



**HAL**  
open science

# DEVELOPPEMENT DE LUNETTES VIDÉOS POUR LA PROTECTION OCULAIRE

Eric Mathieu, Jean-Philippe Farrugia, Aurélien Grivollat

► **To cite this version:**

Eric Mathieu, Jean-Philippe Farrugia, Aurélien Grivollat. DEVELOPPEMENT DE LUNETTES VIDÉOS POUR LA PROTECTION OCULAIRE. 15ème Congrès Francophone de Techniques Laser (CFTL), ONERA Toulouse, Sep 2016, Toulouse, France. hal-01388467

**HAL Id: hal-01388467**

**<https://hal.science/hal-01388467>**

Submitted on 28 Oct 2016

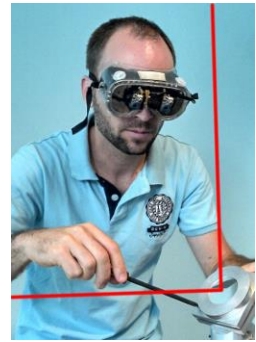
**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# DEVELOPPEMENT DE LUNETTES VIDÉOS POUR LA PROTECTION OCULAIRE



Authors: Eric MATHIEU<sup>1</sup>, Jean-Philippe FARRUGIA<sup>2</sup>, Aurélien GRIVOLLAT<sup>2</sup>

Mots clefs : afficheur tête-haute, sécurité laser, Réalité Augmentée (RA), lunettes vidéo, Equipement de Protection Individuelle (EPI), vision vidéo

## Résumé :

Les travaux présentés dans ce document concernent la protection oculaire pour l'usage des lasers de haute énergie ou haute puissance dans les laboratoires.

En effet, les techniques de filtres actuelles ne permettent pas toujours de se protéger de tous les types de lasers, notamment lorsque trop de longueurs d'ondes sont en présence (lasers blancs ou colorants), ou lorsque l'énergie du laser est trop grande (cas des lasers Femtoseconde notamment).

Ainsi, nous développons des lunettes de protection oculaire via une vision indirecte. Des caméras sont placées devant le regard et diffusent l'image sur des afficheurs devant les yeux. A l'aide de deux caméras, nous conservons une vision stéréoscopique, et donc une perception de la profondeur. L'usage d'une vision synthétique permet également l'ajout d'éléments virtuels (Réalité Augmentée), utiles pour l'utilisateur : puissance laser en phase de réglage, trajet d'un faisceau de longueur d'onde invisible, délimitation de zone de danger... De plus, l'utilisateur conserve la vision de toutes les couleurs, ce qui est particulièrement utile dans certaines phases de réglages (doublage en fréquence...).

Le casque est encore à l'état de prototype. Nous présentons dans ce document les choix d'assemblage optique, mécanique, la programmation associée ainsi que les essais réalisés en situation.

Pour une simplification des développements et des essais, nous avons utilisé une base de PC portable, des caméras industrielles programmables (IDS Ueye) ainsi que des afficheurs du commerce (casque Sony HMZ-T2 et Oculus DK2). Les images sont traitées en temps réel par le PC, avec notamment des calculs sur processeur graphique pour augmenter la vélocité.

Les essais utilisateurs ont été réalisés sur le premier prototype et ont montré la pertinence du concept, avec notamment les limitations suivantes :

- Le décalage temporel entre la prise d'image et l'affichage est extrêmement sensible et doit se limiter à moins de 48ms
- Pour les déplacements, il est fondamental d'avoir un champ de vision large (l'idéal étant 175°)
- Pour le travail sur poste, la résolution est primordiale, le champ de vision peut être restreint (1 arcminute correspondant à la vision humaine)
- Les lasers pulsés induisent un phénomène stroboscopique. Il est nécessaire de se désynchroniser des pulses

Ainsi, nous envisageons les évolutions et développements futurs :

- Casque « tout-en-un » avec traitement des images embarqué sur carte électronique ou base smartphone
- Possibilité d'alterner entre une vision naturelle, une vision par simple filtre « classique » et vision vidéo pour les phases critiques
- Analyse de la fréquence des lasers pulsés pour désynchronisation des images caméras
- Amélioration de la Réalité Augmentée avec notamment une meilleure détection des marqueurs de positionnement

---

<sup>1</sup> CETHIL : Centre d'Energétique et de Thermique de Lyon, 9 rue de la physique - 69621 Villeurbanne cedex, France

<sup>2</sup> LIRIS : Laboratoire d'Informatique en Image et Systèmes d'information, Bâtiment Nautibus, 8 boulevard Niels Bohr - 69622 Villeurbanne Cedex, France

## 1. REMERCIEMENTS

Les auteurs tiennent à remercier le laboratoire du CETHIL et ses directeurs successifs pour leur soutien au développement de ce projet. Le CETHIL a financé en interne l'intégralité des développements présentés dans ce document, ainsi que les stages étudiants associés.

Le laboratoire LIRIS est également partenaire et a fourni une aide précieuse aux développements informatiques à travers l'expertise et le temps consacré par Mr Farrugia.

## 2. INTRODUCTION

Actuellement, la seule technique disponible pour se protéger les yeux lors des expositions dangereuses (soudure, lasers) consiste en des filtres optiques. Il existe 2 types de filtres : les filtres d'intensité réduisent l'intensité lumineuse de l'ensemble du spectre visible, les filtres sélectifs, ou passe-bande, laissent une partie du spectre visible atteindre l'œil (et stoppent les longueurs d'onde liées au laser utilisé par exemple).



Figure 1: Exemple d'un masque de protection laser utilisant un filtre sélectif

Les filtres ne peuvent pas protéger (i) des énergies lumineuses trop fortes (cas des lasers haute énergie) et (ii) de trop de longueurs d'ondes visibles en même temps (cas rencontré avec plusieurs lasers ou lasers blancs). De plus, ces filtres occultent, par nature, certaines couleurs du spectre visible.

Afin de contourner ces inconvénients, nous avons développé un casque de vision indirecte équipé de 2 caméras et 2 afficheurs devant chaque œil.

Cette solution présente plusieurs avantages :

- (i) protection totale, puisque la lumière ne peut pas atteindre l'œil directement,
- (ii) possibilité d'ajouter de l'information pertinente pour l'utilisateur (Réalité Augmentée),
- (iii) conservation de la vision des couleurs.

Pour conserver la perception 3D de la profondeur, nous utilisons un système stéréoscopique avec 2 caméras.

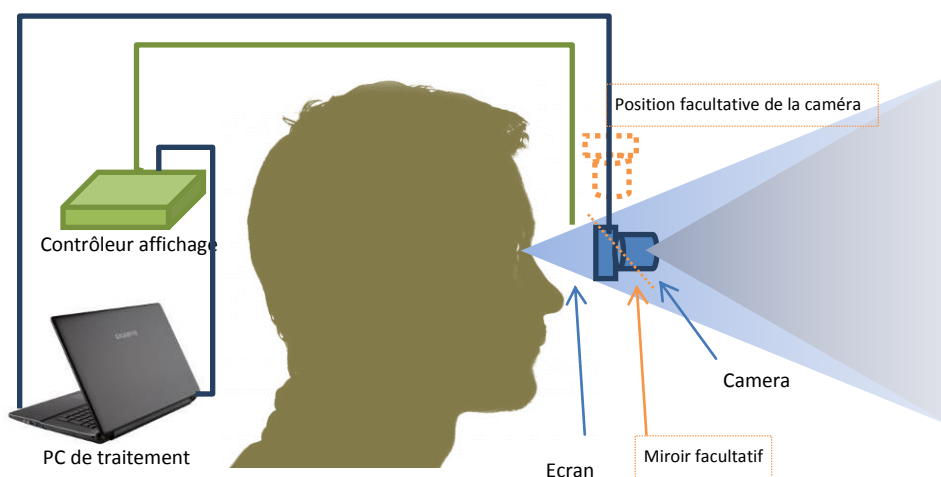


Figure 2: Description du système

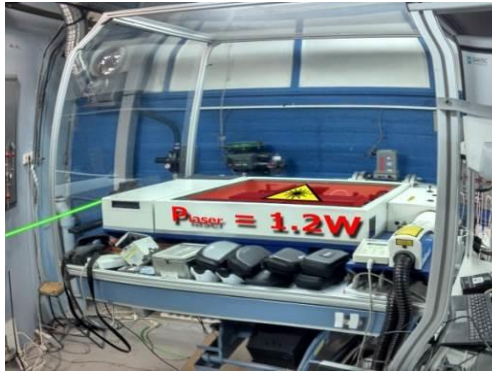


Figure 3: Exemple de Réalité Augmentée sur un laser à colorants. Visualisation de faisceau invisible, délimitation de la zone de danger, affichage de la puissance laser lors des réglages...

Ce projet a été initié en 2012 au laboratoire CETHIL. Le CETHIL utilise en effet des lasers de haute énergie et de nombreuses longueurs d'ondes sont parfois en présence sur les expériences. Depuis 2013, le laboratoire LIRIS est impliqué dans le projet et apporte son expertise dans les domaines de la vision informatique et du traitement sur processeur graphique et système embarqué.

Jusqu'en 2015, le principal objectif du projet était de démontrer la pertinence du concept et sa faisabilité.

Ce document est structuré comme suit.

Après une brève introduction, nous présentons le matériel et la structure du programme informatique (chapitres 3 et 4). Le chapitre 5 résume les tests utilisateurs effectués en condition de travail. Le chapitre 6 conclut sur les développements futurs et les perspectives.

### 3. CHOIX MATÉRIELS

Dans ce chapitre, nous présentons les choix matériels qui ont été réalisés. Connaissant les contraintes, nous avons sélectionné le meilleur matériel possible, disponible dans le domaine public (le marché militaire n'était pas une option).

#### CONTRAINTES

Nous avons identifié les contraintes suivantes pour la sélection du matériel :

- Caméras relativement rapides (au moins 30hz pour chaque œil) afin de garantir un confort de vision raisonnable. Par expérience, nous estimons que 50hz serait préférable. Les systèmes de réalité virtuelle les plus avancés, comme l'Oculus, recommandent 75hz. [1]
- Délai minimal entre la prise d'image et l'affichage. Moins de 20ms est recommandé par l'afficheur Oculus [2], bien que ce délai devienne sensible à l'utilisateur (et soit à l'origine de nausées) au-dessus de 48ms [3].
- Caméras de haute sensibilité pour les environnements sombres.
- Bonne résolution angulaire. Pour se conformer à la vision humaine, il faudrait un afficheur avec une résolution angulaire de 1 minute d'arc (environ  $0.02^\circ$ ) [4].
- Large champ de vision (angle de champ,  $\alpha$ ), notamment pour les phases de déplacement, afin que l'utilisateur n'heurte pas d'objets autour de lui. Idéalement, le champ de vision devrait approcher les  $175^\circ$  (vision humaine [5]).
- Poids léger. A minima pour la partie qui sera placée sur la tête de l'utilisateur.

#### CHOIX DU MATÉRIEL

Afin de faciliter les développements et la maintenance, le logiciel a été mis au point sur un PC portable, avec des caméras USB et des afficheurs tête-haute du commerce. Le code lui-même pourrait tourner sur un smartphone ou une tablette, mais ces systèmes ne possèdent pas actuellement de connexion USB haute vitesse (USB3) pour connecter les caméras.

Le premier prototype a été développé à partir de lunettes VUZIX Wrap 920 AR déjà équipées avec 2 caméras (640\*480px) et 2 afficheurs (640\*480px, angle de champ horizontal de  $30^\circ$ ).



Figure 4: les lunettes Vuzix Wrap 920 AR



Figure 5: caméras et afficheurs Vuzix montés sur un masque de protection

Ce prototype a démontré le potentiel de cette technologie, mais a montré des limites :

- Les caméras avec obturateur déroulant (rolling shutter) sont de type webcams et incompatibles avec des lasers pulsés (effets stroboscopiques)
- Trop faible résolution
- Champ de vision trop étroit

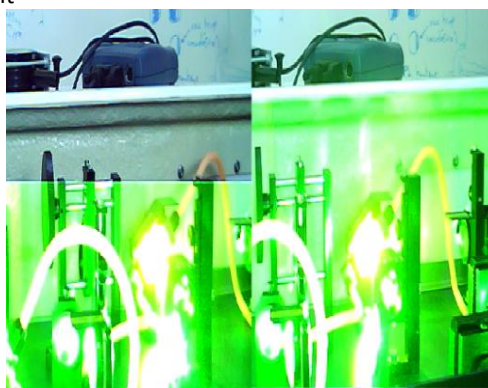


Figure 6: exemple de saturation laser sur des caméras avec obturateur « rolling shutter » (œil droit et gauche)

## CAMERAS

Notre choix a été guidé par la nécessité d'utiliser des ports USB rapides (USB3) permettant de faire transiter les données de 2 caméras haute résolution avec un taux de rafraîchissement supérieur à 50 Hz.

Nous avons choisi les IDS UI-3251LE



Figure 7: IDS UI3251 camera

- Capteur CMOS de diagonale 1/1.8"
- Interface USB3
- 1600\*1200px
- Obturateur : Rolling ou global shutter
- 60 Hz max
- Monture M12 (pour changement facile des optiques)
- Dimensions H/Larg/Long : 36,0 mm x 36,0 mm x 20,2 mm
- Poids : 12g
- Contrôle manuel de : exposition, fréquence, couleurs, etc.

## AFFICHEURS VIDÉO (LUNETTES)

- Sony HMZ-T2
  - 2 écrans OLED
  - 1280x720 pixels (résolution angulaire = 28.4 pixels/degré)
  - Angle de champ horizontal de 45°



Figure 8: Lunettes Sony HMZ-T2

Un second afficheur a été acquis ultérieurement en 2014 afin d'obtenir un angle de champ plus large : Oculus DK2

- Ecran OLED
- Résolution (par œil) : 960 x 1080 px (résolution angulaire = 9.6 pixels/degré)
- Angle de champ horizontal : 100°



Figure 9: Oculus DK2

## UNITÉ DE TRAITEMENT (PC)

Afin de réaliser des traitements en temps réel, la plupart des calculs sont réalisés sur le processeur graphique (GPU). Une partie du code utilise le langage CUDA qui est dédié au GPU NVIDIA. Ainsi, le choix du PC était principalement gouverné par les contraintes suivantes : 2 ports USB3 distincts, un processeur graphique NVIDIA, compacité et poids léger pour les essais en mobilité.

Nous avons choisi un PC Gigabyte P34G-3 PC équipé du processeur Intel i7 4700HQ quad-core et du processeur graphique Nvidia GTX 765M. Ce portable est bien équipé dans une dimension réduite (14 pouces et moins de 2kg).

Les développements ont été réalisés en C++, sous Microsoft Windows.



## ASSEMBLAGE DES COMPOSANTS

Les 2 caméras ont été montées sur une plaque support légère et appuyées sur des entretoises souples permettant ainsi d'ajuster les angles verticaux et horizontaux des caméras.



Figure 10: Caméras montées sur le casque Sony



Figure 11: caméras montées, avec une protection plastique

Les caméras sont séparées d'une distance de 65mm, correspondant à la distance interpupillaire moyenne. Dans un premier temps, afin de simplifier le montage, nous avons placé les caméras à l'avant du casque. Mais l'utilisation d'un miroir serait probablement préférable et améliorerait les vitesses de déplacement de l'utilisateur [9]. Avec un miroir, les caméras peuvent être placées à la même distance que les yeux, d'un objet.

Les caméras ont été montées avec une légère convergence correspondant à une distance de 4 mètres. Ceci procure plus de confort à l'utilisateur, mais est source de distorsion [6]. Néanmoins, nous n'avons pas ressenti cette distorsion lors de nos essais.

Nous avons utilisé 3 paires de lentilles, monture M12:

1. Large angle de champ ( $\alpha$ ) : focale de 2.95mm. Angle de champ horizontal de 101°. Correspondant au champ de vision dans les lunettes Oculus.
2. Angle de champ moyen : focale de 4mm. Angle de champ horizontal de 84°. Utilisé avec les lunettes Sony HMZ, avec un découpage de l'image pour correspondre aux 45° de champ horizontal de l'afficheur.
3. Petit angle de champ : focale de 12mm. Angle de champ horizontal de 33°.



Figure 12:  $\alpha$  avec focale de 2.95mm



Figure 13:  $\alpha$  avec focale de 4mm



Figure 14:  $\alpha$  avec focale de 12mm

Pour faciliter les déplacements de l'utilisateur, le casque peut être connecté à une batterie. Associé au PC portable, nous pouvons intégrer l'ensemble dans un petit sac à dos.

## 4. DÉVELOPPEMENTS INFORMATIQUES

Le logiciel de contrôle des caméras et l'affichage utilisateur ont été développés en C++ sous environnement Windows, et fait appel aux bibliothèques OpenCV pour le traitement des images. Le C++ est le langage utilisé par les SDK des caméras, des lunettes Oculus, ainsi que la bibliothèque OpenCV. C'est un langage de suffisamment bas niveau pour contrôler au mieux les accès mémoire et optimiser le code pour un traitement visant le temps-réel.

### TRAITEMENT DES IMAGES

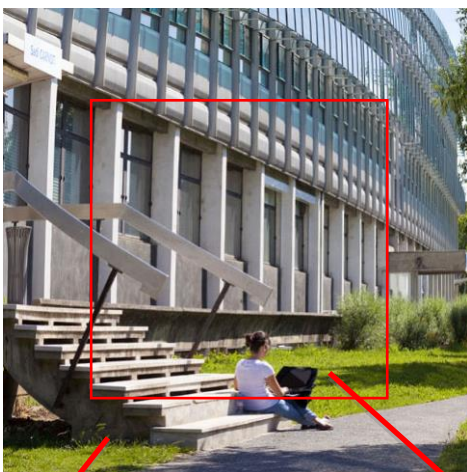
Les images capturées par les caméras ne sont pas directement envoyées à l'afficheur. Notre but est de faciliter les manipulations de lasers dans un environnement complexe, aussi essayons-nous d'améliorer la perception de l'environnement. Pour certains essais, nous avons ainsi simulé un plus grand champ de vision en conservant la zone centrale à l'échelle et sans distorsion, avec des côtés comprimés. L'utilisateur a ainsi une perception de ce qui se trouve sur les côtés de l'image. L'utilisateur peut ainsi éviter des objets.

Les images enregistrées par les caméras doivent donc subir les traitements informatiques suivants :

- Découpage (afin d'ajuster l'angle de champ au champ de vision des afficheurs)
- Zone centrale avec côtés comprimés, pour certains essais.
- Correction de distorsion (non appliquée actuellement), afin de corriger les distorsions optiques.

Toutes ces opérations peuvent être effectuées en une seule passe par une fonction de remapage (remap, disponible dans la bibliothèque OpenCV). Afin d'accélérer le calcul, nous avons utilisé la version GPU de la fonction remap (utilisant du code Cuda).

Comme montré dans la figure 15, la compression des côtés de l'image est exponentielle. Plus on s'éloigne du centre, plus l'image est comprimée.



Côtés comprimés

Centre echelle 1

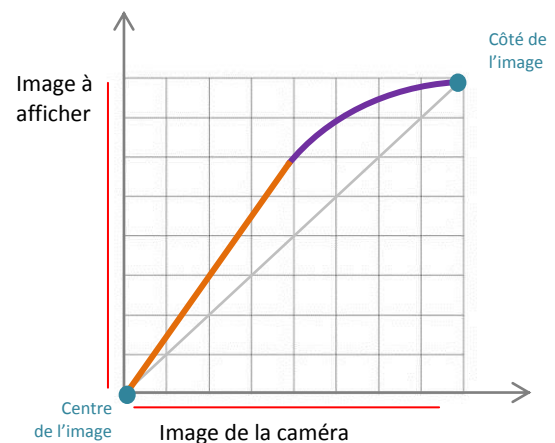


Figure 15: Déplacement des pixels entre l'image originale et l'affichage. Avec compression latérale.

### PROFILS ET REGLAGES

Une Interface Utilisateur (IU) a été développée afin de contrôler les paramètres de base de la caméra tels que le temps d'exposition et le gain. A travers cette interface, l'utilisateur peut également contrôler le programme et alterner entre les profils de visualisation ou en créer de nouveaux. Ceci est précisément nécessaire lors des phases de développement afin de sélectionner le meilleur profil pour un affichage optimal (zoomé ou non, côtés comprimés ou non, etc.)

L'utilisateur peut naviguer dans l'IU à l'aide d'une simple télécommande 2 boutons utilisée pour les présentations.





Figure 16: exemple du menu de l'IU

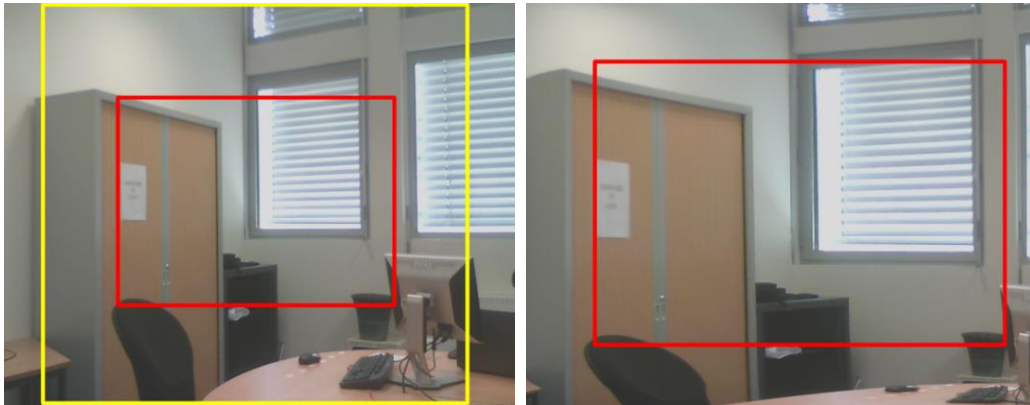


Figure 17 – Copies d'écrans avec les zones pour les réglages et ajustements images

L'interface utilisateur est générée directement sur le processeur graphique GPU, par un script OpenGL.

Les différents profils d'affichage sont décrits dans des fichiers xml, lus par le programme lors de son démarrage. L'utilisateur peut très facilement créer un nouveau fichier xml si nécessaire.

### IMAGE HAUTE DYNAMIQUE - HIGH DYNAMIC RANGE (HDR)

Afin d'optimiser la visibilité pour certaines applications telles que la soudure, nous avons essayé d'intégrer la visualisation haute dynamique, connue sous le nom anglais High Dynamic Range (HDR).

L'affichage HDR est réalisé à partir de plusieurs images (2 images dans notre cas) prises avec des temps d'exposition différents. Ainsi, il est possible d'obtenir une vision non saturée sur l'image finale de zones peu illuminées ainsi que de zones très lumineuses. Ceci peut être un élément très avantageux pour des applications type soudure. L'utilisateur pouvant ainsi conserver une bonne vision du point de soudure ainsi que de l'environnement.

Notre caméra IDS peut fonctionner avec un mode séquentiel double exposition. La plus grande difficulté était de réaliser ce calcul HDR en temps réel.

Notre calcul de construction de l'image HDR utilise la librairie OpenCV/Cuda. Malheureusement, le meilleur taux de rafraichissement que nous ayons obtenu est de 10 images/secondes. Il y a néanmoins de grandes possibilités d'optimisation. Ce taux de rafraichissement est bien trop bas pour une utilisation pratique. Des équipes ont déjà réalisé des images HDR en temps réel par calculs sur FPGA [7]. Cette option pourrait être une solution simple et robuste.

### RÉALITÉ AUGMENTÉE (RA)

L'un des principaux avantages de la vision par vidéo interposée est la possibilité d'ajouter des éléments virtuels ; c'est-à-dire, construire de la Réalité Augmentée.

Les premiers tests que nous ayons réalisés étaient pour des applications lasers. Nous avons ajouté des formes simples à l'image, pour par exemple : visualiser un faisceau laser invisible, délimiter une zone de danger. Nous pouvons également ajouter du texte. Ce texte pourrait par exemple afficher en temps réel des éléments nécessaires en phase de réglage, tels que la puissance du laser.

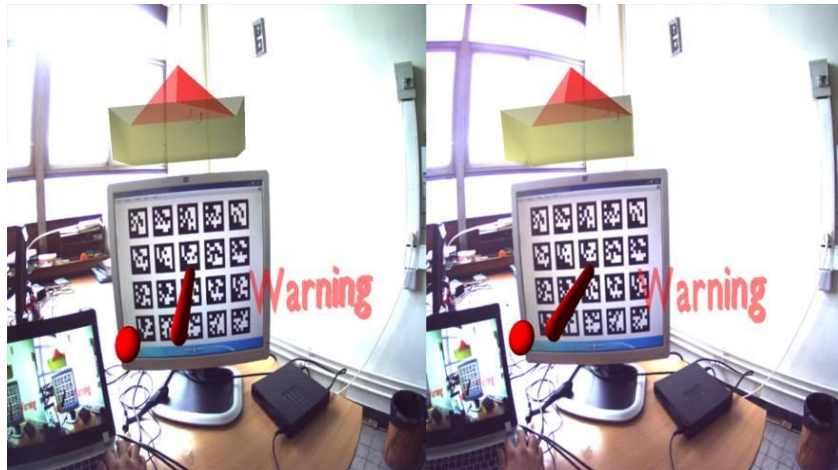


Figure 18: exemple de plusieurs formes que nous pouvons ajouter avec notre programme de Réalité Augmentée (œil gauche et droit)

Pour chaque image, le calcul de réalité augmentée nécessite 2 étapes :

1. Trouver la position de la caméra dans l'espace (camera registration)
2. Calculer la position des formes pour chaque vue correspondant à chaque œil et dessiner les éléments virtuels dans l'image.

Pour l'étape 1, nous utilisons la reconnaissance de marqueurs. Nous pouvons utiliser des marqueurs car l'utilisateur de nos lunettes se trouve dans un environnement connu (laboratoire) avec des éléments se trouvant toujours à la même position. Il est donc possible de coller des marqueurs dans la salle. L'utilisation de marqueurs facilite grandement la tâche de positionnement dans l'espace. Nous pouvons ainsi réaliser l'ajout de Réalité Augmentée en temps réel. Nous utilisons pour cette étape la librairie ARToolKitPlus [8]. Le calcul est réalisé sur le CPU en 7ms.

L'étape 2 est totalement réalisée sur le GPU avec OpenGL (freeGlut pour l'exécution de boucles (loops), GLU pour la définition de perspective). Ce calcul prend 1.5 ms pour afficher 5 formes géométriques (cf. image ci-dessus).

## TEMPS DE CALCUL

La mesure du retard total entre l'affichage et la réalité a été réalisée par le procédé suivant :

- Une horloge défile sur un écran
- Les caméras visent l'écran
- L'affichage des images caméras (après traitement total) est affiché sur le même écran, à côté de l'horloge.
- On fait une copie d'écran à un instant T

La différence du temps affiché entre l'horloge et les images caméra correspond au retard total. L'erreur maximale dépend du taux de rafraîchissement de l'écran, soit 60hz = 17ms.



Figure 19: retard total mesuré par copie d'écran d'une horloge

La mesure du temps d'exécution de chaque étape du programme a été réalisée en utilisant la fonction C++ `std::clock` function.

Ci-dessous, le résumé des temps d'exécution pour chaque étape du programme d'acquisition et d'affichage.

Opération	Temps [ms]
Temps total du programme (de la réception image à la disponibilité pour affichage). Détails :	environ 35
Détection des marqueurs	7
Copie des images sur GPU	jusque 10
Traitement « remap »	2
Calcul et ajout de la Réalité Augmentée	1.5
Temps incompressible, non lié à notre logiciel (réception images caméras via USB3, transfert vers afficheur)	environ 60
<b>Temps total</b>	<b>100 ms</b>

En utilisant le logiciel caméra fournit par IDS (IDS cockpit), nous avons testé différents modes d'acquisition d'images :

Mode d'acquisition	Retard total (entre affichage et réalité)
Device Independent (CPU)	60ms
Direct3D	60ms
OpenGL	110 ms !!

Note : nous avons testé différents retards obtenus avec différentes caméras (webcam intégrée, webcam externe, autre caméra IDS). Il est intéressant de noter que ce retard est toujours autour de 60ms (+-17ms) ! Ceci semble donc être un temps minimum incompressible pour obtenir et afficher des images sur notre PC (windows 7, USB3).

## EFFETS STROBOSCOPIQUES AVEC LES LASERS PULSÉS

Les lasers de haute énergie sont généralement pulsés. Ceci résulte en des effets stroboscopiques avec les caméras. Pour des caméras avec obturateur déroulant, on obtient un balayage de zones saturées (voir plus haut). Avec des caméras à obturateur global, la prise d'image peut être synchronisée sur les flashes lasers afin de les visualiser, ou au contraire de s'en affranchir.

Nous avons imaginé deux façons de gérer ce problème :

1. Remplacer une image surexposée pour l'image précédente
2. Synchroniser, ou désynchroniser les caméras en utilisant un signal de synchronisation externe (trig)

La solution 1 peut très facilement être testée. Néanmoins, la présence ou non du pulse laser dans l'image n'est pas si triviale à déterminer car la lumière occupe parfois une petite portion de l'image. La meilleure solution était de suivre l'exposition moyenne sur un canal couleur et repérer les pics d'intensité (canal vert dans notre cas puisque le laser était vert). Ces développements ont été réalisés avant d'avoir un second prototype opérationnel. Nous n'avons donc pas encore mené d'essais in-situ.

La solution 2 est possible avec nos caméras IDS car elles peuvent recevoir un signal de trig externe. Pour récupérer la fréquence laser, une photodiode pourrait être utilisée et ajoutée sur les lunettes. Nous n'avons pas développé cette solution pour le moment.

## 5. TESTS EN CONDITION

### SENSATION UTILISATEUR : CHAMP DE VISION ET FRÉQUENCE IMAGE

Ces tests ont été difficiles à réaliser en raison du retard d'affichage incompressible (voir précédemment). A cause de ce retard, il est difficile d'estimer si la sensation de l'utilisateur est meilleure ou pas en fonction des configurations testées.

Néanmoins, à l'aide du premier prototype, nous avons conduit des tests en condition réelle sur 15 utilisateurs. Ils étaient divisés en 3 groupes

- Le premier groupe était équipé d'un large angle de champ caméra (120°), sans vision stéréoscopique. Champ de vision de 30° ; donc une sensation d'éloignement.
- Le second groupe avait un affichage entièrement à l'échelle 1. Ainsi, aucune information en dehors du champ de vision, lequel était donc restreint à 30°.
- Le troisième groupe avait un affichage central à l'échelle 1 et une zone comprimée que les côtés afin de visualiser (de façon modifiée) le champ de vision périphérique.

Chaque candidat devait suivre la même procédure :

Etape 1: phase d'acclimatation durant 10 minutes, ensuite

Etape 2: phase de mobilité :

- Marcher dans le couloir
- Entrer dans une salle (en ouvrant une porte fermée)
- Cheminer entre des tables et attraper un objet
- Cheminer en sens inverse et aller s'asseoir

Etape 3: phase statique

- S'asseoir et enfiler des rondelles sur une vis

Toutes ces étapes devaient être réalisées dans un minimum de temps. Toutes les collisions étaient enregistrées.

Les résultats sont les suivants :

#### Phase de mobilité

Table 1 tests mobilité : temps total

	Temps minimum	Temps maximum	Temps moyen
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 : Nombres de collisions

Nb de 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

#### Phase statique

Tableau 3 Phase statique : nombre de rondelles enfilées sur la vis

	Minimum	Maximum	Nb total de rondelles
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 :

Pour la mobilité, il semble qu'une vision latérale soit nécessaire (mouvements plus rapides et moins de collisions). Le groupe B a les plus mauvais résultats. Étonnamment, le groupe A réalise un bon score à ce test alors que sa vision est très déformée car lointaine.

Pour la phase statique, nous avons le résultat opposé : le groupe B est plus rapide pour réaliser cet exercice. La meilleure résolution d'affichage est probablement cruciale dans ce cas afin de contrôler des mouvements précis. Il est également clair que le groupe A a plus de difficultés pour cet exercice. La résolution d'affichage est probablement trop basse pour réaliser ce travail de précision.

En conclusion, nous confirmons qu'il est nécessaire d'avoir un grand champ de vision pour se déplacer, et une grande résolution angulaire (grand champ de vision facultatif ?) pour réaliser un travail statique précis.

Il faudra toutefois confirmer ces essais sur un nombre plus important de sujets.

## TESTS DE RÉALITÉ AUGMENTÉE

Notre première application de Réalité Augmentée est l'affichage d'éléments pour signaler les dangers à un utilisateur de laser. Nous avons ajouté les éléments virtuels suivants :

1. Surface semi-transparente au-dessus d'un laser colorant décapoté. Ceci délimite une zone de danger.
2. Faisceau bleu pour symboliser le trajet du faisceau laser UV

Le suivi de la position caméra est réalisé par des marqueurs disposés dans l'environnement : sur le laser et sur les murs de la salle.



Figure 20: image des marqueurs sur un laser à colorants

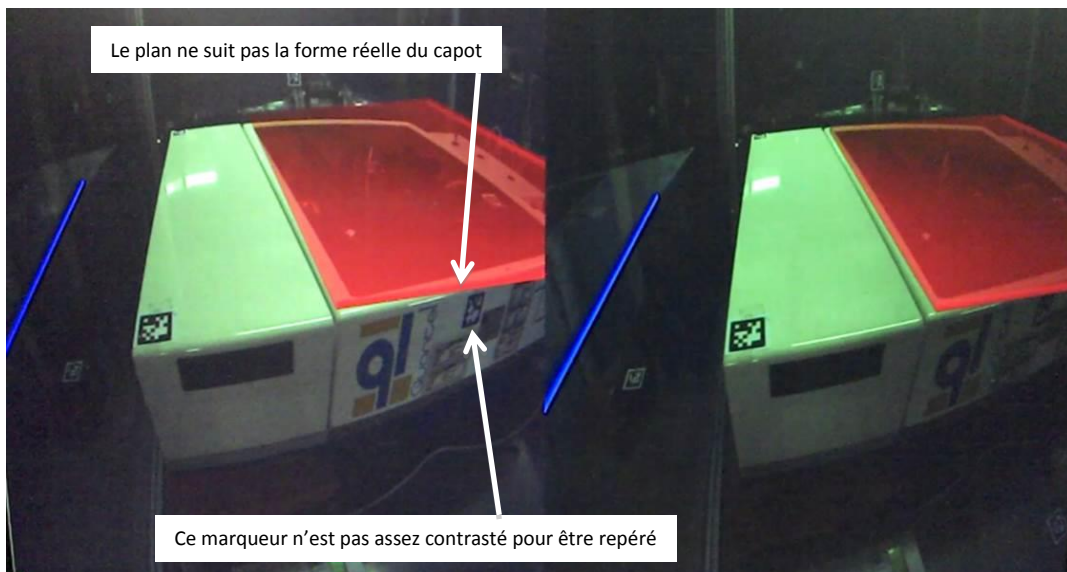


Figure 21: Visualisation des éléments de RA sur le laser colorant (œil gauche et droit)

Nous avons rencontré les difficultés suivantes :

- Clignotement des éléments virtuels en raison des défauts de repérage des marqueurs
- Mauvais recouvrement des éléments virtuels sur l'image en raison de la distorsion optique qui n'était pas prise en compte (essentiellement sur les côtés de l'image)

Les premiers essais ont néanmoins démontré que la Réalité Augmentée est possible et ne ralentit pas de manière perceptible le traitement et l'affichage des images (temps de traitement RA = 8.5ms). Ce pourrait être un outil très utile pour l'utilisateur, mais il nécessite encore de nombreuses améliorations pour devenir confortable :

- Améliorer la détection des marqueurs, notamment lorsque le contraste est bas (zones sombres)
- Prendre en compte la distorsion optique des caméras.



## 6. CONCLUSION ET PERSPECTIVES

### UNE TECHNOLOGIE PROMETEUSE

La vue indirecte à travers des caméras est probablement la seule solution pour protéger les yeux des émissions radiatives à toutes les énergies et sur tout le spectre lumineux. De plus, cette technologie permet d'ajouter des éléments virtuels (RA). Ces éléments peuvent être extrêmement utiles dans un environnement professionnel, par exemple pour visualiser des zones de danger, ou des faisceaux qui seraient invisibles.

### CONTRAINTES

La technologie actuellement mise en œuvre fait face à 4 contraintes principales :

1. Retard entre l'affichage numérique et la réalité
2. Manque de résolutions des caméras et des afficheurs
3. Champ de vision étroit
4. Poids et autonomie

Ces contraintes sont connues et seront peut-être résorbées dans un avenir proche par l'évolution naturelle de ces technologies émergentes. Les afficheurs tête haute et les caméras évoluent très rapidement. Nous pourrions aussi développer un système dédié à cette application avec une électronique spécifique et un montage opto-mécanique adapté.

#### PROBLÈME DE RETARD D'AFFICHAGE

Le retard d'affichage (lag) doit être minimisé. Nous avons confirmé qu'un décalage temporel entre la réalité et l'affichage de 60ms est perceptible et dérangent.

Ce retard a une influence capitale dans la sensation de malaise de l'utilisateur [3].

Nous avons déterminé un retard incompressible de 60ms lié à l'utilisation d'un port USB, à la transmission des données et à la gestion par l'OS (windows).

En ajoutant notre traitement de données, le retard a été augmenté à 100ms, ce qui est trop grand pour un usage confortable des lunettes. C'est donc un axe d'amélioration capital pour le futur.

#### RÉSOLUTION ANGULAIRE

Les afficheurs Sony et Oculus ont une résolution angulaire inférieure à la vision humaine. Il est ainsi difficile de lire des petits caractères, comme sur un clavier par exemple. Pour être plus confortables, les afficheurs devront avoir une meilleure résolution angulaire.

Néanmoins, la possibilité d'alterner entre une vision grand champ avec basse résolution angulaire (Oculus, 9.6px/degré) et une vision champ étroit et haute résolution angulaire (Sony, 28.4px/degré) permettrait d'améliorer l'expérience utilisateur en lui permettant de se déplacer, mais aussi de réaliser un travail statique de précision.

#### CHAMP DE VISION

Le champ de vision le plus large actuellement disponible sur le marché est le masque Oculus (100°), ce qui reste très inférieur à la vision humaine (175°). Néanmoins, on note que l'œil humain n'a besoin d'une haute résolution qu'au centre de la vision et une résolution dégradée en périphérie

A la date des derniers essais (juillet 2015), nous n'avons pas encore testé le casque avec afficheur Oculus DK2. Le traitement des images est opérationnel, il restera à implémenter le SDK Oculus.

#### POIDS ET AUTONOMIE

Notre prototype est basé sur un PC portable. Le poids total du système (PC + contrôleur lunettes + lunettes vidéo + batterie externe lunettes) est d'environ 10kg. C'est un système qui ne pourrait bien-entendu pas être utilisé en l'état de manière opérationnelle. Les câbles entre les afficheurs et le reste du matériel sont également inconfortables.

La version finale et opérationnelle devra être totalement intégrée dans un casque. Elle pourrait ressembler à un casque de 1kg environ. L'autonomie devra également couvrir un usage d'une journée (8h).



## PROCHAINES ÉTAPES

### VALIDER LES CONFIGURATIONS (CHAMP DE VISION, RÉOLUTION, ETC.)

Les tests sur des personnes différentes et des configurations différentes doivent être poursuivis avec le nouveau prototype.

Par exemple, il n'est pas encore certain que l'addition d'une information latérale comprimée soit utile pour l'utilisateur. Une petite portion latérale de l'image pourrait, de manière optionnelle, contenir une vue comprimée et donner à l'utilisateur une perception du contenu des angles morts.

### MONTAGE OPTOMÉCANIQUE POUR ALTENER LES CONFIGURATIONS OPTIQUES

Lorsque nous aurons défini les configurations optimales, nous réaliserons un montage opto-mécanique pour alterner entre un grand champ de vision et un plus faible champ. Nous pensons également conserver la possibilité d'utiliser des filtres lasers classiques. Ainsi, l'utilisateur pourra, en fonction des conditions, choisir une vue naturelle, via des filtres, ou par vidéo interposée.

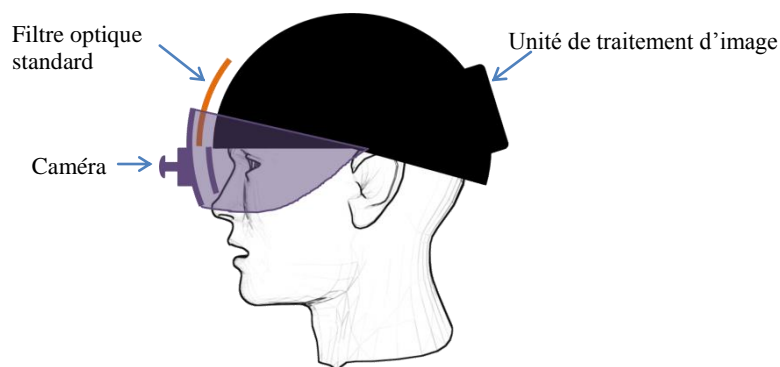


Figure 22: principe d'un casque de protection à vision modulable

### PORTER LES TECHNOLOGIES SUR UN SYSTÈME EMBARQUÉ LÉGER

Nous souhaitons intégrer l'ensemble du processus d'acquisition et de traitement des images dans le casque. Une base type smartphone pourrait répondre à ce besoin. Certains smartphones ont des processeurs graphiques performants (tel que le Nvidia Tegra). Le problème à résoudre est la communication entre les caméras et le module smartphone car le port MIPI standard ne peut pas faire transiter des volumes de données aussi conséquents que 2 caméras haute résolution et haute cadence (60hz).

Une autre piste à explorer est la réalisation d'une électronique dédiée ; base FPGA par exemple. Ceci réduirait le retard d'affichage à un minimum, mais lève la question de l'implémentation de codes complexes pour la Réalité Augmentée dans un système de ce type.

## 7. REFERENCES

- [1] <https://developer.oculus.com/documentation/intro-vr/latest/>
- [2] [https://developer.oculus.com/documentation/intro-vr/latest/concepts/bp\\_intro/](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). Ophthalmogy 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/HDRYeyetap\\_IEEEccece2012paper551.pdf](http://www.eyetap.org/papers/docs/HDRYeyetap_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.