



# Conception et développement de composants logiciels et matériels pour un dispositif ophtalmique

Jessica Combier

## ► To cite this version:

Jessica Combier. Conception et développement de composants logiciels et matériels pour un dispositif ophtalmique. Optique [physics.optics]. Université Paul Sabatier - Toulouse III, 2019. Français. NNT : 2019TOU30014 . tel-03653785

**HAL Id: tel-03653785**

**<https://theses.hal.science/tel-03653785>**

Submitted on 28 Apr 2022

**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.

Université Fédérale



Toulouse Midi-Pyrénées

# THÈSE

En vue de l'obtention du

## DOCTORAT DE L'UNIVERSITÉ FÉDÉRALE TOULOUSE MIDI-PYRÉNÉES

Délivré par :

*l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

---

---

Présentée et soutenue le 23/01/2019 par :

**JESSICA COMBIER**

**Conception et Développement de Composants Logiciels et Matériels  
pour un Dispositif Ophtalmique**

---

---

### JURY

|                   |                         |
|-------------------|-------------------------|
| FRANÇOIS BERRY    | Professeur d'Université |
| ALBERTO IZAGUIRRE | Professeur d'Université |
| ALTUNA            |                         |
| AUDE BOUCHIER     | Docteur et Ingénieur    |
| OLIVIER MOREL     | Maître de Conférences   |
| PATRICK DANÈS     | Professeur d'Université |
| BERTRAND          | Maître de Conférences   |
| VANDEPORTAELE     |                         |

---

**École doctorale et spécialité :**

*EDSYS : Informatique 4200018*

**Double mention :**

*EDSYS : Robotique 4200046*

**Unité de Recherche :**

*Laboratoire d'analyse et d'architecture des systèmes*

**Directeur(s) de Thèse :**

*Bertrand VANDEPORTAELE et Patrick DANÈS*

**Rapporteurs :**

*François BERRY et Alberto IZAGUIRRE ALTUNA*



## Remerciements

Je souhaite tout d'abord remercier mes deux directeurs de thèse Bertrand Vandeportaele et Patrick Danès, pour leur aide et leur soutien.

Je remercie également pour leur accueil et leur conseils l'ensemble des membres de l'équipe de R&D de Essilor du site de Labège et les autres employés de Essilor avec qui j'ai pu être en interaction au cours de ma thèse. Je remercie en particulier Aude Bouchier (mon encadrante entreprise) pour son aide et son soutien.

Je souhaite également remercier Henri Camon (directeur du laboratoire commun OPERA), qui m'a apporté son soutien et son aide en particulier sur les aspects administratifs.

Je tiens à remercier les équipes RAP et GEPETTO du LAAS-CNRS pour ces trois années passées à leur côtés, qui furent fort enrichissantes.

Enfin, je souhaite un grand merci à mon conjoint Kévin, ainsi qu'à ma famille et mes amis, qui m'ont soutenu tout au long de la thèse.



## Résumé

Les recherches menées au cours de cette thèse de Doctorat s'inscrivent dans les activités du laboratoire commun OPERA (OPTique EmbaRquée Active) impliquant ESSILOR-LUXOTTICA et le CNRS. L'objectif est de contribuer au développement des "lunettes du futur" intégrant des fonctions d'obscurcissement, de focalisation ou d'affichage qui s'adaptent en permanence à la scène et au regard de l'utilisateur. Ces nouveaux dispositifs devront être dotés de capacités de perception, de décision et d'action, et devront respecter des contraintes d'encombrement, de poids, de consommation énergétique et de temps de traitement. Ils présentent par conséquent des connexions évidentes avec la robotique.

Dans ce contexte, les recherches ont consisté à investiguer la structure et la construction de tels systèmes afin d'identifier leurs enjeux et difficultés. Pour ce faire, la première tâche a été de mettre en place des émulateurs de divers types de lunettes actives, qui permettent de prototyper et d'évaluer efficacement diverses fonctions. Dans cette phase de prototypage et de test, ces émulateurs s'appuient naturellement sur une architecture logicielle modulaire typique de la robotique.

La seconde partie de la thèse s'est focalisée sur le prototypage d'un composant clé des lunettes du futur, qui implique une contrainte supplémentaire de basse consommation : le système de suivi du regard, aussi appelé oculomètre. Le principe d'un assemblage de photodiodes et d'un traitement par réseau de neurones a été proposé. Un simulateur a été mis au point, ainsi qu'une étude de l'influence de l'agencement des photodiodes et de l'hyper-paramétrisation du réseau sur les performances de l'oculomètre.

**Mots-clés :** Lunettes intelligentes, robotique, réalité augmentée, oculométrie (gaze tracking), architecture modulaire.



## Abstract

The research carried out during this doctoral thesis takes place within the OPERA joint laboratory (OPTique EmbaRquée Active) involving ESSILORELUXOTTICA and the CNRS. The aim is to contribute to the development of "glasses of the future", which feature obscuration, focus or display capabilities that continuously adapt to the scene and the user gaze. These new devices will be endowed with perception, decision and action capabilities, and will have to respect constraints of space, weight, energy consumption and processing time. They therefore show obvious connections with robotics.

In this context, the structure and building of such systems has been investigated in order to identify their issues and difficulties. To that end, the first task was to set up emulators of various types of active glasses, which enable the prototyping and effective testing of various functions. In this prototyping and testing phase, these emulators naturally rely on a modular software architecture typical of robotics.

The second part of the thesis focused on the prototyping of a key component which implies an additional constraint on low consumption, namely the eye tracking system, also known as gaze tracker. The principle of a photodiode assembly and of a neural network processing has been proposed. A simulator has been developed, as well as a study of the influence of the arrangement of photodiodes and the hyper-parametrization of the network on the performance of the oculometer.

**Key-words :** Smart glasses, robotics, augmented reality, gaze tracking, modular architecture.



# Table des matières

|                                                                                          |           |
|------------------------------------------------------------------------------------------|-----------|
| <b>Introduction</b>                                                                      | <b>1</b>  |
| <b>1 Contexte de l'étude : la vision instrumentée</b>                                    | <b>7</b>  |
| 1.1 Le système visuel humain . . . . .                                                   | 8         |
| 1.1.1 Composition et fonctionnement de l'œil . . . . .                                   | 8         |
| 1.1.2 Rappels d'optique pour la modélisation de l'œil . . . . .                          | 9         |
| 1.1.2.1 Points principaux et points nodaux . . . . .                                     | 10        |
| 1.1.2.2 Modèles optiques de l'œil . . . . .                                              | 12        |
| 1.1.3 La rétine et le rôle des photorécepteurs . . . . .                                 | 15        |
| 1.1.4 Les mouvements de l'œil . . . . .                                                  | 15        |
| 1.1.5 La perception 3D . . . . .                                                         | 17        |
| 1.2 Les lunettes du futur . . . . .                                                      | 18        |
| 1.2.1 Dispositif ophtalmique et nouveaux usages . . . . .                                | 18        |
| 1.2.2 Définition de la réalité augmentée et de la réalité virtuelle . .                  | 19        |
| 1.2.3 Technologies, Contraintes, Opportunités . . . . .                                  | 21        |
| 1.2.3.1 Applications potentielles et complexité . . . . .                                | 21        |
| 1.2.3.2 Technologies . . . . .                                                           | 22        |
| 1.2.3.3 Contraintes et opportunités . . . . .                                            | 24        |
| A Contraintes d'embarquabilité . . . . .                                                 | 24        |
| B Contraintes pour le confort visuel . . . . .                                           | 24        |
| C Opportunités . . . . .                                                                 | 25        |
| 1.3 Un système robotique embarqué . . . . .                                              | 25        |
| 1.3.1 Des scénarios types aux fonctions embarquées . . . . .                             | 25        |
| 1.3.2 Architectures logicielles robotiques pour le prototypage de<br>fonctions . . . . . | 26        |
| 1.3.2.1 Architecture modulaire . . . . .                                                 | 26        |
| 1.3.2.2 Interfaces de communication . . . . .                                            | 28        |
| 1.3.3 Résumé - Vue d'ensemble . . . . .                                                  | 29        |
| 1.4 Objectifs de la thèse . . . . .                                                      | 31        |
| <b>2 Composants matériels et logiciels</b>                                               | <b>33</b> |
| 2.1 Affichage de contenu . . . . .                                                       | 34        |
| 2.1.1 Casques de réalité augmentée . . . . .                                             | 35        |
| A Les VST-HMDs . . . . .                                                                 | 35        |
| B Les OST-HMDs . . . . .                                                                 | 35        |
| 2.1.2 Limitations des dispositifs de réalité augmentée . . . . .                         | 36        |
| 2.1.2.1 Limitations des affichages sur des HMDs . . . . .                                | 36        |
| A Résolution . . . . .                                                                   | 36        |
| B Champ de vue . . . . .                                                                 | 36        |
| C Conflit convergence-accommodation . . . . .                                            | 37        |

|         |         |                                                                       |    |
|---------|---------|-----------------------------------------------------------------------|----|
|         | D       | Luminosité et Contraste . . . . .                                     | 37 |
|         | E       | Différence de point de vue . . . . .                                  | 37 |
| 2.1.2.2 |         | Contraintes inhérentes à la réalité augmentée . . . .                 | 38 |
|         | A       | Temps réel, latence et fréquence d’affichage                          | 38 |
|         | B       | Estimation de la pose du dispositif d’affichage                       | 38 |
| 2.1.2.3 |         | Contraintes d’embarquabilité d’un HMD . . . . .                       | 39 |
| 2.1.3   |         | Conclusion . . . . .                                                  | 39 |
| 2.2     |         | Perception de l’environnement . . . . .                               | 39 |
| 2.2.1   |         | Modélisation des caméras . . . . .                                    | 40 |
|         | 2.2.1.1 | Modèle trou d’épingle . . . . .                                       | 40 |
|         | 2.2.1.2 | Modèle trou d’épingle avec distorsion : “Plump bob”                   | 40 |
|         | 2.2.1.3 | Étalonnage . . . . .                                                  | 42 |
|         | 2.2.1.4 | Modèle pour caméra à large champ de vue . . . . .                     | 43 |
|         | A       | Modèle fisheye . . . . .                                              | 43 |
|         | B       | Modèle génériques pour caméras omnidi-<br>rectionnelles . . . . .     | 44 |
|         | 2.2.1.5 | Conclusion . . . . .                                                  | 45 |
| 2.2.2   |         | Perception 3D . . . . .                                               | 45 |
|         | 2.2.2.1 | Capteurs passifs et actifs . . . . .                                  | 45 |
|         | A       | Les caractéristiques des systèmes actifs . .                          | 46 |
|         | B       | Les caractéristiques des systèmes passifs . .                         | 46 |
|         | 2.2.2.2 | Perception 3D par stéréovision . . . . .                              | 47 |
|         | A       | Principe . . . . .                                                    | 47 |
|         | B       | Calcul de cartes de disparité . . . . .                               | 49 |
|         | C       | Stéréovision sur une séquence d’images . .                            | 50 |
|         | 2.2.2.3 | Autres techniques basées sur des capteurs passifs . .                 | 52 |
|         | A       | Caméras plénoptiques . . . . .                                        | 52 |
|         | B       | Perception 3D basée sur des techniques<br>d’apprentissage . . . . .   | 53 |
|         | 2.2.2.4 | Conclusion . . . . .                                                  | 55 |
| 2.3     |         | Localisation des lunettes dans la scène . . . . .                     | 55 |
|         | 2.3.1   | Localisation dans un environnement maîtrisé . . . . .                 | 56 |
|         |         | 2.3.1.1 Systèmes sans caméra embarquée . . . . .                      | 56 |
|         |         | 2.3.1.2 Systèmes avec caméra(s) embarquée(s) . . . . .                | 56 |
|         | A       | Systèmes basés marqueur . . . . .                                     | 57 |
|         | B       | Systèmes basés modèle 3D . . . . .                                    | 57 |
|         | 2.3.2   | Localisation dans un environnement inconnu . . . . .                  | 58 |
|         |         | 2.3.2.1 SLAM basé primitives géométriques . . . . .                   | 59 |
|         | A       | Méthodes basées filtrage . . . . .                                    | 60 |
|         | B       | Méthodes basées ajustement de faisceaux . .                           | 60 |
|         |         | 2.3.2.2 SLAM direct . . . . .                                         | 61 |
|         | 2.3.3   | Conclusion . . . . .                                                  | 62 |
| 2.4     |         | Détection et suivi du regard du porteur . . . . .                     | 62 |
|         | 2.4.1   | Techniques d’oculométrie (suivi du regard, <i>eye-gaze tracking</i> ) | 62 |

|         |                                                                                       |    |
|---------|---------------------------------------------------------------------------------------|----|
| 2.4.2   | Techniques basées caméra et extraction d'éléments caractéristiques . . . . .          | 63 |
| 2.4.2.1 | Éléments caractéristiques détectés . . . . .                                          | 64 |
| 2.4.2.2 | Méthodes basées interpolations . . . . .                                              | 65 |
| A       | Les PCCR et autres techniques s'appuyant sur les reflets . . . . .                    | 65 |
| B       | L'usage de la pupille ou de l'iris seul . . . . .                                     | 66 |
| C       | Les limites de ces méthodes . . . . .                                                 | 66 |
| 2.4.2.3 | Méthodes basées modèle 3D . . . . .                                                   | 66 |
| 2.4.3   | Oculomètre basse consommation . . . . .                                               | 68 |
| 2.4.4   | Conclusion . . . . .                                                                  | 70 |
| 2.5     | Rendu synthétique de l'environnement et affichage d'information augmentée . . . . .   | 70 |
| 2.5.1   | Deux procédés de synthèse d'image . . . . .                                           | 70 |
| 2.5.1.1 | Incrustation d'objets synthétiques . . . . .                                          | 70 |
| 2.5.1.2 | Affichage de contenu synthétique construit à partir de la scène . . . . .             | 71 |
| 2.5.2   | Rendu d'une image par DIBR pour un autre point de vue . . . . .                       | 73 |
| 2.5.2.1 | Synthèse d'images intermédiaires . . . . .                                            | 74 |
| A       | "Direct 3D warping" . . . . .                                                         | 74 |
| B       | Maillage triangulaire . . . . .                                                       | 74 |
| C       | Inverse 3D warping . . . . .                                                          | 76 |
| D       | Fusion des images cibles intermédiaires . . . . .                                     | 77 |
| 2.5.2.2 | Remplissage de zones vides . . . . .                                                  | 78 |
| A       | Remplissage des cracks . . . . .                                                      | 79 |
| B       | Remplissage des trous dus aux occlusions . . . . .                                    | 79 |
| 2.5.2.3 | Génération de nouvelles vues par des techniques d'apprentissage automatique . . . . . | 80 |
| 2.5.2.4 | Limite du DIBR . . . . .                                                              | 81 |
| 2.5.3   | Conclusion . . . . .                                                                  | 82 |
| 2.6     | Étalonnage du dispositif . . . . .                                                    | 82 |
| 2.6.1   | Modèle de projection . . . . .                                                        | 83 |
| 2.6.2   | Étalonnage d'un OST-HMD . . . . .                                                     | 84 |
| 2.6.2.1 | Méthodes basées sur l'alignement de points . . . . .                                  | 84 |
| 2.6.2.2 | Méthodes basées caméras . . . . .                                                     | 85 |
| 2.6.2.3 | Méthodes avec adaptation dynamique au regard du porteur . . . . .                     | 87 |
| 2.6.3   | Étalonnage d'un VST-HMD . . . . .                                                     | 87 |
| 2.6.4   | Conclusion . . . . .                                                                  | 88 |
| 2.7     | Conclusion . . . . .                                                                  | 89 |

|          |                                                                                        |            |
|----------|----------------------------------------------------------------------------------------|------------|
| <b>3</b> | <b>Proposition d'un oculomètre basse consommation</b>                                  | <b>91</b>  |
| 3.1      | Prototypage virtuel d'un oculomètre basse consommation . . . . .                       | 92         |
| 3.1.1    | Exigences pour l'optique embarquée . . . . .                                           | 92         |
| 3.1.2    | Une solution basée photodiodes et apprentissage automatique                            | 93         |
| 3.1.2.1  | Sélection d'un assemblage de photodiodes . . . . .                                     | 93         |
| 3.1.2.2  | Synthèse d'un algorithme pour estimer la direction<br>du regard . . . . .              | 94         |
| 3.1.3    | Simulation de l'oculomètre dans un environnement synthétique                           | 95         |
| 3.1.3.1  | Synthèse d'images du visage et de l'œil . . . . .                                      | 96         |
| 3.1.3.2  | Simulation de l'assemblage de photodiodes . . . . .                                    | 97         |
| A        | Assemblage de photodiodes . . . . .                                                    | 97         |
| B        | Photodiode . . . . .                                                                   | 98         |
| 3.2      | Tests sur données synthétiques . . . . .                                               | 101        |
| 3.2.1    | Définition des hyperparamètres du réseau . . . . .                                     | 102        |
| 3.2.1.1  | Méthodes pour l'évaluation et la comparaison de ré-<br>seaux de neurones . . . . .     | 104        |
| 3.2.1.2  | Prétraitement des données d'entrée . . . . .                                           | 105        |
| 3.2.1.3  | Initialisation des poids et fonction d'activation . . .                                | 108        |
| 3.2.1.4  | Fonction de perte . . . . .                                                            | 109        |
| 3.2.1.5  | Définition d'une architecture de réseau de neurones                                    | 112        |
| 3.2.1.6  | Dimensionnement du nombre d'époques et de la<br>taille de la base de données . . . . . | 114        |
| 3.2.2    | Premiers résultats et analyse . . . . .                                                | 115        |
| 3.2.2.1  | Estimation de la direction du regard . . . . .                                         | 116        |
| 3.2.2.2  | Estimation de la pose de l'œil . . . . .                                               | 118        |
| 3.2.3    | Évaluation de différents assemblages de photodiodes . . . . .                          | 120        |
| 3.2.3.1  | Variation de l'angle d'ouverture des photodiodes . .                                   | 120        |
| 3.2.3.2  | Variation du nombre de photodiodes . . . . .                                           | 122        |
| 3.2.3.3  | Variation de l'orientation des photodiodes . . . . .                                   | 122        |
| 3.2.3.4  | Variation de l'organisation spatiale des photodiodes                                   | 122        |
| 3.2.3.5  | Test avec filtre . . . . .                                                             | 127        |
| 3.2.3.6  | Conclusion sur l'assemblage à considérer . . . . .                                     | 127        |
| 3.3      | Tests sur données réelles . . . . .                                                    | 129        |
| 3.3.1    | Banc de test et procédure d'acquisition de données . . . . .                           | 129        |
| 3.3.2    | Oculomètre basé caméras pour le banc de test . . . . .                                 | 131        |
| 3.3.3    | Simulation des photodiodes à partir d'images réelles . . . . .                         | 132        |
| 3.4      | Conclusion . . . . .                                                                   | 132        |
| 3.5      | Perspectives . . . . .                                                                 | 133        |
| <b>4</b> | <b>Prototypage et évaluation</b>                                                       | <b>135</b> |
| 4.1      | Banc de test basé sur un VST-HMD . . . . .                                             | 136        |
| 4.1.1    | Description générale . . . . .                                                         | 136        |
| 4.1.1.1  | Composants logiciels et matériels . . . . .                                            | 136        |

|         |   |                                                                                                                  |     |
|---------|---|------------------------------------------------------------------------------------------------------------------|-----|
|         | A | Système d’affichage et banc stéréoscopique<br>large champ de vision . . . . .                                    | 136 |
|         | B | Estimation de la direction du regard . . . .                                                                     | 138 |
|         | C | Localisation des lunettes dans la scène . . .                                                                    | 138 |
|         | D | Perception 3D de la scène . . . . .                                                                              | 138 |
| 4.1.1.2 |   | Architecture logicielle . . . . .                                                                                | 139 |
| 4.1.2   |   | Synthèse d’images par DIBR pour le point de vue de l’utilisateur                                                 | 142 |
| 4.1.2.1 |   | Calcul de la carte de profondeur . . . . .                                                                       | 142 |
| 4.1.2.2 |   | Étape de "3D warping" . . . . .                                                                                  | 143 |
| 4.1.2.3 |   | Remplissage des zones vides . . . . .                                                                            | 144 |
| 4.1.3   |   | Projection de contenu augmenté ou altéré . . . . .                                                               | 145 |
| 4.1.4   |   | Suivi du regard . . . . .                                                                                        | 146 |
| 4.1.4.1 |   | Implémentation d’un algorithme basé interpolation                                                                | 148 |
| 4.1.4.2 |   | Implémentation d’un algorithme basé modèle 3D . .                                                                | 150 |
| 4.1.5   |   | Premiers tests, résultats et performances . . . . .                                                              | 153 |
| 4.1.5.1 |   | Test du VST-HMD par une application . . . . .                                                                    | 153 |
| 4.1.5.2 |   | Performance temporelle actuelle sur CPU . . . . .                                                                | 153 |
| 4.1.5.3 |   | Test et analyse de la perception 3D . . . . .                                                                    | 155 |
| 4.1.6   |   | Perspectives . . . . .                                                                                           | 157 |
|         | A | Déport des traitements d’images sur GPU                                                                          | 157 |
|         | B | Ajout d’une centrale inertielle pour améliorer<br>la localisation du SLAM et augmenter<br>la fréquence . . . . . | 158 |
|         | C | Réduction de la latence par la synthèse<br>d’image prédictive . . . . .                                          | 158 |
|         | D | Amélioration de la qualité des images syn-<br>thétisées par DIBR . . . . .                                       | 158 |
|         | E | Adaptation des algorithmes au modèle de<br>caméra omnidirectionnelle . . . . .                                   | 160 |
|         | F | Ré-étalonnage automatique et régulier du<br>banc de stéréovision . . . . .                                       | 160 |
|         | G | Amélioration de l’oculomètre basé modèle<br>3D . . . . .                                                         | 160 |
| 4.1.7   |   | Conclusion . . . . .                                                                                             | 161 |
| 4.2     |   | Banc de test basé sur un OST-HMD . . . . .                                                                       | 161 |
| 4.2.1   |   | Dispositif et composants logiciels . . . . .                                                                     | 162 |
| 4.2.1.1 |   | Choix des composants matériels . . . . .                                                                         | 162 |
| 4.2.1.2 |   | Architecture logicielle . . . . .                                                                                | 164 |
| 4.2.2   |   | Tests et résultats d’étalonnage d’OST-HMD . . . . .                                                              | 165 |
| 4.2.2.1 |   | Méthode d’étalonnage manuelle . . . . .                                                                          | 166 |
| 4.2.2.2 |   | Single Point Active Alignment Method . . . . .                                                                   | 166 |
| 4.2.2.3 |   | Single Point Active Alignment Method et oculométrie                                                              | 167 |
| 4.2.2.4 |   | Évaluations et résultats . . . . .                                                                               | 167 |
| 4.2.3   |   | Limites du système actuel et perspectives . . . . .                                                              | 168 |

|                                                  |                                                                     |            |
|--------------------------------------------------|---------------------------------------------------------------------|------------|
| 4.2.3.1                                          | Oculométrie . . . . .                                               | 168        |
| 4.2.3.2                                          | Étalonnage . . . . .                                                | 169        |
| 4.2.3.3                                          | Applications sur le banc de test . . . . .                          | 169        |
| 4.3                                              | Prototype de lunettes avec obscurcissement automatique des verres . | 170        |
| 4.3.1                                            | Choix des capteurs de luminosité . . . . .                          | 170        |
| 4.3.2                                            | Choix d'une commande du verre en fonction de la luminosité          | 172        |
| 4.3.3                                            | Test sur simulateur et mise en place du prototype . . . . .         | 173        |
| 4.3.4                                            | Perspectives . . . . .                                              | 175        |
| 4.4                                              | Conclusion . . . . .                                                | 178        |
| <b>Conclusion</b>                                |                                                                     | <b>181</b> |
| <b>A Bref historique de la réalité augmentée</b> |                                                                     | <b>185</b> |
| A.1                                              | Avant 2015 . . . . .                                                | 185        |
| A.2                                              | Depuis 2015 . . . . .                                               | 186        |
| A.3                                              | Positionnement d' Essilor . . . . .                                 | 186        |
| <b>Bibliographie</b>                             |                                                                     | <b>191</b> |

# Introduction

## Contexte général

### La lunette active

Une paire de lunettes est un dispositif courant, dont les propriétés (correction ou degré d'obscurcissement) n'évoluent généralement pas au fil du temps. Les lunettes dotées de verres autochromes sont néanmoins un contre-exemple : avec ces dispositifs, l'adaptation du degré d'obscurcissement du verre à la luminosité de l'environnement est obtenue par un procédé chimique, laissant peu de marges de manœuvre à un pilotage fin. L'arrivée récente de verres dont l'obscurcissement est pilotable électriquement (verres électrochromes) change la donne en fournissant l'opportunité d'agir sur le verre finement. Néanmoins, elle rend nécessaire l'intégration de capacités de perception et de raisonnement sur la lunette afin de déterminer la commande adéquate, d'autant plus si le verre est pilotable de manière localisée (grâce à l'adressage de différentes cellules électrochromes). L'intégration d'une source énergétique sur les lunettes est par là même rendue nécessaire et peut prendre différentes formes telles que batterie ou système de récupération d'énergie cinétique ou électromagnétique.

Parallèlement à ces considérations, la paire de lunettes ne doit pas gêner la vision et doit couvrir le champ le plus large possible. Devant être portée sur le nez en permanence, son poids et son encombrement doivent être réduits et son apparence doit être attrayante. Dans le cas où les lunettes disposent de capacités dynamiques, la vitesse de réaction du système doit être compatible avec le système visuel humain.

Ces considérations ont servi de point de départ à la thèse et de nombreux autres usages ont par la suite été envisagés en plus de l'obscurcissement piloté, par exemple : correction adaptative (en fonction de la distance visée dans la scène), affichage d'information (en fonction du contexte), redirection de capacités sensorielles (affichage de texte obtenu par reconnaissance vocale pour une personne déficiente auditive), etc.

### Réalité Virtuelle/Augmentée

Les technologies de réalité virtuelle et plus récemment de réalité augmentée sont devenues accessibles au grand public, principalement dans le domaine du divertissement mais également dans différents domaines professionnels. Elles connaissent aujourd'hui une évolution très rapide. Bien que visant potentiellement différents sens (vision, audition, odorat, toucher et même goût), les systèmes les plus répandus visent à l'immersion de l'utilisateur via le sens de la vision, par le biais de casques ou lunettes. Alors qu'un dispositif de réalité virtuelle propose à son porteur l'affichage d'un modèle synthétique de la scène en le privant de sa perception de

Confidentiel

l'environnement réel, un dispositif de réalité augmentée doit permettre l'intégration d'éléments synthétiques dans sa perception de l'environnement réel, ce qui soulève de nombreuses contraintes supplémentaires.

Grâce aux nouvelles capacités d'affichages procurées par la lunette active, cette thèse s'intéresse à la problématique de la réalité augmentée mais en considérant des contraintes propres aux applications visées et en ne tentant pas dans ce domaine de concurrencer les géants de l'industrie du jeu. Au lieu de cela, la thèse considère les lunettes comme un système robotique doté de capacités de perception et d'analyse de l'environnement, de prise de décision et d'action, ce système devant également prendre en compte des considérations d'autonomie énergétique.

## Laboratoire commun OPERA

C'est dans le contexte énoncé précédemment que le laboratoire commun OPERA (OPTique EmbaRquée Active) composé du LAAS-CNRS (Laboratoire d'Analyse et d'Architecture des Systèmes), et de l'entreprise Essilor, leader mondial dans la production de verres et d'équipements ophtalmiques, mène des recherches sur les "lunettes actives", c'est-à-dire des lunettes équipées de "fonctionnalités dynamiques ou actives". Au sein du LAAS-CNRS, différentes équipes impliquées travaillent sur différents aspects (photonique, technologie, électronique...) L'équipe Robotique, Action, Perception (RAP) étudie spécifiquement les lunettes actives en utilisant des approches propres aux systèmes robotiques.

## Objectifs et contributions de la thèse

Les travaux de thèse présentés dans ce manuscrit s'inscrivent dans le cadre des recherches du laboratoire OPERA et ont été réalisés aux sein des équipes RAP du LAAS-CNRS et de Recherche et Développement d'Essilor. L'objectif principal de la thèse est d'investiguer de nouvelles fonctionnalités et de proposer des solutions d'architectures (matérielles et logicielles) adaptées aux lunettes actives. A cette fin, une première partie des travaux a consisté en la réalisation de bancs de prototypage pour démontrer les solutions proposées. Ensuite, un travail plus spécifique sur la fonction de mesure de l'orientation du regard de l'utilisateur a été mené afin de tenir compte des contraintes d'embarquabilité imposées par l'application, notamment en termes de consommation énergétique et d'encombrement. Parallèlement à ces considérations technologiques, la thèse s'intéresse également au système visuel humain via différents aspects :

- en tant qu'objet sur lequel effectuer des mesures par oculométrie, par exemple pour permettre l'interaction par le regard ;
- en tant qu'objet sur lequel agir via un affichage, par exemple pour informer l'utilisateur ou pour obscurcir des sources lumineuses gênantes ;
- en tant qu'objet pathologique qu'il faut assister, dans le cadre de dispositifs ophtalmique, par exemple en adaptant dynamiquement la correction ou en

reprojetant l'information visuelle sur des zones encore sensibles dans le cas de personnes atteintes de Dégénérescence Maculaire Liée à l'Âge.

Pour ces raisons, et parce qu'il intègre de nombreux mécanismes actifs (telles que l'orientation, la focalisation ou l'ouverture), le système visuel humain doit être considéré en tant que composant à part entière du système de perception augmenté.

## Prototypage

Deux bancs de prototypage complémentaires permettant de simuler différents types de verres actifs ont été mis en place afin de tester et de sélectionner différents algorithmes, et d'étudier les limites inhérentes aux différents systèmes. Pour simplifier les phases de tests et de prototypage, les bancs de test sont basés sur une architecture modulaire, dans laquelle chaque composant peut être développé, évalué indépendamment, et éventuellement interchangé avec un ou plusieurs autres composants. Certains des composants étudiés dans cette thèse sont par exemple dévolus :

- à la perception et à l'analyse de la scène, notamment afin de la modéliser et de permettre la localisation du système ;
- au suivi du regard du porteur ;
- à l'étalonnage géométrique des différentes parties du système ;
- à la génération d'images mixant la scène réelle et des éléments synthétiques.

Le premier banc de prototypage a été construit sur la base d'un casque de réalité augmentée de type VST-HMD (Video See-through Head Mounted Display), avec lequel l'utilisateur perçoit la scène au travers d'écrans. Ce banc a permis de valider un concept original de génération d'images permettant de réduire l'inconfort de l'utilisateur. Ce concept est basé sur la génération de nouvelles images pour les points de vue des yeux de l'utilisateur à partir d'images acquises par des caméras embarquées en tenant compte de différentes contraintes. Ce banc a notamment permis l'évaluation de l'apport de la solution proposée sur la perception 3D de la scène par l'utilisateur.

Le second banc de test a quant à lui été développé sur la base d'un casque de réalité augmentée de type OST-HMD (Optical See-Through Head Mounted Display), avec lequel la scène est perçue directement par l'utilisateur. Ce banc a notamment permis d'étudier les problématiques inhérentes à ce genre de dispositif liées à l'étalonnage géométrique du système.

## Mesure de l'orientation du regard

En parallèle à l'étude à large spectre menée sur le système de lunettes, la thèse se focalise sur le développement d'un des modules : le système de mesure de l'orientation du regard, aussi appelé oculomètre. L'approche proposée vise à réduire la consommation énergétique. Elle consiste à remplacer les caméras, dont l'acquisition

et le traitement des images sont coûteux en énergie, par un assemblage de photodiodes. Le traitement des mesures fournies par les photodiodes est quant à lui dévolu à un réseau de neurones.

Afin de définir des assemblages de photodiodes adéquats, un simulateur a été mis en place, dans lequel différents assemblages peuvent être évalués comparativement en faisant varier le nombre, la position, l'orientation et l'ouverture des photodiodes. Après de nombreux tests sur données synthétiques, des tendances ont pu être dégagées, ce qui a permis de définir des assemblages performants adaptés aux besoins. Ceux-ci feront l'objet d'une étude sur des données réelles à l'aide d'un nouveau banc de test (en cours de développement) après la fin de la thèse.

## Organisation du mémoire

En premier lieu, le chapitre 1 positionne le contexte de la thèse et introduit quelques notions de base. Avant de décrire notre étude, le manuscrit fournit une courte bibliographie sur le système visuel humain. Cette bibliographie a été effectuée au début de la thèse afin d'identifier sur des lunettes actives ce qui est susceptible d'entraver la vision du porteur. Ensuite, le chapitre introduit la notion de "lunettes du futur" en décrivant les applications possibles, les technologies utilisées (avec une définition de la réalité augmentée et de la réalité virtuelle) et les contraintes inhérentes à ces systèmes. Puis, un parallèle est proposé entre ces nouvelles lunettes et les systèmes robotiques. Les notions d'architecture modulaire et de prototypage y sont introduites. La lunette est alors décrite comme un système modulaire équipé de fonctions de perception, d'analyse, de décision et d'action. Ce chapitre se clôture sur le récapitulatif des objectifs de la thèse.

Le deuxième chapitre regroupe l'état de l'art effectué au cours de cette étude et se divise en plusieurs sections, chacune consacrée à la présentation d'un des modules des lunettes actives. La première section porte sur un composant matériel pour l'affichage (réalité augmentée) et détaille les contraintes qui lui sont liées. Les sections suivantes traitent des modules que nous avons jugés essentiels et nécessaires aux applications envisagées : la perception de l'environnement, la localisation des lunettes dans la scène, la détection et le suivi du regard du porteur, la synthèse d'image et l'étalonnage du système d'affichage des lunettes. Les deux derniers modules s'appliquent spécifiquement aux lunettes de réalité augmentée mais pourraient être étendus à tous types de lunettes actives dont les propriétés (par exemple focale ou obscurcissement) sont définies individuellement par zone sur le champ de vision du porteur.

Le troisième chapitre est entièrement consacré au module de détection et de suivi du regard du porteur, pour lequel nous proposons un nouveau concept d'oculomètre basse consommation. La première section du chapitre énonce les contraintes et les enjeux liés à cet oculomètre, présente notre solution basée sur l'utilisation de photodiodes et de méthodes d'apprentissage, et décrit le simulateur mis en place pour identifier des configurations d'assemblage de photodiodes pertinentes. La sec-

tion suivante est consacrée à la définition du réseau de neurones, à la visualisation et à l'analyse des premiers résultats de simulation et à la comparaison de plusieurs propositions d'assemblage de photodiodes. Puis, la dernière section ouvre sur une suite à cette étude, qui portera sur l'évaluation de ce concept d'oculomètre sur données réelles.

Le dernier chapitre présente les bancs de prototypage mis en place pour simuler et tester l'intégration de nouvelles fonctionnalités sur lunettes actives afin d'éviter la fabrication de nouvelles lunettes pour tester chaque nouveau concept. Les deux premières sections sont consacrées respectivement aux bancs de test basés sur un VST-HMD et sur un OST-HMD. Pour chaque banc, une description de l'architecture matérielle et logicielle est proposée, des premiers tests et résultats sont détaillés, puis leurs limites et les améliorations possibles sont exposées. La dernière section du chapitre présente un concept de lunette active proposé par l'entreprise. Après l'avoir simulé et testé sur le banc de test basé VST-HMD, un prototype a été conçu pour vérifier le concept et évaluer ses limites.

Confidentiel

# Contexte de l'étude : la vision instrumentée

## Sommaire

|            |                                                                       |           |
|------------|-----------------------------------------------------------------------|-----------|
| <b>1.1</b> | <b>Le système visuel humain</b>                                       | <b>8</b>  |
| 1.1.1      | Composition et fonctionnement de l'œil                                | 8         |
| 1.1.2      | Rappels d'optique pour la modélisation de l'œil                       | 9         |
| 1.1.3      | La rétine et le rôle des photorécepteurs                              | 15        |
| 1.1.4      | Les mouvements de l'œil                                               | 15        |
| 1.1.5      | La perception 3D                                                      | 17        |
| <b>1.2</b> | <b>Les lunettes du futur</b>                                          | <b>18</b> |
| 1.2.1      | Dispositif ophtalmique et nouveaux usages                             | 18        |
| 1.2.2      | Définition de la réalité augmentée et de la réalité virtuelle         | 19        |
| 1.2.3      | Technologies, Contraintes, Opportunités                               | 21        |
| <b>1.3</b> | <b>Un système robotique embarqué</b>                                  | <b>25</b> |
| 1.3.1      | Des scénarios types aux fonctions embarquées                          | 25        |
| 1.3.2      | Architectures logicielles robotiques pour le prototypage de fonctions | 26        |
| 1.3.3      | Résumé - Vue d'ensemble                                               | 29        |
| <b>1.4</b> | <b>Objectifs de la thèse</b>                                          | <b>31</b> |

Les systèmes étudiés dans cette thèse ont pour objectif de doter des lunettes de fonctionnalités avancées telles que l'assombrissement des verres ou l'affichage de contenu synthétique dans le champ de vision du porteur, en s'adaptant à son besoin et à la nature de la scène. Par conséquent, ces dispositifs doivent tenir compte du fonctionnement du système visuel de l'utilisateur, pour ne pas altérer la vision du porteur et pour connaître les objectifs à atteindre. Par exemple, un dispositif, générant un affichage synthétique dans le champ de vision du sujet, doit dans l'idéal atteindre une résolution angulaire équivalente à celle du système visuel. Il n'est pas nécessaire d'aboutir à une plus grande résolution. La première section se propose de décrire les caractéristiques de la vision humaine. La deuxième section présente la lunette instrumentée, avec ses composants logiciels et matériels, ses applications et les contraintes auxquelles elle doit répondre. Ensuite, le sujet est abordé sous un autre angle en comparant la lunette instrumentée à un système robotique équipé de capteurs et d'actionneurs. Dans ce contexte, nous proposons d'utiliser des outils propres à la robotique et de les appliquer au problème étudié. Le chapitre est conclu par la présentation du sujet de la thèse et de ses objectifs.

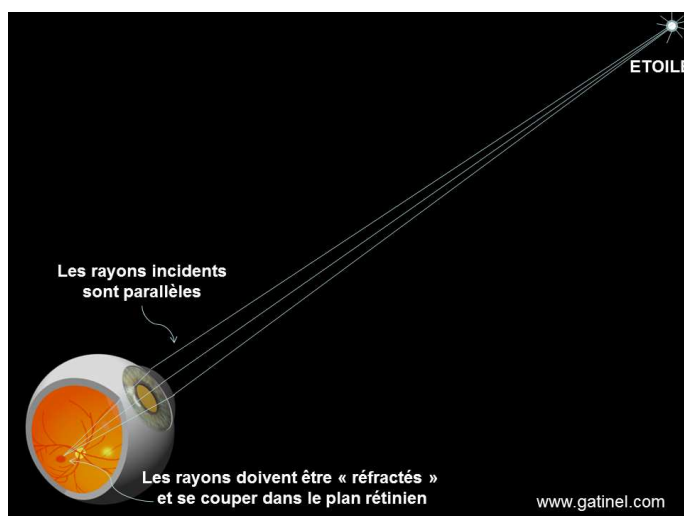


FIGURE 1.1 – Image d'un point à l'infini sur la rétine. (Extrait de [Gatinel 2013b])

## 1.1 Le système visuel humain

La courte bibliographie du système visuel humain présentée dans cette section est basée sur les cours et livres suivants [Gatinel 2013a], [Tasman 2006], [Missal 2018], [Dragoi 2018], [Schubert 2006].

### 1.1.1 Composition et fonctionnement de l'œil

Le fonctionnement de l'œil est comparable à celui d'une caméra : la lumière collectée par le dispositif optique est focalisée sur une surface, un capteur pour une caméra et la rétine pour un œil (figure 1.1). Tout comme la matrice photosensible de la caméra, la rétine est recouverte de photorécepteurs qui reçoivent l'information lumineuse et la retransmettent au cerveau via le nerf optique. La figure 1.2 montre une coupe horizontale schématique de l'œil, où l'on retrouve la rétine et le nerf optique.

Le cristallin et la cornée entrent en jeu dans la focalisation des rayons lumineux et la construction d'une image nette sur la rétine. Le rôle de ces deux éléments peut être comparé aux assemblages de lentilles dans une caméra. Notamment, à l'instar d'une caméra dont la focale de l'objectif est manuellement modifiable, la focale de l'œil peut varier par le biais de muscles ciliaires modifiant l'épaisseur du cristallin. La cornée, quant à elle, est la première surface rencontrée par la lumière. Sa paroi extérieure agit en première approximation comme un dioptre sphérique. L'importante variation d'indice de réfraction entre l'air et le film lacrymal recouvrant la cornée induit sa surface à se comporter aussi comme un miroir convexe (Toutefois la proportion de rayons réfléchis est plus faible que la proportion de rayons réfractés). Une partie de la lumière incidente est alors réfléchie, ce qui donne lieu à un reflet cornéen. Suivant le même principe, la paroi intérieure se comporte elle aussi comme un dioptre sphérique, générant un second reflet cornéen beaucoup

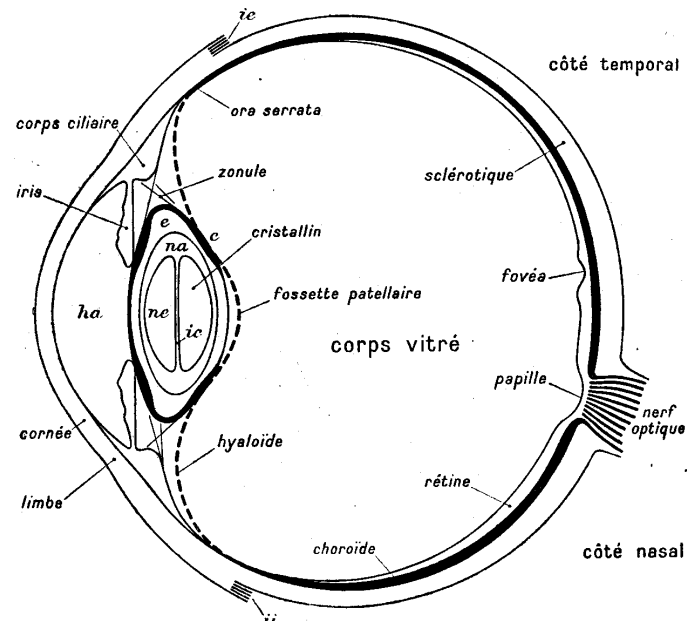


FIGURE 1.2 – Coupe horizontale schématique de l'œil. ha : humeur aqueuse; c : capsule du cristallin (épaisseur très exagérée); e : écorce; na : noyau adulte; ne : noyau embryonnaire; ic : intervalle central; ie : insertion du muscle droit externe; ii : insertion du muscle droit interne. (Extrait de [Le Grand 1964])

plus faible. Cependant, en pratique, la courbure de la cornée n'est pas constante sur toute sa périphérie. On observe une toricité locale, qui peut être modélisée par une ellipsoïde torique.

En plus d'un procédé de focalisation, l'œil a besoin de réguler la quantité de lumière projetée sur la rétine. Ce rôle est rempli par l'iris, situé entre la cornée et le cristallin, qui agit donc comme le diaphragme d'une caméra. Ainsi, son ouverture, appelée pupille, admet un diamètre variant entre 1,5 mm et 8 mm. Le processus de contraction-rétraction sous-jacent est induit par deux types de réflexes pupillaires.

- Lors de l'illumination soudaine et brève d'un œil, sa pupille réagit par un rétrécissement de son diamètre (réflexe pupillaire direct). La pupille de l'autre œil réagit de façon identique (réflexe pupillaire consensuel). Cette réaction survient après un temps de latence d'environ 200 ms, d'autant plus court que l'intensité du stimulus lumineux est grande.
- Lorsque les yeux convergent, il se produit une accommodation et un rétrécissement pupillaire, qui permettent d'obtenir une image rétinienne plus nette. La profondeur de champ est augmentée, au sens où l'intervalle de distances dans l'espace objet conduisant à une vision nette est accru.

### 1.1.2 Rappels d'optique pour la modélisation de l'œil

Pour mesurer la direction du regard, il est possible d'observer (sur l'image issue d'une caméra) des reflets cornéens ou des éléments distinctifs tels que la pupille ou

l'iris [Hansen 2010]. Il peut alors être utile de considérer un modèle géométrique de l'œil pour retracer le chemin des rayons lumineux réfractés ou réfléchis à sa surface. Par la suite, ce tracé de rayons permet d'estimer la position et l'orientation de l'œil relativement à la caméra qui l'observe.

Cette sous-section donne un aperçu de quelques modèles importants.

### 1.1.2.1 Points principaux et points nodaux

Pour la description des modèles géométriques de l'œil qui vont suivre, il est nécessaire d'introduire la notion de points principaux et de points nodaux.

Les systèmes optiques complexes constitués de plusieurs lentilles et considérés centrés (c'est-à-dire admettant un axe de révolution dit axe optique) peuvent être modélisés par une boîte noire contenant six points cardinaux, qui déterminent pour un rayon incident donné, un rayon réfracté en sortie (figure 1.3). Les six points sont les suivants : deux points focaux, deux points principaux et deux points nodaux.

Pour rappel, on distingue deux espaces de chaque côté du système optique : l'espace objet, d'où viennent les rayons incidents émis ou réfléchis par un objet réel, et l'espace image, où sortent les rayons réfractés et où se forment les images perçues par l'œil ou un capteur photosensible. Les points focaux, appelés respectivement point focal image  $F'$  et point focal objet  $F$ , sont des points particuliers situés sur l'axe optique, respectivement dans l'espace image ou objet. Tout rayon passant par le point focal objet émergera du système en étant parallèle à l'axe optique (Schéma 1.3(b)). De même tout rayon incident parallèle à l'axe optique passera par le point focal image après réfraction. Les plans focaux objet et image, qui sont les plans perpendiculaires à l'axe optique et passant par les points focaux, suivent le même principe. Tout objet situé dans le plan focal objet a une image située à l'infini de l'autre côté du système. De même, tout objet situé à l'infini, tel qu'une étoile, aura une image dans le plan focal image.

Pour définir les points principaux, on peut considérer le cas des rayons parallèles à l'axe optique et provenant de l'infini. Après avoir traversé le système optique, les rayons émergents correspondants intersectent l'axe optique en  $F'$ . Si l'on trace les lignes définissant ces rayons incidents et émergents, elles s'intersectent toutes dans un même plan perpendiculaire à l'axe optique comme l'illustre le schéma 1.3(b). Ce plan se nomme plan principal image. Similairement, tout rayon émergent parallèle à l'axe optique croise le rayon incident correspondant dans le plan principal objet. Les points principaux objet et image notés  $H$  et  $H'$  sur le schéma sont les intersections entre les plans principaux et l'axe optique.

Les points nodaux définissent eux aussi un trajet particulier d'un rayon (Figure 1.4). En effet, un rayon incident passant par le premier point nodal ressortira du second point nodal avec une direction inchangée. Dans certain cas, des simplifications peuvent être faites. Si l'objet et l'image sont dans un milieu de même indice de réfraction, alors le premier et le second point nodal seront respectivement confondus avec le premier et le second point principal. En revanche, si le système optique se restreint à une lentille mince pour la quelle l'image et l'objet sont dans un même

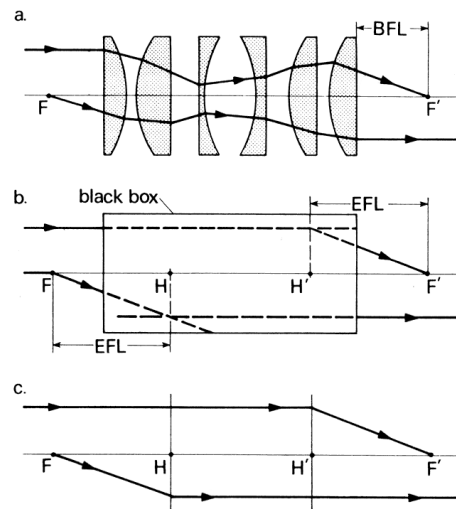


FIGURE 1.3 – Plans et points principaux d'un système optique complexe. (a) Schéma d'un système optique complexe et du tracé de deux rayons, l'un parallèle à l'axe optique et provenant de l'infini en entrée du système, l'autre parallèle à l'axe optique en sortie du système. Ces rayons, après ou avant réfraction, croisent l'axe optique en  $F'$  et  $F$ , les points focaux image et objet.  $BFL$  désigne la distance focale arrière (back focal length – BFL), qui représente la distance entre la paroi de sortie du système optique et le point focal image. (b) Le système optique est remplacé par une boîte noire dans laquelle sont définis deux plans principaux situés à l'intersection du rayon émergent et du rayon incident. On définit les points principaux  $H'$  et  $H$  comme les intersections entre l'axe optique et les plans principaux image et objet.  $EFL$  désigne la distance focale effective (effective focal length – EFL), qui correspond à la distance séparant le point focal image (respectivement objet) du point principal image (respectivement objet). (c) Le système optique est finalement modélisé par les points focaux  $F$  et  $F'$  et deux plans principaux orthogonaux à l'axe optique aux points  $H$  et  $H'$  (Extrait de [Tasman 2006]).

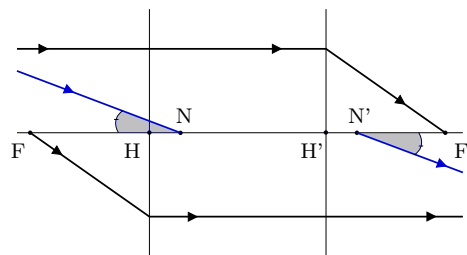


FIGURE 1.4 – Schéma d'un système optique modélisé par les points focaux image et objet  $F'$  et  $F$ , les points principaux image et objet  $H'$  et  $H$  et les points nodaux image et objet  $N'$  et  $N$ . Tout rayon incident (en bleu sur le schéma) passant par le point nodal objet  $N$  ressort du système par le point nodal image  $N'$  en une direction inchangée.

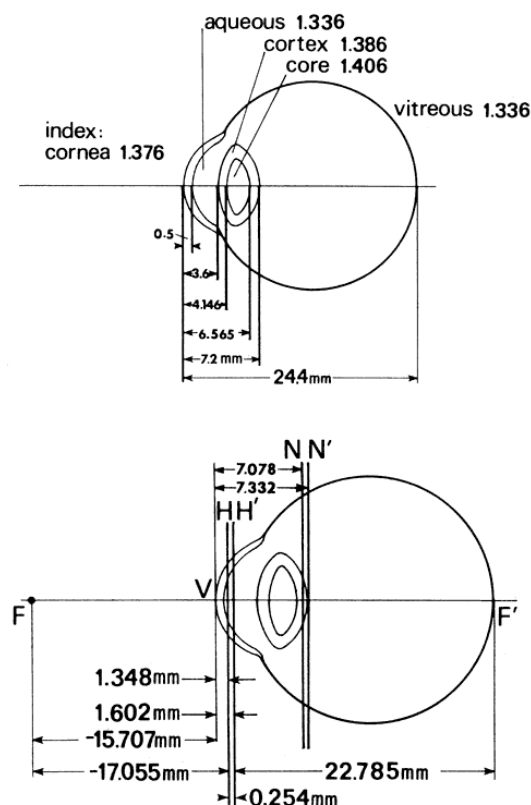


FIGURE 1.5 – Modèle théorique de Gullstrand. (Haut) Indices et positions des surfaces réfractantes. (Bas) Positions des points cardinaux et des plans définissant les lentilles théoriques simulant le système optique complexe de l'œil (Extrait de [Tasman 2006]).

milieu, alors les points nodaux et principaux sont tous confondus avec le centre optique (point par lequel les rayons incidents ne sont pas déviés).

### 1.1.2.2 Modèles optiques de l'œil

Le modèle le plus connu de l'œil est celui proposé par Gullstrand en 1909 [von Helmholtz 1909], qui considère un œil sans anomalie et au repos<sup>1</sup>. Dans cette représentation, les points nodaux et principaux ne sont pas confondus, afin de rendre compte de la différence d'indice de réfraction entre l'air et le milieu vitreux situé devant la rétine (Figure 1.5).

Emsley [Emsley 1952] propose un modèle dit réduit, qui ne comprend qu'une surface réfractive (Figure 1.6). Pour nos algorithmes de suivi du regard (Section 4.1.4), certaines données géométriques sont reprises de ce modèle simplifié. Cette approximation nous permet de proposer des dispositifs matériels moins compliqués (Section 2.4.2.3).

1. L'œil au repos est un œil qui n'accommode pas. Les muscles du cristallin sont au repos.

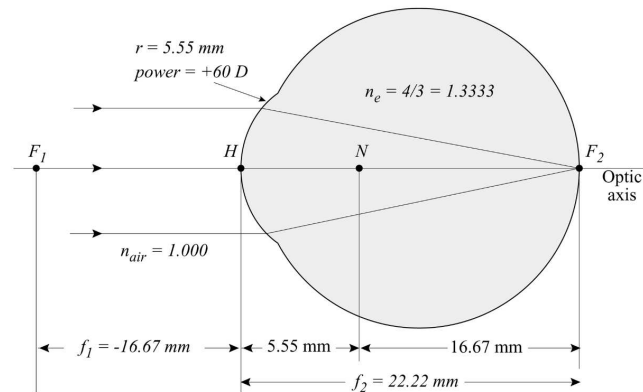


FIGURE 1.6 – Modèle théorique de Emsley et ses paramètres. (Extrait de [Carvalho 2003])

D'autres modèles plus récents se rapprochent davantage de l'anatomie de l'œil, en proposant notamment une cornée asphérique ([Lotmar 1971], [Kooijman 1983], [Navarro 1985]). Parmi ces modèles asphériques, on retrouve des modèles réduits similaires à celui de Emsley ([Thibos 1992], [Thibos 1997]).

En pratique, les différentes surfaces de la cornée et les couches successives du cristallin (Figure 1.2) n'ont pas d'axe commun. C'est pourquoi, différents axes sont définis en fonction des surfaces ou des points particuliers de l'œil (Figure 1.7).

- L'axe pupillaire est la droite perpendiculaire à la cornée passant par le centre  $C_p$  de la pupille.
- L'axe visuel correspond intuitivement au trajet le plus court emprunté par un rayon lumineux issu d'une source ponctuelle vers la fovéa (zone de la rétine ayant l'acuité maximale). En théorie, il est défini par une ligne brisée : une première droite qui relie le point  $P$  observé au premier point nodal  $N$  (Figure 1.5) du système optique et une seconde droite qui relie le deuxième point nodal  $N'$  avec le point focal.
- La ligne de visée est la droite qui relie le centre  $C_p$  de la pupille au point  $P$  de la scène observée par le sujet, appelé point de fixation.
- La ligne de regard est la droite connectant le point de fixation  $P$  et le centre de rotation de l'œil  $C_r$ .
- L'axe médian ("Best Fit Axis" – BFA) est un axe optique approximatif construit à partir des axes optiques de la cornée et du cristallin.
- Pour un point lumineux  $P$ , l'axe kératométrique est l'axe reliant  $P$  à son reflet cornéen. Il passe par le centre de courbure local de la cornée  $C_c$ .

L'axe pupillaire forme un angle  $\lambda$  avec la ligne de visée et un angle  $\kappa$  avec l'axe visuel, qui sont de l'ordre de 6 à 8 degrés. En pratique, la proximité entre l'axe visuel et la ligne de visée permet de considérer ces angles comme équivalents.

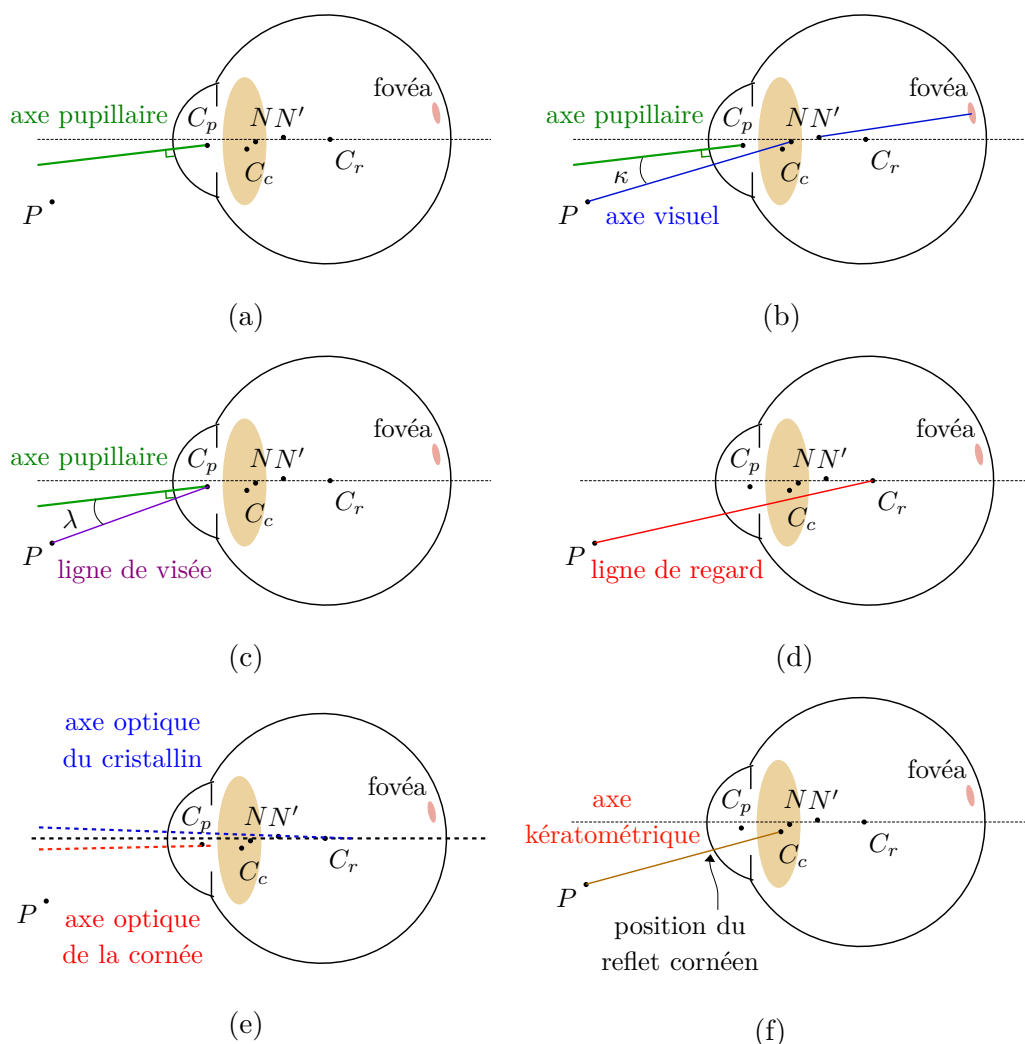


FIGURE 1.7 – Six schémas d'une coupe horizontale de l'œil droit avec ses axes et points particuliers. Les angles sont exagérés pour faciliter la lecture du schéma (a) montre l'axe pupillaire passant par le centre  $C_p$  de la pupille. (b) présente l'axe visuel segmenté en deux droites. L'une passe par le point de fixation  $P$  et par le point nodal objet  $N$ . L'autre émerge du point nodal image  $N'$  et atteint la fovéa. L'axe visuel forme un angle  $\kappa$  avec l'axe pupillaire. (c) illustre la ligne de visée passant par le point de fixation  $P$  et par  $C_p$ . Cette ligne forme un angle  $\lambda$  avec l'axe pupillaire. (d) montre la ligne de regard traversant le centre de rotation de l'œil  $C_r$  et le point de fixation  $P$ . (e) illustre les différents axes optiques de l'œil. Les axes optiques de la cornée (en pointillé rouge) et du cristallin (en pointillé bleu) ne sont pas confondus. Souvent, un axe médian (en pointillé noir) est considéré. (f) présente l'axe kératométrique passant par un point  $P$ , émettant de la lumière, et par le centre de courbure local de la cornée  $C_c$ . Le reflet cornéen de  $P$  est visible là où l'axe kératométrique coupe la cornée.

Dans notre étude, nous nous intéressons essentiellement à l'axe visuel puisqu'il correspond à la "direction du regard", c'est à dire à la direction vers laquelle nous portons notre attention.

Suivant les modèles optiques de l'œil, la ligne de vue est définie différemment. Elle désigne parfois l'axe reliant le centre de la cornée et la fovéa ou le centre de la pupille et la fovéa ([Witzner Hansen 2010], [Zhu 2007a], [Morimoto 2005]). De même pour certains modèles très simplifiés, tels que celui de Emsley, tout ces axes sont confondus.

### 1.1.3 La rétine et le rôle des photorécepteurs

Le système visuel humain ne présente pas une sensibilité uniforme sur l'ensemble du spectre lumineux et n'offre pas une précision et des performances identiques sur la totalité du champ de vision. Ces caractéristiques permettent de définir un cahier des charges pour des lunettes intelligentes équipées de verres actifs. Ce phénomène résulte de l'organisation des cellules photosensibles à la surface de la rétine. Celles-ci se répartissent en deux catégories complémentaires.

- Les cônes discernent les couleurs et offrent une vision nette et précise. De plus, grâce à leur relativement faible sensibilité à la luminosité, ils sont plus efficace en vision photopique (en environnement éclairé). Il en existe trois variétés, chacune percevant dans des gammes de longueurs d'onde différentes (rouge, bleu, vert).
- Les bâtonnets, contrairement aux cônes, sont très sensibles à la luminosité et peu à la couleur, ce qui les rend adéquats pour la vision scotopique (en environnement peu éclairé).

Les bâtonnets sont essentiellement répartis sur la périphérie du champ de vision, tandis que les cônes se situent pour la plupart sur la macula avec un pic de densité sur la fovéa, offrant ainsi une vision centrale précise. On parle alors de forte acuité visuelle, un terme qui qualifie le pouvoir séparateur de l'œil, c'est à dire la capacité à distinguer deux points en fonction de leur écartement. Plus exactement, l'acuité visuelle correspond à l'angle de vue minimal séparant deux points encore discernables. Elle atteint en moyenne une minute d'arc au centre de la vision.

La figure 1.8 donne un aperçu des conséquences de cette répartition hétérogène des cellules sur la variation de l'acuité dans le champ de vision.

De même, les différents types de cônes ne sont pas présents en quantité égale. Les cônes sensibles au vert et au rouge sont prédominants, ce qui induit une vision photopique plus intense dans le vert-jaune de longueur d'onde 555 nm (figure 1.9). En vision scotopique, l'œil est plus sensible au cyan 507 nm, qui correspond à la sensibilité spectrale des bâtonnets.

### 1.1.4 Les mouvements de l'œil

L'œil effectue différents types de mouvements, dont les vitesses de rotation peuvent être élevées. La plupart des systèmes de suivi du regard ne sont pas capables

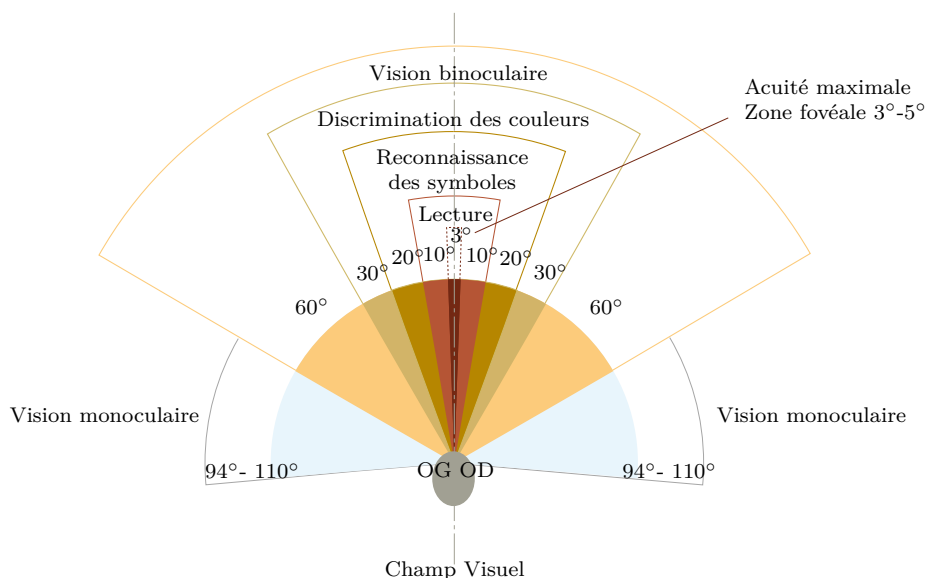


FIGURE 1.8 – Champ visuel binoculaire présenté par catégorie en fonction de l'acuité. (image extraite de wikipedia, sous licence CC BY-SA 3.0, sur [https://upload.wikimedia.org/wikipedia/commons/e/ee/Champ\\_vision.svg](https://upload.wikimedia.org/wikipedia/commons/e/ee/Champ_vision.svg))

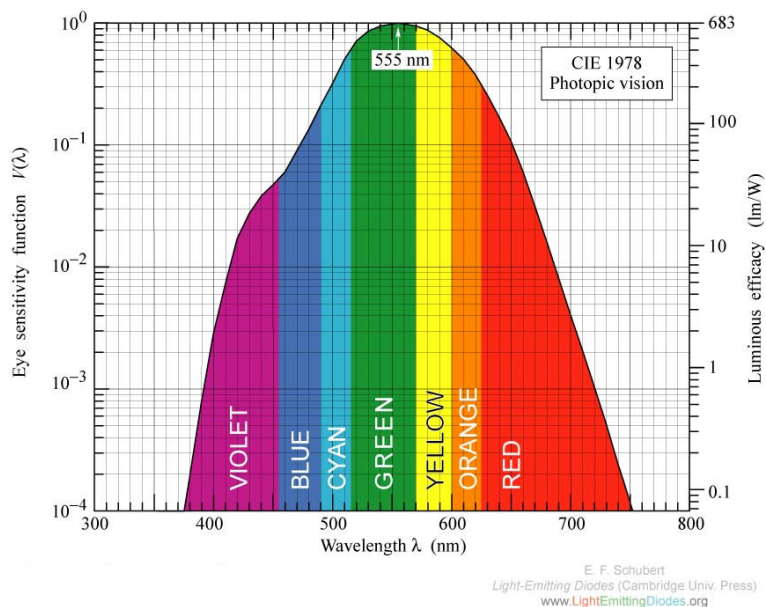


FIGURE 1.9 – Fonction d'efficacité lumineuse spectrale en vision photopique définie par la CIE (Commission Internationale de l'Éclairage) en 1978. Cette fonction fournit pour une longueur d'onde donnée la sensibilité de l'œil. (Extrait de [Schubert 2006])

de détecter les mouvements les plus rapides.

On distingue deux types de mouvements oculaires.

- Les mouvements oculaires de stabilisation consistent à stabiliser l'image rétinienne lors de mouvements de la tête ou du corps. Ils peuvent atteindre des vitesses de l'ordre de  $500^\circ/\text{s}$ .
- Les mouvements oculaires d'orientation consiste à explorer le champ visuel ou à maintenir l'image d'une cible sur la fovéa, même quand celle-ci est en mouvement. Les saccades font partie de ces mouvements et peuvent atteindre jusqu'à  $700^\circ/\text{s}$ .

Tous ces mouvements de rotation s'effectuent autour d'un centre de rotation. Celui-ci n'est pas rigoureusement fixe mais ses déplacements sont minimes. On peut considérer qu'il se trouve au centre de l'œil assimilé à une sphère.

### 1.1.5 La perception 3D

Les systèmes d'affichage 3D, tels que les casques de réalité augmentée ou virtuelle, visent à tromper notre perception tridimensionnelle. Celle-ci dépend de nombreux facteurs.

- La vision binoculaire, également appelée vision stéréoscopique, permet d'estimer finement les distances et le relief. Elle repose sur le fait que les deux yeux voient le monde sous des angles légèrement différents. Cependant, cette stéréoscopie n'opère que sur une plage de distance finie. Pour un objet situé au delà de 30 mètres, les images rétiniennes des deux yeux sont pratiquement identiques.
- Le processus d'accommodation opéré par le cristallin, quand il raffine la netteté de l'image sur la fovéa, fournit une information supplémentaire sur la distance des objets. Si les objets sont situés sur le plan de mise au point, alors ils apparaissent nets.
- Par analyse de la scène perçue, il est possible de déduire quel est objet le plus proche : il suffit de comparer les tailles apparentes de deux objets connus tels qu'ils se présentent dans notre champ de vision, ou d'observer l'occlusion d'un objet par un autre.
- Un déplacement latéral de la tête induit une vitesse apparente des objets de la scène en sens inverse. Plus la vitesse apparente d'un objet est élevée, plus celui-ci est proche.

Si l'ensemble de ces règles ne sont pas satisfaites, alors le porteur des lunettes est susceptible de ressentir un inconfort. Ainsi, bien que la plupart des casques d'affichage 3D proposent une vision stéréoscopique, ils ne génèrent pas des images à différentes profondeurs. L'image perçue est toujours localisée à la même distance. L'utilisateur ne fait pas varier la vergence de son cristallin. Il se produit alors un conflit convergence-accommodation, au sens où les yeux convergent vers un élément synthétique affiché par le casque, mais les cristallins n'accommodent pas à la distance correspondante. Ils s'accommodent à la distance de l'image affichée par le

casque. Ce phénomène génère une sensation de mal-être et de maux de tête après plusieurs minutes d'utilisation du dispositif [ANSES 2014].

## 1.2 Les lunettes du futur

### 1.2.1 Dispositif ophtalmique et nouveaux usages

La thèse présentée dans ce manuscrit s'inscrit dans les travaux de recherche du groupe ESSILOR, leader mondial dans la production de verres et d'équipements ophtalmiques<sup>2</sup>. Très investi dans les activités de recherche et développement, l'entreprise possède trois principaux centres « Innovation & Technologies » à Paris, Dallas, Singapour. De nouvelles technologies ont été développées, telles que Varilux, Crizal, Transitions, Eyezen, Xperio, ou Optifog [Essilor 2018]. D'autres centres existent dont un situé à Labège, qui s'intéresse aux problématiques de lunettes instrumentées dont les propriétés des verres sont commandables par signaux électriques. La thèse s'est déroulée en collaboration avec le site de Labège.

Toujours dans cet objectif d'innovation, ESSILOR et le CNRS ont créé en décembre 2015 le laboratoire commun OPERA (OPTique EmbaRquée Active), dont l'objectif est de développer des verres et des lunettes instrumentées. Par exemple, ces nouveaux verres pourraient s'assombrir à la teinte demandée (fonction d'assombrissement), permettre de faire la mise au point (fonction de focalisation), ou encore afficher du contenu synthétique dans le champ de vision du porteur (fonction d'affichage).

Lorsqu'elles sont dotées des fonctions précédemment évoquées, les lunettes peuvent répondre à des besoins et à des usages liés à la santé ou au confort visuel.

- Les verres s'assombrissent en réponse à un éblouissement soudain et/ou localisé. Il peut par exemple s'agir d'une gêne causée par l'apparition du soleil bas sur l'horizon dans le champ de vision d'un conducteur.
- La focale (puissance) des verres s'adapte en fonction de la distance de l'objet visé par le porteur, pour proposer une correction adéquate.
- Du contenu synthétique s'affiche dans le champ de vision du porteur pour apporter des solutions à des défauts visuels tels que le manque de contraste ou la DMLA<sup>3</sup> (Figure 1.10). En effet, le dispositif pourrait ré-hausser les contours de scènes peu contrastées pour en améliorer la visibilité, ou encore ré-afficher le contenu perdu par la DMLA sur une autre portion de la rétine encore active.
- Les verres affichent du contenu synthétique pour des exercices de ré-éducation orthoptique (Figure 1.11).

2. ESSILOR détenait 37% du marché mondial des verres et lentilles en 2013.

3. La dégénérescence maculaire liée à l'âge (DMLA) est une maladie touchant la zone centrale de la rétine et entraînant une perte progressive de la vision centrale. Elle laisse habituellement intacte la vision périphérique ou latérale.

- Le contenu oral d’une conversation s’affiche sous forme de texte à travers les lunettes pour aider les personnes mal-entendantes.



FIGURE 1.10 – Illustration de l’influence de la DMLA sur le champ de vision.

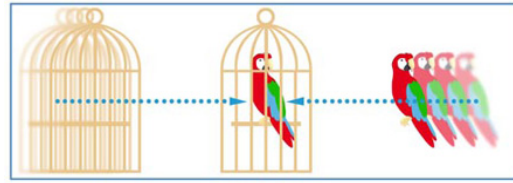


FIGURE 1.11 – Illustration d’un exercice d’orthoptie visant à ré-éduquer la vision binoculaire.

Hormis pour le domaine de la santé, axe d’étude traditionnel de ESSILOR, d’autres besoins peuvent aussi être satisfaits par un système d’affichage de réalité augmentée. En voici quelques-uns :

- Aide à la localisation et la navigation.
- Assistant électronique mentionnant les prochains rendez-vous, les noms des interlocuteurs ou encore le nom d’enseignes et de monuments.
- Support visuel pour l’éducation [Liarokapis 2004] et les musées [Huang 2009].
- Outils de travail facilitant la visualisation de données en 3D ou proposant une assistance visuelle dans des tâches manuelles. Par exemple, ce dispositif peut aider à la conception de nouvelles pièces mécaniques sur ordinateur ou apporter un soutien pour des opérations chirurgicales difficiles [Haouchine 2013].
- Système améliorant l’interaction et les conversations à distance en affichant la personne distante à proximité du porteur [Orts-Escolano 2016].
- Dispositif pour le divertissement proposant des jeux ou films mêlés à la réalité.

### 1.2.2 Définition de la réalité augmentée et de la réalité virtuelle

Au sens large, la réalité augmentée est une technologie qui permet d’ajouter des informations visuelles, auditives, et éventuellement gustatives, olfactives ou tactiles à la perception de l’environnement réel. Le principe général est d’introduire un élément (objet ou personne) virtuel au sein de l’environnement réel en s’appuyant sur les capacités de perception de l’utilisateur. Inversement, la réalité virtuelle est complètement immersive et isole l’utilisateur de son environnement réel. Par exemple, les casques de réalité virtuelle (Oculus Rift, HTC Vive, Morpheus) affichent une scène synthétique (Figure 1.14). Comme ces casques sont opaques, l’utilisateur ne perçoit plus la scène réelle. Les figures 1.12 et 1.13 illustrent une scène perçue respectivement à travers d’un casque de réalité augmentée et d’un casque de réalité virtuelle.

Les casques de réalité virtuelle sont généralement constitués d’un écran devant lequel sont placées deux lentilles, qui permettent d’observer une scène virtuelle avec un champ de vision de l’ordre de 100 degrés. De leur côté, les casques de réalité

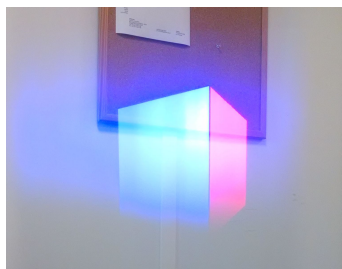


FIGURE 1.12 – Aperçu d'un affichage synthétique (cube coloré) pour un œil généré par un casque de réalité augmentée, le lumus DK50.

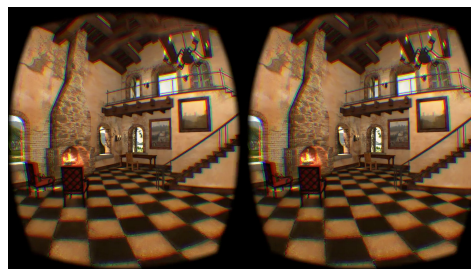


FIGURE 1.13 – Aperçu d'une scène synthétique pour les deux yeux générée pour un casque de réalité virtuelle, l'oculus DK2.

augmentée sont la plupart du temps équipés d'une surface semi-réfléchissante en avant de l'œil et d'un afficheur (écran LCD, OLED ou projecteur laser). Alors que l'environnement réel reste directement visible à travers le dispositif, l'information affichée est réfléchiée par la surface semi-réfléchissante vers l'œil (Figure 1.15).



FIGURE 1.14 – Un casque de réalité virtuelle Oculus Rift (Image extraite de [https://commons.wikimedia.org/wiki/File:Boy\\_wearing\\_Oculus\\_Rift\\_HMD.jpg](https://commons.wikimedia.org/wiki/File:Boy_wearing_Oculus_Rift_HMD.jpg))

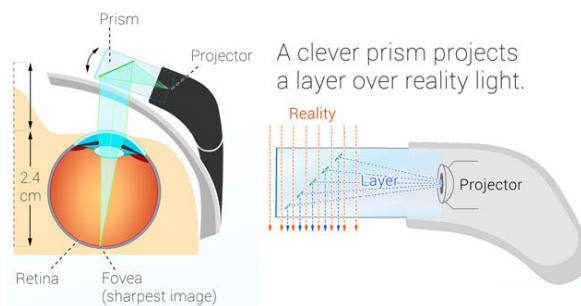


FIGURE 1.15 – Schéma illustrant le fonctionnement des Google Glasses (Image extraite de <https://www.varifocals.net/google-glass/>)

Plus généralement, la réalité augmentée et la réalité virtuelle sont des cas particuliers de la réalité mixte (Figure 1.16). La réalité mixte, définie par Milgram [Milgram 1994], désigne les *réalités*, où l'environnement réel est mélangé à l'environnement virtuel. Le *continuum virtuel* correspond à l'ensemble des réalités mixtes possibles. Comme l'indique la figure 1.16, l'environnement réel et l'environnement virtuel sont les deux extrémités de cet ensemble, la réalité virtuelle et la réalité augmentée étant deux exemples de réalité mixte existant dans le *continuum virtuel*.

Dans la littérature, on retrouve trois catégories de systèmes d'affichage de réalité

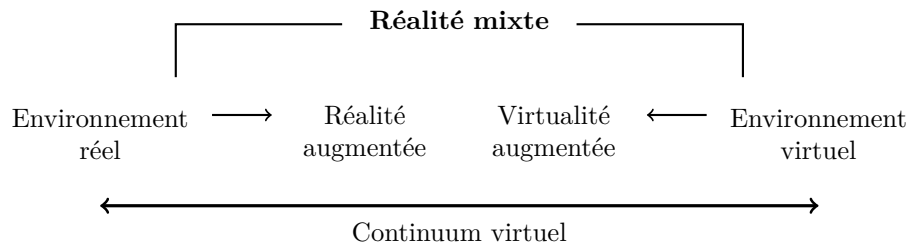


FIGURE 1.16 – Représentation du concept de réalité mixte de Milgram.



FIGURE 1.17 – Exemple de HHD, où une voiture est affichée à l'emplacement d'un tag perçu par un téléphone.

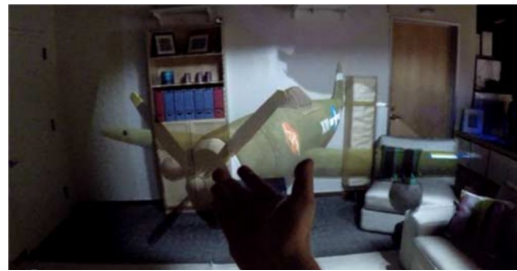


FIGURE 1.18 – Exemple de système d'affichage spatial, où un projecteur affiche un avion en tenant compte du relief de la scène [Benko 2014].

augmentée *visuelle* :

- Les systèmes d'affichage spatiaux (spatial displays) projettent ou affichent du contenu virtuel directement sur les objets ou les surfaces de l'environnement (Figure 1.18).
- Les systèmes d'affichage portés à la main (Hand-Held Displays – HHDs) implémentent les méthodes d'affichage du contenu augmenté sur un écran (smartphone, tablette) tenu à la main (Figure 1.17).
- Les systèmes d'affichage montés sur la tête (Head Mounted Displays – HMDs) sont des dispositifs d'affichage fixés sur les lunettes ou un casque.

### 1.2.3 Technologies, Contraintes, Opportunités

#### 1.2.3.1 Applications potentielles et complexité

Parmi les différentes fonctionnalités citées plus haut, on peut distinguer différents degrés de difficulté et de dépendance avec la scène réelle. Ces degrés se traduisent en termes de complexité de mise en œuvre.

- *Degré 1* : Les lunettes affichent des données indépendantes de la scène. Par exemple, pour les exercices d'orthoptie (Figure 1.11), les deux objets synthétiques (le perroquet et la cage) flottent dans l'air. Seule l'estimation de

la direction pointée par les yeux peut être utile si l'on cherche à évaluer la réussite de cet exercice.

- *Degré 2* : Le contenu synthétique est affiché dans une direction et à une distance approximative, en s'aidant de l'information de localisation fournie par un GPS et une centrale inertielle<sup>4</sup> et d'une carte issue d'un Système d'Information Géographique (SIG)<sup>5</sup>. Les applications de localisation et navigation (Figure 1.19), où le système affiche une flèche dans la direction que le porteur doit emprunter, entrent dans cette catégorie.
- *Degré 3* : De l'information visuelle est ajoutée à proximité d'un objet détecté dans la scène (Figure 1.20). Cette tâche nécessite, quant à elle, un système de vision et un algorithme de détection et de suivi de l'objet en question. Un dispositif équipé d'un assombrissement dynamique peut être basé sur un fonctionnement similaire, qui détecterait des sources lumineuses dans l'environnement pour déduire la teinte du verre adaptée.
- *Degré 4* : Des éléments synthétiques sont superposés sur un ou plusieurs éléments réels préalablement détectés et segmentés. Ici, la dépendance avec la scène réelle est plus forte. En effet, un léger décalage de l'objet virtuel par rapport au réel est facilement perceptible. Pour que la superposition soit correcte malgré les mouvements de tête du porteur et le délai de traitement de la détection et la segmentation, il est nécessaire d'estimer ce mouvement afin d'anticiper l'affichage. Un tel système peut être utilisé pour l'aide à la réalisation de tâches de maintenance en environnements statiques (Figure 1.21).
- *Degré 5* : Le dispositif demande une perception plus fine de la scène réelle, par le biais d'une reconstruction en trois dimensions de l'environnement. Cette capacité peut être utile pour des applications interactives de jeux ou de films, où le virtuel se déplace en tenant compte des contraintes de l'environnement réel.

### 1.2.3.2 Technologies

Quel que soit leur niveau de complexité, ces systèmes sont tous dotés de capacités de perception, d'analyse et d'action. Parmi les capacités de perception, on distingue deux catégories : les capteurs proprioceptifs d'un système renseignent sur son état interne (e.g., les codeurs d'une liaison d'un robot manipulateur), alors que les capteurs extéroceptifs renseignent sur son interaction avec son environnement (e.g., la température ou la luminosité ambiante). Ainsi, pour les lunettes du futur,

---

4. Une centrale inertielle, aussi appelée IMU, "inertial measurement unit", est un instrument permettant d'estimer l'orientation. Il est constitué de deux ou trois variétés de capteurs : trois accéléromètres mesurant les accélérations selon trois axes différents, trois gyromètres mesurant la vitesse angulaire sur trois axes et optionnellement trois magnétomètres fournissant le cap à partir de la mesure du champ magnétique.

5. Un système d'information géographique est un outil informatique permettant de recueillir, de stocker, d'analyser et de gérer un grand nombre de données réparties dans le monde entier.



FIGURE 1.19 – Exemple d’affichage synthétique pour la navigation.



FIGURE 1.20 – Exemple d’affichage synthétique pour l’assistance.



FIGURE 1.21 – Affichage synthétique pour la réalisation de tâches de maintenance en environnement statique.

les technologies associées à la proprioception se limitent essentiellement aux centrales inertielles, qui estiment l’orientation des lunettes. En revanche, les capteurs extéroceptifs peuvent par exemple consister en des caméras passives ou actives<sup>6</sup>, des capteurs temps de vol, des photodiodes ou des microphones, en vue de recueillir des informations sur la scène (e.g., pour des tâches de localisation, détection, suivi, reconstruction 3D) et le porteur (e.g., pour le suivi de son regard ou l’estimation de la taille de sa pupille).

Dans le cadre de la thèse, les actions envisagées sur des lunettes sont au nombre de trois.

- *Obscurcissement du verre* L’obscurcissement est obtenu au moyen d’une technologie développée par l’équipe d’ESSILOR Labège : le verre électro-chrome. Ce verre est une cellule constituée d’une solution électrolytique comprise entre deux coques fines agissant comme des électrodes. En appliquant une tension entre les bords des deux coques, une réaction d’oxydo-réduction se déclenche au sein de la cellule entraînant un changement de teinte de la solution.
- *Changement de focale (puissance)* Différentes technologies permettent de modifier la puissance de verres de lunettes. Citons deux exemples. Le premier s’appuie sur le principe qu’une goutte de liquide agit comme une lentille, dont la focale dépend de l’étalement de la goutte. Plus la goutte est bombée, plus sa focale est faible et inversement. Partant de cette idée, il existe des lentilles<sup>7</sup> liquides, dont la forme change lorsqu’elles sont mises sous tension. Un second exemple de verres de lunettes de focale ajustable consiste en deux lentilles montées l’une contre l’autre. La focale du système s’ajuste en couissant les deux lentilles l’une par rapport à l’autre. Les lentilles de Alvarez-Lohmann en sont un exemple.

6. Les caméras actives projettent de la lumière sur la scène, puis la capturent après réflexion sur les surfaces des objets de la scène. Par opposition, les caméras passives observent la scène avec la luminosité naturellement présente.

7. Les lentilles de la société Optotune [www.optotune.com](http://www.optotune.com) et les Varioptic de Corning [www.corning.com](http://www.corning.com)

- *Affichage d'éléments synthétiques* Dans la littérature, plusieurs technologies permettent l'affichage d'éléments synthétiques par le biais d'un dispositif installé sur une lunette (voir chapitre 2.1). ESSILOR développe également son propre système. Celui-ci est basé sur un miroir holographique réfléchissant vers l'œil le flux lumineux en provenance d'un écran LCD fixé sur une branche de la lunette. L'intérêt du miroir holographique, en plus de sa transparence, est de réfléchir l'information lumineuse dans une direction choisie, contrairement à un miroir traditionnel.

### 1.2.3.3 Contraintes et opportunités

L'utilisation de ces nouveaux dispositifs implique un certain nombre de contraintes. En effet, l'ensemble de l'appareillage doit être incorporé dans des lunettes et doit donc répondre à des besoins de confort, légèreté, forme et apparence. Il doit aussi être compatible avec la vision du porteur sans altérer sa perception ni sa santé visuelle.

**A Contraintes d'embarquabilité** Le caractère embarqué du système impose des restrictions de taille, poids et consommation énergétique. Ces limitations évoluent en fonction de l'application visée. Pour des lunettes grand public à usage quotidien, ces exigences sont élevées : poids et encombrement comparables aux paires de lunettes traditionnelles et autonomie énergétique de plusieurs jours. Inversement, des systèmes dédiés à la santé peuvent s'autoriser des limitations moins fortes : les calculs importants sont pris en charge par un ordinateur déporté dans une poche alimenté par une batterie plus imposante.

**B Contraintes pour le confort visuel** Les caractéristiques et le fonctionnement de la vision du porteur définissent des limitations et des règles que des lunettes actives doivent respecter. Ces principes sont présentés ci-dessous :

- *Fréquence de rafraîchissement et latence de l'affichage* : Une fréquence trop faible ou une latence trop importante altère la perception de notre mouvement. Ce dernier est habituellement ressenti à la fois par le système visuel et l'oreille interne. Ainsi lorsque le déplacement détecté par ces deux procédés ne coïncide pas, l'utilisateur peut ressentir une sensation de malaise.
- *Largeur du champ de vision et acuité visuelle* : Selon la finalité du dispositif, il peut être nécessaire de couvrir l'ensemble du champ visuel, notamment pour les solutions proposées à la DMLA, pour l'amélioration de contraste et pour la protection contre l'éblouissement. De plus, une grande partie du champ de vision requiert une précision angulaire importante, car malgré une vision nette restreinte à la zone centrale, les saccades de l'œil comblent ce manque et permettent au cerveau de reconstruire une image nette plus grande.
- *Perception de la 3D et conflit convergence-accommodation (définis à la section 1.1.5)* : Pour générer la 3D, les systèmes d'affichage actuels utilisent la

stéréoscopie mais génèrent un conflit convergence-accommodation. Ainsi, ces dispositifs affichent une image différente pour chaque œil en tenant compte de la parallaxe. Cependant l'assemblage optique génère une image sur un seul et même plan à une distance fixe devant l'utilisateur. Pour permettre un usage confortable, le processus d'accommodation doit aussi être pris en compte soit en proposant un système optique capable d'afficher une image située à différente distance, soit en adaptant la distance de mise au point en fonction de la distance séparant l'utilisateur du point d'intérêt qu'il regarde.

- *Risque de paresse des yeux* : Les deux procédés proposés précédemment de verres à focalisation ou obscurcissement variables agissent en concurrence avec le cristallin et l'iris. C'est à dire que le verre ajuste sa distance de focalisation (respectivement son obscurcissement) en même temps que le cristallin (respectivement que la rétraction de l'iris), ce qui peut avoir pour conséquence d'alléger le travail habituellement exercé par l'œil et de le rendre paresseux, en altérant sa capacité de focalisation ou d'adaptation à la variation de luminosité.

**C Opportunités** De nouveaux services peuvent être sans doute envisagés, qui tirent parti de la plasticité cérébrale. Les capacités du cerveau à apprendre pour appréhender des situations imprévues permettent, par exemple, qu'un individu atteint de DMLA puisse exploiter, à l'issue d'une période d'adaptation, le ré-affichage sur sa vision périphérique des informations qui ne peuvent être perçues par ses récepteurs visuels déficients. Un phénomène d'adaptation voisin a déjà été observé lors d'immersion en réalité virtuelle : suite à une phase d'apprentissage, le système visuel parvient à compenser l'inconfort initialement observé [Regan 1995].

## 1.3 Un système robotique embarqué

### 1.3.1 Des scénarios types aux fonctions embarquées

Il existe de nombreuses similarités entre la robotique et les lunettes actives. Tout d'abord, un système robotique est une machine dotée de capacités de perception, de raisonnement et d'action. Ainsi, à partir de sa perception, elle interagit de manière autonome avec l'environnement, les hommes et les robots qui l'entourent. Ses capacités de raisonnement lui permettent de réagir face à des scènes et des situations nouvelles. En résumé, un robot est basé sur le schéma de fonctionnement suivant "perception, décision, action".

Tout comme la robotique, la lunette instrumentée se caractérise par le même schéma de fonctionnement (Figure 1.22) : le système perçoit l'environnement et le porteur, analyse les données perçues pour prendre une décision et agit sur le verre en conséquence (obscurcissement, focalisation, affichage). Les lunettes, ayant besoin de capacités de perception et de décision, sont donc confrontées aux mêmes difficultés et besoins qu'un robot. Par exemple, il peut être nécessaire de construire une carte d'un environnement inconnu, localiser le système dans celle-ci, détecter

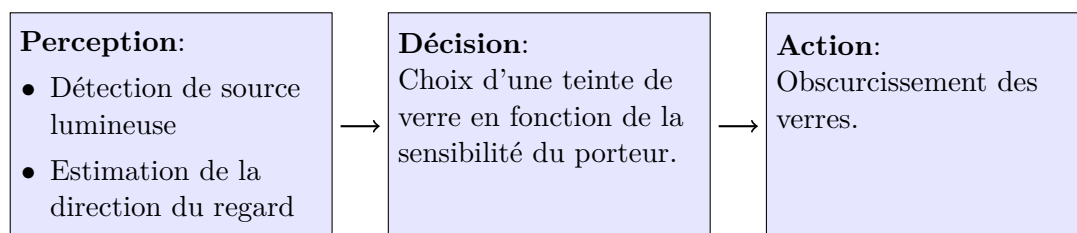


FIGURE 1.22 – Exemple du processus "perception, décision, action" pour des lunettes obscurcissantes dont la teinte varie suivant la position et l'intensité de la source lumineuse par rapport à la direction du regard du porteur et à sa sensibilité.

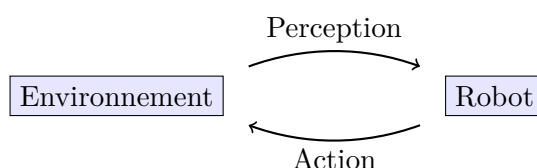


FIGURE 1.23 – Boucle de rétro-action, où le robot agit et adapte son action sur l'environnement en fonction de la perception qu'il en a.

et suivre des objets d'intérêt. Par conséquent, les algorithmes et outils développés pour la robotique peuvent être ré-utilisés pour les lunettes.

Pour satisfaire les processus de perception, de décision et d'action, la robotique regroupe de nombreuses disciplines : l'informatique, la mécanique, l'électronique, l'automatique, les capteurs, le traitement du signal et de l'information, la décision, l'intelligence artificielle, etc. Pour les mêmes raisons, la lunette active recouvre diverses thématiques avec l'informatique, la mécanique, l'électronique, l'optique, la chimie, etc.

Autre similitude, les deux systèmes, robots et lunettes instrumentées, présentent tout deux des contraintes d'embarquabilité (taille, poids, temps de calculs et consommation énergétique).

En revanche, une différence importante peut être notée : la boucle de rétro-action sur le processus "perception, décision, action" est plus complexe pour les lunettes. En effet, la perception du porteur intervient dans cette boucle et ajoute une composante basée sur le ressenti, qui est difficilement quantifiable et évaluable. De plus, cette notion est relative à chaque individu et à chaque contexte. Les figures 1.23 et 1.24 illustrent cette différence.

### 1.3.2 Architectures logicielles robotiques pour le prototypage de fonctions

#### 1.3.2.1 Architecture modulaire

Pour exécuter une mission, un robot doit exécuter un grand nombre d'activités de manière simultanée. Par exemple, pour planifier et exécuter un déplacement

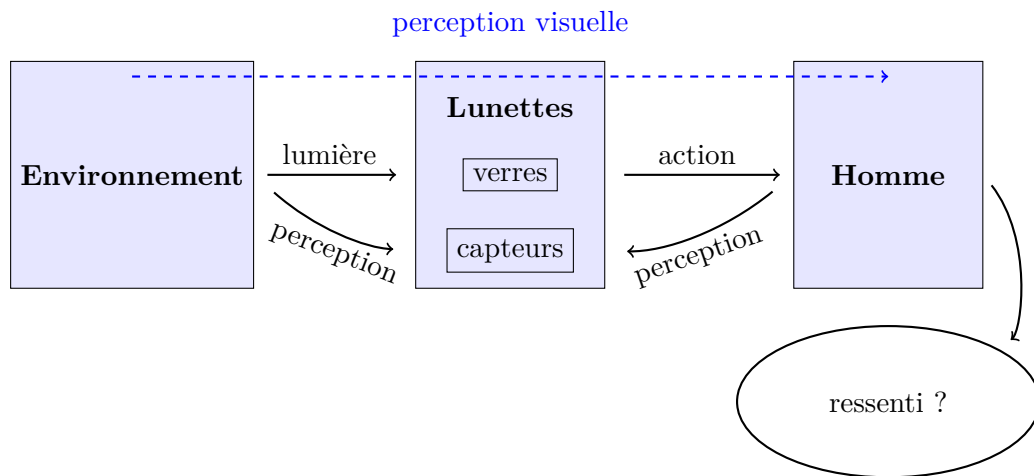


FIGURE 1.24 – Schéma illustrant la difficulté de mise en place d’une boucle de rétro-action entre les lunettes, le porteur et l’environnement. Les capteurs équipés sur les lunettes acquièrent des informations sur la scène et l’utilisateur. C’est l’étape de perception. Ensuite sur la base de ces informations et des capacités de raisonnement des lunettes, les verres changent leur état. Ce sont les étapes de décision et d’action des lunettes. La lumière issue de la scène, qui traverse les lunettes et atteint les yeux de l’utilisateur, est transformée par l’action du verre. La perception visuelle de l’utilisateur est donc modifiée. Pour ajuster l’action du verre de manière à améliorer cette perception visuelle, il est nécessaire d’avoir accès au ressenti de l’utilisateur. En d’autres termes, le système a besoin de savoir quel est l’effet d’une action des verres sur la perception visuelle, le confort et le ressenti de l’utilisateur. Toutefois, ces informations peuvent être difficiles à obtenir.

dans un lieu inconnu, il doit percevoir l'environnement au moyen d'une ou plusieurs caméras embarquées, appliquer un ensemble de traitements adéquats aux images obtenues, acquérir à une fréquence plus élevée les données provenant de son odométrie et/ou de sa centrale inertielle, fusionner l'ensemble de ces informations dans un schéma de SLAM (simultaneous localization and mapping) pour construire une carte de l'environnement et se localiser concomitamment dans celle-ci, générer une trajectoire faisable dans la carte obtenue, puis s'asservir sur celle-ci (Figure 1.25). Il est par conséquent nécessaire de gérer l'exécution parallèle de ces fonctions (en respectant leurs priorités respectives), leur synchronisation, leur accès à des données partagées, etc.

Cette complexité inhérente à la robotique justifie le déploiement d'architectures logicielles temps réel. Celles-ci sont constituées de composants (modules), organisés en niveaux hiérarchiques selon les contraintes auxquelles ils sont soumis et le degré d'abstraction dans les données et modèles qu'ils manipulent. Ainsi, au-dessus du matériel et de l'instrumentation associée se situe le niveau fonctionnel. Les modules correspondants doivent respecter des exigences sévères en termes de temps de traitement, communications, etc. C'est pourquoi, ils sont généralement codés dans des langages tels que C ou C++. On trouve ici des serveurs de données capteurs, des modules de traitement de l'information perçue, des composants en charge de la locomotion, du calcul d'odométrie, de la planification et de l'exécution réactive de trajectoires, etc. Les composants du/des niveaux hiérarchiques supérieurs manipulent des données, des modèles et des bases de connaissances plus abstraits. Ils concernent le contrôle d'exécution, la connaissance de situation, la planification de tâches, le raisonnement sur des buts, l'action délibérée, l'apprentissage, etc. Ils constituent un ensemble indispensable pour doter le robot de capacités de raisonnement, d'adaptation (e.g., à des environnements/situations inconnus), et d'interaction (avec d'autres robots et/ou des humains). Du fait qu'ils hébergent des processus plus symboliques soumis à des contraintes temporelles moins sévères, ils sont généralement implémentés au moyen de langages interprétés. Le caractère modulaire de telles architectures facilite le développement de fonctions et permet leur ré-utilisabilité [Alami 1998], [Nana 2005]. Il simplifie également l'évaluation de différents algorithmes pour la réalisation d'une même fonction ; il suffit pour cela de développer un lot de composants interchangeable, chacun étant basé sur un algorithme distinct, mais utilisant des interfaces identiques pour l'échange de données.

### 1.3.2.2 Interfaces de communication

Le réseau qui supporte l'appel de procédures et l'échange de messages entre les composants logiciels du niveau fonctionnel est créé au moyen d'un "middleware", ou "intergiciel". Ainsi, le middleware ROS (Robot Operating System)[Quigley 2009] est omniprésent en robotique. Il permet de déployer un ensemble de noeuds ("nodes") s'exécutant en parallèle (éventuellement sur des machines distinctes) et communiquant via des services ("actions") et échangeant des données via un mécanisme de publication/souscription à des ports ("topics"). De nombreux paquets ROS sont

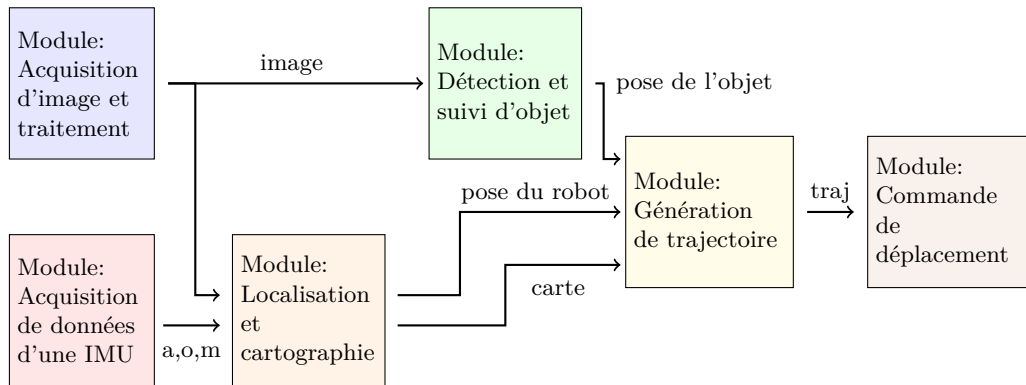


FIGURE 1.25 – Exemple d’architecture modulaire pour un robot devant effectuer une tâche de déplacement pour se positionner face à un objet de la scène. Sur le schéma, *a,o,m* représentent les données issues respectivement des accéléromètres, oscillomètres et magnétomètres de l’IMU et *traj* désigne la trajectoire calculée du robot. Les modules sont représentés par des rectangles et les messages par des flèches.

disponibles en open-source.

Comme indiqué plus haut, les lunettes instrumentées s’apparentent à un système robotique. Par conséquent, le déploiement de fonctions sur des architectures logicielles pour la robotique facilite les phases de conception et de tests de prototypes, et permet accessoirement de bénéficier de nombreux composants disponibles sur étagère en open-source. C’est pourquoi l’ensemble de nos logiciels dédiés à des prototypes de lunettes a été intégré sur le middleware ROS. Il convient de noter qu’une telle architecture fonctionnelle ne conviendra certainement pas pour des produits finaux devant être optimisés en terme d’empreinte mémoire, de cadence, de latence et de consommation énergétique, alors que la rapidité et la facilité de mise en œuvre sont prioritairement recherchées lors du prototypage.

### 1.3.3 Résumé - Vue d’ensemble

En résumé, les lunettes proposées ne sont plus uniquement des systèmes optiques et passifs pour la correction de la vue. Elles deviennent des dispositifs actifs s’adaptant au porteur et à l’environnement, dans le but d’améliorer toujours plus le confort de l’utilisateur. Elles pourront s’obscurcir pour prévenir des éblouissements, changer de focale pour corriger la presbytie, ré-afficher du contenu visuel pour compenser la perte de vision due à la DMLA ou améliorer la perception dans les scènes peu contrastées. En plus des applications pour la santé, ces lunettes constitueront un outil supplémentaire pour l’assistance au travail, l’éducation, la culture et le divertissement.

Pour assurer ces nouvelles fonctionnalités, de nombreux composants seront nécessaires, tels que des verres actifs, des capteurs, une unité de calcul et de communication, et une source d’énergie. Toutefois, ils devront être intégrés en respectant

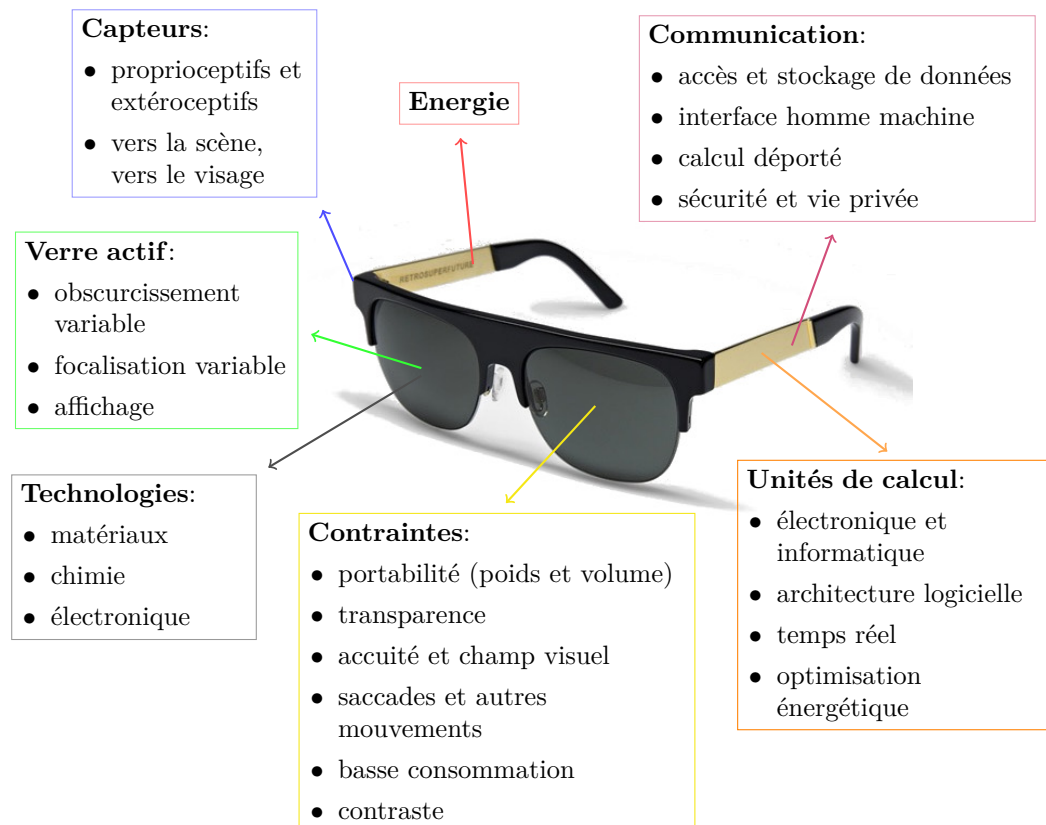


FIGURE 1.26 – Récapitulatif des lunettes instrumentées : contraintes, technologies, fonctions.

des contraintes fortes d'embarquabilité (encombrement, légèreté et consommation énergétique). D'autre part, les lunettes devront s'accorder aux caractéristiques et exigences du système visuel humain (acuité visuelle, champ de vision, perception du mouvement, convergence, accommodation, focalisation et rétraction de l'iris), pour ne pas altérer son fonctionnement et ne pas être rejetées par le potentiel porteur.

De par la nature et la complexité des processus mis en jeu dans les fonctions qu'elles permettent de réaliser, les lunettes peuvent être comparées à un système robotique. Que ce soit pour un robot ou des lunettes, on trouve en effet des primitives de perception, de localisation, de décision, et d'actionnement. Ainsi, certaines techniques et méthodes peuvent être empruntées à la robotique pour les appliquer à l'optique embarquée et active.

Un récapitulatif de la description des lunettes actives est illustré dans la figure 1.26.

## 1.4 Objectifs de la thèse

Cette thèse vise à étudier une lunette instrumentée comme un système robotique, et à tester divers algorithmes et méthodes sur des prototypes. Plus précisément, ses objectifs se décomposent en deux sous-parties : la mise en place de composants logiciels et matériels pour la réalisation de fonctions élémentaires, puis l'intégration de ces briques sur des démonstrateurs. Parmi ces fonctions élémentaires figurent l'affichage de contenu synthétique, l'oculométrie, la localisation des lunettes, la cartographie et l'analyse de la scène. Certaines fonctions sont le résultat de travaux de recherche en robotique ou vision par ordinateur ; leurs auteurs ont mis à disposition en open-source des bibliothèques ou composants logiciels associés. D'autres fonctions ont été mises au point au cours de la thèse. Les composants correspondants peuvent comporter plusieurs aspects : le matériel (assemblages, capteurs, actionneurs) ; l'interface de programmation applicative<sup>8</sup> ("Application Programming Interface"—API) permettant de communiquer avec l'électronique ; l'algorithme principal ; le composant ROS encapsulant ce contenu algorithmique dans un noeud.

Le manuscrit est organisé comme suit. L'état de l'art lié aux modules principaux est présenté dans le chapitre 2. Une étude plus approfondie est menée sur l'élaboration d'un oculomètre basse consommation au chapitre 3. Des démonstrateurs ont été mis au point comme preuves de concept en vue de tester le fonctionnement de chacun des modules, de les évaluer et de les valider. Ainsi, un outil de simulation a été développé pour l'évaluation d'un module d'oculométrie (Chapitre 3). D'autre part, plusieurs démonstrateurs matériels ont été réalisés sur la base de technologies variées, en vue d'intégrer diverses fonctions dans des composants et de procéder à leurs tests (Chapitre 4).

---

8. En informatique, une API regroupe plusieurs outils logiciels permettant à un programme informatique de communiquer avec d'autres logiciels. Par exemple, pour interagir avec un capteur et accéder aux mesures effectuées par ce dernier, un programme informatique a besoin d'une API pour communiquer avec les logiciels associés au capteur et disposer de ses services.

Confidentiel

# Composants matériels et logiciels

## Sommaire

|            |                                                                                  |           |
|------------|----------------------------------------------------------------------------------|-----------|
| <b>2.1</b> | <b>Affichage de contenu</b>                                                      | <b>34</b> |
| 2.1.1      | Casques de réalité augmentée                                                     | 35        |
| 2.1.2      | Limitations des dispositifs de réalité augmentée                                 | 36        |
| 2.1.3      | Conclusion                                                                       | 39        |
| <b>2.2</b> | <b>Perception de l'environnement</b>                                             | <b>39</b> |
| 2.2.1      | Modélisation des caméras                                                         | 40        |
| 2.2.2      | Perception 3D                                                                    | 45        |
| <b>2.3</b> | <b>Localisation des lunettes dans la scène</b>                                   | <b>55</b> |
| 2.3.1      | Localisation dans un environnement maîtrisé                                      | 56        |
| 2.3.2      | Localisation dans un environnement inconnu                                       | 58        |
| 2.3.3      | Conclusion                                                                       | 62        |
| <b>2.4</b> | <b>Détection et suivi du regard du porteur</b>                                   | <b>62</b> |
| 2.4.1      | Techniques d'oculométrie (suivi du regard, <i>eye-gaze tracking</i> )            | 62        |
| 2.4.2      | Techniques basées caméra et extraction d'éléments caractéristiques               | 63        |
| 2.4.3      | Oculomètre basse consommation                                                    | 68        |
| 2.4.4      | Conclusion                                                                       | 70        |
| <b>2.5</b> | <b>Rendu synthétique de l'environnement et affichage d'information augmentée</b> | <b>70</b> |
| 2.5.1      | Deux procédés de synthèse d'image                                                | 70        |
| 2.5.2      | Rendu d'une image par DIBR pour un autre point de vue                            | 73        |
| 2.5.3      | Conclusion                                                                       | 82        |
| <b>2.6</b> | <b>Étalonnage du dispositif</b>                                                  | <b>82</b> |
| 2.6.1      | Modèle de projection                                                             | 83        |
| 2.6.2      | Étalonnage d'un OST-HMD                                                          | 84        |
| 2.6.3      | Étalonnage d'un VST-HMD                                                          | 87        |
| 2.6.4      | Conclusion                                                                       | 88        |
| <b>2.7</b> | <b>Conclusion</b>                                                                | <b>89</b> |

Les lunettes décrites section 1.2.1 doivent permettre de modifier la perception de l'utilisateur en fonction de ses besoins à un instant donné. Parmi ces besoins, on retrouve l'aide à la focalisation sur des objets de la scène, la protection solaire

adaptative et l’affichage de contenu synthétique intégré à la scène réelle. Ces besoins sont liés à des éléments de l’environnement. Ainsi, les lunettes doivent percevoir conjointement l’environnement et l’utilisateur. Les informations extraites sont ensuite analysées pour permettre la commande adéquate des verres, en s’adaptant en temps réel à l’environnement et à l’utilisateur.

Du fait que les lunettes constituent un dispositif mobile, l’environnement évolue sans cesse. Pour faciliter la tâche d’acquisition et d’analyse, il est souvent utile de le cartographier et de localiser les lunettes dans la carte obtenue. De cette façon, la position d’un élément précédemment détecté et localisé peut être prédite sans avoir à effectuer une nouvelle étape de détection. De plus, la cartographie permet d’assimiler davantage de données, et d’enrichir ainsi la phase d’analyse.

En résumé, des lunettes actives doivent intégrer un ensemble de composants matériels et logiciels. Ceux-ci peuvent être vus comme les briques élémentaires d’une construction indispensable au bon fonctionnement. Ce chapitre présente un état de l’art des différents composants :

- *Première section* : Les composants d’affichage pour la réalité augmentée sont décrits.
- *Deuxième section* : Nous présentons les composants permettant de percevoir les informations photométriques et géométriques de l’environnement, avec notamment une description des modélisations des caméras et une présentation des solutions de perception de la 3D de la scène.
- *Troisième section* : Différentes solutions de composants de localisation des lunettes dans la scène sont abordées.
- *Quatrième section* : Nous nous intéressons au suivi du regard du porteur. Cette modalité constitue un moyen de communication avec les lunettes (e.g., *via* la sélection de boutons affichés en sur-impression) ainsi qu’un moyen de paramétrisation des actions effectuées par la lunette (e.g., pour des verres à focale variable, *via* le calcul de l’intersection entre les directions de regard de l’œil droit et de l’œil gauche).
- *Cinquième section* : Les méthodes d’affichage pour les lunettes de réalité augmentée sont étudiées.
- *Sixième section* : Nous présentons les techniques d’étalonnage associées aux lunettes équipées de système d’affichage. Ces techniques visent à assurer que le contenu synthétique s’affiche à la position ciblée dans le champ de vision du porteur.

## 2.1 Affichage de contenu

Les casque ou lunettes de réalité augmentée (“Head Mounted Display” - HMD) sont des dispositifs portés sur la tête de l’utilisateur qui affichent dans son champ de vision du contenu synthétique n’existant pas dans la scène réelle.

### 2.1.1 Casques de réalité augmentée

Les casques de réalité augmentée peuvent être rangés dans deux catégories [Carmigniani 2011] : les *Video See-Through Head-Mounted Displays* (VST-HMDs), et les *Optical See-Through Head-Mounted Displays* (OST-HMDs).

**A Les VST-HMDs** Un utilisateur équipé d'un Video See-Through Head-Mounted Display (VST-HMD) ne reçoit pas la lumière de l'environnement réel. Il visionne sur un écran l'image éventuellement modifiée de la scène réelle capturée par une ou deux caméras (Figure 2.1). Le champ de vision disponible est généralement de l'ordre de 100 degrés.

Ce type de dispositif a l'avantage d'offrir une bonne luminosité et un bon contraste. De plus, la composition entre réel et virtuel se fait au niveau d'une image numérique, ce qui permet d'atténuer les décalages temporels et spatiaux entre les éléments virtuels et réels (Figure 2.2). En revanche, la résolution est généralement faible et l'utilisateur peut ressentir un certain inconfort lié à l'affichage indirect de la scène. L'encombrement peut également être prohibitif.



FIGURE 2.1 – Un prototype de VST-HMD issue des travaux de [Steptoe 2014], comprenant un casque de réalité virtuelle et deux caméras.

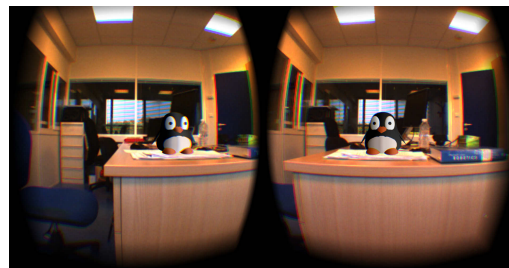


FIGURE 2.2 – Aperçu d'un affichage généré pour un VST-HMD, où le pingouin est l'élément synthétique ajouté sur les images de la scène réelle.

**B Les OST-HMDs** Un Optical See-Through Head-Mounted Display (OST-HMD) par opposition au VST-HMD laisse passer l'information lumineuse de la scène réelle directement vers les yeux de l'utilisateur. Il comprend un système optique, appelé combineur, qui redirige vers l'œil la lumière issue d'un écran ou un laser intégré dans les branches des lunettes. Un exemple d'OST-HMD a été décrit précédemment (Figure 1.15).

Les OST-HMDs ont l'avantage de permettre la visibilité directe de l'environnement par l'utilisateur. En revanche, ils offrent généralement une luminosité et un contraste faible, particulièrement en extérieur. En effet, ce dispositif ajoute de la lumière en plus de celle provenant de la scène (Figure 2.3). En l'état, l'occultation d'un objet réel par un objet virtuel n'est pas possible. Un deuxième inconvénient

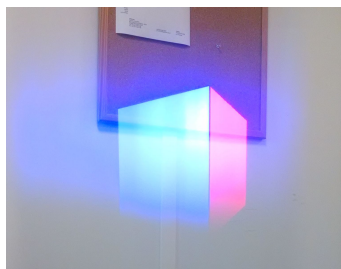


FIGURE 2.3 – Affichage synthétique (cube coloré) généré par le lumus DK50.

des OST-HMDs est de ne permettre l’affichage d’un contenu virtuel que sur un champ de vision réduit.

### 2.1.2 Limitations des dispositifs de réalité augmentée

De nombreuses difficultés freinent encore la commercialisation de casques et lunettes de réalité augmentée. Ces limites peuvent être classées dans trois catégories : celle liée au confort du système d’affichage, à la réalité augmentée elle-même et au caractère embarqué du dispositif total.

#### 2.1.2.1 Limitations des affichages sur des HMDs

L’affichage sur un HMD présente plusieurs défis, détaillés ci-dessous.

**A Résolution** Sur un HMD, la résolution angulaire atteint difficilement celle du système visuel humain (une minute d’arc en moyenne), sauf si l’angle de vue de l’afficheur est de petite taille. Néanmoins, on peut considérer que pour certaines applications, la résolution n’est pas un critère décisif. Un compromis est souvent établi entre résolution et champ de vue. Par exemple, le casque Oculus DK2 propose un champ de vue de l’ordre de  $100^\circ$  et une résolution angulaire d’environ 5 minutes d’arc. Les Google Glasses proposent, quant-à-elles, un champ de vue de  $14^\circ$  et une résolution d’environ 2 minutes d’arc proche de celle du système visuel humain.

**B Champ de vue** Le champ de vue du système d’affichage est souvent trop faible, souvent inférieur à  $40^\circ$  pour un OST-HMD. C’est d’ailleurs une critique récurrente des lunettes Hololens de Microsoft. Deux technologies plus exotiques conduisent à des solutions alternatives, les lunettes *Pinlight Display*<sup>1</sup> de

---

1. Le *Pinlight Display* utilise une matrice transparente de micro-projecteurs devant les yeux. Cette matrice est réalisée par l’association d’un écran SLM (« Spatial Light Modulator ») et d’une plaque transparente recouverte de points lumineux. Pour chaque point lumineux, la lumière issue de ce point est modulée par le SLM avant d’atteindre la pupille.

[Maimone 2014] et le système *Emacula*<sup>2</sup> de Innovega<sup>3</sup>. De plus, si l'application nécessite que les lunettes perçoivent l'environnement (voir section 1.2.3.1), alors les capteurs mis en jeu doivent être dotés d'un large champ de vision.

**C Conflit convergence-accommodation** Il s'agit certainement d'un enjeu phare, puisque ce conflit apporte maux de tête et inconfort à l'utilisateur (voir section 1.1.5). Une pratique trop courante peut même engendrer une anomalie sur le système visuel [ANSES 2014]. Pour éviter ce problème, des alternatives existent [Schowengerdt 2003], [Hua 2014].

**D Luminosité et Contraste** Ces deux caractéristiques sont généralement insuffisantes en extérieur avec les OST-HMDs et posent particulièrement problème en extérieur.

**E Différence de point de vue** Il est important de noter que les capteurs (IMU, caméras, photodiodes) équipant un HMD ont des poses (positions et orientations) propres, qui diffèrent de celles des yeux de l'utilisateur. Une étape d'étalonnage est donc nécessaire, en vue d'évaluer les transformations qui les unissent.

Par ailleurs, la différence de pose entre la(les) caméra(s) filmant la scène et les yeux pose un problème particulier. En effet on ne doit pas ré-utiliser telles quelles les images de l'environnement perçu par la(les) caméra(s) afin de générer du contenu, au risque de provoquer un inconfort pour l'utilisateur, car la différence de points de vue altère sa perception de l'environnement et de son mouvement. Lorsque la tête bouge comme sur la figure 2.4, le porteur du casque observe un mouvement de la scène correspondant aux points de vue des caméras, plutôt que de ses yeux.

Pour synthétiser une image adaptée au point de vue de l'utilisateur, un *système visuel étendu* est donc défini à partir de l'afficheur du HMD et des yeux du porteur et son modèle de projection doit être estimé, en vue d'associer à un point sur l'écran de l'afficheur la droite de projection correspondante dans la scène. Sur cette base, les images sur la rétine de l'utilisateur d'un point de la scène et du point correspondant de l'afficheur doivent être convenablement superposées. Cependant, dans la pratique, les yeux du porteur ne sont pas fixes par rapport au casque. Il peut donc être intéressant de compléter le modèle du système visuel étendu de façon à prendre en compte des variables telles que la position du centre optique de l'œil par rapport aux lunettes, qui peut être estimée au moyen d'un oculomètre.

---

2. Le système *Emacula* comprend une lentille et une paire de lunettes équipé d'un système d'affichage. La lentille a la particularité de rendre possible la mise au point de l'œil sur des distances très courtes telle que la distance lunette-œil. La lentille est composée de deux filtres, un disque au centre de la lentille et une couronne en périphérie. Le filtre périphérique focalise la lumière de l'environnement sur la rétine et ne focalise pas l'affichage des lunettes. Le filtre centrale ne focalise pas la lumière de l'environnement mais focalise celle provenant des lunettes. Ainsi l'œil est toujours capable de focaliser à deux distances, lui permettant de voir à la fois la scène réelle et le contenu virtuel des lunettes.

3. <http://www.emacula.io/>

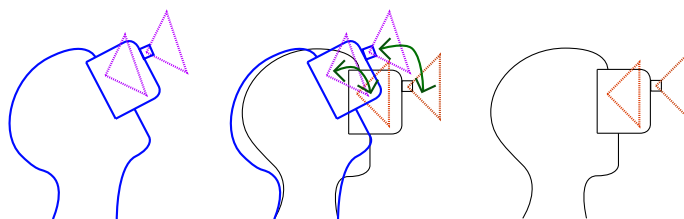


FIGURE 2.4 – Schéma explicatif de l'inconfort causé par la différence de point de vue entre les yeux de l'utilisateur et les caméras. Les schémas de droite et de gauche montrent deux postures différentes de la tête d'un utilisateur, et celui du centre superpose les deux autres images. Les champs de vision des yeux de l'utilisateur et des caméras sont représentés par des triangles. Sur le dessin central, les flèches vertes montrent que les mouvements des centres optiques des caméras du VST-HMD et des yeux de l'utilisateur sans VST-HMD diffèrent.

### 2.1.2.2 Contraintes inhérentes à la réalité augmentée

Certaines contraintes relèvent de la réalité augmentée en général, quel que soit le support (casque, lunettes, tablette ou téléphone, ).

**A Temps réel, latence et fréquence d'affichage** Suivant l'application (cf. la section 1.2.3.1), les HMDs sont soumis à des contraintes temps réel plus ou moins sévères. Comme la tête de l'utilisateur peut être en mouvement, le contenu affiché doit continuellement s'adapter aux nouveaux points de vue des yeux. Pour les OST-HMDs, la latence est clairement perceptible lorsque les éléments virtuels sont décalés par rapport à la scène réelle. En revanche sur les VST-HMDs, l'image affichée de l'environnement réel et l'information augmentée sont tous deux retardées, ce qui génère un inconfort dû à l'incompatibilité entre le mouvement perçu par le système visuel et celui perçu par l'oreille interne. Ce phénomène se nomme « cinétose », ou « motion sickness », ou « VR-induced sickness effect » [Le Corre 2016].

En conséquence, la latence entre le mouvement du casque et l'affichage de l'image doit être limitée. Selon Adelstein *et al.* [Adelstein 2003] elle ne doit pas dépasser 17 ms. Concernant la fréquence de rafraîchissement de l'affichage, voici quelques chiffres applicables aux VST-HMDs et casques de réalité virtuelle. Cette fréquence dépend bien sûr de l'application visée. Pour l'affichage d'environnements virtuels sans mouvement rapide de l'utilisateur, Chen et Thropp [Chen 2007] suggèrent une fréquence minimum de 17.5 Hz. En revanche, lorsque des mouvements rapides sont nécessaires, comme c'est le cas pour les jeux sur HMD, la fréquence doit être plus élevée. Ainsi, la société Oculus conseille un taux de rafraîchissement d'au moins 70 Hz. D'autre part, Claypool *et al.* [Claypool 2006] montrent que la fréquence impacte les performances des joueurs sur HMD.

**B Estimation de la pose du dispositif d'affichage** Lorsque une superposition fine de l'information augmentée avec des éléments de la scène est souhaitée, il est préférable de localiser la lunette. Ainsi, sur la base de l'estimation de sa pose

par rapport à un repère lié à la scène, cette superposition peut être maintenue en dépit du déplacement de l'utilisateur. Pour estimer cette pose, plusieurs solutions existent. Certaines consistent à munir la scène d'éléments dédiés, tels que des tags ou marqueurs de référence, ou bien à installer des caméras filmant le porteur. Cependant, cette option restreint l'usage du HMD à des environnements instrumentés. Suivant l'application recherchée pour le casque, il peut être nécessaire de le rendre fonctionnel pour des milieux inconnus en ayant recours à des techniques permettant de cartographier la scène et, concomitamment, de localiser la lunette dans la carte obtenue. Cette problématique est détaillée dans la section 2.3.

### 2.1.2.3 Contraintes d'embarquabilité d'un HMD

Les autres contraintes actuelles de la réalité augmentée sur HMD relèvent des problématiques classiques des systèmes embarqués grand public.

- Le système doit être léger, car il doit généralement être porté sur la tête pendant de nombreuses heures.
- Il doit disposer d'une interface intuitive et d'une bonne ergonomie.
- L'apparence (discrétion, design) est également un facteur déterminant.

### 2.1.3 Conclusion

Les composants d'affichage, ainsi que leurs limitations et contraintes, ont été présentés au cours de cette section, ce qui nous a permis d'identifier les points cruciaux, nécessitant d'être pris en compte lors de la conception d'un dispositif d'affichage. Sur la base de cette étude, nous avons mis au point deux dispositifs d'affichage (un VST-HMD et un OST-HMD) détaillés au chapitre 4.

## 2.2 Perception de l'environnement

Les problématiques traitées dans cette section ne concernent que la perception visuelle de l'environnement pour acquérir des informations photométriques ou géométriques. Dans la majorité des cas évoqués, des données photométriques sont nécessaires pour effectuer certaines tâches mentionnées dans la section 1.2.1, comme par exemple l'analyse de la luminosité de la scène pour déduire l'obscurcissement des verres, ou encore la détection de personne ou d'objet de la scène pour des fonctions avancées de réalité augmentée. Pour acquérir ces informations, nous nous sommes intéressés à deux capteurs différents : les photodiodes et les caméras. Les photodiodes fournissent une information de luminosité sur un angle de vue défini et dans un spectre de longueur d'onde donné, qui peut par exemple correspondre à la vision humaine de jour. Elles ont la particularité de proposer une dynamique plus étendue et de consommer très peu. Par exemple, le composant MAX44009 intégrant une photodiode ne consomme qu'environ  $5 \mu W$  (pour une tension de  $3 V$  et un courant de  $1,6 \mu A$  et code la luminosité sur 22 bits de 0,045 à 188000 lux,

tandis que la caméra UI-3241LE-C-HQ de IDS consomme de 1,3 à 1,5  $W$  et fournit une information RGB pour chaque pixel où chaque canal est codé sur 8 *bits*.

Seul l'utilisation de caméras sera abordée dans cette section, dont la première partie traite des modélisations des caméras et la seconde de la perception 3D.

### 2.2.1 Modélisation des caméras

Définir un modèle de projection d'une caméra (le capteur et son objectif) permet de connaître pour un point 3D de la scène sa projection sur l'image. On parle alors de modèle direct. Pour reconstruire le rayon de projection associé à un pixel, on a recours à un modèle inverse calculé à partir du modèle direct.

#### 2.2.1.1 Modèle trou d'épingle

Le modèle de caméra le plus répandu et utilisé en vision par ordinateur est le modèle trou d'épingle ("pinhole"), que l'on complète généralement par une modélisation des distorsions dues aux lentilles de l'objectif. Ce modèle correspond à un "filtre" sélectionnant les rayons lumineux issus de la scène passant par un point commun appelé centre optique et se focalisant sur le capteur de la caméra. Il est décrit par les équations 2.1 et 2.2, où :  $(x, y, z)$  désigne le vecteur des coordonnées métriques, dans un repère  $R_c$  lié à la caméra, d'un point  $P_c$  quelconque de la scène ;  $(u, v)$  désigne le vecteur des coordonnées pixel, dans le repère  $R_i$  de l'image, de sa projection  $p$  sur le plan image ;  $f$  désigne la focale ;  $k_u$  et  $k_v$  définissent la densité pixellique suivant les axes  $U_i$  et  $V_i$  en pixels par mètre ;  $s$  encode l'angle entre les axes  $X$  et  $Y$  du capteur ;  $(p_u, p_v)$  désignent les coordonnées pixelliques du point principal défini comme l'intersection entre le plan image et l'axe optique ;  $(u_h, v_h, w_h)$  sont les coordonnées homogènes. Les paramètres  $(f, k_u, k_v, s, p_u, p_v)$  du modèle trou d'épingle sont appelés paramètres intrinsèques du modèle pinhole. Lorsque le point de la scène est exprimé dans un repère  $R_s$  lié à la scène, des paramètres extrinsèques sont alors ajoutés pour décrire la rotation et la translation entre  $R_c$  et  $R_s$ . La figure 2.5 illustre le principe du modèle trou d'épingle.

$$\begin{bmatrix} u_h \\ v_h \\ w_h \end{bmatrix} = \begin{bmatrix} k_u f & s & p_u \\ 0 & k_v f & p_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.1)$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_h/w_h \\ v_h/w_h \end{bmatrix} \quad (2.2)$$

#### 2.2.1.2 Modèle trou d'épingle avec distorsion : "Plump bob"

Le modèle trou d'épingle avec distorsion couramment utilisé est celui proposé par [Heikkila 1997]. Dans ce modèle, pour modéliser les distorsions, on considère dans un premier temps la projection du point 3D  $P_c$  sur le plan image normalisée,

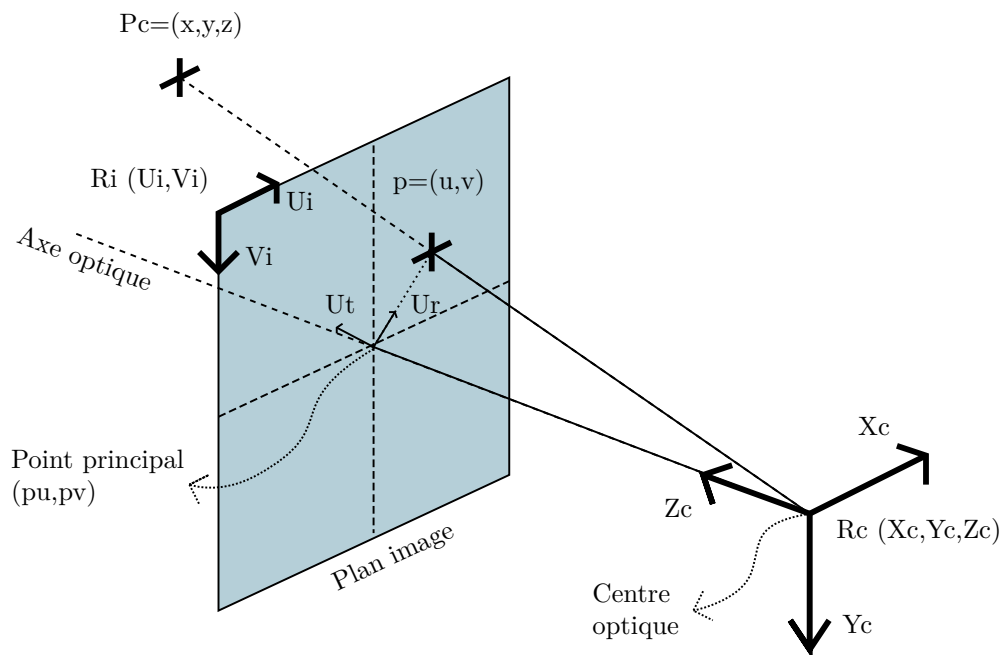


FIGURE 2.5 – Schéma illustrant le modèle trou d'épingle. Un point  $P_c$  de coordonnées  $(x, y, z)$  exprimé dans  $R_c$  le repère de la caméra est projeté sur l'image en un point  $p$  de coordonnées  $(u, v)$  dans le repère  $R_i$  du plan image.

c'est à dire l'image théorique pour une focale de 1 unité métrique (équation 2.3). Sa projection a pour coordonnées métriques  $(x_n, y_n)$ .

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x/z \\ y/z \end{bmatrix} \quad (2.3)$$

Ensuite, le modèle de distorsion est appliqué au point normalisé selon l'équation 2.4 pour déduire le point  $(x_d, y_d)$  correspondant. Le pixel sur l'image réelle est alors déduit des équations 2.1 et 2.2 appliquées à  $(x_d, y_d, 1)$ . Dans ce modèle, on distingue deux distorsions : radiales et tangentielles. La distorsion radiale encode la composante de la distorsion exprimée sur l'axe radial  $U_r(P_c)$  passant par le point principal et le point  $p$  correspondant à la projection de  $P$  par le modèle trou d'épingle sans distorsion. La distorsion tangentielle désigne la distorsion suivant l'axe  $U_t(P_c)$  perpendiculaire à  $U_r(P_c)$ . Elle est due à un décentrement des lentilles le long de l'axe optique. Dans l'équation 2.4, les paramètres  $(k_1, k_2, k_5)$  et  $(k_3, k_4)$  correspondent respectivement aux distorsions radiales et tangentielles. Ces paramètres sont également des paramètres intrinsèques, car ils ne dépendent pas de la pose de la caméra relativement à la scène.

$$\begin{aligned} \begin{bmatrix} x_d \\ y_d \end{bmatrix} &= (1 + k_1 r^2 + k_2 r^4 + k_5 r^6) \begin{bmatrix} x_n \\ y_n \end{bmatrix} + dx \\ \text{où } r^2 &= x_n^2 + y_n^2 \\ \text{et } dx &= \begin{bmatrix} 2k_3 x_n y_n + k_4 (r^2 + 2x_n^2) \\ k_3 (r^2 + 2y_n^2) + 2k_4 x_n y_n \end{bmatrix} \end{aligned} \quad (2.4)$$

### 2.2.1.3 Étalonnage

Pour estimer les paramètres du modèle pour une caméra donnée, il est nécessaire de procéder à son étalonnage. Pour ce faire, la procédure la plus courante consiste à prendre plusieurs images d'une mire (échiquier,...) située en différentes positions et orientations. Par la suite, l'algorithme d'étalonnage extrait la position 2D des coins<sup>4</sup> de l'échiquier dans l'image, puis estime les paramètres intrinsèques du modèle de la caméra  $((f, k_u, k_v, s, p_u, p_v, [k_1, k_2, k_3, k_4, k_5])$  pour le modèle de [Heikkila 1997]), et les paramètres extrinsèques qui expriment les changements de pose (position et orientation) entre le repère caméra et les repères liés à la mire pour chacune des images.

Cette procédure se déroule en plusieurs étapes. Après une estimation peu précise des paramètres intrinsèques, les paramètres extrinsèques sont estimés une première fois en estimant l'homographie<sup>5</sup> entre le plan de la mire et sa projection dans l'image [Zhang 2000]. Ensuite, l'ensemble des paramètres est raffiné par des techniques

4. Les coins sont les points d'intersection entre les carrés blanc et noirs de la mire.

5. En vision par ordinateur, dans un espace à trois dimensions, une homographie (ou transformation projective) est une transformation d'un plan  $P_1$  vers un autre  $P_2$ , qui conserve les droites. Elle s'écrit sous la forme d'une matrice  $H$  3x3, telle que pour  $x_1$  appartenant à  $P_1$ , on ait  $x_2 = Hx_1$  où  $x_2$  appartient à  $P_2$ . En d'autres termes, toutes images projectives d'un plan dans une scène sont

de moindres carrés pour la résolution de problème non linéaire. Cette technique cherche à minimiser l'erreur de reprojection<sup>6</sup> entre les coins extraits de l'image et la projection des points 3D correspondants.

Les outils d'étalonnage utilisés pour la thèse sont OpenCV<sup>7</sup>, qui reprennent le modèle de [Heikkila 1997].

Dans la pratique, à partir de l'image brute délivrée par la caméra, il est d'usage de construire une image synthétique dite sans distorsion sur laquelle se basent ensuite les algorithmes de vision par ordinateur. Le modèle de caméra pinhole seul pouvant alors être utilisé.

#### 2.2.1.4 Modèle pour caméra à large champ de vue

Parmi les caméras large champ de vue, il y a des caméras équipées de lentille fisheye et les systèmes catadioptriques.

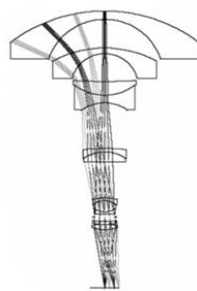


FIGURE 2.6 – Schéma illustrant les assemblages de lentilles dans les optiques fisheye (Extrait de [Scaramuzza 2013])



FIGURE 2.7 – Image d'un échiquier obtenue à partir d'une caméra fisheye.

**A Modèle fisheye** Les optiques fisheye sont constituées d'un assemblage de lentilles (Figure 2.6) leur conférant un large champ pouvant dépasser 180 degrés. Les caméras équipées de ces lentilles et dotées d'un champ de vision supérieur à 140 degrés sont généralement mal modélisées par un modèle trou d'épingle même en incluant des distorsions. Ce modèle ne convient pas, car la distorsion radiale est trop importante comme le montre cette figure 2.7. Il est donc nécessaire de définir un modèle capable de capturer ce phénomène. Plusieurs modélisations de caméra fisheye ont été proposés, [Hughes 2008] en expose un modèle de projection unifié axial.

liées par une homographie.

6. L'erreur de reprojection d'un point 3D  $P$  est la distance en pixel entre un point 2D  $p$  détecté sur l'image et la reprojection  $p'$  du point 3D  $P$  correspondant à l'aide du modèle de projection estimé.

7. <https://opencv.org/>

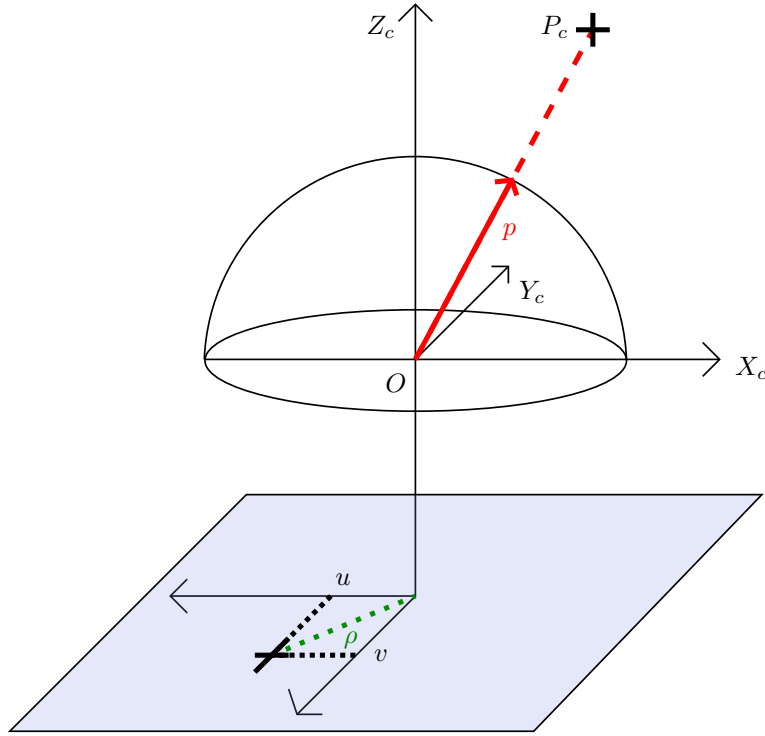


FIGURE 2.8 – Illustration du modèle Ocam de [Scaramuzza 2006a]. Le point  $P_c$  de la scène est projeté en  $(u, v)$  sur le plan image normalisé.

**B Modèle génériques pour caméras omnidirectionnelles** D'autres modèles plus génériques, tels que celui de [Scaramuzza 2006a], ont aussi été proposés. Ces modèles sont compatibles avec les caméras catadioptriques et les caméras fisheye.

[Scaramuzza 2006a] définit un modèle inverse, où une fonction polynomiale  $f$  s'applique sur les coordonnées  $(u, v)$  du point dans l'image normalisée (2.5 et 2.6).  $p$  représente le vecteur directeur passant par le point 3D  $P_c$  et le centre optique  $O$  de la caméra.

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} u \\ v \\ f(\rho) = f(\sqrt{u^2 + v^2}) \end{bmatrix} \quad (2.5)$$

$$f(\rho) = a_1 + a_2\rho + a_3\rho^2 + a_4\rho^3 + a_5\rho^4 + \dots \quad (2.6)$$

Dans le modèle de [Scaramuzza 2006a], les défauts d'alignement entre le capteur et l'objectif de la caméra (lentilles ou miroirs) sont modélisés par la transformation affine 2D décrite dans l'équation (2.7).  $(u', v')$  représentent les coordonnées avec distorsion sur l'image réelle et  $(u, v)$  sont les coordonnées idéales sur l'image normalisée.

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} c & d \\ e & 1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} x'_c \\ y'_c \end{bmatrix} \quad (2.7)$$

### 2.2.1.5 Conclusion

Cette section a présenté plusieurs modèles de caméras, qui ont été utilisés par la suite sur nos prototypes présentés dans le chapitre 4.

Pour l'étalonnage des caméras fisheye du VST-HMD, nous avons tout d'abord opté pour la suite logicielle de **OpenCV**<sup>8</sup>, qui se base sur un modèle proche de celui de [Kannala 2006], pour lequel l'implémentation sur **OpenCV** n'inclue pas tous les paramètres modélisant les distorsions.

De plus, leur algorithme d'étalonnage diffère lui aussi des travaux de [Kannala 2006], en reprenant la même procédure d'étalonnage que celle réservée au modèle trou d'épingle, au sens où il détermine une première initialisation des paramètres extrinsèques par calcul d'homographie [Zhang 2000], sans initialiser les paramètres de distorsion. Or, sur une image issue d'une caméra fisheye, les mires sont fortement déformées à cause de la distorsion (Figure 2.7). C'est pourquoi, lors de nos tests, l'étape d'estimation de paramètres extrinsèques ne parvenait généralement pas à converger vers une solution. De plus, même en triant les images pour sélectionner celles qui réussissaient cette étape, le résultat de l'étalonnage ne demeurerait pas concluant.

Comme nos premiers essais avec **OpenCV** n'ont pas été fructueux, nous avons porté notre second choix sur la boîte à outils *OCamCalib* de Scaramuzza sur MATLAB ([Scaramuzza 2013] et [Scaramuzza 2006b]), qui a fourni des résultats plus corrects (plus faible erreur de reprojection sur toute la surface du capteur) pour l'étalonnage d'une caméra.

## 2.2.2 Perception 3D

En plus des informations photométriques, il peut être intéressant d'acquérir des données géométriques sur la scène, notamment avec des images de profondeur où chaque pixel est associé à une distance entre le point 3D observé et un plan fictif parallèle au plan du capteur et passant par le centre optique. On parle aussi de "z-map" au lieu de carte de profondeur.

### 2.2.2.1 Capteurs passifs et actifs

Il existe deux catégories de capteurs, ceux actifs qui projettent de la lumière sur la scène et ceux passifs qui n'utilisent que la lumière déjà présente dans l'environnement. Ces deux technologies ont chacune leurs avantages et désavantages, que nous rappelons ci-dessous.

8. [https://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html](https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html)

**A Les caractéristiques des systèmes actifs** Les systèmes actifs basés sur la projection de lumière présentent les caractéristiques suivantes :

- Ils fournissent généralement des informations 3D à plus haute cadence et plus faible latence que les systèmes passifs. Néanmoins, il est courant de filtrer ces images pour réduire le bruit sur la mesure de distance.
- Ils fonctionnent indépendamment de la texture de la scène, qu'elle soit unie ou très texturée.
- La mesure peut être perturbée par la lumière environnante et particulièrement en extérieur. Ces capteurs sont donc généralement réservés pour une utilisation en intérieur.
- Les distances acquises sont restreintes à un intervalle. Par exemple, le capteur temps de vol Kinect2 de Microsoft, qui est un capteur RGB-D<sup>9</sup>, détecte les surfaces entre 0,8 et 4,2 mètres.
- Les largeurs des champs de vue des capteurs de profondeur disponibles sont inférieurs à 100 degrés, ce qui ne permet pas de couvrir la vision humaine avec un seul capteur.

Les références [Horaud 2016] et [Breuer 2014] proposent une étude comparative entre différents capteurs de profondeur et listent leur caractéristiques (résolution, intervalle de distance, champ de vue, fréquence d'acquisition).

**B Les caractéristiques des systèmes passifs** Les systèmes passifs s'appuient sur l'acquisition de plusieurs images de la scène (pour des points de vue ou des mises au point différentes) sans utiliser d'éclairage additionnel. Ces techniques présentent les avantages et inconvénients suivants :

- Ils peuvent fonctionner en intérieur et extérieur, mais sont tributaire d'un éclairage suffisant.
- La profondeur ne peut être calculée que sur des portions de la scène observées sur chaque image. De plus pour les bancs de stéréovision (des systèmes équipés de deux caméras, partageant une partie de leur champ de vision), le champ de vision stéréoscopique est défini par le champ de vision commun entre les caméras, jusqu'à une certaine distance à partir de laquelle les différences de points de vue deviennent négligeables. Ainsi, l'écartement des caméras et leurs focales influent sur la plage des profondeurs pouvant être estimées. Pour sélectionner cette plage, les caméras et leur agencement doivent être judicieusement choisis.
- Ils tendent à commettre des erreurs sur les zones de la scène peu texturées ou lors de motifs répétitifs.

---

9. Les capteurs actifs fournissent souvent une image couleur et l'image de profondeur associée. On parle alors de capteur RGB-D.

- Les calculs nécessaires sont lourds et doivent être parallélisés sur des architectures matérielles particulières (par exemple sur GPU ou FPGA<sup>10</sup>) pour espérer atteindre des cadences et résolutions élevées. Le temps de calcul dépend étroitement de la résolution de l'image de profondeur et de sa qualité. Pour les bancs de stéréovision, un comparatif de différents algorithmes est disponible en ligne<sup>11</sup>.

### 2.2.2.2 Perception 3D par stéréovision

**A Principe** Un banc de stéréovision est un système constitué de deux caméras  $C_1$  et  $C_2$ , synchronisées et solidarisées sur un même support, qui partagent une partie de leurs champs de vision. La portion de champ de vue commune est appelée vision binoculaire et le reste vision monoculaire. L'emplacement de cette perception commune dépend de l'assemblage des deux caméras, plus précisément de leur espacement  $b$ , de leurs focales  $f$  et des dimensions des capteurs. Pour la suite, nous considérons deux caméras  $C_1$  et  $C_2$  de centres optiques  $O_1$  et  $O_2$ , dont on acquiert les images  $I_1$  et  $I_2$ .

L'objectif est de construire une carte de profondeur qui associe une distance dans la scène à chaque pixel de l'image de gauche  $I_1$  (ou de droite  $I_2$ ) appartenant au champ de vision binoculaire (pour notre explication, les distance sont calculée pour l'image de gauche). Un pixel  $p_1$  peut être associé à un point 3D  $P$  de la scène projeté sur  $I_1$ . La position 3D de  $P$  est calculable par triangulation si on connaît sa projection  $p_2$  sur l'image de droite  $I_2$  et le changement de pose entre les deux caméras, appelé interpose.  $p_1$  et  $p_2$  sont alors appelés homologues, car ils sont les images d'un même point 3D de la scène.

Une étape essentielle, pour le calcul de la position de  $P$ , est de chercher la projection  $p_2$  de  $P$  sur  $I_2$  par comparaison des pixels situés dans une sous-fenêtre centrée sur  $p_1$  dans  $I_1$  avec les pixels situés dans une sous-fenêtre mobile dans  $I_2$ , laquelle est supposée centrée sur  $p_2$  lors du meilleur appariement. Or,  $p_2$  se situe le long d'une ligne dans  $I_2$ , appelée ligne épipolaire, correspondant à la projection sur l'image de droite du rayon incident passant par  $P$  et  $O_1$  (Figure 2.9). En tenant compte de cette contrainte géométrique, la recherche de  $p_2$  peut s'effectuer sur la ligne épipolaire plutôt que sur l'image entière, ce qui réduit le coût calculatoire et la probabilité de faux appariements.

Cette recherche peut encore être optimisée en considérant deux nouvelles caméras de même focale et de même résolution, orientées de telle sorte que leur lignes épipolaires soient parallèles à l'axe  $U$  (axe horizontal de l'image) et à la ligne passant par  $O_1$  et  $O_2$  (Figure 2.10). Connaissant l'écart pixellique  $d$  entre les deux projections  $p_1$  et  $p_2$ , appelé disparité, on en déduit la profondeur en unité métrique  $p$  par

10. Les FPGA (Field-Programmable Gate Array) sont des circuits intégrés reprogrammables, c'est-à-dire que l'on peut les configurer afin d'exécuter une application donnée. Il est donc possible de les configurer pour des objectifs de performance temporelle.

11. <http://vision.middlebury.edu/stereo/eval3/> et [http://www.cvlibs.net/datasets/kitti/eval\\_stereo\\_flow.php?benchmark=stereo](http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php?benchmark=stereo)

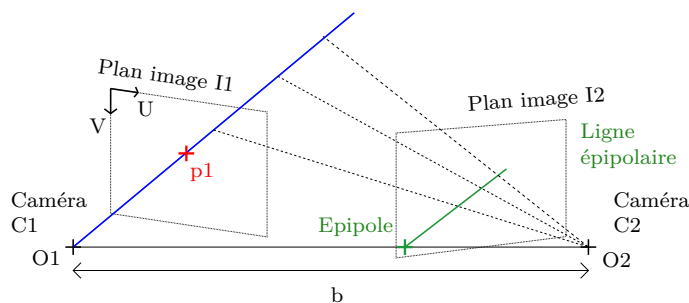


FIGURE 2.9 – Illustration d’un banc stéréoscopique constitué de deux caméras  $C_1$  et  $C_2$  de centres optiques  $O_1$  et  $O_2$ . Ces deux centres optiques sont distants de  $b$ . Le rayon de projection (en bleu) passant par  $O_1$  et le pixel  $p_1$  (en rouge) de l’image  $I_1$  de  $C_1$  est projeté sur l’image  $I_2$  de  $C_2$  en la ligne épipolaire (en vert).

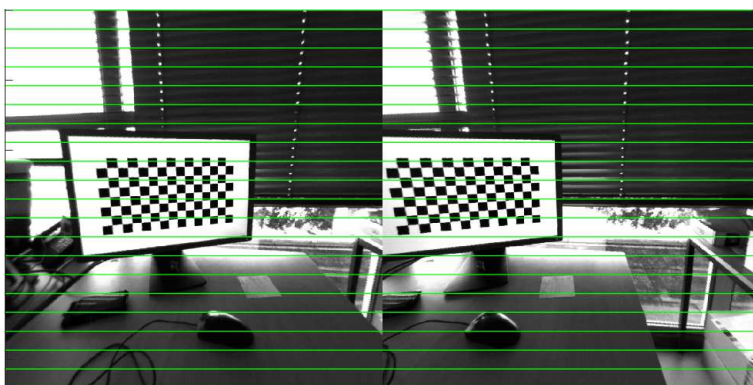


FIGURE 2.10 – Exemple d’images rectifiées issues d’un banc stéréoscopique. Les lignes vertes représentent des lignes épipolaires.

l’équation  $p = f k_u b / d$ , dans laquelle  $f$  est la focale en mètre ;  $k_u$  est la densité pixelique en pixels par mètre ;  $b$  (“baseline”) est la distance en unité métrique séparant les deux caméras. Pour se ramener à ce cas idéal où les lignes épipolaires sont parallèles à l’axe  $U$  de l’image, il suffit d’effectuer une transformation dite « rectification épipolaire » sur les deux caméras, en appliquant une homographie aux deux images, afin de les projeter sur un même plan (le calcul est détaillé par [Fusiello 2000]). Sur cette paire d’images synthétiques, les homologues sont situés sur une même ligne et la recherche d’appariement se fait plus rapidement, les pixels étant stockés de manière contiguë en mémoire. De plus les lignes peuvent être traitées quasiment de manière indépendante.

Des états de l’art des techniques de mise en correspondance des pixels d’un système stéréoscopique sont proposés dans [Hirschmuller 2009] et [Chambon 2011]. La fonction de coût nommée CENSUS se démarque par ses bonnes performances [Hirschmuller 2009], qui chutent néanmoins en présence de bruit de mesure élevé sur les données photométriques de l’image. La référence [Chambon 2011] précise aussi que cette fonction fournit de bons résultats dans les zones peu texturées. En

revanche, elle n'est ni la plus rapide ni la plus performante pour les zones très texturées. OpenCV, via `opencv-contrib`, propose plusieurs implémentations sur CPU de fonctions de coût. Ces outils ont été testés et évalués lors de la thèse, ce qui nous a permis de constater que CENSUS semble être un bon compromis, compte tenu de la qualité de l'image de profondeur obtenue et du temps de calcul. Néanmoins la durée du calcul sur CPU est élevée avec plus d'une seconde et ne s'accorde donc pas avec les contraintes temps réel des applications de réalité augmentée.

Comme il a été expliqué plus tôt, les images d'un banc stéréoscopique peuvent être rectifiées épipolairement pour accélérer la recherche des points homologues. Le calcul de cette transformation requiert de connaître les paramètres intrinsèques du modèle des deux caméras et leur pose relative, que l'on estime lors de l'étalonnage du banc. A l'instar d'un étalonnage monoculaire, la procédure consiste à acquérir plusieurs images avec les deux caméras, de points définis sur un plan, généralement les coins d'un damier. Ensuite, sur la base de ces points, on calcule une première estimation des paramètres intrinsèques et extrinsèques par un étalonnage monoculaire. La pose relative est initialisée à partir des paramètres extrinsèques des deux étalonnages. Puis, on raffine l'ensemble des données estimées (les paramètres des modèles de projection des deux caméras, leur pose relative et les poses des damiers) en minimisant conjointement l'erreur de projection sur les images des deux caméras.

**B Calcul de cartes de disparité** La référence [Scharstein 2002] distingue deux familles principales d'algorithmes pour le calcul de cartes de disparité : les méthodes locales et globales.

**Méthodes locales** Les méthodes locales suivent les étapes suivantes :

- calcul de la fonction de coût pour l'ensemble des couples de pixels gauche-droite possibles le long de la ligne épipolaire jusqu'à une disparité limite ;
- calcul du coût par agrégation, pour chaque couple de pixels, sur une fenêtre de  $n \times n$  pixels ;
- calcul de la carte de disparité en cherchant pour chaque pixel gauche, le pixel droite présentant la plus grande similitude définie par la fonction de coût.

Ces techniques sont dites locales car la valeur de disparité choisie dépend uniquement des données internes aux fenêtres de calculs. Elles ne tiennent pas compte des correspondances gauche-droite trouvées sur les pixels voisins. En conséquence, un pixel de l'image de droite peut être associé à plusieurs pixels de l'image de gauche (violation de la contrainte d'unicité). De plus, il est possible que des appariements ne respectent pas la contrainte d'ordre.

**Méthodes globales** Les méthodes globales en revanche ajoutent une étape d'optimisation visant à minimiser une fonction de coût globale combinant un terme lié à la différence d'apparence et un terme de lissage considérant les valeurs de disparité à proximité. Comme pour la méthode locale, le terme d'apparence peut être calculé en se basant sur des sous-fenêtres centrées sur les pixels concernés. La

disparité de chaque pixel de l'image de gauche est déduite de l'étape d'optimisation sur la fonction de coût globale. Ainsi la carte de disparité obtenue est lissée, ce qui améliore sa qualité sur les zones unies. En revanche, les frontières entre les objets conduisant à des discontinuités de disparité ont tendance à être mal estimées. Pour éviter ce phénomène, certains termes de lissage sont définis de manière à conserver les bordures [Scharstein 2002].

Ensuite, quelle que soit la méthode, des étapes de raffinement sont mises en place. Elles peuvent, par exemple, lisser la carte de disparité par un filtre bilatéral, ou estimer des disparités à une résolution sous-pixelles.

La référence [Bleyer 2013] présente un état de l'art plus récent, dans lequel elle mentionne la méthode locale avec fenêtre de poids adaptatif nommée ASW (adaptive support weight) qui donne de meilleurs résultats aux frontières d'objets [Yoon 2006], et celle avec fenêtre oblique qui tient compte des surfaces non fronto-parallèles à la caméra [Gallup 2007]. Néanmoins malgré ces nouveautés, les performances des méthodes locales sur de grandes surfaces faiblement texturées restent inférieures à celles des méthodes globales.

Peu d'implémentations d'algorithmes de correspondance stéréoscopique (*stereo-matching*) sont disponibles en open-source. Parmi elles, on distingue le semi-global block matching de OpenCV inspiré du semi-global matching (SGM) de [Hirschmuller 2008]. Cette implémentation consiste à comparer les pixels droite et gauche par blocs (sur une fenêtre de pixels) avec la mesure proposée par Birchfield et Tomasi [Birchfield 1998], puis à appliquer une optimisation sur une fonction de coût globale qui inclut un terme de lissage particulier tenant compte des disparités des pixels plusieurs directions autour du pixel (8 dans la version originale [Hirschmuller 2008]). Cette technique est robuste au manque de texture, mais les bordures des objets sur la carte de disparité sont assez grossières.

Certaines étapes des algorithmes peuvent être parallélisées sur des architectures matérielles de type GPU ou FPGA. Par exemple, [Mei 2011] propose un stéréo matching SGM avec le descripteur AD CENSUS sur GPU. Pour l'image des cônes du dataset de Middlebury<sup>12</sup> de 450 x 375 pixels et de disparité maximale de 60 pixels, le calcul de la carte de disparité prend moins de 0.1 seconde sur GPU et 15 secondes sur CPU. Plus récemment, [Wang 2015b] atteint, avec un algorithme similaire sur FPGA, 67 Hertz pour une image de 1024 x 768 pixels et une disparité maximale de 96 pixels.

**C Stéréovision sur une séquence d'images** Sur un flux vidéo, la stéréovision peut tirer avantage des images stéréo de l'instant précédent pour réduire l'effet du bruit et l'incertitude sur l'estimation des cartes de profondeur. D'autre part, considérer l'aspect temporel permet de réduire les artefacts, c'est-à-dire les zones de pixels erronées qui apparaissent et disparaissent sur une séquence d'images de profondeur. Si la scène et la caméra sont stationnaires, alors les données de disparité

12. <http://vision.middlebury.edu/stereo/data/>

anciennement estimées peuvent directement être réutilisées. En revanche, si la scène est dynamique (c'est à dire des objets de la scène se déplacent) ou si la caméra est mobile, alors la carte de disparité subira des variations au cours du temps. Les anciennes valeurs de disparité doivent alors subir différents traitements pour tenir compte des mouvements et éventuellement être remises en cause.

Une de ces techniques se nomme la stéréovision spatio-temporelle et consiste à évaluer la meilleure correspondance pixel à pixel suivant trois dimensions, incluant une dimension temporelle en plus des deux dimensions spatiale de l'image. Ainsi, par ce procédé, lorsque plusieurs associations sont possibles, on discrimine grâce à la dimension temporelle (Figure 2.11). Cette méthode fait l'hypothèse que la disparité au sein de l'image n'évolue pas pendant le laps de temps concerné dans la fenêtre de coût. Il est donc préférable de sélectionner peu d'images, comme le proposent [Jain 2014] avec seulement trois paires d'images. Les auteurs précisent qu'utiliser davantage d'images n'améliore pas le résultat.

Pour lisser la carte de disparité tout en préservant les contours des objets, les auteurs de [Richardt 2010] ont adapté la méthode de [Yoon 2006] au cas spatio-temporel en la modifiant pour une exécution plus rapide et une empreinte mémoire moins importante. Ils atteignent ainsi 75 millisecondes sur les images de cônes du dataset de Middlebury ( $450 \times 375$  pixels et une disparité maximale 60 pixels).

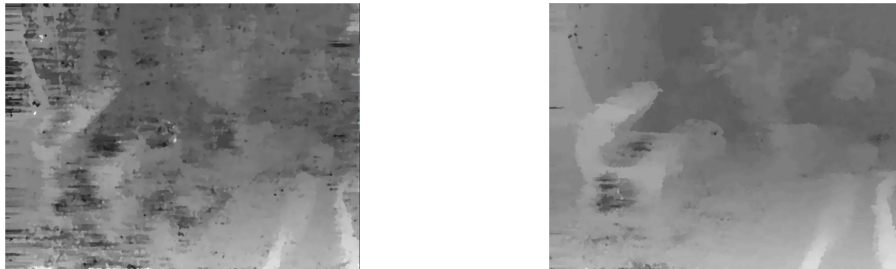


FIGURE 2.11 – Deux exemples de cartes de disparité de la même scène obtenues par un algorithme ne considérant pas d'aspect temporel (gauche) et un second intégrant une dimension temporelle dans l'agrégation de coût (droite) (Extrait de [Kowalczyk 2013])

D'autres idées ont été proposées pour tirer parti du flux vidéo. L'une d'elles est d'accélérer l'étape de recherche de pixels homologues sur les images gauche et droite en restreignant l'intervalle de recherche. Pour cela, le nouvel intervalle, défini pour le pixel  $p$  de l'image gauche de l'instant  $t$ , est centré sur la position du précédent pixel homologue à  $p$  extrait de l'image droite à l'instant  $t - 1$ . Cette méthode fait l'hypothèse que la profondeur de la scène perçue par la caméra pour un pixel évolue peu au cours du temps. Elle s'appuie sur une méthode de stéréovision locale, où l'on définit une disparité minimale et maximale. La carte de profondeur doit être initialisée par des méthodes traditionnelles de stéréovision. Cette technique a l'avantage d'accélérer l'estimation des disparités, car l'intervalle de recherche est réduit. Néanmoins, les erreurs d'association des pixels gauche-droite s'accumulent peu à peu sur les cartes de disparité calculées, en les propageant d'une image à

l'autre, à cause de la restriction de l'intervalle de recherche. Un mouvement de la caméra ou d'un objet de la scène induit aussi en erreur cet algorithme, qui suppose que la scène et la caméra sont stationnaires. Une alternative simple, pour tenir compte de la mobilité de la scène et de la caméra, est de réinitialiser régulièrement la carte de disparité. Pour s'adapter aux mouvements de la scène, [Mun 2016] propose de s'appuyer sur des points caractéristiques détectés par FAST [Rosten 2005] (*Features from Accelerated Segment Test*) sur des images consécutives. L'intervalle de recherche des pixels homologues dépend alors du déplacement de ces points.

La référence [Dobias 2011] se base sur un principe similaire. Les auteurs prédisent la prochaine carte de disparité en s'appuyant sur le mouvement de la caméra et effectuent une recherche localisée autour des valeurs prédites. Par détection de points caractéristiques dans les paires d'images précédentes et actuelles, ils estiment le changement de pose du banc stéréoscopique entre les deux instants. Puis, connaissant l'ancienne carte de disparité, ils expriment chaque pixel comme des points 3D de la scène qu'ils projettent sur la nouvelle image pour construire une carte de disparité associée à cette pose. Par la suite, ils valident la disparité calculée par comparaison pixel à pixel. Les valeurs de disparité des pixels invalidés sont supprimées, générant des trous dans l'image. Pour les remplir, ils proposent deux solutions : l'une estime les disparités manquantes par les techniques de stéréovision traditionnelles, l'autre propage les valeurs de disparités aux pixels voisins par un algorithme de segmentation, appelé *seed-growing*. Cet algorithme segmente une forme de manière itérative. Il part d'un point initial, puis itérativement agrège les pixels voisins, s'ils respectent les conditions définies pour appartenir à la forme segmentée.

Le lecteur intéressé peut accéder à un état de l'art plus complet dans [El Jaafari 2016] et [Jain 2014].

### 2.2.2.3 Autres techniques basées sur des capteurs passifs

En plus de la stéréovision, d'autres méthodes permettent d'estimer la profondeur sur la base d'images délivrées par des caméras, avec notamment des caméras plénoptiques ou des techniques basées apprentissage.

**A Caméras plénoptiques** Les capteurs “light field” (aussi appelé plénoptiques) sont basés sur le même principe général que la stéréovision. Ces derniers comprennent une matrice de lentilles disposée devant le capteur. Ainsi, l'image obtenue est constituée de plusieurs petites images, pouvant chacune être assimilée à l'image d'une caméra peu résolue en des points de vue différents (Figure 2.12). Par exemple, la caméra Illum de Lytro est dotée de près de 200.000 micro-lentilles hexagonales.

Cette technologie, récemment en expansion grâce à l'arrivée de caméras plénoptiques dans le commerce (Lytro<sup>13</sup> et Raytrix<sup>14</sup>), est principalement orientée vers

13. <https://support.lytro.com/hc/en-us>

14. <https://raytrix.de/>

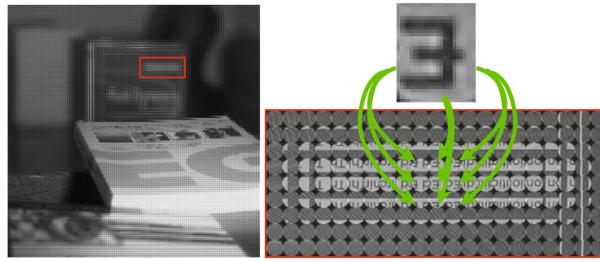


FIGURE 2.12 – Exemple de photographie acquise par une caméra light field (Extrait de [Bishop 2010]). L'image de droite montre une vue agrandie d'une portion de celle de gauche. On y observe que la lettre E sur l'ouvrage est observée à travers plusieurs micro-lentilles pour des points de vue différents.

la photographie et le post-traitement des photos, tel que le changement de focalisation et de profondeur de champ. Néanmoins, cet outil permet aussi l'estimation de cartes de profondeur comme le propose [Tao 2015]. Toujours dans un objectif de post-traitement (c'est-à-dire sans objectif de temps réel), la plupart des travaux de recherche visent à améliorer la qualité de l'image de profondeur [Wang 2015a] ou sa résolution [Bishop 2010]. Cependant, les auteurs de [Vasko 2015] sont parvenus à atteindre des durées de traitement de 32 ms pour une image de 1024x1024 pixels en parallélisant les calculs sur le GPU NVIDIA GeForce GTX TITAN. La référence [Hänsel 2017] envisage également une application temps réel de ces capteurs en portant les calculs sur FPGA.

Malgré ces résultats intéressants, il est à noter que ce domaine de recherche est encore récent, et peu de travaux sont disponibles en open-source.

**B Perception 3D basée sur des techniques d'apprentissage** Depuis l'essor des méthodes d'apprentissage, et plus particulièrement à base de réseaux de neurones profonds convolutionnels,<sup>15</sup> rendu possible grâce à l'avancée technologique de systèmes de calculs massivement parallèles, de nombreux travaux de recherche se sont intéressés à l'estimation de cartes de profondeur à partir d'images de caméras.

**Monoculaire** Parmi ces travaux, certains s'orientent sur l'utilisation d'une seule caméra. Le système mis en place doit donc apprendre à détecter des structures

15. Les réseaux de neurones constituent une sous-discipline de l'apprentissage automatique (machine learning). Le principe est de modéliser un système complexe (régression ou classification) par un réseau constitué de plusieurs couches de neurones. Chaque neurone, hormis ceux de la première couche, effectue une somme pondérée sur les valeurs en sortie des neurones de la couche précédente à la quelle il est connecté puis une fonction généralement non linéaire est appliquée à cette somme. Les pondérations des nœuds représentent les paramètres du modèle qu'il faut estimer par une étape d'étalonnage appelé apprentissage. Le réseau de neurones profond est un réseau de neurones admettant plus d'une couche intermédiaire en plus de la première et de la dernière couche, associées respectivement aux données d'entrée et de sortie. Le principe général est que plus le réseau est profond, plus le modèle admet de paramètres et plus il peut modéliser des systèmes complexes et des idées abstraites.

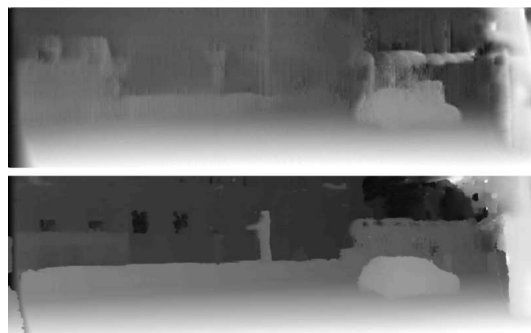


FIGURE 2.13 – Images de profondeur (Extrait de [Smolyanskiy 2018]) estimées par des réseaux profonds, de [Godard 2017] (en haut) basé monoculaire et [Smolyanskiy 2018] (en bas) basé stéréovision.

géométriques particulières dans l'image et en déduire leur profondeur.

Comme l'explique [Garg 2016], l'inconvénient de ces méthodes est la réalisation de bases de données précises et massives. Pour l'estimation de cartes de profondeur à partir d'images couleurs, les données recherchées doivent comprendre une série d'images avec leurs cartes de profondeur associées. Cependant, pour obtenir une vérité terrain de la profondeur, des systèmes LIDAR sont utilisés, et ces derniers sont sujets à erreurs en présence du soleil. C'est pourquoi on retrouve beaucoup de projets visant un apprentissage semi-supervisé voire non-supervisé, qui évitent l'acquisition de données de profondeur.

Le lecteur intéressé pourra accéder à un état de l'art de l'estimation par apprentissage de cartes de profondeur sur la base d'une image caméra sur [Godard 2017]. Par ailleurs, la plupart de ces travaux rendent leur code disponible sur internet [Eigen 2014], [Liu 2015], [Garg 2016], [Godard 2017], [Chen 2016].

En termes de temps de calcul, Godard et al. [Godard 2017] obtiennent des performances intéressantes avec moins de 35 ms de calculs pour une image de  $512 \times 256$  pixels sur le GPU Titan X.

**Stéréovision** Face aux efforts et progrès récents pour l'estimation de cartes de profondeur en monoculaire, les auteurs de [Smolyanskiy 2018] revendiquent que le stéréo matching reste une meilleure approche, si l'on cherche à obtenir des résultats précis pour tous types d'environnements même non familiers (Figure 2.13). Ils proposent un réseau de neurones profond qui estime une carte de profondeur à partir des images issues d'un banc stéréoscopique. Leur solution a été testée sur une carte graphique embarquée Jetson TK2 et met 11s pour traiter une image de  $1025 \times 321$  pixels avec une disparité maximum de 136 pixels, et 650 ms sur un GPU Titan XP.

La référence [Kendall 2017] propose un état de l'art sur les techniques basées apprentissage pour l'estimation de profondeurs sur la base d'images issues d'un banc stéréoscopique. Contrairement au cas monoculaire, ces travaux sont plus rarement disponibles en open-source [Zbontar 2016].

#### 2.2.2.4 Conclusion

Dans cette section, plusieurs méthodes de perception 3D de l'environnement ont été présentées : la vision active et la vision passive (avec la stéréovision sur une paire d'images ou sur un flux vidéo, les capteurs plénoptiques et les méthodes basées apprentissage). Peu de ces travaux de recherche aboutissent à des codes open-source. Ainsi pour nos deux prototypes conçus au cours de la thèse et décrits dans le chapitre 4, des compromis ont été faits et nous avons opté pour l'utilisation d'un banc stéréoscopique et de l'algorithme de mise en correspondance stéréo SGBM implémenté par OpenCV avec la fonction de coût CENSUS.

## 2.3 Localisation des lunettes dans la scène

La localisation des lunettes équipées de verres actifs dans la scène est nécessaire pour construire une carte et stocker des informations géométriques (par exemple un nuage de points 3D) ou photométriques de la scène. Grâce à cette carte, l'analyse de la scène ne se base pas uniquement sur les dernières données perçues par les capteurs, pour une zone restreinte de l'environnement, mais elle prend en considération l'ensemble des informations enregistrées. Lors de la construction de la carte de la scène, la pose des lunettes doit être estimée au sein de cette carte, pour que l'information acquise soit intégrée au bon emplacement. De même, la pose des lunettes est nécessaire pour exploiter la carte.

Par ailleurs, en réalité augmentée, la localisation de la lunette permet de générer un affichage stable, qui tient compte du déplacement de l'utilisateur. En effet, pour afficher un élément synthétique fixe dans la scène, la première étape est de définir sa pose au sein de la scène. Ensuite, le contenu synthétique est projeté sur chaque image générée par les lunettes et affichée devant les yeux de l'utilisateur. Si les lunettes bougent dans la scène, alors la pose de l'élément synthétique par rapport aux lunettes doit être réestimée.

Il existe différentes techniques de localisation, qui sont utilisées en robotique et en réalité augmentée. Parmi elles, on peut distinguer deux catégories : celles nécessitant une instrumentalisation de l'environnement et celles fonctionnant en environnement inconnu. Les solutions basées sur un environnement contrôlé ajoutent des éléments au sein de la scène. Les lunettes sont alors localisées par rapport à ces éléments. En revanche, les techniques adaptées aux environnements inconnus ne requièrent pas l'ajout d'objet au sein de la scène. Cette section présente un état de l'art sur les solutions de localisation pouvant être appliquées aux lunettes actives. Elle se scinde en deux parties. La première décrit les techniques opérant en environnement maîtrisé. La seconde traite les solutions fonctionnant en environnement inconnu.

### 2.3.1 Localisation dans un environnement maîtrisé

Une grande variété de capteurs ont été utilisés pour la localisation de personnes en mouvement dans une scène [Welch 2002] : les capteurs ultrasons, les capteurs magnétiques, les systèmes optiques utilisés pour la capture de mouvement, les caméras intégrées aux lunettes et les structures mécaniques. Parmi ces dispositifs, nous distinguerons deux sous-catégories : les systèmes basés sur l'utilisation d'une ou de plusieurs caméras embarquées sur les lunettes et les systèmes n'utilisant pas de caméra embarquée.

#### 2.3.1.1 Systèmes sans caméra embarquée

Le premier casque de réalité augmentée de [Sutherland 1968] se localisait à l'aide d'un système mécanique, qui reliait physiquement le casque à la scène. Le système proposé est articulé et permet au casque de gérer une ou plusieurs rotations. L'amplitude du mouvement est mesurée par un encodeur intégré dans chacune des articulations. Cette solution contraint fortement les mouvements du casque par rapport à la scène et impose un dispositif encombrant.

Les capteurs ultrasons et magnétiques sont constitués de plusieurs modules physiques : des émetteurs disposés à différents endroits de la scène et un capteur fixé sur les lunettes. Par comparaison au système mécanisé, ces deux types de capteurs peuvent localiser les lunettes dans un plus grand espace, et permettent une plage de mouvement plus large. Malgré cet avantage, les solutions magnétiques et ultrasons présentent un inconvénient important : leur sensibilité au bruit. Les dispositifs ultrasons sont mis en difficulté en présence de vent et de variation de température ou d'humidité. Les capteurs magnétiques, quant à eux, sont gênés par la proximité de matériaux conducteurs ou ferromagnétiques.

Une autre méthode appelée système de capture de mouvements (MoCap – “motion capture”) consiste à équiper la scène de plusieurs capteurs actifs, comportant une caméra et un éclairage infrarouge. Le dispositif détecte dans l'infrarouge des marqueurs placés sur les lunettes et les localise ainsi à une fréquence de 200Hz avec une précision de l'ordre du millimètre.

Ces systèmes ont tous pour point commun de ne fonctionner que dans un espace restreint délimité par le positionnement des émetteurs ou des capteurs. Ils ne sont pas adaptés pour une utilisation en extérieur (trop de contraintes et de sources de bruits sur les capteurs). Par ailleurs, ils peuvent nécessiter, pour certains, une longue procédure de mise en place et d'étalonnage, ce qui les rend difficilement transportables d'une pièce à l'autre.

#### 2.3.1.2 Systèmes avec caméra(s) embarquée(s)

Les systèmes de localisation avec une caméra embarquée sur les lunettes surmontent certains des problèmes rencontrés précédemment. Pour estimer la pose des lunettes, ils localisent la caméra dans la scène en détectant sur ses images des marqueurs ou des objets de forme géométrique connue.

**A Systèmes basés marqueur** Un ou plusieurs marqueurs (tag plan ou objet texturé) de dimension et d'apparence connues sont positionnés dans la scène. Lorsque la caméra filme l'un d'eux, un algorithme détecte sa projection sur l'image acquise par la caméra (Figure 2.14(a)).

- Les tags, souvent définis par des motifs géométriques noir et blanc, présentent un fort contraste. Ils sont donc facilement détectables par des techniques de seuillage et de détection de contour ou de ligne. La bibliothèque Apriltag [Olson 2011] offre par exemple un ensemble de fonctions de traitement.
- Pour la détection de marqueurs texturés (c'est-à-dire présentant des variations de luminance et/ou de teinte forte), l'algorithme cherche des points caractéristiques dans l'image, dont l'apparence locale les rend facilement détectables et reconnaissables. Il existe différentes méthodes de détection de points caractéristiques : SIFT [Lowe 2004], FAST [Baya 2008], BRIEF [Calonder 2010], ORB [Rublee 2011], etc. Lors de la détection, ces points sont associés à un descripteur, qui permet de les comparer à d'autres points caractéristiques et, par la suite, de les reconnaître sur l'image (ou l'objet) texturée de référence. Ensuite, une seconde étape cherche des correspondances entre les points détectés dans l'image de la caméra et ceux détectés sur le marqueur. Pour que la recherche de points fonctionne, l'objet ou l'image de référence doit être texturé sans motif répétitif, ce qui fausserait l'association de points caractéristiques.

Une fois le tag ou l'élément texturé détecté, on procède à l'association entre des points 2D issus des projections des marqueurs sur l'image de la caméra et les points 3D correspondants appartenant au marqueur. Les points 2D sont exprimés dans le repère associé à l'image et les points 3D dans le repère associé au marqueur fixe dans la scène. La transformation rigide (rotation et translation) entre le repère 3D de la caméra et le repère 3D du marqueur est alors estimée à l'aide des points 2D et 3D par la méthode des  $n$  points (PnP<sup>16</sup> – Perspective- $n$ -Point) ou par un calcul d'homographie lorsque les points du marqueur appartiennent à un même plan.

**B Systèmes basés modèle 3D** Une autre solution pour localiser la caméra sans ajouter de marqueur dans la scène est de détecter sur l'image un objet dont la forme géométrique est connue (avec un modèle maillé constitué de points et de segments). Les conditions sur l'apparence de l'objet sont moins fortes que précédemment : il peut être peu texturé. Par exemple, cette solution peut localiser la caméra dans la scène en détectant des meubles de cuisine, dont on connaît le modèle maillé. Par cette méthode basée modèle 3D, il n'est pas nécessaire d'ajouter des tags ou des objets dans la scène. Pour effectuer la localisation, l'algorithme considère une transformation rigide initiale approximative, qui peut être estimée par la méthode précédente (détection de points caractéristiques et application du PnP). À partir de cette transformation initiale, le modèle maillé est projeté sur une image théorique

16. PnP, Perspective- $n$ -Point, est une méthode permettant de connaître la pose d'une caméra, préalablement étalonnée, par rapport à un repère fixe de la scène à l'aide de la position de  $n$  points 3D de la scène et de la position de leur projection sur l'image de la caméra.

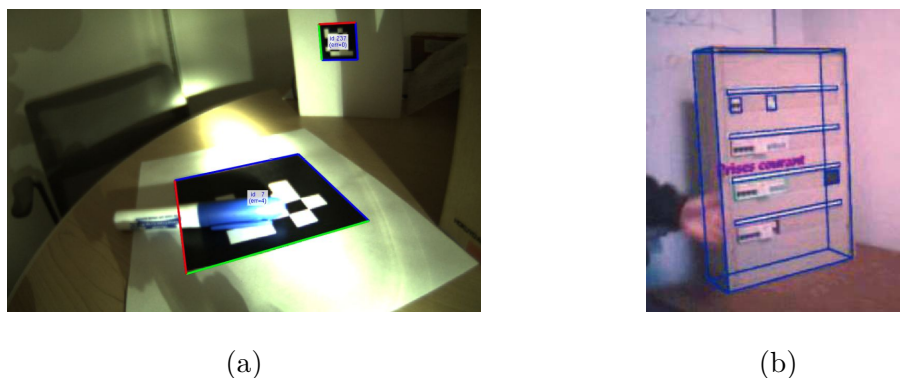


FIGURE 2.14 – Deux exemples de localisation avec caméra embarquée. (a) : Illustration de la localisation basée marqueur où des tags sont détectés sur l'image (Extrait de [Olson 2011]). (b) Illustration de la localisation basée modèle 3D, où un tableau électrique est détecté sur l'image et sa pose relative par rapport à la caméra est estimée à l'aide d'un modèle 3D de l'objet (Extrait de [Comport 2006]).

de la caméra, où le maillage est tracé en blanc sur un fond noir. Ensuite, en calculant les gradients de l'image réelle acquise par la caméra, les contours des objets sont mis en évidence. L'algorithme peut alors estimer une nouvelle transformation en minimisant la distance entre les segments du modèle maillé projeté sur l'image théorique et les contours présents dans l'image réelle. La figure 2.14(b) montre un exemple de localisation basée modèle 3D.

Pour ces deux méthodes de localisation basées caméra embarquée, la procédure d'installation dans la scène est moins contraignante : pose de tag ou d'objet tout au plus. En revanche, il est nécessaire de connaître les dimensions et l'apparence du marqueur ou de l'objet utilisé pour la localisation. D'autre part, les lunettes ne peuvent être localisées que dans une zone prédéfinie contenant les marqueurs. Cette technique peut être utilisée pour des applications s'exécutant dans un cadre connu. Par exemple, pour une application d'aide à la maintenance de véhicule, l'algorithme peut alors détecter des éléments ou des objets connus de la voiture, tels que le moteur, les bouchons de réservoir ou la batterie. Un état de l'art plus détaillé sur ces méthodes est disponible dans [Marchand 2016].

### 2.3.2 Localisation dans un environnement inconnu

Pour les techniques présentées précédemment, un élément connu (capteur, émetteur, objet) doit être présent dans la scène où l'utilisateur se déplace. On dit alors que l'environnement est instrumenté. Ces solutions ne peuvent donc pas fonctionner en environnement inconnu. Suivant les applications visées pour les lunettes actives, le système doit être capable de gérer les situations nouvelles, où l'utilisateur se déplace dans des lieux non connus à l'avance. Dans le cadre du projet, nous nous intéressons donc plus particulièrement aux techniques de localisation en environne-

ment inconnu.

Plutôt que d'utiliser des amers connus dans la scène (tag ou objet), le système construit lui même une carte de l'environnement contenant des informations photométriques ou des amers visuels (points, segments, etc). De la sorte, la caméra est localisée grâce aux données enregistrées dans la carte. Les algorithmes estimant simultanément la pose de la caméra et la carte de l'environnement sont appelés "localisation et cartographie simultanée" ("Simultaneous Localisation And Mapping" - SLAM).

Au contraire, si aucune carte n'était construite, alors le mouvement effectué par la caméra entre deux acquisitions d'images serait estimé uniquement à partir des informations disponibles dans la dernière image. En conséquence, les erreurs liées à l'estimation de pose s'accumuleraient à chaque nouvelle image. C'est pourquoi il est préférable d'enregistrer dans une carte un grand nombre d'amers visuels, que l'algorithme utilise pour la localisation. Deux autres avantages existent lorsque la carte contient des amers. D'une part, l'algorithme peut réduire l'erreur d'estimation accumulée lors d'un déplacement, à chaque fois que la caméra observe d'anciens amers depuis des nouveaux points de vue. Dans ce cas de figure, une fermeture de boucle est effectuée, en réestimant les poses et les amers observés au cours du déplacement. D'autre part, si le suivi de la caméra est perdu, une procédure de relocalisation cherche dans les nouvelles images d'anciens amers présents dans la carte.

Il existe deux catégories principales de SLAM : le SLAM basé primitives et le SLAM direct.

### 2.3.2.1 SLAM basé primitives géométriques

Le SLAM basé primitives construit une carte à l'aide d'amers visuels (aussi appelés primitives), qui peuvent être des points 3D (ou des droites, des patches plans, ...). Pour la localisation, l'algorithme détecte sur les images des primitives connues issues de la carte. Ensuite, la pose de la caméra est estimée afin de minimiser un critère d'erreur de reprojection. De manière générale, le SLAM effectue les actions suivantes :

- détection d'amers sur l'image de la caméra ;
- ajout de nouveaux amers à la carte
- suivi des amers connus ;
- estimation de la pose de la caméra ;
- relocalisation de la caméra quand le suivi des primitives a échoué ;
- fermeture de boucle lorsque la caméra observe une ancienne portion de la carte.

L'étape de cartographie est plus longue et plus calculatoire que celles de localisation et sont effectuée en tâche de fond, à une fréquence moins importante dans un SLAM temps réel.

Il existe deux principaux types de méthodes d'estimation de pose de la caméra : les méthodes basées filtrage et les méthodes basées ajustement de faisceaux ("bundle adjustment").

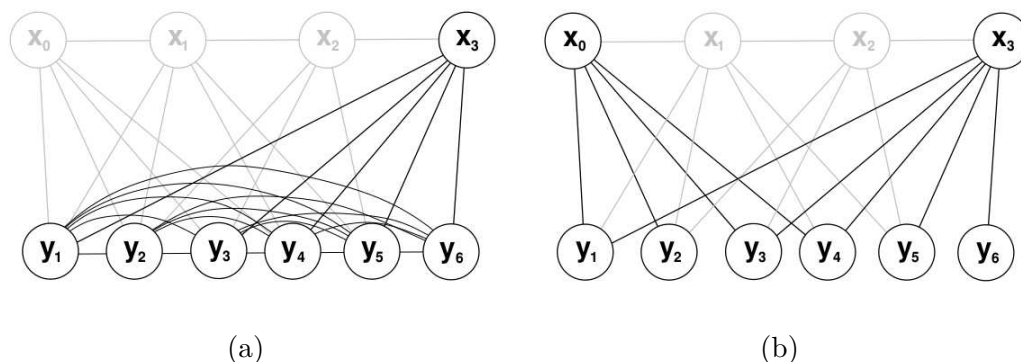


FIGURE 2.15 – Deux schémas illustrant les données mises en jeu pour l'estimation de la pose de la caméra, ainsi que leur dépendance (Extrait de [Strasdat 2010]).  $y_i$  pour  $i \in [1, 6]$  et  $x_j$  pour  $j \in [0, 3]$  désignent respectivement les positions des amers visuels et les poses successives des caméras. Les données prises en compte dans le calcul de l'estimation de la pose de la caméra sont écrites en noir. (a) illustre le SLAM basé filtrage, où les lignes reliant les données entre elles correspondent aux interdépendances entre les données codées par la matrice de covariance. (b) illustre le SLAM basé ajustement de faisceaux, où les lignes désignent quels amers sont visibles sur les images clefs.

**A Méthodes basées filtrage** Les méthodes basées filtrage [Harris 1988][Davison 2003] réévaluent pour chaque nouvelle image l'estimé et la covariance a posteriori du vecteur d'état, constitué de la pose (et éventuellement la vitesse) de la caméra ainsi que de la position des amers (Figure 2.15(a)). Cette technique ne réestime pas les poses passées de la caméra. Les incertitudes liées aux anciennes poses sont intégrées dans la matrice de covariance (Cette méthode n'est pas robuste face aux données erronées, car elle suppose des bruits gaussiens et quand elles sont intégrées à la matrice de covariance, il n'est plus possible de les supprimer ou de minimiser leur impact). D'autre part, si la carte contient beaucoup de primitives, la réévaluation du vecteur d'état nécessitera beaucoup de calculs. Pour satisfaire des contraintes temps réel, cette méthode doit donc être restreinte à un nombre limité de primitives. Toutefois, elle a pour avantage que la pose peut être mise à jour à haute fréquence en intégrant des données issues de capteurs tels qu'une centrale inertielle et que le vecteur d'état avec vitesse permet de prédire des poses du futur.

**B Méthodes basées ajustement de faisceaux** Pour chaque nouvelle image, l'algorithme estime la pose actuelle de la caméra et réestime les anciennes poses, ainsi que la position des primitives. Pour éviter que l'estimation ne s'effectue sur

un nombre de poses trop élevé, le SLAM ne considère qu'un sous-ensemble de poses particulières associées à des "images clefs" ("keyframes") sélectionnées dès qu'une image contient une grande proportion d'amers nouveaux (Figure 2.15(b)). Grâce au nombre réduit d'images clefs, l'algorithme peut s'exécuter en temps réel. D'autre part, [Strasdat 2010] montre que les méthodes basées ajustement de faisceaux avec images clefs sont plus performantes dans la plupart des situations que les méthodes basées filtrage.

### 2.3.2.2 SLAM direct

Le SLAM direct, contrairement au SLAM basé primitive ne nécessite pas d'extraction d'amers. Il compare les données photométriques pixel par pixel. Ainsi, au lieu de minimiser une erreur de reprojection (distance pixellique), il minimise une erreur photométrique (distance de valeur de pixels). Pour estimer la pose de la caméra, la comparaison peut se faire de deux manières. La nouvelle image est comparée soit avec la ou les dernières images clefs ([Meilland 2013], [Engel 2014]), soit avec les données photométriques stockées dans la carte. Si la comparaison s'appuie sur une carte, la reconstruction de la scène doit être dense et non éparse comme avec les méthodes basées primitives. Pour cela, il existe différents modèles de cartes denses : basés voxels [Damen 2012], surfaces [Newcombe 2011] ou un nuage de points dense.

La référence [Forster 2014] propose une technique semi-directe ("Semi-Direct Visual Odometry" - SVO) robuste au large mouvement entre images et sans recherche de points d'intérêt. Comme pour les techniques basées ajustement de faisceaux, les auteurs construisent une carte éparse de points 3D. Au lieu de détecter sur toute l'image des points d'intérêts, ils calculent, par projection des points 3D de la carte, une première estimation des positions des points 2D correspondants sur la nouvelle image. Ensuite, l'algorithme réajuste la position des points 2D en comparant des sous-fenêtres de pixels centrées sur les points projetés. Les sous-fenêtres issues de la nouvelle image sont comparées avec les sous-fenêtres correspondantes issues des anciennes images. Puis dans une dernière étape, la pose de la caméra et les positions des points 3D sont optimisés de manière à minimiser l'erreur de reprojection. Cette technique a l'avantage d'être rapide (environ 700 Hz) sur ordinateur de bureau et robuste au large mouvement entre images. Cependant, elle est mise en difficulté lorsque l'apparence des objets varie entre deux images.

Les méthodes de SLAM présentées dans cette section s'appuient sur les images d'une ou de plusieurs caméras ou sur les images d'un capteur RGB-D. Avec les capteurs RGB-D et les bancs stéréoscopiques, l'information de profondeur associée aux pixels est soit directement accessible, soit estimée par stéréovision. Pour les systèmes monoculaires (c'est-à-dire n'ayant qu'une seule caméra), du fait de l'absence des données de profondeur, la carte est construite à un facteur d'échelle près. L'ajout d'une centrale inertielle ("inertial measurement unit" - IMU), composée d'accéléromètres, de gyromètres et de compas, permet éventuellement d'estimer le facteur d'échelle et apporte plus de précision à la localisation et à la cartographie

[Mur-Artal 2017b].

### 2.3.3 Conclusion

Plusieurs méthodes de localisation utilisées en réalité augmentée et en robotique ont été présentées dans cette section. Pour notre banc de test présenté au chapitre 4, nous avons choisi de localiser le dispositif avec l'algorithme open source ORB-SLAM2 pour banc stéréoscopique de [Mur-Artal 2017a]. C'est un SLAM basé sur les primitives ORB à l'aide de sacs de mots visuels [Gálvez-López 2012], qui ont la particularité d'offrir une relocalisation et une fermeture de boucle robustes.

## 2.4 Détection et suivi du regard du porteur

Les mouvements oculaires fournissent une information riche sur l'état et l'intention d'une personne. Cette information peut être utile pour définir une commande d'obscurcissement, de focalisation et d'affichage sur des lunettes actives.

L'oculométrie (*eye tracking*, *gaze tracking*) désigne l'étude du mouvement de l'œil. On peut distinguer deux dénominations : l'*eye tracking* qui mesure le mouvement de l'œil et le *gaze tracking* qui estime la direction du regard (ou le point 3D observé) relativement à la scène [Chennamma 2013].

Ainsi, l'étude [Hansen 2010] des systèmes utilisant une ou plusieurs caméras filmant l'œil distingue les techniques de *eye tracking*, qui estiment la position de l'œil dans une image, et les techniques de *gaze tracking*, qui estiment la direction du regard. Cependant, les travaux de recherche en oculométrie ou suivi du regard utilisent généralement de manière interchangeable les vocables *eye tracking*, *gaze tracking* ou *eye-gaze tracking*.

Dans cette section, un état de l'art des techniques de *eye-gaze tracking* est présenté. Pour plus de références, le lecteur pourra consulter [Hansen 2010] et [Chennamma 2013].

### 2.4.1 Techniques d'oculométrie (suivi du regard, *eye-gaze tracking*)

Il existe de nombreuses techniques d'oculométrie, dont [Chennamma 2013] en propose un état de l'art. Les auteurs mentionnent, tout d'abord, les techniques d'électro-oculographie, où des électrodes posées sur la peau tout autour de l'œil mesurent les variations des différences de potentiel en fonction des mouvements de l'œil. Ces méthodes sont capables de détecter de larges mouvements des yeux ouverts ou fermés. Néanmoins, le contact des électrodes sur la peau les rend intrusives. Une autre solution, elle aussi intrusive mais en revanche très précise et rapide, est la bobine d'exploration sclérale ("scleral search coil"), qui enregistre les variations de courant générées par le mouvement d'une bobine intégrée dans une lentille de contact.

Les deux dernières catégories mentionnées dans [Chennamma 2013] sont toutes deux basées sur la réflexion de la lumière à la surface de l'œil. L'une d'elles consiste à mesurer la réflexion de la lumière (infra-rouge ou visible) par l'œil à l'aide de plusieurs capteurs de type photodiodes et phototransistors placés en périphérie. La lumière peut être projetée par des LEDs [McKay 1987] ou un laser [Irie 2002].

Un second ensemble de techniques, dites “basées vidéo”, se distingue par l'utilisation de caméras. La référence [Hansen 2010] en fournit un état de l'art. Ces méthodes peuvent être regroupées en deux sous-catégories : celles basées sur l'apparence et celles basées sur l'extraction d'éléments caractéristiques (pupille, iris, reflets cornéens, coins des yeux, paupières) dans l'image.

- La référence [Sheela 2011] décrit cinq techniques basées sur l'apparence. Aucune d'elles ne nécessite d'étalonnage géométrique de(s) caméra(s). Ces méthodes utilisent les données photométriques de l'image pour en déduire directement les coordonnées d'un point observé sur un écran. Elles ne requièrent pas d'étape d'extraction d'éléments caractéristiques. Parmi ces méthodes basées apparence, on retrouve aujourd'hui de nombreuses solutions ayant recours à des réseaux de neurones convolutionnels estimant la position d'un point sur un écran [Krafka 2016].
- Les autres méthodes basées vidéo se décomposent en deux étapes : la détection de l'œil et de certains de ses éléments caractéristiques dans l'image ; puis l'estimation de la direction du regard à l'aide de ces éléments caractéristiques. La deuxième étape se base soit sur un modèle de l'œil et de ses réflexions cornéennes, soit sur une interpolation, qui ne nécessite aucune modélisation de l'œil.

Les techniques basées sur l'usage de caméra(s) ont l'avantage de ne pas être en contact avec la peau ou l'œil. Elles sont donc plus facilement intégrables dans des lunettes avec des contraintes fortes d'apparence et de forme. Cependant, elles peinent à être robustes aux changements de luminosité, à la non-homogénéité de la lumière projetée (présence d'ombre), à l'ouverture et la fermeture des paupières et aux réflexions sur le cristallin.

Par la suite, nous présenterons plus en détail les méthodes basées caméra avec extraction d'éléments caractéristiques. Ces dernières, en plus de leur facilité d'intégration, peuvent fournir la pose de l'œil dans le référentiel de la caméra. Or, connaître la position de l'œil peut être utile pour assurer un affichage précis en réalité augmentée (voir section 2.6).

#### 2.4.2 Techniques basées caméra et extraction d'éléments caractéristiques

Les oculomètres basés caméra existent sous différentes configurations. Ainsi, la caméra est soit positionnée sur une table devant l'utilisateur, soit fixée sur des lunettes.

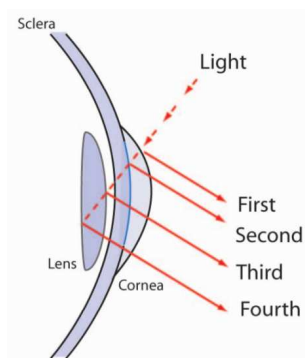


FIGURE 2.16 – La lumière se réfléchit sur les différentes couches successives composant la cornée (“cornea” sur le schéma) et le cristallin (“lens” sur le schéma), pour générer différentes images de Purkinje à différentes intensités. (Image extraite de [Hansen 2010])

#### 2.4.2.1 Éléments caractéristiques détectés

Comme précisé précédemment, les méthodes basées caméra et extraction d’éléments caractéristiques se déroulent en deux étapes. Pour la première, des éléments reconnaissables de l’œil (pupille, iris, contour des paupières) ou des reflets à la surface de la cornée appelés reflets cornéens sont détectés [Hansen 2010]. Pour une source lumineuse, on distingue jusqu’à 4 reflets, appelés images de Purkinje, correspondant aux réflexions successives sur les couches de la cornée et du cristallin (Figure 2.16). Le reflet cornéen correspond à la première image de Purkinje, la plus intense. Les autres images sont plus difficiles à détecter et nécessitent un environnement lumineux contrôlé.

La pupille a l’avantage d’être facilement détectable dans le spectre de l’infrarouge et son apparence reste similaire d’une personne à une autre. En effet, suivant la position de l’éclairage [Nitschke 2013b], la pupille apparaît noire ou blanche, ce qui facilite sa segmentation (Figure 2.18). Ainsi, de nombreux travaux se basent sur la détection de pupille dans l’infrarouge et la modélisation de son contour dans l’image par une ellipse ([Kumar 2009], [Świrski 2012], [Fuhl 2016]). Cependant, l’utilisation de la pupille présente deux principaux défauts. Sa taille varie, ce qui ajoute un paramètre supplémentaire à estimer quand on souhaite reconstruire sa pose 3D, et sa forme est altérée par réfraction à travers la cornée. C’est pourquoi [Nitschke 2013a] préfère détecter dans le spectre du visible les bords externes de l’iris, qui ne subit ni changement de taille, ni réfraction. Toutefois, l’iris est plus difficilement détectable et n’est pas toujours entièrement visible car partiellement caché par les paupières (Figure 2.17).

La cornée agit aussi comme un miroir et réfléchit la lumière vers la caméra. Partant de ce principe, la plupart des recherches se basent sur la détection des reflets spéculaires de LEDs infra-rouge (Figure 2.18) dont la pose est préalablement estimée [Guestrin 2006]. Néanmoins, d’autres chercheurs s’intéressent aussi à la détection dans le domaine visible de la réflexion de la scène sur la cornée [Nitschke 2013a]

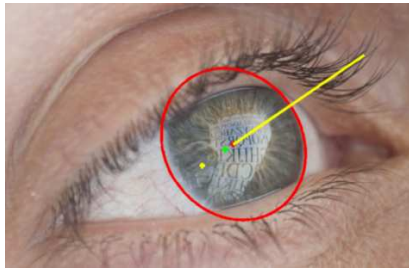


FIGURE 2.17 – Œil observé dans la lumière visible (extrait de [Nitschke 2013a]). Le contour de l'iris est détecté et tracé en rouge. La direction du regard est schématisée en jaune. Le reflet de la scène est visible sur la cornée.



FIGURE 2.18 – Œil observé dans l'infrarouge avec un éclairage hors axe générant une pupille noire. Le reflet cornéen d'une LED infra-rouge forme un point blanc sous la pupille (extrait de [Kolakowski 2006]).

(Figure 2.17), la caméra et l'œil formant alors une caméra catadioptrique.

#### 2.4.2.2 Méthodes basées interpolations

Les méthodes basées interpolation utilisent une fonction estimant la position 2D du point observé (sur un écran d'ordinateur ou sur l'image d'une caméra filmant la scène) à partir des éléments caractéristiques 2D détectés dans l'image de la caméra de l'oculomètre. Cette fonction peut être une fonction polynomiale [Li 2006], un réseau de neurones [Sewell 2010] ou une homographie [Li 2005].

**A Les PCCR et autres techniques s'appuyant sur les reflets** La solution la plus courante parmi les méthodes basées interpolation se nomme le PCCR (pupil center corneal reflection). Elle considère en entrée de la fonction d'interpolation les vecteurs différences entre le centre de la pupille dans l'image et les reflets cornéens. Cette technique a l'avantage d'être précise avec une erreur angulaire sur l'estimation de la direction du regard inférieure au degré.

Cependant, les PCCR présentent certains inconvénients liés à la détection de reflets cornéens, notamment il nécessitent un environnement lumineux contrôlé. En effet, de tels systèmes peuvent tomber en déroute si d'autres reflets viennent interférer, spécialement en extérieur où le soleil émet lui aussi de l'infrarouge. De plus, suivant les mouvements des yeux, le reflet n'est pas toujours situé sur la cornée, limitant ainsi les orientations de l'œil pouvant être estimées.

D'autres méthodes considèrent les différentes images de Purkinje. Elles présentent l'avantage d'être très précises mais requièrent un contrôle de la luminosité encore plus important. À titre d'exemple, la méthode de [Cornsweet 1973] détecte la première et quatrième images de Purkinje d'une LED infra-rouge et obtient ainsi une précision très élevée d'une minute d'arc pour un balayage de l'œil de 10 à 20 degrés.

**B L’usage de la pupille ou de l’iris seul** Les systèmes basés sur la détection de reflets cornéens requièrent un environnement lumineux contrôlé et ne peuvent donc pas être utilisés sur des lunettes au quotidien. C’est pourquoi certaines approches n’utilisent que la position de la pupille ou de l’iris comme paramètre d’entrée. Par exemple pour la détection de la pupille, ces méthodes atteignent de 1 à 2 degrés de précision pour [Li 2006] et 0.6 degrés pour [Kassner 2014]. Cependant, [Li 2005] montre que l’utilisation de la pupille seule offre de moins bons résultats.

**C Les limites de ces méthodes** L’ensemble de ces techniques basées interpolation présente l’inconvénient de ne pas être robustes au changement de pose de la tête de l’utilisateur par rapport à la caméra. De plus, leur usage est contraint aux conditions définies lors de l’étalonnage. Par exemple pour les méthodes estimant un point 2D observé sur un écran, [Nitschke 2013a] montre que si l’écran est éloigné du sujet après l’étalonnage, l’erreur angulaire augmente rapidement. Ces méthodes ne fonctionnent que sur le plan défini par l’écran et à la distance choisie lors de l’étalonnage. De même, pour le système proposé par [Kassner 2014] qui estime un point sur l’image d’une caméra, l’utilisateur ne peut pas se déplacer dans la scène sans perdre l’étalonnage de l’oculomètre.

Pour remédier au second problème lié au changement de pose du sujet par rapport à la caméra, certains auteurs utilisent des fonctions d’interpolation tenant compte d’un modèle de mouvement de la tête. Par exemple [Zhu 2007b] inclut un modèle de compensation de ce mouvement dans l’estimation. La méthode autorise un déplacement dans un volume de  $L \times H \times P = 200 \times 200 \times 300\text{mm}$  pour une caméra à 450mm du sujet avec une précision angulaire de 1,17 et 1,38 degrés suivant l’horizontale et la verticale.

#### 2.4.2.3 Méthodes basées modèle 3D

Généralement, les méthodes basées interpolations ne calculent pas la direction du regard dans le référentiel de la caméra. Elles fournissent un point 2D observé sur un écran. De plus, la plupart du temps, elles ne sont pas robustes au changement de pose de la caméra.

En revanche, les méthodes dites “basées modèle 3D” répondent à ces besoins. Leur principe est de reconstruire le trajet lumineux entre les éléments caractéristiques détectés et la caméra sur la base d’un modèle géométrique de l’œil (Figure 2.19). Pour les reflets cornéens, la surface de la cornée est alors assimilée à un miroir sphérique et le chemin de la lumière est calculé depuis la source lumineuse (LED). Ces méthodes requièrent une étape d’étalonnage pour estimer les paramètres du(des) modèle(s) de projection de la(des) caméra(s) du gaze tracker, ainsi que les positionnements relatifs des LEDs et caméra(s).

Différents modèles géométriques d’œil ont été utilisés, généralement celui de Gullstrand (Section 1.1.2.2). Les plus complexes tiennent compte de l’indice de réfraction du milieu cristallin, ainsi que de l’angle  $\kappa$  entre l’axe optique de l’œil (défini comme l’axe pupillaire à la section 1.1.2.2) et l’axe visuel [Guestrin 2006].

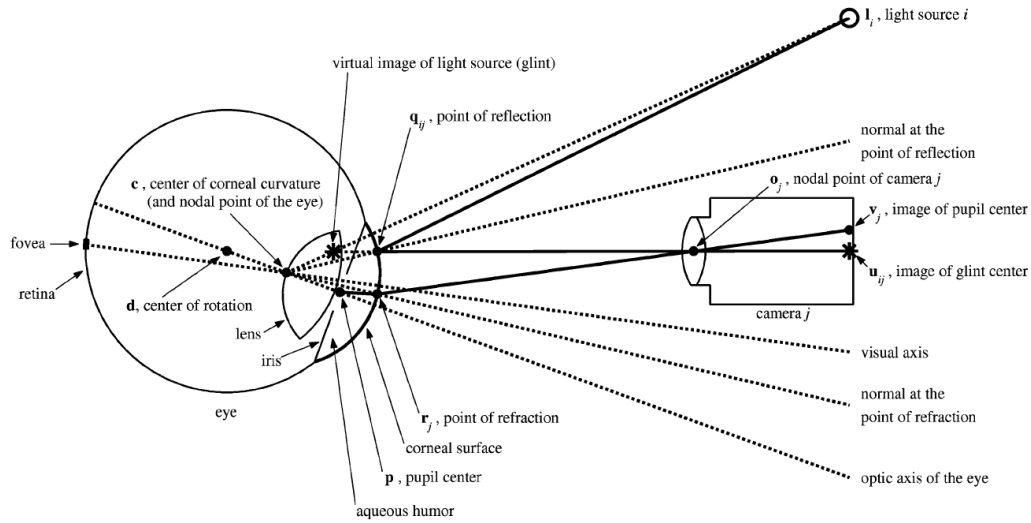


FIGURE 2.19 – Modèle de l'œil proposé par Guestrin et Eizenman. Les différents tracés de rayon sont représentés sur le schéma. La lumière issue des LEDs est réfléchiée sur la cornée et est focalisée par l'objectif de la caméra sur son capteur. De même, le rayon lumineux passant par centre de la pupille est réfracté par la cornée avant d'atteindre la caméra de l'oculomètre.

Les auteurs de [Guestrin 2006] ont étudié l'influence du nombre de caméras et de LEDs sur la procédure d'étalonnage. Une configuration constituée d'une caméra et de deux LEDs infrarouge est proposée, et atteint une erreur (Root Mean Squared Error - RMSE)<sup>17</sup> de 0.9 degré sur l'axe visuel estimé. Cette erreur est expliquée par l'asphéricité de la cornée et le bruit sur l'estimation des éléments caractéristiques. Par ailleurs, [Taba 2012] montre qu'un modèle intégrant cette asphéricité obtient de meilleurs résultats.

La référence [Nitschke 2013a] simplifie le système en estimant la pose de l'œil à partir de l'iris, qui ne subit pas de réfraction comme le fait la pupille. Pour déterminer l'orientation de l'axe pupillaire, les auteurs s'aident d'une projection perspective simplifiée. Dans leur configuration, cette approximation est possible puisque la taille de l'iris est bien plus faible que la distance entre la caméra et l'œil. Par la suite, l'axe visuel est déduit à 0.9 degrés près.

Les auteurs de [Swirski 2013] proposent une nouvelle méthode uniquement basée sur la détection de la pupille et ne nécessitant pas d'étape d'étalonnage pour chaque utilisateur. Pour cela, ils considèrent un modèle simple de l'œil en rotation autour d'un point situé sur l'axe optique, ici assimilé à l'axe visuel. Ainsi, à partir de plusieurs vues de l'œil ayant changé d'orientation, il est possible d'estimer sa position et sa rotation pour chacune des images. Une hypothèse initiale est formulée sur la distance de l'œil à la caméra afin de permettre l'estimation d'une

17. La racine de l'erreur quadratique moyenne (Root Mean Squared Error - RMSE) se calcule comme la racine carrée de la moyenne des erreurs quadratiques.

pose. Le résultat est ensuite raffiné dans un processus d'optimisation par moindres carrés sur la base des dernières images acquises. Par cette méthode, une précision de 2 degrés est atteinte sur des données simulées qui ne tiennent pas compte de la réfraction. Cependant, [Villanueva 2007] montre que les modèles sans réfraction peuvent conduire jusqu'à 5 degrés d'erreur angulaire.

### 2.4.3 Oculomètre basse consommation

L'essentiel des oculomètres développés et étudiés par la communauté scientifique et industrielle exploitent des caméras, car cette technologie est non intrusive. Cependant l'usage de caméras est énergivore pour la prise d'images et leur traitement, ce qui réduit la durée de fonctionnement dans le cas d'un oculomètre alimenté par batterie embarqué sur des lunettes.

Partant de ce principe, [Ando 2012] propose de réduire le coût énergétique induit par le traitement. Au lieu d'utiliser un ordinateur doté d'un système d'exploitation, les auteurs atteignent une consommation énergétique à 375 mW à l'aide d'un FPGA et d'un traitement peu coûteux en calculs s'exécutant à la fréquence de 8 Hz.

Cependant l'acquisition d'images et leur traitement consomment encore trop d'énergie pour permettre une utilisation continue sur plusieurs jours. Une solution peut être de s'orienter vers des caméras spécifiques à faible consommation énergétique telle que celle utilisée par [Nehani 2015]. Les auteurs donnent un exemple de système embarqué énergétiquement neutre, c'est à dire dont l'énergie totale consommée est produite par des cellules solaires embarquées. Il repose sur une caméra analogique en niveau de gris sélectionnée pour sa très faible consommation énergétique de 2 mW sous une tension de 3 V. De plus, elle autorise un accès individuel aux pixels, ce qui permet de réduire le coût énergétique résultant de leur transfert et de leur traitement. C'est sur cette technologie que s'appuie [Mayberry 2014]. Considérant que l'image acquise par la caméra d'un système de suivi du regard contient de nombreuses données redondantes, les auteurs préfèrent ne lire que certains pixels censés contenir à eux seuls suffisamment d'information. Dans leur solution, appelée "iShadow", un réseau de neurones est entraîné pour minimiser le nombre de pixels lus et déduire la direction du regard (Figure 2.20). Ainsi, sur les 112×112 pixels offerts par la caméra, l'application n'en lit que 10% et ne consomme que 70 mW pour un fonctionnement à 30 Hz et une précision angulaire de 3 degrés. Néanmoins iShadow n'est pas robuste au changement de luminosité, d'utilisateur et de pose relative entre l'oculomètre et le visage.

Un second système de suivi du regard, nommé "CIDER", a été développé en tant qu'amélioration de iShadow [Mayberry 2015]. Ce dernier ne consomme que 7 mW à 4 Hz ou 32 mW à 250 Hz, et fournit une précision de 0.6 degrés. CIDER propose plusieurs nouveautés :

- détection de la nature (intérieur *vs* extérieur) de l'environnement, et lancement d'un mode de fonctionnement associé; chacun des deux modes de fonctionnement s'appuie sur un réseau de neurones entraîné sur une série de données adéquate;

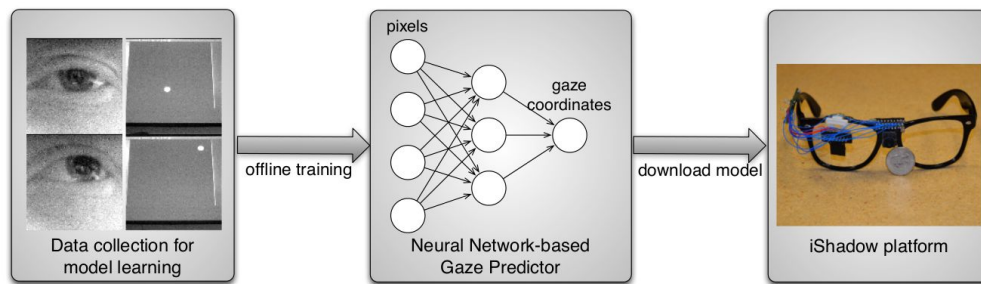


FIGURE 2.20 – Schéma de fonctionnement du système de suivi du regard basse consommation “iShadow”. Les pixels sont traités en entrée du réseau pour déduire la position du point 2D observé sur un écran (Image extraite de [Mayberry 2014]).

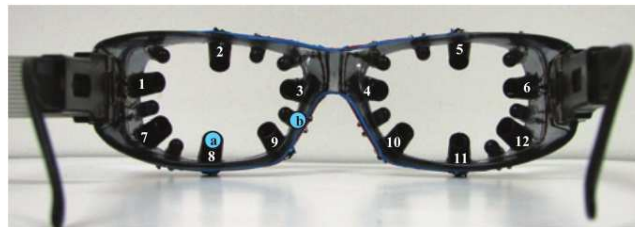


FIGURE 2.21 – Prototype proposé par [Topal 2014] intégrant (a) 6 photodiodes et (b) 6 LEDs infra-rouges disposées sur le pourtour du verre (Image extraite de [Topal 2014]).

- fonctionnement en lumière infrarouge (contrairement à iShadow) à l’aide de LEDs infra-rouge pour le mode intérieur et de la lumière infrarouge du soleil pour le mode extérieur ;
- augmentation de la fréquence de fonctionnement en proposant deux méthodes d’estimation, l’une basée réseau de neurones et l’autre basée sur un raffinement plus rapide par recherche locale sur l’image ;
- estimation du centre et du rayon de la pupille sur l’image (assimilée ici à un cercle par simplification) ;
- adaptation au changement d’utilisateur et au changement de pose entre oculomètre et visage grâce à un ré-apprentissage des réseaux de neurones, déporté sur un téléphone via une connexion Bluetooth.

Dans le but de diminuer la consommation énergétique, une autre solution a été proposée par [Topal 2014] en exploitant des photodiodes au lieu d’une caméra. Les auteurs montrent qu’avec 6 photodiodes et 6 LEDs infra-rouge par œil (Figure 2.21), il est possible d’estimer la position d’un point 3D visé sur un écran d’ordinateur par régression linéaire avec une précision de 0.93 degrés. Néanmoins, ni la robustesse de cette solution au changement de pose entre l’oculomètre et le visage, ni sa robustesse au changement de luminosité et d’utilisateur ne sont prouvées. Ce sont des problèmes que nous souhaitons résoudre.

### 2.4.4 Conclusion

Les techniques d'oculométrie ont été présentées dans cette section. Deux de nos prototypes détaillés au chapitre 4 disposent d'un oculomètre basé caméra, avec lesquelles deux techniques ont été testées : l'une basée interpolation [Kassner 2014], l'autre basée modèle 3D [Swirski 2013]. Une autre étude menée au cours de la thèse (chapitre 3) vise à développer un composant de suivi du regard basse consommation se basant sur les principes défendus par [Mayberry 2014] et [Topal 2014] : les données issues d'une caméra filmant l'œil présentent de nombreuses redondances, de sorte qu'il doit être possible de diminuer leur nombre sans diminuer la précision du système de suivi du regard. C'est pourquoi nous choisissons d'exploiter plusieurs photodiodes et LEDs infra-rouge, et de les disposer sur des montures de lunettes. Les données issues des photodiodes seront ensuite utilisées par un réseau de neurones préalablement entraîné hors ligne pour estimer la direction du regard.

## 2.5 Rendu synthétique de l'environnement et affichage d'information augmentée

Certains verres actifs sont adressables, c'est à dire que le champ de vision du porteur est partitionné en zones qui font chacune l'objet d'une action spécifique. Ainsi, la commande est une matrice à deux dimensions exprimant la commande pour chacune des zones adressées. La problématique étudiée dans cette section concerne sa détermination.

L'action appliquée peut consister en un affichage, un assombrissement ou une focalisation. Toutefois, la bibliographie disponible sur le domaine ne concerne que l'affichage de contenu synthétique pour la réalité augmentée. Nous traitons donc cette problématique d'un point de vue affichage. Certaines méthodes peuvent néanmoins être adaptées à d'autres types d'actions (assombrissement, changement d'indice de réfraction).

### 2.5.1 Deux procédés de synthèse d'image

Suivant les applications recherchées et les dispositifs d'affichage ciblés, les techniques de rendu diffèrent. Dans le cadre de la thèse, deux procédés d'affichage sont présentés : un premier pour générer du contenu synthétique à une pose spécifique de la scène (Figure 2.22), un second pour afficher du contenu synthétique lié à la géométrie et/ou à l'apparence de la scène réelle (Figure 2.23). Ces deux approches peuvent se faire l'une après l'autre sur la même image.

#### 2.5.1.1 Incrustation d'objets synthétiques

L'objet synthétique est généralement défini par un maillage de points 3D texturé, c'est à dire qu'une image de texture ou une couleur est associée à chacune de ses facettes. La projection de cet objet sur une image se déroule en plusieurs étapes. Au



FIGURE 2.22 – Exemple d'un affichage de contenu synthétique à une position et orientation souhaitées.



FIGURE 2.23 – Exemple d'un affichage de contenu synthétique dont la structure géométrique est issue de la scène réelle.

départ, une caméra virtuelle est définie ( focale, résolution, placement) de manière à générer l'image souhaitée et la pose du maillage est exprimée dans le repère de la caméra. Les points 3D du maillage sont ensuite projetés sur l'image de la caméra en des points 2D. Puis, chaque facette du maillage projeté suit un processus de rendu par rasterisation (matricialisation). Cette technique de rendu est disponible dans la librairie OpenGL<sup>18</sup> et s'exécute sur la carte graphique. Deux ensembles de paramètres doivent être renseignés pour réaliser ces opérations : la transformation rigide entre le repère associé à l'objet synthétique et celui associé à la caméra, ainsi que la matrice de projection perspective.

Sur des lunettes de réalité augmentée (binoculaire), deux caméras virtuelles doivent être définies pour générer des images des objets synthétiques pour chaque œil. La pose et le modèle de ces caméras doivent être estimés dans une procédure d'étalonnage introduite dans la section 2.6.

Par ailleurs, sur un système mobile, ce changement de pose entre l'objet synthétique et les caméras est amené à varier. Il est décomposé en plusieurs parties (Figure 2.24) :

- La transformation entre chaque caméra virtuelle et le repère associé aux lunettes. Elle doit être estimée dans une procédure d'étalonnage du système d'affichage des lunettes.
- La transformation entre le repère des lunettes et le repère de la scène, qui est estimée par un algorithme de localisation.
- La transformation entre le repère de la scène et les position et orientation ciblées pour l'objet synthétique, qui est exprimée après analyse de la scène. Par exemple, la pose de l'objet peut être définie sur un plan de la scène préalablement détecté.

### 2.5.1.2 Affichage de contenu synthétique construit à partir de la scène

Le contenu synthétique affiché est construit à partir de la géométrie et/ou de l'apparence de la scène. Pour cela, des images de la scène sont acquises par des

18. OpenGL (Open Graphics Library) est une librairie logicielle dédiée à la génération d'images 2D et 3D.

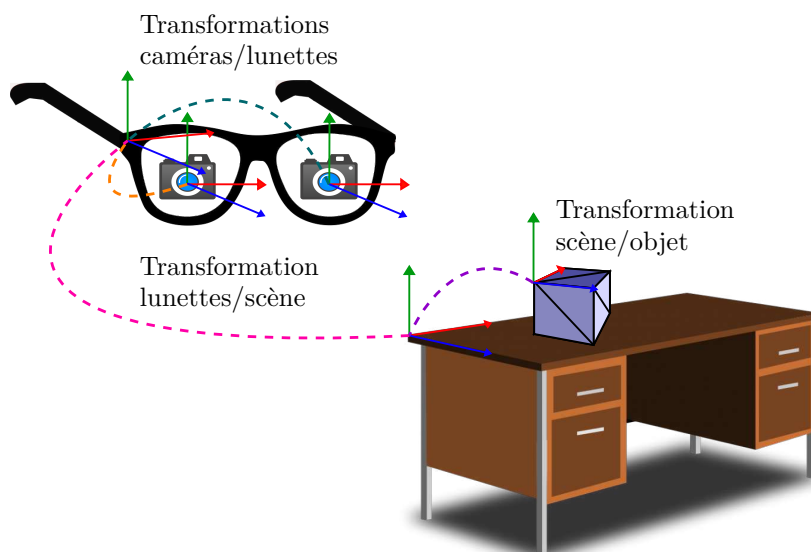


FIGURE 2.24 – Le changement de pose entre les caméras virtuelles et l’objet synthétique est composé à partir de trois transformations : de la caméra virtuelle au repère des lunettes (en orange et en vert), des lunettes à la scène (en rose) puis de la scène au repère de l’objet synthétique (en violet).

caméras, appelées caméras de scène ou caméras réelles, orientées vers la scène et fixées sur la monture des lunettes. Ensuite, de nouvelles images sont générées sur la base des informations acquises (données photométriques et géométriques de la scène).

Cette technique peut être utile lorsqu’on souhaite changer l’apparence des objets dans la scène, en variant leur couleur par exemple. En suivant cette idée, les lunettes pourraient améliorer le contraste d’une scène où mettre en évidence des objets particuliers en réhaussant leurs contours. D’autre part, ce procédé est nécessaire pour des dispositifs de type VST-HMD, où la scène réelle n’est pas directement perceptible et doit donc être ré-affichée à l’intérieur du casque. Comme expliqué à la section 2.1.2, les images acquises par des caméras de scène ne peuvent pas être directement affichées par le casque à cause d’une différence de point de vue entre les yeux de l’utilisateur et les caméras. Généralement, la différence de point de vue est négligée en positionnant les caméras de scène devant les yeux au bon écart inter-oculaire [Pankratz 2015] et [Steptoe 2014]. Cependant, lors d’une rotation de la tête, la perception du mouvement de la scène diffère toujours entre le point de vue des caméras de scène et celui des yeux de l’utilisateur. Les auteurs de [Keller 2005] proposent d’utiliser un système orthoscopique où les points de vue des caméras et des yeux sont confondus grâce à un dispositif matériel particulier. Une dernière alternative est de définir deux caméras virtuelles à l’emplacement des yeux et de générer des images pour ces caméras virtuelles en exploitant les données issues des caméras de scène. Ce procédé se nomme rendu basé image (“image based rendering” – IBR) [Shum 2000]. On distingue trois catégories.

- Une première catégorie ne considère aucune information géométrique et se base uniquement sur le contenu **photométrique**. Le principe est d'acquérir de nombreuses images depuis différents points de vue pour construire une fonction plénoptique  $f$  telle que l'intensité lumineuse  $I$  associée à un rayon lumineux passant par un point 3D  $(x, y, z)$ , orienté suivant les angles  $(\theta, \phi)$ , de longueur d'onde  $\lambda$ , à l'instant  $t$  s'écrit  $I = f(x, y, z, \theta, \phi, \lambda, t)$ . Après estimation de la fonction plénoptique, de nouvelles images peuvent être générées en appliquant la fonction à chacun des pixels. Pour que cette méthode génère des images fidèles à la réalité, un grand nombre d'images doit être acquis.
- Une deuxième catégorie **basée géométrie** exprime la scène comme un maillage texturé. Pour acquérir des données géométriques, la scène est analysée pour estimer un modèle 3D texturé. Ensuite une nouvelle image est générée en utilisant les méthodes de rendu traditionnelles. Cette technique présente l'avantage de nécessiter peu d'images. Néanmoins, elle gère mal l'influence de l'éclairage sur les textures.
- Une troisième catégorie est basée sur une **géométrie implicite ou partielle**. On parle de géométrie implicite car les informations géométriques ne sont pas directement disponibles, mais sont estimées à partir des images : calcul de carte de profondeur ou du flux optique<sup>19</sup>. Ces données géométriques sont alors utilisées pour générer de nouvelles images. Par exemple, pour construire l'image d'une caméra placée à mi-distance de deux autres caméras, il est possible de calculer le flux optique entre les deux images de départ et de déduire le flux optique associé à la nouvelle image. Les pixels de la nouvelle image sont alors décalés de la moitié de leur translation observée entre les deux premières images.

### 2.5.2 Rendu d'une image par DIBR pour un autre point de vue

La référence [Sun 2010a] conseille le rendu basé image et carte de profondeur ("depth image based rendering" – DIBR), une technique basée géométrie implicite, car elle reprend les avantages des autres méthodes. Elle consiste à synthétiser une image pour un nouveau point de vue à partir d'images réelles, où chaque pixel est associé à une profondeur. Les cartes de profondeur peuvent provenir de capteurs temps de vol [De Sorbier 2010] et [Saito 2011] ou peuvent être calculées par des algorithmes de stéréovision [Lai 2016a] et [Kauff 2007]. Toutefois, malgré son coût calculatoire important, la stéréovision s'associe bien avec le DIBR. En effet, sur les zones unies, même si la mise en correspondance stéréo estime mal la profondeur, l'image reconstruite par DIBR ne sera pas pénalisée.

Le DIBR se déroule en plusieurs étapes (Figure 2.25) : calculs de cartes de profondeur, génération d'images cibles intermédiaires, fusion des images cibles et remplissage des pixels indéterminés. Considérons le cas simple où une image  $I_c$  (image

19. Le flux optique correspond au mouvement apparent des objets entre deux images successives d'une même caméra. Il est décrit comme un ensemble de vecteurs de vitesse associés aux pixels de l'image. Ce mouvement peut être causé par le déplacement des objets de la scène et de la caméra.

cible) est construite à partir de deux images réelles  $I_{s1}$  et  $I_{s2}$  (images sources). Dans un premier temps, des cartes de profondeur sont calculées pour les images  $I_{s1}$  et  $I_{s2}$ . Puis deux images cibles intermédiaires  $I_{c1}$  et  $I_{c2}$  sont construites à partir respectivement de  $I_{s1}$  et  $I_{s2}$  et fusionnées en une image  $I_c$ , où réside des pixels indéterminés, qui sont remplis lors d'une dernière étape.

Cette section explique tout d'abord les étapes de synthèse et de fusion des images intermédiaires et de remplissage des pixels indéterminés, puis, des approches basées apprentissages sont présentées.

### 2.5.2.1 Synthèse d'images intermédiaires

Cette étape (nommée aussi “3D warping”) construit autant d'images cibles intermédiaires qu'il y a de caméras de scène comme le montre le schéma 2.25. Elle requiert tout d'abord pour chaque caméra de scène une image d'origine (niveau de gris ou couleur) et une carte de profondeur (ou de disparité) associée. Le processus nécessite également de connaître le modèle de projection de chaque caméra virtuelle et réelle, ainsi que la transformation rigide entre chaque couple de caméras virtuelle et réelle.

Plusieurs méthodes existent pour réaliser cette étape de “3D warping”. Certaines sont décrites dans [Morvan 2009]. Trois d'entre elles sont détaillées ci-dessous.

**A “Direct 3D warping”** Le “direct 3D warping”, aussi appelé “forward 3D warping”, vise à construire un nuage de points 3D à partir des pixels de l'image source et à le projeter sur l'image cible. L'avantage de cette méthode, facilement parallélisable sur GPU, est sa rapidité [Do 2012]. L'inconvénient est l'approximation faite sur le calcul du pixel de l'image cible. Un pixel de l'image source ne correspond pas parfaitement à un pixel de l'image cible. Après projection du point 3D, le pixel cible a une position non-entière. Pour assigner la valeur du point 3D à un pixel, nous choisissons le pixel le plus proche, ce qui aboutit à une approximation. Autre inconvénient, cette méthode laisse apparaître des trous de deux natures différentes :

- les cracks, qui sont des trous causés par des étirements de texture,
- les trous dus aux occlusions correspondant à des zones de l'image cible qui ne sont pas visibles dans l'image source, car un objet plus proche cache l'information souhaitée.

Par ailleurs, pour éviter qu'un pixel de l'arrière plan ne vienne obstruer un pixel du premier plan, il est possible d'utiliser un “z-buffer”<sup>20</sup>.

**B Maillage triangulaire** Une autre technique de “3D warping” utilise le maillage triangulaire, où l'image et la carte de profondeur sont divisées en triangles. Les valeurs des pixels inclus dans les triangles définissent la texture associées aux triangles (Figure 2.26). Par la suite, un point 3D est calculé pour chaque sommet de

20. Un z-buffer est un tableau de deux dimensions stockant la profondeur associée à chaque pixel. Ce buffer est rempli lors de l'étape de rendu et permet de gérer les occultations.

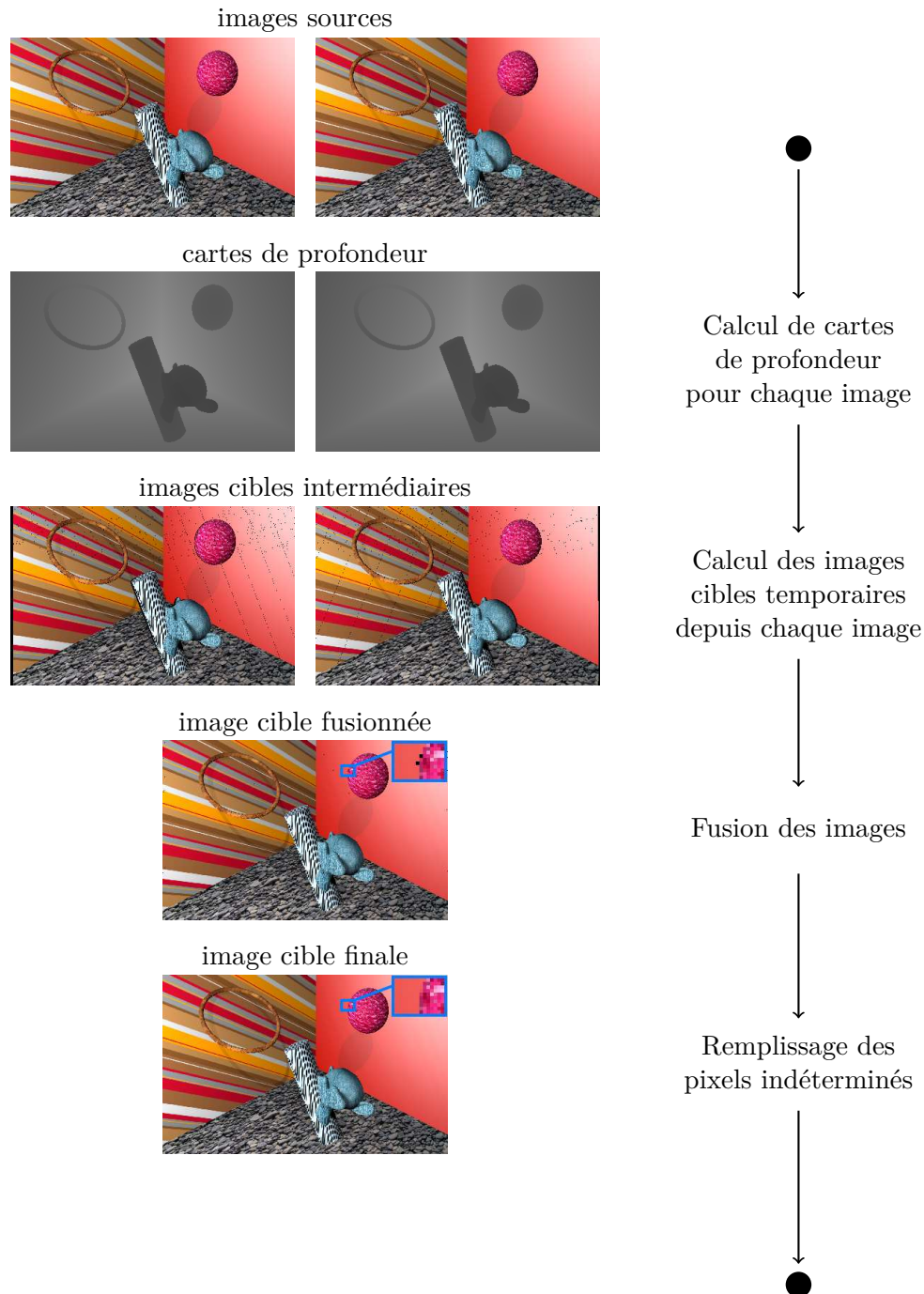


FIGURE 2.25 – Schéma résumant les étapes du DIBR, où deux images de départ sont utilisées pour générer une nouvelle image. On retrouve chaque étape du processus illustrée pour des images de tests (les images sources et les cartes de profondeur sont synthétisées par un logiciel dédié). Des images cibles intermédiaires sont calculées par “3D warping” pour chaque image source. On observe sur ces images des pixels noirs, qui n’ont pas été affectés lors du “3D warping”. Les deux images intermédiaires sont ensuite fusionnées, ce qui permet de faire disparaître la majorité des pixels indéterminés. Une valeur est ensuite affectée aux pixels indéterminés (zoom dans l’encadré bleu) restants lors d’une dernière étape de remplissage afin d’assurer la continuité dans l’image.

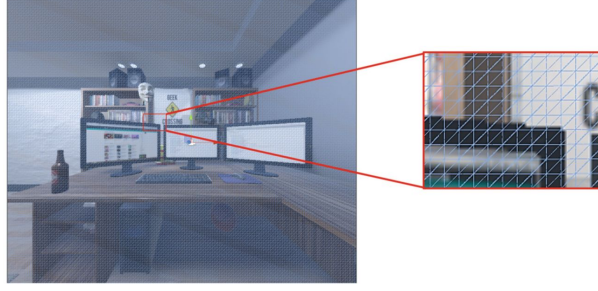


FIGURE 2.26 – Exemple d’image source divisée en triangles (Extrait de [Lai 2016b])

chaque triangle et est projeté sur l’image de la caméra cible. Par des méthodes traditionnelles de rendu d’image sur GPU, le maillage précédemment défini est projeté sur l’image cible et la texture est appliquée. Les avantages de cette méthode sont sa rapidité, l’absence de trou de type “cracks” et un rendu plus réaliste. Néanmoins, elle peut générer des artefacts sur les zones précédemment cachées par occlusion. Au lieu de laisser ces zones vides, la texture sera étirée sur toute la surface des trous, interpolant les valeurs de pixels correspondant à l’arrière plan et au premier plan. [Lai 2016b] reprend cette méthode mais l’adapte pour tenir compte des occlusions.

**C Inverse 3D warping** L’“inverse 3D warping”, aussi appelé “backward 3D warping”, consiste à calculer pour chaque pixel de l’image cible  $I_c$  une valeur à partir de l’image source  $I_s$  (Figure 2.27).

Afin d’accélérer la correspondance entre les deux images  $I_s$  et  $I_c$ , on considère un banc stéréoscopique synthétique rectifié épipolairement avec les images source rectifiées  $I_{sr}$  et cible rectifiées  $I_{cr}$ . Ensuite, on construit la carte de profondeur de  $I_{sr}$ , en calculant la valeur de profondeur  $depth$  pour chaque pixel  $(u, v)$  de  $I_s$  et la position du pixel correspondant sur  $I_{sr}$  par homographie. La valeur de profondeur d’un pixel de  $I_{sr}$  se calcule en appliquant la matrice de rotation entre les caméras non rectifiée et rectifiée, dont on ne considère que la troisième composante  $(r_{31}, r_{32}, r_{33})$ , au point 3D correspondant  $(x, y, depth)$  à  $(u, v)$  (Équation 2.8).

$$depth_{rectified} = \begin{bmatrix} r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ depth \end{bmatrix} \quad (2.8)$$

Ainsi, le processus pour remplir un pixel cible de  $I_c$  est le suivant (Figure 2.27) :

- calcul du pixel correspondant sur  $I_{cr}$  par homographie ;
- recherche du point associé sur  $I_{sr}$  le long de la ligne épipolaire en vérifiant que la disparité corresponde à la valeur de profondeur enregistrée ;
- calcul du point sur  $I_s$  par homographie ;
- interpolation bilinéaire sur les pixels à proximité ;
- affectation de la valeur interpolée au pixel de  $I_c$ .

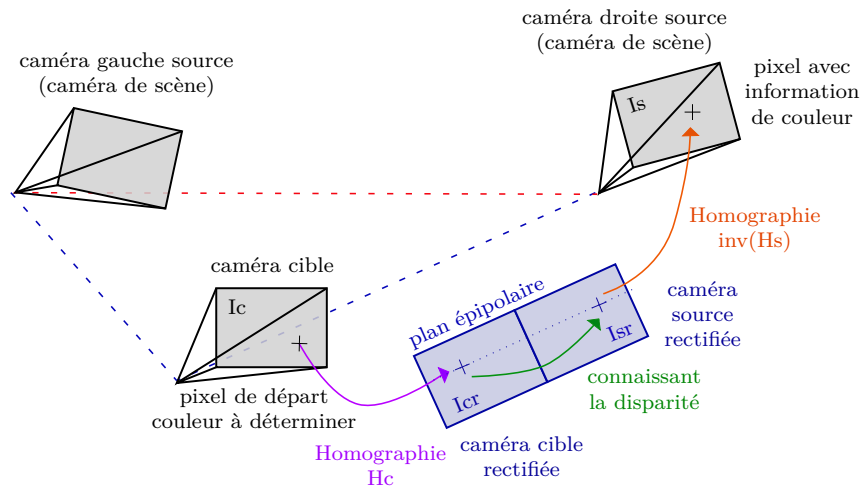


FIGURE 2.27 – Schéma retraçant les calculs nécessaires afin d'établir pour un pixel de l'image cible la couleur appropriée dans l'image source. Les caméras rectifiées partagent le même centre optique que les caméras de départ. Le plan épipoire a été représenté sur le côté pour clarifier le schéma.

L'avantage de cette méthode est sa précision. Elle fournit une image qui se rapproche le plus de l'image qui serait délivrée par une caméra cible réelle. En effet les pixels sont interpolés et les “cracks” sont réduits sans être tous supprimés. En revanche, elle est coûteuse en temps de calcul et n'est pas compatible avec des applications temps réel

**D Fusion des images cibles intermédiaires** Après “3D warping”, chaque image source a généré une image cible intermédiaire. Ensuite, ces images sont fusionnées pour réduire au plus les zones laissées indéterminées. La fusion doit tenir compte de deux facteurs : les différences de distorsions (étirement) et de couleurs. En effet, l'étape de “warping” peut générer des distorsions, et plus particulièrement lorsque les caméras sources et cibles sont éloignées l'une de l'autre. Différentes techniques permettent de fusionner les images intermédiaires :

- Une première technique de fusion réduisant ces artefacts consiste à pondérer les contributions des images sources en fonction de la proximité entre la caméra cible et la caméra source correspondante.
- Une seconde méthode consiste à remplir tout d'abord l'image cible avec l'image source la plus proche puis de remplir les pixels manquants successivement avec les autres images. Cette technique est adaptée lorsque la caméra cible est très proche de l'une des caméras sources. Néanmoins, elle ne tient pas compte des distorsions et différences de couleur, qui seront visibles sur les pixels remplis par les autres caméras plus éloignées.
- Une troisième solution est de choisir individuellement pour chaque pixel indéterminés l'image source par comparaison des valeurs de profondeur. Le pixel source ayant la profondeur la plus faible est sélectionné. Cette dernière mé-

thode fonctionne bien pour des cartes de profondeur de bonne qualité. Dans le cas contraire, elle introduit des artefacts.

Le lecteur intéressé pourra trouver plus de détails dans [Zhao 2013] et [Sun 2010b].

### 2.5.2.2 Remplissage de zones vides

Comme précisé plus haut, certaines techniques de “3D warping” génèrent des trous soit par étirement de texture soit par occlusion (Figures 2.28). Ces trous peuvent être remplis en grande partie lors de l’étape de fusion des images calculées à partir des différentes images sources.

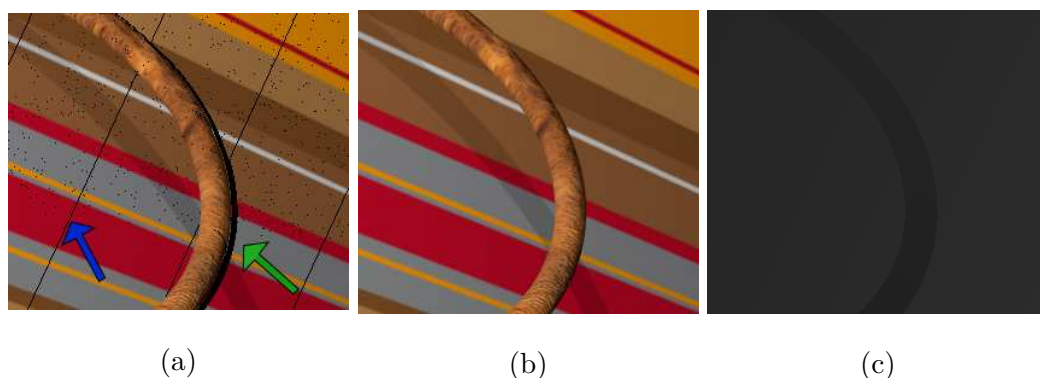


FIGURE 2.28 – (a) Portion d’une image générée après DIBR. (b) Portion correspondante de l’image source. (c) Carte de profondeur associée. Des trous de type “cracks” sont visibles sur le fond rayé (dont un indiqué par une flèche bleue.) Un trou causé par une occlusion est visible sur le côté droit de l’arceau marron (indiqué par une flèche verte).

Deux méthodes permettent de différencier les “cracks” des occlusions :

- Une première consiste à vérifier la variation de profondeur des pixels autour du trou : Si elle est importante, il s’agit d’une occlusion, sinon il s’agit d’un “crack”.
- D’autre part, comme les “cracks” sont des trous très fins (voir figure 2.28), ils peuvent être détectés en mettant en place une opération morphologique d’ouverture<sup>21</sup> sur l’image binaire du trou (pixel blanc pour le trou, noir sinon) [Oliveira 2015]. Ainsi, une image de “cracks” s’obtient par différence entre l’image binaire de départ et l’image obtenue après opération morphologique.

Les méthodes de remplissage des zones vides en DIBR se basent en partie sur les travaux d’in-painting, techniques qui admettent généralement un coût calculatoire non compatible avec des contraintes temps réel.

21. Une opération morphologique d’ouverture consiste à éroder les formes blanches sur l’image puis à la dilater. Cette technique permet de supprimer le bruit après binarisation de l’image. Ainsi, les tâches blanches de faible épaisseur disparaissent.

Un état de l'art des méthodes d'in-painting est disponible dans [Guillemot 2014]. Les auteurs expliquent que l'on peut distinguer deux principales catégories :

- La méthode basée diffusion utilise des équations aux dérivées partielles pour propager les éléments structurels (lignes) présents sur la périphérie du trou. Elle fonctionne bien pour des zones de petites tailles mais a tendance à flouter l'image pour des surfaces plus grandes et plus texturées.
- La méthode basée copie de données, appelée “exemplar based in-painting” (EBI), traite itérativement des patches de pixels en commençant sur la bordure de la zone vide (Figure 2.29). Un patch partiellement rempli et centré sur un pixel vide est comparé à d'autres patches de l'image. Si les pixels non vides des deux patches sont similaires, alors le reste du patch source est copié dans le patch cible. Cette méthode fonctionne très bien sur des grandes zones texturées mais est particulièrement lourde en calculs. En effet, dans l'algorithme d'origine proposé par [Criminisi 2003], les patches sources sont recherchés sur l'ensemble de l'image.

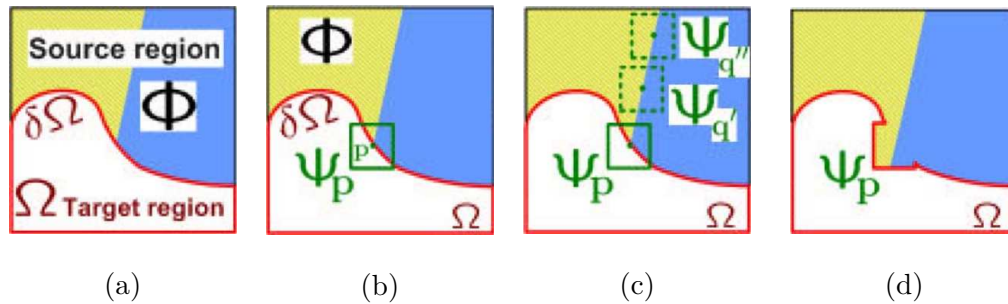


FIGURE 2.29 – Illustration de l'EBI (image extraite de [Criminisi 2003]) : (a) Image partiellement remplie avec  $\Omega$  la zone vide,  $\delta\Omega$  son contour et  $\Phi$  la zone remplie. (b) Image avec le patch  $\Psi_p$  centré sur le point  $p$ . (c) Exemples de patches ( $\Psi_{q'}$  et  $\Psi_{q''}$ ) compatibles avec  $\Psi_p$ . (d) Image avec le patch  $\Psi_p$  rempli à partir du meilleur patch candidat.

**A Remplissage des cracks** Plusieurs méthodes permettent de remplir les “cracks”. La plus courante est l'interpolation sur les pixels non vides voisins [Oliveira 2015], [Mao 2013]. Les trous générés par les étirements de texture étant petits, l'interpolation ne risque pas d'introduire un aspect flouté inapproprié. Une autre technique est d'appliquer une étape de “inverse 3D warping” supplémentaire sur ces pixels [Zinger 2010], [Gui 2013] ou d'adapter l'étape de “3D warping” en copiant chaque pixel source sur les voisins du pixel cible correspondant [Tian 2009], [Zinger 2010] ou en redimensionnant l'image source [Zinger 2010].

**B Remplissage des trous dus aux occlusions** Les trous générés par des occlusions peuvent nécessiter des techniques d’“in-painting” plus complexes. Lorsque

ces trous sont de petite taille, ils peuvent être remplis par des techniques d'interpolation à l'instar des trous de type cracks [Zinger 2010], [Solh 2012]. Pour l'interpolation, [Solh 2012] propose de réduire la dimension de l'image, en excluant les pixels indéterminés du calcul. L'image est réduite jusqu'à ce qu'il n'y ait plus de pixel vide. Ensuite, la dimension de l'image est ramenée à sa valeur d'origine. Cependant, quand la largeur de la zone de pixels vides est trop importante, l'interpolation provoque un aspect flouté mixant les différents éléments (premier plan et arrière plan) à la frontière du trou (figure 2.30). Pour éviter l'aspect flouté, on utilise souvent l'EBI.

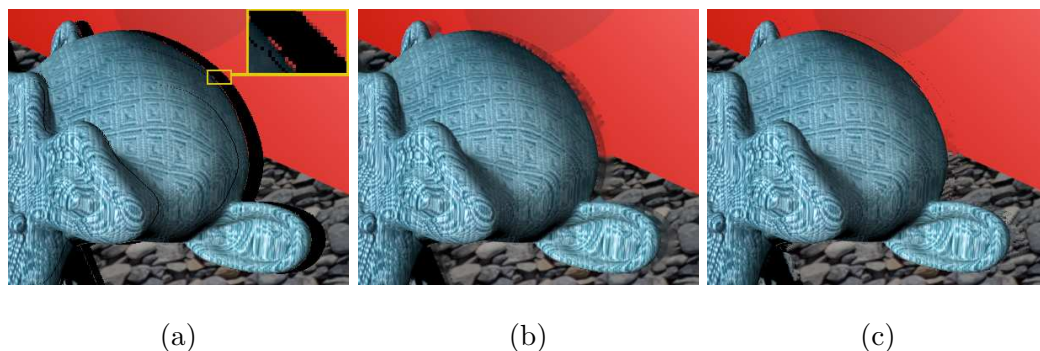


FIGURE 2.30 – (a) Portion d’une image générée après DIBR. (b) Même image après remplissage par interpolation. (c) Même image après remplissage par “exemplar based in-painting”. Il peut être noté que (a) comprend des pixels indéterminés à cause d’une occlusion (en noir, avec zoom en haut à droite).

Au sein d’un algorithme de DIBR, l’EBI est adapté au cas des occlusions, de telle sorte que les trous soient remplis de préférence avec des pixels appartenant à l’arrière plan [Choi 2013], [Reel 2013], [Ahn 2012], [Xu 2013].

D’autre part, il se produit souvent un effet de “pixels fantômes” sur les bords des trous dus aux occlusions, qui est causé par la présence de pixels dont les valeurs sont interpolées entre les éléments en avant et en arrière de l’image source. On observe sur la figure 2.30(c) des pixels bleu/gris dans la zone encadrée en jaune à la périphérie du trou du côté de la surface unie rouge (un zoom de cette zone est affichée dans un cadre jaune). De même, on trouve des pixels rouges sur la périphérie du trou du côté de l’objet bleu/gris. Pour les supprimer, on applique parfois une autre étape connue du DIBR appelée suppression des pixels fantômes (“ghost removal”) [Oliveira 2015], [Zinger 2010].

### 2.5.2.3 Génération de nouvelles vues par des techniques d’apprentissage automatique

Une approche naïve du DIBR basée sur les techniques d’apprentissage automatique consiste à calculer une carte de profondeur par apprentissage profond et d’appliquer les méthodes traditionnelles présentées plus haut pour le reste du processus. Cependant [Xie 2016] montre qu’un réseau de neurones dédié au DIBR surpasse ce

type de solution.

Ainsi, le concept le plus direct est d'estimer la valeur des pixels de la nouvelle vue associée à la caméra virtuelle en fournissant en entrée du réseau l'image source issue des caméras réelles et la pose ciblée de la caméra virtuelle, comme le propose [Tatarchenko 2016]. Néanmoins, [Zhou 2016] revendique que l'estimation directe des valeurs des pixels tend à fournir des images floues et conseille de prédire le flux optique de l'image cible. De la sorte, il ne reste plus qu'à copier les pixels depuis l'image source vers la nouvelle image à partir des données de flux estimées. Le second avantage de cette méthode est qu'elle peut prédire des zones ou faces d'un objet non observées en copiant plusieurs fois les pixels de l'image source. Ainsi, les objets d'apparence symétrique se prêtent bien à cette méthode.

Une autre solution appelée DeepStereo, proposée par [Flynn 2016], estime tout d'abord la profondeur, puis la valeur des pixels de l'image cible. Tout d'abord, la profondeur de la scène est discrétisée et  $n$  images cibles temporaires sont reconstruites avec l'hypothèse que la scène est un plan fronto-parallèle à la caméra à la  $k$ -ième distance (avec  $k \in [1, n]$ ). A partir de ces images, un réseau estime  $n$  cartes de probabilités indiquant pour chaque pixel une probabilité d'appartenir aux plans à la  $k$ -ième distance. Puis ces  $n$  cartes sont utilisées comme filtre de sélection appliqué aux  $n$  images cibles temporaires pour générer une image cible finale. Sur le même principe, DeepStereo peut reconstruire une nouvelle vue en se basant sur plusieurs images sources.

La référence [Xie 2016] présente Deep3D, une variante basée sur les travaux de [Flynn 2016]. Les auteurs cherchent à estimer une nouvelle vue à partir d'une seule image en exploitant plus de cinq millions d'images de film 3D comme base de données. Leur solution a pour avantage de fournir une carte de profondeur probabiliste intégrant un aspect d'"in-painting" et fournit des images cohérentes à 100 Hz sur GPU. Néanmoins, le réseau a été entraîné pour estimer, à partir d'une image perçue par un œil droit, l'image qui serait perçue par un œil gauche pour un écart inter-oculaire fixe. Cette banque de données d'apprentissage ne peut donc pas être reprise telle quelle sur un VST-HMD.

#### **2.5.2.4 Limite du DIBR**

La qualité de l'image reconstruite dépend essentiellement du changement de pose entre les caméras réelle et virtuelle et de la carte de profondeur, quand celle-ci est nécessaire. En effet, plus les caméras sont éloignées, plus il est difficile de construire une nouvelle vue. Les éléments/objets de la scène ne sont pas perçus de la même manière pour les deux points de vue : changement d'éclairage, manque de détails sur une texture, voire absence d'information causée par une occlusion. Une solution atténuant ce problème est d'utiliser des images issues de caméras réelles situées de part et d'autre des vues à reconstruire.

De plus, la qualité des cartes de profondeur impacte fortement la qualité de l'image finale, particulièrement si la vue calculée est éloignée des caméras d'origine. Pour améliorer les cartes de profondeur, il est possible d'appliquer des filtres bila-

téraux qui floutent l'image de profondeur dans les zones de gradient faible tout en conservant les contours des objets liés aux gradients forts. Pour les cartes de disparité calculées par stéréo-matching, [Kauff 2007] ajoute une étape de vérification appelée “consistency check”, où la disparité est évaluée et réestimée si la ressemblance entre les deux images n'est pas suffisante. L'idéal, cependant, est de mettre en place un outil fournissant les cartes de profondeur adaptées pour le DIBR à la fois dans les zones texturées et à la frontière entre plusieurs objets. En revanche, cette contrainte est moins forte dans les zones unies. Les cartes de profondeur reconstruites par [Xie 2016] tiennent compte de cette considération.

### 2.5.3 Conclusion

Cette section a présenté deux procédés de synthèse d'images : l'un pour l'incrustation d'objet synthétique dans la scène ; l'autre pour le rendu basé image, qui nous intéresse plus particulièrement pour le VST-HMD conçu au cours de la thèse et détaillé au chapitre 4. Les tests et le choix de la méthode de rendu basé images sont d'ailleurs approfondis au chapitre 4.

Deux de nos prototypes (VST-HMD et OST-HMD) affichent du contenu synthétique dans la scène à l'aide de la bibliothèque OpenGL.

## 2.6 Étalonnage du dispositif

Parmi les lunettes actives décrites au chapitre 1, certaines peuvent être adressables, c'est-à-dire qu'elles consistent en un système dont les propriétés du verre peuvent être modifiées localement. Suivant les applications visées avec ces dispositifs, il peut être nécessaire de sélectionner les zones des verres à activer pour que seule une portion ciblée du champ de vision soit affectée. Ainsi, pour un rayon passant par un point  $P$  de la scène et se projetant sur la rétine, on doit être capable d'agir sur le verre là où le rayon l'intersecte. La section précédente suppose que l'on connaît, pour chacun des yeux, une caméra virtuelle, ainsi que son modèle de projection. En pratique, la pose des caméras virtuelles et leur modèle de projection doivent être estimés lors d'une étape d'étalonnage. Cette section présente un état de l'art sur ces méthodes d'étalonnage.

L'étalonnage d'un tel système se décompose en plusieurs étapes :

- estimation du modèle de projection de chaque caméra ;
- estimation du changement de pose entre les verres actifs et les différents capteurs intégrés aux lunettes ;
- estimation du modèle de projection des verres.

Les verres adressables sur deux dimensions posés devant les yeux de l'utilisateur sont assimilés à des plans image de caméras. Ainsi, l'ensemble constitué du système visuel humain et des verres correspond à deux caméras, dont on doit déterminer les modèles de projection. Par ailleurs, ces caméras ne sont pas rigidement liées, c'est

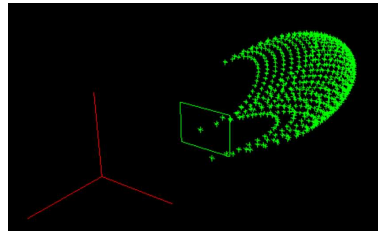


FIGURE 2.31 – Illustration de l’écran virtuel d’un HMD modélisé par [Owen 2004]. La modélisation suit une forme parabolique et non plane comme le propose le modèle trou d’épingle. (Extrait de [Owen 2004])

à dire que les lunettes peuvent bouger par rapport aux yeux de l’utilisateur et ceci doit être considérée dans le modèle de projection.

Cette section se focalise sur les systèmes d’affichage de type OST-HMD et VST-HMD pour présenter les méthodes d’étalonnage de lunettes actives. Par la suite, les techniques appliquées pour les HMDs pourront être transposées à d’autres systèmes actifs adressables sur deux dimensions (adaptation de l’obscurcissement du verre ou de la focale).

### 2.6.1 Modèle de projection

Le modèle de projection le plus couramment choisi est le modèle trou d’épingle ([Janin 1993], [Fuhrmann 1999], [Kato 1999], [Tuceryan 2002], [Gilson 2008], [Kellner 2012], [Itoh 2014]), dans lequel généralement le centre optique et le plan image sont respectivement définis comme le centre optique de l’œil et l’écran virtuel visible à travers les verres. En revanche, la méthode d’alignement de points (Single Point Active Alignment Method – SPAAM) de [Tuceryan 2002] fait abstraction de la notion d’écran virtuel et de centre optique en n’estimant que la matrice de projection, ce qui rend le problème facilement soluble.

[Owen 2004] montre que les optiques des HMDs génèrent des distorsions non modélisées par le modèle trou d’épingle, qui déforment la surface de l’écran virtuel supposé plan (Figure 2.31). Pour y remédier, [Hua 2002] propose d’intégrer des distorsions optiques radiales au modèle. [Lee 2015] les complexifie en ajoutant des distorsions tangentielles et des degrés de libertés supplémentaires aux distorsions radiales. [Owen 2004] modélise l’écran virtuel par une surface parabolique. [Itoh 2015b], quant à lui, introduit une fonction modélisant le flux lumineux, associant un rayon réfracté par les optiques du HMD à un rayon incident (Figure 2.32). Ainsi, pour faire correspondre un point de la scène à un pixel de l’afficheur, deux fonctions sont définies pour décrire chacune le trajet lumineux depuis l’écran vers l’œil et depuis le monde réel vers l’œil.

[Klemm 2016] et [Klemm 2017] conseillent de recourir à des modèles non paramétriques capables de modéliser les discontinuités des systèmes d’affichage complexe, et atteignent des résultats dépassant ceux du SPAAM.

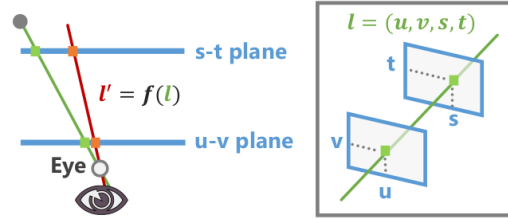


FIGURE 2.32 – Illustration du flux optique. Un rayon lumineux est modélisé par deux points 2D définis comme l’intersection entre le rayon et deux plans distants parallèles. La fonction  $f$  proposée par [Itoh 2015b] associe au rayon vert  $l$ , un autre rayon rouge réfracté  $l' = f(l)$  par les optiques du HMD. Ce rayon vert (resp. rouge) symbolise un trajet passant par l’œil sans réfraction du HMD (resp. après réfraction). (Extrait de [Itoh 2015a])

## 2.6.2 Étalonnage d’un OST-HMD

L’une des difficultés sur l’étalonnage d’un OST-HMD est l’acquisition des points 2D et 3D nécessaires pour l’estimation du modèle. En effet, les 2D points projetés ne peuvent pas être lus par l’ordinateur puisque la perception de l’utilisateur entre en jeu dans le processus de projection.

### 2.6.2.1 Méthodes basées sur l’alignement de points

La solution la plus répandue est l’alignement de points, comme le proposent [Janin 1993] et [Tuceryan 2002]. Lors de la procédure, l’utilisateur regarde à travers les lunettes et se déplace de manière à faire coïncider dans son champ de vision un point 2D affiché par le dispositif et un point 3D réel de la scène. De cette manière, de nombreuses correspondances sont acquises pour différents points 2D sur l’écran et pour des points 3D à différentes profondeurs. Par ailleurs, [Axxholt 2011] montre que, pour un étalonnage satisfaisant, la position des points 3D doit balayer tout le volume de travail. La méthode SPAAM proposée par [Tuceryan 2002] est souvent utilisée comme référence de comparaison pour les chercheurs, car elle offre des résultats précis et est facilement implémentable. Ses principaux points faibles sont la lenteur de la procédure, l’expertise requise par l’expérimentateur et la dépendance de l’étalonnage au positionnement des lunettes sur le visage. En effet, [Axxholt 2011] explique que lors du port des OST-HMD, ces derniers ont tendance à légèrement glisser, faussant ainsi l’étalonnage au bout d’un certain temps.

Pour accélérer la procédure et réduire la complexité de la phase d’acquisition, [Kellner 2012] et [Fuhrmann 1999] proposent d’aligner non plus deux mais trois points à chaque acquisition. Dans cette configuration, le porteur doit tendre un marqueur (localisé dans la scène par un système de caméra(s)) à bout de bras et se déplacer pour assurer sa superposition avec le point 2D de l’afficheur et le point 3D fixe dans la scène (Figure 2.33).



FIGURE 2.33 – Exemple d’alignement effectué lors de l’étalonnage proposé par [Kellner 2012]. L’utilisateur se déplace dans la scène de manière à superposer le marqueur posé sur le trépied, celui tenu à la main et le point 2D affiché par le casque. (Extrait de [Kellner 2012])

### 2.6.2.2 Méthodes basées caméras

Une autre méthode pour acquérir un ensemble de points 2D et 3D nécessaires à l’estimation du modèle consiste à positionner une caméra derrière les lunettes, à l’emplacement théorique des yeux du porteur. [Owen 2004] propose un étalonnage en deux phases. Une première étape hors ligne permet d’acquérir les paramètres intrinsèques du modèle de projection du dispositif en plaçant une caméra derrière les lunettes en différentes positions le long de l’axe horizontal. La caméra observe alors des points 2D affichés par les lunettes. Ses différentes positions latérales permettent d’estimer la position de l’écran virtuel et les paramètres du modèle de surface parabolique. La seconde phase vise à adapter l’étalonnage à l’écart inter-pupillaire du porteur, en déterminant la position latérale du centre de projection du système formé par les yeux et le HMD. [Gilson 2008] présente une méthode d’étalonnage similaire, qui ne nécessite pas d’intervention humaine. Ce choix permet d’éviter les erreurs de mesure dues à l’utilisateur. Pour obtenir un bon positionnement de la caméra dès le départ, les auteurs la positionnent à l’emplacement d’un œil dans la tête d’un mannequin (Figures 2.34 et 2.35). D’autre part, ils observent les points 3D de la scène à travers les optiques des lunettes, contrairement à [Owen 2004].



FIGURE 2.34 – Photo de la caméra montée dans la tête d’un mannequin à l’emplacement de l’œil gauche. (Extrait de [Gilson 2008])

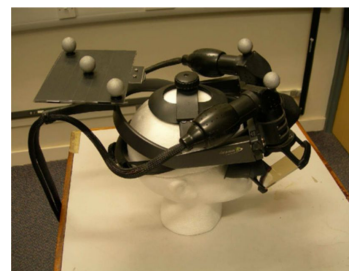


FIGURE 2.35 – Photo de la tête de mannequin équipée d’un HMD. La caméra est placée derrière les lunettes et dans la tête. (Extrait de [Gilson 2008])

Pour les modèles non paramétrique, il est nécessaire d’estimer le rayon de pro-

jection correspondant à chaque pixel de l’afficheur, ce qui exclut les méthodes d’étalonnage basées sur l’alignement de points. Pour ce faire, [Klemm 2016] positionne successivement une caméra derrière les lunettes à cinq positions différentes. Afin d’associer à chaque pixel de l’afficheur une position sur l’image de la caméra, une succession de motifs tel que ceux visibles sur la figure 2.36 sont affichés. En observant ces motifs depuis plusieurs poses de la caméra, il est possible de trianguler les positions 3D de chaque pixel de l’écran virtuel. Néanmoins, cette technique a le désavantage de ne pas être robuste aux systèmes optiques complexes basés guide d’onde<sup>22</sup>, car, pour différents points de vue de la caméra, les rayons passant par un pixel donné peuvent avoir des trajets optiques différents.



FIGURE 2.36 – Exemple de modulation de luminance par un motif sinusoïdal affiché par les afficheurs du HMD et observé par les caméras placées derrière les lunettes. (Extrait de [Klemm 2017])

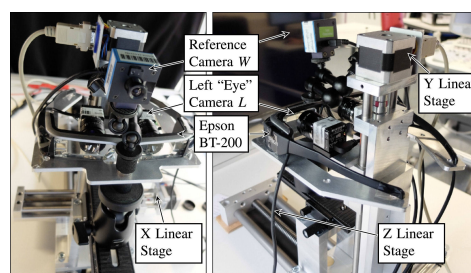


FIGURE 2.37 – Vue avant et arrière du banc d’étalonnage de [Klemm 2017]. Les caméras se déplacent suivant les axes  $X$  et  $Y$  par rapport aux lunettes et l’ensemble {lunettes, caméra} bouge le long de l’axe  $Z$ . (Extrait de [Klemm 2017])

Les auteurs améliorent le procédé dans [Klemm 2017] en modifiant la procédure d’étalonnage. Ainsi, au lieu de calculer les positions 3D des pixels de l’écran virtuel par triangulation, ils estiment la matrice de projection perspective pour une position fixe de l’œil. Pour modéliser les distorsions, ils appliquent sur l’image projetée une correction pixel par pixel, définie par une carte de distorsion. Ce modèle a l’avantage d’être facilement implémentable sur GPU à l’aide d’OpenGL, ce qui permet un rendu rapide. La procédure d’étalonnage reste assez similaire à [Klemm 2016], néanmoins un banc automatisé déplace la caméra et les lunettes (Figure 2.37). Par ailleurs, pour adapter la position de la caméra à celle de l’œil de l’utilisateur, les auteurs proposent d’effectuer neuf étalonnages pour différentes positions des caméras derrière les lunettes. Ensuite, pour de nouvelles positions de l’œil, les matrices de projection perspective et les cartes de distorsion correspondantes sont calculées

22. Il existe différentes technologies permettant d’acheminer la lumière émise par une source lumineuse vers les yeux de l’utilisateur. Les systèmes basés guide d’onde (“guided-wave optical combiner”) peuvent être constitués de nombreux éléments optiques, qui réfléchissent la lumière de nombreuses fois pour la guider jusqu’à l’œil. Les chemins empruntés par les rayons lumineux diffèrent les uns des autres. Suivant les dispositifs, les rayons ne traversent pas les mêmes éléments optiques avant d’atteindre l’œil. Par exemple, le système optique du casque Lumus DK50 de la société Lumus Optical présente cette particularité.

par interpolation sur les quatre plus proches positions étalonnées. Cette méthode obtient la meilleure précision angulaire parmi les techniques issues de l'état de l'art. Les performances sont proches de la résolution angulaire de l'œil humain avec une erreur entre 0.4 et 2.26 minutes d'arc contre de l'ordre de 0.6 à 0.8 minutes pour l'œil (acuité visuelle).

### 2.6.2.3 Méthodes avec adaptation dynamique au regard du porteur

Les auteurs de [Itoh 2014] sont les premiers à proposer un OST-HMD dont l'étalonnage s'adapte automatiquement à la position courante de l'œil délivrée par un oculomètre. Ils considèrent le système (œil plus affichage) comme un modèle trou d'épingle sans ajout de distorsion optique. Un premier étalonnage permet de connaître les paramètres intrinsèques et extrinsèques pour une pose du HMD donnée. Par la suite, la matrice de projection est modifiée en fonction de la pose de l'œil fournie par l'oculomètre.

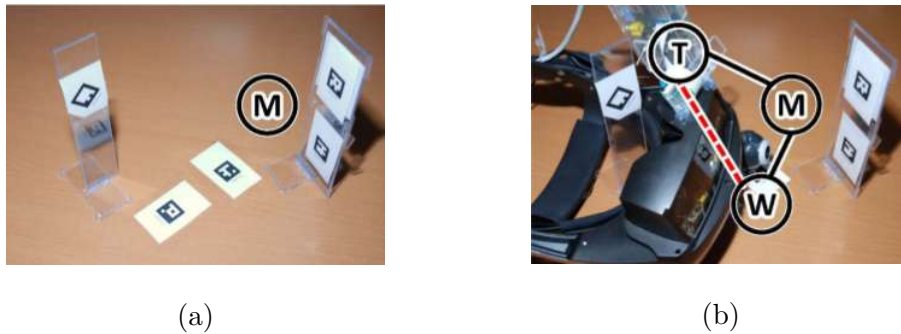


FIGURE 2.38 – (a) Photo de la table et des tags dispatchés. (b) Photo de la même configuration avec le HMD ajouté.  $M$ ,  $T$  et  $W$  désignent respectivement les tags, la caméra de l'oculomètre et celle filmant la scène. (Images extraites de [Itoh 2014])

Une méthode d'étalonnage des paramètres extrinsèques (Figure 2.38) est également proposée pour déterminer la transformation rigide entre la caméra  $W$  fixée sur le HMD filmant la scène et la caméra  $T$  de l'oculomètre. Pour y parvenir, des tags  $M$  sont disposés sur une table et localisés les uns par rapport aux autres avec une troisième caméra filmant l'ensemble sous différentes poses. Ensuite, le HMD est placé sur la table de sorte que chaque caméra observant au moins un tag puisse être localisée relativement à celui-ci. Ensuite, connaissant chaque transformation rigide entre les repères caméra et tag, on en déduit celle entre les deux repères caméras  $W$  et  $T$ .

### 2.6.3 Étalonnage d'un VST-HMD

L'étalonnage de VST-HMD est un sujet moins étudié par la communauté scientifique car il est difficile d'évaluer sa qualité par la superposition du contenu réel et virtuel. En effet, l'environnement réel n'est pas visible à travers un VST-HMD, ce qui empêche l'usage de techniques d'étalonnage dédiées aux OST-HMD.

Bien que l'erreur de superposition entre les éléments réels et virtuels ne soit pas perceptible avec un VST-HMD, l'étalonnage reste néanmoins nécessaire pour éviter une mauvaise évaluation de la distance des objets [Kuhl 2009] et des nausées ou inconforts ressentis par le porteur comme expliqué précédemment.

Des premières méthodes d'étalonnage ([Kuhl 2009], [Steinicke 2009]) ont été proposées en s'inspirant de celles définies pour les OST-HMD. Elles consistent à évaluer et corriger un paramètre du système, tel que le champ de vue ou la position de l'œil, en demandant au sujet d'itérativement porter et retirer le casque pour comparer des objets de référence dans la scène réelle avec le contenu synthétique affiché par le casque. Cependant, ces techniques ne permettent pas d'évaluer de nombreux paramètres simultanément et sont sujettes aux erreurs de perception humaine.

Pour modéliser les distorsions des VST-HMD, une autre approche se base sur la modélisation géométrique des systèmes optiques des HMD [Robinett 1993]. Par la suite, ces modèles sont repris pour définir une carte de distorsion applicable pixel par pixel sur l'image calculée par projection perspective [Bax 2004]. Ces calculs étant parallélisable sur GPU, les performances temporelles obtenues sont élevées et permettent d'utiliser des applications dynamiques de jeux par exemple. Néanmoins, ces méthodes nécessitent les données des constructeurs sur le HMD et les optiques, qui ne sont pas toujours disponibles. De plus ces paramètres peuvent varier d'un exemplaire à un autre et souvent les données fournies ne représentent pas des valeurs mesurées sur le système réel.

C'est pourquoi [Gilson 2011] propose d'adapter sur les VST-HMD la solution [Gilson 2008] pour OST-HMD. Le principe général est de positionner une caméra derrière le casque à la position théorique de l'œil. Ces deux systèmes sont localisés dans un référentiel commun de la scène. Le HMD affiche alors un damier, dont les coins sont détectés sur l'image fournie par la caméra, permettant par la suite d'associer les pixels de l'afficheur à ceux de la caméra. Ensuite, le casque est retiré et la caméra (qui elle est restée fixe) observe un damier mis en mouvement. De cette procédure, on peut donc extraire un ensemble de points 3D de la scène issus du damier et de points 2D associés sur l'afficheur par correspondance entre les pixels de la caméra et du HMD. Néanmoins, la technique pourrait être améliorée en effectuant la correspondance par affichage de motif comme [Klemm 2016].

Enfin, [Jones 2015] propose une solution semblable à celle de [Gilson 2011] pour laquelle les distorsions pour les trois longueurs d'onde (rouge, vert bleu) sont estimées. De plus, l'impact de la variation de la position de l'œil sur les distorsions observées est évalué. Le lecteur intéressé trouvera dans cet article un état de l'art approfondi sur l'étalonnage de casque VST-HMD.

#### 2.6.4 Conclusion

Un état de l'art des techniques d'étalonnage de casque (ou de lunettes) de réalité augmentée a été présenté dans cette section. L'étalonnage de casque est indispensable pour permettre une superposition acceptable du contenu synthétique affiché sur la scène réelle. Un prototype d'OST-HMD (Chapitre 4) a été mis au point pour

servir de banc de test, afin d'évaluer différentes techniques d'étalonnage. Sur ce principe, la méthode du SPAAM a déjà été mise en place et testée sur ce prototype. Le prototype de VST-HMD, le second casque de réalité augmentée conçu et étudié au cours de la thèse, a déjà été étalonné par le constructeur. Seules les poses relatives du banc stéréoscopique filmant la scène et de l'oculomètre par rapport au casque ont dû être estimés.

## 2.7 Conclusion

Ce chapitre a fourni une étude détaillée de différents composants essentiels pour les lunettes actives, qui nous a guidé pour la suite de nos recherches :

- les composants d'affichage pour générer du contenu synthétique dans le champ de vision de l'utilisateur,
- les composants de perception de l'environnement pour acquérir des données photométriques ou géométriques de la scène,
- les composants de localisation des lunettes pour construire une carte dynamique de l'environnement avec les données acquises par les composants de perception,
- les composants de détection et suivi du regard,
- les composants de synthèse d'image pour générer une image adaptée au point de vue de l'utilisateur, qui sera ensuite affichée par les lunettes,
- les composants d'étalonnage de système d'affichage pour permettre qu'un point affiché par les lunettes puisse être superposé à un point de la scène réelle.

Ces composants ont ensuite été implémentés dans des prototypes présentés au chapitre 4. Le choix des composants s'appuie sur l'état de l'art présenté dans ce chapitre et sur nos contraintes de moyen et de temps.

Un composant de suivi du regard basse consommation est proposé au chapitre 3, dans lequel il est testé et évalué sur des données de simulation.

Confidentiel

# Proposition d'un oculomètre basse consommation

## Sommaire

|                                                                      |            |
|----------------------------------------------------------------------|------------|
| <b>3.1 Prototypage virtuel d'un oculomètre basse consommation .</b>  | <b>92</b>  |
| 3.1.1 Exigences pour l'optique embarquée . . . . .                   | 92         |
| 3.1.2 Une solution basée photodiodes et apprentissage automatique    | 93         |
| 3.1.3 Simulation de l'oculomètre dans un environnement synthétique   | 95         |
| <b>3.2 Tests sur données synthétiques . . . . .</b>                  | <b>101</b> |
| 3.2.1 Définition des hyperparamètres du réseau . . . . .             | 102        |
| 3.2.2 Premiers résultats et analyse . . . . .                        | 115        |
| 3.2.3 Évaluation de différents assemblages de photodiodes . . . . .  | 120        |
| <b>3.3 Tests sur données réelles . . . . .</b>                       | <b>129</b> |
| 3.3.1 Banc de test et procédure d'acquisition de données . . . . .   | 129        |
| 3.3.2 Oculomètre basé caméras pour le banc de test . . . . .         | 131        |
| 3.3.3 Simulation des photodiodes à partir d'images réelles . . . . . | 132        |
| <b>3.4 Conclusion . . . . .</b>                                      | <b>132</b> |
| <b>3.5 Perspectives . . . . .</b>                                    | <b>133</b> |

Pour les applications visées, Essilor préfère centrer sa recherche sur des systèmes destinés à un usage quotidien et aussi discrets que des lunettes de vue classiques. Ainsi, ces systèmes doivent tenir compte des contraintes d'embarquabilité et plus particulièrement de consommation énergétique. Pour le confort de l'utilisateur, la batterie et les composants électroniques doivent être légers. Les capteurs ainsi que les traitements effectués par le calculateur intégré doivent être peu énergivores, pour permettre à la batterie de durer plus d'une journée (idéalement, une semaine). À titre d'exemple, Essilor utilise actuellement sur un prototype des batteries de 60mAh pour alimenter deux verres électro-chromes, chacun étant commandé par un micro-contrôleur. La masse et les dimensions des batteries sont respectivement d'environ 1g et 3×10×30mm, ce qui permet d'en disposer une dans chaque branche. Grâce à la faible consommation de ces composants, les lunettes présentent une autonomie supérieure à la semaine.

S'appuyant sur ce principe, cette section est consacrée au développement d'un oculomètre basse consommation, consécutivement à l'état de l'art ayant fait l'objet de la section 2.4. Il s'agit d'une problématique relativement peu étudiée, qui apporte des fonctionnalités intéressantes aux lunettes actives.

Tout au long de notre étude, l'oculomètre est supposé n'observer qu'un seul œil et estimer sa pose ou la direction qu'il pointe. Une variante intéressante, non étudiée ici, serait de définir un oculomètre en charge de l'estimation conjointe des données liés aux deux yeux, en tenant compte de contraintes supplémentaires telles que : l'intersection des directions de regard en un point et la distance inter-oculaire constante.

### 3.1 Prototypage virtuel d'un oculomètre basse consommation

Dans cette section, nous présentons tout d'abord le contexte et les exigences liées aux oculomètres embarqués. Puis, dans un second temps, nous proposons un oculomètre basse consommation basé photodiodes et apprentissage automatique, qui respecte les contraintes énoncées. Un tel système peut être configuré de différentes manières : nombre, caractéristiques et emplacements des photodiodes. Afin de nous affranchir des délais de fabrication et de tests de multiples réalisations matérielles, nous avons conçu et déployé un simulateur sur ordinateur. Ainsi, de nombreuses configurations peuvent être rapidement évaluées et comparées. Cette solution de prototypage virtuel fait l'objet de la troisième partie.

#### 3.1.1 Exigences pour l'optique embarquée

L'optique embarquée correspond ici à des dispositifs ophtalmiques dotés de fonctions actives pour un usage quotidien en tout lieu, que la personne soit fixe ou en déplacement, qu'elle soit en intérieur ou extérieur. En conséquences, ces systèmes intègrent de nombreuses exigences :

- *Taille, poids, apparence* : Comme le port de ces systèmes est permanent, leur apparence et leur poids doivent être similaires à ceux de lunettes de vue comme le propose [Bohn 2018].
- *Coût énergétique* : Les lunettes sont alors un système embarqué consommant de l'énergie et doivent donc répondre à des contraintes de coût énergétique en vue de satisfaire une autonomie d'au moins une journée pour le confort de l'utilisateur.
- *Simplicité d'utilisation* : Par ailleurs, la fonction d'oculométrie assurée doit être automatique et nécessiter le minimum de réglages de la part de l'utilisateur. Si un étalonnage est nécessaire pour adapter le système à un individu, alors il doit être simple et rapide. De plus, l'estimation doit être robuste au changement de pose des lunettes sur le nez de l'utilisateur.

A cause des contraintes de taille, de poids et de consommation énergétique, les lunettes actives ne peuvent pas inclure de caméras dont l'acquisition et le traitement des images consomment trop pour permettre un usage en continu sur plus d'une journée.

### 3.1.2 Une solution basée photodiodes et apprentissage automatique

#### 3.1.2.1 Sélection d'un assemblage de photodiodes

Pour proposer un oculomètre basse consommation, nous nous sommes basés sur les travaux de [Mayberry 2015] et [Topal 2014]. La référence [Mayberry 2015] démontre que dans une image, de nombreuses données sont redondantes et seuls quelques pixels sont utiles pour l'estimation de la direction du regard. Sur le même principe, [Topal 2014] montre que six photodiodes disposées sur le pourtour du verre suffisent pour déduire la position d'un point observé sur un écran. Dans ces travaux, l'expérience est menée dans un environnement stable sans variation de luminosité. D'autre part, aucun mouvement n'est considéré entre les lunettes et le visage de l'utilisateur. Ces deux restrictions ne correspondent pas à l'usage visé, que nous souhaitons atteindre avec notre proposition d'oculomètre basse consommation.

Notre proposition d'oculomètre basse consommation est constituée d'un assemblage de photodiodes orientées vers l'œil que l'on dispose sur le pourtour des verres de lunettes dans le but de les dissimuler dans la monture (Figure 3.1). Toutefois, il est nécessaire d'investiguer parmi les assemblages possibles ceux qui sont les plus adaptés au besoin :

- Quel nombre de photodiodes choisir ? Vraisemblablement plus le nombre de photodiodes est élevé, plus le résultat est précis. Néanmoins, la complexité et le coût du système augmentent.
- Quel angle d'ouverture pour chacune des photodiodes ? Pour choisir un angle d'ouverture, les photodiodes peuvent être diaphragmées ou équipées de lentilles. Avec de grands angles d'ouverture, le système total sera capable de couvrir toute la surface de l'œil. Cependant, les photodiodes risquent d'observer la scène autour du visage.
- Quelle position affecter aux photodiodes le long du contour du verre ? Est-ce utile d'en disposer à droite et à gauche du verre, là où elles sont les plus éloignées de l'œil ?
- Comment orienter les photodiodes ? Doivent-elles toutes pointer vers un même point ?
- Quelle bande spectrale utilisée ? Doit-on mesurer dans le visible, ou bien dans l'infrarouge où la pupille apparaît nettement plus noire que le reste du visage ? Pour son premier prototype, [Mayberry 2014] a considéré le visible, mais s'est tourné vers l'infrarouge pour le second [Mayberry 2015].
- Faut-il embarquer un dispositif d'éclairage pour obtenir un capteur actif, ou se contenter de l'éclairage présent dans l'environnement (capteur passif) ?
- Est-ce que la forme du verre (grand, petit, rond, rectangulaire) a un impact sur le résultat ? Il est possible que, pour certains verres, le point de vue des photodiodes n'offre pas une bonne observation de l'œil.

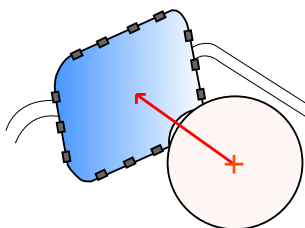


FIGURE 3.1 – Illustration d'un assemblage de photodiodes disposées sur le pourtour d'un verre de lunettes. L'œil est représenté ici par une sphère et la direction du regard par une flèche rouge.

Un des objectifs de cette étude est de choisir un assemblage de photodiodes répondant à ces contraintes. Pour ce faire, plusieurs assemblages doivent être testés. Une approche possible est de fabriquer des échantillons et de les tester en conditions réelles. Cependant, cette approche est longue et fastidieuse. Pour éviter cela, nous choisissons plutôt de simuler ces assemblages en considérant un maillage 3D texturé pour le visage. Une démarche exploratoire nous permet alors de tester et comparer de nombreux agencements de photodiodes.

### 3.1.2.2 Synthèse d'un algorithme pour estimer la direction du regard

Afin de limiter l'encombrement, il est préférable de diminuer autant que possible le nombre de photodiodes. Cependant, les algorithmes traditionnels d'estimation de la direction du regard basés sur l'extraction d'éléments caractéristiques de l'œil (Section 2.4) ne peuvent pas fonctionner sur un petit nombre de photodiodes (par exemple 9 ou 25 photodiodes). De plus, leur coût calculatoire et leur coût en mémoire sont prohibitifs si on souhaite exploiter un simple microcontrôleur.

D'autre part, pour notre étude, de nombreux assemblages de photodiodes doivent être testés. Or, nous ne souhaitons pas développer des algorithmes spécifiques pour chaque oculomètre. Notre algorithme doit donc apprendre par lui-même comment traiter au mieux les données, afin qu'il soit capable de s'adapter à d'autres assemblages de photodiodes. Nous nous orientons donc vers des outils généralistes d'apprentissage automatique pouvant s'adapter à des assemblages quelconques sans nécessiter de modification. Toutefois, ceux-ci requièrent une étape d'apprentissage, au cours de laquelle les paramètres du modèle sous-jacent sont estimés. Ainsi, pour effectuer cette étape d'apprentissage, il faut réunir une base de données d'entrée et de sortie du modèle, qui sont respectivement les valeurs fournies par les photodiodes et la direction du regard (ou pose de l'œil). La constitution d'un tel jeu de données peut parfois être longue et fastidieuse.

Un algorithme remplissant les contraintes de faible coût calculatoire et d'adaptabilité est la régression polynomiale. La référence [Kassner 2014] utilise cette méthode (décrite section 4.1.4.1) pour estimer les coordonnées 2D d'un point (point ciblé sur l'image d'une caméra de scène) à partir des coordonnées d'un autre point (centre de l'image de la pupille sur l'image de la caméra de l'oculomètre). Dans

notre cas, l'algorithme estime l'orientation de l'œil (ou bien son orientation et sa position) à partir des signaux issus des photodiodes. Ainsi, si l'on utilise cette méthode, une fonction polynomiale doit être définie pour chaque quantité à estimer, avec pour variables d'entrée les données délivrées par les photodiodes. Cette solution nécessite une étape d'étalonnage, qui consiste à estimer les coefficients des fonctions polynomiales. Pour effectuer cet étalonnage, seules quelques mesures suffisent.

Cependant, nous cherchons à résoudre un problème complexe où les lunettes sont susceptibles de bouger, où la luminosité de l'environnement peut varier et où la forme du visage diffère d'un utilisateur à un autre. L'outil mis en place doit donc être capable de s'adapter à toutes ces variables. Or, les fonctions polynomiales ne suffisent pas pour modéliser un problème aussi complexe. C'est pourquoi nous avons choisi d'effectuer la régression par un réseau de neurones (Figure 3.2). Ce dernier présente l'avantage de pouvoir modéliser des systèmes complexes, d'être suffisamment générique pour s'adapter à d'autres assemblages moyennant un nouvel étalonnage et de s'implémenter rapidement, en quelques lignes de code, grâce à des outils logiciels adaptés. Pour notre étude, nous utilisons scikit-learn<sup>1</sup> et Keras<sup>2</sup> avec le backend Tensorflow<sup>3</sup>. Néanmoins cette technique requiert un jeu de données important, de plusieurs dizaines de milliers d'échantillons pour notre cas. En simulation, ces données peuvent facilement être recueillies en grand nombre, car le processus de synthèse est entièrement automatique. En revanche, pour constituer un jeu de données permettant l'étalonnage d'un oculomètre physique, une solution est proposée section 3.3.

### 3.1.3 Simulation de l'oculomètre dans un environnement synthétique

Pour tester et évaluer différents oculomètres, nous proposons d'effectuer du prototypage virtuel. Pour cela, un outil de simulation a été conçu, où les assemblages de photodiodes peuvent être définis avec de nombreux degrés de liberté. Les photodiodes observent un environnement synthétique constitué d'un visage et d'un œil pouvant être mis en rotation. Dans un premier temps, la section 3.1.3.1 explique comment l'environnement (le visage et l'œil) est synthétisé. Puis, dans un second temps, la section 3.1.3.2 décrit les paramètres configurables de l'assemblage de photodiodes et détaille le procédé de simulation des données fournies par des photodiodes.

---

1. Scikit-learn (<http://scikit-learn.org/>) est une bibliothèque logicielle Python open-source proposant de nombreux outils d'apprentissage automatique, dont la régression polynomiale et les réseaux de neurones. Cette bibliothèque ne propose pas tous les outils dont nous avons besoin. Nous l'avons donc utilisée en complément d'une autre bibliothèque.

2. Keras (<https://keras.io/>) est une API Python open-source s'ajoutant aux bibliothèques de réseau de neurones (TensorFlow, CNTK ou Theano). Elle permet une utilisation simplifiée de ces outils. Elle peut également être utilisée avec scikit-learn, qui propose de nombreux outils d'analyse pour l'évaluation et la comparaison de réseaux de neurones.

3. Tensorflow (<https://www.tensorflow.org/>) est une bibliothèque logicielle Python open-source proposant des outils d'apprentissage automatique. Elle permet de définir des architectures de réseau de neurones variées et dispose de nombreux paramètres et outils associés.

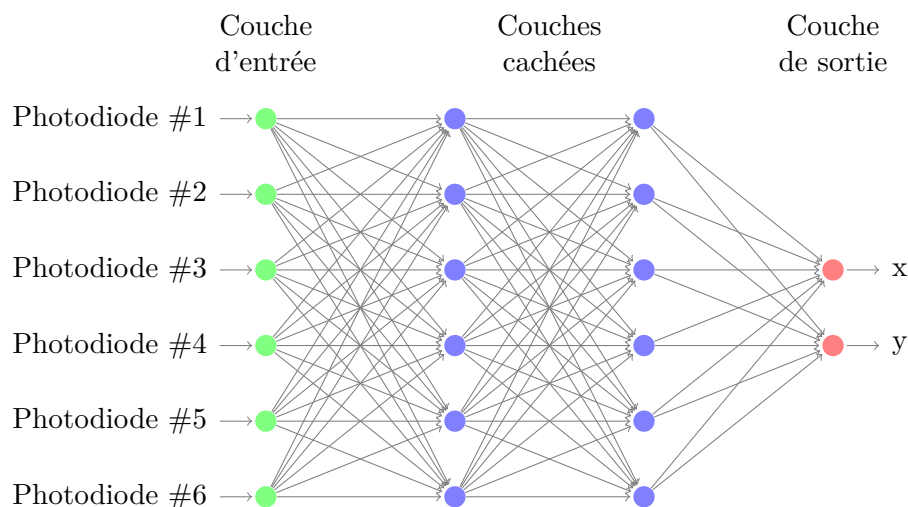


FIGURE 3.2 – Représentation schématique d'un réseau de neurones estimant la direction du regard sur la base de six photodiodes. Le réseau est constitué d'une couche d'entrée (en vert) alimentée par les mesures issues des six photodiodes, de deux couches cachées (en bleu) et d'une couche de sortie (en rouge) fournissant une estimée des deux degrés de liberté en rotation de l'œil.

### 3.1.3.1 Synthèse d'images du visage et de l'œil

Pour synthétiser le visage d'une personne et son œil pointant vers diverses directions, nous avons choisi d'utiliser Blender, un outil permettant de générer des vues synthétiques sur la base de maillages 3D texturés. En plus de son interface graphique, cet outil est configurable par script python, rendant possible la génération automatique de milliers d'images, en faisant varier différents paramètres (orientation du regard, illumination, etc).

Le modèle du visage (Figure 3.3) a été repris des travaux de [Swirski 2013]. Il est configuré de façon à générer des mouvements du visage et des yeux visuellement

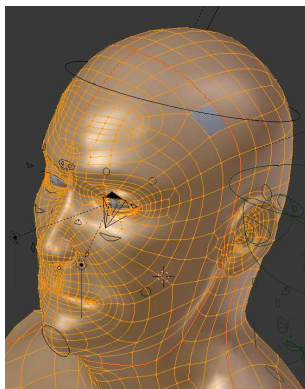


FIGURE 3.3 – Illustration du maillage 3D du visage repris de [Swirski 2013]

cohérents : par exemple, on observe sur les images de la figure 3.4 qu'une rotation de l'œil déforme les paupières inférieure et supérieure. La texture du modèle est elle aussi reprise des travaux de [Swirski 2013]. Elle permet de simuler l'apparence du visage en infrarouge. En cas de besoin, le logiciel peut aussi simuler dans la lumière visible avec une autre image de texture. L'éclairage lumineux est assuré par deux sources ponctuelles situées de part et d'autre de l'œil.

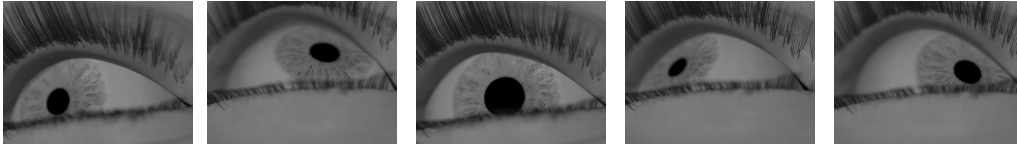


FIGURE 3.4 – Exemple d'images générées par Blender à partir du modèle de [Swirski 2013], où l'orientation de l'œil et la taille de la pupille varient.

La base de données d'apprentissage doit être aussi variée que possible et couvrir des configurations très différentes. Ainsi, après apprentissage, le réseau sera plus à même de gérer un large éventail de situations (capacité à généraliser). Pour cela, le modèle de l'œil sous Blender intègre des paramètres modifiables tels que son orientation ou la taille de sa pupille. L'environnement peut lui aussi être configuré en modifiant les sources lumineuses : nombre, positions, natures (ponctuelles, surfaciques, à l'infini) et intensités. De plus, les propriétés liées à la texture de l'œil peuvent également être modifiées pour générer des reflets cornéens, ce qui n'est pas proposé au départ sur le modèle de [Swirski 2013].

### 3.1.3.2 Simulation de l'assemblage de photodiodes

**A Assemblage de photodiodes** L'assemblage de photodiodes est en premier lieu contraint par la forme et la taille du verre. Ainsi, chaque photodiode est placée et orientée individuellement sur le pourtour de celui-ci. Elle peut être munie d'une lentille ou d'un diaphragme afin d'ajuster son angle d'ouverture, et d'un filtre afin de modifier la forme du cône de lumière qu'elle perçoit. Nous appellerons par la suite ce filtre “filtre de forme”. Par exemple, le filtre de forme de la figure 3.5 privilégie la direction de perception définie par la fente. Toutefois, il ne faut pas que cette fente soit trop étroite, sous peine d'apparition d'un phénomène de diffraction.

Plusieurs paramétrisations de l'assemblage de photodiodes sont possibles. Nous proposons la suivante (Figure 3.6) :



FIGURE 3.5 – Exemple de filtre de forme appliqué devant les photodiodes.

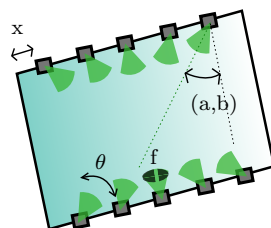


FIGURE 3.6 – Exemple d'assemblage de photodiodes : 5 d'entre elles sont placées sur le bord supérieur et 5 autres sont placées sur le bord inférieur. Leurs champs de vue sont illustrés par des cônes verts. Chaque photodiode est définie par les paramètres (ouverture  $\theta$  ; position  $x$  ; orientation  $(a,b)$  ; filtre  $f$ ).

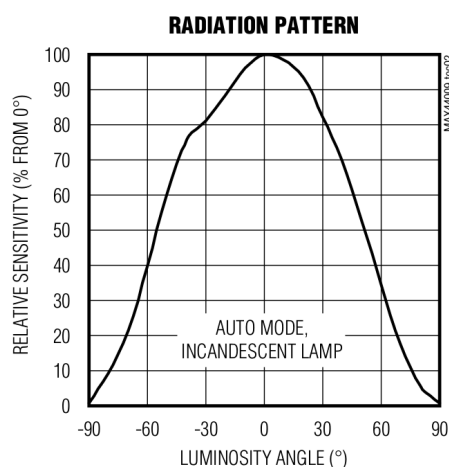


FIGURE 3.7 – Sensibilité d'une photodiode (MAX 44009) en fonction de l'angle d'incidence du rayon (Extrait de la datasheet du composant).

- liste de points 3D, exprimés dans un repère associé à l'oculomètre, définissant le contour du verre ;
- nombre  $n$  de photodiodes ;
- positions  $x_i$ ,  $i \in \{1, \dots, n\}$ , des photodiodes sur le contour du verre (d'autres possibilités peuvent néanmoins être envisagées) ;
- orientations  $(a_i, b_i)$ ,  $i \in \{1, \dots, n\}$ , des photodiodes suivant deux directions ;
- angles  $\theta_i$ ,  $i \in \{1, \dots, n\}$ , de champ de vue des photodiodes ;
- filtres nommés  $f_i$ ,  $i \in \{1, \dots, n\}$ , devant les photodiodes.

Par ailleurs, pour la génération de la base de données, l'assemblage peut être placé en différentes positions et orientations par rapport au visage, dans le but de considérer les déplacements des lunettes sur le nez du porteur.

**B Photodiode** Une photodiode est un dispositif analogique qui intègre le flux lumineux et le convertit en une valeur numérique, que nous souhaitons simuler. Pour

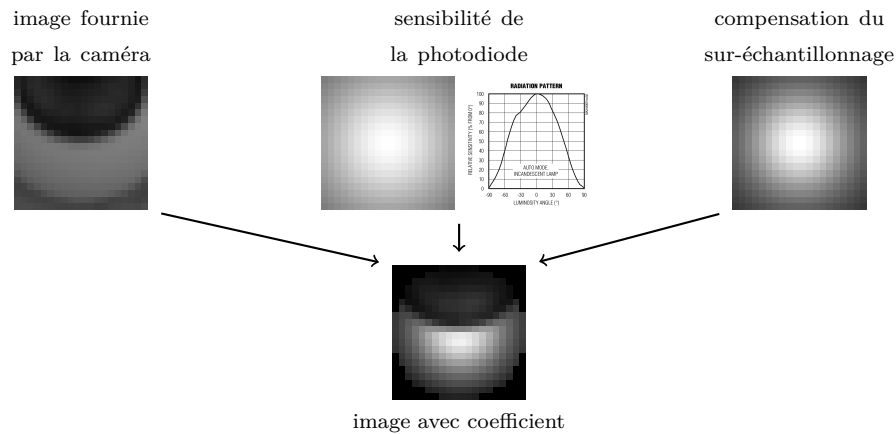


FIGURE 3.8 – Processus de simulation de la mesure fournie par une photodiode. On multiplie pixel par pixel l'image synthétisée par Blender (haut gauche) avec une première matrice de coefficients modélisant la sensibilité angulaire de la photodiode (haut milieu), avec une deuxième matrice compensant le sur-échantillonnage angulaire causé par la projection de l'image sur un plan plutôt qu'une sphère (haut droite) et avec un masque délimitant le cône d'observation.

cela, une caméra est définie dans Blender pour chaque photodiode. Sa position, son orientation et son angle d'ouverture correspondent à ceux de la photodiode. Les images délivrées par cette caméra permettent de simuler les mesures de la photodiode. Toutefois, comme nous ne pouvons pas définir de caméra un pixel sous Blender, la mesure est simulée à partir d'une image de 10x10 pixels à laquelle on applique des filtres particuliers :

- masque modélisant l'ouverture circulaire ;
- filtre de forme ;
- sensibilité angulaire de la photodiode, fonction du rayon incident (Figure 3.7) ; un exemple de filtre est visible sur l'image centrale en haut de la figure 3.8 ;
- compensation du sur-échantillonnage angulaire induit par l'utilisation d'un modèle trou d'épingle pour la synthèse de l'image par Blender : chaque pixel de l'image ne couvre pas un angle solide de même taille, car le modèle trou d'épingle consiste à générer une image par projection sur un plan et non sur une sphère ; si on somme simplement les valeurs de chaque pixel, des portions du champ de vision de la photodiode seront sur-représentées ; un exemple de filtre est présenté en haut à droite de la figure 3.8.

Pour déduire la mesure fournie par la photodiode, ces filtres définis dans des matrices 10x10 sont appliqués sur l'image synthétisée par Blender en multipliant élément par élément les filtres avec l'image.

Pour la suite du chapitre, nous considérons trois repères : un pour l'oculomètre supposé solidaire des lunettes, un pour le visage et un autre pour l'œil. Le modèle géométrique de l'œil considéré ici est simplifié : le centre optique est confondu avec

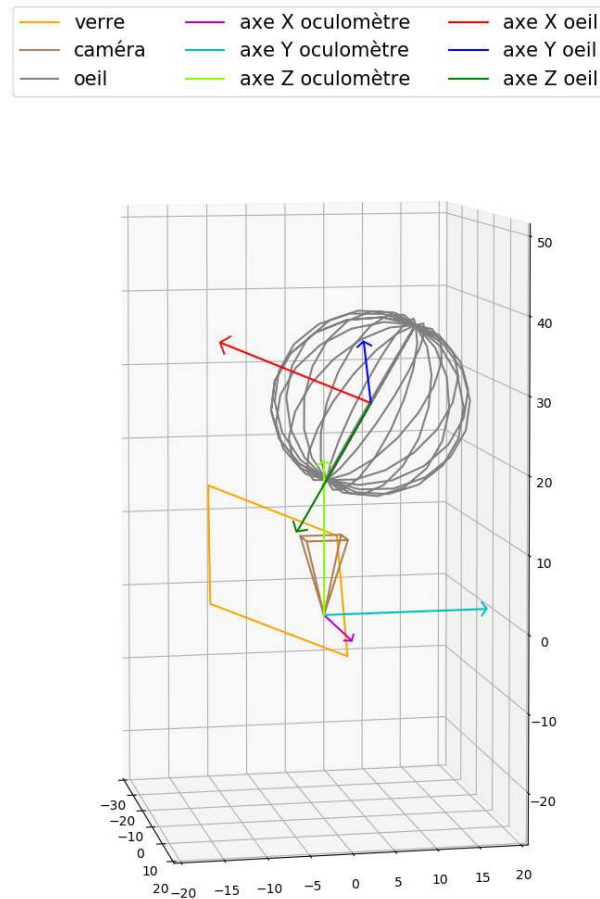


FIGURE 3.9 – Représentation de l'enveloppe de l'œil par une sphère (en noir) et du verre de lunettes par son contour (en jaune). Les axes  $X$ ,  $Y$  et  $Z$  associés à l'œil (resp. à l'oculomètre) sont indiqués en rouge, bleu et vert (resp. magenta, cyan et vert clair). En marron est indiquée la caméra fictive définie par les auteurs de [Swirski 2013] pour leur oculomètre basé caméra.

le centre de rotation et l'axe pupillaire est confondu avec la direction du regard.

- Le repère de l'oculomètre  $R_{oculomètre}$  a pour centre  $O_{oculomètre}$  et pour axes  $(X_{oculomètre}, Y_{oculomètre}, Z_{oculomètre})$ . L'oculomètre est susceptible de bouger par rapport au visage, car l'utilisateur peut retirer et remettre ses lunettes sans les replacer exactement au même endroit. Lors de la simulation, nous considérons un déplacement de l'oculomètre de  $\pm 4mm$  suivant les axes  $X_{oculomètre}$  et  $Z_{oculomètre}$  et une inclinaison de 0.1 radian autour de  $X_{oculomètre}$  et  $Y_{oculomètre}$ .
- Le repère du visage  $R_{visage}$  a pour centre  $O_{visage}$  et pour axes  $(X_{visage}, Y_{visage}, Z_{visage})$ . Il correspond au repère monde de Blender car le modèle du visage ne bouge pas.
- Le repère de l'œil  $R_{œil}$  a pour centre  $O_{œil}$ , le centre de rotation de l'œil (assimilé ici à son centre optique), et pour axes  $(X_{œil}, Y_{œil}, Z_{œil})$ . Pour l'étude menée dans ce chapitre, l'axe  $X_{œil}$  correspondant à l'axe pupillaire est considéré comme la direction du regard (direction de l'œil) par simplification. Ce dernier est en mouvement de rotation et aucune translation n'est considérée entre le visage et l'œil.

Les repères  $R_{oculomètre}$  et  $R_{œil}$  sont visibles sur le schéma 3.9.

L'oculomètre doit estimer la direction du regard (ou la pose de l'œil) exprimée dans le repère de l'oculomètre en minimisant une fonction de perte (définie section 3.2.1.4) appliquée aux directions estimées et réelles du regard (ou aux directions estimées et réelles du regard et aux positions estimées et réelles de l'œil)

En sortie du réseau de neurones, la direction est représentée par deux variables : les projections sur les axes  $X_{oculomètre}$  et  $Y_{oculomètre}$  du vecteur unitaire supportant la direction du regard. Cependant, cette approche présente deux inconvénients. D'une part, on ne peut considérer que le cas où la composante sur  $Z_{oculomètre}$  du vecteur directeur désigné par  $X_{œil}$  est négative (voir le schéma 3.9), ce qui est toujours le cas dans nos simulations. D'autre part, les variations locales des projections sur  $X_{oculomètre}$  et  $Y_{oculomètre}$  n'évoluent pas linéairement en fonction des variations angulaires locales de la direction du regard. Ainsi la mesure d'erreur n'a pas le même impact suivant la direction du regard.

La pose de l'œil est représentée en sortie du réseau par les deux composantes sur  $X_{oculomètre}$  et  $Y_{oculomètre}$  du vecteur unitaire supportant la direction du regard et par la position 3D de  $O_{œil}$  dans le repère  $R_{oculomètre}$ .

## 3.2 Tests sur données synthétiques

Cette section regroupe tous les résultats des tests sur données synthétiques. Tout d'abord, le réseau estimant la direction du regard et sa procédure d'apprentissage sont paramétrés pour un assemblage de photodiodes donné : 20 capteurs de 10 degrés d'ouverture disposés en haut et en bas d'un verre rectangulaire, tous orientés vers un même point derrière la pupille (Figure 3.10). Pour sélectionner une paramétrisation satisfaisante, de nombreuses paramétrisations candidates sont envisagées, et celle qui conduit aux meilleures performances est conservée (Section 3.2.1). Le

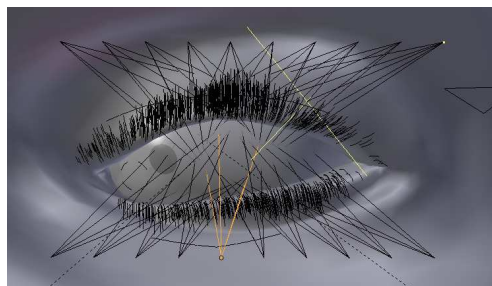


FIGURE 3.10 – Image de l’assemblage de photodiodes de référence, issue de l’interface graphique de Blender. Les photodiodes sont représentées par des pyramides, et sont positionnées en les sommets de celles-ci.

réseau de neurones est alors testé et évalué exhaustivement (Section 3.2.2). Ensuite, dans l’optique d’améliorer les performances de l’estimation de la direction du regard, nous varions les assemblages des photodiodes. Nous faisons l’hypothèse que la paramétrisation du réseau et de sa procédure d’apprentissage restent satisfaisantes quels que soient ces assemblages, et présentons les résultats d’estimation (Section 3.2.3).

### 3.2.1 Définition des hyperparamètres du réseau

Dans une étape préliminaire, le réseau de neurones et la procédure d’apprentissage nécessitent d’être paramétrés : les données d’entrée peuvent nécessiter d’être prétraitées, les paramètres des calculs internes aux nœuds doivent être définis et la manière d’effectuer l’apprentissage doit aussi être spécifiée. Ces degrés de liberté seront appelés “hyperparamètres” du réseau, à ne pas confondre avec ses paramètres définis comme les poids associés aux nœuds et les biais (définis plus tard dans cette section).

Après une brève description des hyperparamètres pouvant être ajustés, la sous-section 3.2.1.1 présente les outils considérés pour l’évaluation des performances des réseaux de neurones. Ceci nous permet de tester et comparer différentes paramétrisations, afin de choisir une paramétrisation satisfaisante.

Le premier hyperparamètre à définir concerne le prétraitement des données d’entrée, que l’on appelle couramment “standardisation” ou “normalisation” [Sola 1997]. Différentes méthodes existent. Elles sont exposées à la section 3.2.1.2. Cette étape permet de redéfinir l’intervalle de définition des variables pour limiter les erreurs numériques sur les calculs.

D’autres hyperparamètres sont liés aux opérations effectuées au sein de chaque nœud (figure 3.11). Un poids  $w_i$  est associé à chaque entrée du nœud et s’applique sur les valeurs calculées  $x_i$  par les neurones placés en amont. Ensuite, l’ensemble de ces données pondérées ainsi que le biais  $b$  sont sommés et une fonction d’activation  $f$  leur est appliquée. En régression, la fonction  $f$  définie pour les nœuds de la couche de sortie est la fonction identité. On distingue donc deux hyperparamètres permettant de paramétrer le réseau : la méthode d’initialisation des poids et le choix de la

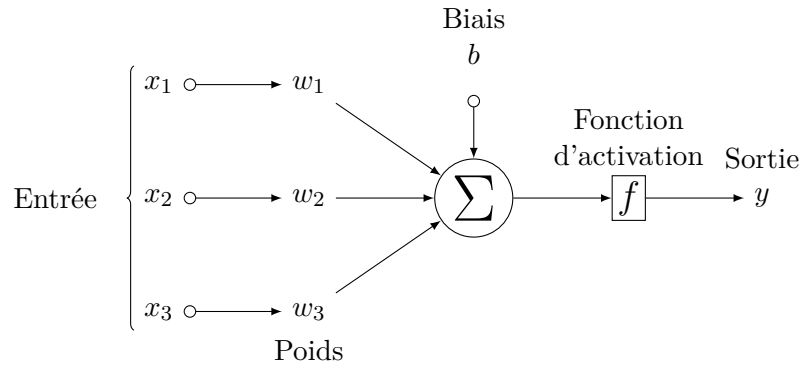


FIGURE 3.11 – Calculs effectués sur un nœud d’un réseau de neurones.  $x_1$ ,  $x_2$  et  $x_3$  sont les sorties des nœuds précédents auxquelles ce nœud est connecté en amont. On multiplie ces valeurs par des poids  $w_i$ . Puis, l’ensemble des données pondérées ainsi qu’un biais  $b$  sont sommés et une fonction d’activation  $f$  est appliquée. La valeur  $y$  déduite en sortie de ce nœud constitue la sortie du réseau ou bien l’entrée des nœuds placés en aval.

fonction d’activation.

Lors de l’apprentissage, les pondérations  $w_i$  et les biais  $b$  sont estimés de manière à minimiser une fonction de perte  $f_{\text{perte}}(y_{\text{cible}}, y_{\text{estime}})$  caractérisant l’erreur entre le résultat estimé  $y_{\text{estime}}$  et la valeur ciblée  $y_{\text{cible}}$  issue de la base de données enregistrée. Avec l’outil Keras, plusieurs fonctions de perte peuvent être testées, dont l’erreur absolue et l’erreur quadratique. Nous testons différentes fonction de pertes, dont une que nous avons spécifiquement exprimée pour notre problème dans la section 3.2.1.4.

Un autre hyperparamètre lié à la procédure d’apprentissage influe sur le résultat : il s’agit du nombre d’itérations du processus d’optimisation sur l’ensemble de la base de données. Une itération est souvent désignée par le terme “époque” (epoch). La taille de la base de données d’apprentissage a elle aussi une influence. Par ailleurs, l’architecture du réseau a également un impact : nombre de couches cachées et de nœuds par couches, organisation des connexions. Plus le nombre de couches cachées et de nœuds du réseau est élevé, plus celui-ci peut modéliser des phénomènes complexes. Cependant, le coût calculatoire et la consommation énergétique augmentent eux aussi.

Il est possible de tester toutes les paramétrisations possibles. Toutefois comme de nombreux hyperparamètres doivent être définis, le temps nécessaire au test serait très long. Nous avons donc choisi, pour la plupart des hyperparamètres, de les évaluer indépendamment les uns des autres, de manière alternée.

### 3.2.1.1 Méthodes pour l'évaluation et la comparaison de réseaux de neurones

Pour évaluer le réseau de neurones après apprentissage, une fonction de score compare le résultat estimé et les valeurs issues de la base de données. Dans notre cas, nous utilisons la “variance expliquée” définie par l'équation 3.1 et l'appliquons aux  $n$  données estimées et  $n$  données réelles ciblées notées respectivement par  $y_{estime_k}$  et  $y_{cible_k}$  pour  $k \in \{1, \dots, n\}$ . Cette dernière dépend de l'erreur quadratique moyenne relativement à la variance des données. Le principe est de réduire l'impact d'une erreur forte lorsque la variance des données est élevée, et inversement, d'amplifier l'importance d'une erreur faible lorsque la variance est faible. Ce score, tel qu'il est exprimé, s'interprète comme suit : son maximum de 1 correspond à une correspondance parfaite entre les données estimées et les données réelles et plus le score est bas, plus l'estimation s'éloigne des valeurs ciblées. Nous utiliserons ce score pour comparer les performances des réseaux les uns par rapport aux autres. Cependant pour évaluer un réseau seul, nous préférons considérer l'erreur angulaire absolue moyenne sur un jeu de données de validation, car cette mesure est plus parlante.

$$score = 1 - \frac{\sum_{k=1}^n (y_{cible_k} - y_{estime_k})^2}{\sum_{k=1}^n (y_{cible_k} - \mu_{cible})^2} \quad (3.1)$$

où  $\mu_{cible} = \frac{1}{n} \sum_{k=1}^n y_{cible_k}$

La variance expliquée permet d'évaluer et de comparer des réseaux de neurones après apprentissage sur des bases de données spécifiques. Deux autres pratiques essentielles ont été appliquées ici pour effectuer ces comparaisons. La base de données initiale a été divisée en deux sous-ensembles<sup>4</sup> : un premier sur lequel s'effectue l'apprentissage, et un second de plus petite taille pour l'évaluation du réseau. Le principe de cette séparation est d'évaluer la capacité du réseau à extrapoler sur de nouvelles données différentes de celles sur lesquelles il a été optimisé.

D'autre part, comme les performances du réseau dépendent significativement des données d'apprentissage, il est préférable de tester plusieurs apprentissages sur différents jeux de données. C'est pourquoi nous avons recours au “K-fold” pour l'évaluation [Kohavi 1995]. Le tableau 3.1 illustre cette technique. La base de données initiale est divisée en  $K$  parties, que nous appellerons “fold”.  $K - 1$  sous-ensembles sont utilisés pour l'apprentissage et le sous-ensemble restant sert à la validation. Cette étape est répétée  $K$  fois en permutant le sous-ensemble utilisé pour l'évaluation. La performance du réseau est alors associée au score moyen sur l'ensemble des  $K$  procédures.

---

4. Il arrive également que la base de données soit divisée en trois parties : une pour l'apprentissage, une pour la validation et une pour le test. Le principe est alors de comparer différentes paramétrisations de réseaux de neurones sur la base de données de validation puis d'évaluer une dernière fois la meilleure paramétrisation sur la base de test. Pour notre étude, nous ne considérerons que deux sous-ensembles de données : celui d'apprentissage et celui de validation. Lors de l'évaluation du réseau de neurones avec la paramétrisation finale choisie, nous conserverons le vocable “validation”, au lieu de “test” par simplification pour le lecteur.

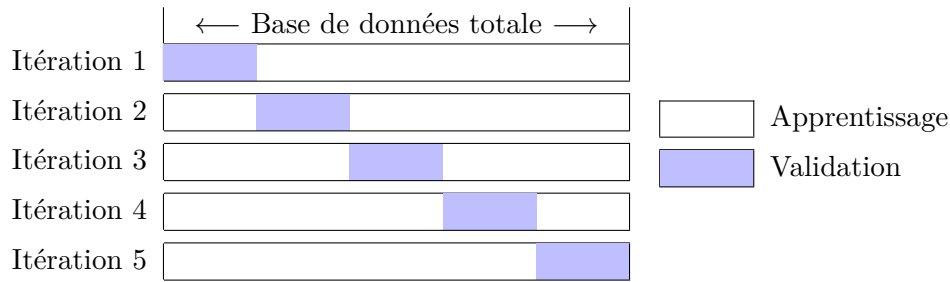


TABLE 3.1 – Schéma illustrant le principe du K-fold. La base de données initiale est divisée en  $K$  sous-ensembles (ici cinq). Pour chaque itération, un ensemble d'apprentissage (en blanc) est construit avec  $K - 1$  sous-ensembles et celui restant est réservé à la validation (en bleu). Cette procédure est effectuée  $K$  fois pour différents ensembles de validation.

La base d'apprentissage sur cette section 3.2.1 comporte plus de 45000 séries de valeurs de photodiodes et directions du regard. Comme expliqué précédemment, pour nos tests nous divisons nos données en deux parties (dans des proportions usuelles) : 90% pour l'apprentissage et 10% pour la validation.

### 3.2.1.2 Prétraitement des données d'entrée

Le tableau 3.2 montre les différentes méthodes de prétraitement proposées par Scikit-learn que nous avons appliquées sur nos données d'entrée. Chacune de ces méthodes s'applique sur un vecteur  $\mathbf{x}_i$  composé des mesures  $x_{ij}$  fournies par la  $i$ -ème photodiode pour toutes les poses, indicées par  $j$ , de l'œil.

Hormis le "Robust scaling" et le "Quantile transform", les techniques présentées dans le tableau 3.2 ont l'inconvénient de ne pas réduire l'impact des données aberrantes. Pour diminuer cet impact, le "Robust scaling" propose une mise à l'échelle ne tenant pas compte des valeurs minimales et maximales, mais du premier et troisième quartile<sup>5</sup>, et le "Quantile transform" transforme chaque donnée de  $\mathbf{x}_i$  en vue de conférer une distribution  $d$  uniforme ou gaussienne à la variable  $x$  prenant les valeurs dans  $\mathbf{x}_i$ . "L'histogramme" de cette variable correspond alors à la distribution  $d$ . Cette méthode calcule dans un premier temps une fonction de répartition<sup>6</sup> ("cumulative distribution function" – CDF) empirique, qui associe à chaque  $x_{ij}$  la probabilité  $p$  qu'une variable  $x$  (prenant valeur dans  $\mathbf{x}_i$ ) soit inférieure ou égale à  $x_{ij}$ . Ensuite, la fonction quantile<sup>7</sup> relative à la distribution  $d$  est appliquée à chaque probabilité  $p$  obtenue. Malgré sa robustesse aux données aberrantes, cette

5. Les données sont triées par ordre croissant et séparées en quatre parts égales. Les trois valeurs séparant chaque portion sont des quartiles : le premier quartile se situe entre la première et deuxième part, tandis que le troisième quartile est à la frontière entre la troisième et quatrième part.

6. La fonction de répartition de la variable aléatoire  $X$  donne pour tout  $x$  la probabilité que  $X$  soit inférieure ou égale à  $x$ .

7. La fonction quantile associée à la distribution d'une variable aléatoire  $X$  fournit pour une

| Prétraitement      | équation                                                                                                  | Conséquences                                       |
|--------------------|-----------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| Simple scale       | $x'_{ij} = x_{ij}/\max(\mathbf{x}_i), \forall j$                                                          | $ x'_{ij}  < 1$                                    |
| Standard scaling   | $x'_{ij} = (x_{ij} - \mu)/\sigma^2, \forall j$                                                            | $\mu' = 0$ et $\sigma'^2 = 1$                      |
| Min-max scaling    | $x'_{ij} = \frac{x_{ij} - \min(\mathbf{x}_i)}{\max(\mathbf{x}_i) - \min(\mathbf{x}_i)}, \forall j$        | $x'_{ij} \in [0, 1]$                               |
| Max-abs scaling    | $x'_{ij} = \frac{2(x_{ij} - \min(\mathbf{x}_i))}{\max(\mathbf{x}_i) - \min(\mathbf{x}_i)} - 1, \forall j$ | $x'_{ij} \in [-1, 1]$                              |
| Robust scaling     | $x'_{ij} = \frac{x_{ij} - \mu}{q_3 - q_1}, \forall j$                                                     | $\mu' = 0$ et la plupart des $x'_{ij} \in [-1, 1]$ |
| Quantile transform | $x'_{ij} = \text{Quantile}(\text{CDF}(x_{ij})), \forall j$                                                | $\mathcal{N}(0, 1)$<br>ou $\mathcal{U}(0, 1)$      |

TABLE 3.2 – Méthodes de prétraitement.  $\mu$ ,  $\sigma^2$  et  $\sigma$  correspondent respectivement à la moyenne, la variance et l'écart-type des données stockées dans  $\mathbf{x}_i$  (de même pour  $\mu'$ ,  $\sigma'^2$  et  $\sigma'$  avec l'ensemble des  $x'_{ij}$  pour tout  $j$ ).  $q_3$  et  $q_1$  désignent respectivement les troisième et premier quartiles des données dans  $\mathbf{x}_i$ .  $\mathcal{N}(\mu, \sigma^2)$  et  $\mathcal{U}(a, b)$  désignent respectivement la distribution normale de moyenne  $\mu$  et de variance  $\sigma^2$ , et la distribution uniforme sur  $[a, b]$ . Après pré-traitement par “quantile transform”, les données de  $\mathbf{x}_i$  suivent une distribution gaussienne ou uniforme.

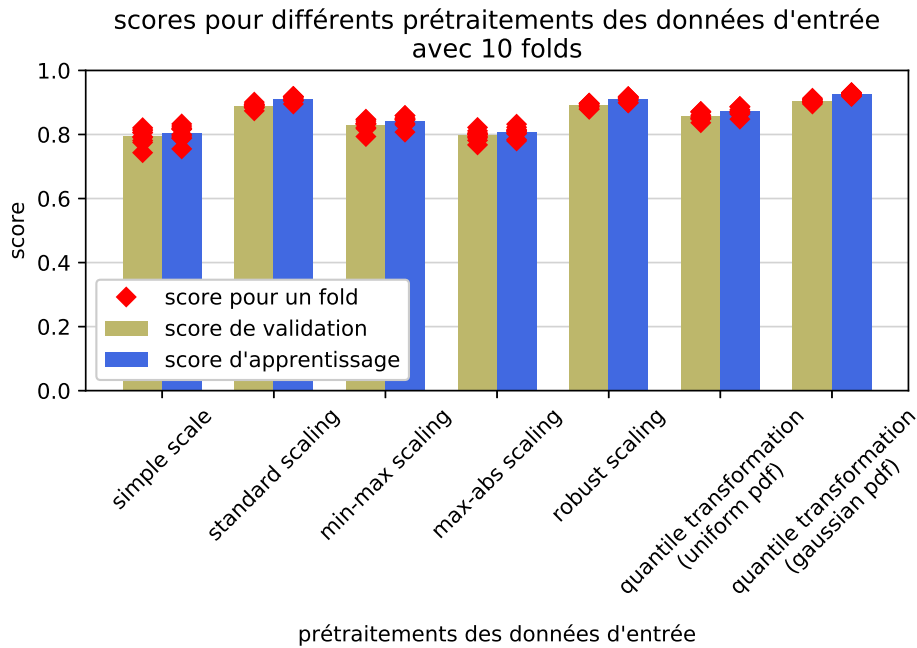


FIGURE 3.12 – Score des réseaux de neurones pour différents prétraitements. Le score moyen sur les données d'apprentissage (resp. de validation) est affiché en bleu (resp. en vert). Les scores de chaque fold sont indiqués par un losange rouge.

méthode de normalisation a tendance à altérer les corrélations entre variables.

Le graphique 3.12 montre le score d'apprentissage (défini en 3.2.1.1) pour les différentes méthodes de prétraitement. Sur chacun de ces tests, nous avons fixé les hyperparamètres suivants :

- le nombre d'époques est 50 ;
- la fonction d'activation est la fonction "relu" (définie section 3.2.1.3) ; cette fonction est souvent utilisée pour les réseaux de neurones profonds [Glorot 2011] ;
- la méthode d'initialisation des poids est la méthode "Glorot uniform" [Glorot 2010] ;
- la fonction de perte est "l'erreur quadratique moyenne".

L'ensemble des méthodes citées dans le tableau 3.2 a été testé. Celle qui fournit le meilleur résultat pour notre application est la "quantile transform" associée à une distribution gaussienne de moyenne nulle et de variance unitaire.

---

probabilité  $p$  une valeur  $x$ , telle que la probabilité pour que  $X$  soit inférieur ou égale à  $x$  soit égale à  $p$ .

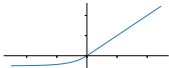
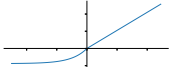
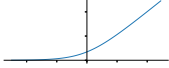
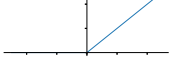
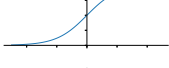
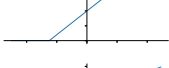
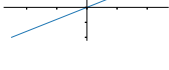
| Activation    | Equation                                                                                                   | Graphique                                                                             |
|---------------|------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| elu           | $f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$         |    |
| selu          | $f(x) = \lambda \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$ |    |
| softplus      | $f(x) = \log(1 + e^x)$                                                                                     |    |
| softsign      | $f(x) = x / (1 +  x )$                                                                                     |    |
| relu          | $f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$                       |    |
| tanh          | $f(x) = \frac{2}{1 + e^{-2x}} - 1$                                                                         |    |
| sigmoïde      | $f(x) = \frac{1}{1 + e^{-x}}$                                                                              |    |
| hard sigmoïde | $f(x) = \max(0, \min(1, x * 0.2 + 0.5))$                                                                   |  |
| linéaire      | $f(x) = x$                                                                                                 |  |

TABLE 3.3 – Tableau représentant les différentes fonctions d'activation. Sur les graphiques,  $\alpha = 1$  pour elu et  $(\alpha, \lambda) = (1.6732, 1.0507)$  pour selu.

### 3.2.1.3 Initialisation des poids et fonction d'activation

L'initialisation des poids et la fonction d'activation sont des hyperparamètres dépendants qu'il faut évaluer conjointement.

Les poids du réseau à l'initialisation sont fixés aléatoirement suivant une distribution de probabilité. Plusieurs possibilités sont proposées par l'outil Keras telles que des distributions normales et uniformes centrées autour de zéro. Nous avons testé les méthodes suivantes : normale  $\mathcal{N}(0, 0.05)$ , normale  $\mathcal{N}(0, 0.05)$  tronquée au-delà de deux fois l'écart-type, uniforme  $\mathcal{U}(-0.05, 0.05)$ , Lecun uniforme, Lecun normale [LeCun 1998], Glorot normale, Glorot uniforme [Glorot 2010], He normale, He uniforme [He 2015].

Les différentes fonctions d'activation testées et proposées par Keras sont présentées dans le tableau 3.3. Certaines sont basées sur des fonctions affines (ou affines par morceaux) et d'autres sur les fonctions logarithmiques et exponentielles. Or, un des critères pour le choix de la fonction d'activation est son coût calculatoire, dans le but de limiter la consommation énergétique du dispositif. Les fonctions affines par morceaux sont donc à privilégier.

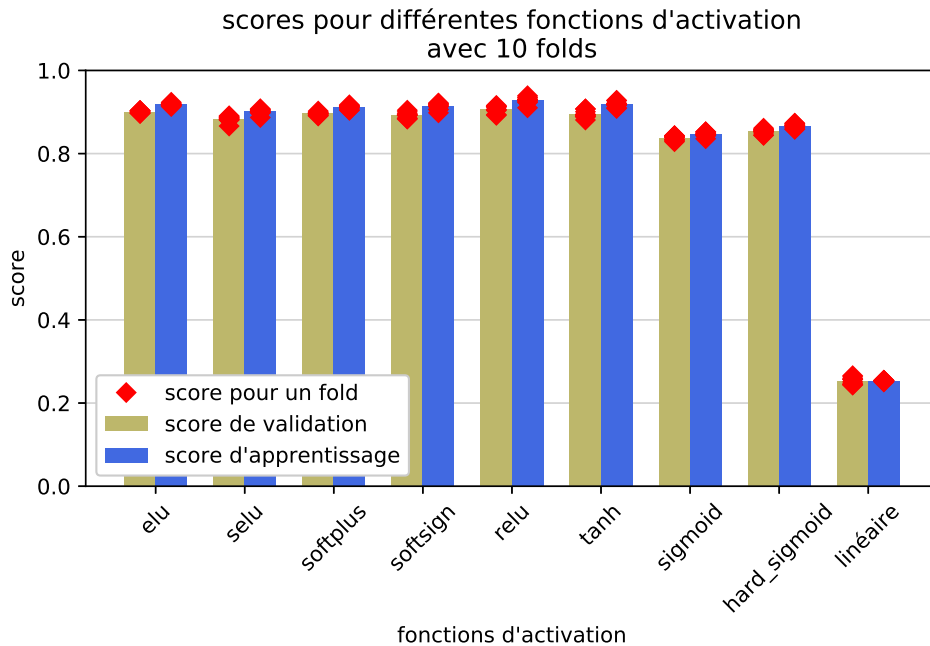


FIGURE 3.13 – Score des réseaux de neurones pour différentes fonctions d'activation. Le score moyen sur les données d'apprentissage (resp. de validation) est affiché en bleu (resp. en vert). Les scores de chaque fold sont indiqués par un losange rouge.

Après avoir testé l'ensemble des couples d'initialisation et de fonction d'activation, la meilleure paramétrisation est l'initialisation avec la distribution Glorot uniforme et la fonction d'activation relu, qui fournit de bons résultats avec un score moyen de 0.909 sur les données de validation. Par ailleurs, la fonction d'activation relu a l'avantage de nécessiter peu de calculs. On peut observer l'impact du choix de la fonction d'activation sur le graphique 3.13, où l'initialisation des poids est Glorot uniforme, excepté selu qui est conseillée avec l'initialisation Lecun normale. Pour ces tests, le nombre d'époques est 50 et la fonction de perte, qui guide la mise à jour des poids lors de l'apprentissage, est l'erreur quadratique moyenne.

D'autres fonctions d'activation plus sophistiquées existent avec notamment le PRelu [He 2015], correspondant à un relu paramétrique dont la valeur du paramètre est lui aussi estimé durant l'apprentissage. Elles pourront être testées par la suite pour aller plus loin dans la paramétrisation du réseau.

#### 3.2.1.4 Fonction de perte

En régression (contrairement aux problèmes de classification), deux fonctions de perte sont généralement utilisées : l'erreur quadratique moyenne et l'erreur absolue moyenne. Dans notre cas, ces fonctions ont comme objectif de réduire la différence entre les projections en millimètre sur les axes  $X_{oculomètre}$  et  $Y_{oculomètre}$  des vecteurs directions estimés et ciblés. Cependant, le choix de la projection orthonormée

| Mesure de l'erreur         | équation                                                  |
|----------------------------|-----------------------------------------------------------|
| Erreur quadratique moyenne | $\frac{1}{n} \sum_{k=1}^n (y_{cible_k} - y_{estime_k})^2$ |
| Erreur absolue moyenne     | $\frac{1}{n} \sum_{k=1}^n  y_{cible_k} - y_{estime_k} $   |
| Erreur personnalisée       | $\frac{1}{n} \sum_{k=1}^n \Delta angle_k$                 |

TABLE 3.4 – Tableau représentant les différentes fonctions de perte appliquées sur  $n$  séries de données estimées  $y_{estime_k}$  et réelles  $y_{cible_k}$ .

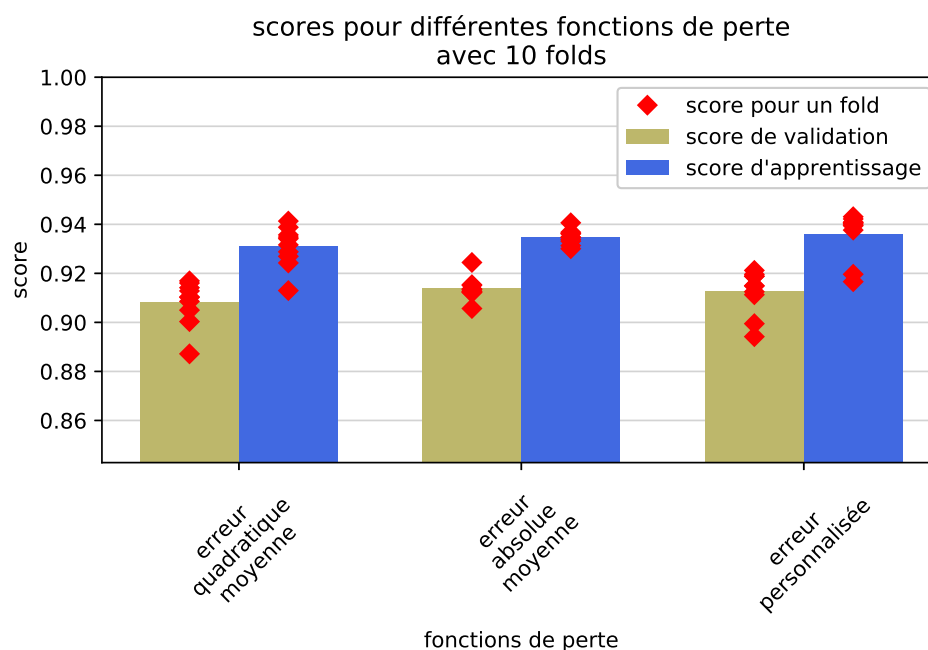


FIGURE 3.14 – Score des réseaux de neurones pour différentes fonctions de perte. Le score moyen sur les données d'apprentissage (resp. de validation) est affiché en bleu (resp. en vert). Les scores de chaque fold sont indiqués par un losange rouge.

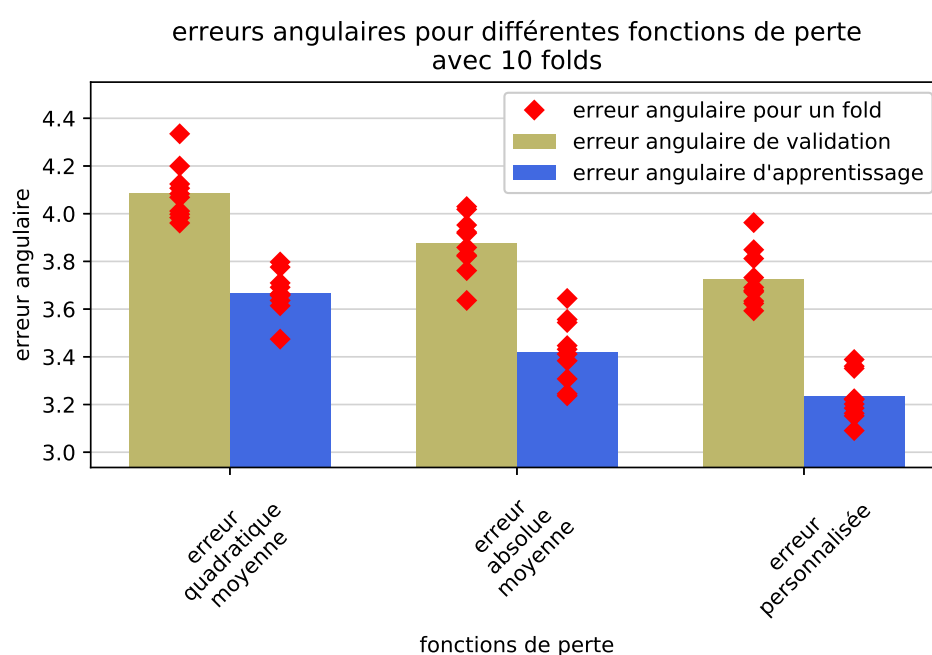


FIGURE 3.15 – Erreur angulaire absolue moyenne en degrés des réseaux de neurones pour différentes fonctions de perte. L'erreur moyenne (sur tous les folds) sur les données d'apprentissage (resp. de validation) est affichée en bleu (resp. en vert). Les résultats de chaque fold sont indiqués par un losange rouge.

sur  $(X_{oculomètre}, Y_{oculomètre}, Z_{oculomètre})$  pour l'expression des données de sorties génère une erreur angulaire de la direction de regard non uniforme sur le champ de vision. En effet pour une même erreur angulaire, une direction de regard proche de l'axe  $Z_{oculomètre}$  aura une erreur sur  $X_{oculomètre}$  et  $Y_{oculomètre}$  plus importante que pour une direction de regard presque perpendiculaire à  $Z_{oculomètre}$ . Ainsi, cette non-uniformité est susceptible de contraindre davantage les directions proches de l'axe  $Z_{oculomètre}$  lors du processus de minimisation de l'erreur d'estimation. Pour éviter ce problème, une solution est de définir une nouvelle fonction de perte calculant directement l'erreur angulaire absolue moyenne en degré, que nous nommerons "erreur personnalisée". Les trois fonctions décrites dans le tableau 3.4 ont été testées sur notre réseau de neurones de référence avec les hyperparamètres suivants : 50 époques, Glorot uniforme et relu. La figure 3.14 montre que la fonction d'erreur a peu d'influence sur le score. En revanche, son impact est visible sur l'erreur angulaire absolue moyenne (Figure 3.15). Ce résultat de l'erreur personnalisée s'explique car le réseau est alors optimisé pour réduire l'erreur angulaire.

### 3.2.1.5 Définition d'une architecture de réseau de neurones

La question abordée dans cette section est de savoir quelle architecture de réseau de neurones convient à nos besoins. Nous souhaitons donc définir le nombre de couches cachées intégrées dans le réseau et le nombre de nœuds sur chaque couche. Pour notre étude, les couches du réseau sont densément connectées, c'est-à-dire que les nœuds sont connectés à chaque nœud de la couche précédente et à chaque nœud de la couche suivante.

Plusieurs architectures ont été testées et évaluées. Elles sont mentionnées sur le graphique de la figure 3.16. La notation (a,b,c, ...) désigne le nombre de nœuds par couches cachées. Ainsi, (80,50) est un réseau de deux couches cachées, où la première contient 80 nœuds et la secondes 50.

Lors de nos tests, nous avons pu confirmer l'idée selon laquelle plus nous ajoutons de degrés de liberté au réseau (que ce soit en ajoutant des couches et des nœuds par couche), plus le système est capable d'estimer correctement la direction du regard. Néanmoins, les coûts calculatoire et énergétique augmentent eux aussi.

D'autre part, nous observons un léger phénomène de surapprentissage ("overfitting"). Ce phénomène se produit lorsque le réseau a appris trop spécifiquement sur la base de ce jeu de données d'apprentissage. Dans ce cas de figure, le modèle n'est plus capable de généraliser sur de nouvelles données. Le problème de surapprentissage est récurrent en apprentissage automatique lorsque le nombre de degrés de liberté d'un modèle est trop important. En général, ce phénomène s'observe quand les performances sur données d'apprentissage sont bien meilleures que sur les données de validation. Pour le contrer, une des solutions est d'augmenter le nombre de données d'apprentissage, permettant ainsi d'améliorer la capacité du réseau à généraliser.

Parmi les architectures testées, aucune ne se démarque particulièrement des autres. Le choix de l'architecture se base donc sur un compromis entre performance

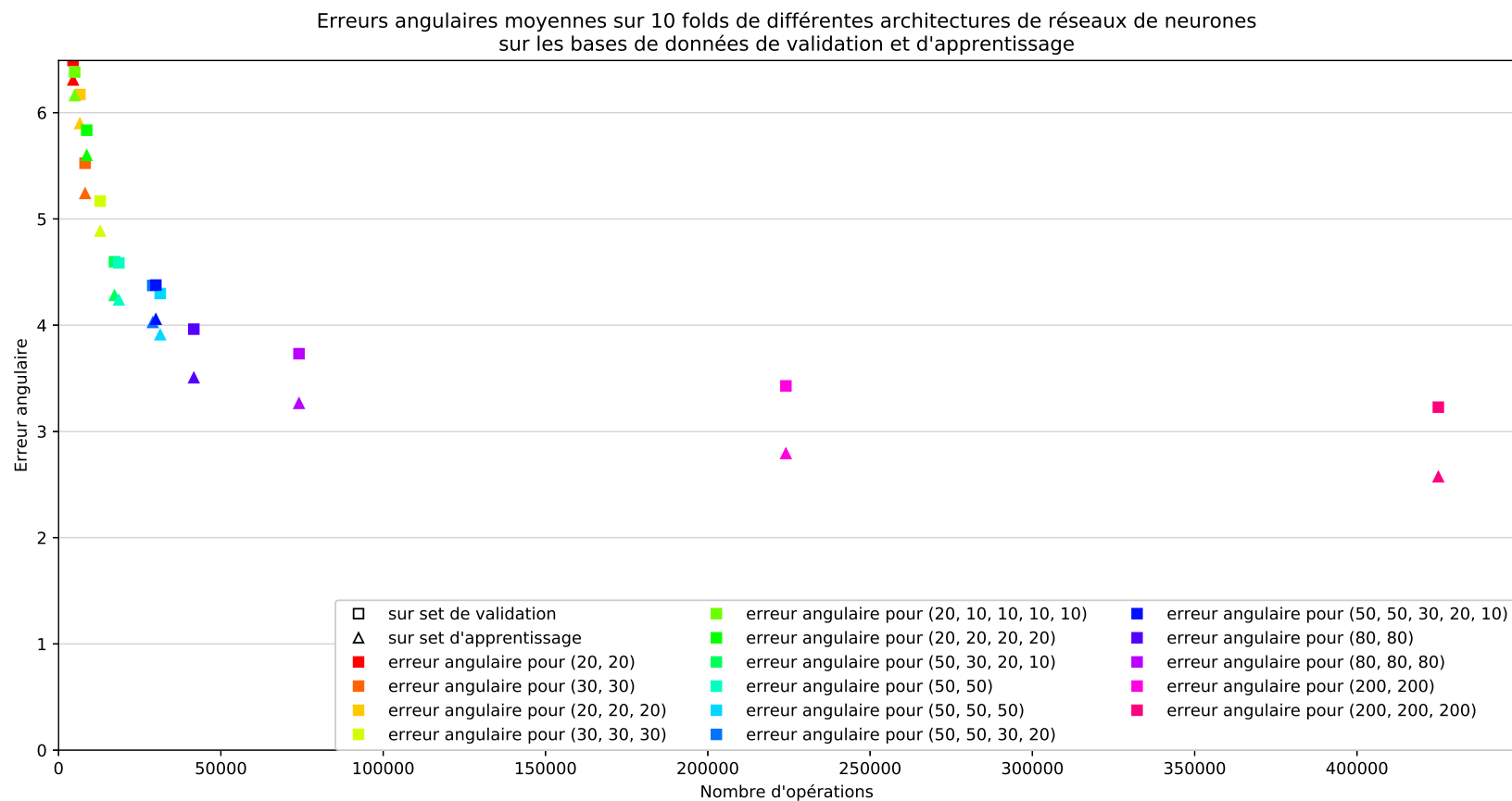


FIGURE 3.16 – Performance en termes d'erreur angulaire absolue moyenne (en degré) pour différentes architectures neuronales classées selon le nombre d'opérations de multiplication incluses dans le calcul d'une direction du regard. Les résultats sur les bases de données de validation (resp. apprentissage) sont représentés par un carré (resp. triangle).

| Opération      | Proportion moyenne | Ecart-type des proportions |
|----------------|--------------------|----------------------------|
| Multiplication | 0.1945             | 0.002425                   |
| Addition       | 0.1992             | 0.0007411                  |
| Lecture        | 0.3996             | 0.0003843                  |
| Affectation    | 0.2068             | 0.003449                   |

TABLE 3.5 – Tableau listant la moyenne et l'écart-type des proportions de chaque opération impliquée dans le processus d'estimation d'une direction du regard pour toutes les architectures testées dans le graphique 3.16. L'écart-type étant faible, on peut constater que la proportion de chaque type d'opération ne change pas suivant l'architecture du réseau.

et coût calculatoire et énergétique.

Toutefois, la consommation du système ne se prédit pas facilement. Elle dépend du matériel choisi (processeur généraliste, plate-forme dédiée) pour effectuer cette tâche : certaines opérations seront plus coûteuses que d'autres et certains calculs pourront être parallélisés. Nous nous intéressons donc au coût calculatoire plutôt qu'au coût énergétique en distinguant les opérations suivantes : multiplication, addition, lecture de variable et affectation.

Pour illustrer l'impact du nombre de multiplications sur les performances du système, le graphique 3.16 classe différentes architectures de réseaux de neurones en fonction de l'erreur angulaire moyenne et du nombre de multiplications impliquées dans le processus d'estimation d'une direction du regard. Il est alors intéressant de constater que la relation entre le nombre d'opérations et l'erreur angulaire absolue moyenne du réseau semble être exponentiellement décroissant. Ainsi, au delà d'un certain seuil, pour améliorer les performances du système, le coût calculatoire à engager devient de plus en plus conséquent.

Concernant les autres opérations (addition, lecture de variable et affectation), l'allure du graphique est la même. En effet, la proportion de chacune des opérations est similaire pour toutes les architectures testées dans le graphique 3.16. Le tableau 3.5 montre la moyenne et l'écart-type des proportions de chaque opération sur l'ensemble des architectures testées.

### 3.2.1.6 Dimensionnement du nombre d'époques et de la taille de la base de données

Sur le graphique 3.17, on observe qu'un nombre élevé d'époques améliore l'estimation de la direction du regard. Néanmoins, pour un très grand nombre d'époques, il se produit une saturation du score de validation, voire un phénomène de surap-

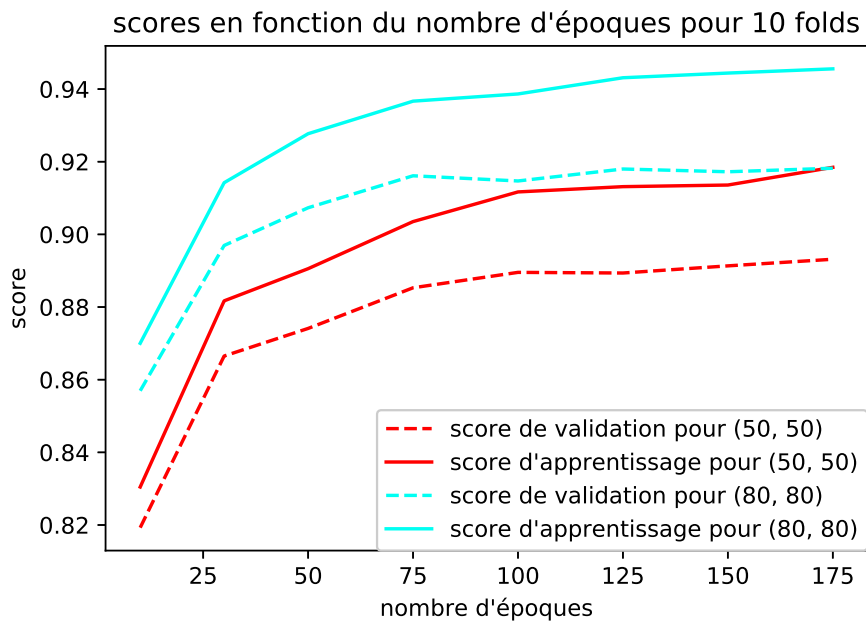


FIGURE 3.17 – Évolution du score en fonction du nombre d'époques pour deux architectures de réseaux différents, chacun de deux couches de 50 ou 80 nœuds par couches.

apprentissage où le score d'apprentissage croît davantage que celui de validation.

Jusqu'ici, pour nos tests nous avons fixé le nombre d'époques à 50 pour des raisons de temps de traitement. Dans la pratique, il serait plus intéressant de considérer 100 époques. Au-delà, l'amélioration apportée n'est pas significative voire inexistante sur les données de validation.

En comparaison, l'augmentation de la base de données d'apprentissage permet de réduire l'effet de surapprentissage, ce qu'on observe avec le rapprochement des scores de validation et d'apprentissage lorsque le nombre de données d'apprentissage augmente, comme l'indiquent les courbes pleines et en pointillé du graphique 3.18. D'autre part, les performances du système définies par les courbes en pointillé augmentent avec la taille de la base d'apprentissage. Ainsi plus la base d'apprentissage est importante, meilleur sera le résultat. Cependant, cette croissance tend à devenir de moins en moins marquée à mesure que l'on ajoute des données.

En conséquence, lors de nos simulations, nous avons considéré près de 45000 jeux de données (un jeu de données contient les  $n$  mesures fournies par les  $n$  photodiodes de l'assemblage et la direction du regard (ou la pose de l'œil)) que l'on répartit en données d'apprentissage (90%) et de validation (10%).

### 3.2.2 Premiers résultats et analyse

Les tests présentés dans la section précédente nous ont permis de choisir une paramétrisation intéressante pour le réseau de neurones : normalisation des données

scores en fonction de la taille de la base d'apprentissage pour 10 folds

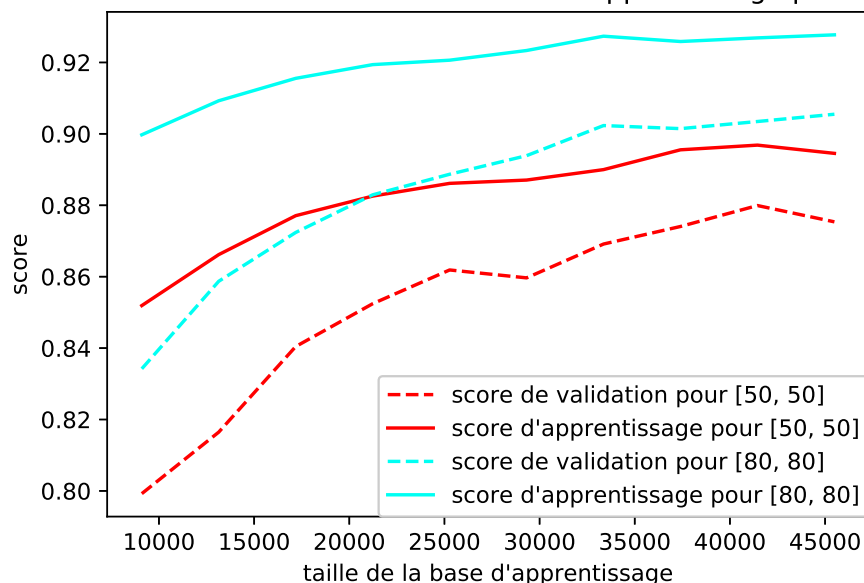


FIGURE 3.18 – Évolution du score en fonction de la taille de la base d'apprentissage pour deux architectures de réseaux différents chacun de deux couches contenant 50 ou 80 nœuds.

d'entrée par “quantile transform”, initialisations des poids du réseau par “Glorot uniforme”, fonction d'activation “relu” et fonction de perte erreur angulaire absolue moyenne. Avec cette paramétrisation, de nouveaux apprentissages ont été effectués pour mener une analyse plus approfondie sur les performances du réseau et visualiser l'erreur d'estimation. La première partie de cette section est consacrée à l'étude de l'erreur d'estimation de la direction du regard. La deuxième partie porte sur les performances du réseau pour l'estimation de la pose de l'œil.

### 3.2.2.1 Estimation de la direction du regard

Tout en considérant la même base de données synthétiques issue du même assemblage de photodiodes, on s'intéresse maintenant à la visualisation et à l'analyse de l'erreur.

Le graphique 3.19(a) permet de visualiser la répartition de l'erreur angulaire absolue en sortie du réseau sur l'ensemble du jeu de validation en utilisant des boîtes à moustache pour chaque fold. On observe que la majorité de l'erreur se situe entre 1 et 5 degrés, visible avec le premier et troisième quartiles sur le graphique. Néanmoins, le réseau ne parvient pas à estimer la direction du regard pour certaines données d'entrée, pour lesquelles il fournit des résultats erronés entre 8 et 70 degrés.

Ces résultats erronés ne semblent pas être dues à un surapprentissage, car l'allure des boîtes à moustache est la même sur les données d'apprentissage (Figure 3.19(b)).

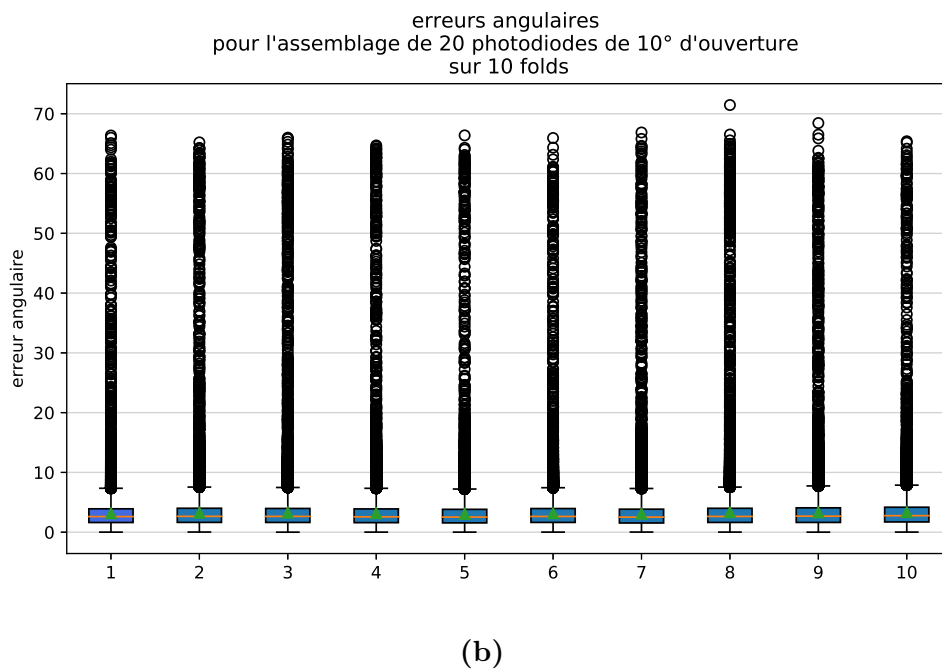
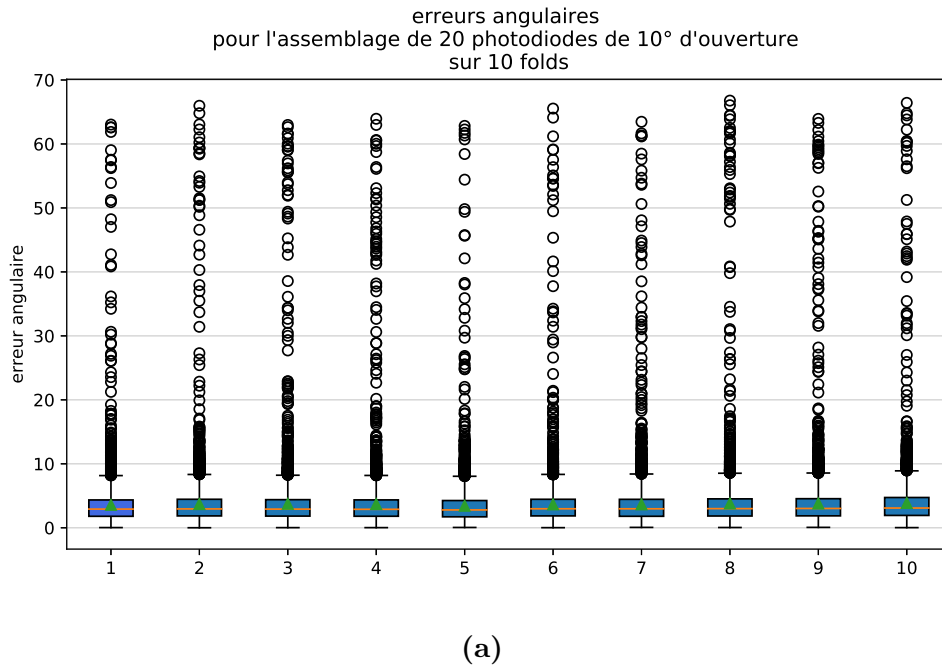


FIGURE 3.19 – Répartition de l'erreur angulaire absolue en degrés sur des données de validation (diagramme - (a)) ou d'apprentissage (diagramme - (b)), représentée par des boîtes à moustaches pour chacun des 10 folds. La moyenne est symbolisée par un triangle vert et la médiane par un trait rouge. La boîte bleue regroupe l'ensemble des données entre le premier ( $Q_1$ ) et troisième ( $Q_3$ ) quartiles. Les moustaches sont définies par rapport au premier et troisième quartiles, avec  $Q_3 + 1.5 * (Q_3 - Q_1)$  pour la moustache supérieure et  $Q_1 - 1.5 * (Q_3 - Q_1)$  la moustache inférieure. Les résultats situés en dehors des moustaches sont assimilés à des résultats erronés.

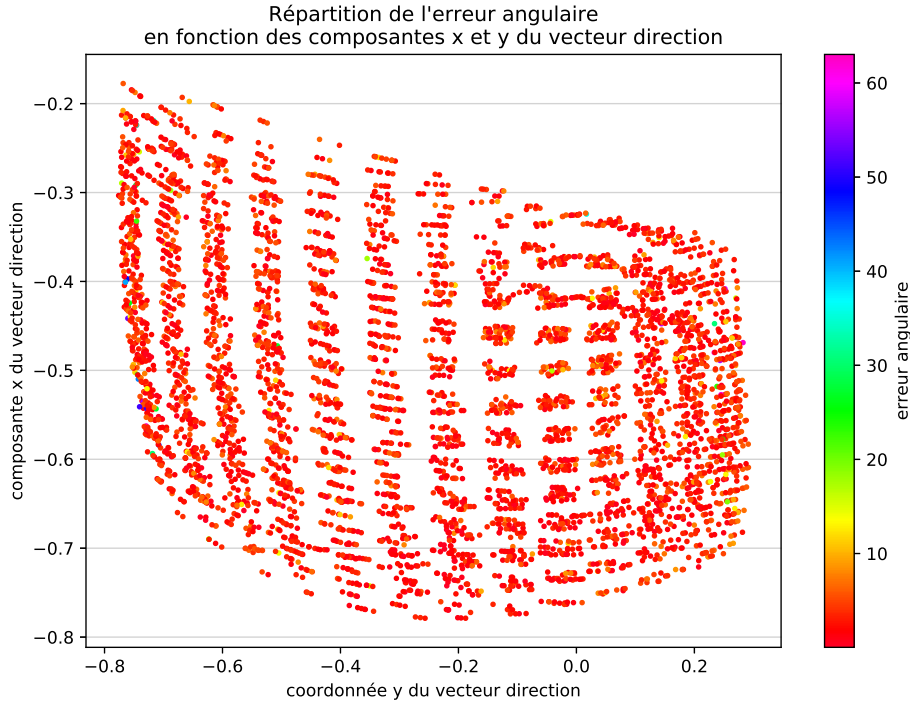


FIGURE 3.20 – Répartition spatiale de l'erreur angulaire absolue en degré (issue du fold 1 du graphique 3.19) suivant les deux projections sur  $X_{oculomètre}$  et  $Y_{oculomètre}$  du vecteur direction. La valeur de l'erreur est codée en couleur. On observe que la majorité des points sont rouges c'est-à-dire que l'on a moins de 10 degrés d'erreur. Les données aberrantes se manifestent en vert, bleu et rose et sont principalement situées dans la périphérie du champ de vision.

La figure 3.20 présente la répartition spatiale de l'erreur. On remarque alors que la périphérie du champ de vision concentre les résultats les plus erronés. On peut trouver deux explications possibles à ce phénomène : le manque de données d'apprentissage en périphérie permettant de bien estimer un modèle, et l'incapacité de l'assemblage de photodiodes choisi à détecter la pupille en périphérie.

### 3.2.2.2 Estimation de la pose de l'œil

Dans cette section, le réseau estime la position du centre de l'œil dans le repère de l'oculomètre en plus de la direction du regard. On ajoute trois composantes à notre problème d'estimation et donc à nos données de sorties : la position selon  $X_{oculomètre}$ ,  $Y_{oculomètre}$  et  $Z_{oculomètre}$  du centre de l'œil, que l'on assimile au centre optique.

La fonction de perte *personnalisée* est définie par l'équation 3.4, qui tient compte de l'erreur angulaire absolue moyenne en degré (Équation 3.2) et de la distance moyenne en millimètre entre la position estimée  $X_{estime_k}$  et la position réelle ciblée  $X_{cible_k}$  (Équation 3.3) sur  $n$  jeux de données avec un facteur de pondération  $\alpha$

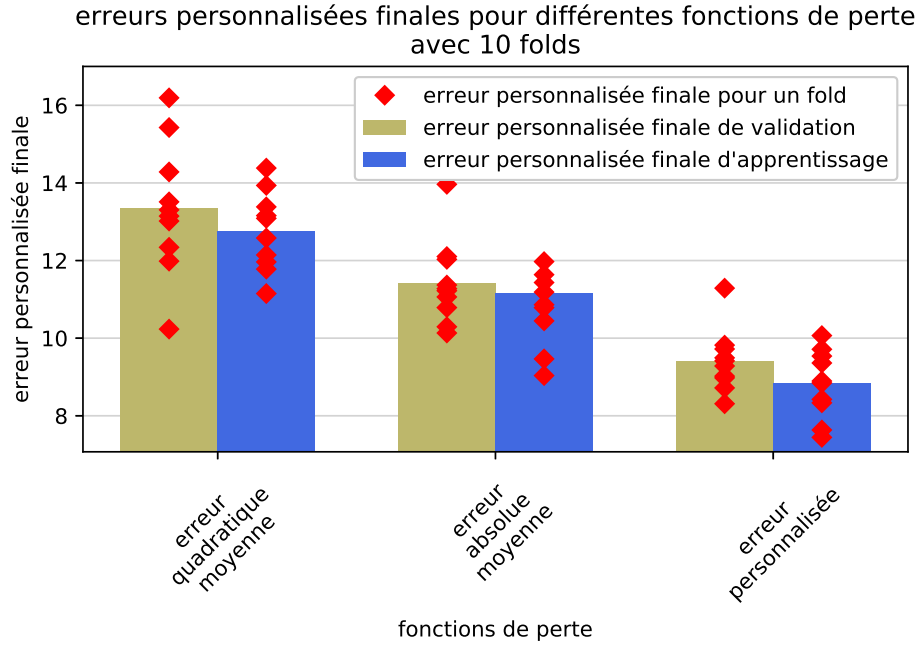


FIGURE 3.21 – Erreur personnalisée du réseau de neurones pour différentes fonctions de perte. L'erreur moyenne sur les données d'apprentissage (resp. validation) est affichée en bleu (resp. vert). Les résultats de chaque fold sont indiqués par un losange rouge.

pour homogénéiser les composantes du vecteur d'erreur. L'erreur angulaire entre la  $k$ -ième direction du regard estimée et la direction réelle ciblée correspondante est notée  $\Delta angle_k$ . Le graphique 3.21 trace le gain obtenu par le biais de la fonction de perte personnalisée, qui se démarque des deux autres fonctions de perte (l'erreur quadratique moyenne et l'erreur absolue moyenne). L'erreur affichée en ordonnée reprend la fonction définie par l'équation 3.4.

$$err_{angle} = \frac{1}{n} \sum_{k=1}^n | \Delta angle_k | \quad (3.2)$$

$$err_{position} = \frac{1}{n} \sum_{k=1}^n \| X_{cible_k} - X_{estime_k} \| \quad (3.3)$$

$$err_{perso} = err_{angle} + \alpha * err_{position} \text{ avec } \alpha = 7 \quad (3.4)$$

Les graphiques de la figure 3.22 montrent la répartition de l'erreur angulaire (diagramme - (a)) et de l'erreur de position (diagramme - (b)). On observe que l'erreur angulaire augmente légèrement par rapport au réseau précédent qui n'estimait que la direction du regard. L'erreur de position, quant à elle, est généralement inférieure à 2 mm et admet une moyenne de 0.73 mm sur l'ensemble des 10 folds,

sachant que le déplacement simulé de la lunette est de  $\pm 4$  mm sur deux axes.

Dans le cadre d'un oculomètre stéréo (estimant la pose des deux yeux), une erreur d'estimation de 2 mm sur la position d'un œil peut être acceptable pour estimer par triangulation la position d'un point 3D observé par l'utilisateur face à lui à moins de quelques mètres de distance. En revanche, cette erreur reste trop importante pour la mise à jour d'un modèle de projection d'un OST-HMD (voir section 2.6).

### 3.2.3 Évaluation de différents assemblages de photodiodes

Pour évaluer différents assemblages de photodiodes, on doit dans un premier temps simuler les valeurs retournées par chacune des photodiodes en variant l'orientation de l'œil et la pose des lunettes par rapport au visage. Cependant, la génération des données synthétiques mise en place sous Blender n'est pas optimisée en terme de temps de calcul. Afin de pouvoir simuler rapidement de nombreux assemblages, nous avons seulement généré 5000 données pour chacun d'eux. De plus, afin d'obtenir des résultats fiables malgré ce faible nombre, nous n'avons pas considéré de mouvement des lunettes, qui induirait un changement de position de l'œil. En conséquence, les scores obtenus sont très élevés et ne sont pas représentatifs des performances du réseau dans des conditions réelles incluant un mouvement des lunettes. C'est pourquoi, nous utilisons ce score uniquement pour comparer les réseaux entre eux et non pour évaluer la performance d'un réseau seul.

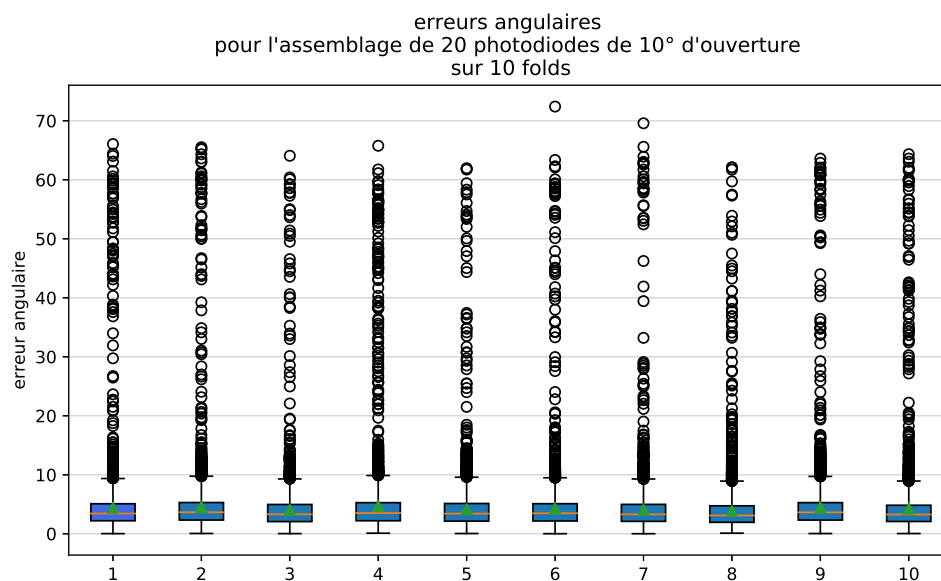
Le paramétrage du réseau de neurones est celui défini à la section 3.2.2.1. Pour chaque assemblage, deux architectures de réseau sont considérées : trois couches à 30 neurones par couche (A1) et deux couches à 80 neurones par couche (A2). D'après le graphique 3.16, ces deux architectures semblent être des compromis entre performance et coût calculatoire. Nous prenons ces deux exemples pour observer l'impact du choix de l'architecture sur les assemblages de photodiodes possibles.

#### 3.2.3.1 Variation de l'angle d'ouverture des photodiodes

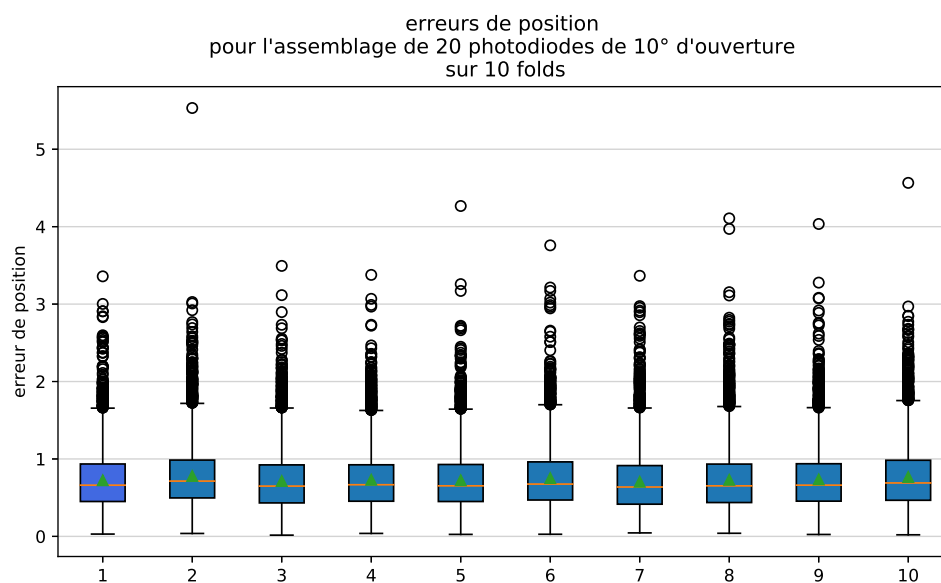
Pour observer l'impact de l'angle d'ouverture sur la capacité du réseau, cinq assemblages différents ont été simulés avec des angles d'ouverture allant de 40 à 4 degrés :

- sur l'assemblage 1, toutes les photodiodes ont 4° de champ de vue ;
- sur l'assemblage 2, toutes les photodiodes ont 7° de champ de vue ;
- sur l'assemblage 3, toutes les photodiodes ont 10° de champ de vue ;
- sur l'assemblage 4, toutes les photodiodes ont 20° de champ de vue ;
- sur l'assemblage 5, l'angle d'ouverture des photodiodes positionnées depuis le nez vers la tempe varie linéairement de 30 à 40 degrés, afin d'éviter que les photodiodes proches du nez observent hors du visage.

La forme du verre et la disposition des vingt photodiodes (position et orientation) restent les mêmes que précédemment (Figure 3.10).



(a)



(b)

FIGURE 3.22 – Répartition de l'erreur angulaire absolue en degrés (diagramme - (a)) et de l'erreur de position (distance) en millimètre (diagramme - (b)) sur les données de validation. Une représentation par des boîtes à moustaches est utilisée pour chacun des 10 folds. Le réseau de neurones réalise l'estimation conjointe de la direction du regard et de la position du centre de l'œil.

Le graphique 3.23(a) montre le score de chaque assemblage. On observe que les faibles angles d'ouverture (assemblages 3,2 et 1) fournissent de bons résultats. 10 degrés d'ouverture (assemblage 3) semble être un bon compromis pour la performance et la facilité de mise en œuvre. En effet plus l'angle est petit plus il sera difficile de le fabriquer en une version compacte intégrable dans une monture.

### **3.2.3.2 Variation du nombre de photodiodes**

Plusieurs assemblages avec différents nombre de photodiodes ont été simulés et testés. Le reste de la configuration est similaire : positions des photodiodes distribuées uniformément sur le haut et le bas du verre, toutes orientées vers un même point derrière la pupille, et ouverture égale à 10 degrés. La forme du verre est toujours rectangulaire comme précédemment.

Le graphique 3.23(b) présente l'influence du nombre de photodiodes sur le score du réseau. Comme on peut s'y attendre, moins il y a de photodiodes et moins bon est le score. Cependant, cette diminution du score est faible. Considérant ce résultat, on pourra choisir un compromis entre les performances du système et son encombrement conditionné par le nombre de photodiodes.

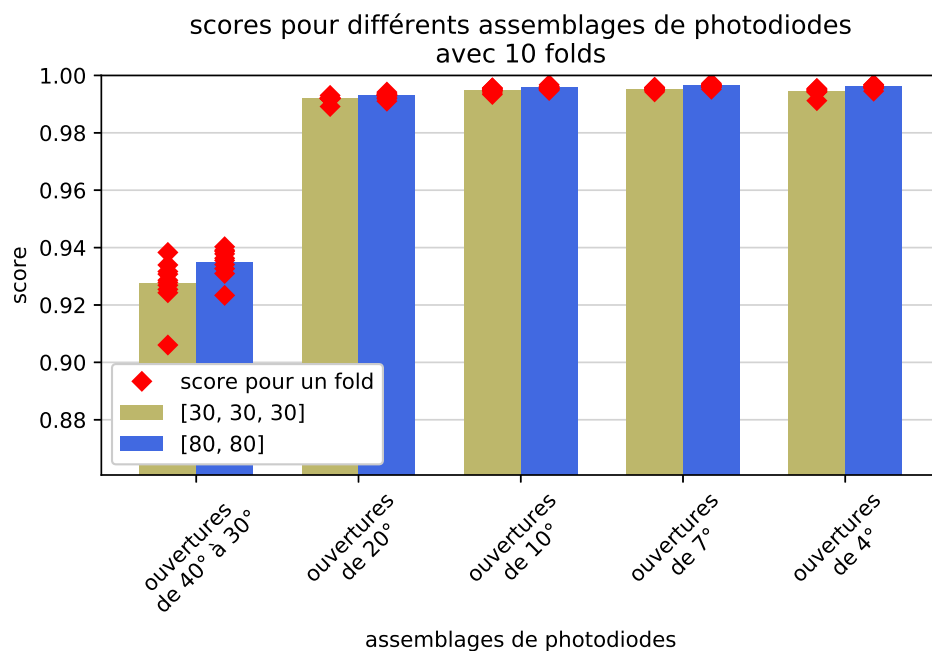
### **3.2.3.3 Variation de l'orientation des photodiodes**

D'autres assemblages ont été évalués pour analyser l'effet de l'orientation des photodiodes sur le système. L'idée est de vérifier si le choix d'un même point ciblé pour toutes les photodiodes n'appauvrit pas les données, du fait qu'elles observent des zones de l'œil très similaires. Pour y répondre, nous avons simulé un assemblage où les photodiodes observent soit un point  $P_1$ , soit un point  $P_2$  à une profondeur différente de  $P_1$  derrière la pupille, pour une pose de l'œil orientée droit devant (direction perpendiculaire au verre). Trois autres assemblages, où l'orientation de chaque capteur est fixée aléatoirement dans une plage de valeurs, ont aussi été testés pour évaluer une configuration sans motifs répétitifs et sans symétrie. Ces cinq assemblages ont tous 20 photodiodes de  $10^\circ$  de champ de vue disposées uniformément sur le haut et le bas du contour rectangulaire du verre.

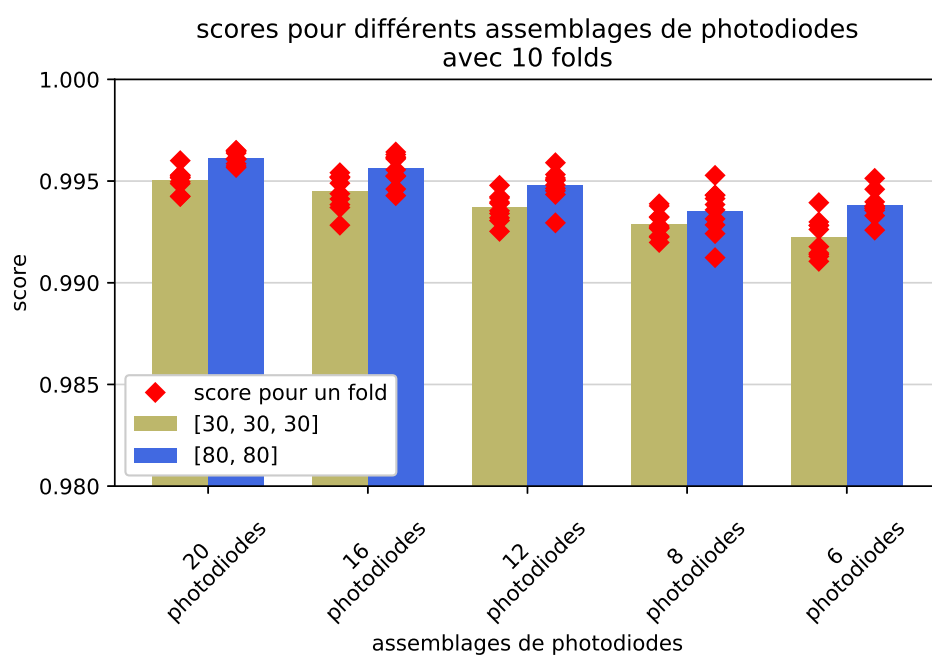
Le graphique 3.24 affiche les résultats des cinq assemblages, on constate que l'hypothèse est vérifiée : il est préférable de définir des orientations aussi variées que possible. Le second assemblage du graphique semble être un bon choix.

### **3.2.3.4 Variation de l'organisation spatiale des photodiodes**

Une autre question s'est posée concernant l'impact de l'organisation spatiale des photodiodes sur le score du réseau. Nous avons donc simulé trois formes différentes : rectangle, cercle et ellipse. Initialement, les capteurs étaient uniquement positionnés sur les bords haut et bas du contour du verre, pensant que les photodiodes sur les bords latéraux auraient un trop grand angle d'incidence pour fournir des données de qualité. Nous avons donc testé un autre assemblage sur verre rectangulaire où les photodiodes sont également réparties sur l'ensemble de la monture. Concernant



(a)



(b)

FIGURE 3.23 – Score des réseaux de neurones pour différents assemblages de photodiodes dont l'angle d'ouverture (diagramme - (a)) ou le nombre de photodiodes (diagramme - (b)) varie. Le score moyen sur données de validation est affiché en vert pour un réseau d'architecture A1 et en bleu pour A2. Les résultats de chaque fold sont indiqués par un losange rouge.

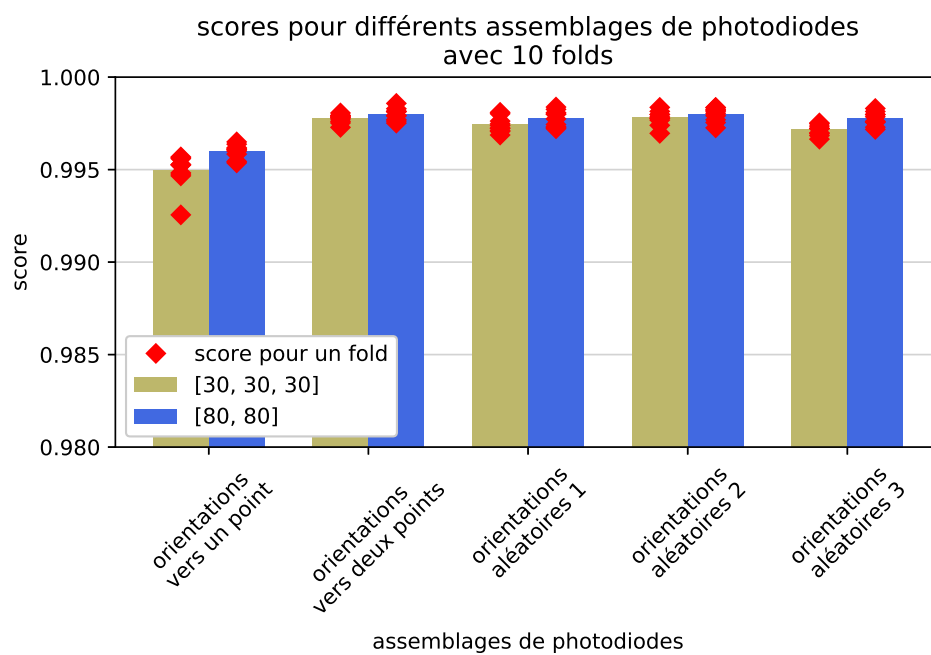


FIGURE 3.24 – Score des réseaux de neurones pour différents assemblages dont l'orientation des photodiodes varie. Le score moyen sur données de validation est affiché en vert pour un réseau d'architecture A1 et en bleu pour A2. Les résultats de chaque fold sont indiqués par un losange rouge.

la forme en cercle, deux assemblages ont été simulés avec deux orientations différentes. Chacun des assemblages comporte 20 photodiodes de  $10^\circ$  d'ouverture, toutes orientées vers un même point.

Ces cinq assemblages sont visibles sur les figures 3.25.

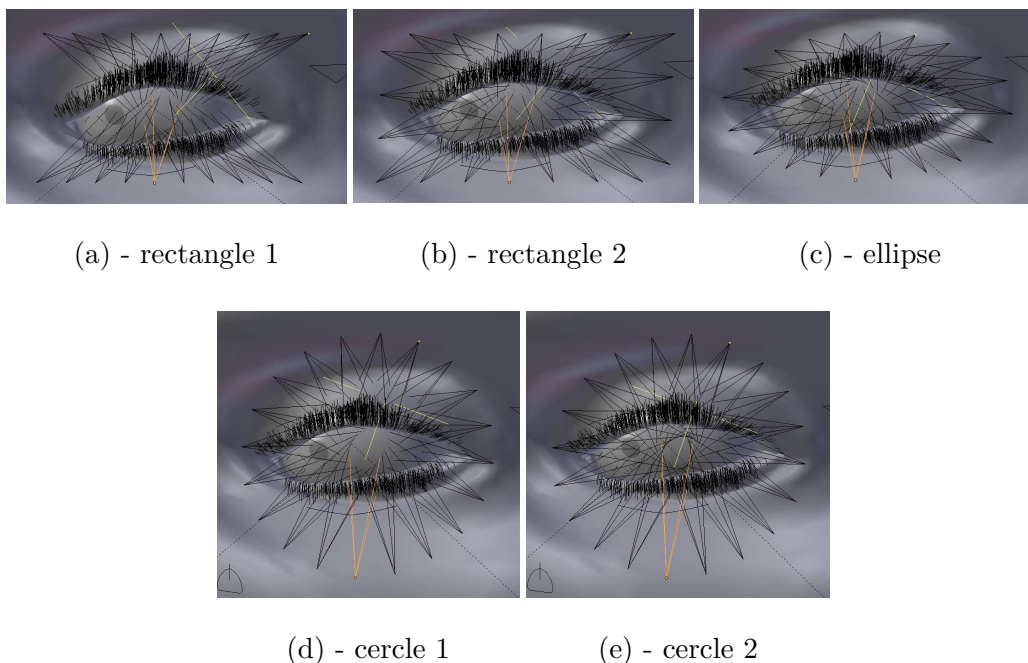


FIGURE 3.25 – Visualisation de différents assemblages de vingt photodiodes devant le modèle de visage. Ces dernières sont représentées par des pyramides à base carrée dont le sommet correspond à la position de la photodiode et la normale de la base à sa direction (les bases des pyramides sont cachées à cause du Z-buffer). (a) est l'assemblage pris comme exemple jusqu'ici, où la forme est rectangulaire et les photodiodes sont réparties en haut et en bas de la monture. (b) est un assemblage avec la même forme de verre, mais avec des photodiodes réparties sur l'ensemble du contour. (c) a une forme elliptique et les photodiodes sont régulièrement espacées. (d) et (e) suivent un contour circulaire, avec des orientations de photodiodes différentes. Les capteurs de (d) visent un point plus éloigné de la pupille contrairement à (e).

Le graphique 3.26 montre le score de chacun des cinq assemblages. On constate tout d'abord qu'une répartition homogène des photodiodes le long du verre de lunette offre de meilleurs résultats par comparaison du score de "rectangle 2" et "rectangle 1". Ensuite entre les différentes formes, l'assemblage sur verre rectangulaire semble être plus performant avec "rectangle 2". D'autre part, pour le contour circulaire, les résultats de "cercle 1" et "cercle 2" montrent qu'il est préférable que les photodiodes soient orientées vers un point éloigné de la pupille.

Pour expliquer ce comportement, on peut analyser les figures 3.25. Ces dernières montrent que la forme circulaire n'offre pas une aussi bonne couverture de la surface de l'œil. Soit les photodiodes n'observent pas le centre de l'œil, soit elles n'observent

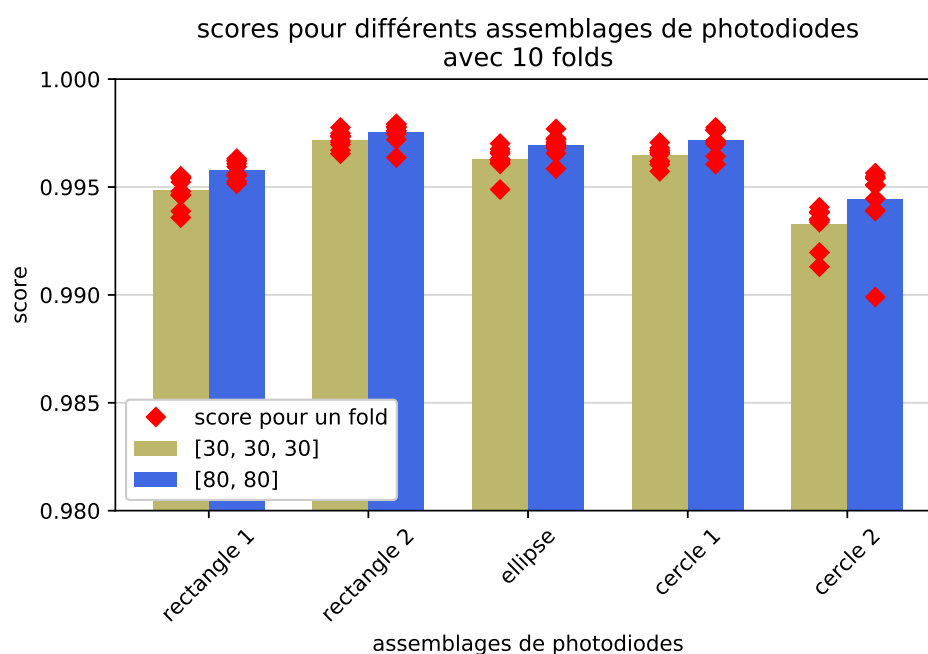


FIGURE 3.26 – Score des réseaux de neurones pour différents assemblages dont l'organisation spatiale des photodiodes varie. Le score moyen sur données de validation est affiché en vert pour un réseau d'architecture A1 et en bleu pour A2. Les résultats de chaque fold sont indiqués par un losange rouge.



FIGURE 3.27 – Filtre appliqué devant les photodiodes.

pas la périphérie et plus particulièrement les coins. La forme rectangulaire, quant à elle, semble proposer une meilleure couverture des coins des yeux.

Ces résultats ainsi que ceux de la section 3.2.3.3 montrent qu’une bonne couverture de l’œil est primordiale dans les performances du système.

### 3.2.3.5 Test avec filtre

De nouveaux assemblages ont été testés afin d’évaluer l’apport de filtres placés devant les photodiodes. Ils sont basés sur un verre de forme rectangulaire. Des photodiodes sont homogènement réparties sur les parties haute et basse du contour, et fixent toutes un même point situé derrière la pupille. Pour la simulation, les filtres (visibles sur la figure 3.27) sont appliqués sur les images calculées par Blender.

Deux assemblages différents ont été évalués avec les filtres : un premier où les photodiodes ont 40 à 30 degrés d’ouverture et un second où elles ont toutes 20 degrés. L’orientation des filtres varie d’une photodiode à l’autre : la ligne définie par le filtre correspond à la projection de la direction de l’œil sur l’image d’une caméra positionnée à la place de la photodiode. Ce filtre a pour conséquence de réduire le champ de vue des photodiodes suivant une direction seulement.

Les résultats des deux assemblages ainsi que de trois autres repris de la section 3.2.3.1 sont présentés sur le graphique 3.29. On observe que le gain sur l’assemblage de 40 à 30 degrés est très important. Cependant, il n’excède pas le score de l’assemblage de 20 degrés d’ouverture sans filtre. D’autre part, l’assemblage avec filtre de 20 degrés atteint les mêmes performances que l’assemblage de 10 degrés sans filtre, ce qui peut s’expliquer par la disposition des photodiodes. Comme les photodiodes voisines ont une orientation similaire et sont placées le long du contour qui est globalement perpendiculaire à la fente des filtres (ligne blanche sur le filtre), la redondance observée sur les données est exprimée suivant la direction définie par le contour du verre (Figure 3.28(a)). Le filtre permet de supprimer cette redondance, tout en conservant un large champ de vision suivant la direction de la fente, couvrant ainsi mieux la surface de l’œil (Figure 3.28(b)).

### 3.2.3.6 Conclusion sur l’assemblage à considérer

En conclusion de ces différents tests, on peut retenir que l’angle d’ouverture des photodiodes doit être faible, sinon leurs champs de vision se recouvrent, ce qui induit une forme de redondance dans leurs mesures. Ce recouvrement peut être vu comme l’application d’un filtre passe bas sur les mesures. Ainsi les mesures sont

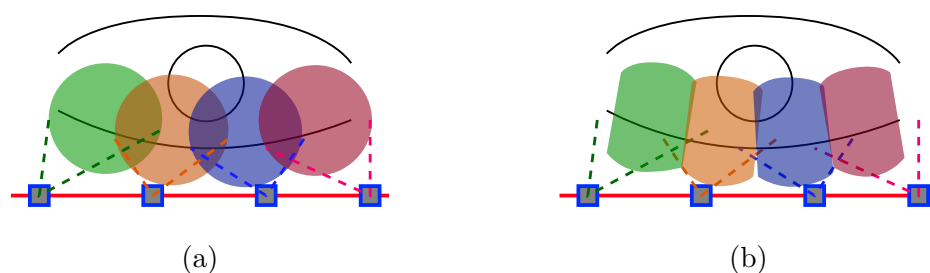


FIGURE 3.28 – Exemple de quatre photodiodes placées le long du contour tracé en rouge. L'illustration (a) montre que les champs de vision des photodiodes (représentés par des disques colorés) se superposent globalement suivant la direction définie par le contour du verre. En ajoutant un filtre sous forme de fente orienté de manière à supprimer ces redondances, le champ de vision des photodiodes est alors tronqué et leurs intersections sont réduites, comme le révèle l'illustration (b). Ainsi on rend les mesures plus discriminantes.

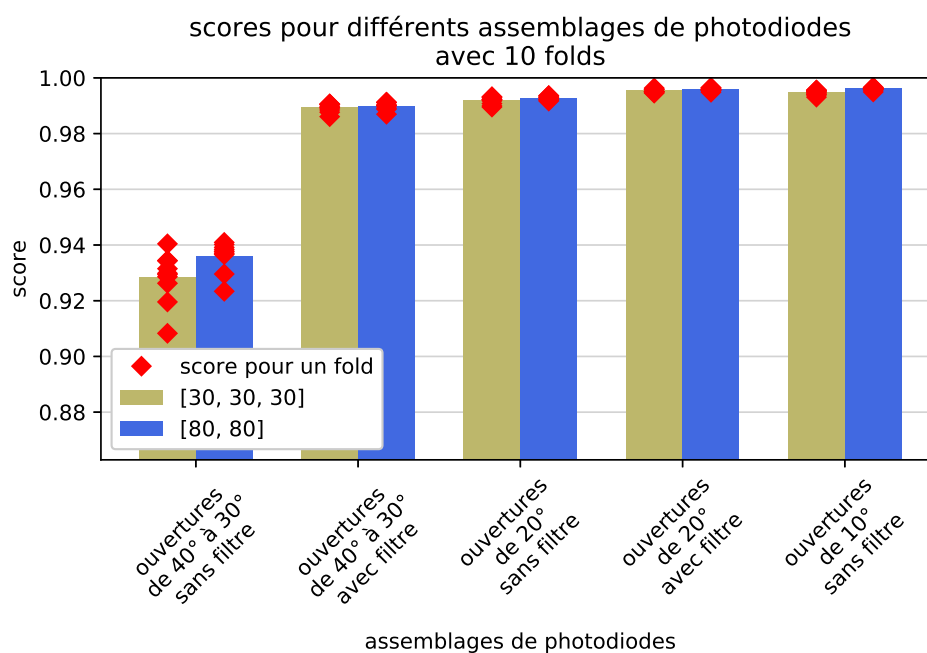


FIGURE 3.29 – Score des réseaux de neurones pour différents assemblages pour lesquels l'angle d'ouverture des photodiodes et la présence de filtre changent. Le score moyen sur données de validation est affiché en vert pour un réseau d'architecture A1 et en bleu pour A2. Les résultats de chaque fold sont indiqués par un losange rouge.

convoluées entre elles, et d'après nos résultats les réseaux de neurones considérés semblent ne pas parvenir à les déconvoluer.

Peut-être qu'un réseau de plus grande taille accompagné d'une grande base de données pourrait remplir cette fonction de déconvolution. Cependant, ce cas sort de notre champ de recherche, étant données les contraintes précitées de faible coût calculatoire et énergétique.

D'autre part comme l'angle d'ouverture doit être faible, la surface de l'œil risque de ne pas être entièrement observée. Il faut donc choisir le positionnement et l'orientation des photodiodes de manière à couvrir toute la surface de l'œil. Pour l'instant, l'assemblage semblant être le plus intéressant suit le contour d'un verre rectangulaire, où les photodiodes sont réparties uniformément mais possèdent des orientations variables comme le propose le second montage de la section du graphique 3.24.

Le nombre de photodiodes devra quant à lui être fixé en fonction des contraintes d'encombrement et des performances attendues.

Ces tests ont été menés en considérant les lunettes immobiles par rapport au visage. Cependant, il serait intéressant de proposer un assemblage qui offre une bonne couverture de la surface de l'œil, quelle que soit la pose des lunettes sur le nez.

### 3.3 Tests sur données réelles

Une fois des premiers résultats obtenus sur simulation, la prochaine étape du projet est d'évaluer le concept de l'oculomètre basse consommation sur données réelles.

Une première approche préalable à la construction d'un assemblage réel est de simuler les données des photodiodes à partir d'images réelles acquises par des caméras haute résolution au lieu d'images synthétiques générées par Blender. Ce principe permet de considérer plus de variabilité de l'environnement, par exemple sur la forme des yeux et du visage. Pour atteindre cet objectif, un banc expérimental capable de générer une base de données de plusieurs dizaines de milliers d'images et de directions du regard a été mis en place. Ce banc peut ensuite être utilisé pour évaluer la précision d'oculomètres lors de tests.

Cette partie du travail, encore en phase de développement, est décrite ci-dessous.

#### 3.3.1 Banc de test et procédure d'acquisition de données

Le banc de test que l'on propose est le suivant (Figure 3.30). Il est constitué :

- d'un oculomètre stéréoscopique de référence noté  $\mathcal{O}_{ref}$  (basé sur une paire de caméras), qui délivre les données alimentant la base de données d'apprentissage de l'oculomètre basse consommation,
- d'un écran affichant une croix en différentes positions contrôlables,
- d'une caméra de scène fixée sur un portique derrière le sujet et filmant l'écran,

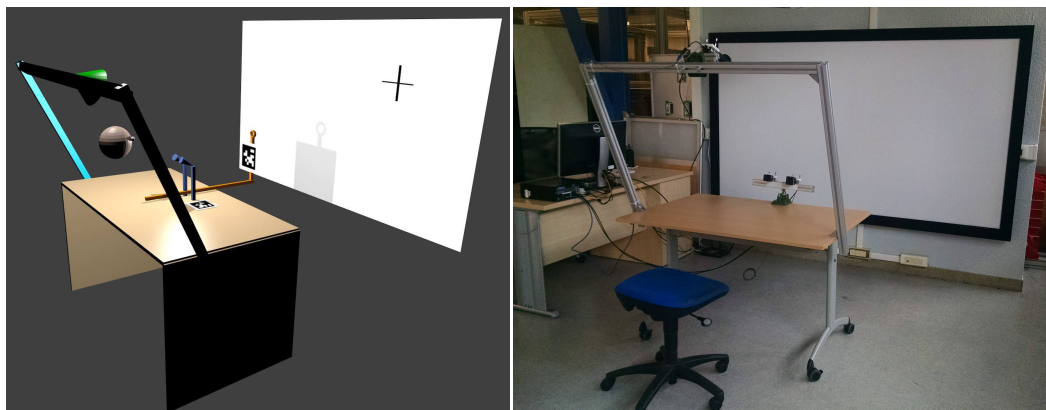


FIGURE 3.30 – Banc de test pour l'acquisition de données et l'évaluation d'oculomètres. (gauche) Simulation du banc de test avec en bleu les deux caméras de l'oculomètre, en vert la caméra de scène fixée sur un portique et en orange la cible tendue à la main par l'utilisateur. (droite) Photo du banc de test réel où l'affichage sur l'écran est généré par un projecteur fixé au plafond.

- d'une cible (en orange sur l'image de gauche de la figure 3.30) pouvant être détectée et localisée grâce à la caméra de scène,
- d'une table sur laquelle est fixée l'oculomètre  $\mathcal{O}_{ref}$  et le portique avec la caméra de scène.

L'utilisateur, dont l'œil est filmé par les deux caméras de l'oculomètre  $\mathcal{O}_{ref}$ , est assis devant le banc de test et regarde en direction de l'écran. Comme la tête de l'utilisateur est susceptible de bouger au cours de l'acquisition, l'oculomètre  $\mathcal{O}_{ref}$  doit tenir compte de ce mouvement et estimer en conséquence la pose de l'œil dans son repère.

Le dimensionnement des éléments du banc de test a été effectué à l'aide d'un modèle synthétique réalisé sous Blender (Figure 3.30). Cette simulation nous a notamment permis de sélectionner les caméras et leurs optiques, de manière à disposer d'un champ de vision et d'une résolution adéquats.

Le banc de test remplit deux objectifs : acquérir une base de données d'apprentissage suffisante et évaluer les performances d'oculomètres existants notés  $\mathcal{O}_{ex}$ . D'une part, pour acquérir un grand nombre de données pour l'apprentissage du réseau de neurones, la procédure d'acquisition doit être simplifiée en nécessitant le minimum de manipulation de la part de l'utilisateur. D'autre part, pour l'évaluation d'oculomètres, le banc de test doit fournir une estimée de la direction du regard précise et fiable. Afin de remplir ces deux objectifs, deux procédures d'acquisition de données sont proposées.

- *Procédure d'acquisition d'une base de données d'apprentissage* : L'utilisateur suit visuellement une croix se déplaçant sur l'ensemble de l'écran, ce qui permet de s'assurer que son regard balaie différentes directions. Pendant ce temps, les caméras de l'oculomètre  $\mathcal{O}_{ref}$  filment ses yeux, et l'algorithme estime la di-

rection de son regard. Puis, les images acquises par les caméras sont reprises pour la simulation des photodiodes. Cette technique permet d'acquérir un grand nombre de données (directions du regard et mesures simulées fournies par les photodiodes) sans nécessiter d'interaction forte de la part de l'utilisateur : seul son regard est contraint, aucune manipulation n'est nécessaire. En revanche, les données sont entachées de l'erreur sur l'estimation effectuée par l'oculomètre  $\mathcal{O}_{ref}$ .

- *Procédure d'acquisition des données pour l'évaluation d'un oculomètre existant*  $\mathcal{O}_{ex}$  : l'utilisateur regarde une croix immobile sur l'écran, aligne la cible sur cette croix de manière à les superposer dans son champ de vision et déclenche la prise d'images par la caméra de scène et la(les) mesure(s) par l'oculomètre  $\mathcal{O}_{ex}$ . Cette étape est répétée plusieurs fois. Par la suite, la direction du regard est déduite des positions 3D de la croix sur l'écran et de la cible, toutes deux détectées sur l'image de la caméra de scène. Cependant pour la calculer et la comparer à celle estimée par l'oculomètre  $\mathcal{O}_{ex}$ , il est nécessaire de connaître les transformations entre les repères de l'écran, de la cible, de la caméra de scène et de l'oculomètre  $\mathcal{O}_{ex}$ .

Pour que le banc de test soit fonctionnel, une étape d'étalonnage doit préalablement être menée pour obtenir les transformations et les modèles géométriques des caméras. Tout d'abord, le banc stéréo de l'oculomètre  $\mathcal{O}_{ref}$  ainsi que la caméra de scène doivent être étalonnés. Puis, pour la seconde étape consistant à estimer les différentes transformations, nous préférons l'effectuer en ligne à chaque prise d'images, car en pratique, les repères associés à l'écran, à la caméra de scène et à l'oculomètre  $\mathcal{O}_{ref}$  bougent les uns par rapport aux autres, au cours de la procédure d'acquisition, à cause de jeux mécaniques. La solution proposée est de fixer des tags sur la cible et sur la table (où l'oculomètre  $\mathcal{O}_{ref}$  est solidement fixé), qui soient détectables sur l'image de la caméra de scène. L'écran est supposé parfaitement plan, et sa pose est calculée à partir de ses quatre coins, eux aussi détectés sur l'image de la caméra de scène. Il reste alors à estimer le changement de pose entre les caméras du banc stéréo et le tag posé sur la table, ce qui peut être effectué en calant un miroir devant la caméra de manière à réfléchir le tag vers la caméra. La pose de ce miroir peut elle-même être estimée en détectant ses coins ou un autre tag collé sur lui.

### 3.3.2 Oculomètre basé caméras pour le banc de test

L'oculomètre de référence du banc de test doit répondre à certaines contraintes. Tout d'abord, il doit être robuste aux changements de pose de l'utilisateur et fournir une précision suffisante pour définir une base de données d'apprentissage fiable. De plus, il doit s'adapter au changement de luminosité et de personne, afin de fournir des données d'apprentissage dans des configurations variées. Par exemple, l'oculomètre choisi ne doit pas avoir recours aux reflets cornéens, car ceci restreindrait la base de données d'apprentissage aux cas particuliers nécessitant la présence de ces reflets.

Nous avons donc choisi un oculomètre basé modèle 3D estimant la pose de l'œil dans l'infrarouge à partir de la détection de son iris, lequel présente l'avantage de ne pas subir de réfraction à travers la cornée et de présenter un diamètre constant. Toutefois, la détection de l'iris sur une image est complexe. Pour restreindre son espace de recherche, la pupille est d'abord détectée. La seconde étape réalisée par l'oculomètre est d'estimer la pose et le diamètre du cercle de l'iris en s'appuyant sur les deux ellipses détectées sur l'image gauche et droite du banc stéréo. Ce calcul se déroule en deux étapes. La première reprend l'approche de [Nitschke 2013a] pour obtenir une valeur approchée. La seconde raffine le résultat par optimisation à l'aide des données des deux images. En dernier lieu, la pose de l'œil est reconstruite à partir de celle de l'iris en tenant compte d'un modèle 3D utilisé par [Nitschke 2013a] intégrant l'angle  $\kappa$  défini à la section 1.1.2.2.

### 3.3.3 Simulation des photodiodes à partir d'images réelles

Contrairement aux images générées par Blender, celles-ci ne sont pas prises depuis le point de vue des photodiodes simulées. Une étape supplémentaire doit donc être effectuée pour calculer l'image qu'observerait une caméra à l'emplacement de la photodiode à partir de l'image obtenue avec la caméra réelle. Cette nouvelle image peut être générée par DIBR en considérant soit un modèle simple de la surface du visage soit une carte de profondeur calculée à l'aide du banc stéréo. Le calcul de correspondance stéréo pourrait être contraint, car la surface du visage est continue.

La simulation des photodiodes à l'aide d'images de caméras présente une limite due au fait que les caméras proposent des mesures de faibles dynamiques (voir section 2.2). De plus, la position des reflets de la lumière issue de la scène sur l'œil dépend alors du point de vue de la caméra et non du point de vue des photodiodes.

## 3.4 Conclusion

Un concept d'oculomètre basse consommation basé sur un assemblage de photodiodes a été présenté dans ce chapitre. Pour mener à bien l'étude, un simulateur a été mis en place pour tester et évaluer l'impact de la configuration de l'assemblage sur les performances. Sur ce principe, plusieurs assemblages ont pu être testés et des conclusions ont pu en être établies : le champ de vue des photodiodes doit être faible ; elles doivent observer différentes portions de l'œil pour éviter de fournir des données redondantes ; elles doivent être disposées tout du long du contour du verre ce qui permet une observation plus complète de la surface de l'œil. Après ces premiers résultats sur données synthétiques, l'étude se poursuit par des tests sur données réelles. Pour cela, un banc de test est en construction pour acquérir des données d'apprentissage en grand nombre et tester des prototypes d'oculomètres.

## 3.5 Perspectives

La mise en place du banc de test a été effectuée avec l'aide de Thomas Dedieu lors de son stage de Master 2. Elle sera bientôt finalisée (montage mécanique, procédure d'étalonnage et d'analyse de scène, adaptation d'un oculomètre basé caméras à nos contraintes). Ensuite, les premiers tests décrits dans la section 3.3 pourront débiter.

Par la suite, d'autres pistes de travail sont envisagées :

- Nous envisageons de fabriquer un premier assemblage de photodiodes sur la base de nos conclusions de la section 3.2 pour évaluer la faisabilité du concept en situation réelle. Ce montage pourra être évalué avec le banc de test en suivant la procédure décrite précédemment.
- On a pu constater à la section 3.2.3 que les photodiodes doivent disposer d'un faible champ de vue (par extrapolation, la zone de l'œil observée doit être petite) et doivent percevoir des portions différentes de l'œil de telle manière que toute sa surface soit observée. Partant de cette analyse, il serait envisageable de chercher à nouveau plus finement un assemblage optimal en considérant cette fois-ci pour critère la minimisation pour chaque photodiode de la section observée et la maximisation de la surface de l'œil observée par le système total.
- Il peut être intéressant d'ajouter un éclairage dynamique au système à l'aide de quelques LEDs infrarouges réparties le long du verre. Cet éclairage peut apporter davantage de robustesse au changement de luminosité et augmenter le nombre de variables en entrée du réseau pour un minimum d'encombrement. En effet, il serait intéressant d'allumer régulièrement à haute fréquence différentes combinaisons de LEDs et de récupérer les observations des photodiodes. L'ensemble de ces observations effectuées sous différentes illuminations serait ensuite placé en entrée du réseau, en faisant l'hypothèse que la pose de l'œil n'a pas évolué entre les différentes observations. Toutefois, cette approche risque d'être confrontée à certaines difficultés telles que les mouvements rapides de l'œil. Comme les observations sont faites à des instants différents, il est possible que la pupille ait bougé.
- Un démonstrateur final pourrait être mis en place pour voir en action l'oculomètre basse consommation. Pour ce faire, un de ces oculomètres pourrait être intégré sur le prototype de OST-HMD présenté plus tard section 4.2.

Confidentiel

# Prototypage et évaluation

## Sommaire

|            |                                                                                    |            |
|------------|------------------------------------------------------------------------------------|------------|
| <b>4.1</b> | <b>Banc de test basé sur un VST-HMD . . . . .</b>                                  | <b>136</b> |
| 4.1.1      | Description générale . . . . .                                                     | 136        |
| 4.1.2      | Synthèse d'images par DIBR pour le point de vue de l'utilisateur                   | 142        |
| 4.1.3      | Projection de contenu augmenté ou altéré . . . . .                                 | 145        |
| 4.1.4      | Suivi du regard . . . . .                                                          | 146        |
| 4.1.5      | Premiers tests, résultats et performances . . . . .                                | 153        |
| 4.1.6      | Perspectives . . . . .                                                             | 157        |
| 4.1.7      | Conclusion . . . . .                                                               | 161        |
| <b>4.2</b> | <b>Banc de test basé sur un OST-HMD . . . . .</b>                                  | <b>161</b> |
| 4.2.1      | Dispositif et composants logiciels . . . . .                                       | 162        |
| 4.2.2      | Tests et résultats d'étalonnage d'OST-HMD . . . . .                                | 165        |
| 4.2.3      | Limites du système actuel et perspectives . . . . .                                | 168        |
| <b>4.3</b> | <b>Prototype de lunettes avec obscurcissement automatique des verres . . . . .</b> | <b>170</b> |
| 4.3.1      | Choix des capteurs de luminosité . . . . .                                         | 170        |
| 4.3.2      | Choix d'une commande du verre en fonction de la luminosité                         | 172        |
| 4.3.3      | Test sur simulateur et mise en place du prototype . . . . .                        | 173        |
| 4.3.4      | Perspectives . . . . .                                                             | 175        |
| <b>4.4</b> | <b>Conclusion . . . . .</b>                                                        | <b>178</b> |

Ce chapitre présente successivement les trois prototypes mis au point : un VST-HMD, un OST-HMD et des lunettes actives équipées de verres électro-chromes.

Les technologies de verres actifs présentées plus haut n'étaient pas toutes disponibles au début de cette thèse. Les deux premiers prototypes (VST-HMD et OST-HMD) ont donc été envisagés pour servir de bancs de prototypage ou de test. Ils permettent de simuler des verres actifs et de tester rapidement différentes solutions matérielles et logicielles pour des primitives de perception, décision et action sur les verres. L'évaluation de la pertinence de diverses solutions pourrait également être menée sur différents prototypes fabriqués et non simulés. Cette option serait cependant très coûteuse en temps.

Pour le premier banc de test, le choix du VST-HMD apporte deux avantages sur l'OST-HMD. D'une part, le champ de vision étudié à travers un casque VST-HMD est plus important que sur un OST-HMD. D'autre part, la simulation de verres obscurcissants ou de focalisation dynamique est plus aisée sur les écrans opaques d'un VST-HMD.

Pour le second banc de test, la technologie du OST-HMD permet de tester l'impact des verres transparents sur la perception de la scène par l'utilisateur. En effet, sur un VST-HMD la scène et le contenu synthétique sont affichés par le biais d'un même écran, contrairement aux OST-HMD, où les chemins optiques empruntés par les rayons issus de la scène réelle diffèrent de ceux empruntés pour la génération d'image synthétique. Cette différence de perception engendre des procédures d'étalement différenciées pour les OST-HMD et les VST-HMD (Section 2.6).

Le troisième prototype est issu d'un concept de lunettes actives proposé par Essilor. Derrière ce concept, l'objectif est d'intégrer dans un système basse consommation des algorithmes de perception de l'environnement, de décision, et d'action sur des verres obscurcissants. Pour l'étude de ce système, les algorithmes de perception ont été préalablement développés et testés sur le banc de test basé sur le VST-HMD. Le prototype a été fabriqué par la suite pour servir de démonstrateur.

## 4.1 Banc de test basé sur un VST-HMD

L'objectif défini pour le premier banc de test est de développer, tester et évaluer les composants logiciels et matériels requis pour les applications citées dans la section 1.2.1. Parmi ces composants, on retrouve *la perception* de la scène *la synthèse de l'image* affichée par le casque, *le suivi du regard* et *la localisation* du casque dans la scène.

Au commencement de la thèse, Essilor ne disposait pas de prototype de lunettes équipées de verres actifs adressables sur lesquels les développements auraient pu être menés. Nous avons donc choisi de construire un banc de test pouvant simuler tous types de verres actifs et nous l'avons équipé de fonctions essentielles. Un casque de VST-HMD a donc été conçu pour servir de banc de test. Son écran opaque et son large champ de vision rendent plus aisée la simulation des propriétés de verres actifs. D'autre part, les OST-HMD disponibles dans le commerce au début de la thèse disposaient d'un affichage ne couvrant qu'approximativement 20 degrés de champ de vision, contrairement à notre prototype qui couvre 110 degrés. La conception de ce premier banc de test a pris en compte deux types de contraintes : celles liées aux lunettes actives et celles liées au délai de fabrication et de développement.

Dans cette section, nous proposons, tout d'abord, une description d'ensemble du banc de test. Certains des modules le constituant sont ensuite détaillés. Puis, nous présentons les premiers tests effectués sur le prototype et les perspectives envisagées.

### 4.1.1 Description générale

#### 4.1.1.1 Composants logiciels et matériels

**A Système d'affichage et banc stéréoscopique large champ de vision**  
Le prototype de VST-HMD (visible sur la figure 4.1(a)) est constitué d'un casque

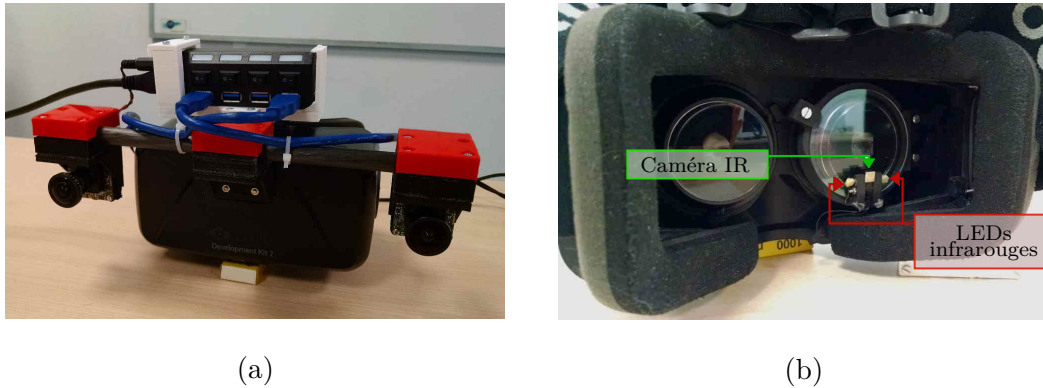


FIGURE 4.1 – Prototype mis au point au début de la thèse. (a) Face avant. (b) Face arrière avec l'oculomètre.

immersif Oculus DK2 et d'un banc stéréoscopique de caméras<sup>1</sup> fisheye synchronisées de 1.3 Mégapixel, d'environ 180 degrés et d'écart intra-oculaire 24 cm. Ce grand angle d'ouverture couvrant la presque totalité du champ de vision humain a été choisi pour permettre au dispositif de réagir à tout événement visible par le porteur. Les caméras sont fixées de part et d'autre du casque afin de se rapprocher de la configuration d'un OST-HMD ou de lunettes actives, dont les verres sont transparents (ou semi-opaques). En effet, sur ces dispositifs, les caméras doivent être intégrées dans la monture près des charnières (de part et d'autre du visage) pour plus de discrétion. D'autre part, un grand écart inter-oculaire permet d'estimer plus finement la profondeur d'objets de la scène plus éloignés qu'avec un petit écart-interoculaire, ce qui améliore le processus de localisation d'algorithmes de SLAM. Cependant, les images issues des caméras ne doivent pas être directement affichées dans le VST-HMD. Les points de vue des caméras et des yeux de l'utilisateur diffèrent, ce qui est susceptible de provoquer un inconfort (maux de tête, nausée). Pour y remédier, de nouvelles images de la scène doivent être générées pour le point de vue des yeux du porteur. La synthèse de ces images est réalisée à l'aide d'un algorithme de DIBR (Depth Image Based Rendering) dont une étude bibliographique est disponible dans la section 2.5.2. Cet outil s'applique sur les images acquises par les caméras du VST-HMD et tient compte du changement de pose entre les caméras et l'œil de l'utilisateur. Pour estimer ce changement de pose, nous nous sommes inspirés des travaux de [Itoh 2014]. Le principe est de positionner approximativement une caméra  $C_0$  à la position de l'œil et d'estimer la transformation rigide entre le banc stéréo et la caméra  $C_0$  (Figure 4.2). Un damier est alors placé devant le casque de façon qu'il puisse être observé par les caméras du banc stéréo. À partir des images du banc filmant le damier, on déduit la pose du damier par rapport aux caméras. Par la suite, on maintient la caméra  $C_0$  dans sa situation, mais on enlève le casque. On estime alors la pose du damier par rapport à la caméra  $C_0$ . On

1. Des caméras Ueye UI-3241LE-C-HQ de IDS Imaging de 1280x1024 pixels avec une optique Lensation de focale 2.95mm.

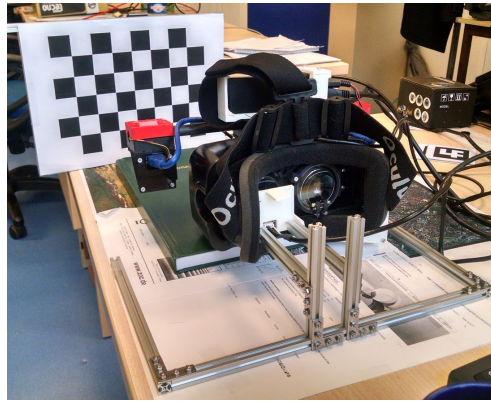


FIGURE 4.2 – Illustration de la procédure d'étalonnage où une caméra est placée dans le casque à l'emplacement d'un œil et un damier est positionné devant le casque.

déduit du calcul de ces deux transformations rigides le changement de pose entre le banc stéréo et la caméra positionnée à l'emplacement de l'œil. Cette procédure est répétée pour chacun des yeux.

**B Estimation de la direction du regard** Pour mesurer la direction du regard, un oculomètre utilisant une micro-caméra et un système d'éclairage infrarouge a été conçu (avec les conseils d'une équipe francilienne de R&D d'Essilor) et inséré dans le casque devant l'œil droit (Figure 4.1(b)). Ce système détaillé section 4.1.4 estime la position de l'œil droit et la direction pointée.

**C Localisation des lunettes dans la scène** Le composant de localisation, estimant la pose des lunettes, ne doit pas s'appuyer sur des équipements présents dans la scène (caméra, tag), pour que le dispositif puisse fonctionner en environnement inconnu. La solution choisie est un algorithme de SLAM (Simultaneous Localization And Mapping) appliqué aux images du banc stéréo (Section 2.3).

**D Perception 3D de la scène** Pour la perception de la géométrie de la scène, il serait intéressant d'équiper le banc de test d'un système de vision active pour bénéficier de ses avantages en matière de fréquence d'acquisition et de qualité de la carte de profondeur. Toutefois, comme le dispositif doit s'utiliser en intérieur comme en extérieur, on choisit d'éviter l'usage de la vision active, dont les mesures sont bruitées en présence du soleil. De plus, les capteurs commerciaux actifs disponibles ne possèdent pas un grand champ de vision. C'est pourquoi la perception de la profondeur de la scène est effectuée par un algorithme de stéréovision passive. Le choix de cet algorithme présente un autre avantage : la compatibilité entre le DIBR et la correspondance stéréo. Les points faibles de la vision stéréo (zones unies ou motifs répétitifs) sont compensés par le DIBR, qui reconstruit une image visuelle

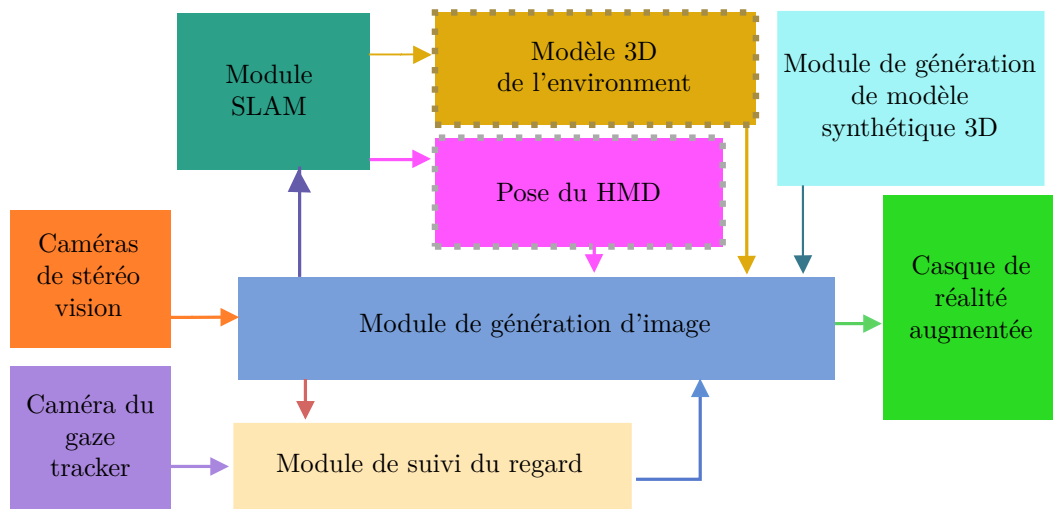


FIGURE 4.3 – Schéma-bloc de l'architecture logicielle générale.

cohérente malgré les erreurs de profondeur. Cette règle reste vraie si les points de vue de l'image synthétisée et de la caméra réelle sont proches.

Le casque et les caméras sont connectés à un ordinateur sur lequel s'effectue l'ensemble des calculs.

#### 4.1.1.2 Architecture logicielle

Pour simplifier le développement et le portage des solutions sur d'autres dispositifs, l'architecture logicielle choisie est modulaire et se base sur le middleware ROS (Robot Operating System). Chaque composant mis en place est ainsi défini dans des modules séparés qui communiquent entre eux. De la sorte, les logiciels sont écrits indépendamment les uns des autres et sont interfacés par la suite, facilitant alors le développement. D'autre part, nous pouvons facilement intégrer des modules déjà existants. Dans les schémas-blocs présentés ci-dessous, les modules définis ne correspondent pas strictement aux noeuds ROS. Pour plus de clarté et pour éviter de surcharger les schémas, nous choisissons de définir également des modules plus conceptuels (tel que le module génération d'image) intégrant d'autres modules (tel que le DIBR).

La figure 4.3 résume les principales parties du système avec leurs interactions. Le module génération d'image (GI), détaillé dans la figure 4.4, génère les images couleur et les cartes de profondeur nécessaires pour le système. Tout d'abord, les images issues du banc stéréo de caméras fisheye, préalablement étalonnées à l'aide du modèle de Scaramuzza décrit à la section 2.2.1.4, sont rectifiées épipolairement. Les images rectifiées sont générées à l'aide d'une projection perspective sur un plan commun pour les deux caméras (en configuration rectifiée épipolairement) en s'appuyant sur le modèle trou d'épingle (Figure 4.5). Ce choix nous permet d'utiliser ces images pour des algorithmes (SLAM, correspondance stéréo) déjà disponibles sur étagère, qui ne gèrent pas d'autre modèle de projection. Cependant, si la projection

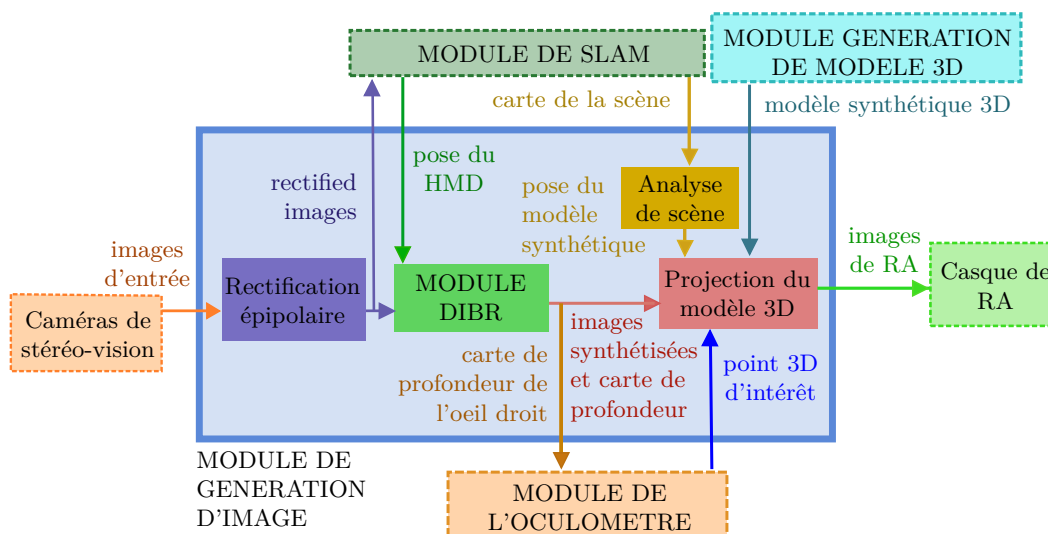


FIGURE 4.4 – Schéma-bloc du processus de génération d'images.

perspective est appliquée à l'ensemble du champ de vision de la caméra, l'image générée est alors sur-échantillonnée<sup>2</sup> dans la périphérie du champ de vision de la caméra. Pour éviter ce phénomène, nous n'utilisons que 110 degrés du champ de vision des caméras.

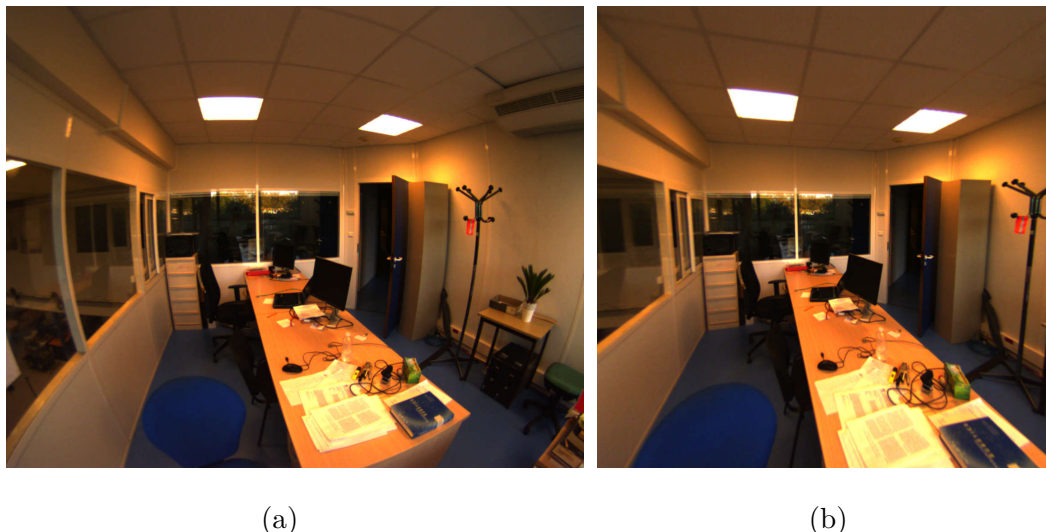


FIGURE 4.5 – Image acquise par la caméra gauche du banc stéréo monté sur le VST-HMD (a) et sa projection perspective après rectification épipolaire (b).

Les images rectifiées sont ensuite utilisées par le SLAM et l'algorithme de stéréovision (inclu dans le module DIBR), qui calcule une carte de profondeur par des

<sup>2</sup>. L'information de luminance de la scène est définie sur plus de pixels. L'image est comme étirée.

techniques standard de corrélation stéréoscopique. Puis les images sont converties par le module DIBR en des images synthétiques adaptées au point de vue des yeux de l'utilisateur. Ces étapes sont détaillées à la section 4.1.2. Par la suite, le module général GI ajoute le contenu augmenté en tenant compte du modèle 3D de la scène et du point 3D fixé par l'utilisateur, qui est déduit de la carte de profondeur et de la direction du regard calculée par le module de l'oculomètre (OC) décrit à la section 4.1.4. Le modèle de la scène est défini comme un nuage de points 3D dynamiquement généré par le processus de SLAM. Ce modèle 3D est analysé pour détecter des structures géométriques caractéristiques (par exemple planaires) et déterminer les zones de la scène où le contenu virtuel pourrait être ajouté. Pour le moment, nous avons implémenté une simple détection de plan robuste sur le nuage de points 3D par la méthode RANSAC<sup>3</sup>, dans le but de poser un modèle 3D se déplaçant à la surface du plan détecté. Des processus plus complexes pourront être envisagés dans le futur.

Certains algorithmes cités précédemment sont disponibles sur étagère en open-source :

- Le SLAM choisi est ORB-SLAM2, un logiciel C++ développé par [Mur-Artal 2017a], utilisant les éléments caractéristiques “Oriented FAST and Rotated BRIEF” (ORB). Cette implémentation est plus robuste de part sa fiabilité de relocalisation de la caméra. Lors de la conception du banc de test, ORB-SLAM2 était le SLAM open source basé stéréovision le plus performant.
- Le logiciel de mise en correspondance stéréo est un code C++ issu des contributions externes distribuées avec OpenCV 3.1. Il s'agit d'un algorithme de Semi-Global Block Matching (SGBM) inspiré des travaux de [Hirschmuller 2008] et s'appuyant sur le descripteur CENSUS (Section 2.2.2.2).

D'autres algorithmes ont été spécifiquement développés pour notre projet :

- Une boîte à outils open source pour l'étalonnage de caméra omnidirectionnelle basée sur le modèle de [Scaramuzza 2006a] a été étendue à une configuration binoculaire. Cet outil permet d'étalonner le capteur stéréo large champ de vision, afin de réaliser par la suite des mesures 3D à partir des images. L'étalonnage de notre banc stéréoscopique a atteint une erreur de reprojection moyenne de l'ordre de 0.2 pixels.
- Un algorithme de DIBR a été mis au point pour générer des images synthétiques pour différents points de vue à partir d'une ou plusieurs caméra(s) de position variable.

---

3. La méthode RANSAC (RANDOM Sample Consensus) estime les  $x$  paramètres d'un modèle associé à un jeu de données, tout en étant robuste aux données aberrantes. Le principe est d'estimer les paramètres à partir de  $y$  données choisies au hasard (avec  $y \geq x$ ) et de compter le nombre de données respectant le critère de proximité au modèle. Cette étape est répétée  $n$  fois et le plus grand ensemble de données ayant respecté le critère est conservé. Le modèle est alors recalculé sur cet ensemble de données débarrassé des valeurs aberrantes. La probabilité d'obtenir un bon résultat dépend de la taille de  $n$ .

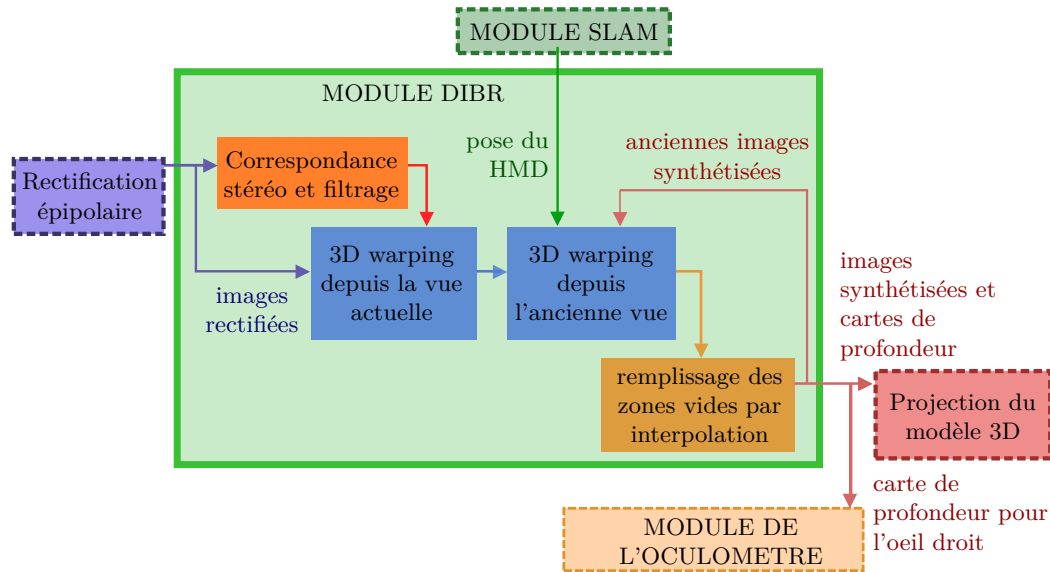


FIGURE 4.6 – Schéma-bloc résumant les étapes du DIBR. Le processus inclut l'algorithme de correspondance stéréo estimant une carte de profondeur. La carte est ensuite transmise au module de "3D warping", qui synthétise deux premières images pour les deux yeux. Les deux étapes suivantes remplissent les zones restées vides.

- Un algorithme estimant la pose de l'œil a été implémenté. Il s'inspire des travaux de [Swirski 2013].
- L'analyse de la scène développée pour notre banc de test consiste à détecter des surfaces planes pour y incruster des éléments synthétiques.

#### 4.1.2 Synthèse d'images par DIBR pour le point de vue de l'utilisateur

Le module responsable de la génération de nouvelles images pour le point de vue de l'utilisateur est le DIBR, dont la structure interne est visible sur la figure 4.6. Il se déroule en plusieurs étapes, détaillées ci-dessous, et dont les résultats sont visibles sur les images de la figure 4.7.

##### 4.1.2.1 Calcul de la carte de profondeur

La première étape consiste à calculer la carte de profondeur à partir des images rectifiées à l'aide d'un algorithme de corrélation stéréoscopique. Nous utilisons l'implémentation open source C++ de Semi-Global Block Matching (SGBM) issue de OpenCV 3.1 s'appuyant sur le descripteur CENSUS (Section 2.2.2.2). Cette implémentation étant sur CPU, elle ne permet pas d'atteindre les fréquences acceptables pour un usage confortable du casque. Néanmoins, elle nous permet d'avoir un premier rendu et d'évaluer ce que l'emploi du DIBR apporte à notre système. D'autres algorithmes de correspondance stéréo existent avec de meilleures performances temporelles, mais généralement au détriment de la qualité de la carte calculée. Nous

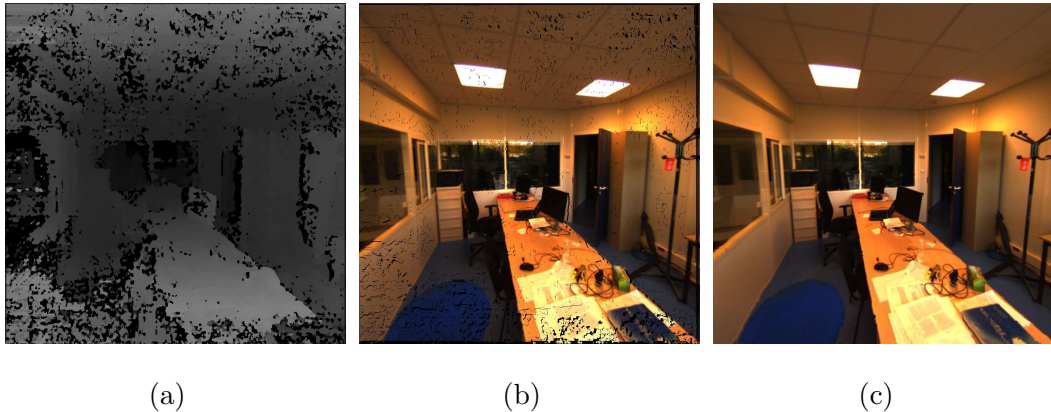


FIGURE 4.7 – Résultats des différentes étapes du DIBR appliquées sur l'image de la figure 4.5(b). (a) est la carte de disparité calculée à partir de la paire d'images stéréo. (b) est l'image générée par le "3D warping", contenant encore des pixels vides affichés en noir. (c) est l'image synthétique finale après l'étape de remplissage.

avons choisi au départ de mettre de côté la contrainte de temps réel pour étudier les performances du DIBR appliqué sur des cartes de profondeur issues de la stéréovision. L'algorithme SGBM avec le descripteur CENSUS est un choix intéressant de par la qualité des cartes obtenues et sa disponibilité en open source.

Malgré ce choix, la carte de disparité calculée par corrélation stéréoscopique présente encore de nombreux défauts. Pour y remédier, deux techniques sont reprises de [Kauff 2007]. Tout d'abord, la ressemblance des patches de pixels gauche et droite est évaluée, pour effacer la profondeur des patches dissemblables. Les valeurs de profondeurs manquantes résultant de cette opération sont ensuite calculées par des techniques de in-painting. Ce procédé permet de supprimer le bruit ou les données aberrantes en les remplaçant par des valeurs cohérentes avec leur voisinage. Un second traitement, appelé filtre bilatéral, a été testé pour lisser la carte de disparité tout en conservant les contours des objets. Cette technique réduit les effets de craquelure (appelé "cracks") observés après projection de l'image d'origine sur la nouvelle image. Son implémentation actuelle est également sur CPU et nécessiterait de passer sur GPU pour réduire son temps d'exécution.

#### 4.1.2.2 Étape de "3D warping"

La seconde étape est celle du 3D warping. Elle consiste à projeter l'image et les cartes de profondeur sur de nouvelles images définies pour une pose différente. Dans notre cas, nous choisissons d'appliquer le "direct 3D warping" pour des raisons de coût calculatoire et de rapidité d'implémentation. De plus, cette technique est facilement parallélisable et pourra plus tard être implémentée sur GPU. Une description des différentes techniques de "3D warping" est disponible à la section 2.5.2. Pour notre application, cette étape s'applique sur l'image réelle la plus proche de la nouvelle image à générer. Ainsi, l'image gauche (resp. droite) du banc stéréo est

prise en compte pour la génération de l'image correspondant à l'œil gauche (resp. droit) de l'utilisateur. L'utilisation de la seconde image permettrait de remplir en grande partie les pixels restés vides après la première projection. Toutefois dans notre cas, nous avons pu constater que la deuxième caméra est trop éloignée pour que le processus de 3D warping fournisse de belles images. En effet, plus les caméras réelles et synthétiques sont éloignées, plus les valeurs de profondeur erronées engendrent des défauts dans l'image reconstruite.

#### 4.1.2.3 Remplissage des zones vides

La dernière étape consiste à associer une valeur photométrique aux pixels non affectés après "3D warping". Une bibliographie sur ces techniques est disponible à la section 2.5.2.2. Sur notre banc de test, deux techniques ont été testées : un remplissage par interpolation et un remplissage par "exemplar based in-painting" (EBI) [Criminisi 2003]. La technique de EBI a l'avantage de reconstruire des portions texturées en concordance avec le reste de l'image. Mais la méthode originale nécessite quelques secondes d'exécution pour un seul trou puisqu'elle effectue, pour chaque pixel à remplir, une recherche sur toute l'image. Afin d'accélérer le processus, trois variantes ont été testées. La première effectue la recherche sur une sous-image centrée autour du trou. La seconde cherche sur le pourtour du trou. La dernière focalise son exploration à proximité du pixel dont la valeur doit être estimée. Ces trois variantes atteignent de meilleures performances pour le remplissage d'une zone vide. Toutefois, elles ne sont pas applicables à notre situation. Comme l'image générée par le "3D warping" est remplie de nombreuses zones vides, leurs remplissage requiert plusieurs secondes de calculs. Une solution possible est d'effectuer cette tâche à basse fréquence sur certaines images uniquement et de propager le remplissage sur les images suivantes par "3D warping". [Álvarez 2017] propose cette solution pour effacer un objet d'un flux vidéo. Toutefois, cette approche nécessite une procédure de remplissage de durée d'exécution stable. Dans notre cas, le nombre et la taille des zones vides fluctuent d'une image à l'autre et le temps nécessaire pour l'étape de remplissage n'est donc pas prévisible et peut parfois être très long, rendant alors cette solution non envisageable. Par ailleurs, l'EBI ne peut pas être porté sur GPU, car le processus est séquentiel et remplit les pixels les uns après les autres suivant un ordre de priorité. Après quelques tests, nous avons préféré opter pour un remplissage par interpolation. Pour les zones vides de petites tailles, les valeurs des pixels sont interpolées sur leurs pixels voisins. Pour les trous de plus grande taille, le remplissage est itératif et s'effectue depuis le bord jusqu'au centre en s'aidant de la méthode "distance transform" [Felzenszwalb 2012] pour déduire l'ordre des pixels à remplir. Cette méthode associe à chaque pixel de la zone vide sa distance au pixel non vide le plus proche. Plus tard, d'autres techniques de remplissage devront être envisagées afin de réduire le temps de calcul et permettre la portabilité sur GPU. Un exemple d'image générée par DIBR suivant notre implémentation est visible sur la figure 4.7(c).

Une solution supplémentaire a été proposée pour remplir les pixels vides et

réduire l'incohérence temporelle du remplissage de trous susceptible de générer du scintillement. L'idée est d'effectuer un lissage temporel en réalisant une étape de "3D warping" pour projeter la dernière image générée par DIBR sur les pixels vides, avant d'effectuer l'étape d'interpolation. Cette technique n'a cependant pas encore été incorporée. Comme elle pratique le "3D warping", la distance entre les points de vue de l'ancienne et nouvelle image doit être faible. Cette contrainte sur la distance se traduit par le besoin de faibles mouvements ou d'une fréquence de traitement élevée. Or le processus de DIBR est actuellement très lent (de l'ordre d'une seconde), de sorte qu'on ne peut pas appliquer cette étape de "3D warping".

Les méthodes basées apprentissage présentées à la section 2.5.2.3 n'ont pas pu être appliquées ici, car la seule méthode open-source, temps réel, et générant des images acceptables pour notre application, appelée Deep3D [Xie 2016], ne peut générer que des images à une certaine position par rapport à l'image de référence. Le réseau de neurones a été entraîné sur des paires d'images stéréoscopiques pour générer une l'image de gauche à partir de l'image de droite distante de l'écart interoculaire. Cette technique n'est donc pas applicable au banc de test. Il nous faudrait un outil capable de générer une nouvelle image à des positions variables comme le DeepStereo [Flynn 2016], qui n'est pas disponible en open source. Mettre en place un tel outil nécessite du temps et une très grande base de données pour l'apprentissage du réseau. Nous n'avons donc pas opté pour ce type de solution. Toutefois, dans de futurs travaux, il pourrait être intéressant de mettre au point un DIBR basé apprentissage spécifique pour notre application, car ces méthodes génèrent des images visuellement acceptables en très peu de temps.

#### 4.1.3 Projection de contenu augmenté ou altéré

Après DIBR, les images synthétisées subissent d'autres traitements avant d'être affichées dans le casque. Dans un premier temps, elles peuvent être traitées pour simuler des verres actifs obscurcissants ou de focale variable. Ces traitements consistent à foncer l'image ou à simuler des défauts de mise au point. Les étapes suivantes de synthèse d'images sont effectuées avec la bibliothèque OpenGL. Dans un second temps, les images préalablement traitées sont affichées comme image de fond pour chaque œil et les cartes de profondeur sont utilisées en entrée pour initialiser le Z-buffer, qui permet de gérer les occultations en comparant les profondeurs des pixels calculés. L'augmentation des images est ensuite obtenue par projection de modèles maillés texturés. Lors de cette étape, le test du Z-buffer détermine pour chaque pixel s'il appartient à la scène réelle ou au contenu virtuel visible en avant-plan. Les caméras considérées pour l'étape de DIBR et de rendu sous OpenGL sont basées sur le même modèle géométrique utilisé par le SDK du casque Oculus DK2. En conséquence, les objets 3D virtuels sont intégrés de manière cohérente dans l'image de la scène réelle. À la fin, le kit de développement de l'Oculus DK2 applique un dernier traitement, qui prend en compte les distorsions et les aberrations chromatiques causées par les optiques bon marché de l'Oculus.

La figure 4.8 montre une paire d'images stéréoscopiques augmentées, générée



FIGURE 4.8 – Images stéréoscopiques affichées sur l’écran du VST-HMD incluant la projection d’un modèle synthétique de pingouin sur la surface d’un bureau.

par notre système et affichée dans le HMD. Elle possède quelques artefacts près des tiroirs du bureau, dus aux erreurs dans la carte de profondeur. L’image 4.9 montre les corrections des aberrations chromatiques engendrées par les lentilles du VST-HMD.

#### 4.1.4 Suivi du regard

De nombreuses solutions d’oculométrie (estimation de la direction du regard) existent et sont présentées à la section 2.4. Ces systèmes peuvent également fournir d’autres informations : la pose des yeux, la taille de la pupille et les clignements. Sur ce dispositif, nous souhaitons estimer la pose de l’œil (orientation et position) relativement à l’oculomètre fixé sur les lunettes. Pour choisir le dispositif adapté à nos besoins, nous avons défini plusieurs critères :

- Le système doit être compact, afin d’être intégrable à terme dans des lunettes et confortable pour l’utilisateur.
- Il doit fournir des mesures correctes indépendamment des changements de pose des lunettes sur le nez. En pratique, elles se déplacent souvent le long du nez et l’utilisateur peut être amené à les retirer et le remettre plusieurs fois par jour. Ces mouvements réguliers posent problème lorsque l’oculomètre ne s’adapte pas aux changements de pose, ce qui concerne une grande partie des systèmes existants.
- Certains de ces dispositifs nécessitent un étalonnage pour chaque individu. En effet, comme la forme du visage et les dimensions géométriques de l’œil varient d’une personne à l’autre, un étalonnage personnalisé permet souvent d’atteindre de meilleures performances en matière de précision angulaire sur

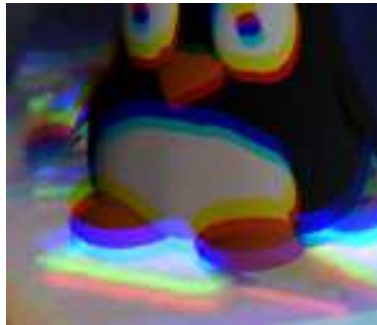


FIGURE 4.9 – Zoom sur les pieds du pingouin de l'image stéréoscopique destinée à être affichée dans le VST-HMD (Figure 4.8). On observe sur l'image un décalage des composantes rouge, vert et bleu, correspondant à une compensation logicielle (proposée par le SDK de l'Oculus DK2) des aberrations chromatiques provoquées par les lentilles du VST-HMD. Ainsi, quand l'image est perçue au travers des lentilles, la correction et les aberrations chromatiques se compensent.

l'estimation de la direction du regard. Il est souhaitable que cette procédure soit simple et rapide.

- Les lunettes seront utilisées en intérieur comme en extérieur. Les algorithmes de perception devront donc être robustes aux changements de luminosité.

La première contrainte basée sur le confort et l'esthétique des lunettes met de côté les techniques incluant un contact avec la peau ou l'œil par le biais d'électrodes ou de lentille de contact (Section 2.4.1). Nous nous sommes alors intéressés aux méthodes basées sur l'analyse de la réflexion de la lumière sur la surface de l'œil. Comme le système doit être robuste aux changements de luminosité, nous avons choisi de concevoir un dispositif équipé de LED infrarouge éclairant la surface de l'œil, lui-même observé par une caméra sensible à l'infrarouge. Les algorithmes basés caméra apportent deux avantages. D'une part, nous bénéficions des nombreuses avancées effectuées dans ce domaine, bien plus étudié que les autres. D'autre part, certains de ces algorithmes sont robustes aux changements de pose.

La caméra de l'oculomètre est équipée d'un filtre passe bande placé devant son objectif laissant traverser la lumière infrarouge et coupant la lumière du visible. Ainsi, grâce au filtre, l'image de l'œil ne contient pas de reflets cornéens générés par l'écran du VST-HMD. En pratique, de tels reflets réduiraient (voire obstrueraient) l'image de la pupille et pourraient gêner l'oculomètre. D'autre part, l'espace au sein du casque étant petit, un assemblage particulier a été considéré. Il inclut, entre les lentilles du VST-HMD et l'utilisateur, un miroir semi-réfléchissant, ne réfléchissant que l'infrarouge et laissant traverser la lumière du visible. Ainsi au lieu d'être orientées directement vers l'œil, les LED pointent vers l'écran, où le miroir réfléchit la lumière vers l'œil. De même, le reflet de l'œil est observé par la caméra à travers ce miroir. L'assemblage résultant ne nécessite que quelques millimètres et n'obstrue pas le champ de vision de l'utilisateur. Cette architecture peut aussi être utilisée sur un OST-HMD, si sa configuration le permet. Notre solution est

visible sur la figure 4.1. Les algorithmes considérés pour notre application n'ont pas recours à la détection de reflets spéculaires de LED à la surface de la cornée, car ces détections ne sont pas robustes aux changements de luminosité et peuvent échouer lorsque d'autres reflets provenant de la lumière du soleil apparaissent.

En équipant chaque œil d'un oculomètre, il est possible d'estimer la direction des deux yeux. Puis, en calculant l'intersection de ces deux directions, on déduit la position 3D du point d'intérêt (PI) de la scène pointé par l'utilisateur. Cependant, nous avons choisi pour ce prototype de ne concevoir qu'un seul oculomètre pour l'œil droit afin de minimiser le nombre d'éléments à intégrer. Pour calculer le PI, nous utilisons la carte de profondeur obtenue à partir des images du banc stéréo. Le PI se calcule alors comme l'intersection entre la direction de l'œil droit et la carte de profondeur.

Deux approches ont été implémentées et testées sur l'oculomètre du VST-HMD : une basée interpolation et une basée modèle 3D.

#### 4.1.4.1 Implémentation d'un algorithme basé interpolation

Un premier algorithme d'oculométrie, facile à mettre en place, a été testé sur le VST-HMD. Il s'agit d'une méthode basée interpolation similaire à celle de [Kassner 2014]. Elle se déroule en deux étapes. La première vise à détecter le centre de l'ellipse associée à l'image de la pupille et la seconde déduit la direction du regard. Le centre de l'ellipse sur l'image est souvent assimilé à la projection du centre de la pupille, néanmoins il s'agit d'une approximation.

Pour la première étape, nous nous sommes appuyés sur les travaux de [Świrski 2012], [Kassner 2014] et [Li 2005]. Nous effectuons au départ une recherche approximative de la pupille (technique reprise de [Świrski 2012]), qui nous permet de définir une sous-image de plus petite taille où effectuer les traitements ultérieurs (Figure 4.10.a). Puis à l'aide d'un K-means<sup>4</sup> sur l'histogramme de cette sous-image, on définit un seuil pour binariser l'image de manière à segmenter la pupille sur le blanc de l'œil (Figure 4.10.b). Ensuite après segmentation, les points du contour détecté sont triés sur un critère de continuité de la courbure. Ce procédé permet d'éliminer quelques points aberrants et portions de contours erronées, tous deux causés par des reflets cornéens ou par les paupières (Figures 4.10.c et 4.10.d). Ensuite, à partir des points conservés, on estime une ellipse par la méthode des moindres carrés. Cette estimation est implémentée de manière robuste à l'aide de RANSAC (Figures 4.10.e et 4.10.f), où l'on définit un critère de proximité des points à l'ellipse. Cette proximité se mesure comme la distance entre le point  $P_c$  du contour et un point  $P_e$  de l'ellipse situé le long du rayon de l'ellipse passant par  $P_c$ . Cette distance est assimilable à la distance du point à l'ellipse lorsque l'ellipse est un cercle.

---

4. Le K-means est une technique de partitionnement (ou segmentation) permettant de partitionner un ensemble de données en plusieurs groupes dont le nombre est préalablement fixé. La proximité d'une donnée à un groupe est définie comme la distance la séparant de la valeur moyenne des données du groupe. Le partitionnement se déroule de manière itérative jusqu'à convergence. À chaque itération, on associe les données à la partition la plus proche, puis on calcule la nouvelle moyenne de la partition modifiée.

Une seconde méthode de détection des points du contour de l'ellipse a été testée et prend la relève quand la première échoue en ne satisfaisant pas certaines conditions (taille de l'ellipse, proximité des points à l'ellipse). Elle s'inspire des travaux de [Li 2005]. À partir de la position approximative du centre de l'ellipse, plusieurs directions sont parcourues à la recherche de fort gradient signalant la présence du contour de la pupille. Dès qu'une valeur de module du gradient importante est détectée, le point est enregistré comme un point potentiel du contour. Ensuite, l'ensemble des points stockés suivent la même procédure d'estimation robuste d'ellipse. L'image de module du gradient considérée pour cette méthode est calculée par le filtre de Sobel (Figure 4.11(a)). Pour supprimer le bruit causé par les reflets cornéens, ces derniers sont détectés et segmentés (Figure 4.11(b)). Leurs formes sont ensuite dilatées. Puis tous les pixels de l'image de module du gradient inclus dans ces formes sont remis à zéro (Figure 4.11(c)).

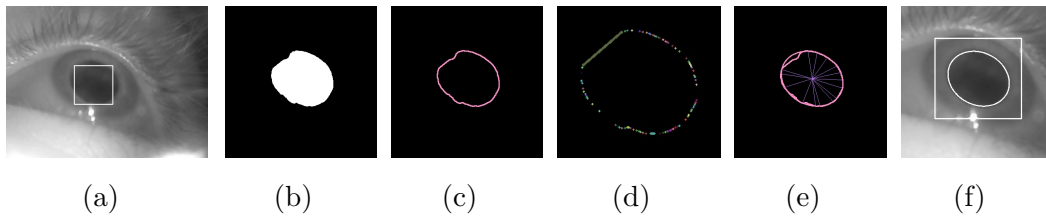


FIGURE 4.10 – Résultats des différentes étapes de la détection de pupille et de l'estimation d'ellipse : (a) détection d'une pupille approximative carrée, (b) segmentation de la pupille, (c) extraction du contour de la forme segmentée, (d) tri des points du contour suivant un critère de continuité de la courbure, (e) estimation d'une ellipse et tracé des points respectant la contrainte de proximité, (f) résultat final.

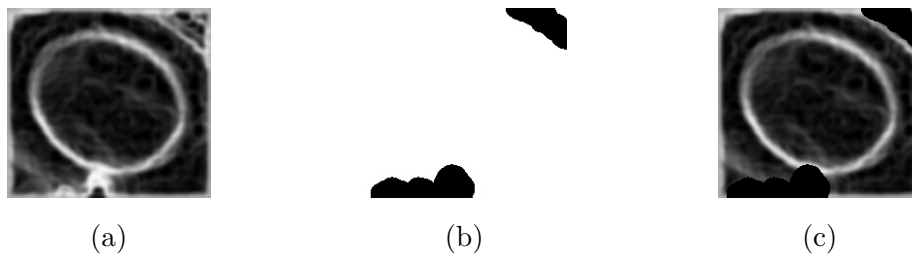


FIGURE 4.11 – Illustration de l'étape de suppression des reflets cornéens : (a) image de contour calculé par un filtre de Sobel, (b) image binaire délimitant les zones très claires, (c) suppression des contours générés par ces zones.

La seconde étape de l'oculomètre basé interpolation est le calcul de la direction du regard à partir des coordonnées du centre de la pupille ( $u_{pc}, v_{pc}$ ). Ici, la direction du regard est définie comme le pixel ( $u_s, v_s$ ) de la caméra de scène (caméra de droite du banc stéréoscopique pour notre implémentation) correspondant à la projection du point d'intérêt. Notre objectif est donc d'estimer une fonction permettant d'associer un point du plan image de la caméra de scène à un point du plan image de l'oculomètre. La fonction choisie est polynomiale de degré deux

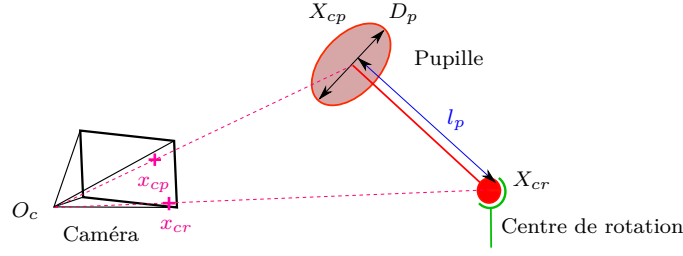


FIGURE 4.12 – Schéma du modèle géométrique et dynamique de l'œil. La pupille est représentée par un disque rouge de diamètre  $D_p$  variable. Ce dernier est en rotation autour d'un point  $X_{cr}$  à une distance  $l_p$  de 9.27 mm du centre de la pupille  $X_{cp}$ . La caméra de l'oculomètre de centre optique  $O_c$  est représentée en noir. Sur son image, la projection du centre de la pupille et du centre de rotation sont représentées par  $x_{cp}$  et  $x_{cr}$ . Le point  $x_{cp}$  mesuré comme le centre de l'ellipse sur l'image est une approximation de la projection de  $X_{cp}$ .

définie par les équations 4.1 et 4.2. Ainsi, deux fonctions  $f_1$  et  $f_2$  sont définies pour estimer respectivement  $u_s$  et  $v_s$  et leurs coefficients  $(a_{1,1}, \dots, a_{1,6})$  pour  $f_1$  et  $(a_{2,1}, \dots, a_{2,6})$  pour  $f_2$  doivent être estimés lors d'une étape d'étalonnage. Au cours de cette étape préliminaire, l'utilisateur doit regarder successivement neuf points distribués homogènement sur un écran et localisés par la caméra de scène.

$$\begin{bmatrix} u_s \\ v_s \end{bmatrix} = \begin{bmatrix} f_1(u_{pc}, v_{pc}) \\ f_2(u_{pc}, v_{pc}) \end{bmatrix} \quad (4.1)$$

$$\begin{aligned} f_i(u_{pc}, v_{pc}) = & a_{i,1} + a_{i,2}u_{pc} + a_{i,3}v_{pc} \\ & + a_{i,4}u_{pc}^2 + a_{i,5}v_{pc}^2 + a_{i,6}u_{pc}v_{pc} \end{aligned} \quad (4.2)$$

La méthode basée interpolation présente l'avantage d'être précise, simple et rapide à mettre en œuvre, permettant ainsi l'obtention rapide de premiers résultats. Toutefois, elle n'est pas robuste aux changements de pose. Cette caractéristique peut être très ennuyeuse sur notre VST-HMD, qui ne reste jamais longtemps immobile par rapport au visage. En effet, il a tendance à glisser et se décaler de quelques millimètres. Ce décalage annule l'étalonnage de l'oculomètre et le rend inopérant. Nous avons donc choisi de tester un second algorithme reprenant la méthode proposée par [Swirski 2013] sans l'étape d'optimisation.

#### 4.1.4.2 Implémentation d'un algorithme basé modèle 3D

Le principe de la méthode proposée par [Swirski 2013] est d'estimer la pose de l'œil en se basant sur un modèle géométrique et dynamique (Figure 4.12). Sur ce modèle, la pupille est un disque de taille variable en rotation autour d'un point  $X_{cr}$ . L'angle  $\kappa$  entre l'axe pupillaire et l'axe visuel (défini section 1.1.2.2), ainsi que la

réfraction de l'image de la pupille à travers la cornée et le cristallin<sup>5</sup> ne sont pas considérés. Comme précédemment, cette méthode n'utilise que la détection de la pupille sur l'image de la caméra. Pour estimer la direction du regard, l'algorithme se base alors sur la détection de la pupille sur plusieurs images consécutives. Ainsi, à partir de plusieurs ellipses de pupille, la projection  $x_{cr}$  du centre de rotation de l'œil  $X_{cr}$  sur l'image est calculée comme l'intersection des axes pupillaires, c'est-à-dire les lignes perpendiculaires aux plans des ellipses et passant par leur centre  $X_{cp}$  (Figure 4.13). Ensuite en faisant une hypothèse sur le diamètre de la pupille  $D_p$ , l'algorithme déduit la position 3D du centre de rotation de l'œil assimilé ici à son centre optique. Puis pour chaque nouvelle image de l'œil, la position 3D du centre de la pupille est calculée connaissant le centre de rotation de l'œil et son modèle simplifié décrit par la figure 4.12. Le détail des calculs mis en jeu est détaillé dans les travaux de [Swirski 2013].

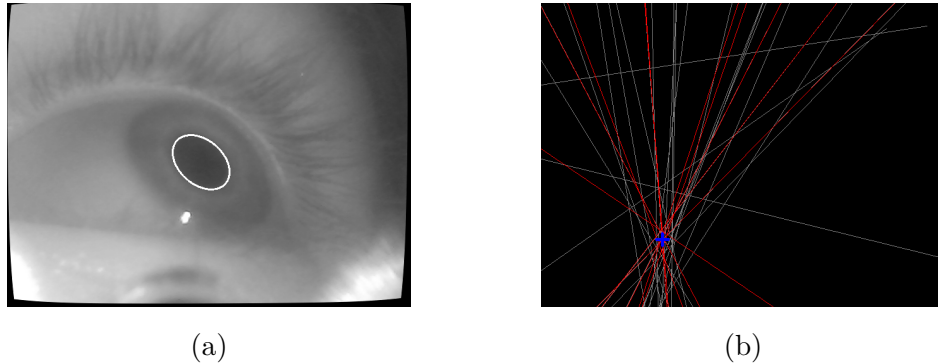


FIGURE 4.13 – Estimation de  $x_{cr}$  à partir d'ellipses détectées dans plusieurs images. (a) Détection de pupille, où l'ellipse estimée est tracée en blanc. (b) Tracé des droites perpendiculaires aux plans des ellipses et passant par leur centre  $X_{cp}$ . Les droites sélectionnées lors de l'estimation robuste par la méthode RANSAC sont représentées en rouge et l'intersection calculée est tracée en bleu.

Pour notre implémentation, nous ne considérons pas l'étape d'optimisation permettant à [Swirski 2013] d'estimer le diamètre de la pupille. Nous le déduisons de la détection de l'iris, qui est de taille fixe et partage le même plan que la pupille, ce qui permet de calculer le diamètre  $D_p$ .

Puis, pour rendre l'algorithme robuste aux changements de pose des lunettes sur le nez, le centre de rotation  $X_{cr}$  est réévalué lorsqu'un mouvement est détecté. Cette détection s'effectue en vérifiant régulièrement si la projection  $x_{cr}$  a changé.

Une seule étape d'étalonnage est nécessaire pour utiliser cet oculomètre avec un casque de réalité augmentée. Elle consiste à estimer le changement de pose entre la caméra de l'oculomètre et l'écran virtuel perçu par l'utilisateur à travers le VST-HMD. Pour calculer ce changement de pose, nous avons recours à l'algorithme du

5. Les indices de réfraction du cristallin, de la cornée et de l'air diffèrent. Les rayons optiques traversant ces milieux sont donc déviés. Certains modèles de l'œil plus complexes intègrent ce phénomène de réfraction [Guestrin 2006].

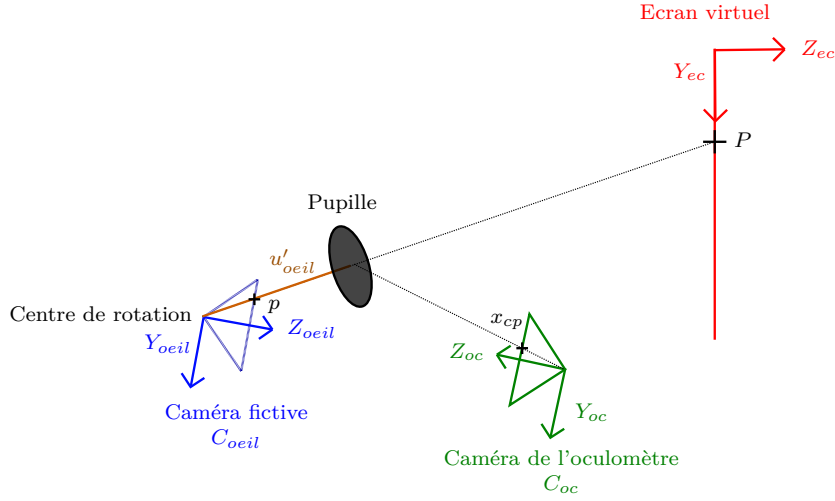


FIGURE 4.14 – Schéma illustrant la procédure d'étalonnage permettant d'estimer la transformation rigide entre  $(X_{oc}, Y_{oc}, Z_{oc})$  et  $(X_{ec}, Y_{ec}, Z_{ec})$ .

PnP appliqué sur des points 3D  $P$  de l'écran virtuel, définis dans un repère associé à cet écran, et sur des points 2D  $p$  construits sur l'image d'une caméra synthétique  $C_{oeil}$ . Cette caméra, illustrée en bleu sur la figure 4.14, admet un centre optique confondu avec le centre de rotation de l'œil. Sa focale  $f$  est définie arbitrairement, car sa valeur n'influe pas sur le résultat final. Le repère  $(X_{oeil}, Y_{oeil}, Z_{oeil})$  associé à la nouvelle caméra correspond au repère  $(X_{oc}, Y_{oc}, Z_{oc})$  de la caméra de l'oculomètre après une rotation de  $180^\circ$  autour de l'axe  $Y_{oc}$ . Les points 2D  $p$  définis sur l'image de  $C_{oeil}$  correspondent à la projection des points 3D de l'écran sur cette caméra.

La procédure d'étalonnage demande à l'utilisateur de fixer différents points répartis homogènement sur l'écran virtuel. De cette procédure, on en déduit les points 3D  $P$  connus dans le repère  $(X_{ec}, Y_{ec}, Z_{ec})$  de l'écran et les points 2D  $p$ . Ces derniers sont calculés par projection du centre de la pupille sur l'image de  $C_{oeil}$ . Les points 2D sont alors exprimés comme dans l'équation (4.3), où  $u'_{oeil}$  de composantes  $(u'_{oeil,x}, u'_{oeil,y}, u'_{oeil,z})$  représente le vecteur direction de l'œil (c'est-à-dire l'axe de visée de l'œil) dans le repère de  $C_{oeil}$ . On détermine  $u'_{oeil}$  à partir de la pose de l'œil fournie par l'oculomètre en appliquant à  $u_{oeil}$  (vecteur direction de l'œil dans le repère de l'oculomètre) le changement de pose entre la caméra de l'oculomètre et la caméra  $C_{oeil}$ .

$$p = \begin{bmatrix} u'_{oeil,x}/u'_{oeil,z} * f \\ u'_{oeil,y}/u'_{oeil,z} * f \end{bmatrix} \quad (4.3)$$

Cette solution d'oculométrie a l'avantage d'être robuste aux changements de luminosité et de pose des lunettes sur le nez. De plus, son étalonnage au sein d'un VST-HMD ne nécessite qu'une procédure unique et simple sans banc d'étalonnage. Toutefois, cette méthode et son intégration au sein du VST-HMD présentent quelques désavantages, qui sont détaillés plus tard section 4.1.6.

#### 4.1.5 Premiers tests, résultats et performances

La première étape de conception du banc de test fut de mettre en place tous les composants logiciels et matériels souhaités pour nos expériences : les perceptions visuelles et 3D de l'environnement, la localisation du banc, la cartographie de la scène, le suivi du regard du porteur et la synthèse d'images adéquates pour l'utilisateur. Par souci de temps, l'implémentation actuelle de ces composants correspond à une première version, qui pourrait encore être optimisée. La contrainte du temps de traitement est le principal point d'amélioration abordé section 4.1.5.2. Pour tester le fonctionnement de notre banc de test, une application présentée section 4.1.5.1 a été mise place et reprend tous les composants du système. Puis comme dernière évaluation exposée section 4.1.5.3, nous avons vérifié que la perception 3D du porteur à travers le casque est améliorée quand l'algorithme de DIBR rectifie l'image affichée.

##### 4.1.5.1 Test du VST-HMD par une application

Une application d'exemple a été mise en place pour tester l'ensemble des composants intégrés au banc de test et pour valider le système total en situation de fonctionnement. Le principe de cette application est d'afficher un objet synthétique sur une surface de la scène dans la direction du regard de l'utilisateur. Dans un premier temps, une surface de la scène réelle est détectée à partir d'un nuage de points 3D par la méthode d'estimation robuste RANSAC. Ce nuage de points peut être issu de la carte construite par l'algorithme de SLAM ou de la carte de profondeur calculée par des techniques de correspondance stéréo. Pour cette application, nous avons choisi la carte de profondeur afin de disposer de nombreux points répartis sur toute la surface. La recherche du plan est accompagnée de contraintes supplémentaires pour ne détecter que les plans horizontaux. Pour cela, on fait l'hypothèse que l'axe optique de la caméra est quasi-horizontale. Après détection, un repère est associé à la surface et sa pose est mémorisée. Comme le changement de pose entre la scène et le VST-HMD est mis à jour par le SLAM, la pose de la surface par rapport au VST-HMD peut être facilement déduite. L'objet synthétique est ensuite affiché sur la surface détectée. L'utilisateur interagit avec l'objet en indiquant une nouvelle position sur la surface. Pour définir cette position, il la pointe du regard et presse une touche du clavier relié à l'ordinateur. La position de ce point d'intérêt est calculée comme l'intersection entre la surface détectée dans la scène et la direction du regard, elle-même estimée par l'oculomètre.

La figure 4.15 montre les images stéréoscopiques affichées dans le VST-HMD pendant nos tests. Ici, un modèle 3D de pingouin est projeté sur la surface d'un bureau détectée dans la scène.

##### 4.1.5.2 Performance temporelle actuelle sur CPU

La garantie de performance temporelle (latence et fréquence d'affichage) sur les traitements du banc de test est indispensable pour le confort de l'utilisateur.



FIGURE 4.15 – Images stéréoscopiques affichées sur l’écran du VST-HMD pour l’application d’exemple. Un modèle synthétique de pingouin est projeté sur la surface d’un bureau dans la direction du regard de l’utilisateur.

Lorsque l’image affichée présente une latence importante, le mouvement résultant perçu par le système visuel ne coïncide pas avec celui mesuré par l’oreille interne. Pour cette raison, la fréquence d’affichage et la latence sont des contraintes essentielles. La fréquence nécessaire dépend de l’application visée. Plus les mouvements de tête du porteur sont rapides, plus la fréquence d’affichage devra être élevée. Pour une simple visualisation d’environnements virtuels dans un HMD, [Chen 2007] conseille une limite minimale de 17.5Hz. En revanche, pour les applications impliquant des mouvements rapides, une fréquence élevée est obligatoire. Claypool et al. [Claypool 2006] décrivent l’impact de la fréquence d’affichage sur la perception de l’utilisateur et sur les performances de jeu. De même, la société Oculus conseille au moins 70 images par seconde (frame per seconds - fps) pour l’Oculus DK2. La latence entre le mouvement du HMD et l’affichage des images doit aussi être limitée pour ne pas être perceptible. Les auteurs de [Adelstein 2003] revendiquent qu’elle ne doit pas dépasser 17ms.

De premiers résultats encourageants de notre banc de test (présentés section 4.1.5.1) ont été obtenus. Cependant, les images générées sont acceptables pour des scénarios statiques. La fréquence d’affichage est bien trop basse pour permettre de vraies interactions, où le porteur est en mouvement dans la scène. Elle atteint actuellement environ 1 fps. De plus, la latence de l’affichage est supérieure à 1 seconde, ce qui est clairement au-dessus des attentes voulues. Lors de nos tests, l’ordinateur portable utilisé contient un processeur Intel Core i7-4800MQ de 8 coeurs tournant à 2.70GHz et une carte graphique Quadro K1100M/PCIe/SSE2.

L’étude des performances des différents composants logiciels est détaillée ci-dessous :

Confidentiel

- La durée de la rectification épipolaire peut être négligée, puisque ce processus implique seulement une interpolation bilinéaire pour des positions de pixels enregistrées dans une carte précalculée.
- Le SLAM binoculaire est utilisé sur les images de résolution réduite (540x540) avec une limite de 1000 points caractéristiques détectés et suivis. La fréquence maximum atteinte pour le SLAM seul est de 13 fps avec trois cœurs utilisés sur le CPU. Comme l’affichage se fait à 60 fps, une pose calculée par le SLAM est utilisée plusieurs fois par le module de génération d’image.
- L’étape de génération d’images est la plus chronophage. L’implémentation sur CPU du Semi Global Block Matching disponible avec la bibliothèque **OpenCV** requiert près d’une seconde pour calculer la carte de profondeur. L’étape de remplissage de trous peut elle aussi être longue, lorsque l’image reconstruite possède de nombreuses zones vides ou de grandes tailles. En comparaison, les étapes de 3D warping stéréo et d’analyse de scène (consistant en une estimation robuste de plan) nécessitent un temps négligeable. De même, la projection de contenu synthétique étant effectuée par GPU et les modèles projetés étant composés de quelques centaines de triangles texturés, cette étape atteint une fréquence d’affichage très élevée. Cependant, le temps de transfert des images générées par DIBR et des cartes de profondeur est le plus limitant pour cette étape de projection.
- L’oculomètre nécessite peu de calculs et peut s’exécuter à plus de 50 fps sur un seul cœur. Cependant, cette tâche ne peut dépasser 30 fps à cause de la fréquence d’acquisition de la caméra.

#### 4.1.5.3 Test et analyse de la perception 3D

Pour évaluer l’impact du VST-HMD sur la perception 3D de l’environnement et sur la coordination vision-manipulation, nous avons mis en place une seconde expérience. L’idée est de vérifier si l’utilisateur voit les objets 3D synthétiques à la position où ils devraient être. L’expérience se base sur le principe suivant : l’utilisateur doit être capable de pointer une cible dans la scène après avoir fermé les yeux. Il s’appuie uniquement sur le souvenir de la position de la cible perçue à travers le HMD. Pour évaluer la capacité naturelle de l’utilisateur à effectuer cette action sans casque, nous avons fait une première expérience sans HMD. Elle se déroule par les étapes suivantes : l’utilisateur est assis à un bureau, regarde un tag posé sur le bureau, ferme ses yeux et pose un second tag à la position mémorisée du premier tag. Ce processus est répété plusieurs fois pour différentes positions du tag sur la table. Les deux tags sont ensuite localisés par une caméra avec la bibliothèque AprilTags [Olson 2011] pour calculer la distance qui les sépare. Cette distance indique l’erreur de perception. Lors de l’expérience, le tag cible a été trouvé avec une racine de l’erreur quadratique moyenne (Root Mean Square Error - RMSE) de 1.84 cm.

Pour évaluer la perception 3D à travers le HMD, nous appliquons la même expérience quand l’utilisateur porte le dispositif. La procédure est la suivante : un

carré synthétique est affiché par le VST-HMD sur un plan détecté dans la scène réelle (le bureau pour notre expérience) (Figure 4.16(a)). Ici, le carré représente la cible que l'utilisateur doit mémoriser. Ensuite et comme précédemment, le porteur du casque ferme les yeux et pose un tag à la position ciblée (Figure 4.16(b)). Puis en dernier lieu, l'erreur de perception est caractérisée au moyen de la RMSE entre le carré ciblé et le tag (Figure 4.16(c)). Cette expérience est répétée deux fois. Pour la première, les images du banc stéréo sont directement affichées sur le casque, sans étape de DIBR. En revanche lors de la seconde expérience, les images affichées sont synthétisées pour la position des yeux par DIBR.

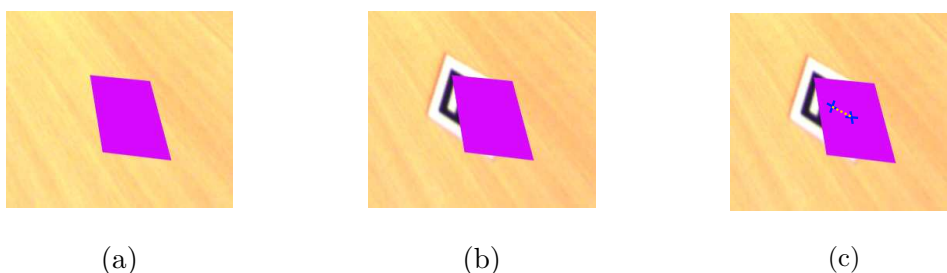


FIGURE 4.16 – Illustration des étapes de l'évaluation de la perception de la profondeur à travers le VST-HMD. (a) carré synthétique affiché par le HMD. (b) tag déposé par l'utilisateur à la position mémorisée du carré. (c) différence de position entre le tag et le carré synthétique.

Lors de l'expérience, il est important que l'utilisateur ferme les yeux quand il pose le tag. S'il voit sa main, il est susceptible de corriger automatiquement (par rétroaction) son mouvement, même si la perception 3D à travers le HMD est mauvaise.

Pour les deux dernières expériences, le porteur du casque trouve la bonne position avec une RMSE de 4.42 cm sans DIBR et de 3.13 cm avec DIBR. La procédure a été répétée 55 fois pour différentes positions de carrés et le détail des erreurs de perception est visible sur la figure 4.17. Par le biais de ces résultats, on observe bien une amélioration de la perception à travers le HMD quand le DIBR corrige l'image. Toutefois, cette perception peut encore être améliorée, car elle n'atteint pas les performances obtenues sans casque. Pour cela, plusieurs pistes sont possibles :

- ré-étalonner le casque plutôt que d'utiliser les paramètres par défaut proposés par la société Oculus (modèle de projection trou d'épingle suivi d'une compensation des distorsions) ;
- inclure un déplacement possible de l'œil (localisé par un oculomètre) dans le modèle de projection du HMD comme le propose [Itoh 2014] ;
- améliorer la qualité de l'image générée par DIBR, pour éviter de perturber l'utilisateur.

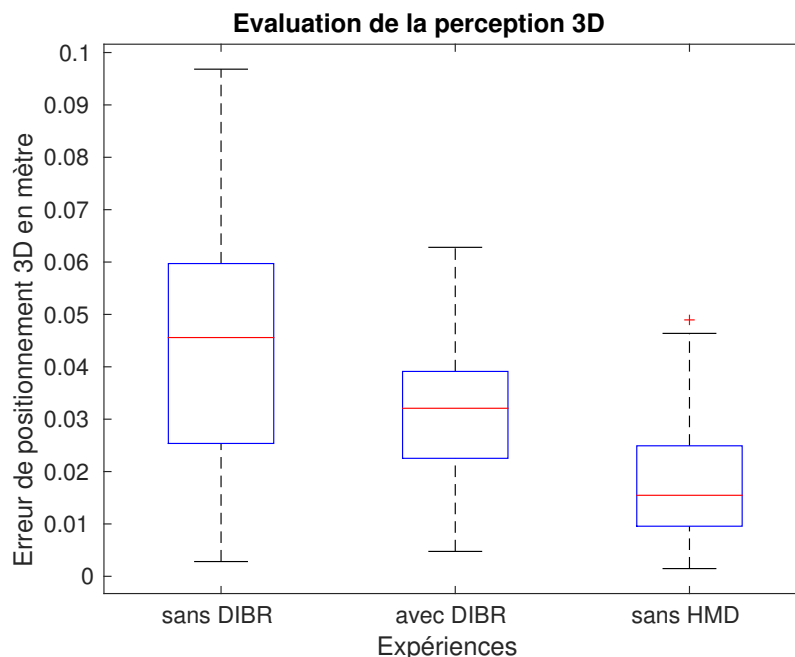


FIGURE 4.17 – Erreur de perception en mètre pour les 3 expériences : de la gauche vers la droite, résultats sans DIBR, résultats avec DIBR et résultats sans HMD.

#### 4.1.6 Perspectives

L'amélioration principale à apporter au banc de test est d'augmenter la fréquence de traitements des algorithmes et de diminuer la latence de l'image affichée dans le casque. Toutefois, de nombreux autres aspects sont à perfectionner.

**A Déport des traitements d'images sur GPU** Les algorithmes n'ont pas été entièrement parallélisés en utilisant efficacement l'architecture multicœur de l'ordinateur. Le module de génération d'image (GI), qui tourne essentiellement sur le CPU, pourrait bénéficier d'une implémentation GPU. En effet, les principaux calculs du module GI portent sur la géométrie et l'image. Une accélération de plus de 30x est souvent observée pour de tels problèmes. Par exemple, des algorithmes de correspondance stéréo temps réel, tel que celui proposé par [Atzpadin 2004], pourraient être utilisés pour drastiquement réduire le temps de calcul. Pour le DIBR, les techniques basées sur l'apprentissage profond semblent prometteuses, mais encore aucune implémentation applicable à notre configuration n'est actuellement disponible librement. À défaut, les solutions proposées par [Lai 2016b] et [Solh 2012] sont les plus raisonnables en matière de temps de traitement. Néanmoins, pour espérer obtenir des résultats appréciables, ces deux méthodes doivent être restreintes au remplissage de petites zones.

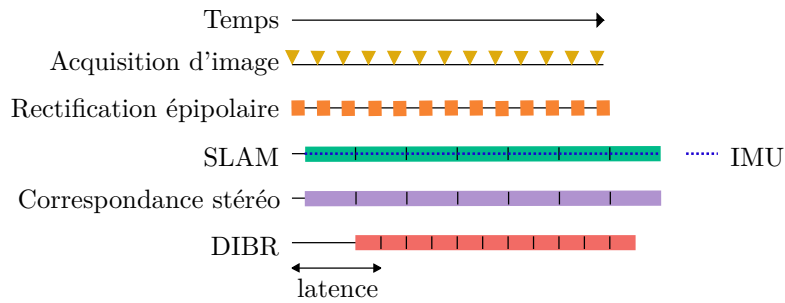


FIGURE 4.18 – Schéma illustrant les fréquences de fonctionnement des différents acquisitions et algorithmes. Les durées indiquées par la longueur des rectangles colorés ne sont pas représentatives des temps réels d'exécution des algorithmes. La latence est représentée par une flèche symbolisant le temps écoulé entre l'acquisition des images par les caméras et la synthèse d'images associées à la position des yeux de l'utilisateur. Ce temps comprend la durée des processus de rectification épipolaire, de correspondance stéréo et de DIBR.

**B Ajout d'une centrale inertielle pour améliorer la localisation du SLAM et augmenter la fréquence** Si une nouvelle image de la scène est affichée à 60 Hz dans le casque, la pose des éléments synthétiques devra être mise à jour à la même fréquence. Or, actuellement à 13 Hz, la fréquence du SLAM est encore trop faible. Lors du processus de localisation, la recherche de points caractéristiques dans l'image est l'étape la plus chronophage. C'est pourquoi pour augmenter la fréquence du SLAM, une solution intéressante serait de tenir compte des données d'une centrale inertielle, qui fonctionne à une fréquence plus élevée. Sur la base d'un modèle du mouvement du HMD, ces données supplémentaires permettraient d'estimer de nouvelles poses du casque entre deux images traitées.

**C Réduction de la latence par la synthèse d'image prédictive** L'image affichée au sein du VST-HMD possède une latence observable lorsque le casque ou la scène se déplace entre l'instant  $t$  de la capture des images par le banc stéréo et l'instant  $t'$  de l'affichage des images synthétisées dans le casque (Figure 4.18). Pour réduire cette latence, nous proposons deux solutions (Figure 4.19). La première consiste à calculer par DIBR une carte de profondeur associée aux dernières images reçues du banc stéréo. Cette solution est intéressante lorsque le processus de DIBR est plus rapide que la mise en correspondance stéréo. Elle permet alors d'éviter la latence propre au calcul de stéréovision. La seconde solution vise à générer une image prédictive par DIBR en se basant sur la pose prédite qu'aura le HMD au moment de l'affichage. La prédiction peut s'appuyer sur un modèle de mouvement du HMD et sur les dernières données fournies par le SLAM.

**D Amélioration de la qualité des images synthétisées par DIBR** Actuellement, nos images synthétisées présentent de nombreux artefacts principalement causés par les erreurs d'appariement dans la carte de profondeur et la présence de

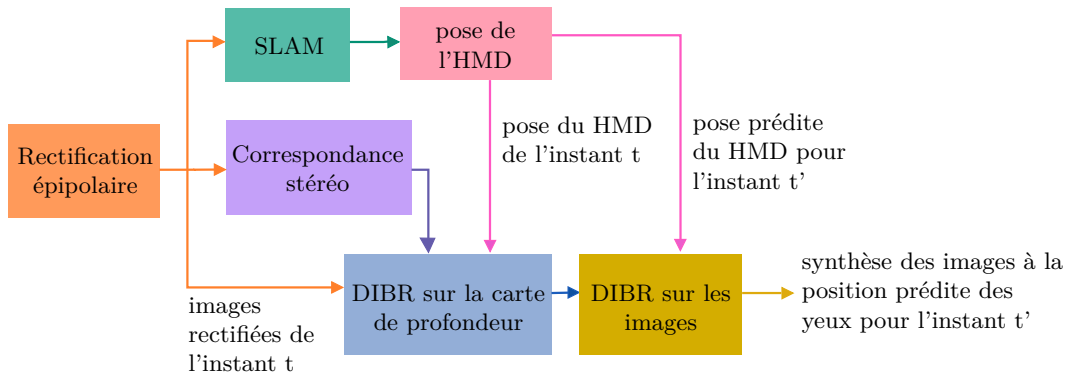


FIGURE 4.19 – Schéma-bloc décrivant le processus de synthèse d'images prédictives. Pour afficher des images stéréoscopiques à un instant  $t'$ , on effectue un algorithme de DIBR sur les dernières images acquises à l'instant  $t$ . Ce processus de DIBR considère la position prédite des yeux de l'utilisateur à l'instant  $t'$  estimée par le module de prédiction de la pose de l'HMD (en rose). Si la correspondance stéréo nécessite plus de temps que le DIBR, alors une carte de profondeur de l'instant  $t$  peut être préalablement estimée par DIBR en considérant la dernière carte calculée.

grandes zones vides après "3D warping". La taille de ces zones est directement liée à la distance des objets de la scène avec le HMD et à la distance entre la caméra réelle et l'œil de l'utilisateur. Lorsque nous avons conçu le banc stéréo du VST-HMD, nous avons considéré des critères de la robotique. Nous avons donc choisi d'éloigner les deux caméras du banc de stéréovision (en augmentant l'écart intra-oculaire) pour percevoir la profondeur sur une plus grande distance. Cette caractéristique est intéressante pour des tâches de localisation, néanmoins elle met en péril l'algorithme de DIBR. Premièrement, comme les caméras sont très éloignées l'une de l'autre, l'algorithme de stéréovision peut difficilement estimer les profondeurs des objets proches (à moins de définir une disparité maximale élevée pénalisant le temps de traitement). Deuxièmement, à cause de cette configuration, les caméras du banc de test sont éloignées des yeux de l'utilisateur. En conséquence, les zones vides des images synthétisées par "3D warping" sont plus grandes.

Pour améliorer la qualité du DIBR, il pourrait être intéressant de se servir d'un banc stéréo, dont les caméras sont plus proches des yeux de l'utilisateur. Une solution idéale serait de concevoir un système optique où une partie de la lumière perçue par les yeux est réfléchiée vers une caméra placée dans la monture. Ce système serait configuré de manière à faire coïncider au mieux le point de vue de la caméra avec celui de l'œil. Toutefois, l'œil se déplace légèrement, car les lunettes n'ont pas une position fixe sur le nez. Une étape de DIBR permettrait de corriger ce décalage résiduel.

Une autre solution pour améliorer la qualité des images synthétisées est de s'appuyer sur un modèle plus riche de la scène. Par exemple, au lieu d'utiliser un nuage de points, les surfaces des objets pourraient être reconstruites. L'étape de 3D warping tiendrait alors compte de cette reconstruction, plutôt que de considérer

directement la carte de profondeur.

### **E Adaptation des algorithmes au modèle de caméra omnidirectionnelle**

Pour le moment, tous les traitements effectués sur les images du banc stéréo fisheye considèrent des images rectifiées synthétiques générées par un modèle trou d'épingle. Néanmoins, il serait intéressant de considérer tout le champ de vision des caméras. Par exemple, l'observation d'une plus grande portion de la scène permettrait au SLAM de détecter davantage de points ou d'éléments caractéristiques, facilitant alors la procédure de localisation. Ainsi, le SLAM aurait moins de risque de traiter des images d'une portion de la scène sans aucune texture. Ce phénomène se produit lorsque la caméra a un faible angle d'ouverture et se trouve face à un objet ou un mur sans texture. Une caméra large champ de vision présente moins de risque d'avoir tout son champ de vision obstrué par un seul objet ou un seul mur. C'est pourquoi pour notre banc de test, il serait intéressant d'ajouter le modèle omnidirectionnel de Scaramuzza [Scaramuzza 2006b] à l'implémentation binoculaire de ORB-SLAM2. Des travaux ont déjà été menés dans cette direction [Urban 2016]. De même, les algorithmes du module de génération d'image pourraient tenir compte de l'ensemble du champ de vision des caméras. Pour cela, d'autres projections, comme la projection sur un cube [Vandepotael 2006], pourraient être considérées, au lieu de la projection perspective sur un plan.

### **F Ré-étalonnage automatique et régulier du banc de stéréovision**

Les caméras du banc stéréo de notre prototype sont maintenues ensemble sur un tube en fibre de carbone à l'aide de pièces imprimées en 3D. À cause de déformations résiduelles inévitables du banc stéréo, le système nécessite d'être régulièrement étalonné. Cette opération longue et fastidieuse peut être évitée si l'étalonnage se déroule en ligne. Une solution envisageable est d'intégrer au SLAM binoculaire large champ de vision l'estimation des paramètres intrinsèques et extrinsèques du banc stéréo fisheye.

### **G Amélioration de l'oculomètre basé modèle 3D**

L'oculomètre basé modèle 3D actuel a l'avantage d'être robuste aux changements de luminosité et de pose des lunettes sur le nez. De plus, son étalonnage au sein d'un VST-HMD ne nécessite qu'une procédure unique et simple sans banc d'étalonnage. Toutefois, cette méthode présente quelques désavantages. Le modèle de l'œil est très simplifié et engendre donc des erreurs importantes d'estimation de la direction de visée de l'ordre de cinq à dix degrés. La réfraction de la pupille à travers la cornée et le cristallin est négligée et l'angle  $\kappa$  (de l'ordre de 6 à 8 degrés) entre la direction du regard et l'axe pupillaire est ignoré. Une autre approximation importante est faite. Le modèle plan de l'écran ne convient pas et contribue aussi à l'erreur angulaire. Il devrait être modélisé par une surface parabolique comme le suggère [Owen 2004]. D'autre part, la robustesse aux changements de pose présente ses limites. La mise à jour du centre de rotation de l'œil n'est pas continue, car elle requiert plusieurs images

consécutives obtenues après le mouvement. Or sur notre VST-HMD, le casque se déplace très fréquemment par rapport au visage à cause de la partie en mousse en contact avec la peau qui se contracte différemment à chaque mouvement de tête ou du visage.

#### 4.1.7 Conclusion

Un premier banc de test basé sur un dispositif de VST-HMD a été mis en place. Des composants logiciels ont été intégrés en tenant compte des contraintes spécifiques aux lunettes actives, de l'état de l'art et du temps et du matériel disponible au cours de la thèse. Des premiers tests ont été réalisés et ont permis de valider l'intégration de composants matériels et logiciels au sein d'une architecture modulaire à l'instar de la robotique. De plus, une expérience a confirmé l'intérêt du DIBR pour la synthèse d'images adaptées au point de vue de l'utilisateur. Par ailleurs, ces tests ont permis d'observer les limites du système actuel énoncées dans la section 4.1.6 et de proposer plusieurs solutions d'amélioration. Des premières simulations de lunettes actives peuvent déjà être menées dans des configurations statiques où l'utilisateur ne bouge pas dans la scène.

## 4.2 Banc de test basé sur un OST-HMD

Un second banc de test a été mis en place afin de simuler une problématique ne pouvant être testée sur un VST-HMD. Il s'agit de la problématique de superposition d'éléments synthétique sur des éléments de la scène, lorsque l'utilisateur perçoit la scène directement et non à travers un écran comme pour le VST-HMD. Pour mener à bien cette tâche, il est nécessaire de connaître trois informations : la pose ciblée de l'objet synthétique dans le référentiel du HMD, la pose de l'œil par rapport au HMD et le modèle de projection de l'écran du HMD. Dans certains cas, le problème est simplifié en considérant l'œil fixe par rapport au casque. Parmi ces informations, l'une d'entre elles ne présente pas le même impact sur un VST-HMD ou un OST-HMD. Il s'agit du modèle de projection du casque. En effet, sur un VST-HMD, la scène et les éléments synthétiques sont projetés sur l'écran en suivant le même modèle de caméra. En conséquence, les éléments synthétiques sont positionnés à l'emplacement ciblé dans la scène. Le dispositif présenté section 4.1 suit ce principe. Sur un OST-HMD, l'utilisateur voit directement la scène à travers les verres. Or, le modèle de projection, associé à la perception de la scène par l'utilisateur, n'est pas connu. Il doit être estimé lors d'une procédure d'étalonnage. De même, le modèle de projection du contenu synthétique (différent du premier), perçu par l'utilisateur, doit être étalonné. Toutefois malgré ces étalonnages, la superposition obtenue n'égale pas celle sur VST-HMD. Comme en témoigne l'état de l'art sur l'étalonnage d'OST-HMD (Section 2.6), la perception de l'utilisateur à travers les lunettes est difficilement modélisable. Cette difficulté est principalement due aux systèmes optiques complexes dédiés à l'affichage et aux déplacements de l'œil par rapport au casque. Ainsi, contrairement aux VST-HMD, la superposition

d'éléments synthétiques sur la scène réelle est une tâche ardue et ne se base pas sur les mêmes techniques d'étalonnage. Cette problématique peut être transposée aux lunettes actives (d'obscurcissement ou de focalisation variable) où les propriétés du verre sont modifiables zone par zone. Par exemple, sur des lunettes d'obscurcissement variable, il faudra teinter la bonne zone du verre, afin d'obscurcir la portion de la scène ciblée telle qu'elle est perçue par l'utilisateur. Les lunettes actives ont donc besoin d'une procédure d'étalonnage comme les OST-HMD. Cependant, le banc de test basé sur un VST-HMD ne peut pas être utilisé pour tester des algorithmes d'étalonnage. Il est donc nécessaire de considérer un autre banc de test basé sur un OST-HMD pour concevoir et évaluer différents algorithmes.

Cette section présente notre banc de test basé sur un OST-HMD (Section 4.2.1) ainsi que des premières implémentations de méthodes d'étalonnage (Section 4.2.2). Ensuite, la section 4.2.3 détaille les difficultés rencontrées avec ce dispositif et les perspectives et améliorations qui pourraient être envisagées. L'ensemble des travaux menés sur ce banc de test ont été effectués avec Tristan Klempka lors de son stage de fin d'études et d'un contrat d'ingénieur.

#### 4.2.1 Dispositif et composants logiciels

Pour ce banc de test, nous avons besoin des composants logiciels nécessaires à la procédure d'étalonnage : la perception de la scène et l'estimation de la position de l'œil. Pour notre application, la perception de la scène permet d'acquérir la position de points 3D dans le référentiel du HMD. La position de l'œil, en revanche, est nécessaire pour certaines méthodes d'étalonnage [Itoh 2014]. Elle est alors intégrée au modèle de projection du casque, pour tenir compte des déplacements constants de l'œil. Les contraintes définies pour ce banc de test sont les mêmes que sur le VST-HMD. Nous avons donc conservé le choix d'un banc stéréoscopique, pour acquérir des informations sur la géométrie de la scène. Pour les mêmes raisons, nous avons repris la solution d'oculométrie développée pour le VST-HMD.

##### 4.2.1.1 Choix des composants matériels

Le dispositif conçu intègre les lunettes de réalité augmentée Lumus DK50 de la société Lumus Optical. Ces lunettes sont équipées d'un système d'affichage de 1280x720 pixels couvrant environ 40 degrés, d'un banc stéréoscopique filmant la scène et d'une centrale inertielle. Le Lumus DK50 est l'un des OST-HMD du marché proposant un affichage avec le plus grand angle d'ouverture. Il est un appareil nomade basé sur le système Android. Pour assurer son autonomie énergétique, il dispose de capacités de calcul limitées, ce qui nous oblige à déporter une partie des calculs sur un ordinateur fixe.

Au départ, cette plateforme ne dispose pas d'oculomètre. Nous l'avons donc ajouté nous-mêmes. Le système choisi est constitué d'une caméra et de deux LED infrarouges. Pour son intégration, un assemblage optique similaire à celui du VST-HMD a été considéré. Cet assemblage permet de pallier le manque de place entre le

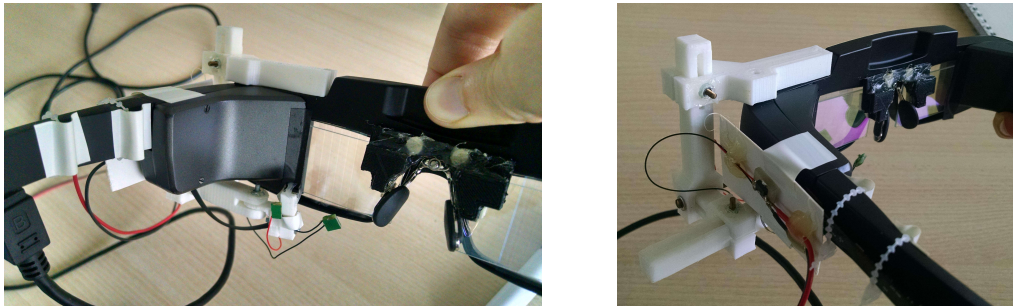


FIGURE 4.20 – Deux photos du montage de l’oculomètre au sein du OST-HMD. La caméra est orientée vers le miroir réfléchissant l’infrarouge. La position de la caméra est ajustable suivant trois axes et son orientation suivant un axe.

visage et les verres du Lumus. Comme la caméra ne dispose pas d’un grand champ de vision, il est difficile de la placer de façon qu’elle observe l’ensemble de l’œil. Nous l’avons donc fixée sur un montage mécanique ajustable, comprenant trois glissières et deux pivots (Figure 4.20). Avec ce montage, le placement de la caméra peut être réglé pour chaque individu.

Des verres électro-chromes ont été ajoutés au banc de test par la suite. Ils nous permettront de tester une application d’obscurcissement des verres, qui tient compte de la direction du regard et de la luminosité de la scène. Le principe sera d’obscurcir les verres quand l’utilisateur regarde en direction d’une source lumineuse éblouissante.

Le schéma 4.21 reprend en détail les différents matériels impliqués dans le banc de test. Les numéros du schéma sont expliqués ci-dessous :

- (1) LUMUS DK50. Prototype d’OST-HMD.
- (2) Dispositif d’affichage transparent délivré par un OST-HMD. C’est au travers de ces lunettes que l’utilisateur observe les objets virtuels et la scène réelle.
- (3) Caméras de scène. Les deux forment un banc stéréoscopique permettant des mesures 3D métriques dans la scène.
- (4) Caméra de l’oculomètre. Elle comprend un filtre laissant passer uniquement la lumière infrarouge.
- (5) Miroir semi-réfléchissant. Il laisse passer la lumière du visible, mais réfléchit la lumière dans le spectre infrarouge.
- (6) LEDs infrarouges. Elles sont dirigées vers le miroir qui réfléchit la lumière vers l’œil de l’utilisateur.
- (7) Liaison USB utilisée pour le versement des applications et de données sur l’appareil, pour le débogage des applications ou la transmission des informations de l’oculomètre.
- (8) Souris Bluetooth servant de pointeur 2D pour les applications.

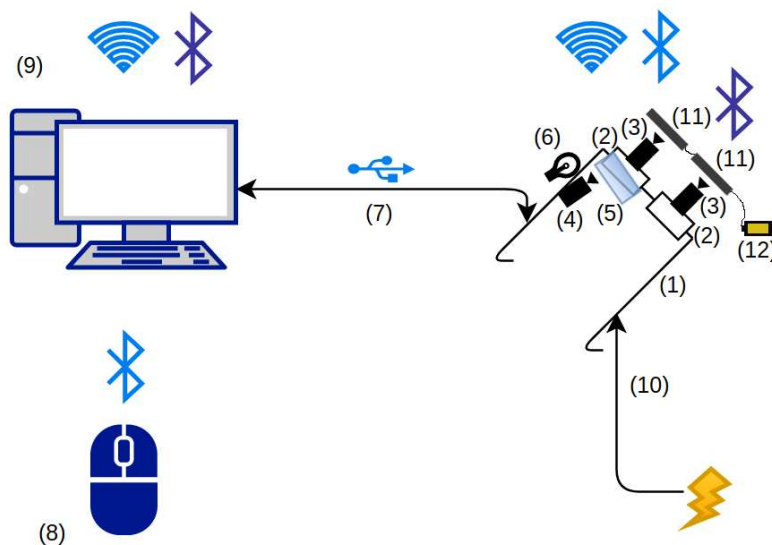


FIGURE 4.21 – Détail du matériel utilisé pour le banc de test : lunettes (1) équipées d'un banc stéréo (3) ; oculomètre (4,5 et 6) ; verres électro-chromes (11). Une partie des calculs est déportée sur un ordinateur (9) et les données sont échangées par wifi. Une souris d'ordinateur (8) connectée par Bluetooth à la lunette sert de pointeur pour les procédures d'étalonnage.

- (9) Ordinateur. Relié en Wifi aux lunettes, il permet de déporter certains traitements et/ou de communiquer avec les lunettes.
- (10) Alimentation
- (11) Verres électro-chromes ajoutés devant les lunettes et communiquant par Bluetooth basse consommation (Bluetooth low energy - BLE) avec l'ordinateur.
- (12) Alimentation par batterie des verres électro-chromes.

#### 4.2.1.2 Architecture logicielle

Parmi les objectifs ciblés du banc de test figurent sa modularité et sa simplicité d'utilisation. Comme pour le VST-HMD, les composants logiciels sont organisés dans une architecture modulaire. Pour mettre en place cette architecture, le middleware ROS est installé sur un ordinateur, sur lequel s'exécute l'ensemble des algorithmes, et sur la plateforme Android intégrée aux lunettes Lumus. Les composants ROS déployés sur la plateforme Android sont clients des composants ROS installés sur l'ordinateur et offrent des services tels que la commande d'affichage et publient des données telles que les images acquises par le banc stéréo. Grâce à ROS, les modules développés fournissent une abstraction du matériel (capteurs, afficheur et électro-chromes) et proposent des fonctionnalités couramment utilisées (suivi du regard, détection de tag dans la scène et étalonnage de l'OST-HMD).

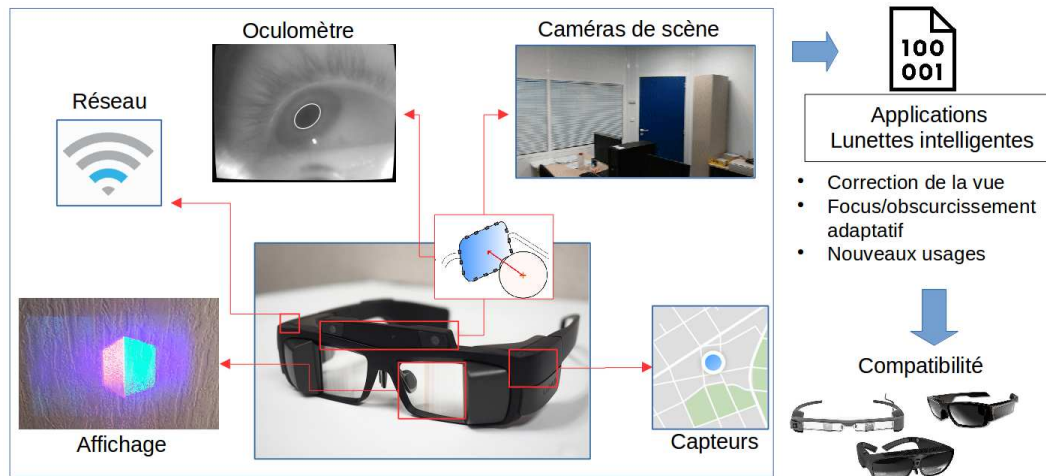


FIGURE 4.22 – Architecture modulaire du banc de test basé sur un OST-HMD. L’acquisition des données de chaque capteur (caméras, IMU et GPS) est intégrée dans des modules. Chaque algorithme est aussi inclus dans un module (oculométrie et détection de tags). La commande des actionneurs (affichage et obscurcissement) est gérée dans différents modules. Un dernier module s’occupe de la communication Wifi avec le PC pour déporter certains des calculs.

Grâce à cette architecture, le développement d’application est simplifié. Les modules développés apportent une abstraction des composants matériels et logiciels, permettant au développeur de programmer sur la base de cette abstraction de plus haut niveau. D’autre part, les modules développés sur le VST-HMD peuvent être facilement intégrés dans l’architecture modulaire de l’OST-HMD. On a donc une compatibilité des outils sur différentes plateformes. La mise en place du middleware ROS pour l’OST-HMD, qui fut un travail important, peut être appliquée sur toute autre plateforme basée sur un système Android. Le schéma 4.22 illustre l’architecture modulaire du banc de test basé sur l’OST-HMD.

#### 4.2.2 Tests et résultats d’étalonnage d’OST-HMD

Trois méthodes d’étalonnage différentes ont été proposées et testées par Tristan Klempka lors de son stage. Ces méthodes s’appuient sur le modèle de projection trou d’épingle traditionnellement utilisé pour les caméras. Ici, la caméra n’est pas réelle, mais désigne une caméra virtuelle modélisant un système visuel étendu constitué de l’OST-HMD (son écran et ses optiques) et de l’œil. Chaque écran de l’OST-HMD s’étalonne individuellement. Le choix de ce modèle a l’avantage de simplifier la procédure d’affichage de contenu synthétique. En effet, ce modèle peut directement être réutilisé par la bibliothèque OpenGL, pour synthétiser les images affichées par le casque et perçues par l’utilisateur. Les caméras virtuelles définies dans OpenGL tiennent alors compte des paramètres estimés du modèle de projection trou d’épingle.

Pour cette étude, on suppose que ce système optique respecte les conditions de Gauss : les rayons lumineux participant à la formation de l'image de la caméra ont une incidence faible par rapport à l'axe optique. Ce modèle ne prend pas en compte les distorsions géométriques et chromatiques ainsi que le flou causés par la lentille.

#### 4.2.2.1 Méthode d'étalonnage manuelle

La première méthode est très simple et consiste à ajuster manuellement les paramètres du modèle trou d'épingle. Ce modèle décrit section 2.2.1.1 comprend des paramètres intrinsèques et extrinsèques. Les paramètres intrinsèques incluent la focale suivant les axes  $X$  et  $Y$ , la position sur l'image du point principal en pixel et le paramètre de non-orthogonalité entre les axes  $X$  et  $Y$ . Les paramètres extrinsèques correspondent au changement de pose (rotation et translation) entre le repère de la caméra de scène et le repère de la caméra virtuelle. En plus des paramètres extrinsèques et intrinsèques, une dernière information est nécessaire pour projeter un point 3D de la scène sur la caméra virtuelle. Il s'agit du changement de repère entre la scène et la caméra de l'OST-HMD. Lors de nos tests, cette transformation rigide est obtenue à l'aide d'un algorithme de détection de tags [Olson 2011]. Au cours de l'expérience, l'ensemble des paramètres sont modifiables par l'utilisateur hormis la rotation (entre la caméra de scène et la caméra virtuelle), que l'on considère comme nulle pour simplifier la procédure. Cette simplification équivaut à faire l'hypothèse que les caméras réelle et virtuelle ont des directions parallèles.

Pour régler les paramètres, l'utilisateur porte le casque et les modifie de manière à ce qu'un objet virtuel se superpose au mieux avec un objet réel (Figure 4.23). Il répète cette étape pour différentes positions de l'objet réel définies dans un espace de 40x40x80 cm devant l'utilisateur. La procédure totale est limitée à quelques minutes afin de se rapprocher des conditions réelles d'utilisation.

#### 4.2.2.2 Single Point Active Alignment Method

La méthode basée sur l'alignement de points appelée "Single Point Active Alignment Method" (ou SPAAM) est une des approches les plus utilisées pour l'étalonnage d'OST-HMD. C'est souvent cette méthode qui sert d'étalon lorsqu'une nouvelle méthode d'étalonnage est proposée. Le SPAAM, présenté section 2.6.2.1, consiste à estimer les valeurs d'une matrice 3x4 correspondant au modèle trou d'épingle. Cette matrice est égale à la multiplication des matrices de paramètres intrinsèques et extrinsèques mentionnés précédemment. Pour l'estimer, nous devons recueillir une série de points 3D de la scène réelle et de points 2D sur l'écran. Ces points 2D représentent la projection des points 3D sur la caméra synthétique. Il s'agit des points 2D de l'écran qui se superposent aux points 3D réels dans le champ de vision de l'utilisateur.

Lors de nos tests, nous avons acquis des séries de dix points. Pour ce faire et comme précédemment, un objet réel est visible par l'utilisateur dans un espace de

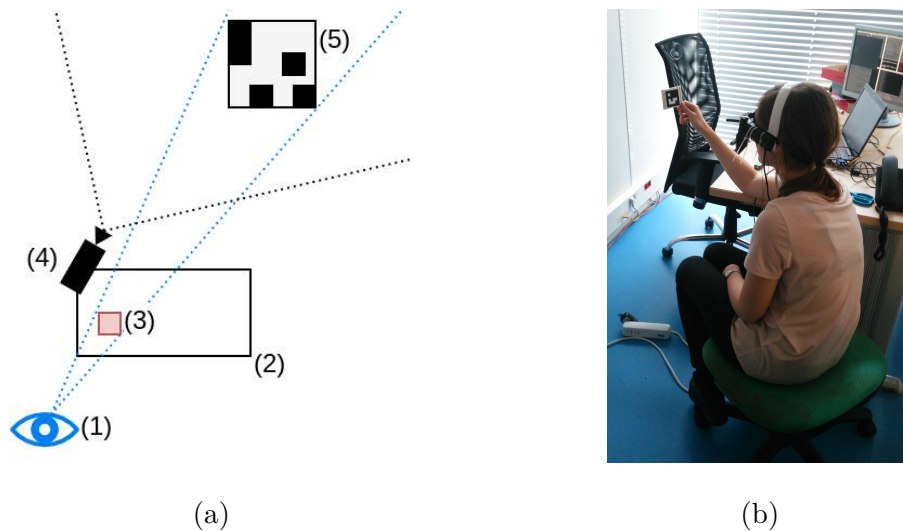


FIGURE 4.23 – (a) Schéma illustrant le principe de superposition lors de l'étalonnage. L'œil (1) observe à travers les verres du OST-HMD (2) l'objet synthétique (3) qu'il superpose à l'objet réel (5). La caméra de scène (4) filme l'objet (5) et un algorithme le localise dans le référentiel de la caméra. (b) Aperçu de la procédure d'étalonnage où le sujet, équipé du casque, regarde l'élément synthétique superposé avec le tag, défini comme objet réel.

40x40x80 cm face à lui. L'objet réel est un tag détecté sur l'image d'une caméra de scène. À l'aide du curseur de la souris Bluetooth, l'utilisateur pointe le centre de l'objet tel qu'il le voit à travers le casque. La position pointée est un point 2D sur l'écran, superposé au point 3D du point de vue du porteur de lunettes.

#### 4.2.2.3 Single Point Active Alignment Method et oculométrie

Cette dernière méthode se base aussi sur le principe du SPAAM. Toutefois, au lieu de pointer la position du point 2D à l'aide du curseur de la souris, l'utilisateur doit fixer du regard le centre de l'objet. Le point observé dans le plan de l'afficheur est donc estimé à l'aide de l'oculomètre. L'objectif de cette technique est de diminuer le nombre de matériels impliqués dans la procédure d'étalonnage et de libérer les mains de l'utilisateur. Pour cette application, l'algorithme d'oculométrie considéré est celui basé interpolation précédemment développé pour le VST-HMD (Section 4.1). Comme nous utilisons un oculomètre, une première étape d'étalonnage est requise, durant laquelle le sujet porte son regard sur les points d'une grille 3x3 couvrant l'ensemble de l'écran.

#### 4.2.2.4 Évaluations et résultats

Afin d'évaluer ces méthodes d'étalonnage, nous avons réalisé avec l'utilisateur une série de mesures sur le prototype. Pour effectuer chaque mesure, l'utilisateur observe un objet réel à travers les lunettes et pointe à l'aide du curseur de la souris

Bluetooth le centre de l'objet dans le plan image de l'afficheur. Ensuite, la position du curseur est comparée avec la position estimée par le modèle de projection. La distance entre ces deux points 2D nous renseigne sur la qualité de l'étalonnage. Pour comparer les méthodes entre elles, nous avons réalisé pour chaque méthode 5 étalonnages suivis chacun d'une évaluation.

Les erreurs moyennes de projection (sur les 5 tests) sont de 26, 7.5 et 99.3 pixels respectivement pour les méthodes manuelle, SPAAM et SPAAM avec oculométrie. Nous avons pu constater qu'une erreur moyenne de plusieurs dizaines de pixels ne permet pas une utilisation correcte des lunettes. L'utilisateur ne perçoit pas l'objet virtuel correctement superposé par rapport à l'objet réel. Pour la méthode manuelle, cette erreur est liée à la difficulté pour le sujet de régler les paramètres du modèle. Il y a de nombreux paramètres à ajuster et certains ont des actions concurrentes (par exemple les paramètres de focale et de translation le long de l'axe  $Z$ ). De plus, cette méthode est très contraignante pour le porteur du casque. Elle demande un savoir technique et beaucoup de temps pour les réglages. La seconde méthode est la plus performante avec une erreur moyenne de moins de 1% par rapport aux dimensions de l'écran. La dernière méthode, qui vise à simplifier la procédure de sélection de points sur l'écran, possède l'erreur moyenne la plus importante. Cette erreur est due principalement aux données erronées fournies par l'oculomètre. En effet, si une des données délivrées par l'oculomètre est fautive, alors l'estimation des paramètres du modèle par la méthode SPAAM est mauvaise. Or, nous avons pu constater que l'oculomètre intégré dans le Lumus ne parvient pas à estimer la direction du regard pour toutes les orientations de l'œil. Cela s'explique par la difficulté à bien placer la caméra. Comme le champ de vision de la caméra est réduit, l'œil n'est observable que dans un espace limité. Cet inconvénient contraint le placement de l'œil par rapport aux lunettes. Or, ce placement est également conditionné par l'observation de l'écran virtuel du Lumus. L'œil ne peut percevoir l'écran que depuis une zone restreinte. En conséquence, il est difficile de trouver un positionnement du HMD adéquat, qui concilie à la fois la contrainte liée à l'écran du Lumus et celle liée à la caméra de l'oculomètre. Ce problème fausse l'étalonnage de l'oculomètre et par la suite de l'OST-HMD.

### 4.2.3 Limites du système actuel et perspectives

Le banc de test possède quelques limites qui nécessiteraient d'être surmontées.

#### 4.2.3.1 Oculométrie

Depuis les tests d'étalonnage présentés section 4.2.2, l'oculomètre a été modifié pour améliorer le point de vue de la caméra, afin qu'elle observe l'œil en entier. Comme expliqué section 4.2.1, l'assemblage mis en place sur le VST-HMD a été copié. Cette solution permet d'éloigner la caméra en la plaçant devant un miroir, qui réfléchit une image de l'œil. Néanmoins, l'espace disponible au sein de l'OST-HMD est plus contraint que dans le VST-HMD. En conséquence, il est difficile de

trouver une position adéquate pour la caméra de manière à concilier deux critères : l'observation de l'ensemble de l'œil et le confort de l'utilisateur. D'autre part et pour les mêmes raisons, son positionnement doit être réajusté pour chaque individu. Une solution possible et simple, pour éviter tous ces problèmes, serait de changer de caméra pour une nouvelle proposant un plus grand angle d'ouverture. Pour des raisons de temps et de difficulté à se fournir des caméras d'aussi petit gabarit, nous n'avons pas remplacé la caméra.

Une solution supplémentaire pour endiguer les données aberrantes serait d'ajouter une étape de filtrage temporel à l'oculomètre, ce qui limiterait les mauvaises détections.

#### 4.2.3.2 Étalonnage

La procédure d'étalonnage, telle qu'elle est présentée section 4.2.2, requiert quelques améliorations. Pour mesurer l'erreur, nous avons demandé à l'utilisateur de cibler un point sur un écran avec un curseur de souris. Il serait intéressant d'évaluer l'impact de l'incertitude liée à ce procédé. Pour cela, le sujet devra pointer à la souris différents points affichés par le casque. Ensuite, l'erreur due à la sélection par un curseur correspond à la distance entre le point 2D affiché à l'écran et le point sélectionné par le curseur. Une seconde partie de la procédure d'évaluation serait à modifier. Elle concerne le point 2D sélectionné lors des tests. Ce dernier est défini comme le centre d'un carré synthétique affiché par le casque. Cependant, choisir le centre de l'objet est discutable, car la projection du centre d'un objet n'est pas le centre de sa projection. Afin de mieux évaluer la qualité de l'étalonnage, on pourrait utiliser des primitives permettant un repérage ponctuel précis, tel que l'intersection de paires de droites.

La méthode choisie pour l'étalonnage a l'avantage d'être performante, simple et rapide à mettre en oeuvre (peu de lignes de code et pas de banc d'étalonnage requis). Toutefois, elle n'est pas robuste aux changements de pose du casque sur le visage du porteur. Certaines méthodes proposent de tenir compte de la position de l'œil en temps réel pour adapter le modèle de projection du système d'affichage ([Itoh 2014] et [Klemm 2017]). De plus, le SPAAM considère un modèle trou d'épingle simple sans distorsion. Or, il a été prouvé que les assemblages optiques intégrés dans les dispositifs d'affichage génèrent des distorsions ([Owen 2004] et [Klemm 2017]). Ainsi, pour obtenir plus de précision, il faudrait considérer des modèles avec distorsions [Itoh 2015b] ou des modèles non paramétriques [Klemm 2017].

#### 4.2.3.3 Applications sur le banc de test

Plusieurs applications sont envisagées sur le banc de test. La première est de tester le système d'oculomètre basse consommation présenté chapitre 3 dans un cadre de réalité augmentée. Un prototype d'oculomètre serait alors conçu et intégré sur l'OST-HMD. Ensuite, plusieurs cas d'utilisation pourraient être testés et évalués, par exemple la sélection de commandes par le regard ou le clignotement des yeux.

Une autre application prévue repose sur la commande de verres électro-chrome. Ici, l'objectif serait d'obscurcir les verres lorsque le porteur des lunettes regarde en direction d'une source lumineuse intense.

### 4.3 Prototypage de lunettes avec obscurcissement automatique des verres

Au cours de la thèse, un concept de lunettes actives basse consommation a été proposé par Essilor. L'objectif principal de ce système est de contrôler la teinte de verres solaires automatiquement suivant la luminosité environnante. En plus de cet objectif, le concept proposé satisfait les conditions de faible encombrement et de basse consommation énergétique. En conséquence, les composants choisis sont discrets et intégrables dans une monture et l'autonomie énergétique visée est d'au moins une journée. Pour tester et évaluer cette idée, des simulations ont été menées sur le banc de test basé VST-HMD. Puis, dans un deuxième temps, un prototype a été conçu.

Après expérimentation, ce concept a donné lieu au dépôt d'un brevet<sup>6</sup>.

#### 4.3.1 Choix des capteurs de luminosité

Pour mesurer la luminosité, il est possible d'utiliser un ALS<sup>7</sup> (ambient light sensor) avec un large champ de vision. L'inconvénient est que ce composant ne détecte pas systématiquement les sources éblouissantes. Par exemple un ALS ne convient pas si l'environnement est globalement sombre avec une source lumineuse ponctuelle. Comme il réalise une moyenne sur l'ensemble de son champ de vision, la luminosité perçue ne sera pas élevée et ne signalera pas la présence d'une source éblouissante.

Réciproquement, un ALS avec un faible angle d'ouverture n'est pas une solution suffisante. En effet, si le capteur est positionné de manière à observer la partie centrale de la vision de l'utilisateur, il ne pourra pas signaler la présence de sources lumineuses intenses en périphérie du champ de vision. En conséquence, un ALS seul ne convient pas pour notre besoin. Notre application requiert de connaître l'intensité lumineuse suivant plusieurs directions.

Une solution possible serait d'utiliser une caméra. En traitant les images acquises par la caméra, nous pourrions détecter et localiser des sources lumineuses ponctuelles. Cependant, les caméras consomment beaucoup d'énergie, ce qui nuit à l'autonomie énergétique du dispositif. D'autre part, ces capteurs sont plus encombrants que les ALS et leur dynamique est moins bonne.

---

6. Combier, Jessica, Bouchier, Aude, Cano, Jean-Paul, Danès, Patrick et Vandeportaele, Bertrand *Method and system for determining an environment map of a wearer of an active head mounted device*. **brevet déposé, numéro EP17305282.0**. 2017.

7. Un ALS est un composant électronique équipé d'une photodiode, d'un filtre photopique et d'un convertisseur analogique-numérique, qui mesure l'intensité lumineuse ambiante telle qu'elle est perçue par l'homme.

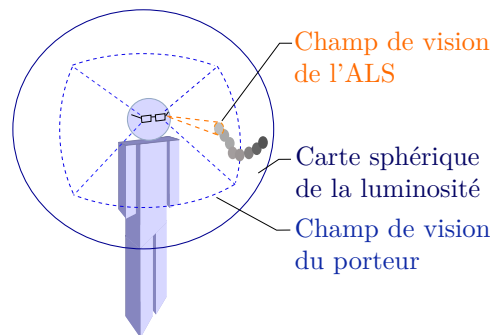


FIGURE 4.24 – Schéma illustrant le principe de cartographie sphérique basé sur l'acquisition d'une donnée de luminosité sur un faible angle d'ouverture et pour une direction de visée connue.

Pour satisfaire le besoin et les contraintes énoncés, nous proposons de construire dynamiquement un modèle de l'environnement lumineux en faisant l'hypothèse que l'environnement évolue lentement et que les sources lumineuses sont loin relativement aux déplacements des lunettes. Afin d'acquérir les données de luminosité, les lunettes sont équipées d'un ou plusieurs ALS de faible ouverture (10 à 20 degrés) couplés à une centrale inertielle (inertial measurement unit - IMU) composée d'accéléromètres, de gyromètres et de compas. À l'aide d'un filtre de Kalman appliqué aux données de l'IMU, on estime les changements de direction des lunettes. Grâce à cette information, on peut définir une direction de visée de l'ALS pour chaque observation par rapport à un repère d'origine. Ainsi, au fur et à mesure que le porteur des lunettes tourne la tête, l'ALS observe différentes portions de la scène. En enregistrant chacune des mesures, il est possible de construire une carte sphérique centrée sur le sujet. Ce principe est illustré sur le schéma 4.24. Par la suite, cette carte peut être analysée pour identifier des sources lumineuses ponctuelles. Pour cela, on calcule l'intersection entre le champ de vision du porteur et la carte sphérique. On en déduit la luminosité maximale observable par le porteur suivant la position angulaire courante de sa tête.

L'usage d'une carte sphérique de luminosité est une approximation de l'environnement lumineux du porteur. Elle représente une bonne approximation si la source lumineuse est très éloignée (par exemple des nuages ou le soleil). Pour réduire la présence de fausses données de luminosité dans la carte causée par cette simplification, trois solutions peuvent être envisagées. La première consiste à dater les données de luminosité sauvegardées et à les supprimer quand elles sont trop anciennes. La seconde est de ne sauvegarder les données que dans la demi-sphère supérieure par rapport à l'horizontale. Nous faisons l'hypothèse que les sources ponctuelles gênantes se trouvent en hauteur. On évite alors les données de luminosité liées à des objets de la scène près du sol. La troisième solution est de détecter si l'utilisateur est en intérieur ou en extérieur à l'aide de capteur ultraviolet. Lorsqu'il est en intérieur, l'analyse de la luminosité est arrêtée, les lunettes sont mises à l'état clair et la carte est initialisée. Ainsi, quand il retourne en extérieur, le processus

d'analyse recommence sur une carte vierge exempte de données de luminosité liées à l'intérieur.

### 4.3.2 Choix d'une commande du verre en fonction de la luminosité

La problématique suivante est de déterminer quelle commande appliquer au verre en fonction de la luminosité perçue. Une première approche développée par Essilor fut de définir des paliers pour la commande. Suivant la luminosité, la commande prenait cinq valeurs différentes correspondant aux classes des verres solaires. On peut représenter la fonction  $f(\text{luminosité}) = \lambda$  (avec  $\lambda$  = taux d'absorption du verre) comme une fonction discontinue, constante par morceaux. Cette approche présente l'inconvénient de ne pas proposer d'état d'obscurcissement intermédiaire. De plus, si la luminosité varie légèrement autour d'une des discontinuités de  $f$ , le verre risque de clignoter entre les deux valeurs des deux paliers. Pour contrer ce problème, une approche de commande avec hystérésis avait été mise en place. Ainsi la valeur de luminosité déclenchant le passage au palier supérieur est plus élevée que celle définie pour le passage du palier haut au palier bas.

Pour notre prototype, nous avons choisi une nouvelle méthode générant une commande continue. Nous nous sommes également fixé d'autres objectifs : la solution doit être facile à mettre en œuvre, avoir un faible coût calculatoire, et proposer des réglages intuitifs. Pour répondre à ces objectifs, nous avons choisi d'utiliser la logique floue [Bouchon-Meunier 2007]. La logique floue est un outil permettant de modéliser des raisonnements basés sur des données imprécises et incertaines. Par comparaison à la logique booléenne qui s'appuie sur des ensembles classiques, la logique floue repose sur des variables définies sur des ensembles flous, traduisant le caractère incertain de leurs valeurs. Grâce à cette notion d'imprécision et d'incertain, cette technique est intéressante pour modéliser le ressenti humain. Elle est souvent employée pour définir des variables jouant sur le ressenti. Un exemple courant est l'asservissement en température d'une pièce de vie. Dans notre cas, la variable à estimer est la commande d'obscurcissement des verres. Elle est, elle aussi, directement liée au ressenti de la personne. Nous ne détaillerons pas ici la théorie liée à la logique floue. Les lecteurs intéressés pourront se tourner vers le livre [Bouchon-Meunier 2007]. Pour notre implémentation, nous avons choisi de définir les fonctions d'appartenance<sup>8</sup> par des trapèzes pour les variables d'entrée et par des singletons pour les variables de sortie. Ce choix est un bon compromis entre complexité et coût calculatoire. Pour les opérateurs et l'implication flous nous avons choisi respectivement les opérateurs de Zadeh MIN/MAX [Zadeh 1965] et l'implication Mamdani [Mamdani 1974]. Le processus de défuzzification est basée sur le calcul du centre de gravité.

Le schéma 4.25 présente l'architecture logicielle du prototype. En entrée du

---

8. En logique floue, les variables d'entrée suivent un processus de fuzzification, où elles sont définies sur des ensembles flous à l'aide de fonctions d'appartenances. De même, les variables de sortie définies sur des ensembles flous sont défuzzifiées pour les exprimer comme des variables réelles continues.

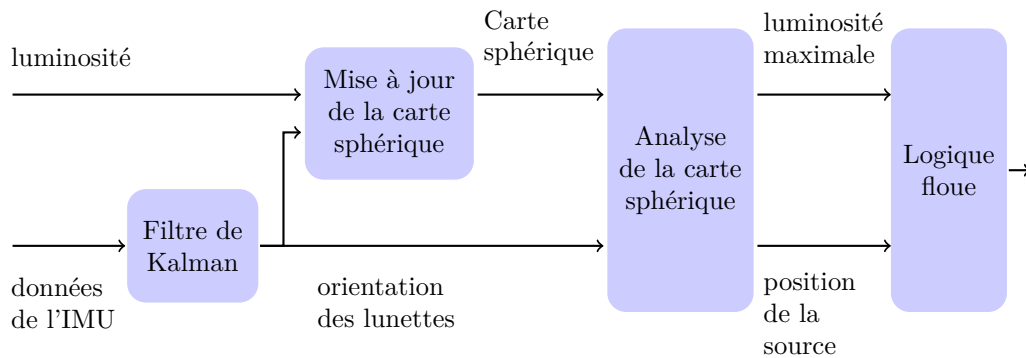


FIGURE 4.25 – Schéma bloc illustrant l'architecture logicielle du prototype. Tout d'abord, les données de l'IMU et de la photodiode sont acquises. À partir des données de l'IMU, l'orientation des lunettes est estimée par un filtre de Kalman. Ensuite, la carte sphérique est mise à jour avec la dernière luminosité perçue et l'orientation des lunettes. La carte sphérique est alors analysée pour détecter la luminosité maximale visible par l'utilisateur. Puis, connaissant l'intensité de la source et sa distance angulaire au centre du champ de vision, on déduit une commande d'obscurcissement en se basant sur le calcul de logique floue.

processus de logique floue, nous définissons deux variables : la luminosité maximale perçue et sa distance angulaire par rapport au centre de la vision de l'utilisateur. La seconde variable permet de modifier la commande en fonction de la position de la source ponctuelle. Ainsi, nous pouvons considérer qu'à intensité égale une source face à l'utilisateur est plus gênante qu'une autre dans la périphérie de son champ de vision.

Le choix de ne détecter qu'une seule source ponctuelle peut se révéler limitant. En effet, si plusieurs sources ponctuelles sont présentes dans la scène, une seule sera détectée, ce qui n'induirait peut-être pas un obscurcissement suffisant. Considérons un exemple où la source de luminosité maximale se trouve en périphérie et la seconde source légèrement plus faible est au centre du champ de vision. L'obscurcissement déduit par logique floue pour la source en périphérie sera peut-être plus faible que pour l'autre source. Une solution intéressante serait de détecter plusieurs maxima locaux. Pour résoudre ce problème, le champ de vision a été découpé en trois parties distinctes : le champ de vision périphérique, le champ de vision intermédiaire et le champ de vision central. Pour chacune d'elles, une luminosité maximale est détectée et la logique floue calcule une valeur d'obscurcissement. La commande appliquée correspond à l'obscurcissement le plus élevé.

#### 4.3.3 Test sur simulateur et mise en place du prototype

Les étapes de mise à jour de la carte sphérique de luminosité et de détection de source ponctuelle dans le champ de vision de l'utilisateur ont été simulées sur le

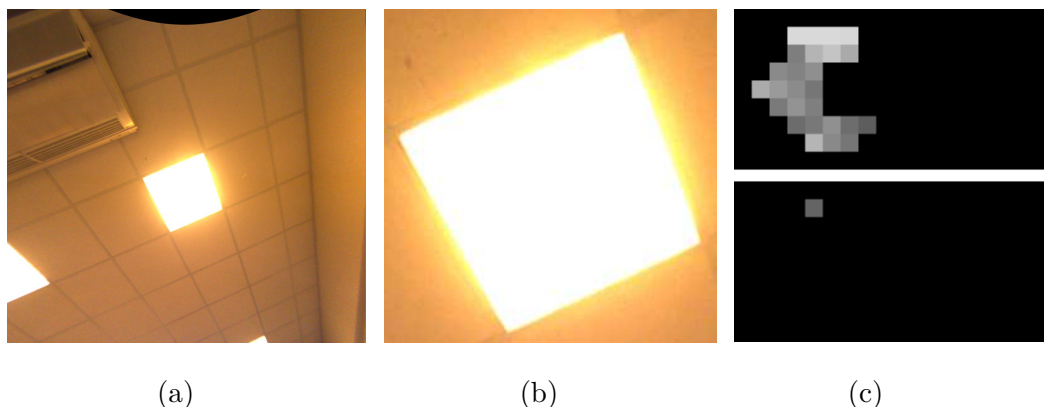


FIGURE 4.26 – Figure illustrant la simulation sur le VST-HMD. (a) est l'image acquise par une des caméras du banc stéréo. (b) est l'image (a) tronquée et agrandie pour simuler la photodiode de 20 degré d'ouverture. (c) regroupe deux images. L'une est la carte de luminosité enregistrée et la seconde désigne par un carré gris la position actuelle pointée par la photodiode sur la carte sphérique.

banc de test basé VST-HMD. Grâce au banc de test, il était plus facile de développer les algorithmes et de les tester rapidement dans différentes conditions lumineuses. L'orientation des lunettes était simulée à partir des données de poses calculées par le SLAM et la valeur de la photodiode (ALS) a été simulée à partir de l'image d'une caméra de scène. La figure 4.26 illustre cette étape de simulation sur le VST-HMD.

La carte sphérique admet deux dimensions et est définie suivant le jeu de coordonnées azimut et élévation. Cette représentation a été choisie pour des raisons de simplicité et rapidité de mise en place. Toutefois, elle présente l'inconvénient de sur-échantillonner les données sur les pôles (nord et sud) de la sphère. Ce sur-échantillonnage n'est pas en adéquation avec nos objectifs de basse consommation.

Après cette simulation, un prototype de lunettes (Figure 4.27) a été conçu en se basant sur des outils de prototypages rapides : cartes et composants électroniques Arduino, bibliothèque logicielle Arduino et interface logicielle codée en python. Un capteur IMU comprenant un accéléromètre, un gyromètre et un compas a été intégré aux lunettes. Une photodiode diaphragmée à 20 degrés est montée sur l'avant des lunettes entre les deux verres électro-chromes. L'acquisition des données des capteurs, la logique floue<sup>9</sup> et le filtre de Kalman<sup>10</sup> appliqué aux données de l'IMU sont gérés par des bibliothèques Arduino open source. Grâce à ces outils, un prototype a rapidement pu être mis en place.

La figure 4.28 montre une capture d'écran de l'interface logicielle construite. Les données calculées sont affichées dans différents graphiques en temps réel. En haut à gauche, on peut visualiser la carte sphérique enregistrée. Les pixels noirs correspondent à des directions encore non observées. De gauche à droite, le premier graphique trace l'orientation en azimut et élévation définie depuis une position

9. <https://github.com/zerokol/eFLL>

10. <https://github.com/RTIMULib/RTIMULib-Arduino>

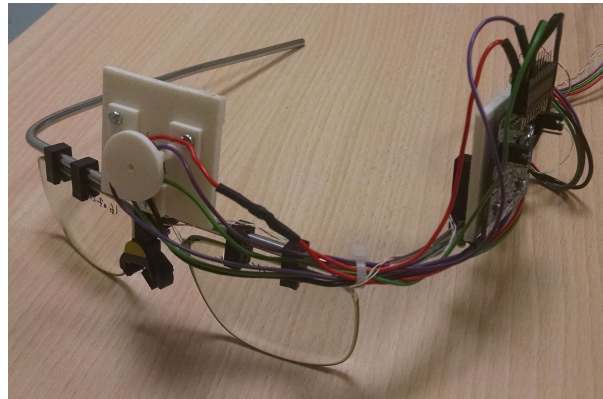


FIGURE 4.27 – Photo du prototype de lunettes équipé des deux verres électrochrome, d’une photodiode diaphragmée positionnée entre les deux verres, et d’une IMU fixée sur le côté. Les calculs sont effectués sur une Arduino Mega alimentée par une batterie nomade. Ces deux composants, non-présents sur la photo, sont embarqués dans une sacoche en bandoulière. Pour visualiser les données, une connexion USB avec un ordinateur peut être établie et un script python affiche les résultats dans des graphiques en temps réel.

d’origine et le second affiche l’évolution de la valeur de commande calculée par la logique floue. Les graphiques du dessous affichent les résultats des calculs pour chacune des zones, avec les luminosités maximales, leur distance angulaire au centre du champ de vision et la commande correspondante.

Lors des tests, nous avons pu constater que malgré la détection de source lumineuse sur l’ensemble du champ de vision de l’utilisateur, les lunettes ne réagissaient pas assez vite. Ce phénomène est dû à la vitesse de migration des molécules au sein de la solution électro-chromique. Les lunettes ont besoin de quelques secondes pour atteindre la teinte recherchée. Une solution pour obscurcir les verres en avance serait d’anticiper la commande à envoyer en s’appuyant sur la prédiction de l’orientation des lunettes. D’autre part, la carte ne se met pas à jour régulièrement sur l’ensemble du champ de vision. Souvent, l’utilisateur garde la tête immobile et ne parcourt pas un grand nombre de directions. Une alternative possible serait de placer plusieurs photodiodes dans différentes orientations.

#### 4.3.4 Perspectives

Ce concept de lunettes de teinte variable a tout d’abord été simulé sur le banc de test pour faciliter les étapes de développement. Puis, sur la même idée, un prototype a été mis en place. Après quelques tests, nous avons pu identifier des limites du système et proposer des solutions :

- Le prototype actuel détecte la luminosité maximale dans trois zones du champ de vision. Cette méthode permet de détecter plusieurs sources ponctuelles dans différentes zones du champ de vision. Néanmoins, la commande des verres peut être discontinuée lorsqu’une source lumineuse se déplace d’une zone à une

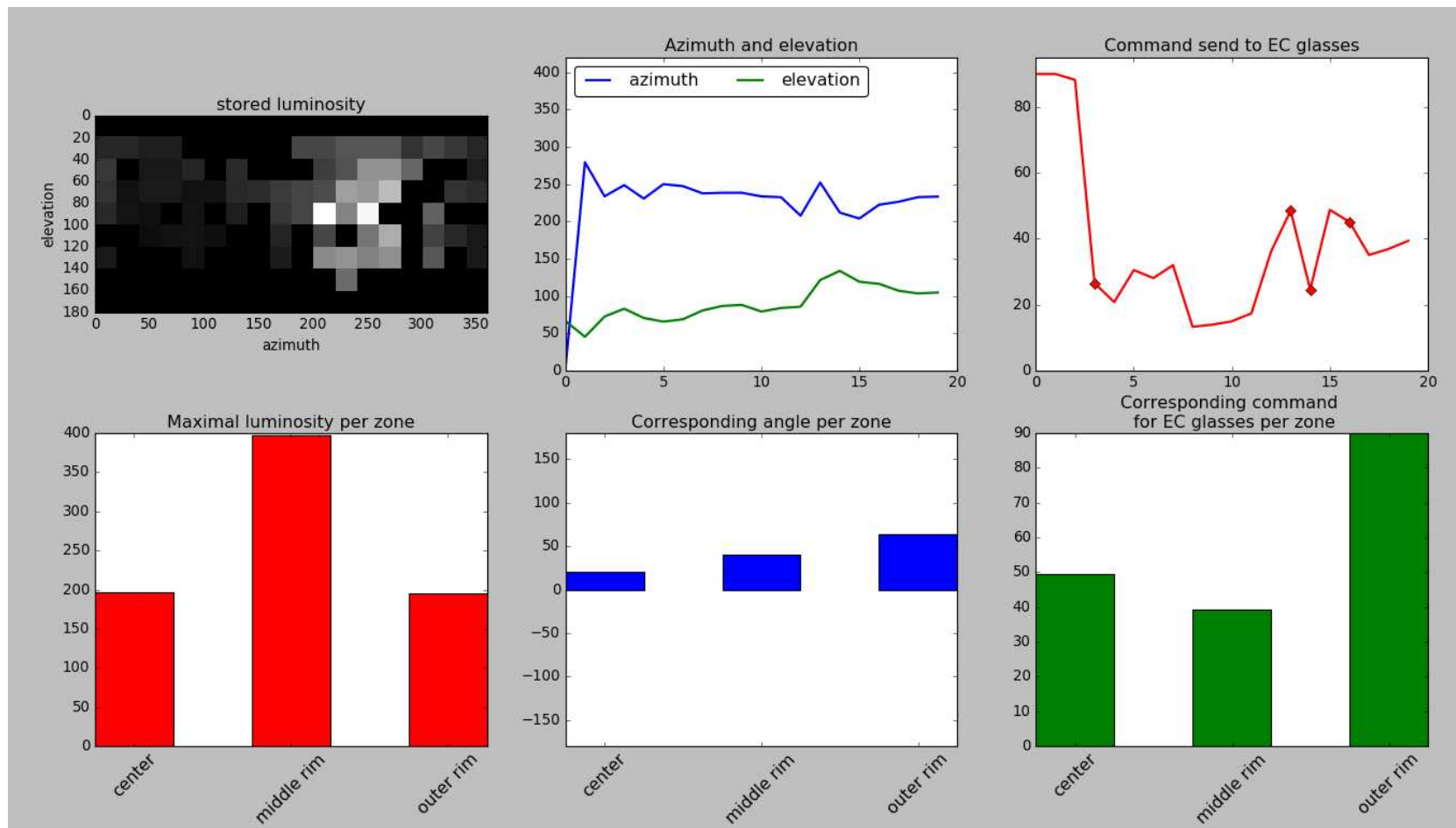


FIGURE 4.28 – Capture d’écran de l’interface logicielle codée en python. Elle affiche en temps réel les données envoyées par USB depuis l’Arduino.

autre. Par exemple, considérons que la scène visible comporte deux sources : une intense en périphérie  $S_1$  et une plus faible  $S_2$  dans la zone intermédiaire. Si la source  $S_2$  se déplace vers la zone en périphérie alors elle ne sera plus détectée, car elle est moins forte que  $S_1$ . Ce changement de détection est susceptible de provoquer une discontinuité de la commande, qui est cependant amortie par la lenteur de l'actionneur. De plus si la source lumineuse  $S_1$  oscille d'une zone à l'autre, la commande du verre oscillera elle aussi. Une solution possible, pour pallier ce problème, serait d'estimer les maxima locaux sur la portion de carte observée, d'appliquer le calcul de logique floue sur chacun d'eux et de conserver la commande ayant l'obscurcissement le plus élevé. Une autre approche pourrait être d'appliquer le processus de logique floue sur l'ensemble des valeurs observées par l'utilisateur parmi celles de la carte de luminosité et d'appliquer la teinte calculée la plus foncée. La méthode choisie ne doit cependant pas être trop coûteuse en calculs pour respecter la contrainte de basse consommation.

- La carte ne se met pas à jour assez régulièrement dans plusieurs directions. Pour régler ce problème, d'autres photodiodes pourraient être montées sur les lunettes.
- La sensibilité à la lumière dépend de la luminosité précédemment perçue par l'utilisateur. Il sera plus facilement ébloui s'il quitte un environnement sombre. Pour tenir compte de ce paramètre, une variable supplémentaire pourrait être ajoutée en entrée du calcul de logique floue : la valeur de luminosité perçue précédemment.
- La carte sphérique est actuellement définie suivant les coordonnées azimuth et élévation. Cette représentation a le désavantage de sur-échantillonner la carte au niveau des pôles. D'autres systèmes de coordonnées pourraient être considérés.
- Les verres mettent quelques secondes pour atteindre l'obscurcissement ciblé. Cependant, la vitesse de teinte est plus importante au début. On observe visuellement les verres se foncer rapidement dès la première seconde. Ce phénomène génère un petit effet de retard, que l'utilisateur perçoit lorsqu'il tourne la tête en direction d'une source lumineuse. Une alternative possible serait d'anticiper la commande à envoyer aux verres, pour qu'ils se teintent en avance. Afin d'atteindre cet objectif, nous pourrions prédire la direction des lunettes et analyser la portion de carte bientôt perceptible.
- Une autre amélioration possible de ce dispositif serait d'incorporer un oculomètre. L'estimation de la distance angulaire entre la source et le regard de l'utilisateur serait raffinée, en tenant compte cette fois-ci, non plus de la direction des lunettes, mais de la direction du regard lui-même. Par ailleurs, avec un oculomètre, nous pourrions estimer le diamètre de la pupille, ce qui nous renseignerait sur la sensibilité de l'individu à la lumière. Cette donnée serait alors ajoutée comme entrée de la logique floue.

## 4.4 Conclusion

Pour étudier la structure et la construction de lunettes actives, trois prototypes ont été réalisés. Ainsi, dans le but de tester et d'évaluer différents composants internes aux lunettes, les deux premiers prototypes mis en place sont des bancs de test (ou prototypage) définis sur des architectures modulaires. Leur architecture modulaire permet de tester plus facilement différents composants logiciels.

Le premier prototype est un VST-HMD composé d'un casque de réalité virtuelle, d'un banc stéréoscopique de caméras fisheye et d'un oculomètre basé caméra. La particularité du VST-HMD est que l'utilisateur observe la scène sur un écran. Il ne l'observe pas directement comme avec un OST-HMD. En conséquence, il est plus aisé de simuler des verres actifs. Le VST-HMD constitue donc un banc de test, où différentes lunettes peuvent être simulées et évaluées. Les premiers tests effectués sur le casque nous ont permis de valider le choix d'une construction d'un dispositif de lunettes actives sur une architecture modulaire. D'autre part, comme le point de vue des caméras du banc stéréoscopique et des yeux de l'utilisateur sont différents, il nous a fallu mettre en place un algorithme de DIBR synthétisant des images de la scène pour les points de vue des yeux de l'utilisateur. Une expérience nous a permis de constater que l'emploi du DIBR améliore la perception 3D de l'utilisateur. Des compromis (entre performances et délais disponible) ont été faits lors de la conception du VST-HMD. Les composants développés ne sont donc pas les plus adaptés pour notre usage. C'est pourquoi de nombreuses améliorations sont nécessaires pour permettre une utilisation confortable du VST-HMD, notamment celles concernant la latence et la fréquence d'affichage, qui est de l'ordre de 1 Hz aujourd'hui.

Le second prototype est un OST-HMD, initialement équipé d'un système d'affichage avec verres transparents, d'un banc de stéréovision et d'un IMU, auquel nous avons ajouté un oculomètre et deux verres électrochromes. Grâce à ces verres transparents, ce dispositif peut être utilisé pour tester différentes techniques d'étalonnage de OST-HMD, qui pourront par la suite être transposées à d'autres verres actifs commandables zone par zone. Des premières techniques d'étalonnage ont été testées et évaluées sur l'OST-HMD et ont permis d'établir des conclusions. La méthode du SPAAM fournit des résultats intéressants. Cependant, elle ne peut pas être utilisée en pratique sur des lunettes actives du quotidien, car l'étalonnage doit être effectué à chaque déplacement des lunettes sur le visage. Pour résoudre ce problème, une solution est d'utiliser pour le système d'affichage un modèle de projection incluant un déplacement possible de l'œil par rapport aux lunettes. La position de l'œil est alors mis à jour à l'aide d'un oculomètre. Toutefois, nous avons pu constater que l'ajout d'un oculomètre sur l'OST-HMD n'est pas simple. La caméra de l'oculomètre doit observer l'ensemble de l'œil. Pour cela, il est préférable qu'elle soit éloignée de l'œil, ce qui est compliqué à réaliser au sein du OST-HMD car l'espace disponible est restreint. L'algorithme d'oculométrie considéré ici mérite également quelques améliorations : choisir un modèle plus complexe pour l'œil et ajouter un filtrage temporel pour le suivi du regard afin d'éviter les données erronées.

Le troisième prototype est une lunette active basse consommation équipée de verres obscurcissant et doté de capacité de perception, d'analyse, de décision et d'action sur les verres. Pour répondre à ces contraintes de coût énergétique, la luminosité de la scène est acquise par une photodiode plutôt qu'une caméra, dont l'acquisition et le traitement des images sont coûteux en énergie. Comme une photodiode fournit une information de luminosité dans une seule direction, nous proposons de construire dynamiquement une carte pour stockées les données de luminosité observée. Nous faisons l'hypothèse que l'environnement lumineux évolue lentement et que les sources lumineuses considérées (soleil, ciel, nuage) sont éloignées relativement aux déplacements des lunettes. Partant de cette hypothèse, nous construisons une carte sphérique de l'environnement lumineux à l'aide d'une photodiode et d'une centrale inertielle, qui fournit l'orientation des lunettes dans la scène au moyen d'un filtre de Kalman. Les algorithmes de perception et de cartographie ont tout d'abord été développés sur le VST-HMD, pour faciliter le développement. Ensuite, un prototype a été conçu en s'appuyant sur des outils de prototypage rapide. Lors des tests, nous avons pu observer certains défauts du système (lenteur du remplissage de la carte et de sa mise à jour, portion de la carte restant vide, lenteur de la réaction des verres) et plusieurs solutions ont été proposées pour améliorer le dispositif.

De ces trois prototypes, nous avons pu acquérir de l'expérience sur la construction de lunettes actives et identifier leurs points bloquants et les solutions envisageables. De nombreuses améliorations ont encore besoin d'être apportées pour que ces systèmes puissent être utilisés en tant que lunettes actives ou banc de test.

Confidentiel

# Conclusion

## Rappel du sujet

L'objectif de cette thèse était d'étudier la construction et le fonctionnement de lunettes actives, en considérant ces dernières comme des systèmes robotiques : Leur fonctionnement nécessite d'effectuer des mesures sur l'environnement et de les analyser, puis de prendre des décisions et finalement d'agir sur la perception de l'utilisateur. De plus, les lunettes actives et les systèmes robotiques partagent des contraintes communes : consommation énergétique, encombrement et poids réduits ainsi que des considérations liés aux aspects temps réel.

Trois types d'actions effectuées par les lunettes sont considérés ici : l'affichage de contenu synthétique dans le champ de vision de l'utilisateur, la correction adaptative de la vue (dans laquelle la correction du verre s'adapte afin d'offrir une vision nette en fonction des objets visés dans la scène) et l'assombrissement des verres en fonction de la luminosité ambiante.

Pour cette étude, les lunettes actives sont associées à une architecture logicielle modulaire. L'ensemble des traitements est ainsi divisé en plusieurs composants élémentaires que l'on appelle modules traitant par exemple de : la perception et la cartographie de la scène, la localisation des lunettes, la génération d'une commande adéquate à envoyer aux actionneurs. Pour étudier le fonctionnement de ces modules, tester et évaluer différentes solutions et méthodes, nous avons mis en place des bancs de test et de prototypage. D'autre part, grâce à ces bancs, il a été possible de simuler des lunettes actives équipées de différents capteurs et actionneurs.

## Étude des modules

Un état de l'art des modules étudiés a été présenté au chapitre 2. Pour chacun d'eux, nous avons exposé des solutions variées permettant d'effectuer les tâches qui leur sont associées. Par la suite, les bancs de tests ont été élaborés sur la base de cet état de l'art et des besoins énoncés. En complément de cette étude, le module d'oculométrie basé photodiodes et réseau de neurones a fait l'objet d'une contribution plus importante que les autres, l'oculomètre devant respecter des contraintes fortes d'encombrement, de poids et de coût énergétique. Une étude a été menée pour étudier l'influence de la configuration de l'assemblage de photodiodes sur les performances de l'oculomètre. Pour effectuer cette étude, nous avons mis au point un simulateur permettant de tester et d'évaluer différents assemblages en évitant les délais de fabrication de dispositifs réels. À partir de ces simulations, nous avons déduit certaines règles sur la configuration de l'assemblage. Premièrement, le champ de vue de chaque photodiode doit de préférence être faible (de l'ordre de 10 degrés). Deuxièmement, elles doivent observer différentes portions de l'œil de manière à couvrir la plus grande surface possible. Ces premiers résultats étant néanmoins limités

Confidentiel

au cadre de la simulation sur des données synthétiques, de nouveaux travaux ont été initiés afin de valider la faisabilité d'un tel oculomètre en situation réelle : l'environnement lumineux complexe constitué de différentes sources lumineuses d'intensité et de position variables, la forme du visage variable d'une personne à l'autre et les réflexions complexes de la lumière sur l'œil. Pour cela, nous proposons de mettre en place un banc de test, qui permettra d'acquérir des données d'apprentissage pour le réseau de neurones intégré à l'oculomètre et d'évaluer ses performances.

## Réalisation de bancs de tests

Deux bancs de tests ont été mis au point pour simuler des lunettes actives. Le premier est basé sur la technologie d'un VST-HMD qui, grâce à son écran opaque, permet de simuler des verres actifs. Par exemple, pour l'obscurcissement automatique des verres, l'image affichée au sein du VST-HMD peut simplement être plus ou moins assombrie. Pour la correction adaptative de la vue, l'image peut être localement floutée suivant la focale du verre et la distance des objets de la scène à l'utilisateur. Ce banc de test est constitué d'un casque de réalité virtuelle, d'un banc stéréoscopique de caméras fisheye filmant la scène et d'un oculomètre basé caméra infrarouge. Une architecture logicielle a été proposée, intégrant les processus de localisation du casque dans la scène, de calcul de carte de profondeur, d'estimation de la direction du regard et de génération d'images.

Pour évaluer le fonctionnement complet du banc de prototypage et de ces modules, une application d'exemple a été réalisée. La qualité de la perception 3D à travers le casque a également été évaluée ; En effet, les images de la scène acquises par les caméras montées sur le VST-HMD ne partageant pas le même point de vue que les yeux de l'utilisateur, il a été nécessaire de générer des images adaptées aux yeux du porteur. Pour cela, nous avons utilisé un algorithme de synthèse basé image et carte de profondeur ("depth image based rendering" - DIBR). La génération de nouvelles images permet de réduire l'inconfort et améliore la perception 3D au sein du casque. Les tests effectués nous ont permis de constater que notre approche basée DIBR propose une meilleure perception 3D que l'utilisation directe des images fournies par les caméras mais tout de même moins bonne que sans casque.

Plusieurs suggestions d'améliorations ont été proposées :

- Tout d'abord, pour réduire la latence et augmenter la fréquence d'affichage, certains traitements sur les images peuvent être déportés sur GPU. Les images peuvent également être générées à l'avance pour des positions prédites des yeux, plutôt que des positions associées aux dernières images acquises.
- Pour améliorer la perception 3D à travers le casque, le modèle de projection du casque devrait intégrer les déplacements de l'œil (localisé par l'oculomètre) et l'algorithme de DIBR devrait être remanié afin de tenir compte de ces déplacements.
- L'estimation de la pose des lunettes relativement à la scène peut être améliorée en intégrant une centrale inertielle au banc de test et en utilisant ces mesures

à l'algorithme de localisation et de cartographie.

- Les algorithmes de perception de la scène doivent encore être adaptés au modèle de caméra omnidirectionnelle, ce qui permettrait d'utiliser l'ensemble du champ de vision des caméras.
- Le banc de stéréovision nécessitant d'être réétalonné régulièrement, un étalonnage automatique en ligne serait appréciable.
- Les modèles géométriques d'œil et d'écran virtuel de l'oculomètre intégré au VST-HMD sont très simples, ce qui aboutit à des faibles précisions pour l'estimation de la direction du regard. Il serait préférable de considérer des modèles plus riches.

Le second banc de test conçu au cours de la thèse est basé sur un OST-HMD. Les spécificités de ce dispositif nous permettent de tester des techniques d'étalonnage différentes de celles utilisées pour les VST-HMD. Par la suite, ces techniques pourraient être étendues pour d'autres lunettes actives (par exemple de focale ou de teinte variable) avec lesquelles l'action sur le champ de vision de l'utilisateur se définit localement. Le banc de test mis en place s'appuie sur des lunettes de réalité augmentée existantes équipées d'un banc stéréoscopique et d'une centrale inertielle, auxquelles nous avons ajouté des verres électro-chromes et un oculomètre basé caméra infrarouge. Sur le même principe que le précédent banc de test, une architecture logicielle modulaire a été utilisée pour l'OST-HMD.

Trois méthodes d'étalonnage ont été testées sur le banc de prototypage. Elles s'appuient sur le modèle de projection trou d'épingle, qui peut être facilement utilisé pour la synthèse d'images avec des outils logiciels adaptés tels qu'OpenGL. Néanmoins, ces approches ne tiennent pas compte du changement de pose possible entre les lunettes et la tête du porteur. Des techniques répondant à ce besoin sont présentées dans l'état de l'art du chapitre 2. Elles s'appuient sur l'estimation de la position de l'œil fournie par un oculomètre intégré au dispositif. Elles pourront plus tard être testées sur le banc de test. L'intégration de l'oculomètre au sein de l'OST-HMD ayant été plus délicate que sur le VST-HMD, du fait de l'espace plus restreint, une caméra avec un plus large champ de vision (permettant d'observer l'ensemble de l'œil depuis une courte distance) permettrait de couvrir une plus grande plage angulaire de directions du regard. D'autre part, les mesures délivrées par l'oculomètre étant impactées par des valeurs aberrantes transitoires dues aux reflets sur la cornée, à du flou de bougé ou à des clignements des yeux, il serait intéressant d'évaluer l'apport d'un filtrage temporel afin d'en réduire l'impact.

Confidentiel

# Bref historique de la réalité augmentée

---

## A.1 Avant 2015

Le premier prototype de réalité augmentée est décrit dans [Sutherland 1968]. Il s'agit d'un système de type OST-HMD fixé sur la tête de l'utilisateur, qui superpose à la scène une image générée par ordinateur (Figure A.1). Cependant, le terme réalité augmentée dans son acception actuelle a été introduit la première fois dans [Caudell 1992].

Au cours des années 70 et 80, la recherche sur ce domaine portait essentiellement sur des systèmes de réalité augmentée pour l'aéronautique et le spatial [Furness III 1986]. Puis dans les années 90, les projets de recherche se sont multipliés sur d'autres applications telles que le médical [Bajura 1992], le travail collaboratif [Rekimoto 1996] et sur l'amélioration de points particuliers des systèmes de RA tels que la mobilité [Feiner 1997], la transparence [Kancherla 1996], le tracking [Azuma 1993], avec notamment l'arrivée de la bibliothèque logicielle de détection de tag ARToolKit [Kato 1999].

Ensuite dans les années 2000, une partie des travaux étaient consacrée aux HHDs, avec en particulier la réalité augmentée sur téléphone. D'autre part, des outils logiciels et des sociétés spécialisés dans la réalité augmentée ont commencé à émerger, telles que l'outil ARToolKit<sup>1</sup> et la société METAIO (rachetée par Apple

---

1. <https://www.hitl.washington.edu/artoolkit/>

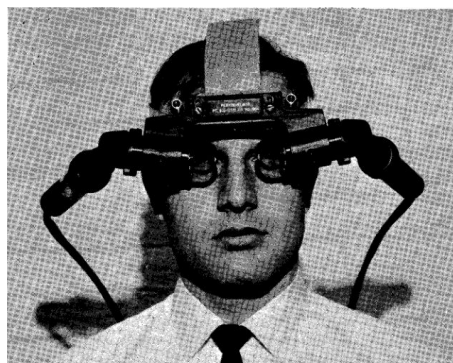


FIGURE A.1 – Premier prototype de réalité augmentée de type OST-HMD proposé par Sutherland en 1968.

en 2015 [Miller 2015]). Grâce à ces outils, à la baisse du coût des caméras et à l'amélioration des puissances de calculs des ordinateurs, la mise en place d'applications de réalité augmentée a été facilitée. C'est pourquoi, à la fin des années 2000, des applications commerciales sont apparues pour la publicité [BMW Group 2008] et le divertissement, avec par exemple, en 2007, le jeu "The Eye of Judgement" sur PlayStation<sup>2</sup>.

## A.2 Depuis 2015

En parallèle des recherches menées au sein des laboratoires, le sujet de la réalité augmentée est de plus en plus étudié dans les entreprises pour des fins commerciales. C'est ainsi qu'au début de la thèse en 2015, de nombreuses entreprises avaient déjà développé et présenté des prototypes expérimentaux de casques ou lunettes de réalité augmentée. Cependant, rares étaient celles qui avaient tenté de les commercialiser (Google Glass<sup>3</sup>, Epson Moverio BT-200<sup>4</sup>). De leur côté, des prototypes de casques de réalité virtuelle, tels que l'Oculus DK2, étaient déjà en vente pour les développeurs, mais non ouverts au grand public.

C'est en 2016 que la société Magic Leap chamboule le marché, lorsqu'elle présente ses prototypes dotés d'une nouvelle technologie proposant un affichage lumineux et focalisant à des distances variables [Kelly 2016]. Cette entreprise très médiatisée suscite d'importants investissements depuis 2014 [Gelles 2014], [Massoudi 2018]. En parallèle, chez les concurrents, des casques de réalité augmentée commencent à être commercialisés pour des publics encore restreints : Meta2 de Metavision<sup>5</sup>, LumusDK50 de Lumus<sup>6</sup>, et Hololens de Microsoft<sup>7</sup>. Seul Hololens semble vouloir viser à terme un public plus large avec le concept de l'holoportation [Orts-Escolano 2016].

Ces casques ont pour points communs leur encombrement et leur forte consommation énergétique qui limite leur autonomie. Ainsi, Intel se démarque avec ses lunettes de réalité augmentée dotées d'une faible consommation énergétique et d'une apparence proche de lunettes traditionnelles [Bohn 2018].

## A.3 Positionnement d' Essilor

De son côté, Essilor, au début de la thèse, n'avait pas encore ciblé de marché particulier : dispositif encombrant ou léger, de forte ou faible consommation énergétique, pour la santé ou pour tout usage. L'arrivée d'acteurs tels que Microsoft (Figure A.3), Magic Leap (Figure A.2) montre que, dans un futur proche, le

2. <https://www.playstation.com/en-us/games/the-eye-of-judgment-ps3/>

3. <https://www.x.company/glass/>

4. <https://www.epson.fr/products/see-through-mobile-viewer/moverio-bt-200>

5. <http://www.metavision.com/>

6. <https://lumusvision.com/>

7. <https://www.microsoft.com/en-us/hololens>

marché du divertissement sera largement occupé. Les dispositifs de ces sociétés proposent de nombreuses possibilités d'interaction avec l'utilisateur et la scène réelle, mais sont énergivores. En conséquence, ils disposent d'une autonomie réduite. Par exemple, elle n'est que de trois heures pour le Magic Leap One. Voilà pourquoi, aujourd'hui, Essilor se focalise essentiellement sur des technologies basse consommation très intégrées, destinées à un port quotidien. L'objectif est d'atteindre une autonomie d'au moins d'une journée complète et de proposer des lunettes discrètes, comparables à des montures de verres solaires.



FIGURE A.2 – Casque Magic Leap One de la société Magic leap (extrait de <https://www.magicleap.com/>)



FIGURE A.3 – Casque HoloLens de Microsoft (extrait de <https://www.microsoft.com/en-us/hololens>)

De nombreuses prévisions annoncent une accélération de l'évolution du marché de la réalité augmentée dans les années à venir. Les investissements devraient augmenter comme le suggère Business Insider (Figure A.4), qui prévoit en 2021 un investissement supérieur à 4 fois l'investissement de 2018. Le *hype cycle* de Gartner (Figure A.5), qui prédit l'évolution des attentes pour les technologies émergentes avant leur introduction sur le marché, prévoit l'apparition de casques et lunettes de réalité augmentée sur le marché grand public d'ici 5 à 10 ans.

Une description plus exhaustive de l'histoire de la réalité augmentée est disponible dans [Billinghamurst 2015].

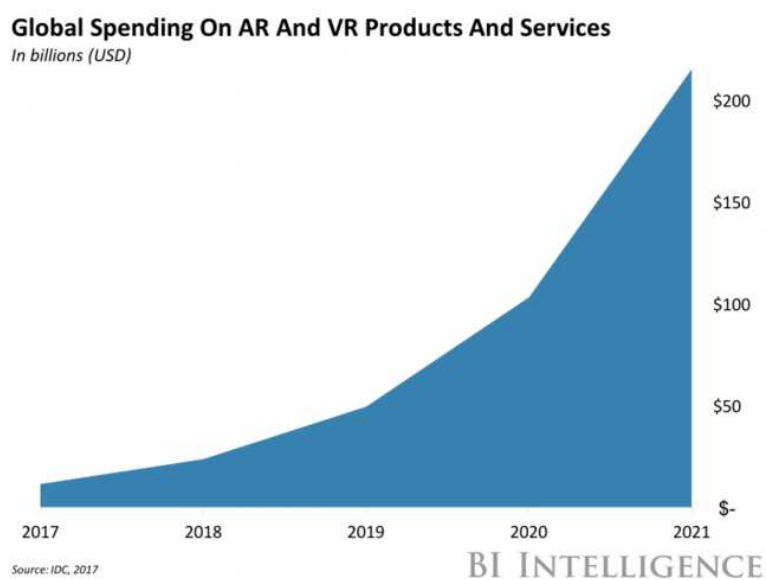
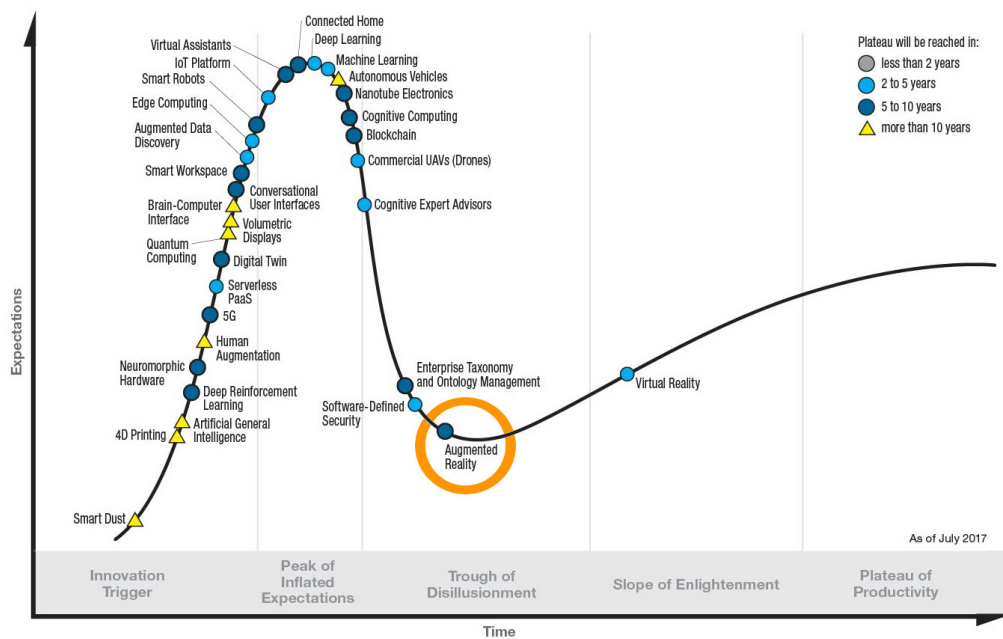


FIGURE A.4 – Pr vision (ao t 2017) des investissements dans le secteur de la r alit  augment e et virtuelle de 2017   2021 par Business Insider (extrait de [http ://www.businessinsider.fr/us/ar-vr-2017-8](http://www.businessinsider.fr/us/ar-vr-2017-8)).

### Gartner **Hype Cycle** for Emerging Technologies, 2017



[gartner.com/SmarterWithGartner](https://www.gartner.com/SmarterWithGartner)

Source: Gartner (July 2017)  
© 2017 Gartner, Inc. and/or its affiliates. All rights reserved.

**Gartner**

FIGURE A.5 – Représentation du *hype cycle* par Gartner en Juillet 2017, qui prédit l'évolution des attentes pour les technologies émergentes d'ici leur introduction sur le marché (extrait de <https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017>).

Confidentiel



# Bibliographie

- [Adelstein 2003] B.D. Adelstein, T.G. Lee et S.R. Ellis. *Head Tracking Latency in Virtual Environments : Psychophysics and a Model*. Dans Human Factors and Ergonomics Society Annual Meeting, volume 47, pages 2083–2087. SAGE Publications, 2003. (Cité en pages 38 et 154.)
- [Ahn 2012] Ilkoo Ahn et Changick Kim. *Depth-Based Disocclusion Filling for Virtual View Synthesis*. Dans IEEE International Conference on Multimedia and Expo, pages 109–114. IEEE, 2012. (Cité en page 80.)
- [Alami 1998] Rachid Alami, Raja Chatila, Sara Fleury, Malik Ghallab et Félix Ingrand. *An Architecture for Autonomy*. The International Journal of Robotics Research, vol. 17, no. 4, pages 315–337, 1998. (Cité en page 28.)
- [Ando 2012] Tomoaki Ando, Vasily G Moshnyaga et Koji Hashimoto. *A Low-Power FPGA Implementation of Eye Tracking*. Dans IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’2012), pages 1573–1576. IEEE, 2012. (Cité en page 68.)
- [ANSES 2014] ANSES. *Technologies 3D et Vision : Usage Déconseillé aux Enfants de Moins de 6 ans, Modéré pour les Moins de 13 ans*, novembre 2014. <https://www.anses.fr/fr/content/technologies-3d-et-vision-usage-deconseille-aux-enfants-de-moins-de-6-ans-moderes-pour-les>. (Cité en pages 18 et 37.)
- [Atzpadin 2004] Nicole Atzpadin, Peter Kauff et Oliver Schreer. *Stereo Analysis by Hybrid Recursive Matching for Real-Time Immersive Video Conferencing*. IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 3, pages 321–334, 2004. (Cité en page 157.)
- [Axholt 2011] Magnus Axholt, Matthew D Cooper, Martin A Skoglund, Stephen R Ellis, Stephen D O’Connell et Anders Ynnerman. *Parameter Estimation Variance of the Single Point Active Alignment Method in Optical See-Through Head Mounted Display Calibration*. Dans IEEE Virtual Reality Conference (VR’2011), pages 27–34. IEEE, 2011. (Cité en page 84.)
- [Azuma 1993] R. Azuma. *Tracking Requirements for Augmented Reality*. Communications of the ACM, vol. 36, no. 7, pages 50–51, 1993. (Cité en page 185.)
- [Bajura 1992] M. Bajura, H. Fuchs et R. Ohbuchi. *Merging Virtual Objects with the Real World : Seeing Ultrasound Imagery within the Patient*. ACM SIGGRAPH Computer Graphics, vol. 26, no. 2, pages 203–210, 1992. (Cité en page 185.)
- [Bax 2004] Michael R Bax. *Real-Time Lens Distortion Correction : 3D Video Graphics Cards are Good for More than Games*. Stanford Electrical Engineering and Computer Science Research Journal (Stanford ECJ), pages 9–13, 2004. (Cité en page 88.)

- [Baya 2008] Herbert Baya, Andreas Essa, Tinne Tuytelaars et Luc Van Gool. *Speeded-Up Robust Features (SURF)*. Computer Vision and Image Understanding, vol. 110, no. 3, pages 346–359, 2008. (Cité en page 57.)
- [Benko 2014] H. Benko, A.D. Wilson et F. Zannier. *Dyadic Projected Spatial Augmented Reality*. Dans ACM Annual Symposium on User Interface Software and Technology, pages 645–655, 2014. (Cité en page 21.)
- [Billinghurst 2015] M. Billinghurst, A. Clark, G. Lee et al. *A Survey of Augmented Reality*. Foundations and Trends® in Human–Computer Interaction, vol. 8, no. 2-3, pages 73–272, 2015. (Cité en page 187.)
- [Birchfield 1998] Stan Birchfield et Carlo Tomasi. *A pixel dissimilarity measure that is insensitive to image sampling*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 4, pages 401–406, 1998. (Cité en page 50.)
- [Bishop 2010] T.E. Bishop et P. Favaro. *Full-Resolution Depth Map Estimation from an Aliased Plenoptic Light Field*. Dans Asian Conference on Computer Vision, pages 186–200. Springer, 2010. (Cité en page 53.)
- [Bleyer 2013] M. Bleyer et C. Breiteneder. *Stereo Matching State-of-the-Art and Research Challenges*. Dans Advanced Topics in Computer Vision, pages 143–179. Springer, 2013. (Cité en page 50.)
- [BMW Group 2008] BMW Group. *The new MINI Cabrio - Launch Campaign With Augmented Reality Technology*. Techcrunch, novembre 2008. <https://www.press.bmwgroup.com/global/photo/detail/P0051421/the-new-mini-cabrio-launch-campaign-with-augmented-reality-technology-11-2008>. (Cité en page 186.)
- [Bohn 2018] D. Bohn. *Intel Made Smart Glasses that Look Normal*. The Verge, février 2018. <https://www.theverge.com/2018/2/5/16966530/intel-vaunt-smart-glasses-announced-ar-video>. (Cité en pages 92 et 186.)
- [Bouchon-Meunier 2007] Bernadette Bouchon-Meunier. *La logique floue : «que sais-je ?»* n° 2702. Presses universitaires de France, 2007. (Cité en page 172.)
- [Breuer 2014] Timo Breuer, Christoph Bodensteiner et Michael Arens. *Low-Cost Commodity Depth Sensor Comparison and Accuracy Analysis*. Dans Electro-Optical Remote Sensing, Photonic Technologies, and Applications VIII ; and Military Applications in Hyperspectral Imaging and High Spatial Resolution Sensing II, volume 9250, page 92500G. International Society for Optics and Photonics, 2014. (Cité en page 46.)
- [Calonder 2010] Michael Calonder, Vincent Lepetit, Christoph Strecha et Pascal Fua. *Brief : Binary Robust Independent Elementary Features*. Dans European Conference on Computer Vision, pages 778–792. Springer, 2010. (Cité en page 57.)
- [Carmigniani 2011] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani et M. Ivkovic. *Augmented Reality Technologies, Systems and Applications*. Multimedia Tools and Applications, vol. 51, no. 1, pages 341–377, 2011. (Cité en page 35.)

- [Carvalho 2003] L.A. Carvalho. *A Simple Mathematical Model for Simulation of the Human Optical System based on in vivo Corneal Data*. Research on Biomedical Engineering, vol. 19, no. 1, pages 29–37, avril 2003. (Cité en page 13.)
- [Caudell 1992] T.P. Caudell et D.W. Mizell. *Augmented Reality : An Application of Heads-Up Display Technology to Manual Manufacturing Processes*. Dans IEEE International Conference on System Sciences (ICSS'1992), volume 2, pages 659–669, 1992. (Cité en page 185.)
- [Chambon 2011] S. Chambon et A. Crouzil. *Similarity Measures for Image Matching Despite Occlusions in Stereo Vision*. Pattern Recognition, vol. 44, no. 9, pages 2063–2075, 2011. (Cité en page 48.)
- [Chen 2007] J.Y.C. Chen et J.E. Thropp. *Review of Low Frame Rate Effects on Human Performance*. IEEE Transactions on Systems, Man, and Cybernetics-Part A : Systems and Humans, vol. 37, no. 6, pages 1063–1076, 2007. (Cité en pages 38 et 154.)
- [Chen 2016] W. Chen, Z. Fu, D. Yang et J. Deng. *Single-Image Depth Perception in the Wild*. Dans Advances in Neural Information Processing Systems, pages 730–738, 2016. Code source disponible sur [https://github.com/umich-vl/relative\\_depth](https://github.com/umich-vl/relative_depth). (Cité en page 54.)
- [Chennamma 2013] H.R. Chennamma et Xiaohui Yuan. *A Survey on Eye-Gaze Tracking Techniques*. Indian Journal of Computer Science and Engineering (IJCSE), pages 388–393, 2013. (Cité en pages 62 et 63.)
- [Choi 2013] Sunghwan Choi, Bumsub Ham et Kwanghoon Sohn. *Space-Time Hole Filling with Random Walks in View Extrapolation for 3D Video*. IEEE Transactions on Image Processing, vol. 22, no. 6, pages 2429–2441, 2013. (Cité en page 80.)
- [Claypool 2006] M. Claypool, K. Claypool et F. Damaa. *The Effects of Frame Rate and Resolution on Users Playing First Person Shooter Games*. Dans Electronic Imaging 2006, volume 6071. International Society for Optics and Photonics, 2006. (Cité en pages 38 et 154.)
- [Comport 2006] Andrew I Comport, Eric Marchand, Muriel Pressigout et Francois Chaumette. *Real-Time Markerless Tracking for Augmented Reality : The Virtual Visual Servoing Framework*. IEEE Transactions on Visualization and Computer Graphics, vol. 12, no. 4, pages 615–628, 2006. (Cité en page 58.)
- [Cornsweet 1973] Tom N Cornsweet et Hewitt D Crane. *Accurate Two-Dimensional Eye Tracker using First and Fourth Purkinje Images*. Journal of Optical Society of America, vol. 63, no. 8, pages 921–928, 1973. (Cité en page 65.)
- [Criminisi 2003] Antonio Criminisi, Patrick Perez et Kentaro Toyama. *Object Removal by Exemplar-Based Inpainting*. Dans IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'2003), volume 2, pages II–721. IEEE, 2003. (Cité en pages 79 et 144.)

- [Damen 2012] Dima Damen, Andrew Gee, Walterio Mayol-Cuevas et Andrew Calway. *Egocentric Real-Time Workspace Monitoring using an RGB-D Camera*. Dans IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2012), pages 1029–1036. IEEE, 2012. (Cité en page 61.)
- [Davison 2003] Andrew J Davison. *Real-Time Simultaneous Localisation and Mapping with a Single Camera*. Dans IEEE International Conference on Computer Vision (ICCV), pages 1403–1410. IEEE, 2003. (Cité en page 60.)
- [De Sorbier 2010] François De Sorbier, Yuki Takaya, Yuko Uematsu, Ismael Daribo et Hideo Saito. *Augmented Reality for 3D TV using Depth Camera Input*. Dans International Conference on Virtual Systems and Multimedia (VSMM'2010), pages 117–123. IEEE, 2010. (Cité en page 73.)
- [Do 2012] Luat Do, German Bravo, Svitlana Zinger et Peter HN De With. *GPU-Accelerated Real-Time Free-Viewpoint DIBR for 3DTV*. IEEE Transactions on Consumer Electronics, vol. 58, no. 2, 2012. (Cité en page 74.)
- [Dobias 2011] M. Dobias et R. Sara. *Real-Time Global Prediction for Temporally Stable Stereo*. Dans IEEE International Conference on Computer Vision Workshops (ICCV Workshops'2011), pages 704–707. IEEE, 2011. (Cité en page 52.)
- [Dragoi 2018] V Dragoi, J. H. Byrne, J. C. Waymire, A. J. Bean, N. Waxham, N. Dafny, P. Dougherty, M. O. Hutchins, L. Gray, J. Knierim, A. Wright et P. Dash. *Neuroscience Online, an Electronic Textbook for the Neurosciences*, 2018. <http://nba.uth.tmc.edu/neuroscience/>, mise à disposition par le département de neurobiologie et de l'anatomie à l'Université du Texas, Section 2, Chap 14-15. (Cité en page 8.)
- [Eigen 2014] D. Eigen, C. Puhrsch et R. Fergus. *Depth Map Prediction from a Single Image using a Multi-Scale Deep Network*. Dans Advances in Neural Information Processing Systems, pages 2366–2374, 2014. Code source disponible sur <https://github.com/hjimce/Depth-Map-Prediction>. (Cité en page 54.)
- [El Jaafari 2016] I. El Jaafari, M. El Ansari, L. Koutti, A. Mazoul et A. Ellahyani. *Fast Spatio-Temporal Stereo Matching for Advanced Driver Assistance Systems*. Neurocomputing, vol. 194, pages 24–33, 2016. (Cité en page 52.)
- [Emsley 1952] H.H. Emsley. Visual optics, volume 1. Butterworths, 5 édition, 1952. (Cité en page 12.)
- [Engel 2014] Jakob Engel, Thomas Schöps et Daniel Cremers. *LSD-SLAM : Large-Scale Direct Monocular SLAM*. Dans European Conference on Computer Vision, pages 834–849. Springer, 2014. (Cité en page 61.)
- [Essilor 2018] Essilor. *À propos d'Essilor*, 2018. <https://www.essilor.fr/a-propos>. (Cité en page 18.)
- [Feiner 1997] S. Feiner, B. MacIntyre, T. Höllerer et A. Webster. *A Touring Machine : Prototyping 3D Mobile Augmented Reality Systems for Exploring the*

- Urban Environment*. Personal Technologies, vol. 1, no. 4, pages 208–217, 1997. (Cité en page 185.)
- [Felzenszwalb 2012] Pedro F Felzenszwalb et Daniel P Huttenlocher. *Distance Transforms of Sampled Functions*. Theory of Computing, vol. 8, no. 1, pages 415–428, 2012. (Cité en page 144.)
- [Flynn 2016] John Flynn, Ivan Neulander, James Philbin et Noah Snavely. *Deepstereo : Learning to Predict New Views from the World’s Imagery*. Dans IEEE International Conference on Computer Vision and Pattern Recognition (CVPR’2016), pages 5515–5524, 2016. (Cité en pages 81 et 145.)
- [Forster 2014] Christian Forster, Matia Pizzoli et Davide Scaramuzza. *SVO : Fast Semi-Direct Monocular Visual Odometry*. Dans IEEE International Conference on Robotics and Automation (ICRA’2014), pages 15–22. IEEE, 2014. (Cité en page 61.)
- [Fuhl 2016] Wolfgang Fuhl, Thiago Santini, Gjergji Kasneci et Enkelejda Kasneci. *PupilNet : Convolutional Neural Networks for Robust Pupil Detection*. arXiv preprint arXiv :1601.04902, 2016. (Cité en page 64.)
- [Fuhrmann 1999] Anton Fuhrmann, Dieter Schmalstieg et Werner Purgathofer. *Fast Calibration for Augmented Reality*. Dans ACM symposium on Virtual Reality Software and Technology, pages 166–167. ACM, 1999. (Cité en pages 83 et 84.)
- [Furness III 1986] T.A. Furness III. *The Super Cockpit and its Human Factors Challenges*. Dans Human Factors Society Annual Meeting, volume 30, pages 48–52. SAGE Publications, Los Angeles, CA, 1986. (Cité en page 185.)
- [Fusiello 2000] A. Fusiello, E. Trucco et A. Verri. *A Compact Algorithm for Rectification of Stereo Pairs*. Machine Vision and Applications, vol. 12, no. 1, pages 16–22, 2000. (Cité en page 48.)
- [Gallup 2007] D. Gallup, J.M. Frahm, P. Mordohai, Q. Yang et M. Pollefeys. *Real-time Plane-Sweeping Stereo with Multiple Sweeping Directions*. Dans IEEE International Conference on Computer Vision and Pattern Recognition (CVPR’2007), pages 1–8. IEEE, 2007. (Cité en page 50.)
- [Gálvez-López 2012] Dorian Gálvez-López et Juan D Tardos. *Bags of Binary Words for Fast Place Recognition in Image Sequences*. IEEE Transactions on Robotics, vol. 28, no. 5, pages 1188–1197, 2012. (Cité en page 62.)
- [Garg 2016] R. Garg, B.G. Vijay Kumar, G. Carneiro et I. Reid. *Unsupervised CNN for Single View Depth Estimation : Geometry to the Rescue*. Dans IEEE European Conference on Computer Vision (ECCV’2016), pages 740–756, 2016. Code source disponible sur [https://github.com/Ravi-Garg/Unsupervised\\_Depth\\_Estimation](https://github.com/Ravi-Garg/Unsupervised_Depth_Estimation). (Cité en page 54.)
- [Gatinel 2013a] D. Gatinel. *Research / Education*, 2013. <https://www.gatinel.com/en/recherche-formation/>. (Cité en page 8.)

- [Gatinel 2013b] D. Gatinel. *Study of the Retinal Image*, octobre 2013. <https://www.gatinel.com/en/recherche-formation/oeil-et-optique-introduction/etude-limage-retinienne/>. (Cité en page 8.)
- [Gelles 2014] D. Gelles et M.J. de la Merced. *Google Invests Heavily in Magic Leap's Effort to Blend Illusion and Reality*. The New York Times, octobre 2014. <https://dealbook.nytimes.com/2014/10/21/google-invests-in-magic-leap-an-augmented-reality-firm/>. (Cité en page 186.)
- [Gilson 2008] Stuart J Gilson, Andrew W Fitzgibbon et Andrew Glennerster. *Spatial Calibration of an Optical See-Through Head-Mounted Display*. Journal of Neuroscience Methods, vol. 173, no. 1, pages 140–146, 2008. (Cité en pages 83, 85 et 88.)
- [Gilson 2011] Stuart J Gilson, Andrew W Fitzgibbon et Andrew Glennerster. *An Automated Calibration Method for Non-See-Through Head Mounted Displays*. Journal of Neuroscience Methods, vol. 199, no. 2, pages 328–335, 2011. (Cité en page 88.)
- [Glorot 2010] Xavier Glorot et Yoshua Bengio. *Understanding the Difficulty of Training Deep Feedforward Neural Networks*. Dans International Conference on Artificial Intelligence and Statistics, pages 249–256, 2010. (Cité en pages 107 et 108.)
- [Glorot 2011] Xavier Glorot, Antoine Bordes et Yoshua Bengio. *Deep Sparse Rectifier Neural Networks*. Dans International Conference on Artificial Intelligence and Statistics, pages 315–323, 2011. (Cité en page 107.)
- [Godard 2017] C. Godard, O. Mac Aodha et G. J. Brostow. *Unsupervised Monocular Depth Estimation with Left-Right Consistency*. Dans IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2017), pages 6602–6611, 2017. Code source disponible sur <https://github.com/mrharicot/monodepth>. (Cité en page 54.)
- [Guestrin 2006] Elias Daniel Guestrin et Moshe Eizenman. *General Theory of Remote Gaze Estimation using the Pupil Center and Corneal Reflections*. IEEE Transactions on Biomedical Engineering, vol. 53, no. 6, pages 1124–1133, 2006. (Cité en pages 64, 66, 67 et 151.)
- [Gui 2013] Haitian Gui, Zhiyong Pang, Dihui Chen, Min Chen et Hongzhou Tan. *A Forward and Reverse Wrapping Depth Image-Based Rendering (FR-DIBR) Method for Arbitrary View Generation*. Dans International Conference on Green Communications and Networks 2012 (GCN 2012), pages 683–690. Springer, 2013. (Cité en page 79.)
- [Guillemot 2014] Christine Guillemot et Olivier Le Meur. *Image Inpainting : overview and Recent Advances*. IEEE Signal Processing Magazine, vol. 31, no. 1, pages 127–144, 2014. (Cité en page 79.)
- [Hänsel 2017] M. Hänsel, M. Rosenberger et G. Notni. *FPGA Implementation of a Multi-View Stereo Approach for Depth Estimation and Image Reconstruction*

- for Plenoptic Cameras*. Dans Ilmenau Scientific Colloquium on Engineering for a Changing World, volume 59, 2017. (Cité en page 53.)
- [Hansen 2010] D.W. Hansen et Q. Ji. *In the Eye of the Beholder : A Survey of Models for Eyes and Gaze*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 3, pages 478–500, 2010. (Cité en pages 10, 62, 63 et 64.)
- [Haouchine 2013] N. Haouchine, J. Dequidt, M.O. Berger et S. Cotin. *Deformation-based Augmented Reality for Hepatic Surgery*. Studies in Health Technology and Informatics, vol. 184, 2013. (Cité en page 19.)
- [Harris 1988] Christopher G Harris et JM Pike. *3D Positional Integration from Image Sequences*. Image and Vision Computing, vol. 6, no. 2, pages 87–90, 1988. (Cité en page 60.)
- [He 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren et Jian Sun. *Delving Deep into Rectifiers : Surpassing Human-Level Performance on Imagenet Classification*. Dans IEEE International Conference on Computer Vision (CV-PR’2015), pages 1026–1034, 2015. (Cité en pages 108 et 109.)
- [Heikkila 1997] J. Heikkila et O. Silven. *A Four-Step Camera Calibration Procedure with Implicit Image Correction*. Dans IEEE International Conference on Computer Vision and Pattern Recognition (CVPR’97), pages 1106–1112, 1997. (Cité en pages 40, 42 et 43.)
- [Hirschmuller 2008] H. Hirschmuller. *Stereo Processing by Semiglobal Matching and Mutual Information*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 2, pages 328–341, 2008. (Cité en pages 50 et 141.)
- [Hirschmuller 2009] H. Hirschmuller et D. Scharstein. *Evaluation of Stereo Matching Costs on Images with Radiometric Differences*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 9, pages 1582–1599, 2009. (Cité en page 48.)
- [Horaud 2016] Radu Horaud, Miles Hansard, Georgios Evangelidis et Clément Ménérier. *An Overview of Depth Cameras and Range Scanners based on Time-Of-Flight Technologies*. Machine Vision and Applications, vol. 27, no. 7, pages 1005–1020, 2016. (Cité en page 46.)
- [Hua 2002] Hong Hua, Chunyu Gao et Narendra Ahuja. *Calibration of a Head-Mounted Projective Display for Augmented Reality Systems*. Dans IEEE International Symposium on Mixed and Augmented Reality (ISMAR’2002), pages 176–185. IEEE, 2002. (Cité en page 83.)
- [Hua 2014] H. Hua et B. Javidi. *A 3D Integral Imaging Optical See-Through Head-Mounted Display*. Optics Express, vol. 22, no. 11, pages 13484–13491, 2014. (Cité en page 37.)
- [Huang 2009] Y. Huang, Y. Liu et Y. Wang. *AR-View : An Augmented Reality Device for Digital Reconstruction of Yuangmingyuan*. Dans IEEE International Symposium on Mixed and Augmented Reality-Arts, Media and Humanities, pages 3–7, 2009. (Cité en page 19.)

- [Hughes 2008] C. Hughes, M. Glavin, E. Jones et P. Denny. *Review of Geometric Distortion Compensation in Fish-Eye Cameras*. Dans IET Irish Signals and Systems Conference (ISSC 2008), 2008. (Cité en page 43.)
- [Irie 2002] Kenji Irie, Bruce A Wilson, Richard D Jones, Philip J Bones et Tim J Anderson. *A Laser-Based Eye-Tracking System*. Behavior Research Methods, Instruments, & Computers, vol. 34, no. 4, pages 561–572, 2002. (Cité en page 63.)
- [Itoh 2014] Yuta Itoh et Gudrun Klinker. *Interaction-Free Calibration for Optical See-Through Head-Mounted Displays based on 3D Eye Localization*. Dans IEEE Symposium on 3D User Interfaces (3DUI'2014), pages 75–82. IEEE, 2014. (Cité en pages 83, 87, 137, 156, 162 et 169.)
- [Itoh 2015a] Yuta Itoh et Gudrun Klinker. *Light-Field Correction for Spatial Calibration of Optical See-Through Head-Mounted Displays*. IEEE Transactions on Visualization and Computer Graphics, vol. 21, no. 4, pages 471–480, 2015. (Cité en page 84.)
- [Itoh 2015b] Yuta Itoh et Gudrun Klinker. *Simultaneous Direct and Augmented View Distortion Calibration of Optical See-Through Head-Mounted Displays*. Dans IEEE International Symposium on Mixed and Augmented Reality (ISMAR'2015), pages 43–48. IEEE, 2015. (Cité en pages 83, 84 et 169.)
- [Jain 2014] A.K. Jain et T.O. Nguyen. *Discriminability Limits in Spatio-Temporal Stereo Block Matching*. IEEE Transactions on Image Processing, vol. 23, no. 5, pages 2328–2342, 2014. (Cité en pages 51 et 52.)
- [Janin 1993] Adam L Janin, David W Mizell et Thomas P Caudell. *Calibration of Head-Mounted Displays for Augmented Reality Applications*. Dans IEEE Annual International Symposium on Virtual Reality, pages 246–255. IEEE, 1993. (Cité en pages 83 et 84.)
- [Jones 2015] J Adam Jones, Lauren Cairco Dukes, David M Krum, Mark T Bolas et Larry F Hodges. *Correction of Geometric Distortions and the Impact of Eye Position in Virtual Reality Displays*. Dans International Conference on Collaboration Technologies and Systems (CTS'2015), pages 77–83. IEEE, 2015. (Cité en page 88.)
- [Kancherla 1996] A. Kancherla, M. Singer et J. Rolland. *Calibrating See-Through Head-Mounted Displays*. Rapport technique, Univ./ North Carolina, 1996. (Cité en page 185.)
- [Kannala 2006] J. Kannala et S.S. Brandt. *A Generic Camera Model and Calibration Method for Conventional, Wide-Angle, and Fish-Eye Lenses*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 8, pages 1335–1340, 2006. (Cité en page 45.)
- [Kassner 2014] Moritz Kassner, William Patera et Andreas Bulling. *Pupil : An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-Based Interaction*. Dans ACM International Joint Conference on Pervasive and

- Ubiquitous Computing : Adjunct Publication, pages 1151–1160. ACM, 2014. (Cité en pages 66, 70, 94 et 148.)
- [Kato 1999] H. Kato et M. Billinghurst. *Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System*. Dans IEEE/ACM International Workshop on Augmented Reality (IWAR'99), pages 85–94, 1999. (Cité en pages 83 et 185.)
- [Kauff 2007] Peter Kauff, Nicole Atzpadin, Christoph Fehn, Marcus Müller, Oliver Schreer, Aljoscha Smolic et Ralf Tanger. *Depth Map Creation and Image-Based Rendering for Advanced 3DTV Services providing Interoperability and Scalability*. Signal Processing : Image Communication, vol. 22, no. 2, pages 217–234, 2007. (Cité en pages 73, 82 et 143.)
- [Keller 2005] Kurtis P Keller, Henry Fuchset al. *Simulation-Based Design and Rapid Prototyping of a Parallax-Free, Orthoscopic Video See-Through Head-Mounted Display*. Dans Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality, pages 28–31. IEEE Computer Society, 2005. (Cité en page 72.)
- [Kellner 2012] Falko Kellner, Benjamin Bolte, Gerd Bruder, Ulrich Rautenberg, Frank Steinicke, Markus Lappe et Reinhard Koch. *Geometric Calibration of Head-Mounted Displays and its Effects on Distance Estimation*. IEEE Transactions on Visualization and Computer Graphics, vol. 18, no. 4, pages 589–596, 2012. (Cité en pages 83, 84 et 85.)
- [Kelly 2016] L. Kelly. *The Untold Story of Magic Leap, the World's Most Secretive Startup*. Wired, avril 2016. <https://www.wired.com/2016/04/magic-leap-vr/>. (Cité en page 186.)
- [Kendall 2017] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach et A. Bry. *End-to-End Learning of Geometry and Context for Deep Stereo Regression*. IEEE International Conference on Computer Vision (ICCV), pages 66–75, 2017. (Cité en page 54.)
- [Klemm 2016] Martin Klemm, Fabian Seebacher et Harald Hoppe. *Non-Parametric Camera-Based Calibration of Optical See-Through Glasses for AR Applications*. Dans International Conference on Cyberworlds (CW'2016), pages 33–40. IEEE, 2016. (Cité en pages 83, 86 et 88.)
- [Klemm 2017] Martin Klemm, Fabian Seebacher et Harald Hoppe. *High Accuracy Pixel-Wise Spatial Calibration of Optical See-Through Glasses*. Computers & Graphics, vol. 64, pages 51–61, 2017. (Cité en pages 83, 86 et 169.)
- [Kohavi 1995] Ron Kohavi et al. *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection*. Dans International Joint Conference on Artificial Intelligence (IJCAI'1995), volume 14, pages 1137–1145. Montreal, Canada, 1995. (Cité en page 104.)
- [Kolakowski 2006] Susan M Kolakowski et Jeff B Pelz. *Compensating for Eye Tracker Camera Movement*. Dans Symposium on Eye tracking research & applications, pages 79–85. ACM, 2006. (Cité en page 65.)

- [Kooijman 1983] A.C. Kooijman. *Light Distribution on the Retina of a Wide-Angle Theoretical Eye*. Journal of the Optical Society of America, vol. 73, no. 11, pages 1544–1550, novembre 1983. (Cit  en page 13.)
- [Kowalczyk 2013] J drzej Kowalczyk, Eric T Psota et Lance C P rez. *Real-Time Temporal Stereo Matching using Iterative Adaptive Support Weights*. Dans IEEE International Conference on Electro/Information Technology (EIT’2013), pages 1–6. IEEE, 2013. (Cit  en page 51.)
- [Krafka 2016] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Su-chendra Bhandarkar, Wojciech Matusik et Antonio Torralba. *Eye Tracking for Everyone*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR),, pages 2176–2184, 2016. (Cit  en page 63.)
- [Kuhl 2009] Scott A Kuhl, William B Thompson et Sarah H Creem-Regehr. *HMD Calibration and its effects on Distance Judgments*. ACM Transactions on Applied Perception, vol. 6, no. 3, page 19, 2009. (Cit  en page 88.)
- [Kumar 2009] Nishant Kumar, Stefan Kohlbecher et Erich Schneider. *A Novel Approach to Video-Based Pupil Tracking*. Dans IEEE International Conference on Systems, Man and Cybernetics (SMC’2009), pages 1255–1262. IEEE, 2009. (Cit  en page 64.)
- [Lai 2016a] Chun-Jui Lai, Ping-Hsuan Han et Yi-Ping Hung. *View Interpolation for Video See-Through Head-Mounted Display*. Dans ACM SIGGRAPH 2016 Posters, page 57. ACM, 2016. (Cit  en page 73.)
- [Lai 2016b] Chun-Jui Lai, Ping-Hsuan Han, Han-Lei Wang et Yi-Ping Hung. *Exploring Manipulation Behavior on Video See-Through Head-Mounted Display with View Interpolation*. Dans Asian Conference on Computer Vision, pages 258–270. Springer, 2016. (Cit  en pages 76 et 157.)
- [Le Corre 2016] B. Le Corre. *Oculus Rift : Jouer ou Vomir, il Faut Choisir ?* Rue89 de Nouvel Obs, mars 2016. <https://www.nouvelobs.com/rue89/rue89-tech/20160329.RUE8971/oculus-rift-jouer-ou-vomir-il-faut-choisir.html>. (Cit  en page 38.)
- [Le Grand 1964] Y. Le Grand. *Optique physiologique : La dioptrique de l’oeil et sa correction*. tome premier., volume 1. Masson, 1964. (Cit  en page 9.)
- [LeCun 1998] Yann LeCun, L on Bottou, Genevieve B Orr et Klaus-Robert M ller. *Efficient Backprop*. Dans Neural Networks : tricks of the Trade, pages 9–50. Springer, 1998. (Cit  en page 108.)
- [Lee 2015] Sangyoon Lee et Hong Hua. *A Robust Camera-Based Method for Optical Distortion Calibration of Head-Mounted Displays*. Journal of Display Technology, vol. 11, no. 10, pages 845–853, 2015. (Cit  en page 83.)
- [Li 2005] Dongheng Li, David Winfield et Derrick J Parkhurst. *Starburst : A Hybrid Algorithm for Video-Based Eye Tracking Combining Feature-Based and Model-Based Approaches*. Dans IEEE Computer Vision and Pattern Recognition Workshops, pages 79–79. IEEE, 2005. (Cit  en pages 65, 66, 148 et 149.)

- [Li 2006] Dongheng Li, Jason Babcock et Derrick J Parkhurst. *openEyes : A Low-Cost Head-Mounted Eye-Tracking Solution*. Dans Symposium on Eye Tracking Research & Applications, pages 95–100. ACM, 2006. (Cité en pages 65 et 66.)
- [Liarokapis 2004] F. Liarokapis, N. Mourkoussis, M. White, J. Darcy, M. Sifniotis, P. Petridis, A. Basu et P.F. Lister. *Web3D and Augmented Reality to Support Engineering Education*. World Transactions on Engineering and Technology Education, vol. 3, no. 1, pages 11–14, 2004. (Cité en page 19.)
- [Liu 2015] F. Liu, C. Shen, G. Lin et I. Reid. *Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 10, pages 2024–2039, 2015. Code source disponible sur <https://github.com/raingo/dcnf-fcsp>. (Cité en page 54.)
- [Lotmar 1971] W. Lotmar. *Theoretical Eye Model with Aspherics*. Journal of the Optical Society of America, vol. 61, no. 11, pages 1522–1529, novembre 1971. (Cité en page 13.)
- [Lowe 2004] David G Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision, vol. 60, no. 2, pages 91–110, 2004. (Cité en page 57.)
- [Maimone 2014] A. Maimone, D/ Lanman, K. Rathinavel, K. Keller, D. Luebke et H. Fuchs. *Pinlight Displays : Wide Field of View Augmented Reality Eyeglasses using Defocused Point Light Sources*. Dans ACM SIGGRAPH 2014 Emerging Technologies, page 20, 2014. (Cité en page 37.)
- [Mamdani 1974] Ebrahim H Mamdani. *Application of Fuzzy Algorithms for Control of Simple Dynamic Plant*. Dans Proceedings of the IEE, volume 121, pages 1585–1588. IET, 1974. (Cité en page 172.)
- [Mao 2013] Yu Mao, Gene Cheung, Antonio Ortega et Yusheng Ji. *Expansion Hole Filling in Depth-Image-Based Rendering using Graph-Based Interpolation*. Dans IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’2013), pages 1859–1863, 2013. (Cité en page 79.)
- [Marchand 2016] Eric Marchand, Hideaki Uchiyama et Fabien Spindler. *Pose Estimation for Augmented Reality : a Hands-On Survey*. IEEE Transactions on Visualization and Computer Graphics, vol. 22, no. 12, pages 2633–2651, 2016. (Cité en page 58.)
- [Massoudi 2018] A. Massoudi et T. Bradshaw. *Magic Leap confirms \$400m Saudi Investment*. The Financial Times, mars 2018. <https://www.ft.com/content/cb7d71a2-221e-11e8-9a70-08f715791301>. (Cité en page 186.)
- [Mayberry 2014] Addison Mayberry, Pan Hu, Benjamin Marlin, Christopher Salthouse et Deepak Ganesan. *iShadow : Design Of A Wearable, Real-Time Mobile Gaze Tracker*. Dans Annual International Conference on Mobile

- Systems, Applications, and Services, pages 82–94. ACM, 2014. (Cité en pages 68, 69, 70 et 93.)
- [Mayberry 2015] Addison Mayberry, Yamin Tun, Pan Hu, Duncan Smith-Freedman, Deepak Ganesan, Benjamin M Marlin et Christopher Salthouse. *CIDER : Enabling Robustness-Power Tradeoffs on a Computational Eyeglass*. Dans Annual International Conference on Mobile Computing and Networking, pages 400–412. ACM, 2015. (Cité en pages 68 et 93.)
- [McKay 1987] D. McKay. <http://www.redcedar.com/owlQandA.html>, 1987. Accessed : May 2018. (Cité en page 63.)
- [Mei 2011] Xing Mei, Xun Sun, Mingcai Zhou, Shaohui Jiao, Haitao Wang et Xiaopeng Zhang. *On Building an Accurate Stereo Matching System on Graphics Hardware*. Dans IEEE Intelligence Conference on Computer Vision Workshops (ICCV Workshops-2011), pages 467–474. IEEE, 2011. (Cité en page 50.)
- [Meilland 2013] Maxime Meilland et Andrew I Comport. *On Unifying Key-Frame and Voxel-Based Dense Visual SLAM at Large Scales*. Dans IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2013), pages 3677–3683, 2013. (Cité en page 61.)
- [Milgram 1994] P. Milgram et F. Kishino. *A Taxonomy of Mixed Reality Visual Displays*. IEICE Transactions on Information and Systems, vol. 77, no. 12, pages 1321–1329, 1994. (Cité en page 20.)
- [Miller 2015] R. Miller et J. Constine. *Apple Acquires Augmented Reality Company Metaio*. Techcrunch, mai 2015. <https://techcrunch.com/2015/05/28/apple-metaio/>. (Cité en page 186.)
- [Missal 2018] M. Missal, A. Mouraux et E. Olivier. *Neurophysiologie*, 2018. <https://www.neurophysiologie.be/index.php>, chap. Vision et oculomotricité et chap. Systèmes sensoriels et potentiels évoqués. (Cité en page 8.)
- [Morimoto 2005] C.H. Morimoto et M.R.M. Mimica. *Eye Gaze Tracking Techniques for Interactive Applications*. Computer Vision and Image Understanding, vol. 98, pages 4–24, avril 2005. Issue 1. (Cité en page 15.)
- [Morvan 2009] Y Yanninck Morvan. *Acquisition, Compression and Rendering of Depth and Texture for Multi-View Video*. PhD thesis, Technische Universiteit Eindhoven, 2009. (Cité en page 74.)
- [Mun 2016] J.H. Mun et Y.S. Ho. *Temporal Domain Stereo Matching based on Feature Points for Restriction of Error Propagation*. Electronic Imaging, vol. 2016, no. 21, pages 3DIPM–402, 2016. (Cité en page 52.)
- [Mur-Artal 2017a] Raul Mur-Artal et Juan D Tardós. *ORB-SLAM2 : An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras*. IEEE Transactions on Robotics, vol. 33, no. 5, pages 1255–1262, 2017. (Cité en pages 62 et 141.)

- [Mur-Artal 2017b] Raúl Mur-Artal et Juan D Tardós. *Visual-Inertial Monocular SLAM with Map Reuse*. IEEE Robotics and Automation Letters, vol. 2, no. 2, pages 796–803, 2017. (Cité en page 62.)
- [Nana 2005] L Nana. *Architectures Logicielles pour la Robotique*. Dans Journées Nationales de la Recherche en Robotique (JNRR'05), 2005. (Cité en page 28.)
- [Navarro 1985] R. Navarro, J. Santamaría et J. Bescós. *Accommodation-dependent Model of the Human Eye with Aspherics*. Journal of the Optical Society of America, vol. 2, no. 8, pages 1273–1281, août 1985. (Cité en page 13.)
- [Nehani 2015] Jetmir Nehani, Davide Brunelli, Michele Magno, Lukas Sigrist et Luca Benini. *An Energy Neutral Wearable Camera with EPD Display*. Dans Workshop on Wearable Systems and Applications, pages 1–6. ACM, 2015. (Cité en page 68.)
- [Newcombe 2011] Richard A Newcombe, Steven J Lovegrove et Andrew J Davison. *DTAM : Dense Tracking and Mapping in Real-Time*. Dans IEEE International Conference on Computer Vision (ICCV'2011), pages 2320–2327. IEEE, 2011. (Cité en page 61.)
- [Nitschke 2013a] Christian Nitschke, Atsushi Nakazawa et Toyoaki Nishida. *I See What You See : Point of Gaze Estimation from Corneal Images*. Dans IAPR Asian Conference on Pattern Recognition (ACPR'2013), pages 298–304. IEEE, 2013. (Cité en pages 64, 65, 66, 67 et 132.)
- [Nitschke 2013b] Christian Nitschke, Atsushi Nakazawa et Haruo Takemura. *Corneal Imaging Revisited : An Overview of Corneal Reflection Analysis and Applications*. IPSJ Transactions on Computer Vision and Applications, vol. 5, pages 1–18, 2013. (Cité en page 64.)
- [Oliveira 2015] Adriano Oliveira, Guilherme Fickel, Marcelo Walter et Cláudio Jung. *Selective Hole-Filling for Depth-Image Based Rendering*. Dans IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2015), pages 1186–1190. IEEE, 2015. (Cité en pages 78, 79 et 80.)
- [Olson 2011] Edwin Olson. *AprilTag : A Robust and Flexible Visual Fiducial System*. Dans IEEE International Conference on Robotics and Automation (ICRA'2011), pages 3400–3407. IEEE, 2011. (Cité en pages 57, 58, 155 et 166.)
- [Orts-Escolano 2016] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P.L. Davidson, S. Khamis, M. Douet *al.* *Holoportation : Virtual 3D Teleportation in Real-Time*. Dans ACM Annual Symposium on User Interface Software and Technology, pages 741–754, 2016. (Cité en pages 19 et 186.)
- [Owen 2004] Charles B Owen, Ji Zhou, Arthur Tang et Fan Xiao. *Display-Relative Calibration for Optical See-Through Head-Mounted Displays*. Dans IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'2004), pages 70–78. IEEE, 2004. (Cité en pages 83, 85, 160 et 169.)

- [Pankratz 2015] Frieder Pankratz et Gudrun Klinker. *AR4AR : Using Augmented Reality for Guidance in Augmented Reality Systems Setup*. Dans IEEE International Conference on Mixed and Augmented Reality (ISMAR'2015), pages 140–143. IEEE, 2015. (Cité en page 72.)
- [Quigley 2009] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler et Andrew Y Ng. *ROS : an open-source Robot Operating System*. Dans ICRA Workshop on Open Source Software, volume 3, page 5. Kobe, Japan, 2009. (Cité en page 28.)
- [Reel 2013] Smarti Reel, Gene Cheung, Patrick Wong et Laurence S Dooley. *Joint Texture-Depth Pixel Inpainting of Disocclusion Holes in Virtual View Synthesis*. Dans Signal and Information Processing Association Annual Summit and Conference (APSIPA'2013), pages 1–7. IEEE, 2013. (Cité en page 80.)
- [Regan 1995] E.C. Regan. *Some Evidence of Adaptation to Immersion in Virtual Reality*. Displays, vol. 16, no. 3, pages 135–139, 1995. (Cité en page 25.)
- [Rekimoto 1996] J. Rekimoto. *Transvision : A Hand-Held Augmented Reality System for Collaborative Design*. Dans Virtual Systems and Multimedia, volume 96, pages 18–20, 1996. (Cité en page 185.)
- [Richardt 2010] C. Richardt, D. Orr, I. Davies, A. Criminisi et N.A. Dodgson. *Real-time Spatiotemporal Stereo Matching using the Dual-Cross-Bilateral Grid*. Dans European Conference on Computer Vision (ECCV'2010), pages 510–523. Springer, 2010. (Cité en page 51.)
- [Robinett 1993] Warren Robinett et Jannick P Rolland. *A Computational Model for the Stereoscopic Optics of a Head-Mounted Display*. Dans Virtual Reality Systems, pages 51–75. Elsevier, 1993. (Cité en page 88.)
- [Rosten 2005] E. Rosten et T. Drummond. *Fusing Points and Lines for High Performance Tracking*. Dans IEEE International Conference on Computer Vision (ICCV'2005), volume 2, pages 1508–1515. IEEE, 2005. (Cité en page 52.)
- [Ruble 2011] Ethan Rublee, Vincent Rabaud, Kurt Konolige et Gary Bradski. *ORB : An Efficient Alternative to SIFT or SURF*. Dans IEEE International Conference on Computer Vision (ICCV'2011), pages 2564–2571. IEEE, 2011. (Cité en page 57.)
- [Saito 2011] Hideo Saito. *Computer Vision for 3DTV and Augmented Reality*. Dans International Symposium on Ubiquitous Virtual Reality (ISUVR'2011), pages 5–8. IEEE, 2011. (Cité en page 73.)
- [Scaramuzza 2006a] D. Scaramuzza, A. Martinelli et R. Siegwart. *A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion*. Dans IEEE International Conference on Computer Vision Systems (ICVS'06), pages 45–45, 2006. (Cité en pages 44 et 141.)
- [Scaramuzza 2006b] D. Scaramuzza, A. Martinelli et R. Siegwart. *A Toolbox for Easily Calibrating Omnidirectional Cameras*. Dans IEEE/RSJ International

- Conference on Intelligent Robots and Systems (IROS'06), pages 5695–5701, 2006. (Cité en pages 45 et 160.)
- [Scaramuzza 2013] D. Scaramuzza. *Camera Calibration Toolbox for MATLAB*, 2013. <https://sites.google.com/site/scarabotix/ocamcalib-toolbox>. (Cité en pages 43 et 45.)
- [Scharstein 2002] D. Scharstein et R. Szeliski. *A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms*. International Journal of Computer Vision, vol. 47, no. 1-3, pages 7–42, 2002. (Cité en pages 49 et 50.)
- [Schowengerdt 2003] B.T. Schowengerdt, E.J. Seibel, J.P. Kelly, N.L. Silverman et T.A. Furness. *Binocular Retinal Scanning Laser Display with Integrated Focus Cues for Ocular Accommodation*. Dans Stereoscopic Displays and Virtual Reality Systems X, volume 5006, pages 1–10. International Society for Optics and Photonics, 2003. (Cité en page 37.)
- [Schubert 2006] E.F. Schubert. Light-emitting diodes. Cambridge University Press, 2 édition, juin 2006. (Cité en pages 8 et 16.)
- [Sewell 2010] Weston Sewell et Oleg Komogortsev. *Real-Time Eye Gaze Tracking with an Unmodified Commodity Webcam Employing a Neural Network*. Dans CHI'10 Extended Abstracts on Human Factors in Computing Systems, pages 3739–3744. ACM, 2010. (Cité en page 65.)
- [Sheela 2011] SV Sheela et PA Vijaya. *Mapping Functions in Gaze Tracking*. International Journal of Computer Applications, vol. 26, no. 3, pages 36–42, 2011. (Cité en page 63.)
- [Shum 2000] Harry Shum et Sing Bing Kang. *Review of IMAGE-BASED RENDERING TECHNIQUES*. Dans Visual Communications and Image Processing 2000, volume 4067, pages 2–14. International Society for Optics and Photonics, 2000. (Cité en page 72.)
- [Smolyanskiy 2018] N. Smolyanskiy, A. Kamenev et S. Birchfield. *On the Importance of Stereo for Accurate Depth Estimation : An Efficient Semi-Supervised Deep Neural Network Approach*. IEEE Computer Vision and Pattern Recognition (CVPR), pages 1007–1015, 2018. (Cité en page 54.)
- [Sola 1997] J Sola et Joaquin Sevilla. *Importance of Input Data Normalization for the Application of Neural Networks to Complex Industrial Problems*. IEEE Transactions on Nuclear Science, vol. 44, no. 3, pages 1464–1468, 1997. (Cité en page 102.)
- [Solh 2012] Mashhour Solh et Ghassan AlRegib. *Hierarchical Hole-Filling for Depth-Based View Synthesis in FTV and 3D Video*. IEEE Journal of Selected Topics in Signal Processing, vol. 6, no. 5, pages 495–504, 2012. (Cité en pages 80 et 157.)
- [Steinicke 2009] Frank Steinicke, Gerd Bruder, Klaus Hinrichs, Scott Kuhl, Markus Lappe et Pete Willemsen. *Judgment of Natural Perspective Projections*

- in Head-Mounted Display Environments*. Dans ACM Symposium on Virtual Reality Software and Technology, pages 35–42. ACM, 2009. (Cité en page 88.)
- [Steptoe 2014] William Steptoe, Simon Julier et Anthony Steed. *Presence and Discernability in Conventional and Non-Photorealistic Immersive Augmented Reality*. Dans IEEE International Conference on Mixed and Augmented Reality (ISMAR'2014), pages 213–218. IEEE, 2014. (Cité en pages 35 et 72.)
- [Strasdat 2010] Hauke Strasdat, JMM Montiel et Andrew J Davison. *Real-Time Monocular SLAM : Why Filter?* Dans IEEE International Conference on Robotics and Automation (ICRA'2010), pages 2657–2664. IEEE, 2010. (Cité en pages 60 et 61.)
- [Sun 2010a] Wenxiu Sun, Lingfeng Xu, Oscar C Au, Sung Him Chui et Chun Wing Kwok. *An Overview of Free View-Point Depth-Image-Based Rendering (DIBR)*. Dans APSIPA Annual Summit and Conference, pages 1023–1030, 2010. (Cité en page 73.)
- [Sun 2010b] Wenxiu Sun, Lingfeng Xu, Oscar C Au, Sung Him Chui et Chun Wing Kwok. *An Overview of Free View-Point Depth-Image-Based Rendering (DIBR)*. Dans APSIPA Annual Summit and Conference, pages 1023–1030, 2010. (Cité en page 78.)
- [Sutherland 1968] I.E. Sutherland. *A Head-Mounted Three Dimensional Display*. Dans ACM Fall Joint Computer Conference, Part I, pages 757–764, 1968. (Cité en pages 56 et 185.)
- [Świrski 2012] Lech Świrski, Andreas Bulling et Neil Dodgson. *Robust Real-Time Pupil Tracking in Highly Off-Axis Images*. Dans Proceedings of the Symposium on Eye Tracking Research and Applications, pages 173–176. ACM, 2012. (Cité en pages 64 et 148.)
- [Swirski 2013] Lech Swirski et Neil Dodgson. *A Fully-Automatic, Temporal Approach to Single Camera, Glint-Free 3d Eye Model Fitting [Abstract]*. Dans Proceedings of ECEM 2013, août 2013. <http://www.cl.cam.ac.uk/research/rainbow/projects/eyemodelfit/>. (Cité en pages 67, 70, 96, 97, 100, 142, 150 et 151.)
- [Taba 2012] Isabella Bahareh Taba. *Improving Eye-Gaze Tracking Accuracy through Personalized Calibration of a User's Aspherical Corneal Model*. PhD thesis, University of British Columbia, 2012. (Cité en page 67.)
- [Tao 2015] M.W. Tao, P.P. Srinivasan, J. Malik, S. Rusinkiewicz et R. Ramamoorthi. *Depth from Shading, Defocus, and Correspondence using Light-Field Angular Coherence*. Dans IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2015), pages 1940–1948, 2015. (Cité en page 53.)
- [Tasman 2006] W. Tasman et E.A. Jaeger. *Duane's ophthalmology on DVD-ROM*, volume 1. Lippincott Williams et Wilkins, 2006. (Cité en pages 8, 11 et 12.)
- [Tatarchenko 2016] Maxim Tatarchenko, Alexey Dosovitskiy et Thomas Brox. *Multi-View 3D Models from Single Images with a Convolutional Network*.

- Dans European Conference on Computer Vision (ECCV/2016), pages 322–337. Springer, 2016. (Cit  en page 81.)
- [Thibos 1992] L.N. Thibos, M. Ye, X. Zhang et A. Bradley. *The Chromatic Eye : A New Reduced-Eye Model of Ocular Chromatic Aberration in Humans*. Applied Optics, vol. 31, no. 19, pages 3594–3600, juillet 1992. (Cit  en page 13.)
- [Thibos 1997] L.N. Thibos, M. Ye, X. Zhang et A. Bradley. *Spherical Aberration of the Reduced Schematic Eye with Elliptical Refracting Surface*. Optometry and Vision Science, vol. 74, no. 7, pages 548–556, juillet 1997. (Cit  en page 13.)
- [Tian 2009] Dong Tian, Po-Lin Lai, Patrick Lopez et Cristina Gomila. *View Synthesis Techniques for 3D Video*. Dans SPIE Optical Engineering+ Applications, pages 74430T–74430T. International Society for Optics and Photonics, 2009. (Cit  en page 79.)
- [Topal 2014] Cihan Topal, Serkan Gunal, Onur Ko deviren, Atakan Dogan et Omer Nezih Gerek. *A Low-Computational Approach on Gaze Estimation with Eye Touch System*. IEEE Transactions on Cybernetics, vol. 44, no. 2, pages 228–239, 2014. (Cit  en pages 69, 70 et 93.)
- [Tuceryan 2002] Mihran Tuceryan, Yakup Genc et Nassir Navab. *Single-Point Active Alignment Method (SPAAM) for Optical See-Through HMD Calibration for Augmented Reality*. Presence : Teleoperators & Virtual Environments, vol. 11, no. 3, pages 259–276, 2002. (Cit  en pages 83 et 84.)
- [Urban 2016] Steffen Urban et Stefan Hinz. *MultiCol-SLAM : A Modular Real-Time Multi-Camera SLAM System*. arXiv preprint arXiv :1610.07336, 2016. (Cit  en page 160.)
- [Vandeportaele 2006] B. Vandeportaele. *Contributions   la Vision Omnidirectionnelle :  tude, Conception et  talonnage de Capteurs pour l’Acquisition d’Images et la Mod lisation 3D*. PhD thesis, Institut National Polytechnique de Toulouse, 2006. (Cit  en page 160.)
- [Vasko 2015] R. Vasko, N. Zeller, F. Quint et U. Stilla. *A Real-Time Depth Estimation Approach for a Focused Plenoptic Camera*. Dans International Symposium on Visual Computing, pages 70–80. Springer, 2015. (Cit  en page 53.)
- [Villanueva 2007] Arantxa Villanueva et Rafael Cabeza. *Models for Gaze Tracking Systems*. Journal on Image and Video Processing, vol. 2007, no. 3, page 4, 2007. (Cit  en page 68.)
- [von Helmholtz 1909] H. von Helmholtz. Handbuch der physiologischen optik, volume 1. Hamburg und Leipzig, 1909. (Cit  en page 12.)
- [Wang 2015a] T.C. Wang, A.A. Efros et R. Ramamoorthi. *Occlusion-aware Depth Estimation using Light-Field Cameras*. Dans IEEE International Conference on Computer Vision (ICCV’2015), pages 3487–3495, 2015. (Cit  en page 53.)

- [Wang 2015b] Wenqiang Wang, Jing Yan, Ningyi Xu, Yu Wang et Feng-Hsiung Hsu. *Real-Time High-Quality Stereo Vision System in FPGA*. IEEE Transactions on Circuits and Systems for Video Technology, vol. 25, no. 10, pages 1696–1708, 2015. (Cité en page 50.)
- [Welch 2002] Greg Welch et Eric Foxlin. *Motion Tracking : No Silver Bullet, but a Respectable Arsenal*. IEEE Computer graphics and Applications, vol. 22, no. 6, pages 24–38, 2002. (Cité en page 56.)
- [Witzner Hansen 2010] D. Witzner Hansen et Q. Ji. *In the Eye of the Beholder : A Survey of Models for Eyes and Gaze*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 3, pages 478–500, mars 2010. (Cité en page 15.)
- [Xie 2016] Junyuan Xie, Ross Girshick et Ali Farhadi. *Deep3d : Fully Automatic 2d-to-3d Video Conversion with Deep Convolutional Neural Networks*. Dans European Conference on Computer Vision (ECCV’2016), pages 842–857. Springer, 2016. (Cité en pages 80, 81, 82 et 145.)
- [Xu 2013] Xuyuan Xu, Lai-Man Po, Chun-Ho Cheung, Litong Feng, Ka-Ho Ng et Kwok-Wai Cheung. *Depth-Aided Exemplar-Based Hole Filling for DIBR View Synthesis*. Dans IEEE International Symposium on Circuits and Systems (ISCAS’2013), pages 2840–2843. IEEE, 2013. (Cité en page 80.)
- [Yoon 2006] K.J. Yoon et I.S. Kweon. *Adaptive Support-Weight Approach for Correspondence Search*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 4, pages 650–656, 2006. (Cité en pages 50 et 51.)
- [Zadeh 1965] Lotfi A Zadeh. *Fuzzy Sets*. Information and Control, vol. 8, no. 3, pages 338–353, 1965. (Cité en page 172.)
- [Zbontar 2016] J. Zbontar et Y. LeCun. *Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches*. The Journal of Machine Learning Research, vol. 17, no. 1, pages 2287–2318, 2016. Code source disponible sur <https://github.com/jzbontar/mc-cnn>. (Cité en page 54.)
- [Zhang 2000] Z. Zhang. *A Flexible New Technique for Camera Calibration*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pages 1330–1334, 2000. (Cité en pages 42 et 45.)
- [Zhao 2013] Yin Zhao, Ce Zhu et Lu Yu. *Virtual View Synthesis and Artifact Reduction Techniques*. Dans 3D-TV System with Depth-Image-Based Rendering, pages 145–167. Springer, 2013. (Cité en page 78.)
- [Zhou 2016] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik et Alexei A Efros. *View Synthesis by Appearance Flow*. Dans European Conference on Computer Vision (ECCV’2016), pages 286–301. Springer, 2016. (Cité en page 81.)
- [Zhu 2007a] Z. Zhu et Q. Ji. *Novel Eye Gaze Tracking Techniques under Natural Head Movement*. IEEE Transactions on Biomedical Engineering, vol. 54, no. 12, pages 2246–2260, décembre 2007. (Cité en page 15.)

- [Zhu 2007b] Zhiwei Zhu et Qiang Ji. *Novel Eye Gaze Tracking Techniques under Natural Head Movement*. IEEE Transactions on Biomedical Engineering, vol. 54, no. 12, pages 2246–2260, 2007. (Cité en page 66.)
- [Zinger 2010] Sveta Zinger, Luat Do et PHN de With. *Free-Viewpoint Depth Image Based Rendering*. Journal of Visual Communication and Image Representation, vol. 21, no. 5, pages 533–541, 2010. (Cité en pages 79 et 80.)
- [Álvarez 2017] Hugo Álvarez, Jon Arrieta et David Oyarzun. *Towards a Diminished Reality System that Preserves Structures and Works in Real-time*. Dans International Conference on Computer Vision Theory and Applications (VI-SAPP'2017), pages 334–343. SCITEPRESS, 2017. (Cité en page 144.)