of: exposure, frame rate, colors, etc.

VIDEO GLASSES

- Sony HMZ-T2
  - 2 OLED screens
  - 1280x720 pixels (angular resolution = 28.4 pixels/degree)
  - Horizontal FOV of 45°

Figure 8: Sony HMZ-T2 head display

A second head display has been later acquired in 2014 for larger field of view testings:
Oculus DK2

- OLED screen
- Resolution (per eye) : 960 x 1080 px (angular resolution = 9.6 pixels/degree)
- 100° horizontal FOV

Figure 9: Oculus DK2 virtual reality mask

COMPUTATION DEVICE (PC)

For real time computation, most of the image processing is performed on the graphic processor (GPU). Part of the code is using the CUDA language, which is dedicated to NVIDIA processors. Thus, the choice of the PC was mainly driven by: having 2 distinct USB3 ports, a NVIDIA GPU, being compact and light for carrying.
We chose a Gigabyte P34G-3 PC equipped with an Intel i7 4700HQ quad-core and an Nvidia gtx 765m GPU. This laptop is well equipped in a small size (14 inches and less than 2kg).
The development has been done in a Microsoft Windows environment.

HARDWARE ASSEMBLY

The 2 cameras have been mounted on a light steel plate which permits to adjust their yaw and pitch angles.

Figure 10: Cameras mounted on the Sony HMD

Figure 11: cameras mounted with protective plastic

The cameras were mounted with a lateral distance of 65mm, which is the commonly agreed average interpupillary distance.
At a first step, and for easy integration, we put the cameras in front of the helmet, but it is suggested that the use of a mirror is preferable [9] and can improve the user motion speed. With the mirror, the cameras could be places at the same distance to the objects as the eyes.

The cameras were slightly tilted to converge at a distance of 4 meters. This gives more comfort for the user, but generates keystone distortion[6]. Nevertheless, we couldn't feel this keystone distortion.

We used 3 set of lenses:

1. Large FOV: focal 2.95mm. Horizontal field angle of 101°. Corresponds to the oculus display
2. Medium FOV : focal 4mm. Horizontal field angle of 84°. Used with the Sony HMZ display, with a crop in the picture to fit the 45° FOV of the display.
3. Small FOV: focal 12mm. Horizontal field angle of 33°.

Figure 12: FOV with the 2.95mm lens

Figure 13: FOV with the 4mm lens

Figure 14: FOV with the 12mm lens

For ease of movement, the Sony headset was connected to a portable battery. Using a laptop, we may get rid of of power cables. The laptop and the portable battery may then be taken and used in a backpack.

4. SOFTWARE DEVELOPMENTS

The software to control the camera and the display to the user has been developed in C++ in a Microsoft Windows environment, using OpenCV library for image processing. C++ is the core language of IDS cameras SDK, Occulus SDK and OpenCV. It is a sufficiently low level language to have a good control of the memory management to optimize the realtime processing.

IMAGE PROCESSING

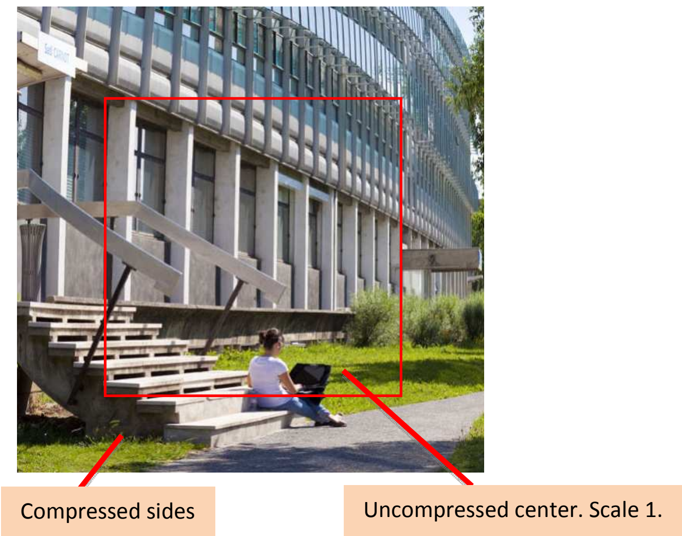
The pictures captured by the camera are not directly sent to the display. Our goal is to facilitate the manipulation of laser hardware in a complex environment, so we need to enhance visual perception of the environment. We chose to simulate a wider field of view by keeping undistorted the central vision and compressing the side vision.

Therefore, the pictures recorded from the cameras needed the following treatment:

- Crop (to adjust the FOV of the cameras to the display
- Center area with compressed sides (for certain tests, to add compressed information on the side)
- Distortion correction (not applied currently), to correct optics aberrations

All these operations can be computed in a single pass by a remap function (available in the Open-CV library). We used the GPU version of the remap function (using Cuda code) in order to speed up the process.

As shown on figure 15, the compression of the sides of the picture is exponential. The more on the side, the more the picture is compressed.



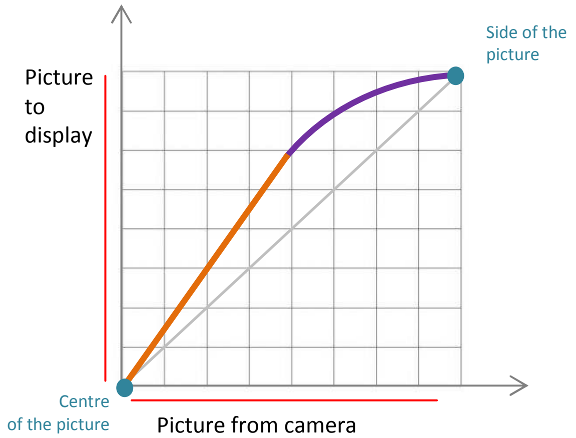


Figure 15: Pixel displacement between original and display. With lateral compression

PROFILES AND SETTINGS

A User Interface (UI) has been developed to control the camera basic parameters such as exposure time and gain. Through this simple UI, the user can also control the software and switch between display profiles or set new profiles. This is especially useful during the development process to select the best profiles (zoom or not, compressed sides or not, etc.)

The user can navigate in the UI using a presentation remote controller with 2 buttons.

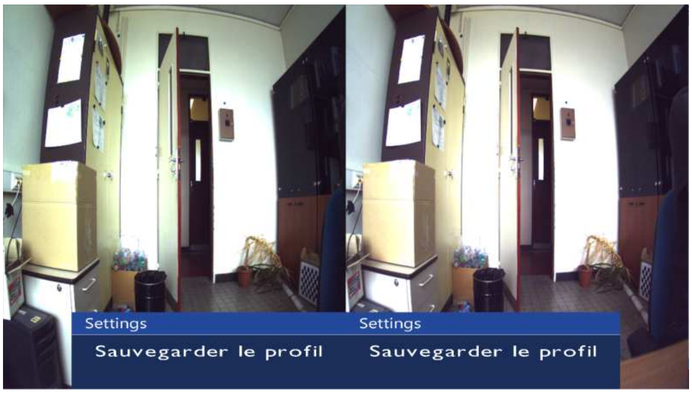


Figure 16: example of the UI menu

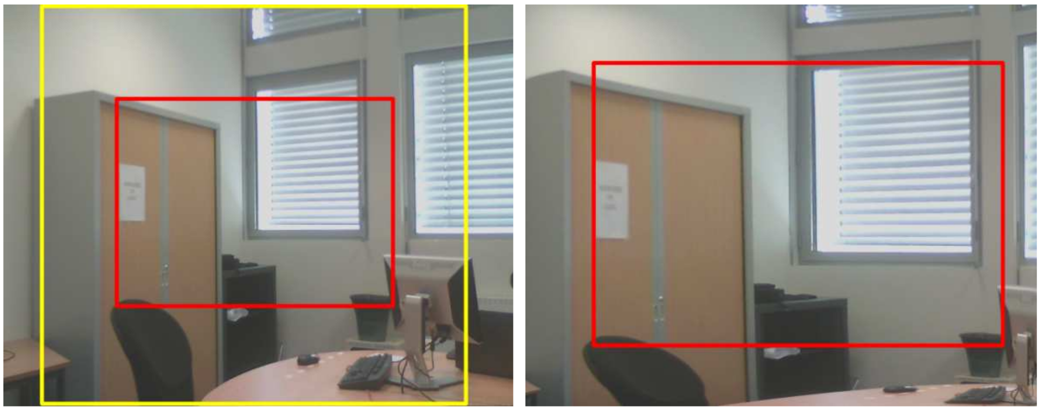


Figure 17 – screen shot with UI display for image selection and adjustment

The UI is generated directly on the GPU, using OpenGL.

The different display profiles are described in an xml file which is read at the program initialization. The user can easily create a new xml profile if necessary.

HIGH DYNAMIC RANGE (HDR)

To insure optimal visibility in certain applications, such as soldering, we tried to implement HDR visualisation. The HDR process builds an high dynamic range image using 2 low dynamic range images taken with different exposure times. The goal is to get a clear vision of low light areas as well as very bright areas. This could be very valuable for soldering work for instance. The user could have a clear view of the soldering arc as well as the environment.

Our IDS cameras can work in a 2 sequential exposure mode with 2 different exposure times.
The main difficulty was to make this HDR in real time.
Our tests have been performed using OpenCV/Cuda library. Unfortunately, the best frame rate we could get was 10 fps, but there is room for optimization. This frame rate is far too slow to allow practical usage. Some teams have developed HDR computation on FPGA board for real-time processing [7], this could be a simple and robust solution to this problem.

AR

One major advantage of our video-see-through glasses is the possibility to add Augmented Reality. The first applications we designed were for laser use. We planned to add simple figures to the scene to show invisible beams and highlight hazardous areas. We also designed the AR capability to display text messages. This text could be used to display and make real-time adjustments, like laser power for instance.

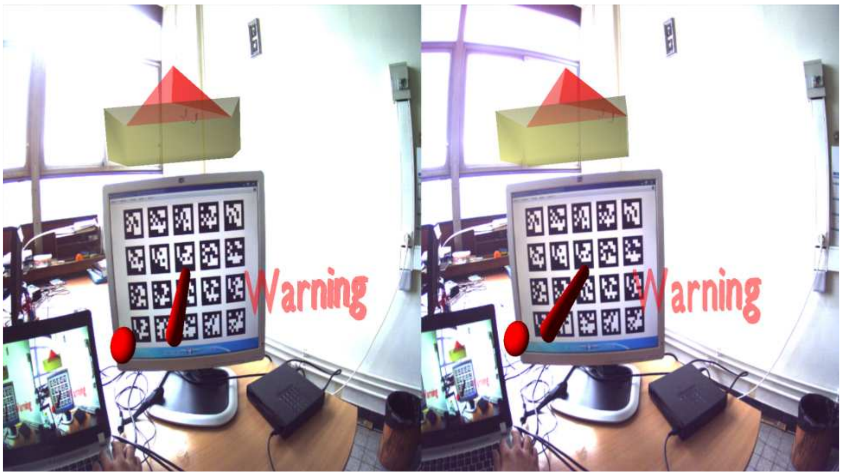


Figure 18: example of different shapes that can be displayed with our AR tool

For each frame, the augmented reality process requires 2 steps:

1. Find the cameras position in space (camera registration)
2. Compute the position of the shapes in each view, corresponding to each eye and draw the virtual shapes in the picture

For the first step, we used markers recognition. We could use markers because we want to use our glasses in a known environment (a lab) with elements always at the same position.
For the markers detection and spatial position computation, we used the ARToolKitPlus library [8]. The computation is performed on the CPU and takes 7ms.

Step 2 is totally performed on the GPU with OpenGL (freeGlut for execution loops, GLU for perspective definition). It takes 1.5 ms to compute with 5 shapes to draw (see figure above).

PROCESSING TIMES

The measurement of the total lag (between reality and display) has been done with the following procedure:

- We displayed a timer on a screen
- The cameras were pointing on the timer
- The camera display (after whole treatment) was displayed in the same screen, on the side of the timer
- We take a screenshot

The difference between the time displayed by the timer and the camera display is the total lag.
The maximum error is directly linked to the refreshing rate of the screen: 60hz = 17ms.

Figure 19: total lag measurement with screenshot of a timer

The measurement of each step of the program was done by using the std::clock function.

Here is a detail of the whole processing time between the image acquisition to its display.

Operation	Time [ms]
Total time of the software (from image reception to availability for display). In which:	Around 35
Markers detection	7
Image copy to GPU	Up to 10
Remap treatment	2
AR computation and drawing	1.5
Incompressible time, not linked to our software. (Camera image reception, through USB3, and display to the HMD)	Around 60
TOTAL time	**100 ms**

The Incompressible times of the cameras were measured using the IDS cockpit software provided by IDS. We change the mode of picture acquisition from Device Independent = through CPU to GPU modes (Direct3D and OpenGL).

Acquisition mode	Total lag (between reality and display)
Device Independent (CPU)	60ms
Direct3D	60ms
OpenGL	110 ms !!

Note: we tested the minimum lag time for different types of cameras (onboard webcam, usb webcam, other IDS camera). It is interesting to see that the lag is always around 60ms (+-17ms)! This looks like an incompressible time for our PC configuration.

STROBOSCOPIC ISSUE WITH PULSED LASERS

Because high energy lasers are generally pulsed lasers, we face stroboscopic problems. With rolling shutter cameras, we had rolling overexposed areas (see above). With global shutter cameras, the flashes can be synchronized to the recorded images, or not. Generally, we see an alternate flashing.

We thought of 2 ways to handle this problem:

1. Replace an overexposed image by the previews image
2. Synchronize, or desynchronize the cameras using an external trigger

Solution 1 can be easily tested, but the laser presence on the picture was not so obvious to determine as it is sometimes only a tiny part of the picture which is over-exposed. The best solution was to check the average exposure of one color canal only, because the laser has a stronger impact in one color only (green in our case). Because this development has been done prior to having an operational second prototype, we have not tested this solution in a real case yet.

Solution 2 is possible with our IDS cameras, which may be externally triggered. To get the laser frequency, a photodiode could be added on the glasses. We have not developed this system yet.

5. TESTS IN CONDITION

USER'S FEELING: FIELD OF VIEW, IMAGE RATE

These tests have been difficult to handle because of the lag between reality and display. Without reducing this lag, it is difficult for the user to judge if the feeling is worst or not in different configuration.
Nevertheless, with the first prototype, we have conducted real tests with 15 different people. They were divided into 3 groups:

A. large field of view (120°) without stereo vision, displayed in the 30° FOV display (thus having the sensation to have everything far from them).
B. The second group had a scale 1 display in the whole display area. Thus having no information of what is situated on the sides of the field of view.
C. The third group had a scale 1 display in the center with a shrinked display of the wide field of view on the sides.

Each candidate should follow the same procedure:

Step 1: get used to the display during 10 minutes, then:

Step 2: Mobility phase

- Walk in a corridor
- Enter a room (by opening a closed door)
- Walk in the room between tables, catch an object
- Walk back to sit

Step 3: static phase

- sit and put washers on a screw

All these steps had to be performed in a minimum of time. All collisions to furniture were recorded.

The results are the following:

Mobility phase

Table 1 mobility tests : total time for the run

	Minimum time	Maximum time	Average time
Group A	35 s	1min 16 s	49,8 s
Group B	50 s	1 min 20 s	56,4 s
Group C	42 s	52 s	47,2 s

Table 2 : Number of collisions

# of collisions	0	1	2	3	4
Group A	5	0	0	0	0
Group B	2	1	1	0	1
Group C	3	2	0	0	0

Static phase

Tableau 3 Static phase : number of washers put on the screw

	Minimum number	Maximum number	Sum of washers put
Groupe A	1 + 1/3	2 + 2/3	10
Groupe B	2+ 2/3	4 + 1/3	17 +2/3
Groupe C	2+2/3	3+2/3	16+2/3

Conclusions :
For mobility, it appears that a lateral view is necessary (faster move and less collisions). Group B has worst results than the 2 others. Surprisingly, the group A, which has a non stereo vision and a large field of view in a shallow display can handle pretty well this test.

In the static phase, we have an opposite result: group B is faster to put washers on the screw. The higher display resolution in that case is crucial to make precise movements. It is also clear that the group A has more difficulties for this exercise. The resolution is probably too low for the needed precision.

In conclusion, we confirm that it is necessary to have a wide FOV for mobility and a high resolution (large FOV unnecessary?) for static and precise work.

AUGMENTED REALITY TESTING

The first usage of Augmented reality is to add information into the display in order to help the user. Our first application is a safety guidance for laser use. We added the following figures:

1. semi-transparent surface over a dye laser, to highlight an hazardous area
2. blue laser beam to show the UV laser beam path.

Camera position tracking is done by sticking reference markers into the scene. We put markers over the laser and on the walls.

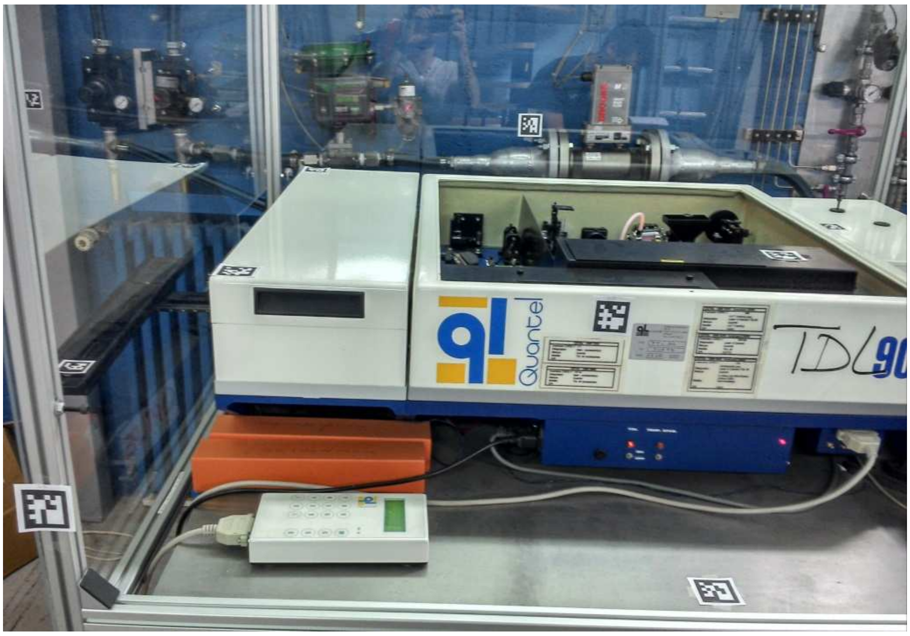


Figure 20: image of the markers stuck on a dye laser

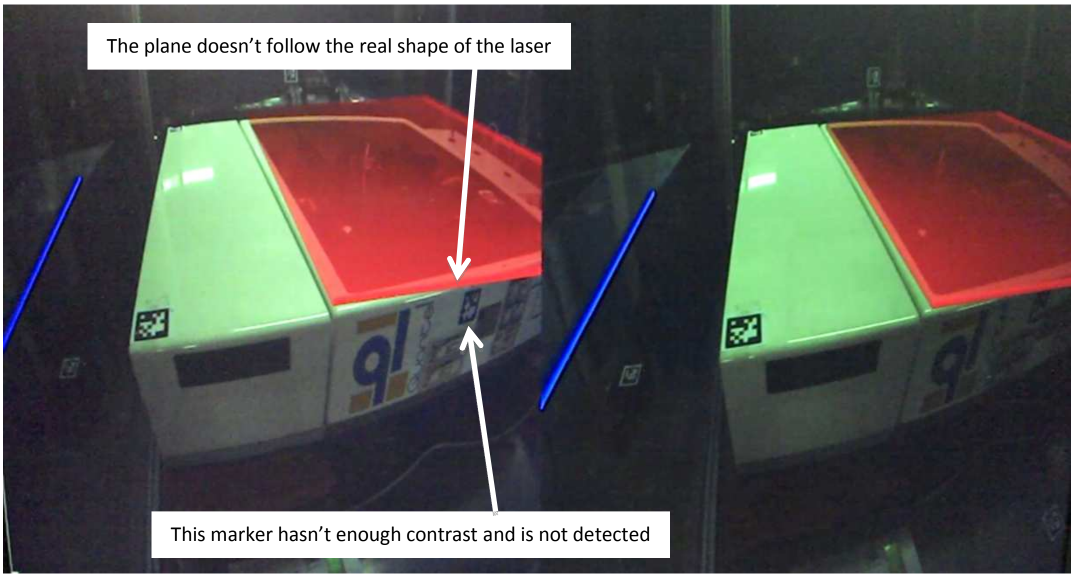


Figure 21: AR over the dye laser (left and right eyes in a side by side display)

We encountered the following issues:

- Blinking of the drawn figures due to a defective analysis of the markers
- Bad matching between the figures and the real object because the camera distortion is not taken into account. On the sides, the drawn figures are not coincident to the real image.

The first tests nevertheless demonstrated that AR is possible and does not slow down the image processing and display in a sensible way (AR treatment takes 8.5ms). It could be a very convenient tool for the user, but it still requires further improvements to be comfortable:

- Improve the markers detection, especially when contrast is weak (dark areas)
- Take into account camera distortion.

6. CONCLUSION AND PERSPECTIVES

PROMISING TECHNOLOGY

The indirect view through cameras is probably the only solution to have a full eye protection at all energies and for the whole light spectra. The technology furthermore allows Augmented Reality. These virtual elements can be very useful in a professional environment, for example by visualising hazardous areas or hazardous beam which could be invisible.

CONSTRAINTS

The current technology has 4 strong constraints:

1. Processing time (lag) between the reality and the display
2. Cameras and display resolutions
3. Field of view
4. Weight and autonomy

The main constraints are known and could be solved in a near future by the natural technology improvements (HMD and cameras are very rapidly improving) or by the development of a dedicated system with an electronic and mechanical design totally built for this usage.

LAG ISSUE

The time between reality and display is called "lag time". This should be minimized as much as possible. We confirmed that 60ms is already sensible.
The lag has a strong influence on the sickness feeling of the user [3].
We determined an incompressible lag of 60ms due to USB port, data transfers and OS management. Adding our computation, the lag was increased to 100ms, which is too high for a comfortable usage of the glasses. This is one improvement to be taken into account in the future.

ANGLE RESOLUTION ISSUE

The Sony, as well as the Oculus HMD, have a lower spatial resolution than human eye. It is difficult to read small characters (such as keyboard for instance). To be more comfortable, the display should have a higher resolution.
Nevertheless, the possibility to switch between a large field of view with low angle resolution (Oculus, 9.6px/degree) to a narrower view with high resolution (Sony, 28.4px/degree) will improve the capabilities of the glasses and make them usable for displacement as well as working.

FIELD OF VIEW

The maximum field of view available on the market is currently the oculus mask (100° FOV) which is still much lower than human vision (175°). Although the human eye doesn't need high resolution on the side since it is only used for navigation.
To date (July 2015), we have not yet tested our prototype with the oculus DK2. The whole image processing is operational, we now need to implement the Oculus SDK.

WEIGHT AND AUTONOMY

Our prototype is based on a laptop PC. The total weight of the system (laptop + HMD controller + HMD glasses + external battery for HMD) is around 10kg. This is definitely a system that couldn't be used in real conditions.
The cables between the HMD and the rest of the system are also uncomfortable.
The final and operational version of the protection glasses should be all weared on the head. It could be a complete helmet with a total weight around 1kg.
The autonomy should also be able to cover one day of work (8 hours).

NEXT STEPS

VALIDATE THE CONFIGURATIONS (FIELD OF VIEW, RESOLUTION, ETC.)

The tests with different people and different configurations should be pursued with the new prototype. For instance, it is still unsure if the addition of lateral compressed information is useful or not when working (static work) with the glasses. Optionally, a small lateral portion of the picture could be compressed to help the user to perceive what lies in the dead angle.

BUILD A MECHANICAL DEVICE TO SWITCH OPTICAL CONFIGURATION

When the optimal configuration will be determined, we will design a mechanical system to switch between the large FOV with low angular resolution to shallow FOV with high angle resolution. We will also add the possibility to use the glasses (or helmet) with laser filters or without any protection. Thus, the user will be able to switch between different shield levels, depending on the hazard and the work to perform.

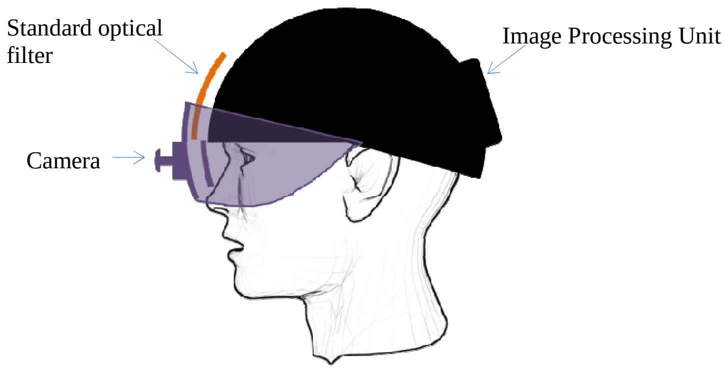


Figure 22: principle of a versatile protection mask

TRANSFER THE TECHNOLOGY TO A LIGHT WEIGHT DEVICE

To come to a versatile system, we must implement the cameras and image processing on a light system to be included on the helmet.
A smartphone-based device would fit most of our constraints. Some smartphones have powerful GPUs such as the Nvidia Tegra. The issue to solve is the communication port between the cameras and the smartphone. The standard MIPI port cannot handle 2 high resolution and high frame rate cameras.
Another way to explore would be to design a dedicated electronic board with FPGA technology for instance. It will reduce the lag to a minimum, but raise the question of implementing a complex process like augmented reality on such a system.

7. REFERENCES

[1] https://developer.oculus.com/documentation/intro-vr/latest/

[2] https://developer.oculus.com/documentation/intro-vr/latest/concepts/bp_intro/

[3] Randall E. Bailey ; Jarvis J. Arthur III ; Steven P. Williams; Latency requirements for head-worn display S/EVS applications. Proc. SPIE 5424, Enhanced and Synthetic Vision 2004, 98 (August 11, 2004); doi:10.1117/12.554462.

[4] *Yanoff, Myron; Duker, Jay S. (2009).* Opthalmogy 3rd Edition*. MOSBY Elsevier. p. 54.* ISBN 0444511415*.*

[5] Wandell, B. (1995). "Foundations of Vision." Sinauer, Sunderland, MA as cited in Neurobiology of Attention. (2005). Eds. Laurent Itti, Geraint Rees, and John K., Tsotos. Chapter 102, Elder, J.H. et al. Elsevier, Inc.

[6] Woods, A. J., Docherty, T., & Koch, R. (1993, September). Image distortions in stereoscopic video systems. In IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology (pp. 36-48). International Society for Optics and Photonics. (http://www.andrewwoods3d.com/spie93pa.html)

[7] Steve Mann et al. REALTIME HDR (HIGH DYNAMIC RANGE) VIDEO FOR EYETAP WEARABLE COMPUTERS, FPGA-BASED SEEING AIDS, AND GLASSEYES (EYETAPS)- 2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 978-1-4673-1433-6
http://www.eyetap.org/papers/docs/HDREyetap_IEEEccece2012paper551.pdf

[8] Daniel Wagner and Dieter Schmalstieg: ARToolKitPlus for Pose Tracking on Mobile Devices. In Proc. 12th Computer Vision Winter Workshop (CVWW'07), Sankt Lambrecht, Austria, February 2007.

[9] M. Park, S. Serefoglou, Hand-Eye Coordination Using a Video-see-through Augmented Reality System, RWTH Aachen University, 2008, 53p.