École Pratique
des Hautes Études

Mention « Systèmes intégrés, environnement, biodiversité »

École doctorale de l'École Pratique des Hautes Études
Laboratoire CHArt (Cognitions Humaine et Artificielle)

Oculométrie Numérique Economique: modèle d'apparence et apprentissage par variétés

Par Ke LIANG

Thèse de doctorat d'Informatique, Mathématique et Application

Sous la direction de M. François JOUEN, directeur d'études

Soutenue le 13 mai 2015

Devant un jury composé de:

Edwige PISSALOUX	Professeur	Université Paris 6	
Stéphane CANU	Professeur	INSA de Rouen	
Marc BUI	Professeur	Université Paris 8	
Charles TIJUS	Professeur	Université Paris 8	
Youssef CHAHIR	Professeur	Université de Caen	
François JOUEN	Professeur	EPHE Paris	(Directeur de thèse)

L'aube adoucie par le vent et la rosée
Entre les rideaux une personne se lève
Le loriot siffle les fleurs sourient
Ce printemps pour qui donc est-il

«Tôt levé» Li Shang Yin (813 - 859)

Remerciements

Je tiens à remercier le directeur de cette thèse, François JOUEN, pour m'avoir encadré pendant cette thèse avec toute confiance. Je lui suis reconnaissant de m'avoir guidé, conseillé, encouragé tout en me laissant travailler très librement. Sa passion, son intelligence, son ouverture d'esprit et sa curiosité scientifique auront fortement marqué mes années de thèse.

Je remercie vivement Mme Edwige PISSALOUX et M. Stéphane CANU d'avoir accepté de porter leur regard expert sur ce manuscrit. Je remercie également les autres membres du jury de cette thèse : M. Marc BUI et M. Charles TIJUS.

Je remercie chaleureusement Youssef CHAHIR pour l'aide et le conseil scientifique qu'il m'a apportés ainsi que pour son extrême gentillesse en toute situation. Ma pensée va également aux autres membres du laboratoire CHArt : Isabelle CARCHON pour son soutien tant scientifique qu'amical, Joëlle PROVASI pour sa motivation et son encouragement. Je remercie Karim, Pierre et Céline au bureau pour leur soutien et amitié.

Je remercie également les personnes de la société Ubiquiet qui m'a financé pendant ma thèse : Dominique, Véranith, Stéphane et Maxime. Je remercie les personnes du musée Tatihou qui ont participé à nos expériences : Mathilde, Ludovic, Hélène et Frédérik. Ils ont été essentiels à la réalisation de l'expérimentation de cette thèse.

Enfin, je remercie ma famille et mes proches pour leur soutien. Mes meilleurs sentiments et ma reconnaissance vont à ceux qui sont toujours là pour moi : ma mère, mon père et Stéphane.

Table des matières

Première partie

Introduction générale

Contexte et problématique

L'oculomètre (Eye tracker) est un système destiné à suivre la direction du regard, à enregistrer et à analyser les mouvement oculaires tels que les saccades, le trémor, etc. Lorsqu'il est apparu à la fin du 19ème siècle, ce système a été conçu de façon mécanique avec l'objectif d'enregistrer les mouvements oculaires au cours de la lecture. Les chercheurs ont ainsi découvert le mouvement saccadique de l'œil, puis les autres différents types de mouvements oculaires comme les vergences, les fixations, etc. Ces travaux ont incité les chercheurs à réaliser des expérimentations dans des domaines variés. Le système oculométrique continue à évoluer encore aujourd'hui pour s'adapter à des domaines de recherche de plus en plus élargis. L'oculomètre peut être utilisé non seulement dans les domaines médicaux, les neurosciences, la psychologie cognitive, mais aussi être appliqué à l'IHM (Interaction Homme-Machine), le marketing, ou les jeux vidéo, etc.

Les techniques pour enregistrer les mouvements oculaires se sont beaucoup développées depuis ces 30 dernières années, et elles deviennent plus complexes et aussi plus performantes. Il existe plusieurs approches pour estimer la direction du regard. L'approche la plus utilisée est celle qui se fonde sur le traitement des caractéristiques des yeux (section 2.3.1) comme la pupille, l'iris, etc. L'oculomètre, tel qu'il est souvent commercialisé, dispose de caméras numériques et utilise la lumière infra-rouge pour détecter la pupille et le reflet de la lumière sur la surface de la cornée. Le moyen le plus utilisé pour estimer la direction du regard est de calculer précisément le vecteur de l'axe visuel par la position du centre de la pupille, et la position du centre de la courbure cornéenne. Le point du regard (PoR) est déterminé comme étant le point de l'intersection de ce vecteur et le plan de la scène (par exemple un écran). Cette technique basée sur le traitement des caractéristiques des yeux fonctionne très bien dans les oculomètres commerciaux, comme les appareils de Tobii, SMI, etc. Mais dans cette approche, la qualité des composants conditionne la performance du dispositif. Pour capturer la pupille et le reflet qui sont de petite taille, les caméras haut de gamme sont indispensables pour enregistrer les détails avec précision. De plus, la façon géométrique de calculer l'axe visuel est sensible aux changements de position des composants.

Le projet ONE (Oculométrie Numérique Economique), vise à développer un système oculométrique à distance avec une webcam et sans utilisation de lumière infra-rouge. Ce système permet de choisir librement les matériels, mais requiert l'efficacité et la robustesse des méthodes de traitement sur les images bruitées où manquent les détails comme l'iris, etc. Cette approche alternative est fondée sur un modèle d'apparence des images des yeux. Une des différences entre les deux approches réside dans la nature de la source de l'image capturée par la caméra, comme le montrent la Figure 1.1 et la Figure 1.2. La technique basée sur le traitement des caractéristiques des yeux a besoin d'identifier la position de la pupille et des reflets dans les images de haute résolution afin de calculer la direction du regard. Mais quand on utilise une caméra normale, les images obtenues nous obligent à trouver une autre solution pour extraire les caractéristiques des yeux, en utilisant des méthodes comme le filtrage d'image, la méthode de la réduction de la dimensionnalité, etc. L'analyse des mouvements oculaires sans extraire les caractéristiques explicites des yeux reste à ce jour un réel défi. La localisation des yeux dans une scène naturelle et complexe est aussi une des problématiques essentielles dans le domaine de la vision par ordinateur.

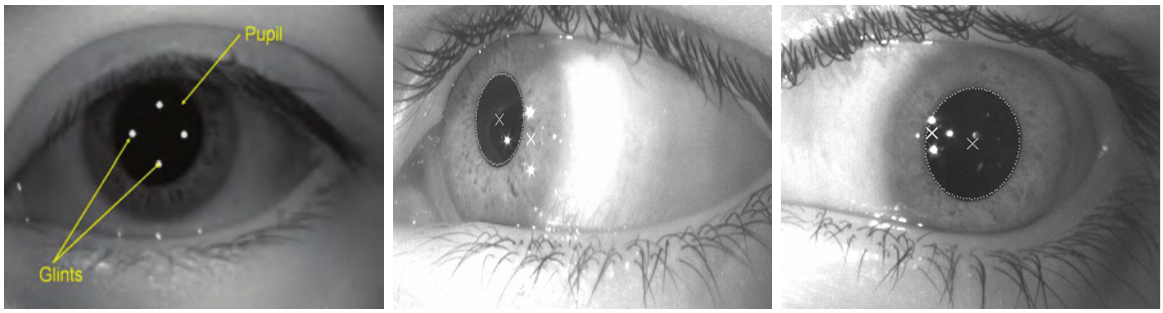


FIGURE 1.1 – La méthode basée sur le traitement des caractéristiques estime la direction du regard en calculant le vecteur Pupille-Reflets, c'est-à-dire en identifiant la position de la pupille et des reflets des sources infra-rouges. (source : [Hong et al., 2007])

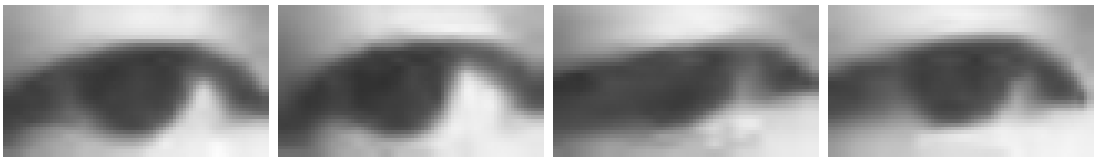


FIGURE 1.2 – Les images des yeux localisées dans les images capturées par une webcam représentent les mouvements des yeux. L'apparence de l'image diffère selon les sujets et les conditions d'illumination. La technique présentée dans la Figure 1.1 n'est pas adéquate pour ces images.

Organisation et contributions

L'objectif de la thèse est de développer un système complet et performant qui permet d'enregistrer les points du regard et de fournir une précision pertinente en temps réel ou "off-line" pour diverses applications d'oculométrie. Vu le problème d'extraction des caractérististiques des yeux à partir d'images bruitées de la webcam, nous utilisons un modèle d'apparence pour non seulement réduire la dimensionnalité de l'image de l'œil de façon locale pour le calcul, mais également pour préserver les caractéristiques de l'apparence de l'image. Ces caractérististiques traitées par ce modèle peuvent être utilisées pour la localisation de l'œil et l'estimation du regard. Pour améliorer la précision du système proposé, nous proposons aussi l'apprentissage par variété sur un ensemble d'images de l'œil pour rendre compte de la diversité des mouvements oculaires présente dans ces images.

Pour la conception et la réalisation, ce système se compose de 2 ensembles : localisation des yeux dans la séquence d'images capturées par la webcam et estimation du regard à partir de l'apparence de l'image des yeux. Concrètement notre travail consiste à mettre en œuvre 4 modules : extraction des caractéristiques, détection et suivi des yeux, apprentissage par variété et estimation du regard par apprentissage automatique.

Les contributions de la thèse sont les suivantes :

- Proposition d'une méthode d'extraction des caractéristiques des yeux avec les motifs binaires locaux centrés-symétriques (CS-LBP). Les images capturées par la webcam nous fournissent moins de détails sur la région des yeux que celles dans la

Figure 1.1. La méthode proposée est basée sur un modèle d'apparence de l'image et permet de caractériser le motif local autour de chaque pixel de l'image. Nous divisons l'image des yeux en blocs et combinons les histogrammes CS-LBP pour former un vecteur de signature de faible dimension. Ce vecteur de caractéristique est facile à calculer et résiste au changement d'illumination. Il est utilisé non seulement pour discriminer l'œil des autres objets pendant la phase de localisation des yeux, mais également pour distinguer les différents mouvements oculaires pendant la phase d'estimation du regard.

- Proposition d'une méthode hybride pour détecter et suivre les yeux dans la séquence d'images capturées par la webcam. Dans un premier temps, un modèle à formes actives (ASM) et une carte des yeux (EyeMap) sont appliqués à la première image pour localiser les composantes faciales, notamment les yeux. Après la localisation de l'œil, on utilise un filtre particulaire pour suivre, de façon stochastique, le déplacement de l'œil dans les images suivantes. Cette méthode permet de détecter et de suivre les yeux plus efficacement et de rélocaliser rapidement les yeux quand ils ont été perdus à cause des mouvements du sujet.

- Introduction de la technique de réduction de la dimensionnalité non-linéaire pour analyser les mouvements oculaires selon les variétés (manifolds) sur un ensemble d'images de l'œil. Considérant que cet ensemble d'images de l'œil se compose de données (images) de haute dimension, le Laplacian Eigenmaps, une méthode fondée sur le laplacien du graphe, permet de trouver une représentation de faible dimension qui préserve les propriétés locales des données. Appliquer cette approche sur l'ensemble des images de l'œil nous permet de déterminer la structure intrinsèque qui décrit la variation des mouvements oculaires. Concrètement l'analyse de la variété contribue à réaliser une calibration automatique qui est cruciale pour le fonctionnement du module d'estimation du regard.

- L'estimation du regard est considérée comme un problème d'apprentissage supervisé. La phase de la calibration permet de générer des exemples et de sélectionner des échantillons corrects pour estimer le regard avec précision. Dans la réalisation du système, nous proposons deux modèles d'estimation du regard :

  - le modèle de prédiction en *régression* estime le regard en coordonnées 2D, dont les valeurs sont un ensemble continu de réels $\mathbb{R}$. Nous proposons une sélection semi-supervisée pour construire un ensemble de données d'apprentissage, et le processus gaussien pour trouver la fonction f qui permet d'estimer la position du regard.
  - le modèle de prédiction en *catégorisation* permet de classifier le regard dans les classes définies de certains types de mouvements oculaires, par exemple, le clignement et les fixations sur les différentes régions à l'écran : en haut à droite, en bas à droite, en haut à gauche, en bas à gauche.

Cette thèse se compose de quatre chapitres qui reprennent les points précédemment évoqués. La partie suivante (**Chapitre 2**) présente une synthèse concernant le système visuel humain, et l'état de l'art de la technique oculométrique. Le **Chapitre 3** présente nos méthodes proposées pour localiser la region de l'œil dans la séquence d'images capturée par la webcam. Ce chapitre se compose de 3 sections : la détection, l'extraction

des caractéristiques et le suivi de l'œil. Le **Chapitre 4** présente les différentes techniques linéaires et non-linéaires de réduction de la dimensionnalité, et l'application du Laplacian Eigenmaps pour générer la variété sur un ensemble d'images de l'œil, qui permettent d'analyser la variation des mouvements oculaires. Ensuite, dans le **Chapitre 5**, nous présentons les méthodes pour estimer le regard et les applications que nous avons réalisées.

Deuxième partie

L'oculométrie

Sommaire

2.1 Le système visuel humain

2.1.1 Le système visuel

Le *système visuel* est constitué de l'ensemble des structures physiologiques qui participent au recueil et au traitement des informations lumineuses pour élaborer des perceptions [Bonnet et al., 1989]. L'œil est le capteur des informations visuelles qui subissent dans les structures rétiniennes leurs premiers traitements. Le système visuel humain, comme celui des autres mammifères, se compose de "modules" interconnectés. Un élément important du système visuel est la rétine qui est constituée par l'ensemble des cellules qui répondent à des stimulations visuelles, dont l'activité est modifiable par la présentation des stimulations lumineuses à l'œil. Les axones des cellules ganglionnaires de la rétine se réunissent pour former le nerf optique. Les deux nerfs optiques se croisent pour moitié au niveau du chiasma optique. Une grande partie des fibres se projette sur une structure thalamique appelée le Corps Géniculé Latéral. De là, partent les fibres qui vont se projeter dans les aires corticales occipitales. La figure 2.1 illustre l'ensemble du système visuel.

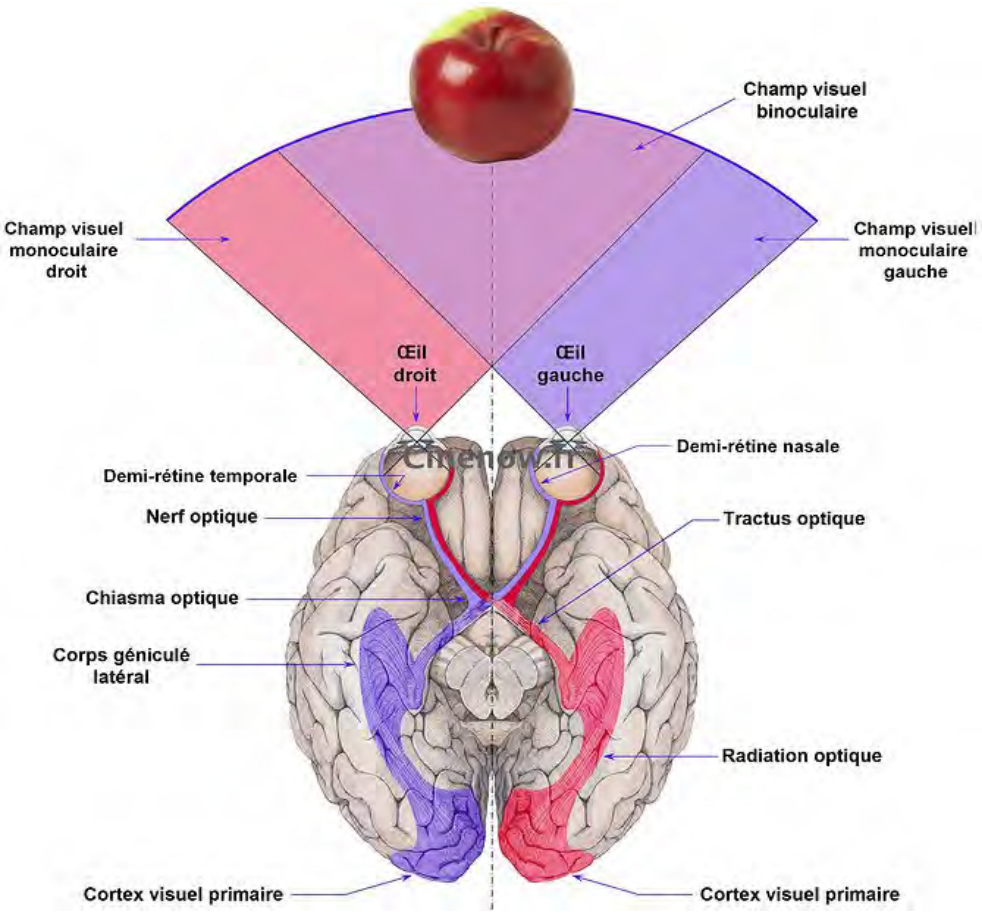


FIGURE 2.1 – Schéma des deux champs visuels et coupe schématique horizontale du cerveau, montrant les projections des nerfs optiques sur le Corps Géniculé Latéral, et du Corps Géniculé Latéral vers le cortex visuel primaire. (source d'image : http ://www.cinenow.fr/tutorials/2110-de-loeil-au-cerveau-des-chemins-qui-se-croisent)

2.1.2 L'anatomie de l'œil

L'œil, "capteur" des informations visuelles, est d'abord un instrument d'optique qui permet la formation d'une image sur la rétine. Il est mobile grâce à 6 muscles extraoculaires et est constitué d'une vingtaine de structures, toutes essentielles pour voir correctement : rétine, iris, cornée, cristallin, etc. Il nous permet de recevoir et de transformer les vibrations électromagnétiques de la lumière en influx nerveux qui sont transmis au cerveau. L'œil humain est un globe de 2,2 à 2,5 centimètres de diamètre (10% de différence entre les adultes [Hammoud, 2008]).

Vus de face, les yeux apparaissent généralement comme un disque sombre (la pupille), entouré par un anneau de couleur (l'iris), avec une occlusion partielle en haut et en bas par les paupières, avec une forme triangulaire de la sclérotique blanche visible sur un ou deux côtés (Figure 2.2).

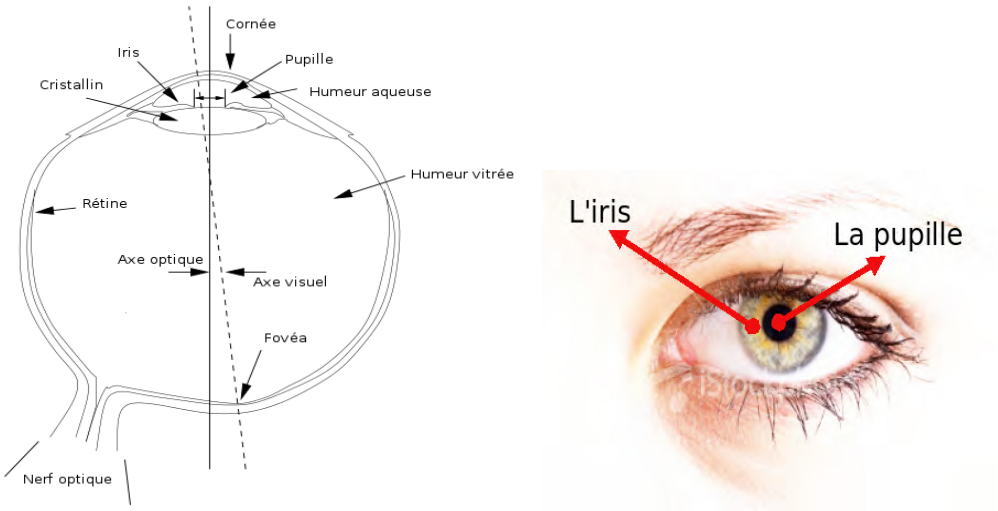


FIGURE 2.2 – Le globe de l'œil humain et les composants principaux.

La figure 2.2 (gauche) représente le schéma horizontal du globe oculaire humain, où nous pouvons trouver les principaux composants suivants :

- **La cornée** : La cornée est une membrane solide et transparente d'environ 11 mm de diamètre, au travers de laquelle la lumière entre à l'intérieur de l'œil. La cornée est nourrie par un liquide : **l'humeur aqueuse**. La cornée contient 78% d'eau et pour maintenir ce degré d'hydrophilie, elle est constamment recouverte de larmes alimentées en continu par les glandes lacrymales et répartie par le battement des paupières. La cornée assure environ 80% de la réfraction.
- **La sclérotique** : La sclérotique est une enveloppe de protection et recouvre environ les cinq sixièmes de la surface de l'œil. Elle donne à l'œil sa couleur blanche et sa rigidité.
- **L'iris** : Il s'agit du diaphragme de l'œil percé en son centre par la pupille. C'est un muscle qui fait varier l'ouverture de la pupille (entre 2,5 et 7 mm) afin de modifier la quantité de lumière qui pénètre dans l'œil pour éviter l'aveuglement en plein soleil ou capter le peu de lumière la nuit.
- **La pupille** : Il s'agit d'un trou au centre de l'iris permettant de faire passer les rayons lumineux vers la rétine.

- **Le cristallin** : Le cristallin est une lentille auxiliaire molle et composée de fines couches superposées. Il se déforme sous l'action des muscles ciliaires. Il se situe derrière l'iris et permet que l'œil adapte sa puissance optique à sa distance de vision, pour avoir une vision nette. Ce phénomène s'appelle l'accommodation.
- **La rétine** : La rétine est organisée selon le schéma de la figure 2.3 : les photorécepteurs recoivent la lumière ; les cellules bipolaires relaient les informations vers les ganglionnaires ; les cellules horizontales et amacrines assurent un contact transversal ; les cellules ganglionnaires, enfin, envoient les signaux vers le cerveau. La rétine possède 2 types de photorécepteurs, qui nous permettent de percevoir les couleurs et les formes : les cônes et les bâtonnets. Les bâtonnets sont environ 130 millions. Ils sont absents de la fovéa et se situent sur toute la périphérie. Ils ont une très grande sensibilité à la lumière, mais ils ne permettent pas la vision des détails. Les cônes sont entre 5 à 7 millions localisés principalement dans la fovéa : ils permettent la perception des détails.

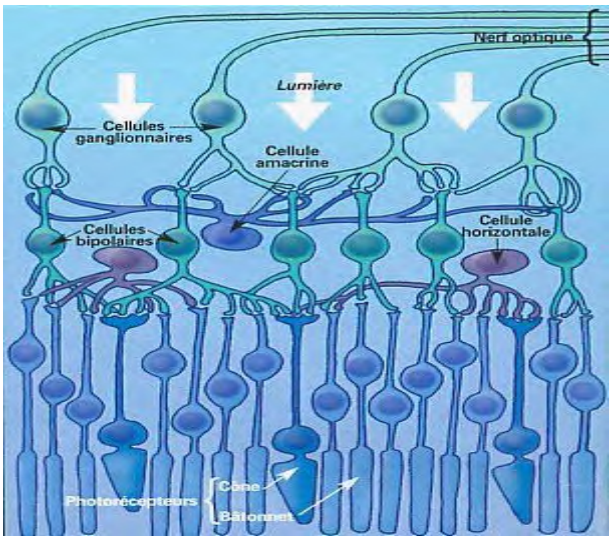


FIGURE 2.3 – La structure de la rétine. De bas en haut, on observe les récepteurs, cônes et bâtonnets, interconnectés par des cellules bipolaires. Celles-ci sont connectées à des cellules ganglionnaires. On notera la présence de connextions horizontales (cellules amacrines). Les axones des cellules ganglionnaires forment le nerf optique. (source d'image : [Imbert, 2001])

- **La fovéa** : La fovéa est la zone de la rétine où la vision des détails est la plus précise. La fovéa ne possède que de cônes qui nous permettent d'avoir la meilleure résolution optique. Elle s'étend sur un diamètre de 300 microns environ. Bien que n'occupant que quelques pourcents de la surface de la rétine, cette zone transmet 90% de l'information visuelle traitée par le cerveau.
- **L'humeur vitrée** : L'humeur vitrée occupe 80% du volume de l'œil, elle est constituée d'une gelée (acide hyaluronique) qui donne à l'œil sa consistance.

Nous pouvons définir deux axes principaux par lesquels la lumière arrive dans l'œil : **l'axe optique** (optical axis) et **l'axe visuel** (visual axis) comme indiqués sur la figure 2.2. La plupart des systèmes d'oculométrie basés sur un modèle 3D calculent la direction du regard à l'aide de l'axe optique ou de l'axe visuel. L'axe optique est la ligne directe qui passe par le centre de la cornée, puis par la pupille, et ensuite par le cristallin (la lentille de l'œil). La partie de la rétine qui est dans le prolongement de l'axe optique permet

de percevoir les éléments de notre environnement avec beaucoup de détails. Pourtant, ce n'est pas cet endroit de la rétine qui est le plus sensible à la lumière et à la couleur. Comme nous l'avons présenté ci-dessus, la partie de la rétine qui est la plus sensible à la lumière et à la couleur se trouve sur la zone fovéale. L'axe qui va du centre de la pupille vers la fovéa ne donne pas une image aussi nette que l'axe optique, parce que cet axe ne passe pas par le centre exact de la cornée et du cristallin, mais c'est cet axe qui donne la meilleure perception de la couleur. Cet axe s'appelle l'axe visuel. Selon la définition théorique l'axe visuel est celui qui relie le point de fixation au premier point nodal, et le deuxième point nodal à la fovéa.

Si un objet intervient dans le champ visuel périphérique, l'axe optique de nos yeux va pivoter dans cette direction afin d'amener l'image de cette partie de scène sur la fovéa. En pratique, il est difficile d'observer la position exacte de la fovéa par le système d'oculométrie. En conséquence l'axe visuel ne peut pas être déterminé facilement. D'après le modèle des yeux, l'axe visuel est juste au-dessus de l'axe optique avec un certain angle dépendant de la forme de l'œil du sujet. La plupart des systèmes d'oculométrie calcule l'axe optique dans un premier temps puis compense la différence pour trouver l'axe visuel afin d'estimer le regard [Hammoud, 2008][Hansen and Ji, 2010].

2.1.3 Types des mouvements oculaires

Nos deux yeux bougent sans cesse, et le plus souvent ensemble. Grâce à leurs mouvements, mais également ceux de notre tête et de tout notre corps, nous avons un accès privilégié au monde qui nous entoure, nous pouvons l'explorer visuellement. D'autre part, étudier le mouvement des yeux est un outil privilégié pour la compréhension des mécanismes de perception et d'action orchestrés par le cerveau.

Six muscles extra-oculaires (ou muscles oculomoteurs extrinsèques) (Figure 2.4), c'est-à-dire trois paires de muscles complémentaires, permettent à chaque œil de se mouvoir dans leur orbite par rapport à la tête (coordonnées craniotopiques). Ils sont innervés par des motoneurones qui génèrent les signaux moteurs permettant d'atteindre rapidement la position désirée et de s'y maintenir pendant la fixation. L'anatomie des muscles et leurs propriétés biomécaniques font actuellement l'objet de recherches et nous n'entrerons pas plus en détail dans la description des muscles.

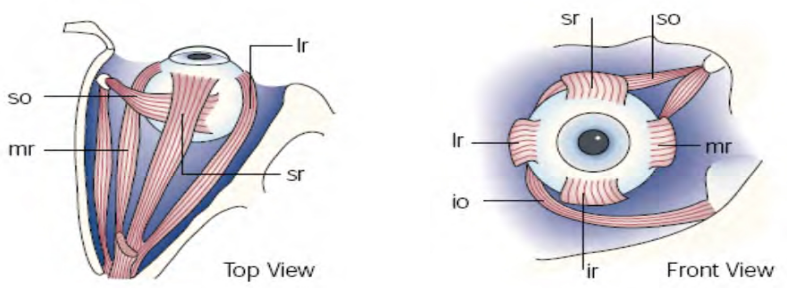


FIGURE 2.4 – Muscles extra-oculaires. mr : droit médial (medial rectus). lr : droit latéral (lateral rectus). sr : droit supérieur (superior rectus). ir : droit inférieur (inferior rectus). so : oblique supérieur (superior oblique). io : oblique inférieur (inferior oblique). Source : [Sparks, 2002].

Les différents types de mouvements oculaires peuvent être classés d'après leur mode d'initiation (mouvements volontaires ou mouvements réflexes), leurs caractéristiques physiques (mouvements lents ou mouvements rapides), la direction du déplacement d'un œil par rapport à l'autre (mouvements de version changeant la direction du regard et mouvements de vergence changeant la profondeur du regard). Selon leurs caractéristiques fonctionnelles, les différents mouvements oculaires permettent d'apporter l'objet d'intérêt au centre de la rétine (saccades, vergences), de garder un objet mobile sur la fovéa (poursuite), de réagir par réflexe à un mouvement du monde visuel par déplacement du monde et/ou déplacement de l'observateur (réflexe optocinétique et réflexe vestibulo-oculaire) ou encore de prévenir la disparition de l'image pendant la fixation (micro-mouvements de fixations).

- **Réflexe vestibulo-oculaire** : Le réflexe vestibulo-oculaire permet de maintenir l'axe visuel constant pendant les mouvements de la tête et du corps. Le mouvement de la tête entraîne un mouvement compensatoire de l'œil dans l'orbite, d'amplitude égale et de direction opposée à celle du mouvement de la tête. Ce réflexe est issu des analyseurs labyrinthiques des canaux semi-circulaires détectant l'accélération angulaire de la tête. Sa latence est très courte, d'environ 10 ms. Ce réflexe est fondamental pour conserver une vision claire quand on se déplace. La plupart des mouvements naturels de la tête sont petits et entraînent de petits mouvements oculaires compensateurs. Des mouvements amples de la tête ou une rotation prolongée du corps provoque un nystagmus. Le nystagmus est un enchaînement rythmique rapide d'une phase lente, corrigeant approximativement la rotation du corps, et de phases rapides (saccades) dirigées dans le sens opposé. Les phases rapides permettent de recentrer l'œil dans l'orbite et de ne pas atteindre les limites mécaniques des positions trop excentrées. Bien que le mode d'incitation soit différent, leur fonction ne diffère pas fondamentalement pas de celle des saccades volontaires.

- **Réflexe optocinétique** : Le réflexe optocinétique provoque des mouvements similaires à celui du réflexe vestibulo-oculaire. Le réflexe vestibulo-oculaire est déclenché directement par le mouvement de la tête et du corps ; le réflexe optocinétique est déclenché par le glissement des images sur la rétine. Ce glissement existe quand le corps se déplace dans le monde mais aussi quand le monde se déplace par rapport au corps, par exemple quand nous regardons le paysage depuis un train en marche. Les systèmes vestibulo-oculaire et optocinétique sont complémentaires.

- **Vergence** : Les vergences sont les mouvements disjonctifs, pendant lesquels les déplacements des deux yeux s'effectuent différemment. Les deux principaux sont la convergence et la divergence. La divergence éloigne le point de fixation ; l'angle de vergence entre les deux axes optiques diminue. La convergence rapproche le point de fixation ; l'angle de vergence entre les deux axes optiques augmente. La divergence et convergence peuvent être déclenchées par plusieurs stimuli. Les principaux sont les disparités, c'est-à-dire les défauts de correspondance rétinienne, le flou rétinien, qui déclenche l'accommodation et une vergence associée ou encore la sensation de proximité.

- **Saccade** : Le terme de saccade désigne tout mouvement rapide permettant de changer la direction du regard. La fonction de la saccade est directement liée à la

présence de la fovéa. Une saccade sert principalement à apporter l'objet d'intérêt au centre de la rétine, mais des saccades existent également chez des espèces dépourvues de fovéas. Les saccades peuvent collaborer avec d'autres systèmes, par exemple lors du nystagmus optocinétique et vestibulo-oculaire, elles sont dans ce cas extrêmement réflexes. Elles peuvent être volontaires, quand nous choisissons de regarder une nouvelle cible d'intérêt. Les saccades sont des mouvements très rapides (leur vitesse peut atteindre 500°/s), qui durent généralement moins de 100 ms. Durant les saccades, aucune information visuelle n'est traitée et les informations n'ont donc pas le temps d'influencer le mouvement une fois qu'il est initié.

- **Poursuite** : La poursuite oculaire permet une vision claire et continue d'objets se déplaçant dans un environnement stable. La vitesse de l'objet mobile est détectée et déclenche des mouvements oculaires dont la vitesse est approximativement celle du mouvement de l'objet sur la rétine. La poursuite peut être entrecoupée de saccades de rattrapage (catch up saccades) qui permettent d'annuler le retard éventuellement accumulé. Lorsque la poursuite est réalisée avec des mouvements de la tête, le système de poursuite peut participer à l'annulation du réflexe vestibulo-oculaire. Le système de poursuite pourrait également participer à la fixation oculomotrice, en éliminant les mouvements involontaires de dérive des yeux. Le glissement rétinien n'est pas le seul signal à pouvoir stimuler la poursuite : d'autres systèmes sensoriels ou des représentations internes du mouvement d'une cible dans l'espace peuvent déclencher une poursuite.

- **Fixation** : Les fixations correspondent à des moments pendant lesquels l'œil reste relativement immobile et le système visuel extrait des informations détaillées autour du point de fixation. Il ne s'agit pas à proprement parler de mouvement oculaire puisque la fixation est l'activité des yeux lorsqu'il restent plus ou moins longtemps positionnés sur le même point. Il s'agit donc de l'activité des yeux lorsque ceux-ci ne bougent pas. A ce moment la cible d'intérêt de l'environnement est reflétée sur la fovéa des deux yeux et peut donc être analysée avec un maximum de discrimination spatiale [Yarbus, 1967][Vurpillot, 1991]. La fonction de maintenance de la fixation sur la fovéa est le type le plus complexe de coordination des mouvements oculaires en interaction constante avec tous les autres systèmes optocinétiques. Pendant une fixation, les yeux continuent de bouger selon trois types de mouvements : la dérive, le tremor et les micro-saccades involontaires.

  - **La dérive** (drift) s'observe comme un déplacement lent et irrégulier des axes optiques indépendamment l'un de l'autre. Néanmoins, on remarque que la dérive des yeux n'est pas complètement aléatoire puisque la cible de la fixation est toujours gardée sur la fovéa.

  - **Le tremor** (tremblement), est un mouvement difficile à observer puisqu'il s'agit d'un mouvement constant et indépendant des yeux, de faible amplitude [1] (20-40 secondes d'angle visuel) mais de haute fréquence (70-90 Hz).

1. En physique classique, l'amplitude désigne la mesure scalaire d'un nombre positif caractérisant l'ampleur des variations d'une grandeur. En oculométrie, l'amplitude est la distance entre 2 points du regard, et elle est mesurée souvent par le degré d'angle visuel (Annexe 1).

– **Les micro-saccades**, sont quant à elles de minuscules saccades, identiques pour les deux yeux. Elles peuvent avoir une dimension minimale de 2-5 minutes d'angle et surviennent de façon involontaire.

La condition nécessaire pour l'analyse des mouvements oculaires, surtout dans le cadre de la conception d'un oculomètre, est l'identification des mouvements comme la fixation, les saccades et la poursuite [Duchowski, 2006]. Il est admis que ces mouvements témoignent de l'attention visuelle volontaire. Les fixations correspondent naturellement au mouvement volontaire pour maintenir le regard sur un objet d'intérêt. De même, la poursuite est utilisée de manière volontaire pour suivre les objets en mouvement. Les saccades sont considérées comme des manifestations de la volonté pour changer le point de fixation afin de modifier notre attention.

2.2 Techniques oculométriques

L'oculométrie est une technologie qui a été étudiée et développée depuis plus de 100 ans. Son émergence, à partir de 1879, est lié aux travaux de Louis Emile Javal sur le mouvement oculaire. Cet auteur a observé que la lecture ne comporte pas un seul balayage des yeux mais une série de mouvements oculaires particuliers : les fixations et les saccades rapides. Cette découverte a soulevé d'importantes questions sur la lecture qui ont été étudiées au début du XX^{e} siècle. A partir de 1930, l'oculométrie commence à être utilisée pour la recherche expérimentale notamment en psychologie. Puis grâce au développement de l'informatique, à partir des années 60, nous avons découvert des usages potentiels dans de nombreuse applications comme les sciences cognitives, l'interaction homme-machine, le diagnostic des maladies, le marketing, etc.

Les premiers systèmes oculométriques, comme celui de [Huey, 1898] ou celui de [Delabarre, 1898] sont mécaniques et donc très inconfortables pour les participants. [Huey, 1898] utilisait une barre de maintien avec la bouche du participant pour que la tête reste immobile. [Delabarre, 1898] anesthésiait l'œil avec une solution à 2 ou 3% de cocaine, puis mettait un anneau sur l'œil qui était connecté à un dispositif mecanique. Au début du 20ème siècle, [Dodge et al., 1901] ont introduit une méthode dont le principe est beaucoup moins invasif que celui de la méthode mécanique, qui consiste à photographier la réflexion de la lumière sur l'œil. Ces 30 dernières années, cette méthode utilisant le reflet cornéen a été beaucoup étudiée et améliorée. Elle est devenue une technique principale très présente dans les systèmes oculométriques actuels.

Dans les sections suivantes, nous allons présenter les différentes techniques et leurs évolutions. Selon la technique employée, on peut classer les systèmes en trois catégories : les systèmes à contact, les systèmes électro-oculographiques et les systèmes photo-oculographique et vidéo-oculographique.

2.2.1 Les systèmes à contact

Les systèmes à contact utilisent souvent des dispositifs qui sont fixés sur l'œil pour analyser les mouvements oculaires. Ils peuvent être plus précis que les autres types d'oculomètre, bien qu'ils soient très invasifs. Le premier appareil capable de mesurer

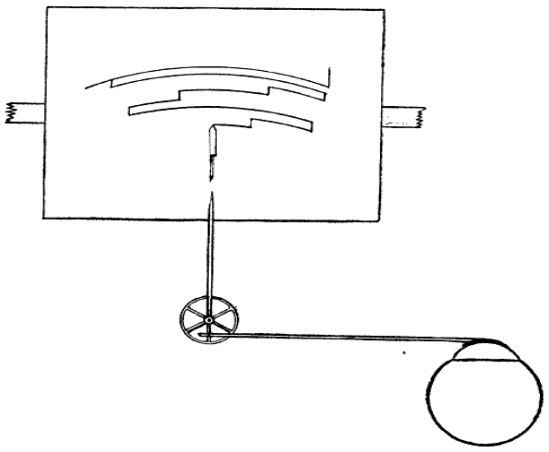


Figure 2.5 – L'appareil mécanique de l'enregistrement des mouvements oculaires conçu par E.B.Huey en 1898.

les mouvements des yeux a été développé par Edmund Burke Huey[Huey, 1898] (Figure 2.5). Son appareil consistait en une membrane très fine trouée au centre qui est placée directement sur l'œil comme une lentille de contact. Cette membrane était attachée à un levier dont l'autre extrémité actionnait une aiguille qui reposait sur un cylindre en rotation. Chaque mouvement actionnait l'aiguille qui enregistrait les traces. Huey s'est intéressé à la manière de lire des textes. Il concluait que lors d'une fixation, les gens lisent plusieurs lettres. Son expérience a confirmé la thèse de Javal : le mouvement des yeux est saccadique. Mais le problème de cet appareil est que l'installation du dispositif sur l'œil était très complexe. Le système a été ensuite amélioré par l'utilisation d'un petit miroir intégré dans un couvercle sur l'œil pour enregistrer la reflexion de la lumière sur les yeux au lieu d'utiliser un levier mécanique.

L'un des fondateurs de la recherche moderne sur les mouvements oculaires est Alfred Lukyanovich Yarbus. Après de nombreuses années de recherches, il a développé un dispositif précis d'enregistrement des mouvements oculaires, et l'a utilisé dans le cadre de différentes disciplines comme les neurosciences, la psychologie expérimentale, l'intelligence artificielle. Son travail sur le sujet comporte plus d'une vingtaine d'articles réalisés à l'Institut de Biophysique de l'Académie des Sciences de l'URSS entre 1954 et 1962. Le système de Yarbus est fondé sur le principe du levier optique et utilise un miroir qui tourne avec l'œil et reflète une source de lumière fixe. La lumière réfléchie est interceptée et enregistrée par un appareil photographique « photokymographe » (Figure 2.6). Le miroir est fixé sur l'œil par un système de ventouse positionné sur la sclérotique. Bien que ce système soit assez léger, de l'ordre de quelques centigrammes, le sujet est obligé de maintenir son œil ouvert à l'aide de ruban adhésif et de plus, la tête doit rester immobile durant toute l'expérimentation. Ceci impose une réduction de la durée expérimentale à environ cinq minutes et cause un certain inconfort au sujet.

Une autre technique qui a été souvent proposée est l'utilisation de la lentille de contact sclérale. Elle se fonde sur l'installation d'une lentille qui doit être collée à l'œil et utilise une bobine d'induction à l'intérieur de la lentille (Figure 2.7). La position de l'œil est déterminée grâce au champ électromagnétique autour de la tête du sujet. Malgré sa précision, cette technique nécessite une bonne dextérité quant à la manipulation des lentilles et le port de la lentille peut provoquer un malaise.

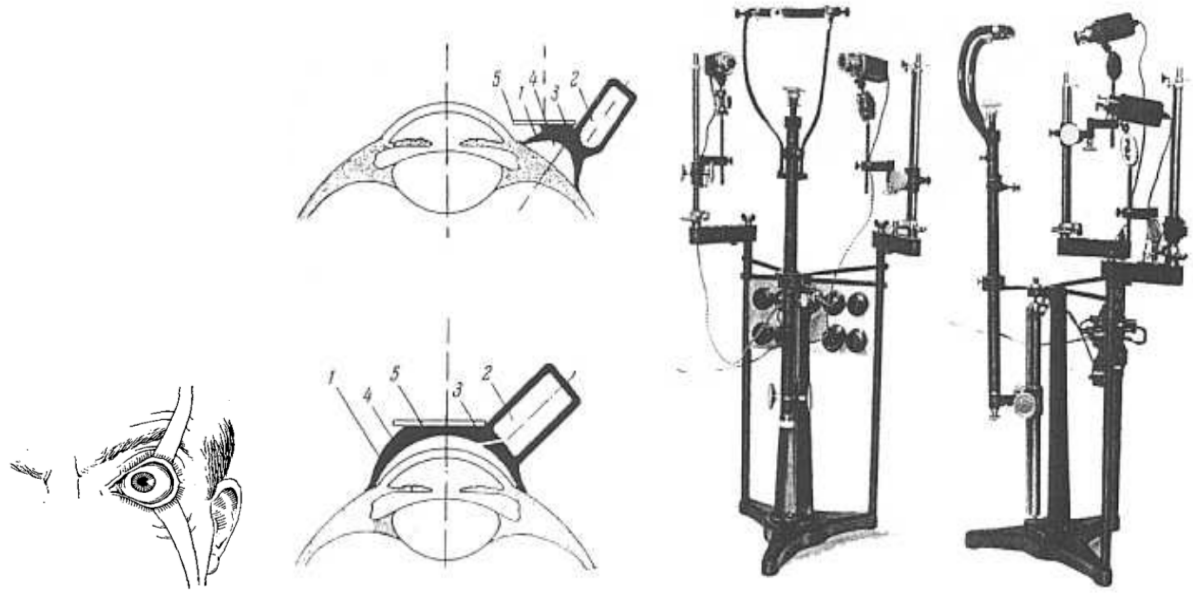


FIGURE 2.6 – Gauche : la technique pour maintenir l'œil ouvert à l'aide du ruban adhésif. Milieu : deux des dispositifs développés par Yarbus. Droit : l'appareil photographique "photokymographe" utilisé pour illuminer les yeux et pour enregistrer le reflet de lumière du mirroirs. (source d'image : [Yarbus, 1967])

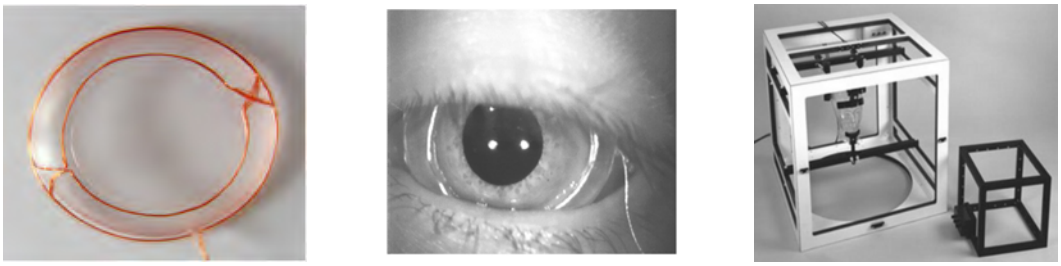


FIGURE 2.7 – Example de la bobine intégrée dans une lentille de contact portée sur l'œil et démonstration de l'utilisation du champ électromagnétique pour mesurer les mouvements des yeux (à droite).

2.2.2 Electro-oculographie (EOG)

Au milieu des années 1970, la technique électro-oculographique a été la plus largement utilisée pour mesurer les mouvements des yeux. Cette technique est l'une des premières techniques utilisées en milieu clinique pour enregistrer la mobilité oculaire et a fait l'objet de plusieurs applications dans le domaine du handicap. Son principe est de mesurer les différences du potentiel bio-électrique, résultant du champ bio-électrique rétino-cornéen modulé par les rotations de l'œil dans son orbite. Cette technique, peu coûteuse, utilise des électrodes (Figure 2.8) cutanées placées autour de l'œil dans le but de suivre les variations des potentiels électriques sur la peau entourant l'œil. L'œil est une pseudo sphère chargée positivement en avant et négativement en arrière.

Les potentiels enregistrés sont compris entre 15-200 μV ; lors du mouvement des yeux, la sensibilité de ce système est de l'ordre de 20 μV. Cette technique mesure les mouvements des yeux par rapport à la position de la tête, elle n'est donc généralement

Figure 2.8 – Exemple de l'éctro-oculographique (EOG) mesurant les mouvements oculaires.

pas appropriée pour mesurer le point de « fixation du regard » (PoR : Point of Regard) si la tête n'est pas maintenue dans une position fixe. Un des avantages est que l'enregistrement peut s'effectuer les yeux fermés ou semi-ouverts. Donc elle est beaucoup appliquée dans les études des mouvements oculaires pendant le sommeil. Bien que cette méthode enregistre les mouvements des deux yeux, elle demeure peu précise et inadéquate pour l'étude de la lecture. Elle est inconfortable et peut provoquer des irritations de la peau.

2.2.3 Photo-oculographie et vidéo-oculographie

La photo-oculographie et la vidéo-oculographie regroupent une grande variété de techniques d'enregistrement des mouvements oculaires, qui se base sur la mesure des différentes caractéristiques de l'œil (ex : forme de la pupille, position du limbe, réflexion cornéenne).

En 1901, Dodge et Cline ont pu mettre en place le premier appareil photo-oculographique ("Dodge Photochronograph" Figure 2.9 gauche), qui consistait à positionner un rayon lumineux sur la cornée et à photographier le mouvement de la réflexion. Ce dispositif est moins intrusif que celui de Huey et est à l'origine des oculomètres actuels qui font appel au reflet cornéen. La même idée est appliquée dans l'appareil de Buswell (Figure 2.9 droite) en 1935 qui est utilisé pour étudier les mouvements oculaires du sujet quand il observe des photos ou des motifs géométriques devant lui. A cause de la limite des techniques de cette époque, les mesures des caractéristiques des yeux fournies par ces dispositifs ne peuvent pas être traitées automatiquement, mais par une inspection visuelle et manuelle de chaque image enregistrée, généralement sur cassette vidéo. Ces mesures sont fastidieuses et limitées aux échantillons observés sur une courte période. Plusieurs de ces méthodes nécessitent l'immobilisation de la tête.

La technique basée sur la poursuite du limbe, qui est la frontière entre la sclérotique (blanc de l'œil) et l'iris (partie sombre), se fonde sur la mesure de la lumière réfléchie suite à l'éclairage du limbe. Si on éclaire cette région de l'œil, la quantité de lumière réfléchie

dépend des surfaces relatives de la sclérotique et de l'iris dans le champ de mesure et donc de la position de l'œil. Elle est peu coûteuse car il suffit d'une simple source de lumière couplée à un détecteur élémentaire, l'ensemble pouvant être fixé sur une monture de lunettes. Sa simplicité lui assure un coût faible. Cependant, les mesures sont facilement perturbées par les mouvements de tête et elles sont limitées aux mouvements horizontaux car le limbe est souvent masqué par les paupières supérieures [Scott et al., 1993].

La technique qui utilise la réflexion de la lumière par l'œil est très répandue. Elle utilise une lumière souvent infra-rouge projetée sur l'œil du sujet. Le dispositif de [Jouen et al., 1995] (Figure 2.10) comprend deux sources de lumière infra-rouge calibrées et deux caméras. Chaque source de lumière infra-rouge est apte à délivrer un faisceau de lumière de référence destiné à être réfléchi sur un œil du sujet, puis détecté par la caméra. Les images des yeux fournies par les deux caméras sont effectuées par les moyens de traitement en parallèle qui calculent les coordonnées du centre du point cornéen en référence à celles du centre de la pupille pour déterminer une direction de fixation.

La surface externe de la cornée n'est pas la seule partie qui peut refléter la lumière. Généralement les rayons émis éclairant l'œil sont réfléchis par quatre parties de l'œil : l'extérieur de la cornée, son intérieur, l'extérieur du cristallin et son intérieur (Figure 2.11). Ces quatre reflets sont connus comme les *reflets de Purkinje* ou les *images de Purkinje*. Nous pouvons utiliser le premier reflet de Purkinje et la position de la pupille comme les deux points de référence pour déterminer le point du regard, comme le fait la plupart des oculomètres vidéo-graphiques. L'oculomètre de [Cornsweet et al., 1973] utilisait le mouvement relatif du premier reflet (l'extérieur de la cornée) et du quatrième reflet (l'intérieur du cristallin) de Purkinje pour estimer le point du regard d'une manière très précise. Cette technique DPI (Dual Purkinje Image) est performante et elle permet d'enregistrer les mouvements oculaires très précis, mais elle exige une stabilisation de la tête, souvent avec une barre de maintien.

Les anciens modèles d'oculomètre présentés dans cette section sont souvent encombrants et gênants étant donné leur manière de fixer la tête du sujet et de filmer les yeux à très courte distance. Depuis ces 30 dernières années, grâce à l'augmentation de la vitesse des microprocesseurs et l'amélioration des techniques et matériels, les oculomètres modernes peuvent être moins gênants, comme les modèles à distance, et plus compacts et légers, comme les modèles portables :

- **Modèles à distance :**

  L'oculomètre à distance (Figure 2.12) permet de mettre le dispositif d'enregistrement (la caméra) plus loin du sujet. La tête du sujet n'a plus besoin d'être fixée avec une barre de maintien. Le système peut être tolérant aux légers mouvements de la tête si elle reste dans la zone de capture. Le participant est plus confortable et par conséquent ses comportements sont plus naturels dans les expérimentations. Le principe du système est similaire aux autres systèmes vidéo-oculographiques qui utilisent le reflet cornéen et la position de la pupille. Malgré la distance du capteur, le système peut fournir une précision pertinente sur le point du regard. L'utilisation de l'oculomètre à distance permet d'élargir les applications comme les études des mouvements oculaires des enfants, l'alerte de fatigue du chauffeur routier, l'assistance et la communication des personnes handicapées, etc.

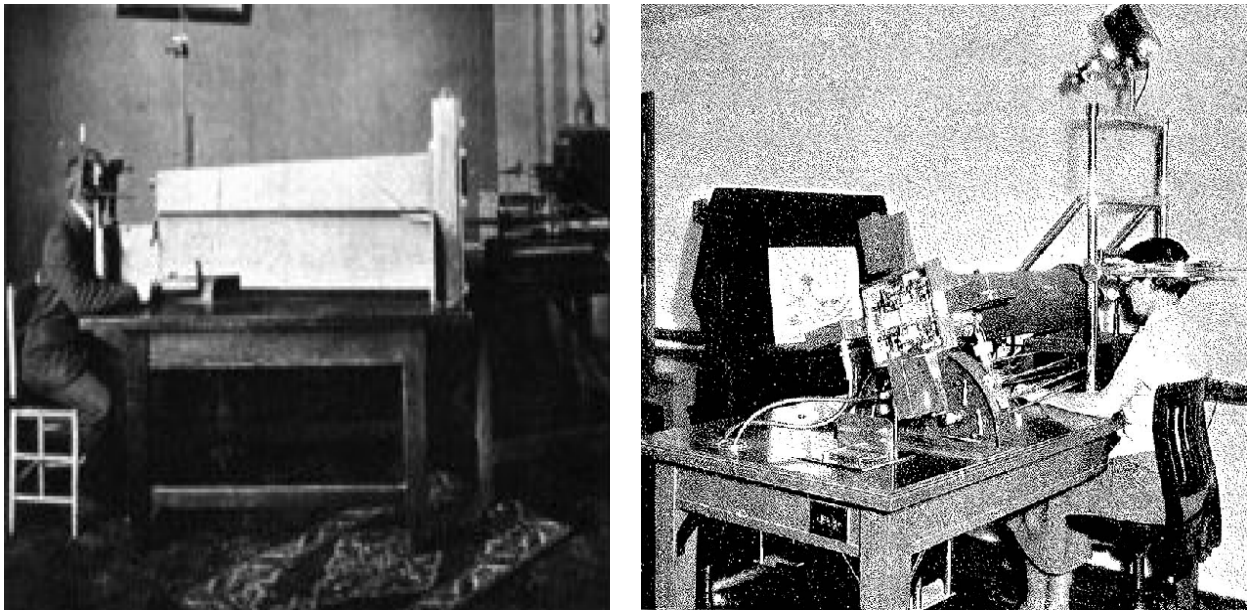


Figure 2.9 – Gauche : le "Dodge Photochronograph" (1908), qui enregistre la reflexion cornéenne de la lumière par la technique photographique. Droite : l'appareil de Buswell (1935) enregistre la reflexion de la lumière à l'aide des lentilles et mirroirs.

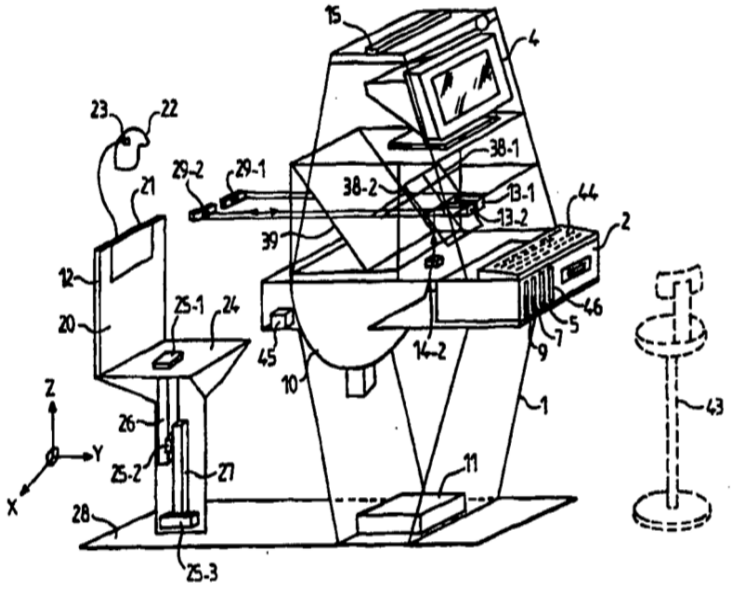


Figure 2.10 – Dispositif de contrôle des mouvements oculaires ([Jouen et al., 1995]). Le sujet est assis sur un support adapté(12), et l'œil est éclairé par un faisceau de lumière infra-rouge(14). L'image de l'œil capturée par la caméra(13) est traitée en calculant les coordonnées du centre du point cornéen en référence aux ceux du centre de la pupille.

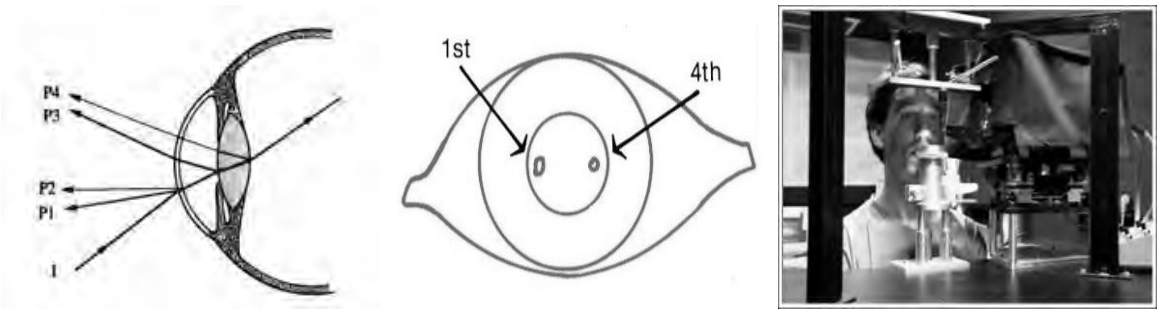


FIGURE 2.11 – Gauche : les 4 réflexions de la lumière par les 4 parties :l'extérieur de la cornée (P1), son intérieur (P2), l'extérieur (P3) du cristallin et son intérieur (P4). Milieu : la vue frontale sur les points P1 and P4. Droite : un oculomètre utilisant la technique DPI.

FIGURE 2.12 – Modèles d'oculomètre à distance

- **Modèles portables :**

  Le premier oculomètre portable a été développé par Hartridge et Thompson en 1948. Aujourd'hui il existe de plus en plus de modèles encore plus légers et moins gênants à porter. Ce type de modèle s'installe souvent sur la tête du sujet sous la forme de casque ou de lunettes (Figure 2.13). L'objectif est que le sujet soit libre de ses mouvements. Nous pouvons l'utiliser pour les applications dans une situation réelle hors du laboratoire. Par exemple, grâce à ce modèle, nous pouvons étudier les regards du consommateur quand il cherche des produits au rayon dans un supermaché, ou étudier l'attention des enfants autistes pendant la communication avec les autres, etc.

FIGURE 2.13 – Modèles de l'oculomètre portable

2.3 Oculométrie numérique

Grâce au développement significatif de la technique informatique depuis ces 30 dernières années, l'oculométrie numérique devient plus en plus performante comparativement aux systèmes oculométriques qui utilisent des caméras analogiques. Les applications de l'oculométrie se trouvent dans les domaines de recherche de plus en plus élargis, comme l'informatique, la psychologie cognitive ou l'éducation. Les images de l'œil capturées par les caméras numériques et les processus de traitement d'images sont indispensables dans un système d'oculométrie numérique. A ce jour les dispositifs du commerce peuvent atteindre une précision de 0.5 à 1.0 degré. Techniquement ces systèmes utilisent souvent la lumière infra-rouge pour la détection des caractéristiques de l'œil et une phase de calibration pour estimer la position du regard.

Dans cette partie, nous allons présenter en détail dans la section 2.3.1 les différents modules de l'oculomètre numérique, et les méthodes utilisées pour localiser les yeux et estimer la direction du regard. Ensuite, nous présenterons les techniques principales des systèmes (2.3.2) et la performance de ce type de modèles dans la section 2.3.3.

2.3.1 Méthodologie

Un système d'oculométrie numérique basé sur la caméra (ou vidéo) est généralement composé de 2 modules importants : la localisation des yeux et l'estimation de la position du regard (PoR : Point of Regard). Le module de localisation des yeux regroupe un ensemble de méthodes nécessaires pour identifier la région des yeux, suivre les yeux détectés dans la séquence d'images et extraire les caractéristiques (la position précise de la pupille, la contour de l'œil, ou la position de la tête, etc.) qui servent pour estimer le PoR. Souvent la position de la tête est prise en compte pour améliorer l'estimation du PoR. Le module pour estimer le PoR utilise d'abord une phase de calibration pour établir un lien entre les caractéristiques extraites de l'image des yeux et la position du regard. Après avoir obtenu les données essentielles par la calibration, le système peut estimer le PoR en coordonnées 2D sur une région précise comme un écran par exemple. La figure 2.14 illustre le schéma global et la relation entre chaque module du système d'oculométrie.

2.3.1.1 Localisation des yeux

L'apparence des yeux dans l'image dépend de plusieurs critères tels que la couleur de la peau, l'illumination, et l'angle de prise de vue. Pour le même sujet, une petite variation de l'angle de vue peut provoquer des changements considérables dans l'apparence de l'image. L'image de l'œil peut être représentée par la distribution de l'intensité de la pupille, l'iris et la cornée, ainsi que par leurs formes. L'angle de vue, la position de la tête, la couleur des yeux, l'illumination, la position de l'iris dans l'orbite de l'œil, et l'état de l'œil (c-à-d., ouverture/fermeture) sont des facteurs importants qui influencent beaucoup l'apparence de l'œil. La détection et le suivi des yeux sont difficiles et complexes en raison de plusieurs problèmes spécifiques, comprenant l'occlusion de l'œil par la paupière, le degré de l'ouverture de l'œil, la variabilité de la taille, la réflectivité ou la position de la tête, etc.

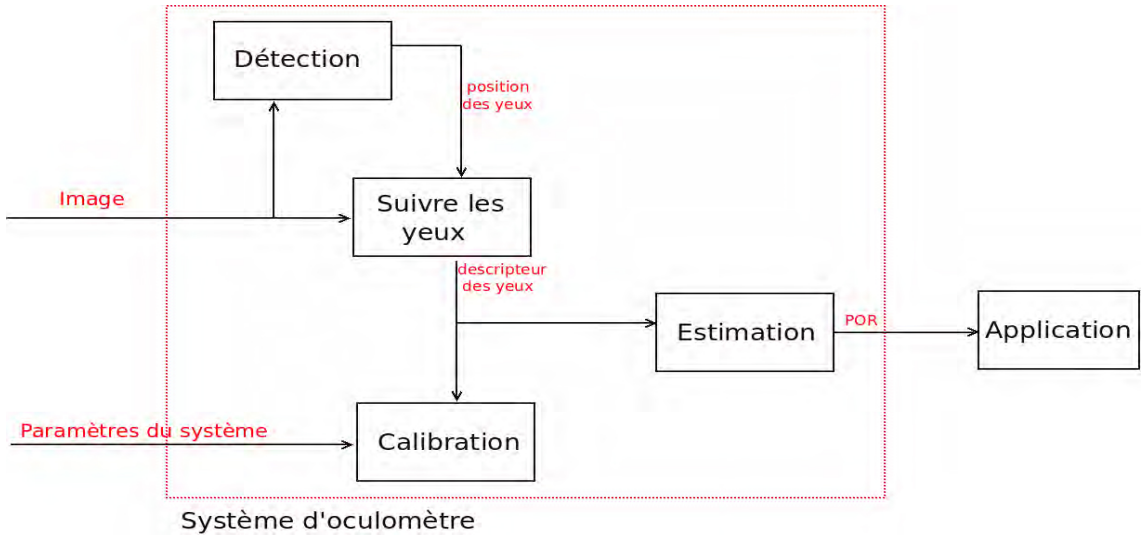


FIGURE 2.14 – Le schéma global des composants dans un système d'oculométrie numérique.

Les techniques de localisation des yeux utilisent souvent les propriétés géométriques (la forme des yeux) et les caractéristiques des yeux (la pupille, l'iris, etc.) pour identifier la présence des yeux dans l'image. L'article de [Hansen and Ji, 2010] résume en détail les deux techniques.

- **Modèle des yeux :**
  Cette approche essaie de construire un modèle des yeux qui décrit la forme des yeux et la contour de l'iris, de la pupille, etc. Généralement le modèle des yeux comprend 2 composantes : un modèle géométrique des yeux qui définit la forme initiale des yeux et une mesure de similarité qui permet au modèle de se transformer pour s'adapter à la forme réelle des yeux dans l'image. L'avantage de cette technique est de traiter les yeux de taille, de forme et de rotation différentes. Les limitations principales sont sur le temps du calcul, l'exigence de la qualité d'image, la position initiale du modèle, etc.

- **Caractéristiques des yeux :**
  Ces méthodes explorent les caractéristiques de l'œil (la pupille, l'iris, le reflet cornéen, etc.) afin de localiser sa position. Généralement ces méthodes sont utilisées avec l'aide de l'illumination infra-rouge qui permet de provoquer l'effet de pupille sombre ou claire selon la position de souce de la lumière infra-rouge. Cet effet est similaire à l'effet "red-eye" qui résulte du flash dans la photographie. Ces méthodes sont robustes contre le changement de la lumière, mais exigent d'avoir des images de résolution et contraste élevés.

En résumé, localiser les yeux reste une tâche difficile et complexe. Il n'existe pas une seule méthode ou un descripteur "parfait" pour détecter les yeux correctement dans 100% des cas, surtout dans les conditions différentes. Chaque approche a ses avantages et ses propres limites. Afin d'obtenir un résultat optimal, nous devons définir les conditions, telles que les composants du système, la condition de la lumière, le comportement du sujet, l'environnement et ensuite choisir les méthodes adéquates pour résoudre le problème de localisation sous ces conditions.

2.3.1.2 Estimation du regard

Le point du regard PoR (ou la direction du regard) est déterminé par l'orientation et la rotation de l'œil, ou plus précisément par l'axe visuel (section 2.1.2), qui relie le centre de la cornée et la fovéa. L'approche fondée sur le traitement des caractéristiques utilise directement les caractéristiques des yeux comme la pupille, les contours ou les coins des yeux, ainsi que le reflet de la lumière infra-rouge sur la cornée, pour estimer le point du regard. L'oculomètre qui applique cette approche utilise souvent une ou plusieurs caméras de qualité pour capturer les images à traiter. En plus, la source de lumière infra-rouge est indispensable pour générer le reflet cornéen.

La méthode P-CR (Pupil-Corneal Reflection) est une méthode de régression classique. Elle utilise la première image de Purkinje comme point de repère et le centre de la pupille pour former un vecteur du descripteur $< g_x, g_y >$ (Figure 2.15). L'objectif est d'estimer le point du regard $< p_x, p_y >$ sur l'écran à partir du vecteur $< g_x, g_y >$. Après la phase de calibration, une corrélation entre les coordonnées du PoR et un ensemble de vecteurs du descripteur est établie. Par conséquent, une fonction de régression f peut être construite pour estimer le point du regard pour un nouveau vecteur du descripteur $< g_x, g_y >$. Cette méthode est relativement simple mais techniquement elle exige l'immobilisation de la tête pour que le reflet cornéen reste stable. En conséquence, une barre de maintien est souvent utilisée pour fixer la tête du sujet. Un autre inconvénient est qu'une calibration longue est indispensable pour obtenir une bonne précision de la mesure.

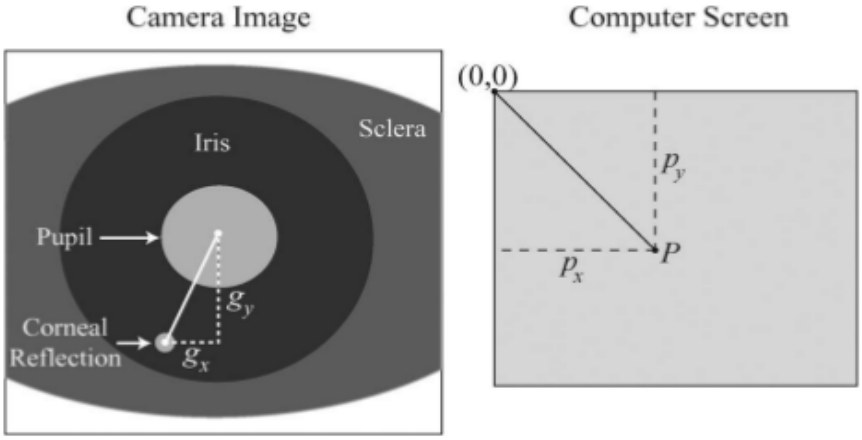


FIGURE 2.15 – La méthode P-CR pour corréler l'œil et le regard. Gauche : dans la région de œil, le vecteur du descripteur $< g_x, g_y >$ est formé par la première image de Purkinje et le centre de la pupille. Droit : sur l'écran, le point P représenté par le vecteur $< p_x, p_y >$ est l'endroit où l'œil regarde.

Une autre technique est basée sur la réalisation d'un modèle géométrique de l'œil en 3D et elle estime le PoR en calculant directement la LoS (Line of Sight) qui représente l'axe visuel. Le principe général, illustré dans la Figure 2.16, est de calculer le vecteur LoS par la position du centre de la pupille et la position du centre de la courbure cornéenne, soit le point C. Le point du regard PoR est déterminé comme étant le point P de l'intersection de ce vecteur LoS et du plan de la scène (par exemple un écran).

Le diamètre de l'œil est environ 25mm et la distance entre le centre de la pupille et

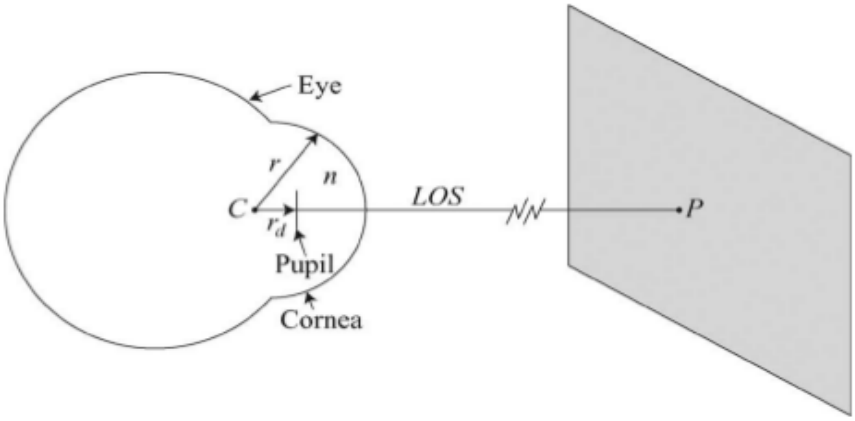


FIGURE 2.16 – La méthode 3D modélise la structure de l'œil en 3D afin d'estimer le regard par l'intersection P entre l'axe visuel et le plan.

le centre de la courbure cornéenne mesure environ 4.5mm. Par conséquent les positions doivent être mesurées très précisément. 0.1mm d'erreur dans le calcul de la position de la pupille peut générer une erreur de 1.2° d'angle visuel, ce qui est équivalent à une erreur de 14.7mm sur un écran à une distance de 700mm. Dans ce cas là, la refraction de la cornée doit être prise en compte pour calculer la position exacte de la pupille.

2.3.2 Techniques

Pour être précis, le système d'oculométrie, qui utilise les méthodes présentées ci-dessus pour détecter les yeux et estimer le regard, utilise souvent une ou plusieurs caméras de qualité pour capturer les détails de l'œil. Par exemple, un appareil stéréoscopique [2] permet de calculer le vecteur LoS en 3D en utilisant la différence des positions du reflet cornéen dans les deux images capturées par l'appareil. En outre, le système nécessite d'appliquer aussi des techniques spécifiques comme la calibration et l'installation de la lumière infra-rouge. Les détails de ces techniques peuvent être référencées dans [Hansen and Ji, 2010] [Hammoud, 2008].

2.3.2.1 Calibration

La calibration est une phase dont l'objectif est de recueillir tous les attributs nécessaires (par exemple les caractéristiques des yeux, la configuration des matériels du système, etc.) pour établir un lien entre les mouvements oculaires et la position du regard sur l'écran. Elle est liée étroitement avec la technique d'estimation du regard et elle existe dans tous les modèles d'oculomètre. Généralement il existe quatre types de calibrations :

2. Un appareil stéréoscopique (plus court : appareil stéréo) est un appareil photographique rassemblant deux chambres photographiques - et donc deux objectifs - placés côte à côte de manière solidaire dans un même boîtier, destiné à produire commodément et dans un même instant un couple stéréoscopique, c'est-à-dire deux photographies jumelles (mais non semblables) en vue de la restitution du relief ou de la 3ème dimension.

- **Calibration de la caméra** :
  Cette procédure détermine les paramètres intrinsèques de la caméra comme : la distance focale, qui correspond à la distance (en mm) séparant le plan rétinien [3]et le point focal [4] ; les paramètres de conversion, qui représentent l'ajustement horizontal et vertical qui permettent de passer d'un repère du plan rétinien (exprimé en mm) au repère image (exprimé en pixels) ; la position du point principal (le centre de l'image) définie comme étant la projection du centre optique sur le plan rétinien, exprimée en pixels.

- **Calibration géométrique** :
  Cette opération détermine la localisation et l'orientation relative des composants du système comme la caméra, la source de la lumière infra-rouge et l'écran de l'expérimentation.

- **Calibration individuelle** :
  L'œil de chacun est légèrement différent par rapport à la taille et la forme du globe, au taux de refraction, etc. L'objectif de cette calibration est de déterminer les facteurs tels que la courbure de la cornée et l'angle entre l'axe visuel et l'axe optique, qui ont des impacts sur la performance d'un oculomètre.

- **Calibration des marqueurs** (mappage "oeil - regard") :
  Pendant cette phase, le sujet est censé fixer son regard sur plusieurs marqueurs affichés un par un sur l'écran. Elle permet de recueillir les informations importantes pour établir la fonction du mappage entre l'œil et les coordonnées du regard à l'écran.

Un oculomètre numérique, surtout celui qui utilise la pupille et le reflet cornéen pour estimer le regard, nécessite de procéder à ces quatre calibrations au cours de l'installation du dispositif. Les trois premières calibrations nécessitent l'intervention de l'experimentateur et il suffit de les faire juste une fois pendant la préparation et l'installation. Ensuite, le sujet va suivre la calibration des marqueurs. Le choix du nombre des marqueurs (points de calibration) est important. S'ils ne sont pas assez nombreux, la performance du système risque de se dégrader à cause du manque d'information nécessaire pour établir une fonction de régression pertinente. S'ils sont trop nombreux, le sujet se fatigue et le risque d'erreur augmente. Beaucoup de techniques sont proposées pour améliorer la performance des trois premières calibrations afin de diminuer le nombre de points de calibration pour le sujet. Il existe des oculomètres qui ont besoin d'un ou deux points de calibration ou qui ne demandent pas de point de calibration.

Pour les systèmes d'oculométrie qui ne font pas les trois premières calibrations, la préparation et l'installation du système sont plus faciles. Par contre pour atteindre une bonne précision, plus de points de calibration sont nécessaires. Pour diminuer le nombre de points et garder une précision correcte, le système doit améliorer la méthode d'extraction du descripteur des yeux et la fonction de régression.

3. Le plan rétinien (plan d'image) est le plan dans lequel l'image se forme à l'aide d'une projection perspective.

4. Le point focal est défini de sorte que les rayons de lumière reflétés par les objets passent par ce point, formant une image perspective de la scène dans le plan rétinien.

2.3.2.2 Illumination infra-rouge

Généralement, l'œil humain couvre les longueurs d'ondes allant de 400 *nm* à 700 *nm*. Les infra-rouges sont des rayons lumineux (des ondes électromagnétiques), dont la longueur d'onde est supérieure à celle de la couleur rouge, mais inférieure aux microondes. L'œil humain ne peut pas distinguer les infra-rouges, qui sont d'une longueur d'onde comprise entre 700 *nm* et 1 000 000 *nm* (Figure 2.17), supérieure au domaine du visible.

La lumière infra-rouge aide à stabiliser la condition de la lumière et éliminer les lumière autour de l'objet dans l'environnement. En plus, on peut avoir le reflet sur la cornée. Si on ajoute des filtres pour laisser passer l'infra-rouge et bloquer seulement la lumière visible, la qualité d'image sera encore meilleure. La technique qui consiste à générer une pupille sombre ou claire facilite la tâche de localisation de la pupille (Figure 2.18). Si une source de lumière infra-rouge se situe près de l'axe optique de la caméra, l'image capturée par la caméra montrera une pupille claire, parce que la plupart des lumières vont retourner directement sur la caméra. Cet effet est similaire à celui de "red-eye" qui résulte du flash dans la photographie. Quand la source de lumière est loin de l'axe optique de la caméra, nous allons observer une pupille sombre dans l'image.

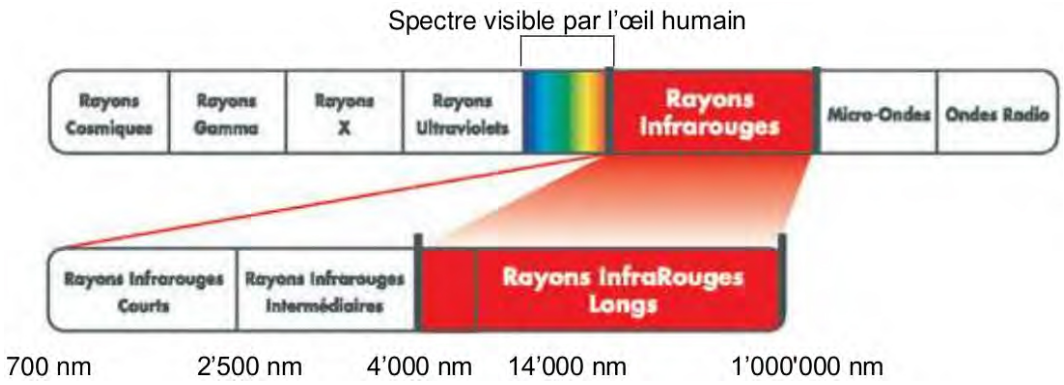


FIGURE 2.17 – Longueurs d'ondes visibles et invisibles pour l'œil humain.

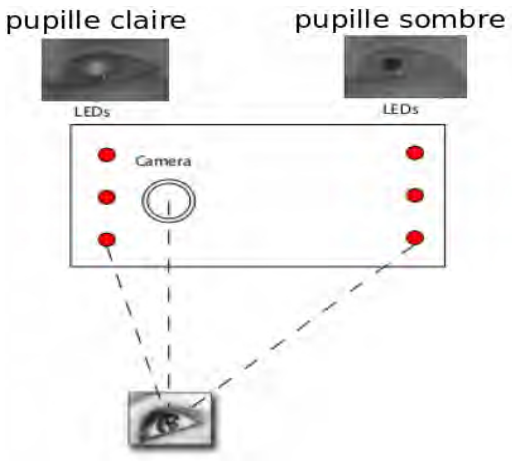
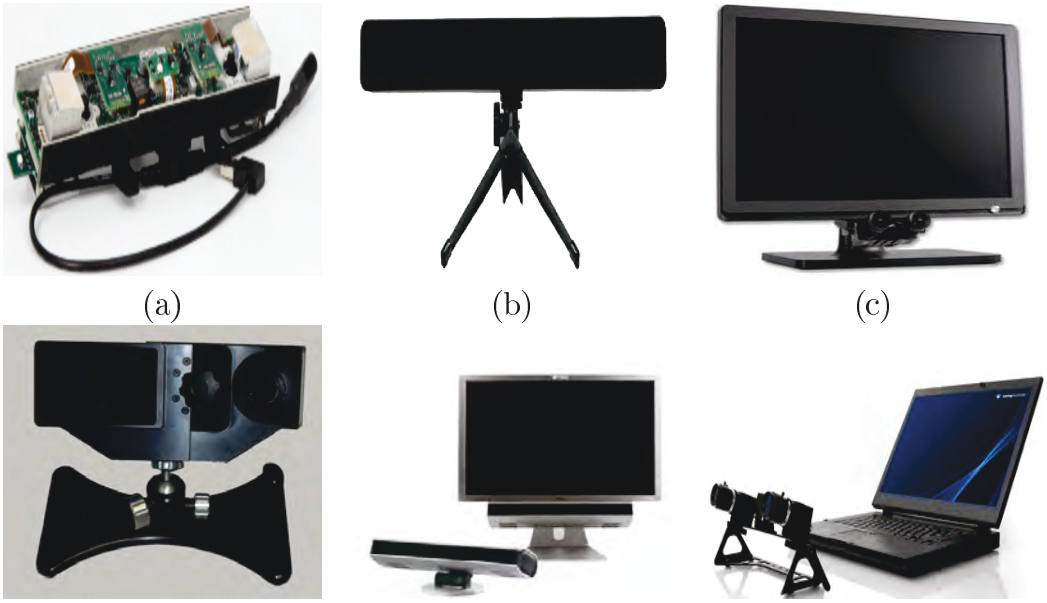


FIGURE 2.18 – L'effet de la pupille sombre et claire qui est utilisé pour identifier l'œil.



(a) (b) (c)

(d) (e) (f)

FIGURE 2.19 – Les différents modèls oculométriques : (a) Tobii IS1 (b) Mirametrix S2 (c) LC EyeGaze Analysis (d) Eyelink 1000 (e) SMI Red250 (f) FaceLab 5

2.3.3 Performance

Au cours de cette dernière dizaine d'années, de nombreux nouveaux systèmes oculométriques sont apparus sur le marché. Ces produits commerciaux permettent des mesures précises et autorisent le mouvement libre ou relativement libre de la tête dans un espace défini. Ici nous présentons les six oculomètres (Figure 2.19) comme examples : Tobii IS1, Mirametrix S2, LC EyeGaze Analysis, Eyelink 1000, SMI Red250 et FaceLab 5. Ces oculomètres à distance intègrent des équipements spécifiques comme des sources infra-rouges, des caméras de haute résolution, l'écran, etc.

Les différents systèmes oculométriques du marché présentent des équipements et des installations différents, mais généralement deux composants sont indispensables : l'illumination infra-rouge et la caméra. Aujourd'hui l'oculomètre n'est pas considéré comme un produit accessible et nécessaire au quotidien pour tout le monde comme le téléphone, le PC, etc. La plupart des utilisations de l'oculomètre restent spécialisées dans la recherche et limitées au laboratoire. En tant que système spécifique, la performance dépend de ces 2 facteurs :

- **Qualité des composants :**
  L'oculomètre video-graphique a besoin d'obtenir la position précise des reflets cornéen et de la pupille afin de déterminer la position du regard. Par conséquent la qualité des composants comme la caméra, le microprocesseur, la source d'illumination, etc. est cruciale pour la qualité de l'oculomètre. Une caméra de haute résolution permet de capturer une image de haute qualité qui donnera plus de détails sur la région des yeux. Une caméra de haute fréquence peut capturer les mouvements oculaires très rapides. En outre, un microprocesseur de grande vitesse, les lumières infra-rouges, la lentille de la caméra sont aussi des composantes importantes pour obtenir un oculomètre de qualité.

- **Développement du matériel et logiciel :**
  Un prototype d'oculomètre nécessite des années de développement. Il s'agit du développement du "hardware", qui concerne la conception et la réalisation concrète du prototype avec tous les matériels concernés, et aussi du développent du "software", qui comprend les bibliothèques et les algorithmes de traitement, SDKs, ou APIs, pour visualiser, interpréter et analyser les résultats.

Les clients de l'oculomètre sont souvent des experts et des chercheurs de laboratoire. La vente de ce produit n'est pas très répandue sur marché. Par conséquent, les vendeurs sont obligés d'augmenter le prix pour atteindre le retour sur les investissements, ou de créer des services de support ou de collaboration pour les experts ou les chercheurs.

Aujourd'hui la technologie oculométrique est largement appliquée dans le domaine d'IHM (Interactions Homme-Machine). Nous pouvons utiliser notre regard pour écrire un message, dessiner, jouer à un jeu video, etc. Cette technologie est plus en plus intéressante pour les applications mobiles, puisque presque tous les mobiles aujourd'hui sont équipés de caméra. Le Samsung Galaxy S3 et S4 possède déjà la fonctionnalité "smart scroll" (Figure 2.20). Lors de la lecture d'une vidéo, celle-ci se met en pause si le regard se détourne de l'écran et la lecture reprend dès que l'utilisateur le regarde de nouveau. Elle permet aussi de faire défiler une page Web ou vos emails sans toucher le téléphone. Les futurs possesseurs de Google Glass pourront déverrouiller leur écran de veille d'un simple coup d'œil grâce à un brevet déposé par le constructeur. Le système d'oculométrie pourrait s'installer sur les Google Glass (Figure 2.21). Il devrait être utilisé, dans un premier temps, pour déverrouiller l'écran de veille. En effectuant un mouvement d'œil, le porteur de l'appareil pourra ainsi accéder à toutes ses données en échange d'un effort minimal. Quant au développement, Tobii a proposé un produit OEM comme *Tobii IS1* relativement moins cher pour les programmeurs qui veulent réaliser leur propre oculomètre.

La condition nécessaire pour l'analyse des mouvements oculaires, surtout dans le cadre de la conception de l'oculomètre, est l'identification des mouvements comme les fixations, les saccades et la poursuite. De plus une fixation se compose de 3 types de mouvements oculaires comme la dérive, le tremor et les micro-saccades involontaires. La plupart des oculomètres ne peuvent observer qu'une partie de ces mouvements oculaires. La Table 2.1 résume les caractéristiques de ces mouvements en amplitude, durée et vélocité. Dans cette section, nous allons présenter les critères pour évaluer la performance d'un oculomètre. Scott et Findlay[Scott et al., 1993] ont proposé plusieurs critères pour décrire un oculomètre "parfait". L'oculomètre est évalué non seulement en fonction des résultats du système, mais aussi par sa technique d'installation, la façon de l'utiliser et l'influence sur les sujets, etc.

En résumé, ces critères sont les suivants :

- **Précision :** La précision est mesurée en degré et actuellement la précision de la plupart des oculomètres du marché peut atteindre $\pm 0.5°$. Cette précision est déjà suffisante pour la plupart des applications d'IHM. Mais le système oculométrique nécessite une précision plus fine afin d'analyser les mouvements comme les micro-saccades et les trémors.

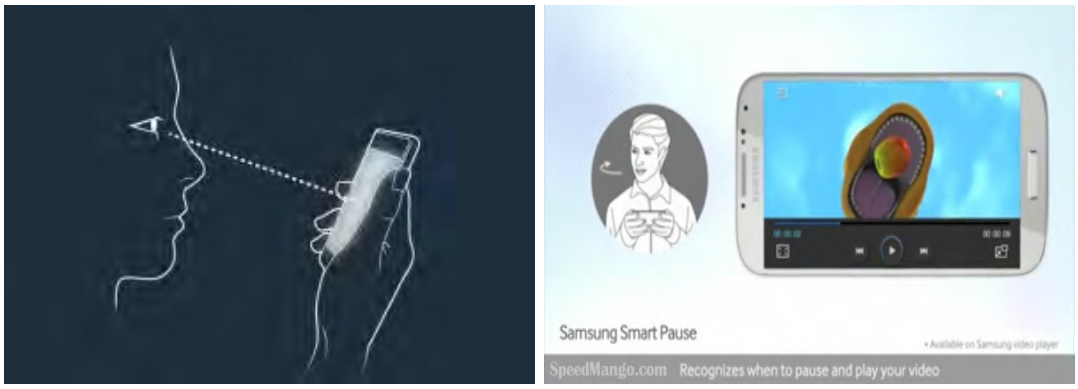


FIGURE 2.20 – La fonctionnalité "smart scroll" dans le Samsung Galaxy smartphone permet de mettre en pause une vidéo, lorsque le regard se détourne de l'écran et de reprendre la lecture dès que l'utilisateur le regarde de nouveau.

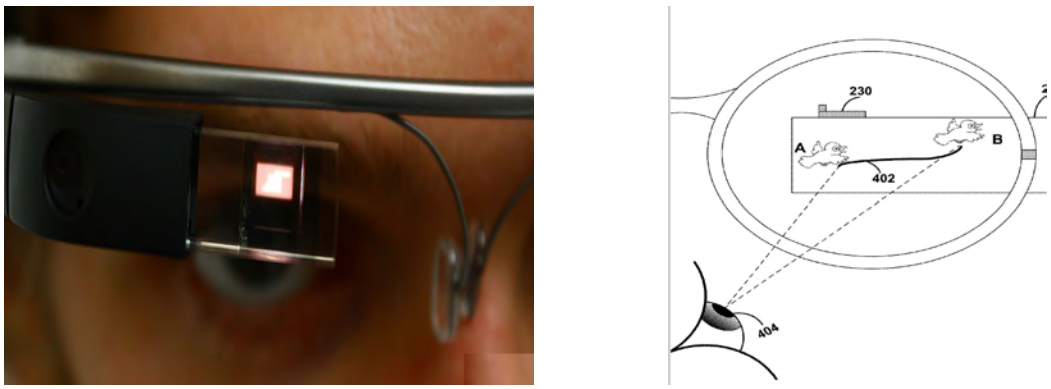


FIGURE 2.21 – Google Glass (GG) permet d'identifier le mouvement oculaire qui suit un objet à l'écran pour déverrouiller l'écran de veille.

- **Temps de résolution et latence :** Tous les systèmes oculométriques en temps réel ont besoin d'un certain temps pour calculer les résultats (le suivi du regard). Cet intervalle comprend la latence et le temps de résolution. La latence est déterminée par la fréquence de la caméra et le temps de résolution désigne le temps de fonctionnement du système (la détection des yeux, l'estimation du regard etc).

- **Robustesse :** Ce critère révèle la capacité du système oculométrique sous différentes conditions, par exemple, le changement de la luminosité, la variété des sujets (âge différent, couleur de peaux différentes, avec ou sans lunettes etc.), le changement de la distance entre le sujet et le stimulus visuel, ou les mouvements de la tête ou du corps du sujet.

- **Filtrage au bas-niveau :** Les données fournies par le système dans un premier temps sont très bruitées à cause de la capture des images ou des clignements des yeux, etc. L'oculomètre a besoin de traitements pour éliminer ces bruits et lisser le signal obtenu.

- **Interface de programmation et de visualisation :** Il s'agit du logiciel du système d'oculométrie qui permet de fournir des interfaces pour les utilisateurs et les experts. L'interface de visualisation permet aux utilisateurs de visualier facilement

Type	Durée (ms)	Amplitude	Vélocité
Fixation	200 – 300	—	—
Saccade	30 – 80	4 – 20°	30 – 500°/s
Poursuite	—	—	10 – 30°/s
Dérive	200 – 1000	1 – 60′	6 – 25′/s
Trémor	—	$< 1'$	—
Micro-saccades	10 – 30	10 – 40′	15 – 50°/s

TABLE 2.1 – Les caractéristiques des mouvements oculaires [Holmqvist et al., 2011]

les résultats, et l'interface de programmation peut aider les experts à réaliser leurs propres applications.

- **Technique d'installation et d'utilisation :** Chaque oculomètre a sa propre méthode d'installation et d'utilisation. Les questions sont de savoir si l'oculomètre nécessite beaucoup de configuration manuelle avant de fonctionner, si l'oculomètre gêne une partie de la vision du sujet, s'il est en contact directement avec le corps, ou s'il a besoin de beaucoup de points de calibration, etc. Ce sont des facteurs importants pour évaluer un oculomètre.

Actuellement, il n'existe pas encore d'oculomètre "parfait" à cause des limites de la technique. Ici nous résumons les critères importants de six oculomètres cités dans notre travail (la Table 2.2) :

	Tobii IS1	Mirametrix S2	EyeGaze Analysis	EyeLink 1000	SMI Red 250	FaceLab 5
Précision [5]	$< 0.5°$	0.5° - 1°	0.4°	0.5°	<0.4°	0.5° - 1°
Fréquence	< 25 - 40 H_z	60 H_z	120 H_z	500 H_z	60 H_z	N/A
'Head box' [6] ($w \times h \times d$cm)	40×30 à 60 cm	$25 \times 11 \times 30$	N/A	$22 \times 18 \times 20$	d = 60 - 80	$35 \times 23 \times 60$
Calibration	N/A	9 points	automatique	N/A	2,5,9 points	N/A
Binoculaire tracking	oui	oui	oui	non	oui	oui

TABLE 2.2 – Les critères de six oculomètres présentés dans la Figure 2.19.

5. Le mouvement oculaire est souvent mesuré par le degré d'angle visuel (°) ou la minute (′), où $1° = 60'$ (Annexe 1). Ayant la distance entre le sujet et l'écran d et la distance de déplacement x sur l'écran, on peut calculer le degré d'angle visuel θ par $tan\theta = \frac{x}{d}$. La précision d'un oculomère est mesurée par le dégré d'angle visuel.

6. 'Head box' est la région où le sujet est permet de bouger la tête sans perdre le contact avec l'oculomètre.

2.4 Le projet ONE (Oculométrie Numérique Economique)

2.4.1 Motivations

La technologie de l'oculométrie aujourd'hui devient plus en plus performante, mais il n'existe pas encore d'oculomètre "parfait" selon les critères comme la précision, la technique de l'installation, l'influence sur le sujet, etc. Par rapport à la précision, la plupart des oculomètres au marché peuvent atteindre $\pm 0.5°$, ce qui est suffisant pour la plupart des études sur les mouvements oculaires et des applications d'IHM. Mais pour certains types des mouvements oculaires comme les micro-saccades et les trémors, le système nécessite une précision encore plus fine. Le choix d'un tel oculomètre dépend du sujet des études ou des applications. Deuxièmement, l'utilisation de l'appareil a besoin d'experts pour son installation et sa configuration. Par exemple, la taille de l'œil est petite, et les capteurs doivent être placés près de l'œil du sujet afin d'obtenir les détails précis sur l'œil. Mais ceci va gêner plus ou moins les comportements du sujet.

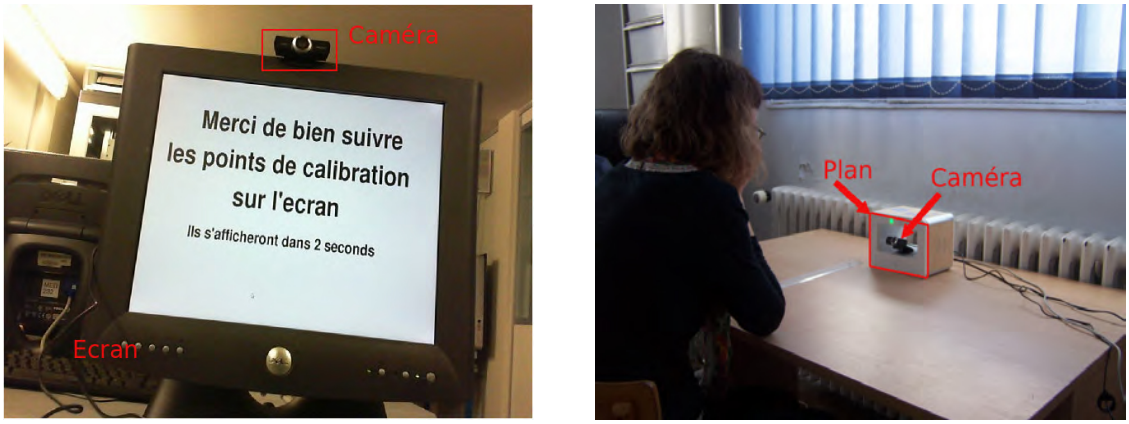


Figure 2.22 – L'environnement de l'expérimention. Gauche : l'installation de la webcam et l'écran. Droit : la position de la webcam et le plan où le sujet doit regarder.

Le projet ONE (Oculométrie Numérique Economique) vise à proposer un système oculométrique à distance, non-intrusif, sans lumière infra-rouge et surtout économique, qui permet d'enregistrer les mouvements oculaires en temps réel (Figure 2.22). Le terme "économique" signifie que le système est moins exigeant pour le choix des composants, moins compliqué pour l'installation et la configuration. Il permet d'être utilisé pour des applications qui ont besoin simplement de mesurer des fixations ou la direction du regard. Depuis ces 30 dernières années, grâce à l'évolution de la technique informatique, les composants comme le processeur, les appareils périphériques deviennent plus en plus performants et accessibles par rapport au prix. Cela motive les chercheurs à développer leurs propres oculomètres "économiques". Ces oculomètres sont souvent développés en modèle portable comme le travail de [Noris et al., 2008], celui de [Kassner and Patera, 2012], et celui de [Martinez et al., 2012]. Dans ces modèles les webcams sont placées à proximité des yeux, ce qui permet de capturer facilement les détails de l'œil. Il existe également des modèles à webcam à distance comme ceux proposés par [Williams et al., 2006], [Nguyen et al., 2009] et [Tan et al., 2002].

Notre système oculométrique est fondé sur un modèle d'apparence de l'œil dont l'image est capturée par une webcam. Ce système utilise une webcam normale de 30 Hz, qui permet de capturer 30 images, d'une résolution entre 640×480 et 960×540 pixels, par seconde. La configuration minimale requise pour installer ce système sur PC comprend un processeur dont la performance est égale ou supérieure à Intel Pentium IV 800M et une mémoire vive au moins 2G. Pour bien capturer les mouvements des yeux sans gêner le comportement du sujet, la webcam se situe à distance du sujet. La distance entre la webcam et les yeux peut être différente selon la condition de l'expérimentation et correspond généralement à la longeur du bras lorsque le sujet est assis (Figure 2.22). La simplicité de l'installation permet au système de s'adapter à des conditions expérimentales variées.

2.4.2 Illumination naturelle

Contrairement aux autres systèmes, notre oculomètre n'utilise pas la lumière infra-rouge. La raison principale pour ne pas utiliser la lumière infra-rouge est le risque avéré pour les yeux de l'utilisateur. A ce jour les opinions sur ce point sont encore partagées. La norme NF EN 62471 relative à la "sécurité photobiologique des lampes et des appareils utilisant des lampes" en 2008 a proposé des limites d'exposition au rayonnement des différentes sources de lumière.

En octobre 2010, l'Anses (Agence nationale de sécurité sanitaire, de l'alimentation, de l'environnement et du travail) a publié un rapport sur "les effets sanitaires des systèmes d'éclairage utilisant des diodes électroluminescentes". Dans cette étude, elle identifie des risques liés aux lumières de courtes longueurs d'ondes et à l'éblouissement. Devant la place qu'occupent désormais les diodes électroluminescentes (LEDs) dans les applications de vision, la question de leur dangerosité soulève de nombreux débats. l'Anses a publié un certain nombre de recommandations visant à restreindre la mise sur le marché des LEDs présentant un certain niveau de risque, inciter les fabricants et intégrateurs à concevoir des systèmes d'éclairage ne permettant pas une vision directe du faisceau émis, etc. Elle recommande également d'adapter la norme NF EN 62471 aux LEDs, et de lever toute ambiguïté sur les protocoles de mesure. Elle propose aussi plusieurs mesures concrètes : éviter les sources de lumière émettant une forte lumière froide auprès des populations à risque, développer des moyens de protection adéquats pour les travailleurs exposés (lunettes de protection optique spécifiques aux LEDs, par exemple), rendre obligatoire le marquage du groupe de risque de sécurité photobiologique, ou encore la mise en place d'un étiquetage intelligible pour le consommateur.

Vu les études et les recommandations présentées ci-dessus, notre système d'oculométrie est conçu sans lumière infra-rouge. Dans nos expérimentations, le système utilise souvent l'illumination naturelle d'intérieur qui est suffisante pour que les yeux soient détectables dans les images.

2.4.3 Webcam

La technologie évolue rapidement dans le domaine de l'acquisition d'image depuis quelques années, que ce soit dans les domaines industriel et scientifique ou dans le do-

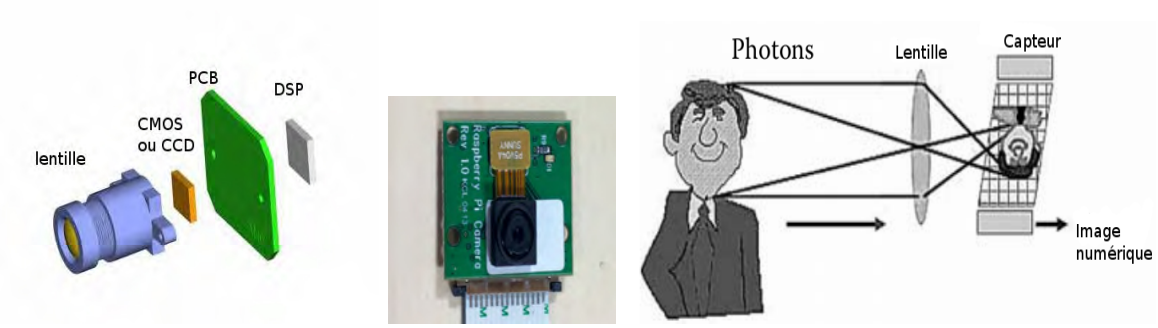


FIGURE 2.23 – Gauche : composants principaux d'une webcam. Milieu : un exemple concret. Droite : l'acquisition d'une image numérique.

maine grand public. Le terme "webcam" désigne une caméra numérique qui est conçue en tant que périphérique, pour transmettre des images en direct à un ordinateur, plutôt que de les stocker pour la lecture ou la transmission ultérieure. La webcam est une des sources principales des images et des vidéos dans le domaine de vision par ordinateur. Elle est très utile dans de nombreuses applications comme la visioconférence, la surveillance, la robotique, etc.

La Figure 2.23 résume les composantes générales d'une webcam, et illustre les mécanismes de conversion qui sont nécessaires pour obtenir une image numérique. Les composants principaux sont :

- Le capteur photographique : CCD [7] ou CMOS [8].
  Un capteur photographique est un composant électronique photosensible servant à convertir un rayonnement électromagnétique en un signal électrique analogique. Ce signal est ensuite amplifié, puis numérisé par un convertisseur analogique-numérique et enfin traité pour obtenir une image numérique. Deux grandes familles de capteurs sont disponibles : les CCD et les CMOS. Depuis l'invention du capteur CCD en 1969 par George Smith et Willard Boyle, le CCD a été utilisé pour des applications de pointe (imagerie astronomique) puis popularisé sur les caméras et appareils photo. Aujourd'hui le CCD existe encore sur les marchés des appareils compacts et les appareils à très haute résolution. Les appareils reflex les plus courant l'ont, quant à eux, délaissé et utilisent majoritairement des capteurs CMOS. Le CMOS a l'avantage d'être miniaturisable facilement, de faible coût. et de faible consommation électrique (beaucoup plus faible que celle du capteur CCD). Aujourd'hui la technologie CMOS est très bien maîtrisée et les capteurs CMOS constituent un concurrent sérieux des CCDs. Pour plus de détail, le lecteur pourra consulter [Nakamura, 2005].

7. Le capteur CCD (Charge Couple Device) est composé d'éléments photosensibles se présentant sous forme linéaire (barrette CCD comme dans les scanners à plat) ou sous forme de matrice (caméras). Un CCD transforme les photons lumineux qu'il reçoit en paires électron-trou par effet photoélectrique dans le substrat semi-conducteur, puis collecte les électrons dans le puits de potentiel maintenu à chaque photosite. Le nombre d'électrons collectés est proportionnel à la quantité de lumière reçue.

8. Le capteur CMOS (Complementarity metal-oxide-semiconductor) se présentent sous forme de matrice de pixels qui vont permettre d'échantillonner une scène. Il est composé de photodiodes, à l'instar des CCD, où chaque photosite possède son propre convertisseur charge/tension et amplificateur (dans le cas d'un capteur APS (Active-Pixel Sensor)).

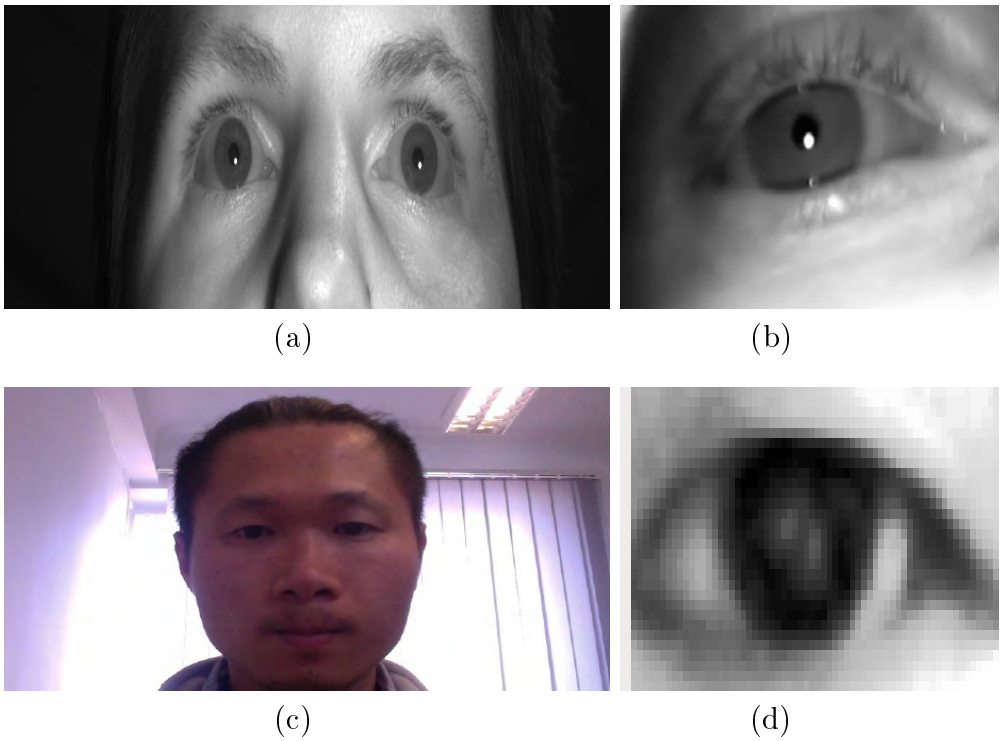
- Le processeur de signal numérique : DSP(Digital Signal Processor)
  C'est un micro processeur optimisé pour exécuter des applications de traitement numérique du signal le plus rapidement possible. Le DSP est un composant important d'un appareil photographique numérique. Il est constitué d'une combinaison de processeurs (hardware) et d'algorithmes (software). Le processeur d'images rassemble les informations de luminance et de chrominance de chacun des pixels et les utilise pour calculer/interpoler les valeurs correctes de couleur et de brillance de chaque pixel. Le processus peut comprend plusieurs opérations différentes comme le dématriçage, la réduction de bruit, la mise au point de l'image, etc. Si le DSP fait bien son travail, le résultat est une image avec des couleurs naturelles et plaisantes, un contraste équilibré et une finesse appropriée. Avec le nombre toujours croissant de pixels dans les capteurs d'image, la vitesse du processeur d'images devient de plus en plus cruciale pour les applications. Par conséquent, les processeurs d'images doivent être optimisés pour traiter plus de données dans un temps plus court.

La qualité de la webcam dépend de la résolution et de la qualité de l'image numérique convertie par le capteur photographique, de la vitesse de son convertissement, de la capacité et de la rapitidé du DSP. En outre, l'interface de connection de la webcam peut jouer un rôle important parce que la vitesse de la transmission des images à l'ordinateur est cruciale pour les applications. Actuellement les webcams au marché sont équipées du capteur CMOS et d'une interface usb 2.0. La fréquence peut atteindre 30 fps pour une image de 640×480 en pixels. Le prix n'est pas élevé, entre 20 et 100 euros. Mais nous pouvons également trouver des webcams de haut qualité comme la Thorlabs camera [9] et l'ImagingSource [10], qui sont spécialisées dans les oculomètres "low-cost" mais peuvent aussi coûter jusqu'à 3000 euros.

2.4.4 Méthodologie

Généralement l'image capturée par une webcam normale ne nous permet pas d'obtenir les deux caractéristiques principales comme la pupille et les images de Purkinje qui sont beaucoup utilisées dans l'approche basée sur le traitement des caractéristiques. Cependant il existe d'autres caractéristiques apparentes ou latentes dans l'image des yeux qui peuvent être utilisées pour concevoir un système oculométrique. L'*apparence* d'une image peut être décrite par la distribution des intensités des pixels de l'image. Le principe de notre approche, basée sur un modèle d'apparence des yeux, est d'extraire les caractéristiques des yeux de manière implicite. Les techniques de réduction de la dimensionnalité sont souvent utilisées pour constuire un modèle de variation de l'objet en analysant les vecteurs d'images, comme l'ACP (Analyse en Composantes Principales)[Pentland et al., 1994]. Le vecteur de faible dimension qui est générée est capable de coder ou caractériser l'objet. Une autre technique pour coder la description d'objet est d'utiliser le descripteur de primitives comme le descripteur de Haar proposé dans les travaux de [Viola and Jones, 2001] et [Lienhart and Maydt, 2002]. Enfin pour estimer le point du regard PoR, on établit une correlation entre ces caractéristiques latentes et la position du regard par une fonction de regression.

9. http ://www.thorlabs.de/navigation.cfm ?guide_id=2025
10. http ://www.theimagingsource.com/en_US/products/oem-cameras



(a) (b)

(c) (d)

Figure 2.24 – La différence des images sources capturée par des caméras différentes. a)b) (*source : Gazegroup.org*) sont les images de haute résolution qui permettent de distinguer la pupille et la réflexion infra-rouge. c) est une image source de la webcam en 640×480 pixels. La région de l'œil en 80×40 pixels (d) présente moins de détails de l'œil que l'image (b).

La Figure 2.24 illustre la différence des images utilisées dans ces deux approches différentes : celle basée sur le traitement des caractéristiques et celle qui s'appuie sur un modèle d'apparence des images. Une caméra de bonne qualité qui se met près du sujet permet de capturer les images de haute résolution et permet de bien localiser la pupille et le reflet de la source infra-rouge. Une webcam standard qui se situe 70cm du sujet donne des détails moins précis sur la région des yeux, et de plus, la dimension de l'image sur la région des yeux est souvent plus petite.

Généralement un système d'oculométrie basé sur l'apparence des images regroupe deux méthodes :

- **Méthode de l'extraction du descripteur des yeux.**
  Des bruits sont souvent présents dans l'image à cause de la distance, de la lumière et de la qualité de la caméra. Pour les éliminer au maximum, différents traitements de l'image sont proposés, comme le filtrage[Williams et al., 2006], le lissage[Baluja and Pomerleau, 1994], ou la méthode d'apprentissage de variété pour obtenir les caractéristiques latentes[Tan et al., 2002]. L'extraction des contour de l'œil et de la pupille est possible même dans l'image de résolution faible, et ces informations peuvent ajoutées pour former un vecteur du descripteur mixe[Fukuda et al., 2011].

- **Méthode de régression.**
  Après avoir obtenu un nombre d'exemples d'images de l'œil durant la calibration, nous pouvons utiliser une méthode de régression ou d'interpolation

pour estimer le point du regard. Ce peut être un apprentissage supervisé ou semi-supervisé. [Baluja and Pomerleau, 1994] ont proposé d'utiliser un réseau du neurone pour résoudre le problème de correlation entre les images des yeux (15×30) et les coordonnées du point du regard sur l'écran. D'autres travaux similaires basés sur les réseaux de neurones peuvent être trouvés dans [Xu et al., 1998][Stiefelhagen et al., 1997]. Pour avoir une bonne précision de l'estimation du PoR, cette méthode nécessite un grand nombre d'exemples d'apprentissage. En conséquence le calcul devient coûteux. [Baluja and Pomerleau, 1994] ont utilisé 2000 exemples pour obtenir une précision de 1.5°. [Williams et al., 2006] ont proposé une technique d'estimation du regard en temps réel par une méthode d'interpolation semi-supervisée à la base de processus gaussien. Ils ont besoin de 91 exemples (16 étiquetés et 75 non-étiquetés) pour avoir une précision de 0.83° environ. [Cadavid et al., 2009] ont utilisé la régression spectrale pour réduire la dimension des images des yeux, et le SVM (Support Vector Machine) pour estimer le PoR. Les autres approches peuvent être trouvées chez [Sugano et al., 2008][Zhang et al., 2011][Nguyen et al., 2009][Sheela and Vijaya, 2011].

L'avantage de l'approche basée sur l'apparence des images est sa simplicité pour la préparation et la configuration du dispositif. Le coût du matériel est moins élevé, et généralement la calibration de la caméra ou du système n'est pas nécessaire. Mais la limite est que le système nécessite un nombre minimal de points de calibration pour obtenir des exemples d'apprentissage permettant d'atteindre une précision pertinente. Enfin, le système doit tenir en compte des facteurs sensibles qui peuvent perturber la précision : le changement d'illumination et le mouvement de tête, etc.

Notre système a l'avantage de la simplicité, mais travailler à partir d'images bruitées et moins nettes nécessite de développer des méthodes efficaces et robustes. Nous proposons d'abord un modèle d'apparence pour extraire les caractéristiques de l'apparence de l'image de l'œil. Cette extraction est un processus de la réduction de la dimensionnalité du point de vue local. Ensuite, pour améliorer la précision du système, nous proposons d'utiliser l'apprentissage par variété sur un ensemble d'images de l'œil pour apprendre la diversité des mouvements oculaires présents dans ces images. Notre système se compose de quatre modules :

- **l'extraction des caractéristiques des yeux** par les motifs binaires locaux centrés-symétriques (CS-LBP). La méthode proposée est basée sur un modèle d'apparence de l'image et permet de caractériser le motif local autour de chaque pixel de l'image. Nous divisons l'image des yeux en blocs et combinons les histogrammes CS-LBP pour former un vecteur de signature des yeux de faible dimension. Ce vecteur de caractéristique est facile à calculer et résiste au changement d'illumination. Il est utilisé non seulement pour discriminer l'œil et d'autres objets pendant la phase de localisation des yeux, mais également pour distinguer les différents mouvements oculaires dans la phase d'estimation du regard.

- **la détection et le suivi des yeux** dans la séquence d'images capturées par la webcam. Dans un premier temps, un modèle à formes actives (ASM) et une carte des yeux (EyeMap) sont appliqués dans la première image pour localiser les composantes faciales, surtout les yeux. Une fois la localisation faite, le filtre particulaire est utilisé pour suivre le déplacement de l'œil dans les images suivantes

selon un algorithme stochastique. Cette méthode permet de détecter et suivre les yeux plus efficacement et de rélocaliser rapidement les yeux quand ils sont perdus à cause des mouvements du sujet.

- **la technique de réduction de la dimensionnalité non-linéaire** pour analyser les mouvements oculaires selon les variétés (manifolds) d'un ensemble d'images de l'œil. Le Laplacian Eigenmaps, une méthode fondée sur le laplacien du graphe, permet de trouver une représentation de faible dimension en préservant les propriétés locales des données (images). Appliquer cette approche sur l'ensemble des images de l'œil nous permet de déterminer la structure intrinsèque qui décrit la variation des mouvements oculaires. Concrètement l'analyse de la variété contribue à réaliser une calibration automatique qui est cruciale pour le fonctionnement du module d'estimation du regard.

- **l'estimation du regard par l'apprentissage supervisé**. Nous proposons deux méthodes différentes selon les besoins de l'application : une méthode de régression par processus gaussien pour estimer le regard en coordonnées 2D, dont les valeurs sont un ensemble continu de réels $\mathbb{R}$; une méthode qui utilise la classification spectrale pour classifier le regard dans les classes définies correspondant à certains types des mouvement oculaires prédéfinis.

Si nous considérons le système comme une boîte noire, l'entrée est l'image de la webcam, la sortie est la direction du regard sous forme de coordonnées 2D ou de catégories de mouvements oculaires. Les quatre modules sont indispensables dans cette boîte noire et ne travaillent pas indépendamment (Figure 2.25). Par exemple, l'extraction des caractéristiques des yeux est non seulement utilisée pour identifier la présence des yeux durant la localisation des yeux sur l'image, mais également utilisée comme un vecteur significatif pour estimer le regard. Nous allons présenter les quatre modules dans les chapitres suivants.

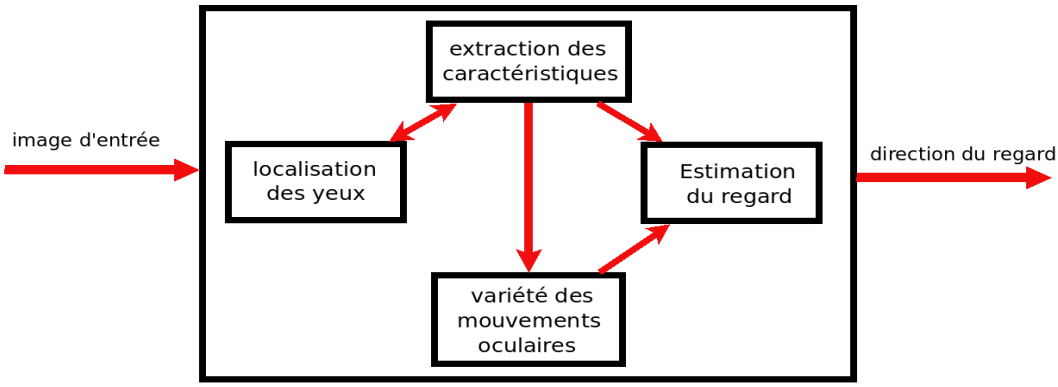


Figure 2.25 – Le schéma de quatre modules développés dans le système proposé.

2.4.5 Bibliothèques de développement

La réalisation d'un système oculométrique regroupe un ensemble de techniques et méthodes dans plusieurs domaines tels que le traitement d'image, la vision par ordinateur, l'apprentissage automatique, l'intelligence artificielle. La conception du système et le développement de programmes sont une partie indispensable de notre travail. Aujourd'hui nous disposons de beaucoup de techniques et outils pour traiter les images et les vidéos. Nous pouvons faire le traitement d'image avec n'importe quel language, vu que c'est souvent de la manipulation de tableaux de données. Le choix d'un language pour une application du traitement d'image ou de vision par ordinateur dépend de plusieurs facteurs : simplicité du codage, performance, lisibilité, disponibilité de bibliothèque, des algorithmes, et aussi de la source des images (l'acquisition depuis une webcam, un flux réseau ou un fichier), etc. Nous avons choisi le C++ comme language du développement pour notre système, parce que le C++ est performant pour une programmation bas niveau, et la plupart des bibliothèques de traitement d'image sont en C++.

Video4Linux ou V4L est une interface de programmation (API) très répandue pour l'acquisition des images ou vidéos depuis la webcam. V4L est intégré dans le noyau linux. Il permet notamment la capture de flux vidéos et d'images en provenance de caméscopes numériques, de cartes d'acquisition vidéo, de tuners TV et radio, de webcams, etc. Il existe deux versions de l'API, Video4Linux (dans les anciens noyaux Linux 2.2, 2.4 et 2.6) et plus récemment Video4Linux2, évolution majeure en standard dans les noyaux Linux 2.6.

En plus de la bibliothèque pour communiquer avec la webcam, nous distinguons aussi les bibliothèques généralistes fournissant des boîtes à outils orientées image, les bibliothèques de programmation, de l'apprentissage, et les logiciels de visualisation, etc. **OpenCV** (http ://opencv.org) est la bibliothèque de base de notre développement. Crée par intel, et depuis 2008 gérée par la société de robotique Willow Garage, OpenCV est une bibliothèque libre du traitement d'image, et de vision par ordinateur. Nous utilisons les autres opensource comme **Itpp** (http ://itpp.sourceforge.net) et **VXL** (http ://vxl.sourceforge.net) pour résoudre les opérations matricielles, et **CImg**(http ://cimg.sourceforge.net) et **Rebol**(http ://www.rebol.com) pour les interfaces.

Nos développements informatiques pour le système oculométrique peuvent se diviser en 3 classes :

- Traitement des images des yeux : les développements de programmation pour la détection des yeux et l'extraction des caractéristiques.

- Suivi d'un objet : la méthode pour suivre un objet dans une séquence d'images par le filtrage particulaire.

- Apprentissage : les méthodes de réduction de la dimensionnalité non-linéaire comme le Laplacian Eigenmaps, le Diffusion Maps et les méthodes comme la classification spectrale et la régression par processus gaussien.

Troisième partie

Détection et suivi des yeux

Sommaire

3.1 Introduction

Dans ce chapitre nous allons présenter le module de détection et de suivi des yeux, dont l'objectif est de trouver la position exacte des yeux dans toutes les images capturées par la caméra (ou dans une vidéo). Dans la section (2.3.1.1) nous avons présenté deux approches de méthodes pour détecter et suivre des yeux. Généralement les techniques utilisent souvent les propriétés géométriques des yeux (la forme, le contour) et les caractéristiques des yeux (la pupille, l'iris et le reflet) pour localiser les yeux.

Etant donné que notre système oculométrique utilise seulement une webcam à distance et sans lumière infra-rouge, il ne nous permet pas d'utiliser les techniques conventionnelles. L'environnement expérimental est souvent le suivant : le sujet est devant la caméra avec une distance d'environ 60-80 cm ; la tête du sujet est libre ; nous utilisons l'illumination naturelle ou de l'intérieur pour l'éclairage. Donc plusieurs considérations doivent être prises en compte pour localiser l'œil.

- Changement d'illumination : La lumière naturelle est utilisée pour l'éclairage. Tant qu'il y a du changement de la lumière, l'apparence de l'image capturée va également changer (voir Figure 3.1). Pour localiser l'œil, la méthode doit résister au changement de luminosité.

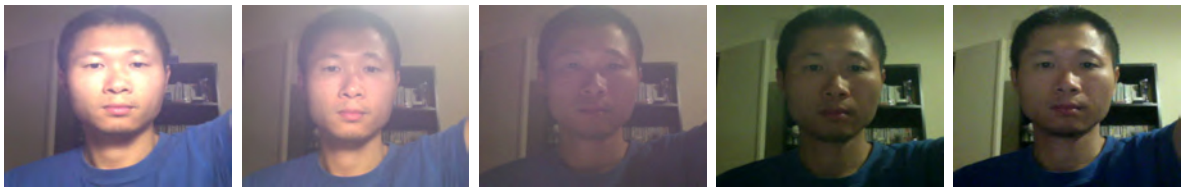


FIGURE 3.1 – Effets du changement de luminance sur la qualité des images à traiter

- Changement de l'apparence de l'œil. La forme de l'œil peut être très variée à cause du comportement du sujet et de la position de sa tête (voir Figure 3.2). Nous devons trouver une solution pour distinguer ces formes variées et en même temps discriminer la région de l'œil des autres régions. En outre, chaque fois qu'on perd l'œil dans l'image pendant la phase du suivi, nous devons être capables de le récupérer plus tard dès qu'il apparait de nouveau dans l'image.

FIGURE 3.2 – Effets du changement de position de la tête sur la qualité des images à traiter

- Performance. La méthode doit localiser les yeux très rapidement dans la séquence d'images capturées par la webcam avec le plus court temps de latence possible. La fréquence d'une webcam normale est de 30 fps. Généralement pour les expérimentations du système oculométrique, une fréquence au-dessus de 15 fps, après avoir appliqué la méthode de localisation, est suffisante pour une application en temps réel.

La capture de la caméra, ou la vidéo génèrent une séquence d'images en fonction du temps t. Généralement nous distinguons deux stratégies pour localiser l'objet dans cette situation :

- effectuer une méthode de détection de la cible sur chaque image capturée par la caméra. La méthode ne dispose pas de connaissance *a priori* sur chaque image avant qu'elle soit capturée. Chaque image est indépendante des autres et la méthode est effectuée de la même manière pour localiser la cible sur chaque image ;
- détecter la région de la cible dans la première image et ensuite effectuer la méthode de suivi pour cette région sur les images suivantes. Dans ce cas, le mouvement de la cible est pris en compte et la position de la cible à l'instant t est dépendante de sa position détectée à l'instant $t-1$.

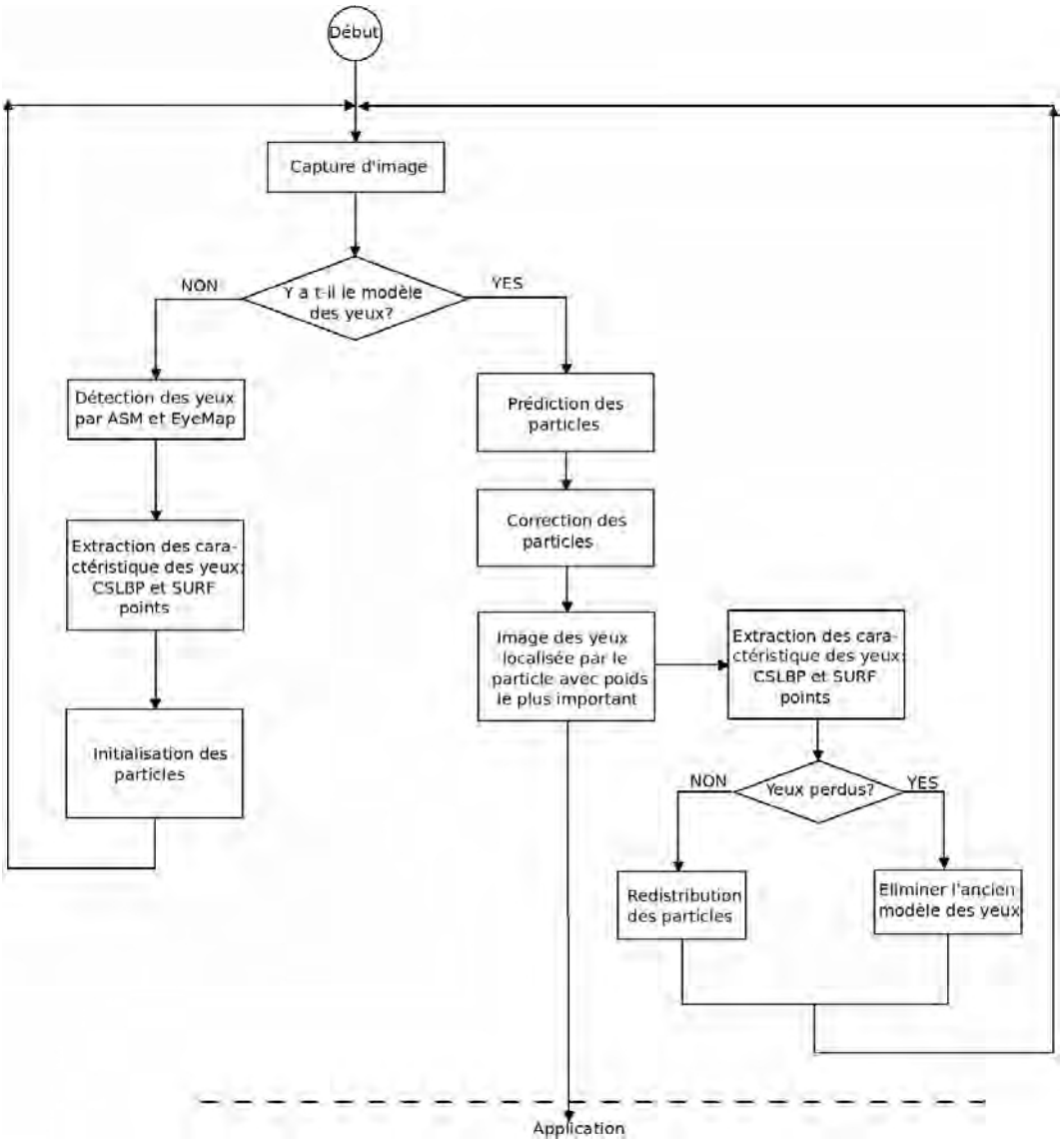


FIGURE 3.3 – Le schéma de la méthode proposée.

Notre méthode, dont le schéma (Figure 3.3) suit exactement la deuxième stratégie, applique d'abord un modèle à formes actives ASM (Active Shape Model) et une carte des yeux (EyeMap) pour localiser la région de l'œil correctement. La région de l'œil détectée est utilisée comme modèle de référence pour la méthode de suivi. Ensuite une méthode stochastique, appelée filtre particulaire (Particle Filter), est utilisée pour suivre la cible

(l'œil détecté) dans les images suivantes. Le filtre particulaire génère un ensemble de "particules". Chaque particule représente un état probable (par exemple la position de l'œil) et chacune évolue d'une manière indépendante au cours du temps t. Une mesure effectuée pour chaque particule représente le degré de confiance sur l'état associé. Pour obtenir cette mesure, qui représente la discrimination entre l'œil et d'autres objets, nous allons extraire les caractéristiques de l'apparence de l'image de l'œil par une variante des motifs binaires locaux CS-LBP (en. Center-Symmetric Local Binary Pattern) ainsi que les points d'intérêt par la méthode SURF (Speeded-Up Robust Features).

Dans les sections suivantes, nous allons présenter d'abord la méthode de détection de l'œil (3.2). Ensuite nous expliquerons la méthode de l'extraction des caractéristiques de l'œil par les motifs binaires locaux centrés-symmétriques (3.3). Dans la section 3.4, nous présenterons le filtre particulaire pour suivre l'œil dans la séquence d'images.

3.2 Détection des yeux

L'objectif de la méthode présentée dans cette section est de détecter la région de l'œil dans une image. D'abord nous utilisons un modèle à formes actives (ASM) pour chercher les composantes faciales dès que le visage est détecté dans l'image. Par ce modèle, nous pouvons localiser la région susceptible de contenir l'œil. Ensuite une carte des yeux (EyeMap) est appliquée dans cette région pour trouver l'endroit où on observe le plus fort contraste. Cet endroit peut être considéré comme la position exacte de l'œil.

3.2.1 Modèle à formes actives

Le modèle à formes actives (Active Shape Model, ASM) est une technique de segmentation introduite initialement par [T.F.Cootes et al., 1995], pour localiser des objets déformables dans des images médicales. L'ASM consiste à déformer itérativement un contour initial afin qu'il se positionne sur le contour de l'objet d'intérêt. C'est un outil de segmentation robuste et fiable permettant l'intégration des connaissances *a priori* sur les formes à l'aide d'un modèle de distribution de points (PDM). Les connaissances sont déduites, à partir de M échantillons d'images présentant les variations possibles de la structure étudiée. L'utilisation de l'ASM dans plusieurs applications prouvent son efficacité. En effet, il a l'avantage de tenir en compte la variabilité anatomique et de gérer automatiquement le changement topologique. L'ASM nécessite toutfois un ensemble de d'apprentissage sur l'objet. En plus, ses résultats ne sont pas satisfaisants dans le cas d'images bruitées, de faible contraste et/ou de faible résolution.

Le modèle à formes actives se compose de deux phases : une phase hors-ligne de construction du PDM par apprentissage, et la phase de localisation de l'objet d'intérêt dans l'image en entrée.

3.2.1.1 Phase de construction

La phase de construction du PDM consiste en premier lieu, à étiqueter manuellement un contour sur l'objet d'intérêt pour chaque image i de la base d'apprentissage (de taille M), par exemple sur la base de données du visage (Figure 3.4 gauche). Sur chaque image, n points caractéristiques sont positionnés sur le contour de la région d'intérêt (Figure 3.4 droite). Chaque forme V_i sera alors modélisée par un vecteur de dimension $2n$ constuit par concaténation des coordonnées des points placés sur son contour :

$$V_i = (x_{i1}, y_{i1}, x_{i2}, y_{i2}, ..., x_{in}, y_{in}) \qquad \forall i \in M$$

avec (x_{ij}, y_{ij}) les coordonnées du point j dans l'image i.

Ensuite un algorithme itératif permet d'aligner les formes et d'obtenir la forme moyenne $\overline{V}$. La détermination de la forme moyenne ainsi que le calcul des principaux modes et amplitudes de déformation sont réalisés par une Analyse en Composantes Principales (ACP). Finalement, un modèle de forme est défini en décrivant les variations de la forme par l'équation :

$$V = \overline{V} + P_f b_f$$

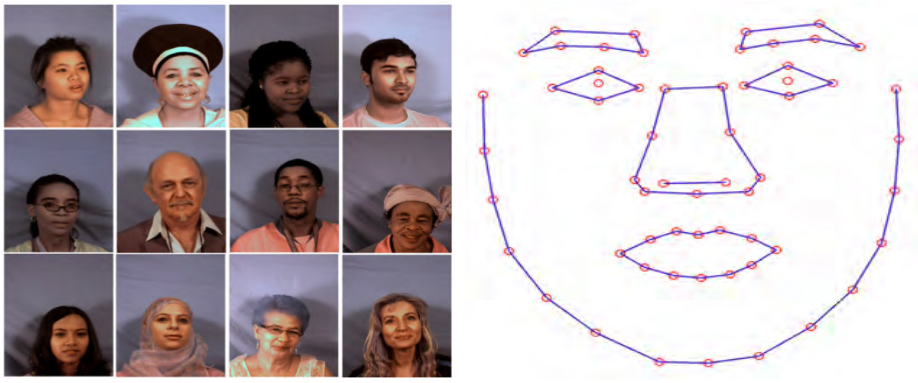


FIGURE 3.4 – Gauche : les exemples de la base de données du visage MUCT [Milborrow et al., 2010] qui contient 3755 images étiquetées avec les points caractéristiques. Droite : 60 points caractéristiques pour décrire le visage.

Ici les variations peuvent être représentées en fonction de la forme moyenne et les principales directions de variations, soit les vecteurs propres de la matrice de covariance C,

$$C = \frac{1}{M}\sum_{i=1}^{M}(V_i - \overline{V})(V_i - \overline{V})^T.$$

$\overline{V}$ est la forme moyenne. $P_f = (p_1, p_2, ..., p_t)$ est la matrice des t vecteurs propres significatifs. $b_f = (b_1, b_2, ..., b_t)^T$ est la matrice des poids représentant la projection de la forme V dans la base P_f. Les valeurs b_i varient classiquement entre $-3\sqrt{\lambda_i}$ et $3\sqrt{\lambda_i}$ où λ est la valeur propre.

3.2.1.2 Phase de localisation

La phase de localisation s'appuie sur ce modèle PDM pour localiser la structure recherchée. Elle consiste à commencer par une estimation initiale à partir de la forme moyenne. Cette estimation est ensuite déformée itérativement vers les frontières de l'objet étudié en utilisant les propriétés de luminance de l'image. A chaque itération, la nouvelle forme obtenue doit appartenir à l'espace autorisé imposé par le modèle de forme et ce jusqu'à convergence.

L'algorithme de la méthode peut être résumé comme suit :

- Calculer un profil pour chaque point caractéristique de la forme moyenne pendant la phase d'apprentissage.
- Positionner la forme moyenne au plus près de la région d'intérêt.
- Répéter jusqu'à satisfaction de la condition de convergence ou épuisement d'un nombre maximum d'itérations.
  - Rechercher, le long de chaque normale de la forme précédemment élaborée, le profil correspondant au mieux à celui calculé pour la forme moyenne. Les nouveaux points de marquage sont les points centraux des profils ainsi trouvés.
  - Rechercher le modèle de forme s'adaptant le mieux aux points trouvés à l'étape précédente. Celui-ci constituera la forme de départ pour l'itération suivante.

Les profils sont des vecteurs calculés à partir de la texture environnante, le long de la droite orthogonale au contour, en chaque point caractéristique de la forme moyenne (Figure 3.5).

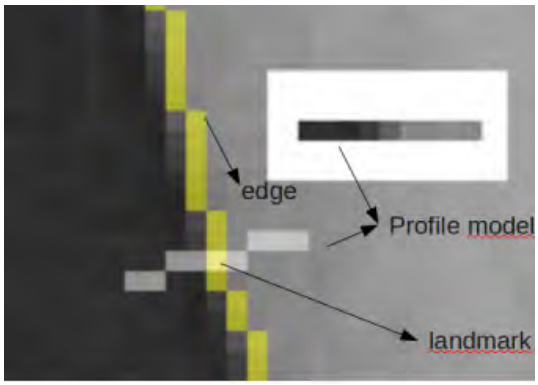
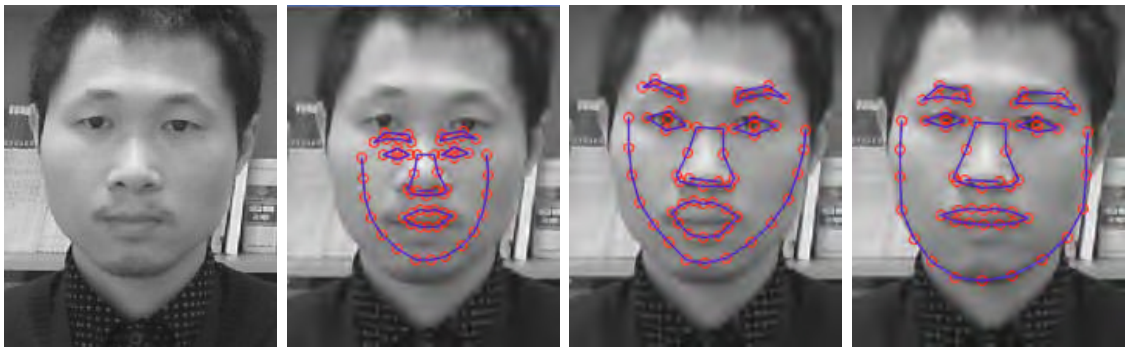


Figure 3.5 – Le profil d'un point caractéristique



a) b) c) d)

Figure 3.6 – Déplacement des points caractéristiques. a) l'image originale b) la première itération c) après 30 itérations d) après 38 itérations

La comparaison entre les profils est réalisée à l'aide de la distance de Mahalanobis qui est basée sur la corrélation entre des variables par lesquelles différents modèles peuvent être identifiés et analysés. Elle permet de déterminer la similarité entre deux jeux de données et est définie par :

$$distance = (g - \overline{g})^T C_g^{-1} (g - \overline{g})$$

où g est le profil construit lors de la recherche, $\overline{g}$ est le profil associé à la forme moyenne et C_g est la matrice de covariance des profils relatifs au point de référence. Le profil le plus similaire au profil de la forme moyenne est celui qui minimisera cette distance. Ainsi, chacun des points de référence est déplacé à chaque itération vers le point marqué sur la normale dont son profil est le plus similaire au profil de la forme moyenne au sens de la distance de Mahalanobis.

La condition de convergence choisie consiste à laisser l'algorithme itérer jusqu'à ce que seulement un faible pourcentage de points de référence continue à se déplacer. Dans le cas où la recherche continue indéfiniment, un nombre maximum d'itérations est fixé pour stopper la recherche et le dernier résultat est donné comme solution finale.

FIGURE 3.7 – Les résultat de ASM sur les exemples des images de la base de données de l'institut de Technologie Géorgie.

La Figure 3.6 illustre la déformation du modèle des points pour conformer au visage dans l'image après une trentaine d'itérations. Dans notre expérimentation, nous utilisons la base de données de l'institut de Technologie Géorgie qui contient les images du visage sur 34 individus. La Figure 3.7 montre les résultats de notre implémentation d'ASM sur les exemples des images.

3.2.2 EyeMap

Les yeux peuvent être détectés non seulement par leur forme, mais également par leur apparence dans l'image. La carte des yeux (EyeMap) est une méthode souvent utilisée pour la détection des yeux en transformant l'image RGB dans l'espace de couleur $YCrCb$[Hsu et al., 2002].

$YCrCb$ est un modèle de représentation de l'espace colorimétrique, où Y est le signal de luminance (noir et blanc). Cr (le rouge moins Y) et Cb (le bleu moins Y) sont les deux informations de chrominance. Pour calculer les valeurs des composantes $YCrCb$ d'une image à partir des composantes RGB (qui varient de 0 à 255), on utilise les formules suivantes :

$$\begin{cases} Y = 0.257 * R + 0.504 * G + 0.098 * B + 16 \\ Cb = -0.148 * R - 0.291 * G + 0.439 * B + 128 \\ Cr = 0.439 * R - 0.368 * G - 0.071 * B + 128 \end{cases}$$

Hsu et al. [Hsu et al., 2002] ont proposé de construire la carte des yeux par la combinaison des deux cartes différentes : EyeMapC qui est calculée sur les composantes de chrominance et EyeMapL qui représente les composantes de luminance.

- **EyeMapC :**
  l'analyse des composantes de chrominance indique, qu'autour des contours des yeux, on observe des fortes valeurs de Cb et de Cr. Pour chaque pixel de l'image, on calcule EyeMapC par :

  $$EyeMapC = \frac{1}{3}\{(C_b)^2 + (\overline{C_r})^2 + C_b/C_r\}$$

  où Cb, Cr, Cb/Cr sont normalisés entre [0,255]. $\overline{C_r}$ est l'inverse de Cr (255-Cr). La Figure 3.8 illustre les résultats de EyeMapC sur les exemples des images du visage.

Figure 3.8 – Résultats de EyeMapC sur les exemples du visage. (Haute : images originales ; Bas : images EyeMapC)

- **EyeMapL :**
  l'image EyeMapL est obtenue par la division des deux images de luminance Y qui sont effectuées respectivement par les opérations de dilatation et d'érosion (voir la Figure 3.9). EyeMapL permet d'accentuer des zones d'ombres et des zones lumineuses dans l'image de la luminance.

$$EyeMapL = \frac{Y(x,y) \oplus g_\delta(x,y)}{Y(x,y) \otimes g_\delta(x,y)}$$

  où$\oplus$ et $\otimes$ sont des opérations de dilatation et d'érosion sur une fonction f avec un élément structurant circulaire g_δ.

EyeMap est générée par la combinaison des deux cartes EyeMapC et EyeMapL (Figure 3.10) :

$$EyeMap = EyeMapC(AND)EyeMapL$$

En multipliant les deux images EyeMapC et EyeMapL, nous obtenons donc une image où les yeux auront un fort contraste. La Figure 3.11 montre le résultat final de EyeMap sur les exemples des images du visage.

3.2.3 Expérimentation

Nous combinons l'ASM et EyeMap pour localiser la région des yeux dans une image capturée par Webcam. La région détectée par l'ASM nous aide à limiter la zone de recherche des yeux, et le traitement d'image EyeMap permet d'identifier la position des yeux en trouvant la zone où il y a le plus fort contraste.

La Figure 3.12 résume notre méthode. La position initiale du modèle des points caractéristiques est très importante pour le résultat de l'ASM. Si le modèle se situe loin du visage dans l'image, l'adaptation des points caractéristiques va être perturbée et donner un mauvais résultat. Pour éviter ce problème, nous utilisons la méthode de

FIGURE 3.9 – Résultats de EyeMapL sur les exemples du visage. (De gauche à droite : l'image originale, dilatation de l'image de luminance, érosion de l'image de luminance, l'image EyeMapL obtenue)

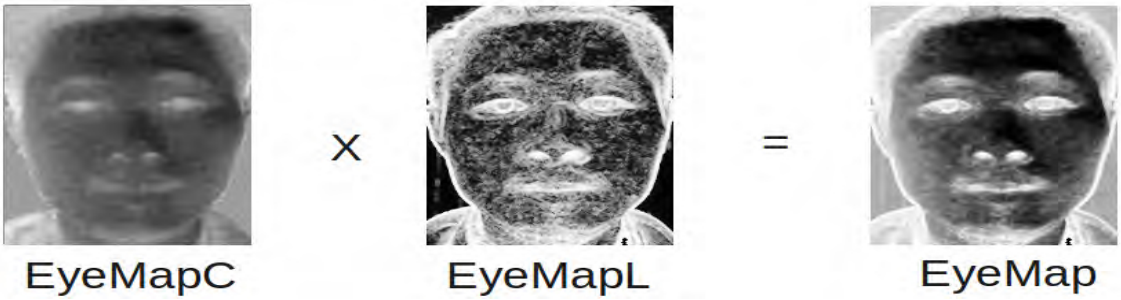


FIGURE 3.10 – Résultat du calcul de l'image

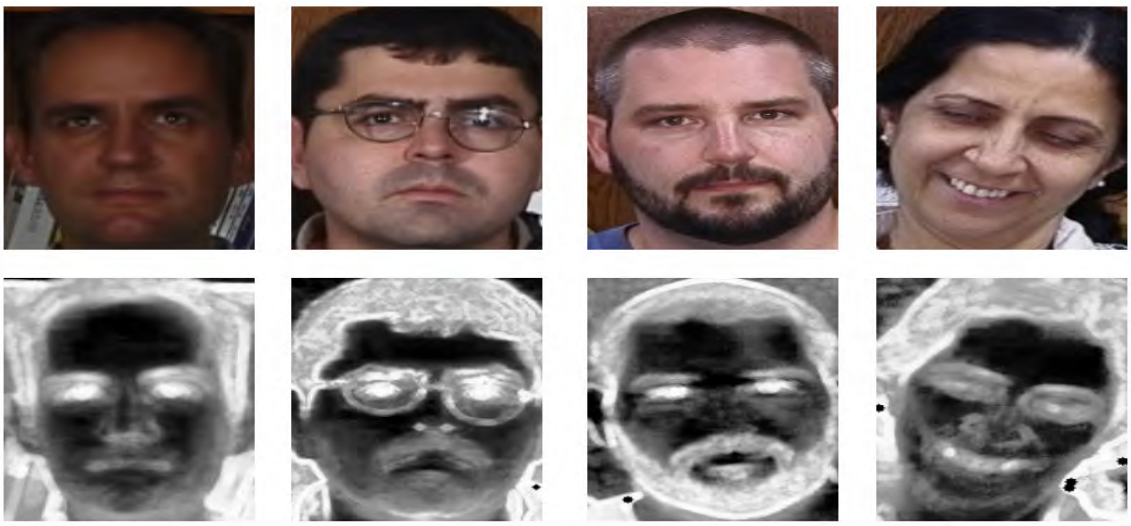


FIGURE 3.11 – Calcul de l'image EyeMap pour différents exemples de visages. (Haute : les images originaires ; bas : les images EyeMap calculées.

FIGURE 3.12 – Le schéma de la localisation des yeux

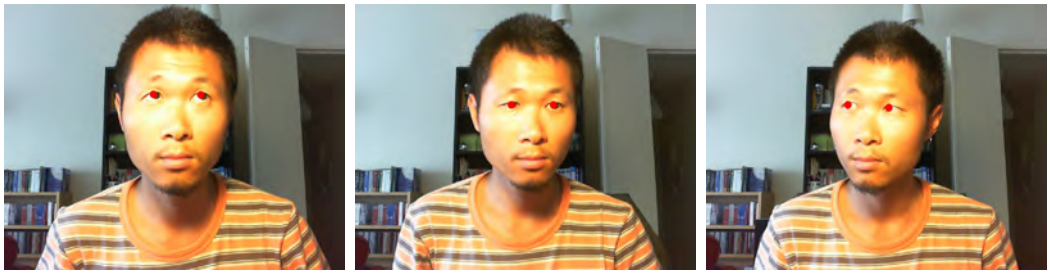
détection du visage proposée par [Viola and Jones, 2004]. Une fois que le visage est détecté, le modèle va être positionné au centre de la région et adapté à la forme du visage pour localiser les composantes faciales. Après avoir localisé la région susceptible de contenir les yeux, nous appliquons la carte des yeux (EyeMap) pour cette région et cherchons une zone circulaire où la somme des pixels est maximale.

Notre expérimentation du développement est basée sur la programmation Opencv en C++. Le développement de l'ASM est basé sur la bibliothèque opensource "*asmlib-opencv*" [11], où le modèle du visage fourni est construit à partir de la base de donnée MUCT qui contient 3755 images. L'algorithme de EyeMap et la recherche de la zone qui a le plus fort contraste appellent les méthodes Opencv de dilatation et d'érosion. Les détails sont présentés dans l'Annexe 7.

Nous avons également appliqué notre méthode sur une autre base de données de visages de l'Institut de Technologie de Géorgie qui contient 510 images de visages de 34 sujets. Nous avons 15 images de visages dans des conditions variées et complexes : rotation de tête, visage avec des lunettes, yeux fermés, couleur de la peau différente, etc. Parmi ces images, il y a 315 images de face, comme dans les exemples illustrés de la Figure 3.13.a, et le taux de réussite de la détection du visage présenté de face est 95%. Pour les images qui contiennent des positions différentes, comme dans les exemples de la Figure 3.14, la détection des yeux est difficile et parfois impossible. Par exemple les yeux dans les images de la Figure 3.14 ne peuvent pas être détectés par la méthode de [Viola and Jones, 2004] à cause de la position de la tête ou de la présence de lunettes. Par contre notre méthode réussit à détecter la région des yeux malgré les difficultés présentées sur ces images, car la carte des yeux calculée est indépendante de la mesure géométrique des yeux et permet d'identifier les yeux par la recherche de la région où il y a un contraste élevé. Nous pouvons aussi appliquer cette méthode pour localiser les yeux dans une séquence d'images fournies par une simple webcam. Cette méthode cherche les yeux de la même façon sur chaque image (voir la Figure 3.13.b). Mais la localisation des yeux peut être interrompue si jamais la position du sujet change. Donc il manque la continuité pour effectuer une localisation correcte. Par rapport au temps de calcul, sur notre machine MacBook Pro 8.1, la méthode a besoin d'environ 100 ms pour détecter les yeux à chaque image. La fréquence de la webcam que nous utilisons dans notre expérimentation est environ 30 fps sur la capture de l'image dont la résolution est 640×480 pixels. Si nous appliquons cette méthode sur la séquence d'images capturées par la Webcam la fréquence descend à environ 11-12 fps.

11. http : ://code.google.com/p/asmlib-opencv

(a)



(b)

FIGURE 3.13 – a) Les résultats de détection des yeux (points blancs) sur les exemples de la base de données de l'Institut de Technologie de Géorgie. b) La détection des yeux (points rouges) dans une séquence d'images capturées par la webcam.

FIGURE 3.14 – Les exemples où les yeux ne peuvent pas être détectés par la méthode de [Viola and Jones, 2004] (en haut) à cause de la rotation de tête ou de la présence de lunettes ; notre méthode réussit à détecter la région des yeux (en bas).

3.3 Extraction des caractéristiques des yeux

Dans les applications de détection et de suivi d'objets, une question clé est la nature des "caractéristiques visuelles" utilisées. La caractéristique (feature) est une notion générale dans le domaine de la vision par ordinateur et elle représente les propriétés distinctives de l'image qui peuvent être des segments de droite, des points, des contours, des régions, des pics de l'objet, etc. Les caractéristiques visuelles permettent généralement de mieux rendre compte de certaines propriétés visuelles de l'image qui vont être utilisées pour des traitements ultérieurs entrant dans le cadre d'applications telles que la détection d'objets ou la recherche d'images par le contenu.

En vision par ordinateur, le terme d'extraction de caractéristiques (features extraction) regroupe un ensemble de méthodes permettant d'extraire des informations à partir d'images complexes sans connaissance à priori sur l'image, informations relatives à la texture mais aussi et surtout au contenu structurel de l'image. L'extraction de caractéristiques permet de créer de nouveaux ensembles de caractéristiques, en utilisant une combinaison des caractéristiques de l'espace de départ ou plus généralement une transformation effectuant une réduction du nombre de dimensions. La sélection de caractéristiques (feature selection) regroupe les algorithmes permettant de sélectionner un sous-ensemble de caractéristiques parmi un ensemble de départ, en utilisant divers critères et différentes méthodes. La sélection des caractéristiques est une technique permettant de choisir les caractéristiques, variables ou mesures les plus intéressantes, pertinentes, adaptées à un système de résolution d'un problème particulier. L'objectif de l'extraction et de la sélection de caractéristiques est d'identifier les caractéristiques importantes pour la discrimination entre classes. Après avoir choisi le meilleur ensemble de caractéristiques, il s'agit de réduire la dimensionnalité de l'ensemble des caractéristiques en trouvant un nouvel ensemble, plus petit que l'ensemble original, qui néanmoins, contient la plupart de l'information.

Un descripteur peut être un vecteur de haute dimension qui décrit les caractéristiques. On distingue usuellement les caractéristiques globales qui sont calculées sur toute l'image et les caractéristiques locales qui sont calculées autour de points d'intérêt. Les caractéristiques locales se distinguent par le fait qu'elles sont distinctes, robustes aux occlusions (car il y en a beaucoup dans un image ou une région) et qu'elles ne nécessitent pas de segmentation. Un descripteur local, calculé pour chaque pixel d'une image ou d'une région obtenue par segmentation, puis accumulé dans un histogramme, est donc une description globale de l'image ou de la région. Généralement le descripteur idéal d'un objet quelconque devrait être :

- **Robuste** au bruit, aux conditions d'acquisition de l'image, au changement de la géométrie de l'objet (la rotation, le changement d'échelle, la translation, etc.) ainsi qu'au changmement de la photométrie (le changement d'intensité des pixels) ;
- **Discriminant** pour identifier l'objet parmi beaucoup d'autres objets différents ;
- **Efficace** pour un calcul rapide, surtout pour une application en temps-réel.

Dans notre expérimentation, l'apparence de l'œil peut être très diverse à cause de la variation des conditions d'acquisition : la distance entre le sujet et la caméra, la luminosité, la position de la tête du sujet, etc. Nous proposons un modèle d'apprence qui permet d'extraire les caractéristiques de manière globale et locale à partir de l'apparence

d'une image. Ce modèle comporte deux méthodes :

- **Les motifs binaires locaux** : cette méthode est appliquée sur toute la région d'intérêt de l'image, par exemple, la région de l'œil. Elle permet d'attribuer à chaque pixel de cette région, une valeur caractérisant le motif local autour de ce pixel. L'histogramme des valeurs de la région peut être utilisé comme descripteur qui représente cette région d'intérêt.
- **La méthode SURF** (Speeded-Up Robust Features) : cette méthode est utilisée pour détecter et suivre les points d'intérêt dans la région de l'œil qui permettent de caractériser la région autour de l'œil.

Ce modèle permet de générer les caractéristiques globales et locales de l'image de œil. La combinaison de ces caractéristiques peut être discriminante pour identifier la présence de l'œil dans l'image. Nous pouvons utiliser ce modèle d'apparence non seulement pour localiser l'œil dans la séquence d'images, mais également pour le module d'estimation du regard.

3.3.1 Motifs binaires locaux et variantes

Les motifs binaires locaux (LBP : Local Binary Patterns) ont initialement été proposés par Ojala et al. dans les années 90 afin de caractériser les textures présentes dans des images en niveaux de gris[Ojala et al., 1994, Ojala et al., 1996]. Ils consistent à attribuer à chaque pixel P de l'image $I(i, j)$, une valeur caractérisant le motif local autour de ce pixel. Ces valeurs sont calculées en comparant le niveau de gris du pixel central P aux valeurs des niveaux de gris des pixels voisins. Cette méthode d'analyse présente une complexité calculatoire faible et permet de différentier des contenus texturaux ; elle est aussi considérée comme une approche unifiant l'approche statistique et d'autres modèles géométriques pour l'analyse de texture.

La méthode LBP est devenue très populaire et très employée pour différentes applications en vision par ordinateur. Elle s'est révélée très efficace pour la classification d'images texturées comme dans les applications de reconnaissance de visage. Elle a été appliquée à la segmentation de la texture dans les images [Heikkilä et al., 2009, Ojala et al., 2002, Maenpaa and Pietikäinen, 2005], et elle est beaucoup utilisée dans le domaine du "tracking" des objets en vidéo et du diagnostic médical. Pour plus de détails sur cette méthode et ses différentes applications on peut se rapporter à cet ouvrage[Pietikäinen et al., 2011]. Grâce à sa flexibilité, la méthode LBP peut être facilement adaptée pour les besoins de divers problèmes. Plusieurs extensions et modifications de la technique LBP ont été proposées dans le but d'augmenter sa robustesse et son pouvoir discriminant.

3.3.1.1 LBP basique

La version originale de l'opérateur LBP est effectuée sur chaque bloc de 3×3 pixels dans l'image. Le concept du LBP est simple : il propose d'assigner un code binaire à un pixel en fonction de son voisinage. Ce code décrivant la texture locale d'une région est calculé par seuillage d'un voisinage avec le niveau de gris du pixel central. Afin de

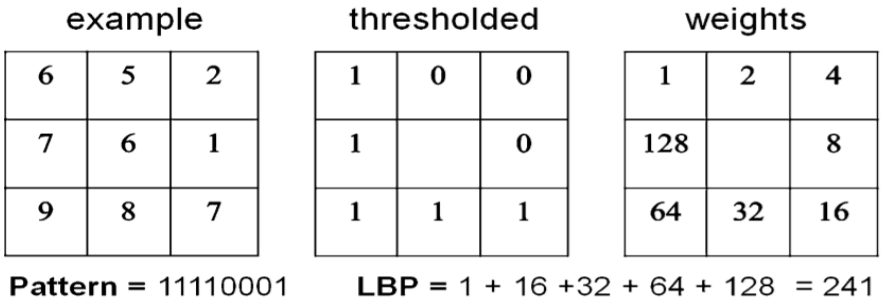


FIGURE 3.15 – Un exemple de calculer LBP basique.

générer un motif binaire, tous les voisins prendront alors une valeur "1" si leur valeur est supérieure ou égale au pixel courant et "0" autrement (Figure 3.15). Les pixels de ce motif binaire sont alors multipliés par des poids et sommés afin d'obtenir un code LBP du pixel courant. On obtient donc pour toute l'image, des pixels dont l'intensité se situe entre 0 et 255 comme dans une image 8 bits. Pour décrire l'image par la séquence des motifs LBP, des histogrammes d'apparition de ces motifs sur l'ensemble de l'image ou une région de l'image peuvent ensuite être calculés, et on peut les choisir comme descripteur de texture de dimension 255.

Plusieurs années après cette première publication, Ojala et al.[Ojala et al., 2002] ont proposé une variante de cette méthode qui formule cet opérateur par une manière générique, appelée LBP multi-échelle. Le concept du LBP multi-échelle, est fondé sur le choix du voisinage afin de calculer un code LBP pour pouvoir traiter les textures à différentes échelles. Un voisinage pour un pixel central est réparti sur un cercle et construit à partir de deux paramètres : le nombre de voisins P sur le cercle et un rayon R pour définir une distance entre un pixel central et ses voisins.

Etant donnée une image $I(x, y)$ en niveau de gris, g_c désigne la valeur de niveau de gris du pixel central et $g_c = I(x_c, y_c)$ où x_c et y_c sont les coordonnées de g_c. On définit g_p qui désigne la valeur de niveau de gris de P pixels espacés régulièrement sur un cercle de rayon R autour de g_c. Les coordonnées de g_p sont données par l'équation suivante :

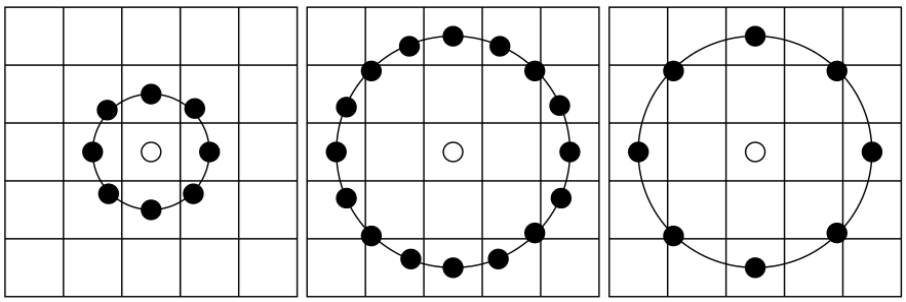
$$\begin{aligned} g_p = I(x_p, y_p), p = 0, ..., P-1 \qquad et \\ x_p = x_c + Rcos(2\pi p/P), \\ y_p = y_c - Rsin(2\pi p/P). \end{aligned}$$

La Figure 3.16 illustre différents voisinages obtenus pour différentes valeurs du couple (P, R). Nous pouvons remarquer que les coordonnées d'un voisin ne sont pas forcément situées au centre d'un pixel. Dans ce cas, le niveau de gris est déterminé par l'intermédiaire d'une interpolation.

Donc cet opérateur générique $LBP_{P,R}$ est défini :

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c)2^P)$$

où $s()$ est une fonction de signe $s(t) = \begin{cases} 1 & t \geqslant T \\ 0 & else \end{cases}$ et le seuil T peut être une valeur petite comme 0.



(P=8, R=1) (P=16,R=2) (P=8,R=2)

FIGURE 3.16 – Le voisinage défini par le couple (P, R), où R est le rayon du cercle et P est le nombre des pixels espacés régulièrement sur un cercle de rayon R. La valeur des pixels est calculée par une interpolation bilinéaire.

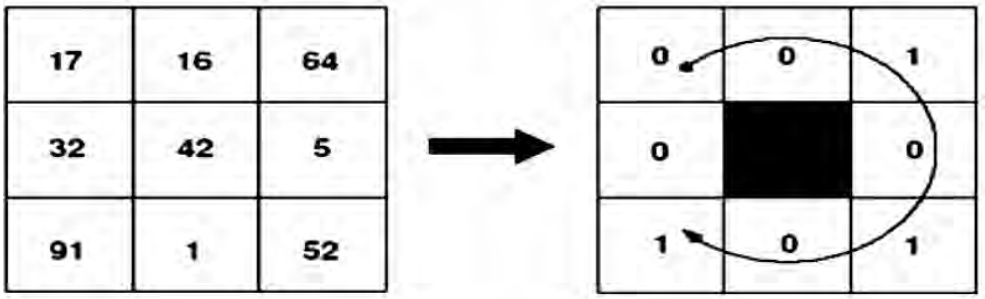


FIGURE 3.17 – Le nombre de transition du motif présenté est 6, donc c'est un motif non-uniforme.

Partant de la définition de voisinage, les auteurs définissent tout d'abord un motif binaire local invariant à toute transformation monotone de l'échelle en niveaux de gris, $LBP_{P,R}$. Pour chaque pixel (x, y) $(g_c = I(x, y))$, comme dans la méthode LBP de la version originale, le pixel central n'est pas utilisé pour la caractérisation des textures. En effet, indépendamment du voisinage g_p, ce pixel décrit uniquement une intensité lumineuse, ce qui n'est pas forcément utile[Ojala et al., 2002]. La méthode LBP de la version originale est similaire avec $LBP_{8,1}$, mais il y a de la différence : les pixels voisins sont indexés par une façon ciculaire, qui est plus avantageux pour générer un descripteur invariant à la rotation. En plus la valeur des pixels voisins peut être calculée par une interpolation bilinéaire (voir Annexe 3).

3.3.1.2 LBP uniforme

Le motif binaire uniforme est la deuxième variante proposée par [Ojala et al., 2002]. Ici, la mesure de l'uniformité pour un motif binaire est définie par "U", qui représente le nombre de transitions bit à bit de "0" à "1" ou vice versa dans un parcours circulaire comme le montre l'exemple de la Figure 3.17 où le nombre de transitions du motif $U = 6$. Un motif binaire local est uniforme si et seulement si $U \leqslant 2$. Par exemple, le motif dans la figure 3.17 (00101010) qui a 6 transitions n'est pas un motif uniforme ; les motifs 00000000 (0 transitions), 01110000 (2 transitions) et 11001111 (2 transitions) sont uniformes ; alors les motifs 11001001 (4 transitions) et 01010011 (6 transitions) sont non-uniformes. Cette notion d'uniformité est importante dans la méthode LBP

pour représenter les informations de primitives structurelles comme les arêtes (coins) et les contours.

La valeur de l'opérateur LBP sur un pixel (x, y) correspond à la somme des bits du voisinage et le nombre des motifs différents dépend du voisinage P. Pour l'opérateur LBP original sur un voisinage $(8, R)$, nous avons 2^P motifs différents. Le nombre des motifs binaires locaux uniformes est calculé par $P(P-1)+3$. Dans le mappage des valeurs, chaque LBP uniforme a sa propre étiquette unique, alors que tous les LBP non-uniformes sont affectés à une seule étiquette. Par exemple, pour le voisinage $(8, R)$, il y a 58 LBP uniformes parmi les 256 motifs. Si nous ajoutons une dernière étiquette pour les LBP non-uniformes, nous obtenons 59 étiquettes au total, comme le montre la Figure 3.18. Dans ce cas, la dimension de l'histogramme LBP peut être réduite de manière importante avec un histogramme de dimension 59. Chacune des 58 premières catégories contiendra le nombre d'occurrences de l'un des motifs uniformes. La dernière contiendra le nombre d'occurrences de tous les motifs non-uniformes. Ce regroupement permet de réduire la dimension du descripteur sans perdre trop d'information.

Il y a deux raisons pour lesquelles nous ignorons les motifs non-uniformes. Tout d'abord, la plupart des motifs binaires locaux dans les images sont uniformes. Ojala et al. ont constaté que dans leurs expériences avec des textures, les motifs uniformes représentent un peu moins de 90% de tous les motifs pour un voisinage (8, 1) et environ 70% pour un voisinage (16, 2). Dans des expériences de reconnaissance du visage [Ahonen et al., 2006] les motifs uniformes atteignent 90.6% pour le voisinage (8, 1) et 85.2% pour le voisinage (8, 2). La deuxième raison d'utiliser uniquement les motifs uniformes est la robustesse statistique. L'utilisation de motifs uniformes au lieu de tous les motifs possibles produit de meilleurs résultats de reconnaissance d'objets dans de nombreuses applications. D'une part, les motifs uniformes sont plus stables, c'est-à-dire présentent moins de bruit et d'autre part, les motifs uniformes réduisent de façon significative le nombre d'étiquettes de LBP possibles.

L'idée du motif binaire uniforme permet de considérer la méthode LBP comme une approche qui unifier l'approche statistique et les approches structurelles ou géométriques de l'analyse de la texture [Maenpaa and Pietikäinen, 2005]. Chaque pixel est représenté par le code de la texture primitive qui correspond au mieux à son voisinage local. Ainsi, chaque code de LBP peut être considéré comme un texton [12]. Les structures primitives locales détectées par la LBP comprennent des points, des arêtes, des extrémités, des courbes, etc. (voir la Figure 3.19). Grâce à cette combinaison d'approches statistiques et structurelles, la méthode LBP a connu un certain succès dans le domaine de la reconnaissance de textures variées.

3.3.1.3 LBP Centré-Symétrique

La méthode LBP Centré-Symétrique (CS-LBP : Center-Symmetric Local Binary Patterns) est une variante de LBP développée pour la description de la région d'intérêt de la texture. Heikkilä a proposé cette méthode la première fois en 2009[Heikkilä et al., 2009]. Ces auteurs ont combiné l'algorithme SIFT et CS-LBP qui extrait les caractéristiques locales. Le descripteur de cette combinaison est utilisé dans les expérimentations de clas-

12. Le terme "texton" est utilisé souvent dans l'analyse de texture. Le texon représente une forme élémentaire qui caractérise la texture.

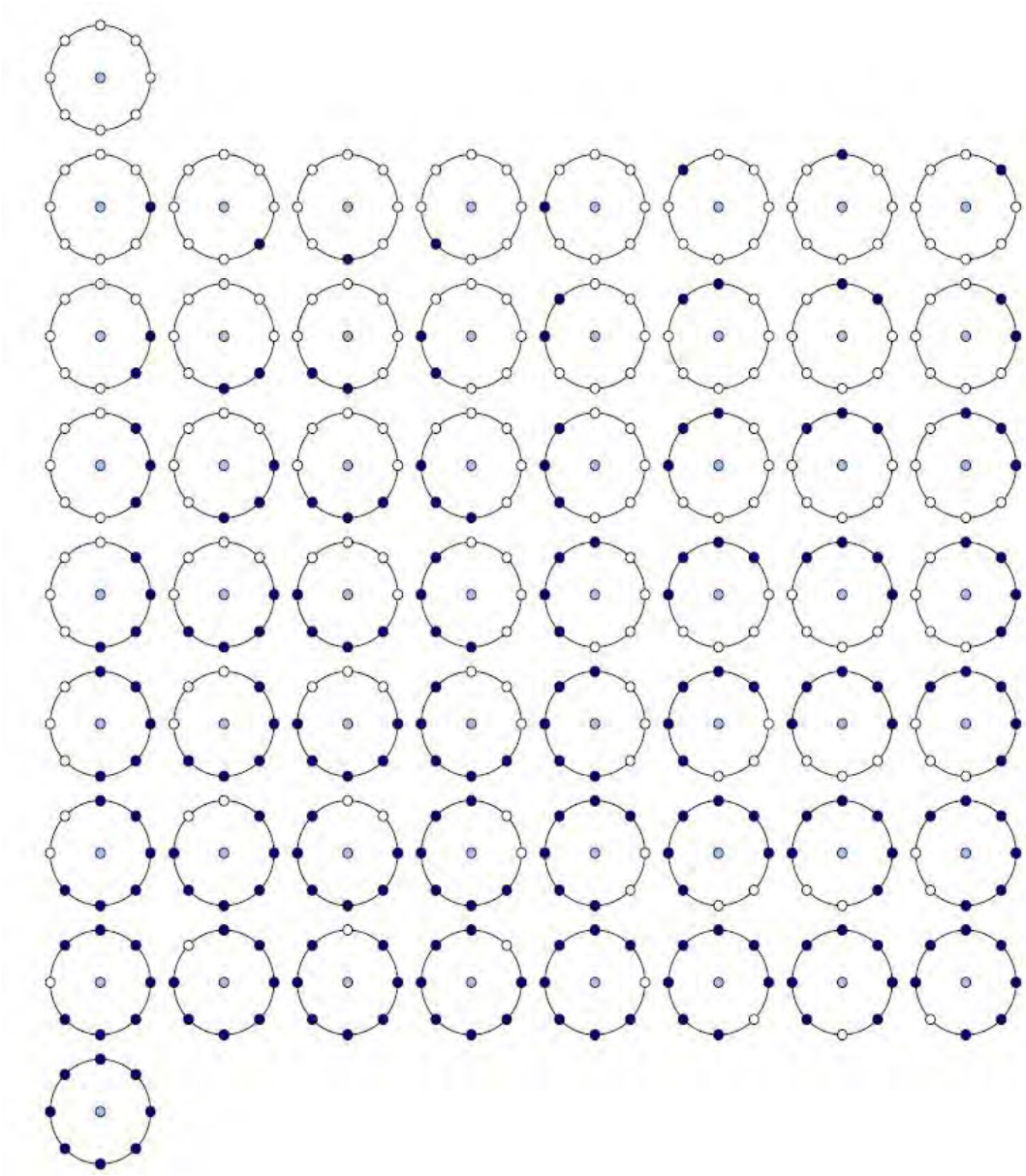


FIGURE 3.18 – Les 58 motifs uniformes avec le voisinage $(8, R)$. Les points de la couleur bleue foncée représentent "1" et les points blancs représentent "0".

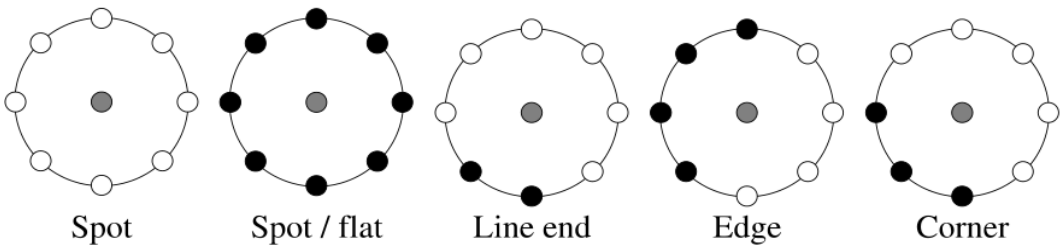


FIGURE 3.19 – Différentes structures primitives locales détectées par LBP.

sification des objets et la recherche (ou "matching") des objets. Les résultats montrent que la performance de cette méthode est meilleure que celle de SIFT. De plus CS-LBP est plus simple à calculer que SIFT.

Le principe de CS-LBP est de comparer, paire par paire, des pixels dans des positions symétriques. La Figure 3.20 montre un voisinage (8,R) où on compare les quatre paires de pixels (n_0, n_4), (n_1, n_5), (n_2, n_6), et(n_3, n_7). La valeur CS-LBP d'un pixel dans la

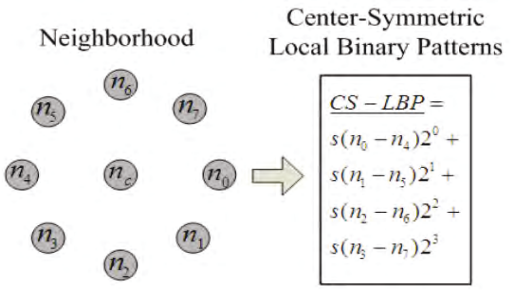


FIGURE 3.20 – Calcul de la valeur CS-LBP sur le point n_c avec un voisinage $(8, R)$.

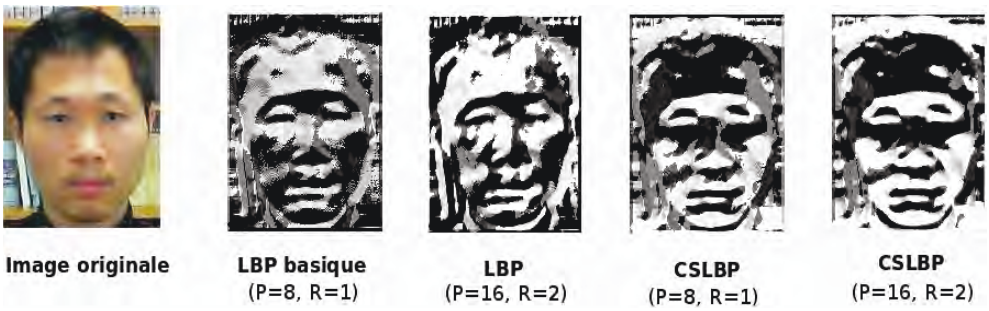


FIGURE 3.21 – Exemples des images LBPs avec un voisinage (P, R) différent.

position (x, y) est calculée comme la formule suivante :

$$CS - LBP_{R,P,T}(x, y) = \sum_{i=0}^{P/2-1} s(n_i - n_{i+(P/2)})$$

où $s(t) = \begin{cases} 1 & t > T \\ 0 & sinon \end{cases}$, n_i et $n_{i+(P/2)}$ sont les intensités des pixels en niveau de gris et T est le seuil défini. Par cette équation, la valeur CS-LBP est toujours un nombre entier entre 0 et $2^{P/2} - 1$. La dimension de l'histogramme est bien $2^{P/2}$. La Figure 3.21 montre les résultats obtenus en appliquant LBP et CS-LBP sur une image de visage. L'image CS-LBP avec un voisinage plus large, par exemple un voisinage (16, 2), permet de mieux réduire les redondances et éliminer le bruit. CS-LBP est souvent utilisé dans le domaine du suivi d'objets car il résiste au changement d'illumination. Par ailleurs, la dimension du vecteur reste relativement petite et efficace et représente un avantage non-négligeable pour les applications [Heikkilä et al., 2009].

3.3.2 Caractéristiques robustes accélérées

Les méthodes de caractérisation et d'interprétation du contenu des images reposent souvent sur l'extraction et la caractérisation de points d'intérêt. L'intérêt de ces approches réside notamment dans leurs propriétés d'invariance au changement de contraste et aux transformations affines des images. La plupart des détecteurs des points d'intérêt sont invariants en translation, comme par exemple le détecteur de Harris[Harris and Stephens, 1988]. Les évolutions telles celles de Harris-Laplace ou les méthodes basées sur les DoG (Difference-of-Gaussian) sont quant à elles invariantes en rotation et en changement d'échelle. La technique telle que MSER[Matas et al., 2004] a été mise au point dans l'objectif d'être invariante aux transformations affines. Cependant, SIFT (Scale-Invariant Feature Transform)[Lowe, 2004] reste la référence en matière de détection de points d'intérêt. Il combine les DoG qui sont invariants en translation, rotation et mise à l'échelle avec un descripteur basé sur les distributions d'orientations de gradient qui, de plus, est robuste aux changements d'illumination et de point de vue. Depuis, quelques variantes et extensions de SIFT telles que PCA-SIFT, ASIFT et SURF ont été mises au point.

L'algorithme des SURF (Speeded-Up Robust Features)[Bay et al., 2006], que l'on peut traduire par descripteurs des caractéristiques robustes accélérées, constitue une bonne alternative aux SIFT. Cette méthode s'appuie largement sur les SIFT mais est plus rapide et se révèle plus robuste quant à certaines transformations.

Le descripteur SURF est principalement reconnu pour sa rapidité de calcul. L'étude comparative[Juan et al., 2010] démontre la supériorité du descripteur SURF par rapport à SIFT et PCA-SIFT d'un point de vue de ces performances en temps d'exécution et de sa robustesse aux changements d'illumination. L'algorithme SURF est composé de trois étapes principales. La première consiste à détecter des points d'intérêt sur l'image, la seconde consiste à décrire ces points d'intérêt à l'aide d'un vecteur de 64 caractéristiques et la troisième est la mise en correspondance des points détectés sur l'autre image.

3.3.2.1 Détection des points d'intérêt

L'approche proposée par SURF utilise une approximation de la matrice hessienne afin de détecter les points d'intérêt qui vont se révéler être des sortes de barycentres des régions recherchées (blobs en anglais). Ces points d'intérêt ont un entourage facilement reconnaissable. La méthode utilise les images intégrales afin de diminuer fortement les temps de calcul car ces dernières permettent le calcul rapide des convolutions par des approximations utilisant les ondelettes de Haar.

1) Image intégrale

Afin de gagner du temps de calcul, l'image à analyser est transformée en image intégrale[Viola and Jones, 2004]. Les images intégrales permettent de faire beaucoup plus rapidement les calculs de convolution et d'aires rectangulaires. Soit i, notre image de départ, $i(x, y)$ représente la valeur d'un pixel de l'image aux coordonnées x et y. L'image intégrale, notée $ii(x, y)$, est une image de même taille que l'image d'origine, calculée à partir de celle-ci. Chaque pixel de l'image intégrale contient la somme des pixels situés au dessus et à gauche de ce pixel dans l'image initiale. La valeur d'un pixel

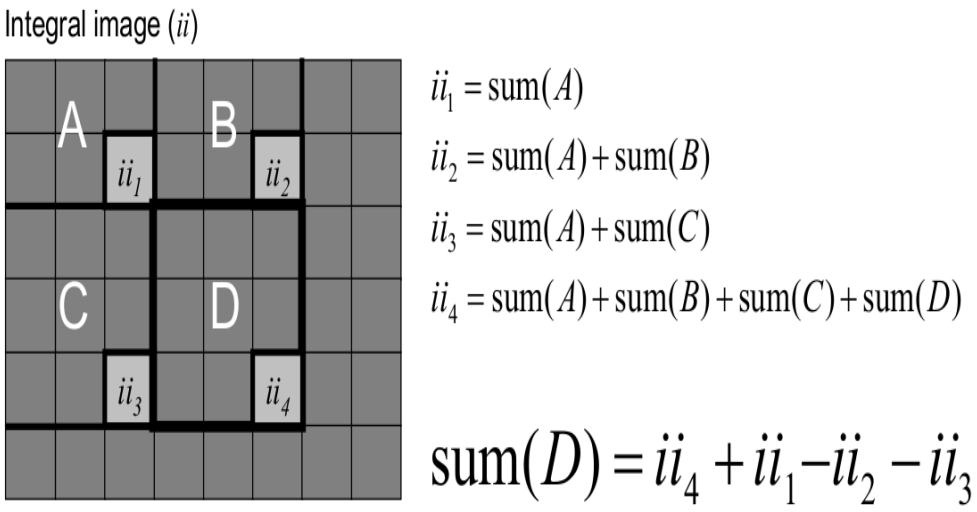


FIGURE 3.22 – Démonstration du calcul de la somme des pixels dans la région D par l'image intégrale.

de l'image intégrale ii est définie à partir de l'image i par l'équation suivante :

$$ii(x,y) = \sum_{i=0}^{i\leqslant x} \sum_{j=0}^{j\leqslant y} i(i,j)$$

Une fois l'image intégrale calculée, l'aire d'un rectangle $ABCD$ de l'image d'origine peut être évaluée en accédant seulement à la valeur des quatre sommets alors qu'il faudrait accéder à toutes les valeurs des pixels du rectangle sans image intégrale (voir la Figure 3.22) :

$$Aire_D = \sum_{i\geqslant x_C}^{i\leqslant x_D} \sum_{j\geqslant y_C}^{j\leqslant y_D} i(i,j) = ii(C) + ii(D) - ii(B) - ii(A)$$

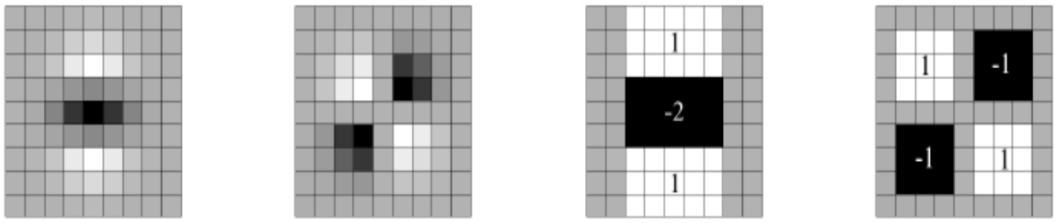
2)Détecteur basé sur la matrice hessienne

Le détecteur localise les points là où le déterminant de la matrice hessienne atteint son maximum. Pour rappel, la matrice hessienne (ou simplement la hessienne) d'une fonction numérique f est la matrice carrée, notée $H(f)$, de ses dérivées partielles secondes. Pour une fonction à deux variables $f(x,y)$, par exemple dans le contexte du détecteur de point $X = (x,y)$ dans l'image I, la matrice hessienne en ce point et à l'échelle σ est définie comme suit :

$$H(X,\sigma) = \begin{bmatrix} L_{xx}(X,\sigma), & L_{xy}(X,\sigma) \\ \\ L_{xy}(X,\sigma), & L_{yy}(X,\sigma) \end{bmatrix}$$

avec $L_{xx}(X,\sigma)$ qui est le résultat de la convolution de la dérivée seconde de la gaussienne $\frac{\partial^2}{\partial_x^2} g(\sigma)$ avec l'image au point X.

En pratique, la gaussienne doit être finie et discrétisée. Pour pouvoir tirer parti des images intégrales, [Bay et al., 2006] construisent une approximation de type «*box filter*» des dérivées secondes de la gaussienne. Ce «*box filter*» est en fait un filtre en utilisant les



a) b) c) d)

FIGURE 3.23 – a)b) : $L_{yy}(x,\sigma)$ et $L_{xy}(x,\sigma)$. c)d) : La gaussienne discretisée et l'approximation de D_{yy} et D_{xy} par les ondelettes de Haar.

ondelettes de Haar. Grâce aux images intégrales, le temps de calcul est indépendant de la taille du filtre. Dans la Figure 3.23 on peut voir les dérivées partielles de la gaussienne. D'abord finies et discrétisées (Figure 3.23 a et b) et puis approximées par un *box filter* suivant les directions y et xy (Figure 3.23 c et d). Les zones grises sont égales à zéro. L'approximation est comme suit :

$$Det(H_{approx}) = D_{xx}D_{yy} - (\omega D_{xy})^2$$

où D_{xx} et D_{yy} représentent respectivement $L_{xx}(X,\sigma)$ et $L_{yy}(X,\sigma)$, D_{xy} représente $L_{xy}(X,\sigma)$, ω est un poids dont la valeur en pratique est 0.9. Le filtre approximé initial de taille 9×9 dont l'échelle $s = 1.2$ correspond à un filtre gaussien d'écart type $\sigma = 1.2$.

Si le déterminant de la matrice Hessienne est positif, alors les valeurs propres de la matrice sont toutes les deux positives ou toutes les deux négatives, ce qui signifie qu'un extremum est présent. Les points d'intérêt seront donc localisés là où le déterminant de la matrice Hessienne est maximal.

Il est intéressant de pouvoir retrouver des points d'intérêt à différentes échelles afin de rendre le détecteur invariant aux changements d'échelle (le même objet peut être représenté en tailles différentes sur deux images). Cet aspect est souvent pris en compte en créant une pyramide d'images. Les images sont répétitivement filtrées avec une gaussienne puis sous-échantillonnées afin d'obtenir une image de plus petite taille. SURF peut procéder différemment grâce aux box filters et aux images intégrales. Au lieu d'appliquer successivement le même filtre à la sortie d'une image filtrée et sous-échantillonnée, on peut utiliser des box filters de diverses tailles directement sur l'image d'origine. Les réponses à une ondelette de Haar à différentes échelles sont donc calculées en agrandissant le filtre plutôt qu'en réduisant itérativement la taille de l'image. Ceci permet d'une part de réduire le temps de calcul et d'autre part d'éviter l'aliasing dû au sous-échantillonnage de l'image. Chaque niveau de la pyramide représente une échelle différente comme $(9 \times 9, 15 \times 15, 21 \times 21, 27 \times 27, ...)$. Par exemple dans le cas d'un filtre de taille 27×27, l'approximation correspond à une gaussienne possédant un $\sigma = \frac{27}{9} \times 1.2 = 3.6 = s$.

3.3.2.2 Description des points d'intérêt

Une fois les points d'intérêt extraits, la seconde étape de SURF consiste à calculer le descripteur correspondant. Le descripteur SURF décrit l'intensité des pixels dans

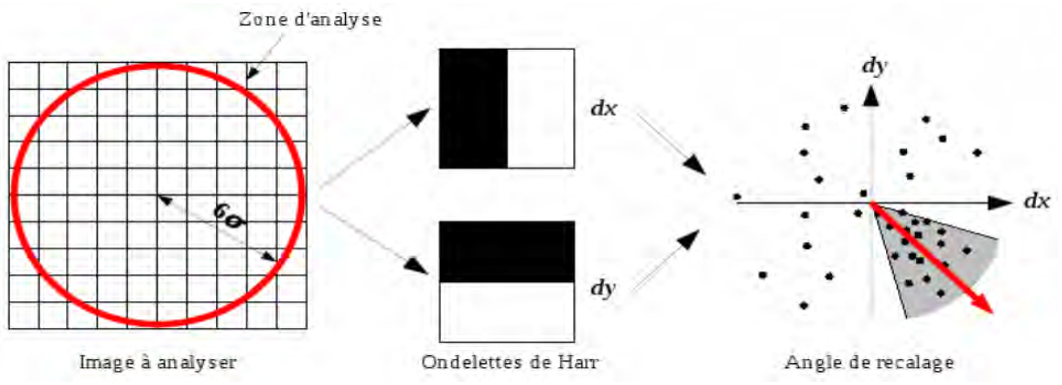


FIGURE 3.24 – Détermination de l'angle de recalage du SURF, en analysant la répartition des réponses des ondelettes de Haar.

un voisinage autour de chaque point d'intérêt. Fortement influencée par l'approche de SIFT, SURF couple une étape de recalage de la zone d'analyse avec la construction d'un histogramme de gradients orientés. La première étape est donc de déterminer l'angle de rotation (ou de recalage) à appliquer à la fenêtre de description locale. Pour cela, on calcule la réponse à des ondelettes de Haar suivant les directions x et y dans le voisinage du point d'intérêt. On applique des ondelettes de Haar sur l'image intégrale permettant ainsi de diminuer les temps de calculs de façon significative. Une fois les réponses calculées, elles sont pondérées par une fenêtre gaussienne centrée sur le point d'intérêt et représentées comme un point dans un espace dont l'abscisse représente la valeur de la réponse horizontale (axe x) et l'ordonnée représente la valeur de la réponse verticale (axe y). Une région qui répondrait mieux aux ondelettes orientées suivant la direction verticale verrait donc la majorité des réponses concentrées le long de l'axe y. Ensuite on calcule la somme de toutes les réponses situées dans une fenêtre de taille $\pi/3$ tournant autour du centre de la région d'intérêt, ce qui permet de définir la norme du vecteur local d'orientation. La direction du plus long vecteur définit l'orientation principale de la région d'intérêt. La Figure 3.24 schématise cette étape : sur l'image initiale le cercle représente la région d'intérêt dont le rayon est égal à $6s$ où s correspond à l'échelle caractéristique extraite du détecteur hessien sur le point d'intérêt détecté.

Les ondelettes de Haar sont constituées d'une partie noire ayant la valeur -1 et d'une partie blanche ayant la valeur $+1$ et leur taille est égale à $4s$. La détermination de l'angle de recalage illustrée dans la Figure 3.24 (droite) se fonde sur la recherche de la répartition majoritaire des réponses des ondelettes dans une zone de rayon $\pi/3$ (zone grise sur le schéma).

Pour calculer le descripteur, on construit une région rectangulaire centrée autour du point d'intérêt et orientée suivant la direction principale sélectionnée au point précédent. La taille de cette région ($20s$) est déterminée par l'échelle à laquelle le point d'intérêt a été trouvé. Cette région est subdivisée en 4×4 carrés. Dans chacune de ces sous-régions, les réponses à une ondelette de Haar sont calculées sur des échantillons régulièrement espacés (voir la Figure 3.25). La réponse suivant la direction horizontale de la sous-région sélectionnée est notée d_x, et d_y désigne la réponse suivant la direction verticale. Notons que les directions verticale et horizontale sont définies par rapport à l'orientation de la zone d'intérêt. Pour augmenter la robustesse par rapport à une erreur de localisation du point d'intérêt, les réponses d_x et d_y sont pondérées par une fenêtre gaussienne.

Ensuite, les réponses pondérées sont sommées sur chaque sous-région et forment les deux premières entrées du vecteur de caractéristiques ($\sum d_x$, $\sum d_y$). Pour apporter une information supplémentaire concernant les changements d'intensité, les sommes des valeurs absolues des réponses sont aussi extraites ($\sum |d_x|$, $\sum |d_y|$) et constituent les deux entrées suivantes du vecteur de caractéristiques de quatre éléments ($\sum d_x$, $\sum d_y$, $\sum |d_x|$, $\sum |d_y|$) (voir la Figure 3.26). Comme il y a 16 sous-régions, on retrouve un vecteur de 64 dimensions, qui constitue la signature de la région d'intérêt.

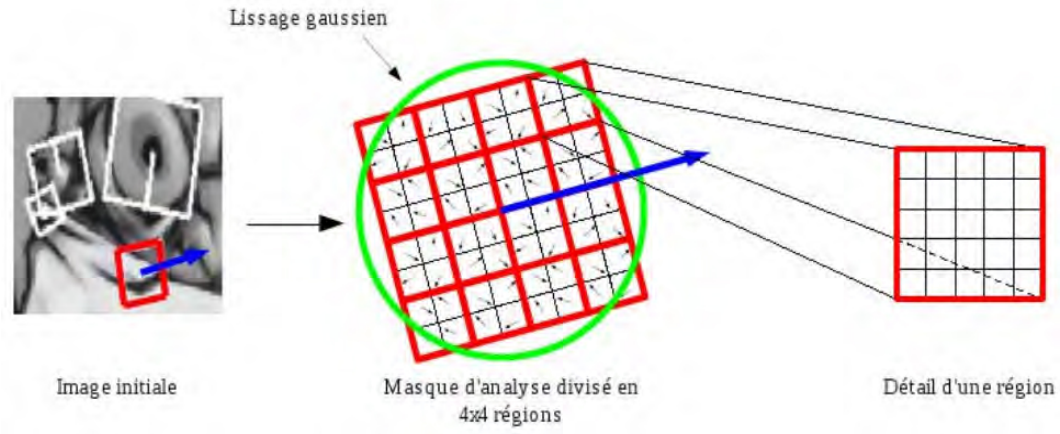


FIGURE 3.25 – Zone d'analyse du SURF divisée en 4×4 régions, elles même divisées en 5×5 sous-régions.

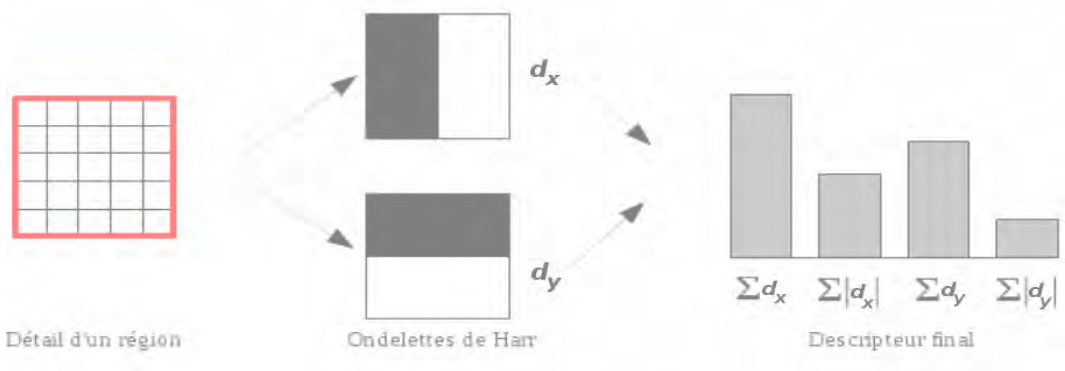


FIGURE 3.26 – Extraction des différents composants du vecteur de caractéristiques

3.3.2.3 Mise en correspondance des points

Dans cette étape, les points d'intérêt d'une image sont mis en correspondance avec les points d'intérêt d'une autre image afin d'estimer le degré de similitude entre ces deux images. Chaque point d'intérêt de l'image de référence est associé aux deux points d'intérêt de l'image requête les plus proches. Le plus proche voisin serait suffisant, mais le second plus proche sera utilisé pour l'étape de filtrage qui suit. Pour trouver les 2-ppv (deux plus proches poisins) la distance euclidienne entre les descripteurs à 64 dimensions est utilisée.

La recherche de plus proches voisins peut s'avérer longue si elle est faite de manière exhaustive. Les arbres k-d permettent de structurer l'espace de recherche afin d'accélérer la comparaison d'un élément avec les autres. La Figure 3.27 illustre la construction et la recherche de plus proches voisins avec un arbre k-d. La performance de recherche d'un arbre k-d se dégrade rapidement lorsque le nombre de dimensions est grand. Pour le descripteur SURF qui a 64 dimensions, Silpa-Anan et al.[Silpa-Anan and Hartley, 2008] ont proposé une version améliorée de l'arbre k-d dont le principe est d'appliquer en parallèle n arbres k-d aléatoires sur 5 premières dimensions dont les variances des données sont les plus grandes. Cet algorithme est implémenté dans la bibliothèque FLANN (Fast Library for Approximate Nearest Neighbors) décrite dans l'article de [Muja and Lowe, 2009].

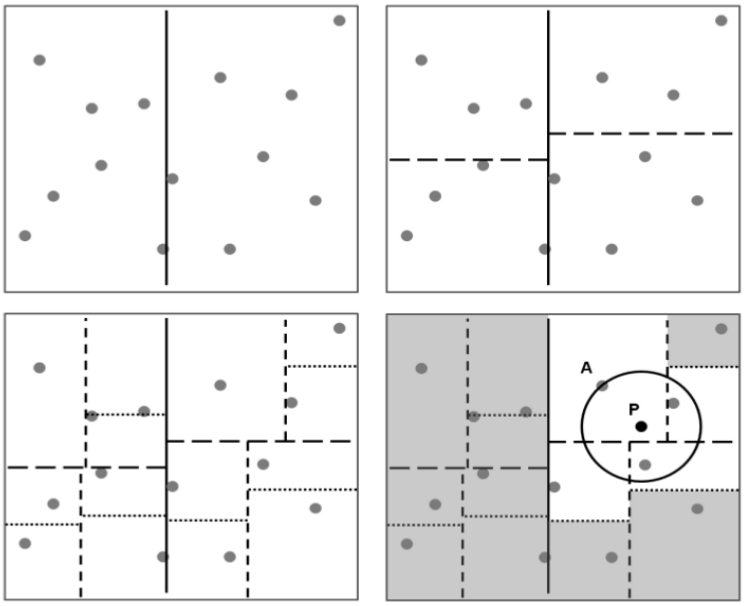


Figure 3.27 – L'espace est récursivement découpé en hyperplans, à hauteur de la médiane selon une dimension (x) puis selon la dimension suivante (y), etc. L'arbre k-d est l'arbre binaire correspondant au découpage. La première partie de la recherche consiste à déterminer dans quelle case de l'arbre est le point P dont le ppv est cherché. Le point A présent dans cette case est marqué comme étant l'actuel ppv. Toutes les cases qui sont plus loin que l'hypersphère de centre P et de rayon PA sont éliminées. Il ne reste plus qu'à trouver si un point dans les cases restantes est plus près de P que A.

Après la recherche de plus proches voisins, chacun des points d'intérêt de référence est associé aux deux points les plus ressemblants dans l'image requête. On souhaite éliminer le plus de fausses mises en correspondance possibles afin de faciliter l'estimation de la transformation. Le premier filtre consiste à supprimer les mises en correspondance dont les 2-ppv sont trop proches l'un de l'autre. C'est le filtrage par unicité, il permet d'éliminer les mises en correspondance ambiguës.

Étant donné deux images A et B, leurs points remarquables p_A et p_B sont détectés et les respectives descripteurs des_A et des_B sont calculés. À partir de la distance entre les descripteurs, sont trouvés les deux plus proches voisins pour chaque élément de p_A en p_B et pour chaque élément de p_B en p_A, en générant deux tableaux de mise en correspondance : $matches_{AB}$ et $matches_{BA}$. Pour chaque point remarquable de A, il y a deux candidats correspondants en B et vice-versa. Le choix de deux candidats par point

est utile pour mesurer la fiabilité de la correspondance. Si la distance D_1 mesurée pour le meilleur candidat (le plus proche) est très faible et la distance D_2 du second candidat est beaucoup plus grande, le premier candidat peut être choisi sûrement comme la bonne correspondance, puisqu'il est indiscutablement le meilleur choix. Dans un cas contraire, où les deux candidats ont des distances très proches l'une de l'autre, il n'est pas possible de choisir le bon candidat avec beaucoup de certitude et dans ce cas la correspondance doit être rejetée. Un seuil est donc choisi de façon arbitraire et expérimentale pour filtrer les fausses correspondances : le taux entre les distances des deux candidats doit être plus petite que le seuil ($\frac{D_1}{D_2} < \text{Seuil}$).

Après ce premier filtrage qui permet d'éliminer un grand nombre d'ambigüités, les deux tableaux de correspondances, $matches_{AB}$ et $matches_{BA}$, sont relativement fiables et un filtrage de symétrie peut être mis en place. Ce deuxième filtrage sélectionne les correspondances qui sont en accord avec les deux tableaux. Ce filtrage de symétrie impose que, pour qu'une correspondance soit acceptée, les deux points remarquables de p_A et de p_B doivent être le meilleur candidat l'un pour l'autre. Ensuite les mises en correspondance peuvent être aussi filtrées en fonction de l'échelle et l'orientation. Les détails de ces techniques de filtrage sont présentés dans [Laganiere, 2011].

3.3.3 Expérimentation

Dans cette section, nous allons présenter les résultats d'application de ces deux méthodes LBP et SURF sur l'image de l'œil pour extraire respectivement les caractéristiques globale et locale de l'image. L'objectif de l'expérimentation est d'extraire les caractéristiques pour former un descripteur de l'œil.

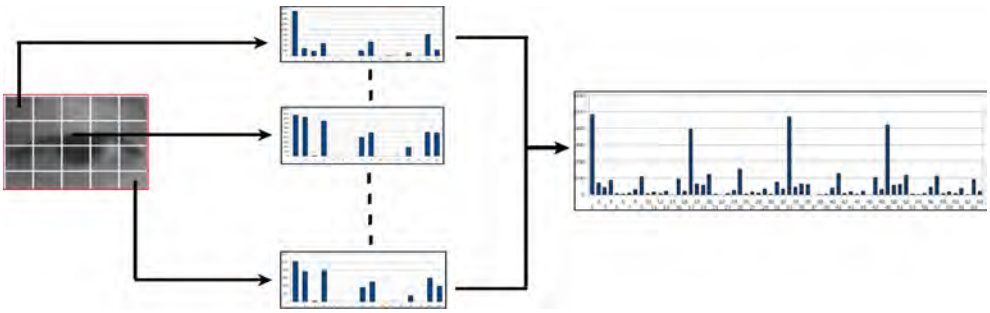


FIGURE 3.28 – Représentation d'un image de l'œil par la combinaison des histogrammes CS-LBP sur chaque bloc de l'image (par exemple 20 blocs).

La dimension de l'image de l'œil est 60×40 pixels. L'histogramme d'une image en tant que descripteur ne montre que les nombres d'apparitions des codes LBP dans la région d'intérêt, mais ne révèle pas la position des caractéristiques. Nous proposons dans un premier temps de diviser l'image de l'œil en blocs, par exemple en 20 blocs (5×4). Ensuite, nous calculons la valeur LBP sur chaque bloc de l'image. Enfin, nous combinons les histogrammes des blocs d'image en un descripteur représentant cette image (voir la Figure 3.28). Pour l'expérimentation, nous avons choisi la méthode CS-LBP à cause de sa robustesse au changement d'illumination et sa faible dimensionnalité. Pour une image de l'œil en 20 blocs, la dimension du descripteur calculé par CS-LBP est $16 * 20 = 320$, qui est largement inférieur à la dimension de l'image originale. La figure 3.29 démontre

la capacité du descripteur CS-LBP à résister au changement d'illumination. Les images capturées dans la région de l'œil représentent le même comportement oculaire du sujet, et nous pouvons observer la différence des apparences lorsque la lumière change (voir Fig 3.29(A)). Les colonnes (C) et (D) dans la figure montrent respectivement la distribution des histogrammes sur toute l'image CS-LBP et des histogrames combinées sur les blocs de l'image CS-LBP. Ces histogrammes préservent les caractéristiques de la texture de l'image. La figure 3.30 illustre la discrimination de l'œil d'un sujet des autres parties de l'image. Dans cette figure nous ne montrons que l'histogramme CS-LBP sur toute l'image sans diviser en blocs, donc l'histogramme a 16 dimensions par rapport à la valeur CS-LBP. La réalisation de la méthode CS-LBP est présentée dans l'Annexe 6.

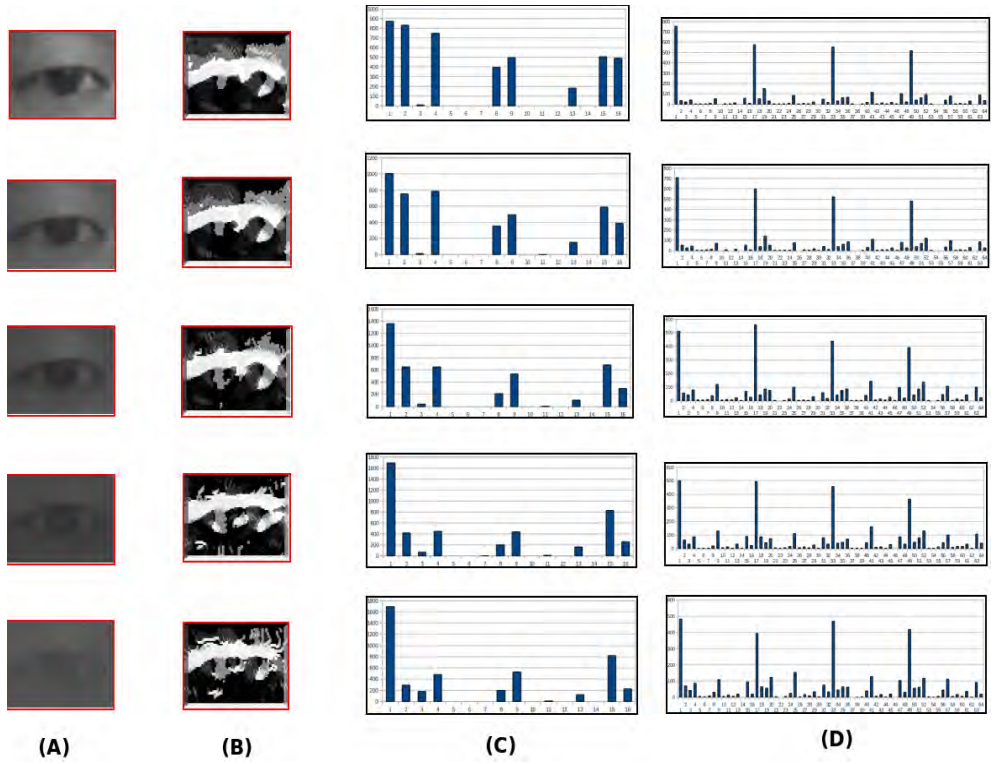


Figure 3.29 – Démonstration de la robustesse de CS-LBP au changement d'illumination. (A) : les images originales des yeux sous différentes luminosités. (B) : les images CSLBP où $r = 1$, $n = 8$. (C) : les histogrames correspondant (16 dimensions). (D) les histogrammes correspondants aux images 2×2 (64 dimensions).

Nous utilisons SURF pour détecter les points d'intérêt de la région de l'œil de référence au départ ainsi que pour mettre en correspondance les points dans les images qui suivent. Afin de détecter assez de points différents, la région de l'œil est élargie en taille de 100×80 en pixels. La figure 3.31 montre les résultats de la correspondance des points par SURF.

La combinaison de ces deux méthodes peut être utilisée pour identifier la présence de l'œil dans une image. Nous pouvons calculer la distance entre les histogrammes CS-LBP de cette image et l'image de référence ainsi que le nombre de points d'intérêt de référence suivis dans la nouvelle image pour déterminer si cette nouvelle image contient l'œil.

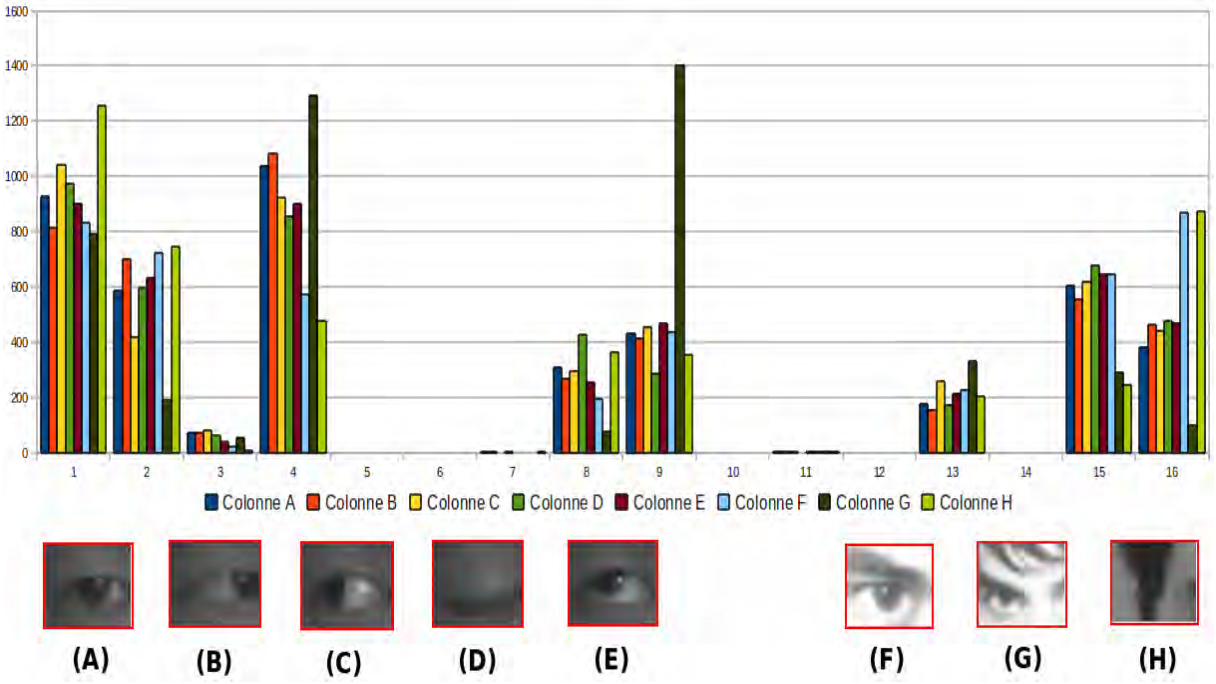


FIGURE 3.30 – Différence des histogrammes CS-LBP des images. Les couleurs différentes dans l'histogramme représentent respectivement les images différentes : (A,B,C,D,E) sont les images des mouvements oculaires du même sujet, et (E,F,G) sont des images quelconques.

(a) (b) (c) (d) (e)

FIGURE 3.31 – Mise en correspondance des points détectés par SURF. (a) les 9 points détectés (numérotés) sur l'image de l'œil de référence. Ensuite nous effectuons la mise en correspondance des points sur les autres images : (b) la fermeture de l'œil ; (c, d) l'œil en rotation ; (e) l'œil a disparu.

3.4 Suivi des yeux

Après avoir détecté la région de l'œil, la méthode stochastique du filtre particulaire est utilisée pour suivre l'œil dans les images suivantes. Cette méthode probabiliste estime la position de l'œil dans l'image actuelle par rapport à la position de l'œil obtenue dans l'image précédente. Le filtre particulaire génère un ensemble de "particules". Chaque particule représente un état probable de l'œil et chacune évolue d'une manière indépendante dans chaque image. Dans cette section, nous allons présenter la technique du filtrage particulaire et l'implémentation de cette méthode pour le suivi des yeux.

3.4.1 Filtrage particulaire

Considérons un système dynamique c'est-à-dire un système d'équation différentielles qui régit l'évolution(détermiste ou stochastique) d'un état inconnu au cours du temps. L'état peut représenter, par exemple, les paramètres cinématiques d'un mobile quelconque (position, vitesse, accélération) ou peut représenter le prix d'une marchandise. L'observateur dispose de mesures partielles entachées d'erreurs(mesures indirectes d'une partie du vecteur d'état) comme des mesures de distance ou d'angle. Le **filtrage** consiste à restituer l'état à partir de ces mesures. Notons qu'il peut s'appliquer dans de nombreux domaines comme le pistage (estimation d'une trajectoire d'un véhicule ou d'un piéton [Dore et al., 2008]), le suivi de contours d'objets dans l'image[Blake et al., 1998], etc. La dimension du vecteur d'état peut être faible (2 en position d'objet dans le vidéo) ou très grande (plusieurs millions en océanographie). Nous sommes conduits à construire un filtre qui fournit une estimation de l'état à chaque instant à partir des mesures récoltées. Ce filtre doit pouvoir être implémenté dans un ordinateur.

En 1961, le filtre de Kalman[Kalman, 1960] a été développé pour les modèles linéaire gaussiens. Ce filtre approche la densité de l'état connaissant l'observation (densité conditionnelle) par une densité gaussienne déterminée par sa moyenne et sa matrice de covariance. La non-linéarité du modèle peut entrainer la multi-modalité de la loi conditionnelle de l'état, et ainsi rendre le filtre de Kalman inadapté. Au début des années 90, les méthodes Monte-Carlo (Annexe 5) ont été proposées pour résoudre le filtrage non linéaire. Ces méthodes, basées sur la loi des grands nombres, ont des performances peu sensibles à la dimension de l'espace d'état. Ainsi, même en grande dimension, les méthodes Monte-Carlo peuvent estimer la densité conditionnelle en temps réel.

Le **Filtrage Particulaire** (Particle Filtering) est une méthode de simulation séquentielle de type Monte-Carlo. Il a été introduit par Del Moral, Rigal et Salut[Moral et al., 1992] et par Gordon, Salmond et Smith[Gordon et al., 1993]. Il propose de représenter la loi conditionnelle de l'état par une somme pondérée de mesures de Dirac. Un ensemble de points appelés "particules" est généré, chacune de ces particules représente un état probable du système. Les coefficients de pondération (poids) sur chaque particule sont une mesure du degré de confiance que l'on peut avoir en ces dernières pour représenter effectivement l'état. Les particules évoluent suivant l'équation d'état du système (état de prediction) et les poids sont ajustés en fonction des observations (étape de correction).

Le filtrage particulaire connait actuellement un fort développement dans de nom-

breux domaines relevant des sciences et techniques de l'information et de la communication, ou des sciences de l'ingénieur :

- localisation, navigation, poursuite d'un ou plusieurs cibles,
- vision par ordinateur,
- robotique,
- traitement du signal audio,
- communications numériques.

De nombreuse versions de filtres particulaires ont été proposées dans la littérature depuis une dizaine d'années. La première version est proposée par Gordon et al. dans [Gordon et al., 1993] sous le nom de « Bootstrap Filter ». La méthode est redécouverte indépendamment dans un contexte de Vision par Ordinateur par [Isard et al., 1998], qui introduisent le vocable CONDENSATION pour « CONditional DENSity propagATION ». L'algorithme générique SIR (Sampling Importance Resampling) englobe ces travaux pionniers ainsi que d'autres variantes. Dans cette section nous présentons la théorie de base sur cette technique d'approximation particulaire. Pour les aspects mathématiques et la lecture approfondie, le lecteur est invité à consulter le numéro spécial[Djuric and Godsill, 2002], le tutoriel de [Arulampalam et al., 2002] et l'article de [Doucet and Johansen, 2008] pour une présentation plus détaillée.

3.4.1.1 Modèle de Markov caché

Le modèle de Markov caché, en anglais « Hidden Markov Model » (HMM), est un cas particulier de systèmes dynamiques stochastiques partiellement observés, où l'état d'un processus de Markov (à temps discret ou continu et à espace d'état fini ou général) doit être estimé à partir d'observations bruitées. Ces modèles sont très flexibles, du fait de l'introduction de variables latentes (non-observées) qui permettent de modéliser des structures de dépendances temporelles complexes, de prendre en compte des contraintes, etc. En outre, la structure markovienne sous-jacente permet d'utiliser des procédures numériques intensives (filtrage particulaire, méthodes de Monte Carlo par chaîne de Markov (mcmc), etc.) dont la complexité est très réduite. Les HMMs sont largement utilisés dans des domaines applicatifs variés, comme la reconnaissance de la parole, l'alignement de séquences biologiques, la poursuite en environnement complexe, la modélisation et le contrôle des réseaux, les communications numériques, etc.

Un modèle markovien d'ordre k considère que l'état du système à un instant t ne dépend que de l'état aux k instants précédents. Cela implique la propriété d'indépendance conditionnelle suivante :

$$\mathcal{P}(x_t|x_{1:t-1}) = \mathcal{P}(x_t|x_{t-k:t-1})$$

où x_t est la variable d'état à l'instant t ($t \in 1, ..., T$), et l'ensemble d'états $X = \{X_1, ..., X_n\}$. Ici les transitions entre les états se produisent entre k instants discrets consécutifs, selon une certaine loi de probabilité.

Un HMM représente, de la même façon qu'une chaîne de Markov, un ensemble de séquences d'observations dont l'état de chaque observation n'est pas visible, mais associé à une fonction de densité de probabilité (*pdf*). L'ensemble d'observations est noté $Z = \{Z_1, ..., Z_m\}$. Il s'agit donc d'un processus doublement stochastique, dans lequel les observations sont une fonction aléatoire de l'état et dont l'état change à chaque instant

en fonction des probabilités de transition issues de l'état antérieure. Généralement un modèle HMM peut être défini par $M = (A, B, \Pi)$ où :

- A **est les probabilités de transitions entre états.**
  Ici conditionnellement aux états passés $(x_0, ..., x_{t-1})$, l'état présent x_t ne dépend que de l'état précédent x_{t-1}.
  $A = [a_{ij}]$ où $a_{ij} \equiv P(x_{t+1} = X_j | x_t = X_i)$

- B **est les probabilités d'émissions.**
  L'hypothèse implicite dans les modèles de Markov cachés pour relier les observations aux états cachés, est de supposer que les observations sont indépendantes, sachant que les états sont cachés, et que la probabilité d'émission de z_t ne dépend que de x_t.
  $B = [b_j(m)]$ où $b_j(m) \equiv P(z_t = Z_m | x_t = X_j)$

- Π **représente les probabilités d'état initial.**
  $\Pi = [\pi_i]$ où $\pi_i \equiv P(x_1 = S_i)$

Ce modèle est completement décrit par le diagramme de dépendance ci-dessous (Figure 3.32) :

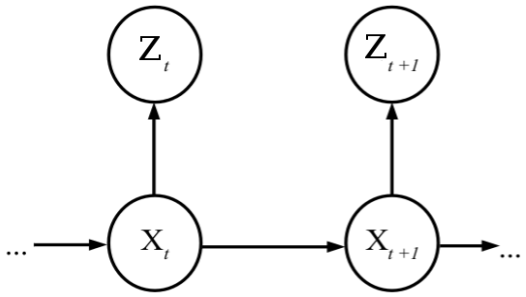


FIGURE 3.32 – Le diagramme de dépendance

Considérons un simple problème de "poursuite", où il s'agit d'estimer l'état x_t ($x_t \in X$)d'un système (par example la position d'un mobile), à partir d'observations $\{z_0, ..., z_t\}$, où $z \in Z$ (par exemple des observations de distance par rapport à une ou plusieurs stations qui contiennent un bruit blanc additif, pas nécessairement gaussien). Ce problème peut être défini comme :

$$\begin{cases} x_t = F_t(x_{t-1}) + W_t \\ z_t = H_t(x_t) + V_t \end{cases} \qquad (1)$$

où x_t est le vecteur d'état du système de dimension n, z_t est le vecteur d'observations de dimension m, F_t est la fonction de transition (ou évolution) de $\mathcal{R}^n$ vers $\mathcal{R}^n$, et H_t est la fonction d'émission (ou observation) de $\mathcal{R}^n$ vers $\mathcal{R}^m$. W_t et V_t sont des vecteurs de bruit (pas nécessairement gaussiens).

3.4.1.2 Approche bayesienne

Le modèle (1) rentre bien dans le cadre d'un système stochastique markovien caractérisé, à chaque instant $t \in T$, par un vecteur d'état x_t. L'évolution de ce système se

traduit par un ensemble d'observations $\{z_0, ..., z_t\}$. On suppose connue la distribution du vecteur d'état à l'instant initial $\mathcal{P}(x_0)$. Toute la connaissance *a priori* sur l'évolution temporelle du vecteur d'état x_t est modélisée par la dynamique du système $\mathcal{P}(x_t|x_{t-1})$. Le lien entre le vecteur d'observation z_t et le vecteur d'état x_t est régi par la densité de probabilité $\mathcal{P}(z_t|x_t)$. Le système est ainsi entièrement caractérisé par :

$$\begin{cases} \mathcal{P}(x_0), & \text{la distribution du vecteur d'état à l'instant initial} \\ \mathcal{P}(x_t|x_{t-1}), & \text{la dynamique } a \textit{ priori} \text{ du système} \\ \mathcal{P}(z_t|x_t), & \text{le lien état-observation} \end{cases} \tag{2}$$

L'objectif du filtrage (particulaire ou autre) est alors d'estimer la loi *a posteriori* de x_t conditionnellement aux mesures $z_{1:t}$, *i.e.* la densité de probabilité $\mathcal{P}(x_t|z_{1:t})$. Dans l'approche bayesienne, si la distribution $\mathcal{P}(x_0)$ est supposée connue, deux étapes sont appliquées récursivement pour estimer la distribution *a posteriori* de $\mathcal{P}(x_t|z_{1:t})$ à chaque instant t :

- **Etape de prédiction**
  La nouvelle observation n'est pas encore arrivée et le modèle d'état permet de calculer la distribution prédite selon l'équation de Chapman-Kolmogorov :

$$\mathcal{P}(x_t|z_{1:t-1}) = \int_{x_{t-1}} \mathcal{P}(x_t|x_{t-1})\mathcal{P}(x_{t-1}|z_{1:t-1})dx_{t-1} \tag{3}$$

  où $\mathcal{P}(x_t|x_{t-1})$ est la fonction de transition définie dans le (1).
- **Etape de correction**
  Comformément à la loi de Bayes, l'estimation de la distribution d'intérêt est mise à jour avec l'arrivée de la nouvelle observation :

$$\mathcal{P}(x_t|z_{1:t}) = \frac{\mathcal{P}(z_t|x_t)\mathcal{P}(x_t|z_{1:t-1})}{\mathcal{P}(z_t|z_{1:t-1})} \tag{4}$$

  où la constante de normalisation $\mathcal{P}(z_t|z_{1:t-1})$ est calculée de la façon suivante :

$$\mathcal{P}(z_t|z_{1:t-1}) = \int_{x_t} \mathcal{P}(z_t|x_t)\mathcal{P}(x_t|z_{1:t-1})dx_t \tag{5}$$

  et dépend de la fonction de vraisemblance $\mathcal{P}(z_t|x_t)$ définie dans le (1). Donc l'équation (4) peut être réécrite comme :

$$\mathcal{P}(x_t|z_{1:t}) \propto \mathcal{P}(z_t|x_t) \int_{x_{t-1}} \mathcal{P}(x_t|x_{t-1})\mathcal{P}(x_{t-1}|z_{1:t-1})dx_{t-1} \tag{6}$$

3.4.1.3 Approximation particulaire

Evidemment la solution récursive décrite dans la section précédente ne peut généralement pas être calculée analytiquement. En effet, les équations (3) et (5) font intervenir des intégrales multiples qu'il est difficile d'évaluer. Dans le cas des méthodes particulaires, la représentation de $\mathcal{P}(x_t|z_{1:t-1})$ est d'une nature tout à fait différente : nous ne cherchons pas une représentation analytique approchée d'une densité de probabilité mais à construire un ensemble d'échantillons (appellés ici "particules").

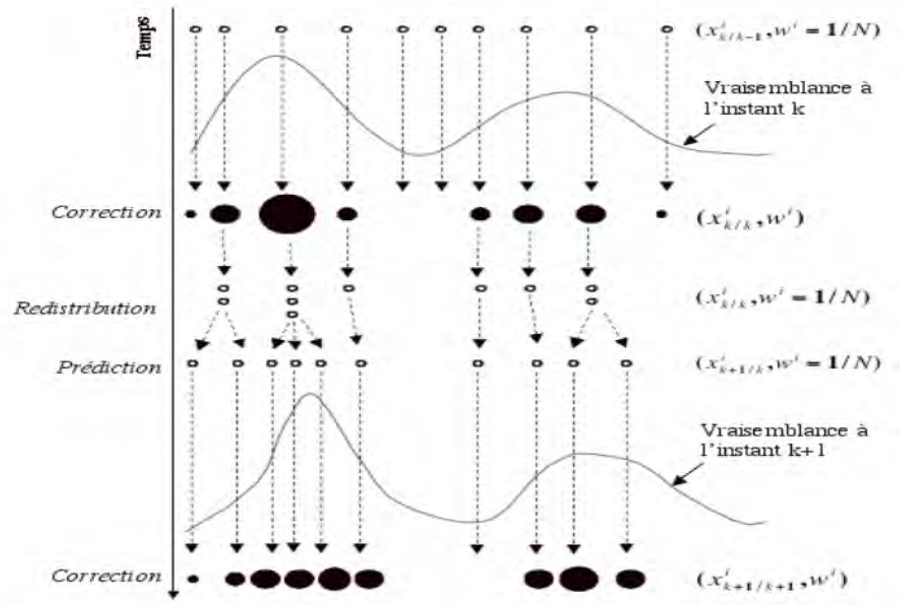


FIGURE 3.33 – Les étapes de l'algorithme du filtrage particulaire.

Le filtrage particulaire consiste à propager un ensemble d'échantillons pondérés (particules) de telle façon qu'à chaque itération k, les particules soient représentatives de la loi $\mathcal{P}(x_t|z_{1:t-1})$. Ceci est effectué récursivement en modifiant l'état et le poids de chaque particule, selon des règles qui dépendent des équations d'état et de mesure du problème. Une particule est un échantillon issu d'une loi de probabilité, auquel on associe un poids. Pour un filtre particulaire constitué de N particules, on note que les particules $x_t^1, ..., x_t^N$ sont des vecteurs de $\mathcal{R}^n$, et les $\omega_t^1, ..., \omega_t^N$ sont des poids positifs de somme 1.

Les particules évoluent dans le temps selon la dynamique de l'état (étape de prédiction) et sont pondérés en fonction de leur vraisemblance en fonction de l'observation courante (étape de correction). Un des premiers algorithmes populaires du filtrage particulaire est apparu en 1993 sous le nom de "boostrap filter"[Gordon et al., 1993] et constitue en fait une application d'une technique Monte-Carlo appelée "Sequential Importance Sampling". L'algorithme du filtrage particulaire classique procède en trois étapes fondamentales : (Figure 3.33)

- **Prédiction (mutation) :** Les particules explorent l'espace d'état de façon indépendante, en imitant le comportement suivi par l'état caché, c'est-à-dire en suivant le modèle *a priori*.
- **Correction (pondération) :** Lorsqu'une nouvelle observation est disponible, l'adéquation de chaque particule avec cette observation est évaluée grâce à la fonction de vraisemblance : chaque particule est alors affectée d'un poids proportionnel à la valeur calculée.
- **Redistribution (sélection) :** Les particules sont éliminées ou multipliées, en fonction de leur poids, c'est-à-dire que les particules auront d'autant plus de descendants à la génération suivante que leur vraisemblance est grande.

La version classique du filtrage particulaire est fondée sur l'approximation d'une densité par une somme pondérée de mesures de Dirac (Figure 3.34). On suppose qu'à l'instant $t-1$, on a une approximation de la distribution conditionnelle $\mathcal{P}(x_{t-1}|z_{1:t-1})$

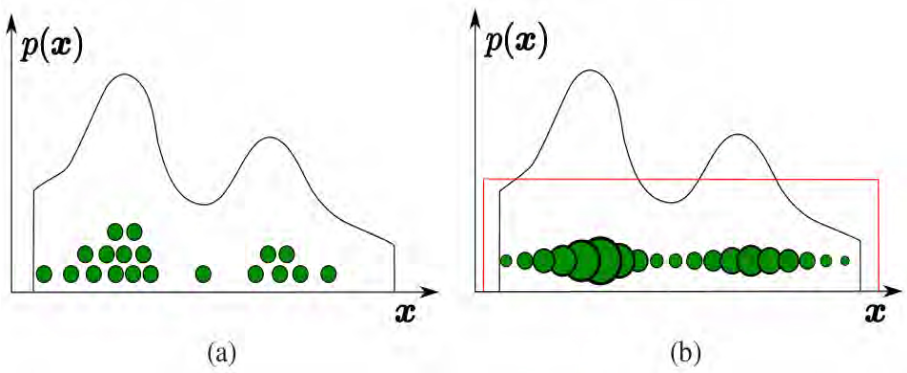


FIGURE 3.34 – Approximation particulaire d'une distribution $p(x)$. Les points représentent les particules, et leur taille est proportionnelle à leur poids. En (a), les particules sont des échantillons i.i.d. selon $p(x)$. En (b), les particules sont placées selon une fonction d'importance, une densité uniforme dans ce cas, et les poids sont corrigés en conséquence afin que le nuage pondéré représente $p(x)$.

de l'état x_{t-1} sachant les observations $z_{1:t-1} = z_1, ..., z_{t-1}$ de la forme

$$\mathcal{P}(x_{t-1}|z_{1:t-1})) \approx \sum_{i=1}^{N} \omega_{t-1}^{(i)} \delta(x_{t-1} = x_{t-1}^{(i)})$$

où δ désigne une mesure de Dirac en x.

Par la manière d'approcher $\mathcal{P}(x_t|z_{1:t})$ par les particules, les 3 étapes suivantes montrent que la récursivité vient du fait que les particules $x_t^{(i)}$ sont produites à partir de $x_{t-1}^{(i)}$, particules à l'instant $t-1$, et l'observation z_t à l'instant t.

- **Etape de prédiction :**
  Considérons $\mathcal{P}(x_t, x_{t-1}|z_{1:t-1})$ comme la densité conjointe (conditionnelle) alors

  $$\mathcal{P}(x_t|z_{1:t-1}) = \int \mathcal{P}(x_t, x_{t-1}|z_{1:t-1}) dx_{t-1}.$$

  Par définition de la probabilité conditionnelle, on a

  $$\mathcal{P}(x_t, x_{t-1}|z_{1:t-1}) = \mathcal{P}(x_t|x_{t-1}, z_{1:t-1})\mathcal{P}(x_{t-1}|z_{1:t-1})$$

  Par hypothèse, on a

  $$\mathcal{P}(x_t|x_{t-1}, z_{1:t-1}) = \mathcal{P}(x_t|x_{t-1}).$$

  Donc

  $$\mathcal{P}(x_t|z_{1:t-1}) = \int \mathcal{P}(x_t|x_{t-1})\mathcal{P}(x_{t-1}|z_{1:t-1}) dx_{t-1}.$$

  Supposons que $\mathcal{P}(x_{t-1}|z_{1:t-1}) \approx \sum_{i=1}^{N} \omega_{t-1}^{(i)} \delta(x_{t-1} = x_{t-1}^{(i)})$, on a alors

  $$\mathcal{P}(x_t|z_{1:t-1}) \approx \sum_{i=1}^{N} \omega_{t-1}^{(i)} \mathcal{P}(x_t|x_{t-1}^{i})$$

  $$\approx \sum_{i=1}^{N} \omega_{t|t-1}^{(i)} \delta(x_t = x_{t|t-1}^{(i)})$$

Avec $x_{t|t-1}^{(i)}$ qui sont obtenues par des réalisations indépendantes de la loi de transition $\mathcal{P}(x_t|x_{t-1}^i)$ et les poids $\omega_{t|t-1}^{(i)} = \omega_{t-1}^{(i)}$.

- **Etape de correction**
  Ici on passe de la loi de densité prédite $\mathcal{P}(x_t|z_{1:t-1})$ à la loi de densité conditionnelle $\mathcal{P}(x_t|z_{1:t})$ grâce à la vraisemblance $g(z_t - H_t(x_{t|t-1}^{(i)}))$ donnée par le modèle du bruit de mesure. La loi de densité conditionnelle est alors approchée

$$\mathcal{P}(x_t|z_{1:t}) \approx \sum_{i=1}^{N} \omega_t^{(i)} \delta(x_t = x_t^{(i)})$$

  et de poids

$$\omega_t^{(i)} = \frac{\omega_{t|t-1}^{(i)} g(z_t - H_t(x_{t|t-1}^{(i)}))}{\sum_{j=1}^{N} \omega_{t|t-1}^{(j)} g(z_t - H_t(x_{t|t-1}^{(j)}))}$$

- **Etape de redistribution**
  Ici il s'agit d'une étape de rééchantillonnage ou de redistribution des particules qui est nécessaire car les algorithmes présentés dans les étapes précédentes souffrent d'un grave défaut : en quelques itérations k, presque tous les poids ω_k^i sont nuls. Idéalement les poids doivent tous rester proches de $1/N$, i.e. les particules sont d'égale importance dans l'approximation. Nous considérons le critère suivant :

$$N_k^{eff} = \frac{1}{\sum_{i=1}^{N} (\omega_k^i)^2} \in [1, N]$$

  qui représente le nombre efficace de particules. Lorsque N_k^{eff} est proche de N alors les particules sont d'égale importance. Il y a dégénérescence des poids lorsque N_k^{eff} est proche de 1. Ce phénomène, appelé "dégénérescence des pondérations", provoque la divergence du filtre. Afin de remédier à ce problème, plusieurs méthodes ont été proposées dans le but de régulariser les poids, comme le tirage multinomial, la redistribution des résidus et la redistribution de Kitagawa. Nous allons les présenter dans la section prochaine.

Les filtres particulaires utilisant la méthode de redistribution des particules s'appellent filtres S.I.R (Sampling Importance Resampling) ou filtres S.I.S (Sampling Importance Sampling).

3.4.1.4 Redistribution des particules

Si le système de particules pondérées $[(x^1, \omega^1), ..., (x^N, \omega^N)]$ est tel que les poids ont une grande variance, on va générer un autre système de particules de même taille en favorisant les particules de grands poids. Si N_k^{eff} est petit alors la plupart des particules a un poids proche de 0 et a donc une contribution négligeable dans l'approximation. L'idée des méthodes de redistribution est donc de "défavoriser" ces particules au bénéfice des particules "importantes" qui ont un poids non négligeable. On "favorise" ces dernières en les dupliquant au détriment de celles que l'on souhaite "défavoriser". Nous allons décrire brièvement les principales méthodes de redistribution des particules utilisées dans le filtre particulaire classique comme le résume [Doucet and Johansen, 2008] :

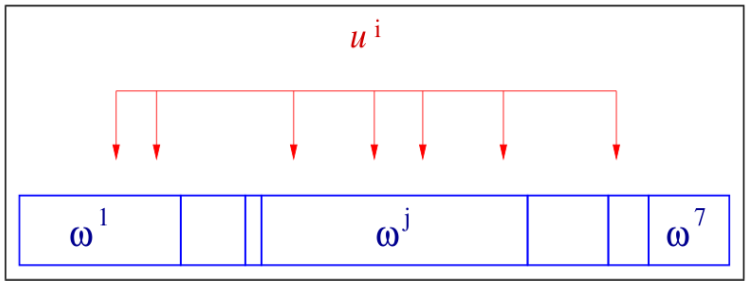


FIGURE 3.35 – Rééchantillonnage multinomial.

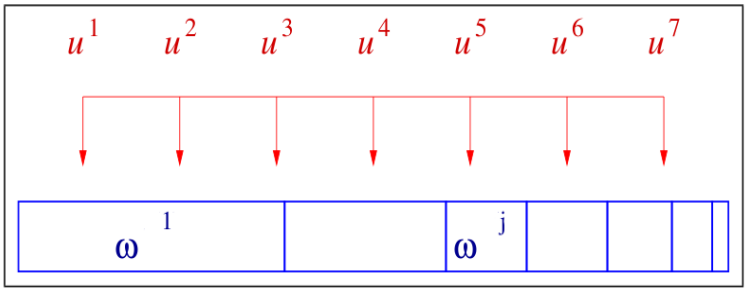


FIGURE 3.36 – Rééchantillonnage de Kitagawa.

- **Le tirage multinomial :** Cette technique est la plus utilisée, chaque particule est sélectionnée en fonction de l'importance de son poids ω^i (voir la Figure 3.35). Voici un exemple :
  Soit un échantillon de particules $x^i, i = 1, ..., 5$ auquelles sont associés les poids $\{\omega^i\}_{i=1}^5 = \{0.105, 0.26, 0.085, 0.43, 0.12\}$ respectivement. On génère 5 *v.a.* uniformes $u^i \in [0, 1], i = 1, ..., 5$ ordonnées $\{u^i\}_{i=1}^5 = \{0.07, 0.27, 0.32, 0.68, 0.88\}$ qu'on compare avec la somme cumulative des poids. Si u^j tombe sur l'intervalle $[\sum_{i=1}^{j} \omega^i, \sum_{i=1}^{j-1} \omega^i]$, la particule de poids ω^j est sélectionnée. On obtient alors un nouveau échantillon de particules $\{x'^i\}_{i=1}^5 = \{x^1, x^2, x^2, x^4, x^5\}$, la particule x^2 est dupliquée et la particule x^3 est éliminée. Les poids seront ré-initialisés en $1/N$. Si on note par N_i' le nombre de particules dupliquées pour chaque particule i, tel que si $N_i' = 0$ la particule i est éliminée. Nous obtenons les résultats suivants $\mathbb{E}[N_i'] = N\omega^i$ et $var(N_i') = N\omega_i(1 - \omega_i)$.
- **La redistribution de Kitagawa :** Cette méthode ne génère pas des variables uniformes ordonnées mais plutôt un échantillon de variables déterministes (voir la Figure 3.36). On simule uniquement u^1 et assigne de façon déterministe les $N - 1$ autres u^i de la manière suivante

$$u^1 \sim \mathcal{U}[0, \frac{1}{N}] \qquad u^i = u^1 + \frac{i}{N}, i = 2 : N$$

- **La redistribution des résidus :** Cette méthode utilise l'idée de Kitagawa pour générer un certain nombre de particules puis elle fait appel à la méthode de redistribution multinomiale pour déterminer les particules restantes. Dans un premier temps, pour chaque i, on sélectionne la particule x^i de façon déterministe $\lfloor \omega^i / N \rfloor$

fois. Dans un deuxième temps, il reste donc

$$N' \leftarrow \sum_{i=1}^{N} \lfloor \frac{\omega^i}{N} \rfloor$$

particules à sélectionner. On applique la méthode de redistribution multinomiale à la loi de probabilité correspondant aux poids restants

$$\omega'^i \leftarrow \omega^i - N\lfloor \frac{\omega^i}{N} \rfloor \qquad et \qquad \omega'^i \leftarrow \omega'^i / \sum_{j=1}^{N} \omega'^j.$$

Cette approche n'a d'intérêt que s'il ne reste que peu de particules N' à générer dans la deuxième étape. C'est le cas lorsque N^{eff} est petit et c'est justement dans ce cas qu'il est nécessaire de redistribuer.

3.4.2 Expérimentation sur le suivi d'objets

Une des applications du filtre particulaire est le suivi des objets en mouvement dans une séquence d'images. Dans cette section, nous allons présenter la réalisation du filtrage particulaire, puis nos expérimentations sur le suivi d'objets, ainsi que notre méthode hybride de détection et de suivi de l'œil présentée dans le schéma général de notre méthode (Figure 3.3).

3.4.2.1 Réalisation du filtre particulaire

En pratique, chaque particule représente une région rectangulaire dans l'image. La structure d'une particule contient notamment les informations suivantes :

- les coordonnées de la position actuelle de la particule (x, y) ;
- les coordonnées de la position précédente de la particule (x_p, y_p) ;
- les coordonnées de la position originale de la particule (x_0, y_0) ;
- la taille de la région (width, height) ;
- le vecteur signature de la région qui peut être décrit par les histogrammes CS-LBP ;
- le poids de la particule ω.

Comme nous l'avons rappelé dans les sections précédentes, l'algorithme du filtrage particulaire a besoin de définir deux modèles : un modèle dynamique $\mathcal{P}(x_t|x_{t-1})$ (ou le modèle de transition) qui explique comment l'objet bouge, et un modèle d'observation qui mesure ou estime la position de l'objet.

Nous choisissons un modèle dynamique auto-régressif d'ordre second pour modéliser le mouvement des particules. Les coordonnées de la particule à l'instant $t+1$, $X_{t+1} = \{x, y\}$ est défini par

$$X_{t+1} = AX_t + BX_{t-1} + Cv_t$$

où A, B, C sont les constantes et v_t est le bruit qui suit la loi normale, $v_t \sim \mathcal{N}(0, \sum)$. Ce modèle dynamique a été largement utilisé dans la litérature consacrée au suivi des objets. [Hess and Fern, 2009] ont par exemple utilisé ce modèle pour suivre les joueurs de rugby.

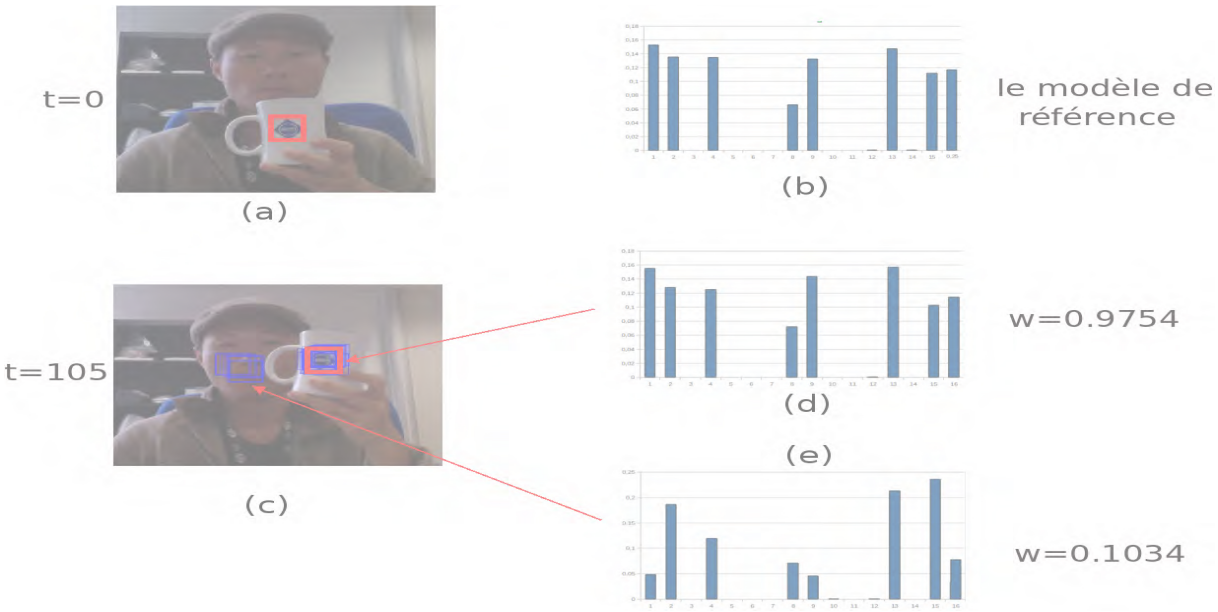


FIGURE 3.37 – Le suivi d'un objet par le filtre particulaire. a) Dans la première image (t=0) nous choisissons la cible (la region rouge) à suivre ; b)l'histogramme de cette region est utilisé comme le modèle de référence ; c) Les résultats du suivi dans la 105-ème image, la region rouge est la position estimée de la cible, et les regions bleues sont les candidats ; d)l'histogramme de la region rouge ayant un poids important ; e)l'histogramme d'une region bleue ayant un poids faible.

Le modèle d'observation est obtenu par la combinaison des histogrammes CS-LBP sur les blocs de la région associée à la particule (voir la section 3.3.3). Cette combinaison des histogrammes forme le vecteur signature de la région. A l'instant t, le modèle d'observation $q_t(X_t)$ est comparé avec le modèle de référence q_0, qui est calculé à l'instant initial 0, soit $q_0(X_0)$. Dans la pratique ce modèle de référence peut être sélectionné manuellement au départ, ou fourni automatiquement par un module de détection d'objet. La similarité de ces deux modèle d'observation q_t et q_0 est définie par la distance de ses vecteurs de signature D, qui est dérivée du coefficient de Bhattacharyya :

$$D(q_0, q_t) = (1 - \sum_{n=1}^{N} \sqrt{q_0[n] * q_t[n]})^{\frac{1}{2}} \tag{3.1}$$

où N est la dimension du vecteur signature. Cette distance est dans $[0, 1]$. Alors le poids ω associé au chaque particule est :

$$\mathcal{P}(z_t|X_t) \propto exp^{-\lambda D^2(q_0,q_t)}$$

En pratique la valeur de λ est 150.

La Figure 3.37 démontre le principe du fonctionnement des particules. La région rouge à l'instant $t = 0$ est choisie manuellement dans la première image de la vidéo, et l'histogramme calculé est utilisé comme le modèle de référence. Dans les images suivantes $t > 0$, tous les particules candidates se déplacent et calculent leurs poids d'importance ω selon la vraisemblance des histogrammes. La particule qui a le poids plus important à l'instant t est choisie comme région identifiée de l'objet et les particules qui ont les poids faibles vont être éliminés lors de la prochaine étape.

3.4.2.2 Suivi d'un objet

Dans cette section nous comparons les résultats en utilisant différentes méthodes du calcul du vecteur signature de la région d'intérêt : HSV, LBP uniforme, CS-LBP.

- Histogramme HSV (Hue-Saturation-Value) : L'espace de couleur HSV caractérise les couleurs de façon plus intuitive, conformément à la perception naturelle des couleurs, en termes de teinte h (Hue), de saturation s et de luminosité v (Value). [Pérez et al., 2002] proposent de calculer l'histogramme de N unités (bins) comme :

  $$N = N_h N_s + N_v$$

  avec les valeurs $h > Threshold_h$ et $s > Threshold_s$. Dans l'expérimentation, $Threshold_h = 0.1$, $Threshold_s = 0.2$ et le nombre d'unités, soit la dimension de l'histogramme $N = 110$.
- Histogramme LBP : La méthode LBP permet de caractériser les textures présentes dans l'image en niveau de gris. L'histogramme calculé représente le vecteur signature de l'image. Ici nous proposons d'utiliser les deux histogrammes LBP (LBP uniforme et LBP centré-symétrique) pour suivre un objet. Ces deux variantes de LBP sont présentées dans la section 3.3.1. Nous divisons la région d'intérêt de l'image en n blocs. Donc la dimension de l'histogramme est calculée :

  $$N_{LBP_{uniforme}} = n * 59 \qquad N_{CS-LBP} = n * 16$$

La Figure 3.38 illustre les résultats du suivi du logo bleu sur la tasse. La région du logo recherché est définie au départ avec une taille de 38×35 pixels. La colonne (A) illustre le résultat en utilisant l'hitogramme HSV dont la dimension est 110. A partir de la 445ème image où le logo est presque caché, le suivi de l'objet est perdu. Les résultats qui utilisent les histogrammes CS-LBP et LBP uniformes montrent que les particules candidates se dispersent autour de l'objet. Les suivis dans les images sont perturbés à cause du changment de trajectoire de l'objet. Nous divisons la région en 4 blocs (D) et 8 blocs (E) : la combinaison des histogrammes CS-LBP permet de bien suivre l'objet, et le suivi est plus précis.

La Figure 3.39 montre les résultats lorsqu'il faut suivre un objet dans une condition où l'illumination change. Nous pouvons observer la position estimée de la cible par le descripteur CS-LBP est moins perturbée et plus précise qu'avec le descripteur HSV.

3.4.2.3 Détection et suivi des yeux

Nous avons présenté dans la Figure 3.3 la méthode pour détecter et suivre l'œil, qui se compose de 3 parties importantes :

- Détection de l'œil. L'objectif est d'obtenir un modèle de l'œil de référence dans la première image. Le modèle est construit par les deux caractéristiques : l'histogramme CS-LBP et les points d'intérêt SURF dans l'image de l'œil.
- Suivi de l'œil. Dès que la région de l'œil est localisée, le suivi de l'œil est effectué par le filtre particulaire dans les images suivantes. Nous utilisons le modèle dynamique auto-régressif d'ordre second pour prédire la position de l'œil et le modèle

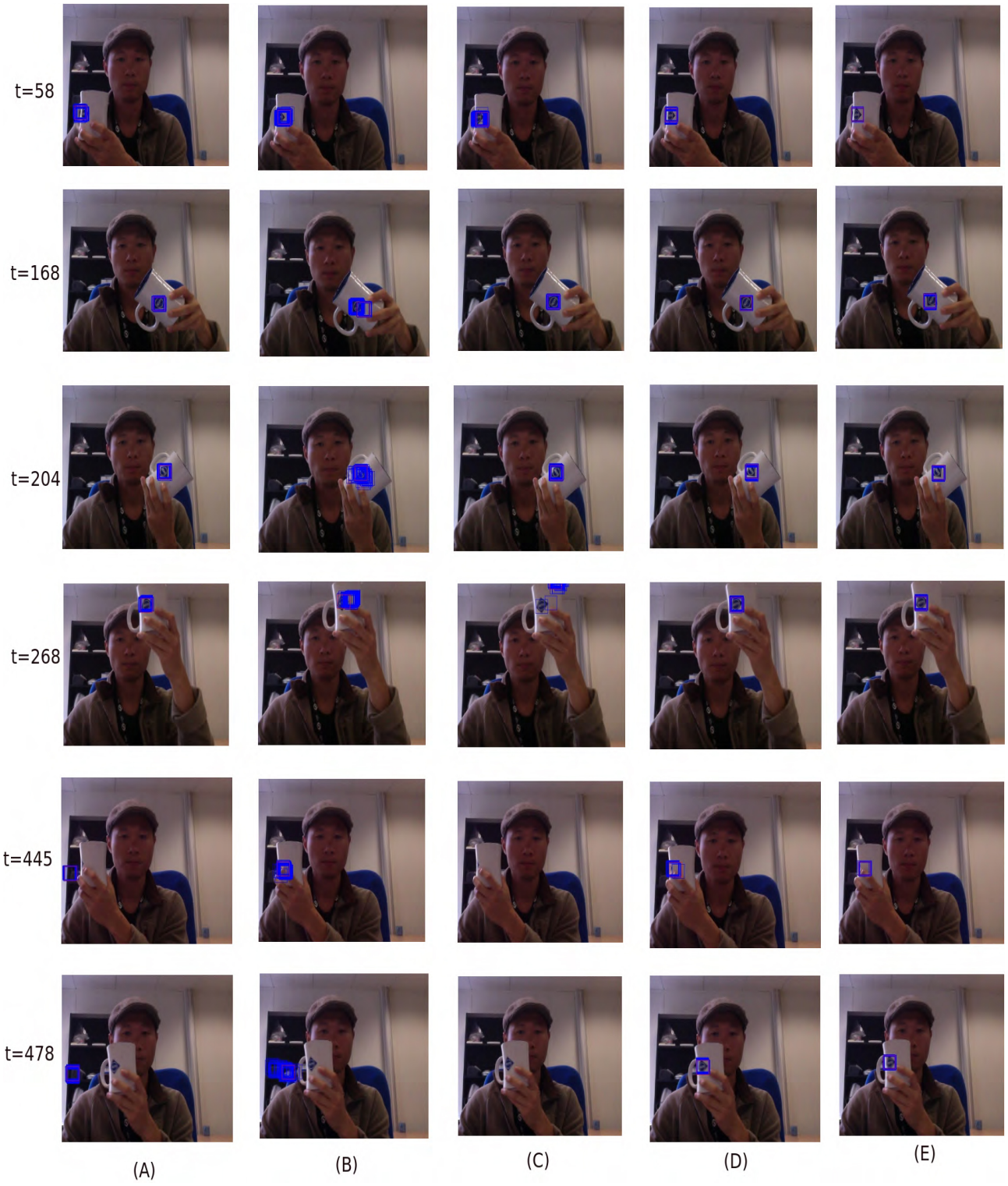


FIGURE 3.38 – Les résultats du suivi du logo par le filtre particulaire en utilisant différent descripteurs. Chaque colonne représente la séquence d'images, et chaque ligne représente la même image. t désigne l'ordre des images. Les 5 descripteurs sont : (A) HSV, la dimension de l'histogramme n est 110. (B) CS-LBP dont la dimension $n = 16$ (C) LBP uniforme dont la dimension $n = 59$ (D) CS-LBP en 4 blocs, $n = 64$ (E) CS-LBP en 8 blocs, $n = 128$.

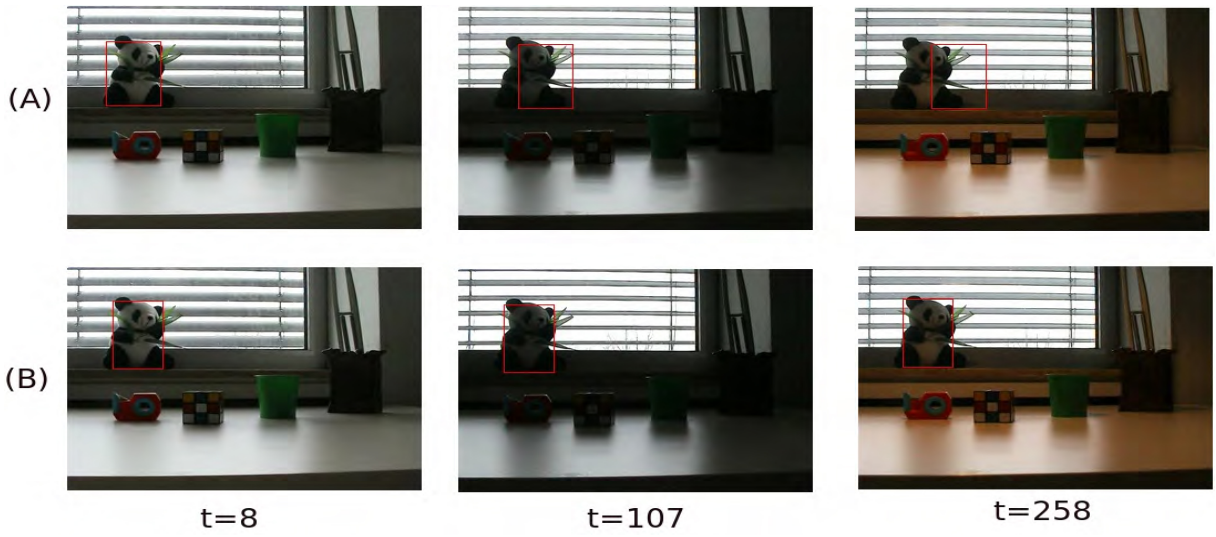


FIGURE 3.39 – Le suivi de l'objet (le rectangle rouge) dans le cas où l'illumination change. Chaque ligne représente la séquence de la vidéo, et chaque colonne représente la même image, t désigne l'ordre des images. Les 2 descripteurs utilisés sont : (A) : HSV (B) CSLBP en 8 blocs.

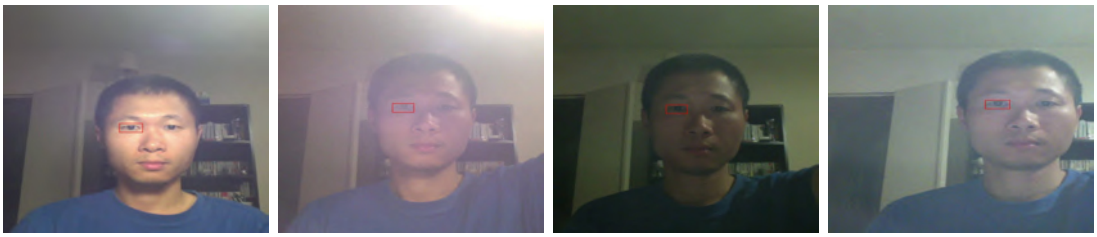


FIGURE 3.40 – Le suivi de l'œil (dans le rectangle rouge) en utilisant le descripteur CS-LBP lorsque l'illumination change.

d'observation par le descripteur CS-LBP. Dans l'expérimentation, le nombre des particules générées est fixé à 100.

- Récupération de l'œil. Les deux paramètres N_{Surf} et D repésentent respectivement le nombre des points SURF et la distance des histogrammes CS-LBP dans l'équation (3.1). Si les deux paramètres sont inférieurs aux seuils définis, on considère que la localisation de l'œil n'est plus valide et que nous avons perdu l'œil. Les valeurs empiriques pour ces seuils sont $N_{Surf} < 1$ et $D < 0.3$. Une fois l'œil est considéré perdu, nous refaisons la detection de l'œil pour obtenir un nouveau modèle.

Dans notre expérimentation, la webcam intégrée dans le PC est capable d'obtenir 30 images par seconde (30fps) et la résolution de l'image capturée est de 960×540 pixels. Les résultats (Figure 3.40) du suivi de l'œil (le rectangle rouge) illustre la robustesse de notre méthode, qui permet de bien suivre l'œil dans les scènes complexes mais également dans le cas où la forme de l'œil change. La Figure 3.41 montre la séquence d'images capturée par la webcam pendant environ 20s. t désigne l'ordre des images capturées. Dans cette séquence nous pouvons remarquer les mouvements du sujet devant la caméra (la rotation, le clignements etc) et le changement d'illumination quand le sujet change de la position. Par rapport le temps du calcul nécessaire, nous pouvons obtenir le résultat avec une

fréquence moyenne de 21 fps, ce qui est plus rapide que celui obtenu par la méthode de [Viola and Jones, 2004], dont la fréquence moyenne est environ 10- 15 fps.

3.5 Conclusion

Dans ce chaptire, nous avons présenté notre méthode pour localiser l'œil dans une séquence d'images capturée par la webcam. Cette méthode hybride utilise notamment le filtre particulaire pour suivre l'œil après avoir détecté l'œil par ASM et la carte des yeux. En outre, les deux caractéristiques utilisées pour représenter l'image de l'œil : l'histogramme CS-LBP et des points d'intérêt générés par SURF sont discriminants pour distinguer l'œil dans les conditions complexes comme un changement de forme de l'œil et un changement d'illumination.

Les résultats de l'expérimentation nous montrent l'efficacité et la performance de notre méthode. Par rapport à d'autres méthodes comme celle de Viola et Jones, qui localisent la region de l'œil par une manière exhaustive sur chaque image, notre méthode probabiliste est plus efficace du point de vue temps de calcul, même si nous devions localiser l'œil dans une séquence d'images de haute résolution. De plus, le filtre particulaire peut également être utilisé pour localiser d'autres objets en plus de l'œil.

FIGURE 3.41 – Le suivi de l'œil en utilisant la méthode proposée sur la séquence d'images capturée par la webcam. t désigne l'ordre de la image.

Quatrième partie

Apprentissage par variété

Sommaire

4.1 Introduction

Les images numériques ont généralement une haute dimensionnalité qui est imputable au grand nombre de pixels qui les composent. Cette grande dimensionnalité représente un obstacle pour le traitement, l'organisation, la recherche, l'analyse ou la visualisation de ces données. Ces problèmes sont classiquement abordés par les techniques de sélection de variables et de réduction de dimension, qui visent à trouver des structures intrinsèques de dimension réduite. Dans la section 3.3, nous avons utilisé deux méthodes d'extraction des caractérististiques pour trouver un descripteur de faible dimension qui représente au mieux une image. Cette manière de réduction de la dimensionnalité peut être considérée comme un processus de traitement local.

Dans ce chapitre, nous allons nous intéresser à analyser la structure topologique d'un ensemble d'images des yeux au point de vue global. Cet ensemble peut être représenté par une base de données qui contient un nombre d'individus de haute dimension, ou par une matrice dont chaque vecteur est une image de haute dimension. Nous voulons utiliser les méthodes de réduction de la dimensionnalité pour trouver un descripteur de faible dimension pour chaque image et aussi la relation (distance) entre ces images. Nous pourrons classifier ces images des yeux par rapport aux différences entre ces descripteurs. La classification nous permet d'avoir les connaissances *a priori* sur les mouvements oculaires. Comme illustré dans la Figure 4.1.1, un ensemble des images des yeux qui représentent les directions différentes du regard (par exemple, vers quatre coins de l'écran (Figure 4.1A) peut être projeté dans un sous-espace 3D. Nous obtenons notamment quatre classes (les cercles rouges dans la Figure 4.1B) qui correspondent aux quatre mouvements des yeux. La réduction de la dimensionnalité est la transforma-

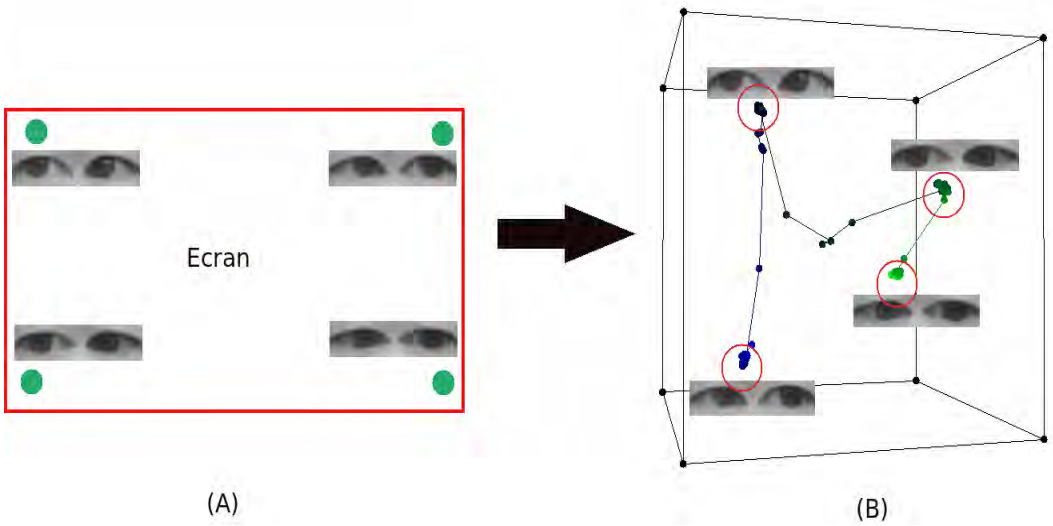


FIGURE 4.1 – A) un ensemble d'images des yeux lorsque les yeux regardent vers les 4 coins sur l'écran (points verts). B) la projection de cet ensemble d'images des yeux dans un sous-espace 3D par la méthode de réduction de la dimensionnalité non-linéaire. On notera les 4 classes trouvées (cercles rouges) qui correspondent aux regards vers les 4 coins.

tion de données de grande dimension dans une représentation significative de dimension réduite. Idéalement, la représentation réduite devrait avoir une dimensionnalité qui corresponde à la dimensionnalité intrinsèque des données qui est le nombre minimum de

paramètres nécessaires pour tenir compte des propriétés observées[Fukunaga, 1990]. En conséquence, la réduction de la dimensionnalité facilite la classification, la visualisation, et la compression de données de grande dimension.

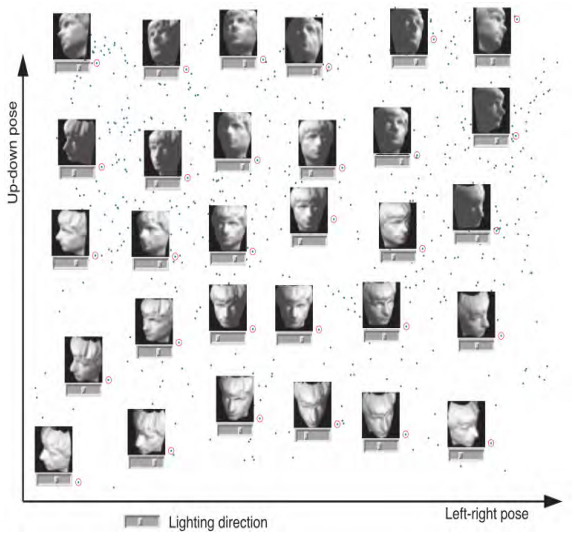
Le problème de la réduction de la dimensionnalité peut être défini comme suit. Supposons que nous ayons un ensemble de données X représentées dans une matrice de taille $N \times D$ comprenant N vecteurs $x_i (i \in 1, 2, ..., N)$ avec une dimensionnalité D. Mathématiquement, la dimensionnalité intrinsèque signifie que les points des données X sont situés sur ou près d'une variété (manifold) avec une dimensionnalité d (où $d < D$, et souvent $d \ll D$) qui est incorporée dans l'espace D-dimensionnel. Les **variétés** sont des généralisations de surfaces. Elles sont des espaces topologiques localement euclidiens (point de vue intrinsèque), mais, dans une échelle plus large, elles sont un peu différentes (point de vue extrinsèque). Un exemple de variété est la surface d'une sphère en R^3, qui localement est topologiquement équivalent au R^2 mais dont les propriétés globales sont très différentes de R^2. L'**apprentissage par variété**, dont l'idée clé est d'apprendre une variété de basse dimension qui préserve les propriétés de données, regroupe un ensemble de techniques de réduction de la dimensionnalité. Ces techniques transforment les ensembles de données X dont la dimensionnalité est D en un nouvel ensemble de données Y avec une dimensionnalité d, tout en conservant la géométrie des données autant que possible. Notons un point de données de grande dimensionnalité par x_i , où x_i est la i^e ligne de la matrice X. Le correspondant de basse dimensionnalité de x_i est noté y_i , où y_i est la i^e ligne de la matrice d-dimensionnelle Y. La représentation de la variété de faible dimension permet de fournir des informations utiles sur la nature et l'organisation des données et peut être exploitée pour des tâches de classification ou de regroupement. En général, ni la géométrie de la variété de données, ni la dimension intrinsèque d de l'ensemble de données X ne sont connues. Par conséquent, la réduction de la dimensionnalité est un problème mal posé qui ne peut être résolu qu'en supposant certaines propriétés des données (telles que la dimension intrinsèque).

Traditionnellement, la réduction de la dimensionnalité est réalisée par des techniques linéaires telles que l'Analyse en Composantes Principales (ACP) [Pearson, 1901], l'analyse factorielle [Spearman, 1904], et la mise à l'échelle classique[Young and Householder, 1938]. Cependant, les limitations de ces techniques sont importantes et bien connues : seules les variétés linéaires sont gérées. Par exemple le SwissRoll [Figure 4.6], qui est une variété non linéaire en forme de spirale à deux dimensions, ne peut clairement pas être décrit par l'ACP.

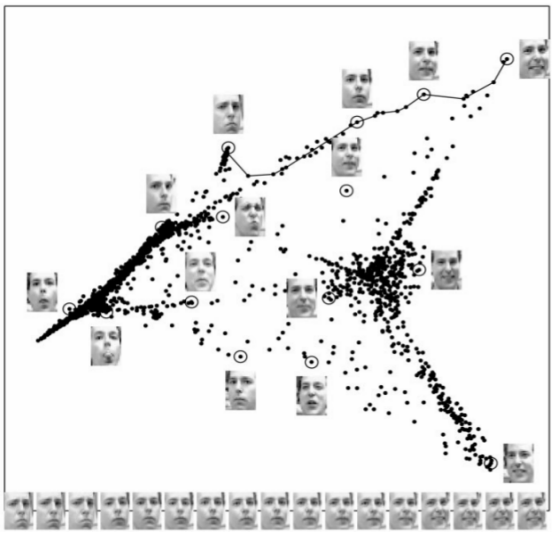
Dans la dernière décennie, un grand nombre de techniques non linéaires de réduction de dimensionnalité ont été proposées, comme Isomap (Isometric Feature Mapping) [Tenenbaum et al., 2000], LLE (Local Linear Embedding) [Roweis and Saul, 2000], le Laplacian Eigenmaps [Belkin and Niyogi, 2001, Belkin and Niyogi, 2003], les cartes de diffusion (Diffusion Maps) [Nadler et al., 2005], etc. Contrairement aux techniques linéaires traditionnelles, les techniques non linéaires ont la capacité de traiter des données complexes non linéaires. En particulier, les techniques non linéaires sont plus performantes sur les données réelles susceptibles de former une variété fortement non linéaire.

Aujourd'hui ces techniques non-linéaires de réduction de la dimensionnalité sont largement utilisées pour résoudre les problèmes dans les domaines comme la reconnaissance de forme, la vision par ordinateur, etc. Tenenbaum et al.[Tenenbaum et al., 2000] ont appliqué IsoMap pour analyser la variation des positions du visage (Figure 4.2 gauche),

des écritures et des mouvements des mains. Roweis et al.[Roweis and Saul, 2000] ont proposé l'utilisation de LLE sur un ensemble d'images pour reconnaitre les expressions faciales (Figure 4.2 droite). Les résultats ont prouvé que ces méthodes non-linéaires permettent de générer une variété dont la structure représente la variation de ces objets présentés dans les images. Les autres références peuvent être trouvées dans [Souvenir and Pless, 2007], [Yang, 2002], [Rahimi et al., 2005], [Pless, 2003], [Rane and Birchfield, 2007], [Zhang et al., 2004], [Pless and Simon, 2002], [Chen et al., 2005].



source :[Tenenbaum et al., 2000]



source :[Roweis and Saul, 2000]

FIGURE 4.2 – Gauche : l'analyse de la variation des positions du visage ; Droite : l'analyse des expressions faciales.

Ici nous présentons nos études sur l'analyse des mouvements oculaires en appliquant les méthodes non-linéaires comme le Laplacian Eigenmaps et le Diffusion Maps. Dans les sections suivantes, nous abordons d'abord les aspects théoriques concernant les méthodes linéaires et non-linéaires de réduction de la dimensionnalité. Ensuite, nous présentons nos expérimentations sur l'ensemble des images de l'œil par ces techniques.

4.2 Méthodes linéaires

L'analyse en composantes principales (PCA) et l'algorithme d'échelle multidimensionnelle (MDS) sont des méthodes linéaires équivalentes de réduction non supervisée de la dimensionnalité.

4.2.1 Analyse en composantes principales

L'analyse en composantes principales[Pearson, 1901] est une méthode projective non supervisée dont le critère à maximiser est la variance originale dans les données projetés. On fait l'hypothèse que les données de départ se trouvent dans un hyperplan et que l'on

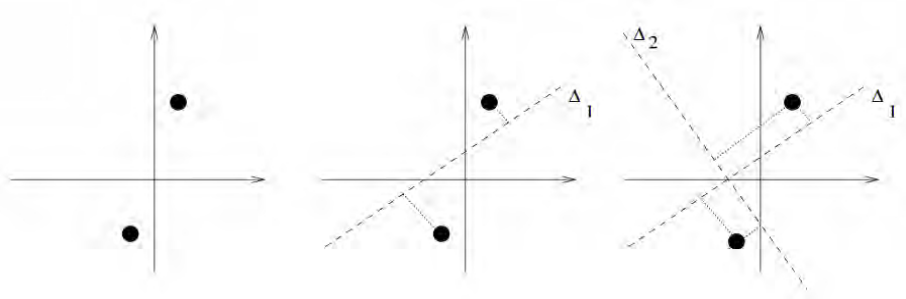


FIGURE 4.3 – A gauche, on indique deux points dans le plan. Au milieu, on a construit la droite Δ_1 qui passe par l'origine et minimise la somme des distances au carré entre chaque point et elle (pointillés fins). A droite, on a construit Δ_2, orthgonale à Δ_1 et qui minimise également la somme des distances au carré à chaque point. Δ_1 et Δ_2 passent par l'origine.

peut les exprimer au moyen des vecteurs qui définissent cet hyperplan. Si l'hypothèse se révèle vraie, on trouve la dimensionnalité intrinsèque des données.

L'idée fondamentale de l'ACP est la suivante : considérant un nuage de N points en D dimensions, on cherche à trouver le plan dans lequel la projection des points du nuage est la moins déformée possible, donc la plus fiable possible. Pour cela, on commence par rechercher la droite Δ_1 qui minimise la somme des distances au carré entre chaque point et la droite (Figure 4.3). Cette droite a la propriété d'être la direction selon laquelle la variance des points est la plus grande. Puis, on cherche une deuxième droite Δ_2 perpendiculaire à Δ_1 qui possède la même propriété. Plus généralement, on définit P droites orthogonales les unes aux autres qui permettent de définir un repère orthonormé. Ces P droites sont les P "axes principaux" d'un repère dans lequel sont situés les échantillons de manière à les décrire de la façon la plus concise. C'est-à-dire que ces composantes principales sont les combinaisons linéaires des variables initiales de variance maximale.

L'algorithme ACP est comme le suivant :

1) Calculer le vecteur moyen $\mu = \{\mu_k\}$, pour chaque demension de données X.

$$\mu_k = \frac{\sum_{i=1}^{N} x_i^k}{N} \qquad k \in D \tag{E 4.1}$$

2) Effectuer la soustraction $B = X - \mathbf{1} \cdot \mu$ où $\mathbf{1}$ est un vecteur colonne $N \times 1$ de 1.
3) Calculer la matrice de variance-covariance $C = \frac{1}{N-1} BB^T$.
4) Calculer les vecteurs propres V et valeurs propres Λ de C :
   $V = \{v_1, v_2, ..., v_D\}$ et $\Lambda = (\lambda_1, \lambda_2, ..., \lambda_D)$ où $\lambda_1 > \lambda_2 > ... > \lambda_D$.
5) Choisir les d premières composantes principales :

$$V^* = \{v_1, v_2, ..., v_d\} \quad (d < D)$$

6) Générer un nouvel ensemble de données X^* de faible dimension à partir des données initiales X.

$$X^* = V^{*T} X$$

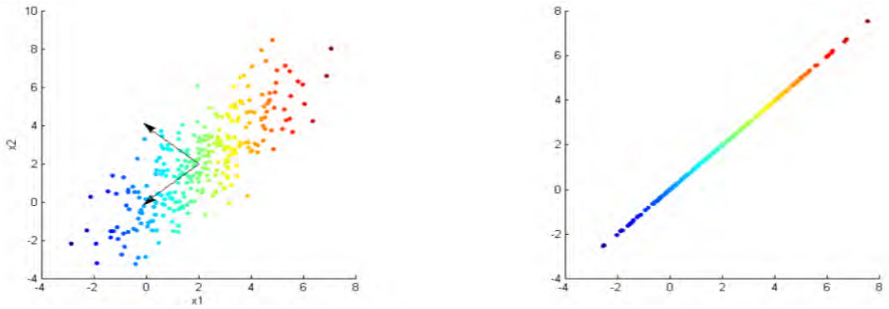


FIGURE 4.4 – Exemple de ACP sur une distribution gaussienne (à gauche) avec ses 2 premières composantes principales (à droite).

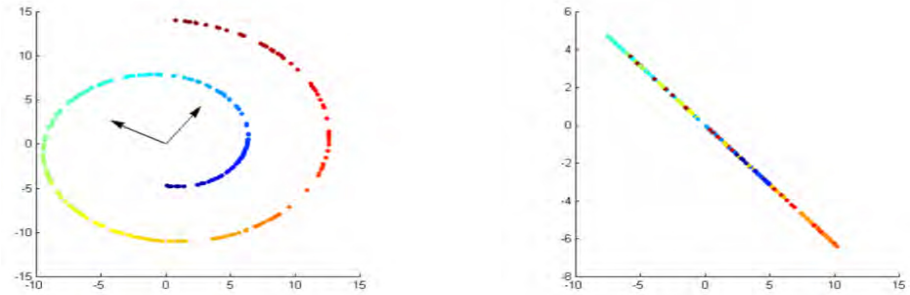


FIGURE 4.5 – Exemple de ACP sur un ensemble de données non-linéaire.

La Figure 4.4 montre un exemple de ACP sur une distribution gaussienne. Les couleurs de données dépendent des valeurs (x_1, x_2) de manière linéaire. La droite de la figure montre la projection des vecteurs propres qui sont les deux premières composantes principales. Comme on l'a vu, la transformation respecte la structure des données initiales. La Figure 4.5 montre un autre exemple de ACP sur les données non-linéaires, par exemple sur la distribution spirale $2-D$ (à gauche). Ici les couleurs dépendent de la valeur t, où la fonction est $f(t) = (tcos(t), tsin(t))$. Par la projection (à droite), on peut observer que les couleurs se chevauchent. Par conséquent les informations géométriques sont perdues pendant la projection.

L'ACP étant une méthode de réduction de la dimensionnalité, il est important de savoir qu'elle ne peut pas retenir la totalité de l'information contenue dans le nuage de points initial. Enfin, l'ACP prend uniquement en compte les dépendances linéaires entre les variables et ne peut donc pas fournir une projection fidèle pour une distribution non-linéaire de points.

4.2.2 Algorithme d'échelle multidimensionnelle

Les techniques d'échelle multidimensionnelle, ou positionnement multidimensionnel (MDS : Multi-Dimensional Scaling) visent à représenter chaque individu (ou vecteur) dans un espace euclidien, habituellement bi ou tridimensionnel, de telle sorte que deux individus semblables soient représentés par deux points proches l'un de l'autre, et un couple dissemblable par des points éloignés. Cette idée a évolué pour considérer des « dissimilarités » ou d'autres types de vraisemblances entre des points à la place des distances euclidiennes. Si les dissimilarités sont exprimées selon une échelle ordinale

(rangs ou classements), les modèles de MDS utilisés sont qualifiés de **non-métriques** ; s'il s'agit d'une échelle de mesure (intervalles, ratios), alors les modèles de MDS sont qualifiés de **métriques**.

Cette approche essaie de préserver la distance entre les points de grande dimension dans une structure de faible dimension. Le problème mathématique fondamental est de trouver la matrice de coordonées X^* de faible dimension pour X minimisant la formule suivante :

$$X^* = \{x_1^*, ..., x_N^*\} = argmin_{\{x^*\}} \sum_{i,j=1}^{N} (x_i \cdot x_j - x_i^* \cdot x_j^*)^2$$

La réalisation concrète de cette idée est de trouver une représentation euclidienne d'un ensemble de points à partir de leurs distances respectives. Ce problème a été résolu en 1938 par Young et Householder[Young and Householder, 1938] au moyen d'une méthode très proche de l'analyse factorielle, faisant appel à la décomposition spectrale d'une matrice symétrique de distances.

La technique d'échelle multidimensionnelle classique[Torgerson, 1952] calcule la matrice de coordonées X^* de faible dimension par résoudre le problème de la décomposition de la matrice de produit scalaire (Annexe 2) $B = XX^T$ en valeurs propres. La matrice B est construite à partir de la matrice de distance S et la matrice centrée $H = I_N - \frac{1}{N}11^{'}$ au moyen d'une double opération de centrage (*double centering*), qui consiste à enlever la moyenne des lignes et la moyenne des colonnes, puis à rajouter la moyenne générale du tableau.

Voici le résumé de cette technique d'échelle multidimensionnelle. La dimension de données initiales X est noté D, et le nombre des données est N :

1) Calculer la matrice de proximité $S = (\delta_{i,j}^2) \quad i, j = 1, ..., N$ par la distance Euclidienne, où

$$\delta_{i,j} = \sum_{d=1}^{D} (x_i^d - x_j^d)^2.$$

2) Calculer la matrice de produit scalaire $B = -\frac{1}{2}HSH$, où la matrice centrée H

$$H = I_N - \frac{1}{N}11^{'},$$

I_N est la matrice unité de taille N et $\mathbf{1}$ est le vecteur de 1 de taille N.

3) Décomposer la matrice de produit scalaire B pour obtenir les vecteurs propres V et valeurs propres Λ :

$$B = V\Lambda V^{'}$$

où $V = \{v_1, v_2, ..., v_D\}$ et $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_D)$.

4) Extraire les d première vecteurs propres et valeurs propres :
$V_d = \{v_1, v_2, ..., v_D\}(d < D)$, $\Lambda_d = diag(\lambda_1, \lambda_2, ..., \lambda_d)$ et $\lambda_1 > \lambda_2 > ... > \lambda_d$.

5) Générer la nouvelle matrice de coordonées de d-dimension.

$$X_d = V_d \Lambda_d^{\frac{1}{2}}$$

Il y a une équivalence entre le MDS classique (analyse des proximités avec une métrique euclidienne) et l'ACP[Desbois, 2005]. Les d coordonnées principales des individus obtenues en MDS classique sont simplement les d composantes principales des individus dans une ACP.

4.3 Méthodes basées sur graphe

Il est fréquent que la dimension intrinsèque des données soit plus petite que la dimension de l'espace d'entrée. Autrement dit, les données représentées dans un espace à D dimensions sont souvent sous-tendues par une variété de dimension $d < D$ plus petite. Les méthodes linéaires comme l'ACP et le MDS classique génèrent des représentations fidèles de faible dimension lorsque les données d'entrée de grande dimension sont principalement confinés dans un sous-espace de faible dimension. Si les données d'entrée sont éparpillés un peu partout dans ce sous-espace, les spectres des valeurs propres de ces méthodes révèlent également la dimension intrinsèque de l'ensemble de données, c'est-à-dire le nombre des modes de variabilité sous-jacents. Mais si la structure de l'ensemble de données est fortement non linéaire, les méthodes linéaires sont vouées à l'échec.

Dans le cas où l'on n'a pas d'*apriori* sur la forme de la variété, les graphes représentent un outil commode pour analyser la structure de voisinage. Le but final de ces méthodes est généralement le même : trouver un plongement du graphe dans lequel la distance euclidienne reflète les distances « naturelles » sur le graphe. Les méthodes basées sur graphes ont récemment émergé comme un outil puissant pour l'analyse des données de grande dimension qui ont été échantillonnées à partir d'une sous-variété de faible dimension. Ces méthodes commencent par la construction d'un graphe peu dense dans lequel les nœuds représentent des modèles d'entrée et les arêtes représentent les relations de voisinage.

4.3.1 Construction du graphe

L'idée de base, issue de la théorie des graphes, est de représenter l'ensemble de données $X = \{x_1, ..., x_N\}$ par un graphe pondéré $G = (V, E, W)$. Les sommet v_i du graphe représentent les vecteurs x_i. Deux sommets v_i et v_j sont adjacents si l'arête $(v_i, v_j) \in E$. On dit que les deux sommets sont voisins (notation $v_i \sim v_j$). Pour déterminer le voisinage pour chaque point, les méthodes souvent utilisées sont k-plus proches voisins (k-ppv) ou ϵ-voisinage qui considère que les deux sommets v_i et v_j sont voisins si et seulement si $\| x_i - x_j \|^2 \leqslant \epsilon$.

La matrice des poids obtenus W est appelée matrice de similarité. Elle est définie par : $W = \omega(v_i, v_j) = \omega_{ij}$, qui obtenu par le **noyau** (voir la définition 4.1). Si les sommets v_i et v_j sont voisins, et $\omega_{ij} = 0$ dans le cas contraire. Les sommets étant liés à eux-mêmes, nous avons pour tout sommet v_i, $\omega_{ii} = 1$.

Définition 4.1 (Noyau). *Un Noyau $k : N \times N \to \mathbb{R}$ sur un ensemble de données*

X est une fonction qui définit la pondération des arêtes pour la matrice de similarité W dans un graphe pondéré. Il possède les propriétés suivantes :

- *symétrique :* $k(x, y) = k(y, x)$
- *positive :* $k(x, y) \geqslant 0$
- *représente la similarité entre les points dans* X

Le degré d'un sommet v_i est défini par : $d(v_i) = d_i = \sum_{v_i \in V} \omega_{ij}$. La matrice de degré D est une matrice diagonale contenant les degrés $d_1, d_2, ..., d_N$. Ainsi, nous définissons la matrice des degrés des sommets D avec : $D_{ii} = D(v_i, v_i) = d(v_i)$ et $D(v_i, v_j) = 0$ pour $v_i \neq v_j$.

Le graphe résultant (supposé connecté, pour simplifier) peut être considéré comme une approximation discrète de la sous-variété échantillonnée par les modèles d'entrée. À partir de ces graphes, nous pouvons alors construire des matrices dont les décompositions spectrales révèlent la structure de faible dimension de la sous-variété (et parfois même la dimensionnalité elle-même). Bien que capables de révéler la structure fortement non linéaires, les méthodes sur graphes pour l'apprentissage par variété sont basées sur des problèmes d'optimisation facilement solubles tels que les problèmes des plus courts chemins, les moindres carrés, et la diagonalisation des matrices.

Dans ce qui suit, nous allons présenter les différents algorithmes basés sur graphe comme Isomap [Tenenbaum et al., 2000], LLE [Roweis and Saul, 2000], Laplacian Eigenmap [Belkin and Niyogi, 2003], et Diffusion Maps [Nadler et al., 2005] [Coifman and Lafon, 2006]. Généralement ces algorithmes se composent de 3 étapes :

1). Construire le graphe G à partir de l'ensemble de données X et déterminer le voisinage pour chaque point.
2). Estimer les propriétés locales par le voisinage défini à l'étape 1.
3). Trouver la variété optimale qui préserve les propriétés locales à l'étape 2.

4.3.2 Isomap

Isomap[Tenenbaum et al., 2000] est une généralisation non-linéaire de l'algorithme MDS. La MDS classique a donné de bons résultats dans de nombreuses applications, mais elle souffre du fait qu'elle vise essentiellement à conserver des distances euclidiennes, et ne tient pas compte de la répartition des points de données voisins. Si les données de grande dimension se trouvent sur ou près d'une variété courbée, comme dans l'ensemble de données *Swiss roll*, la MDS classique peut considérer que deux points sont proches, alors qu'ils sont éloignés sur la variété (voir la Figure 4.6).

Isomap est une technique qui résout ce problème en tentant de préserver les distances géodésiques (ou curvilignes) entre les points de données. Deux mesures de distance peuvent être utilisées sur une variété. Le premier cas est la **distance euclidienne**. Le deuxième cas est la **distance géodésique** qui considère que la distance entre deux points est mesurée tout au long de la surface de la variété en utilisant la métrique riemannienne [13]. la Figure 4.6 illustre la différence entre ces deux mesures. Dans l'Isomap, les distances géodésiques entre les points de données $x_i (i = 1, 2, ..., n)$ sont calculées en

13. Une métrique riemannienne est la donnée, en chaque plan tangent a la surface, d'une forme bilinéaire symétrique définie positive qui soit différentiable.

FIGURE 4.6 – Distance euclidienne (gauche) et géodésique (droite) entre deux points appartenant à la variété appelée « Le bras de Venus » (swiss roll).

construisant un graphe G de voisinage, dans lequel chaque point de données x_i est lié à ses k-plus proches voisins $x_{ij}(j = 1, 2, ..., k)$ dans l'ensemble de données X. Le plus court chemin entre deux points dans le graphe forme une estimation de la distance géodésique entre ces deux points, et peut facilement être calculé à l'aide de l'algorithme du plus court chemin de Dijkstra ou celui de Floyd. Les distances géodésiques entre les points de données dans X sont calculées, de manière à former une matrice de distances géodésiques. Les représentations de faible dimensionnalité y_i des points de données x_i dans l'espace de faible dimension Y sont calculées en appliquant la mise à l'échelle classique sur la matrice de distances géodésiques résultante.

L'algorithme Isomap essaie de découvrir cette géométrie en trois étapes :

1. Recherche les k-plus proches voisins pour les points de l'ensemble $X = (x_1, ..., x_N) \subset \mathcal{R}^D$. D'après les distances euclidiennes $d(i, j)$ on détermine un « voisinage » pour chaque point x_i , soit avec le critère des k-plus proches voisins soit en considérant tous les points à l'intérieur d'une sphère de rayon centré sur x_i . On considère que les distances euclidiennes approchent les géodésiques quand les vecteurs se trouvent à petite distance.
2. On fait une estimation des distances géodésiques $d_G(i, j)$ entre tous les points x_i. Isomap construit un graphe dont les sommets sont les points et les arêtes les distances entre eux. Un sommet est adjacent à un autre seulement s'ils ont été définis comme voisins (étape 1). La distance géodésique est estimée entre chaque paire de données par la distance la plus courte parcourue sur le graphe (algorithme de *Floyd* ou *Djikstra*).

$$d_G(i, j) = min\{d_G(i, j), d_G(i, k) + d_G(i, k)\}$$

3. Finalement, la méthode MDS est appliquée à la matrice de distances $(d_G(i, j)^N_{i,j=1})$ pour obtenir un nouveau système de coordonnées euclidiennes $X^* \subset \mathcal{R}^d, (d < D)$ qui préserve la géométrie intrinsèque de la variété.

Quand le nombre de points grandit, les distances mesurées sur le graphe donnent de meilleures approximations aux géodésiques. Par contre, la complexité des algorithmes de parcours de graphes (étape 2 de l'algorithme) devient une contrainte à considérer :

- $O(n^3)$ pour l'algorithme de *Floyd*. n est le nombre de données de départ.
- $O(kn^2 \log n)$ pour l'algorithme de *Dijkstra*. k est le nombre des plus proches voisins utilisés pour définir le graphe.

4.3.3 LLE

Local Linear Embedding (LLE)[Roweis and Saul, 2000] est une technique qui est similaire à l'Isomap par la construction d'un graphe représentant les données. Contrairement à l'Isomap, il tente de préserver uniquement les propriétés locales des données. En conséquence, le LLE est moins sensible à un court-circuit que l'Isomap, parce que seul un petit nombre de propriétés locales est touché si un court-circuit a lieu. En outre, la préservation des propriétés locales permet une intégration réussie des variétés non-convexes.

Dans le LLE, les propriétés locales des données de la variété sont construites en écrivant les points de données de grande dimension comme une combinaison linéaire de leurs voisins les plus proches. Dans la représentation de faible dimensionnalité, le LLE tente de conserver le poids de reconstruction dans les combinaisons linéaires aussi bien que possible.

LLE décrit les propriétés locales de la variété autour d'un point de donnée x_i en écrivant le point comme une combinaison linéaire W_{ij} (le poids de reconstruction susmentionné) de ses k-plus proches voisins x_j. Par conséquent, LLE correspond à un hyperplan à travers le point de donnée x_i et ses plus proches voisins, ce qui suppose que la variété est localement linéaire. L'hypothèse de linéarité locale implique que le poids de reconstruction W_{ij} du point x_i est invariant par translation, rotation et changement d'échelle. En raison de l'invariance de ces transformations, une application linéaire de l'hyperplan à un espace de dimension inférieure préserve le poids de reconstruction en l'espace de plus faible dimensionnalité. En d'autres termes, si la représentation de faible dimensionnalité des données conserve la géométrie locale de la variété, le poids de reconstruction W_{ij} qui reconstruit les points x_i à partir de leurs voisins dans la représentation des données de grande dimension, reconstruit également les points y_i à partir de leurs voisins dans la représentation de faible dimensionnalité. En conséquence, trouver la représentation des données Y de faible dimension d équivaut à minimiser la fonction de coût.

L'algorithme LLE est résumé en 3 étapes (Figure 4.7) :

1. Détermination des voisins pour chaque point x_i .
2. Calcul des poids W_{ij} qui déterminent le mieux chaque x_i à partir de ses voisins. Les vecteurs x_i sont alors décrits comme une combinaison linéaire des voisins. L'information de la géométrie intrinsèque locale de la variété est codée dans les poidss W_{ij} . On calcule les poids W qui minimisent les erreurs :
$$\epsilon(W) = \sum_i \parallel x_i - \sum_{j \in K(i)} W_{ij} x_j \parallel^2$$
3. Calcul des projections de faible dimension y_i en respectant les mêmes relations de voisinage W_{ij}. Dans cette étape, on cherche l'ensemble $Y = \{y_1, y_2, ..., y_n\}$ tel que $y_i \approx \sum_{j \in K(i)} W_{ij} y_j$, $Y \subset \mathcal{R}^d, d << D$. Les vecteurs de faible dimension y_i représentent les coordonnées sur la variété et celles-ci sont trouvées par la minimisation de la fonction de coût :
$$\Phi(Y) = \sum_i \parallel y_i - \sum_{j \in K(i)} W_{ij} y_j \parallel^2$$

.

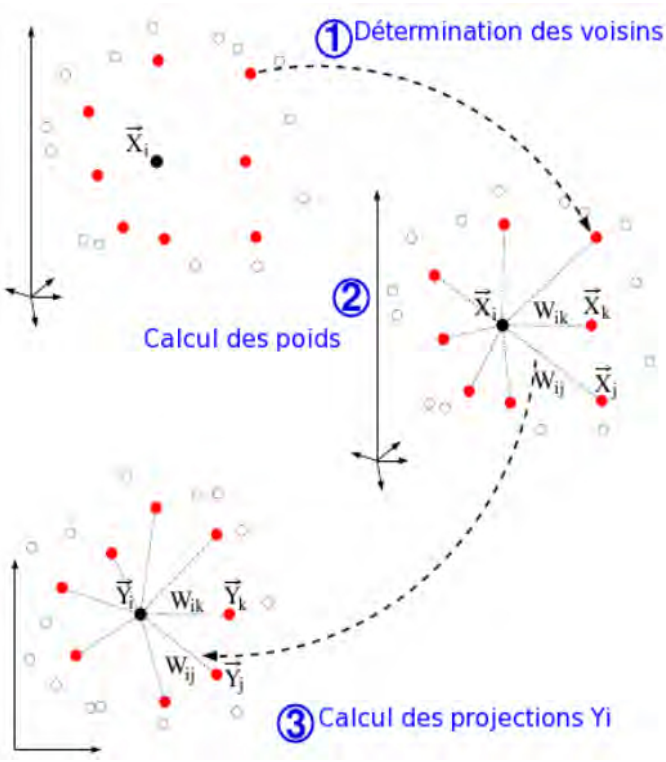


FIGURE 4.7 – Étapes de LLE [Roweis and Saul, 2000].

4.3.4 Laplacian Eigenmaps

Similaire à la technique LLE, l'algorithme Laplacian Eigenmap trouve une représentation de faible dimensionnalité des données en préservant les propriétés locales de la variété[Belkin and Niyogi, 2001, Belkin and Niyogi, 2003]. Cependant il possède un fondement théorique différent : l'information de voisinage est récupérée à l'aide d'un graphe mais les coordonnées de faible dimension sont obtenues à partir de la notion du laplacien du graphe. Dans le Laplacian Eigenmap, les propriétés locales sont basées sur les distances entre les voisins proches. Il calcule une représentation de basse dimensionnalité des données dans laquelle les distances entre un point de données et ses k-plus proches voisins sont réduites au minimum. Cela se fait d'une manière pondérée, à savoir, la distance de la représentation de basse dimensionnalité des données entre un point de données et son voisin le plus proche contribue davantage à la fonction de coût que la distance entre le point de données et son deuxième plus proche voisin. En utilisant la théorie spectrale des graphes, la minimisation de la fonction de coût est définie comme un problème de valeurs propres.

4.3.4.1 Laplacien du graphe

- **Version non normalisée :**
  Soit le laplacien non normalisé, connu aussi sous le nom du laplacien combinatoire.

Il est défini par :

$$L_{un} = L_{un}(v_i, v_j) = \begin{cases} d(v_i), \text{ si } v_i = v_j \\ -\omega_{ij}, \text{ si } v_i \sim v_j \\ 0, \text{ sinon .} \end{cases}$$

Cette matrice peut aussi s'écrire $L_{un} = D - W$. Cette matrice est symétrique et semi-définie positive [14], et possède des valeurs propres positives ou nulles.

- **Version normalisée :**
  On trouve dans la littérature deux versions normalisées du laplacien. La version symétrique est :

$$L_{norm} = L_{norm}(v_i, v_j) = \begin{cases} 1 - \frac{\omega_{ij}}{d(v_i)}, \text{ si } v_i = v_j \\ -\frac{\omega_{ij}}{\sqrt{d(v_i)d(v_j)}}, \text{ si } v_i \sim v_j \\ 0, \text{ sinon .} \end{cases}$$

  La matrice L_{norm} peut s'écrire aussi :

$$L_{norm} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

  La version asymétrique peut s'écrire comme :

$$L_{norm} = I - D^{-1}W = I - L_{rw}$$

  L_{rw} est la marche aléatoire *(random walk)*, ce qui montre clairement le lien entre le laplacien et les marches aléatoires.

$$L_{rw} = D^{-1}W = D^{-\frac{1}{2}}(I - L_{norm})D^{\frac{1}{2}}$$

L_{norm} a 3 proprétés importants :

- λ est une valeur propre de L_{norm} avec le vecteur v si et seulement si λ et v sont la solution de $Lv = \lambda Dv$.
- L_{norm} est la matrice semi-définie positive avec la première valeur propre $\lambda_1 = 0$ et le correspondant vecteur propre qui est un vecteur de constant 1.
- les valeurs propres suivent : $\lambda_1 \leqslant \lambda_2 \leqslant ... \leqslant \lambda_n$.

4.3.4.2 L'algorithme du Laplacian Eigenmap

L'algorithme du Laplacian Eigenmap construit d'abord un graphe de voisinage G dans lequel chaque point de données x_i est relié à ses k-plus proches voisins. Pour tous les x_i et x_j dans le graphe G qui sont reliés par une arête, le poids de l'arête est calculé en utilisant la fonction du noyau gaussien, conduisant à une matrice d'adjacence creuse

14. Une matrice A de $n \times n$ est semi-définie positive si chacun des mineurs principaux de A est $\geqslant 0$. Par exemple pour une matrice 3×3, on a donc 3 mineurs principaux de premier ordre : a_{11}, a_{22} et a_{33}.

W. Dans le calcul des représentations de basse dimensionnalité y_i , la fonction de coût qui est réduite au minimum, est donnée par

$$\Phi(Y) = \sum_{ij} \parallel y_i - y_j \parallel^2 \omega_{ij}$$

.

L'algorithme Laplacian Eigenmaps peut se résumer de la façon suivante :

1. Détermination des voisins pour chaque point (avec les techniques définies pour Isomap et LLE). Dans la construction du graphe, les sommets sont les points et les arêtes sont non nulles seulement si x_i et x_j sont « proches ».
2. Pondération des arêtes. Il y a deux possibilités :
   a. **Noyau de chaleur avec le paramètre $\epsilon \in R$.** Si les sommets i et j sont reliés,
   $$W_{ij} = exp(-\frac{\parallel x_i - x_j \parallel^2}{\epsilon})$$
   b. **Noyau de chaleur avec le paramètre $\epsilon = \infty$.** Si les sommets i et j sont reliés,
   $$W_{ij} = 1$$
3. Calcul des eigenmaps. D'après le graphe $G(V, E; \omega)$ défini auparavant, on calcule les vecteurs et les valeurs propres du système :

$$Ly = \lambda Dy \qquad \text{(E 4.2)}$$

où D est la matrice de dégré dont les éléments sont la somme de chaque ligne de W (voir la section précédente). L est la matrice du laplacien $L = D - W$. L est une matrice symétrique semi-définie positive qui peut être considérée comme l'opérateur laplacien des fonctions définies dans le graphe G.
Si $y_0, ..., y_n$ sont les solutions de l'équation(E 4.2) ordonnées selon leurs valeurs propres,

$$Ly_0 = \lambda_0 Dy_0$$
$$Ly_1 = \lambda_1 Dy_1$$
$$...$$
$$Ly_n = \lambda_k Dy_n$$

$0 = \lambda_1 \leqslant \lambda_2 \leqslant ... \leqslant \lambda_n$ Le vecteur y_0 correspondant à la valeur propre 0 n'est pas pris en compte. Les d vecteurs propres suivants définissent un espace euclidien d-dimensionnel :

$$x_i \longrightarrow (y_{i1}, ...y_{id})$$

4.3.4.3 Cartes de diffusion(Diffusion Maps)

Une autre technique appartenant aux méthodes basées sur le laplacien du graphe est les cartes de diffusion (Diffusion Maps)[Nadler et al., 2005]. Le principe des cartes de diffusion est basé sur la définition d'une marche aléatoire de Markov (random walk) sur le graphe des données. En effectuant la marche aléatoire pour un certain nombre de pas, une mesure de la proximité des points de données est obtenue. En utilisant cette mesure,

la distance de diffusion est définie. Dans la représentation de basse dimensionnalité des données, les distances de diffusion sont conservées le mieux possible. L'idée clé sous-jacente à la distance de diffusion, c'est qu'elle est basée sur l'intégration sur tous les chemins sur le graphe. Cela rend la distance de diffusion plus robuste aux court-circuits que la distance géodésique, par exemple, qui est employée dans Isomap. Le Diffusion Maps a été appliqué avec succès à la mise en correspondance des formes et de nombreuses autres applications.

1) Marches aléatoires sur graphes

Nous allons nous intéresser à un processus de marche aléatoire (ou de diffusion dans le graphe G). Le temps est discrétisé $t = (0, 1, 2, ...)$. A chaque instant, un marcheur est localisé sur un sommet et se déplace à l'instant suivant vers un sommet choisi aléatoirement et uniformément parmi les sommets voisins. La suite des sommets visités est alors une marche aléatoire, et la probabilité de transition du sommet v_i au sommet v_j est définie à chaque étape par :

$$P_{ij} = P(v_i, v_j) = \frac{\omega_{ij}}{d(v_i)}$$

Ceci définit la matrice de transition P, de la chaîne de Markov correspondant à la marche aléatoire sur graphe. Cette technique est connue commme la construction du laplacien du graphe normalisé (voir section précédente). Donc la matrice de transition peut ainsi être définie par $P = D^{-1}W$.

Considérons $P^t(v_i, v_j)$, le noyau correspondant à la t-ème puissance de P, P^t , qui peut être interprété comme la probabilité pour un marcheur d'atteindre le sommet v_j en partant du sommet v_i en passant par t étapes. L'intérêt d'introduire cette matrice de transition est que l'exploration du graphe, par la marche aléatoire qu'elle engendre, permet de déterminer des propriétés topologiques du graphe reliées aux propriétés spectrales de P.

P est généralement symétrique et pour chaque colonne la somme des éléments est égale à 1. Cette matrice est intéressante car elle reflète la géométrie intrinsèque des données. Une marche aléatoire correspond à une chaîne de Markov homogène puisque les probabilités de transition restent les mêmes à chaque fois que l'on revient sur un nœud du graphe. Les chaînes de Markov sont définies en termes d'états et de transitions entre ces derniers. Les états sont, dans notre cas, les nœuds du graphe. Dans une chaîne de Markov, deux états i et j sont dits communicants si l'on peut atteindre l'un à partir de l'autre avec une probabilité finie, ce qui signifie que le graphe est connexe. Si l'on veut décrire la probabilité de transition $P^t(v_i, v_j)$ d'un nœud v_i à un nœud v_j en t étapes, il suffit de considérer des voisinages plus larges, ce qui correspond à élever la matrice P à la puissance t.

2) Distance de diffusion

D'après la théorie de la marche aléatoire, la distance de diffusion entre les deux sommets v_i et v_j à l'étape t dépend de la similarité des distributions de probabilité initialisées par v_i et v_j :

$$D_t^2(v_i, v_j) \triangleq \quad \| \, p(z, t|v_i) - p(z, t|v_j) \, \|_\phi^2 = \sum_{z \in V} (p(z, t|v_i) - p(z, t|v_j))^2 \phi(z).$$

Ici $p(z,t|v_i)$ est la probabilité de la marche aléatoire du point v_i jusqu'à z après t étapes, et ϕ est une fonction de poids.

Cette distance de diffusion peut être calculée en utilisant les vecteurs propres ψ_n et les valeurs propres λ_n de la matrice P^t[Nadler et al., 2005] :

$$D_t^2(v_i, v_j) = \sum_{n \geqslant 1} \lambda_n^t (\psi_n^t(v_i) - \psi_n^t(v_j))^2$$

où $1 = \lambda_0 > |\lambda_1| \geqslant |\lambda_2| \geqslant ... \geqslant |\lambda_n|$.

De cette manière, nous pouvons recouvrir la géométrie de la variété indépendamment de la densité des données. L'ensemble de données $X = \{x_1, ..., x_N\}$ avec dimensiion D, peut être représenté par $Y = \{y_1, y_2, ..., y_N\}$:

$$y_i = \begin{pmatrix} \lambda_1^t \psi_1^t(x_i) \\ \lambda_2^t \psi_2^t(x_i) \\ . \\ . \\ . \\ \lambda_d^t \psi_d^t(x_i) \end{pmatrix}, \qquad d << D$$

4.3.5 Discussion

Dans les sections précédentes, nous avons présenté les différentes techniques de réduction de la dimensionnalité. L'ACP et le MDS classique sont les méthodes linéaires et ne permettent pas de révéler la géométrie réelle de données lorsque la structure de l'ensemble de données est fortement non-linéaire. Par contre, l'ACP est une méthode très rapide. Le MDS génère le même résultat que ACP, mais c'est une méthode relativement lente.

Isomap et LLE sont fondées sur la notion du graphe. Elles sont différentes quant à la manière d'estimer les propriétés du voisinage. Isomap calcule la distance géodésique pour préserver les propriétés du point de vue global, contrairement à LLE qui préserve seulement les propriétés géométriques locales. La préservation de la distance géodésique entre deux points par Isomap provoque souvent une distorsion au niveau du voisinage local, comme la distorsion de la distance entre les deux points A et B (Figure 4.6). Pour LLE, si les 2 points ne sont pas voisins, la distance entre eux peut être sous-estimée. Par conséquent, ces 2 points qui sont loin de l'un à l'autre dans l'espace d'origine X, peuvent être proches dans la structure de la variété Y. Concernant le temps de calcul, Isomap dépend du parcours de tous les points pour calculer les chemins et est par conséquent très lent.

Le Laplacian Eigenmaps est similaire à LLE, mais il permet de révéler la structure géométrique de données en utilisant le laplacien du graphe. Le Laplacian Eigenmaps et LLE peuvent être appliqués pour une base de données de grande échelle, par contre LLE est sensible au bruit. Le Laplacian Eigenmaps et le Diffusion Maps sont relativement plus sensibles à la valeur du paramètre comme le ϵ dans le noyau de chaleur.

Voici la table 4.1 qui résume la différence entre ces techniques présentées.

	MDS	ACP	Isomap	LLE	Laplacian Eigenmaps	Diffusion maps
Vitesse	lent	très rapide	très lent	rapide	rapide	rapide
Géométrie	non	non	oui	oui	oui	peut-être
Bruits	oui	oui	peut-être	non	oui	oui
Paramètres	non	non	oui	oui	oui	oui

TABLE 4.1 – Les comparaisons des techniques (source : [Wittman, 2005])

4.4 Expérimentation

La table 4.1 présentée ci-dessus montre l'avantage d'utiliser les techniques comme le Laplacian Eigenmaps et le Diffusion Maps, ce qui permet de non seulement révéler la géométrie réelle de données bruitées, mais également générer la variété rapidement par la décomposition spectrale de la matrice définie. Dans cette section, nous présentons nos études sur l'analyse des images des yeux par ces méthodes basées sur le graphe. La réalisation des méthodes telles que Laplacian Eigenmaps et Diffusion Maps est détaillée dans l'Annexe 8.

4.4.1 Variété de l'ensemble d'images

Nous appliquons les techniques de réduction de la dimensionnalité sur un ensemble $\mathcal{I}$ des images des yeux capturées par la webcam du système. Le nombre d'images n dans cet ensemble dépend du temps d'acquisition de la webcam dont la fréquence est de 30 Hz pour une résolution de 960×540 pixels. La méthode de localisation des yeux a été présentée dans le chapitre précédent. Dans notre expérimentation, nous capturons l'œil gauche avec une taille de l'image fixée à 80×40 pixels, soit une dimension de 3200. L'ensemble des images $\mathcal{I}$ peut être représenté par une matrice de haute dimension $n \times 3200$. Dans la section 3.3.1, nous avons présenté l'efficacité de l'histogramme CS-LBP en tant que descripteur de l'image. Nous divisons donc l'image de l'œil capturée en bloc et combinons les histogrammes de chaque bloc pour former un descripteur de l'image. Par exemple, si l'image est divisée en 40 blocs (10 en largeur, 4 en hauteur), la dimension de l'histogramme CS-LBP sera $16 \times 10 \times 4 = 640$ pixels. Par conséquent nous obtenons un nouvel ensemble $\mathcal{I}'$ dont la dimension est $n \times 640$. Comparée à la dimension initiale, la dimension de $\mathcal{I}'$ est réduite localement pour le calcul. Ensuite nous utilisons le Laplacian Eigenmaps sur $\mathcal{I}'$ pour apprendre la variété de données en 3D, notée $\mathcal{I}^*$. La dimension de $\mathcal{I}^*$ est $n \times 3$.

La figure 4.8 illustre les représentations des ensembles des échantillons $\mathcal{I}$ et $\mathcal{I}^*$. La figure 4.8 a) montre un ensemble de 120 échantillons des images des yeux lorsque le sujet suit le stimulus visuel (points verts) présenté aux quatre coins de l'écran pendant 4 secondes. Chaque image de cet ensemble $\mathcal{I}$ est d'abord divisée en blocs et nous obtenons

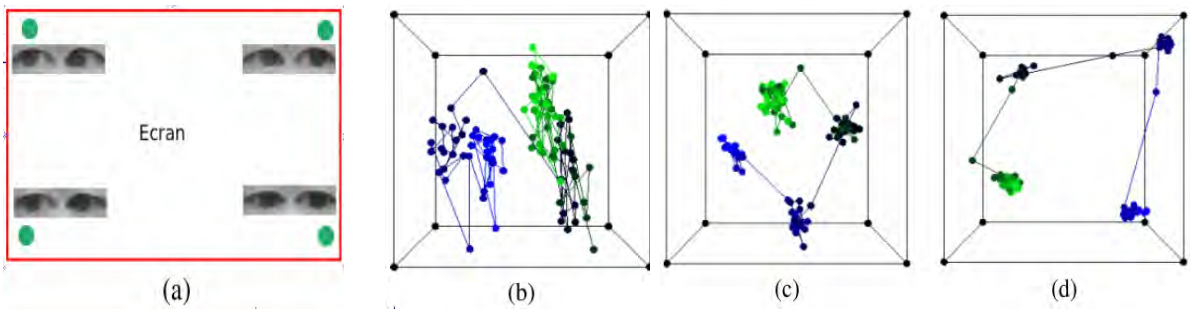


FIGURE 4.8 – a) les échantillons des images de regard vers les 4 coins (120 images). b) la variété des échantillons sans bloc. c) la variété des échantillons divisés en 4 blocs. d) la variété des échantillons divisés en 16 blocs.

donc un nouvel ensemble $\mathcal{I}'$ dont la dimension dépend du nombre des blocs divisés dans l'image. Ensuite nous appliquons le Laplacian Eigenmaps sur $\mathcal{I}'$ pour obtenir sa variété tri-dimensionnelle. La figure 4.8 b), c), d) montre trois variétés différentes de l'ensemble $\mathcal{I}'$ dans lequel chaque image est divisée en blocs de différentes façons (sans blocs, en 4 blocs et en 16 blocs). Les représentations montrent l'avantage de diviser les images des yeux en blocs, ce qui permet de distinguer la variation des mouvements oculaires. Dans les résultats des expérimentations qui suivent, l'image des yeux est divisée en 40 blocs, sauf indication contraire.

La première expérimentation porte sur l'analyse des variétés chez différents sujets dont les yeux sont de formes différentes (Figure 4.9). 5 sujets différents sont placés devant l'écran à une distance variable qui correspond généralement à la longeur du bras. On leur demande de fixer les 16 points sur l'écran pendant que la webcam est en train de capturer les images. Les apparences de l'œil entre les sujets sont différentes car les expérimentations sont réalisées dans des conditions d'illuminations différentes.

Pendant l'acquisition, nous pouvons obtenir un ensemble qui contient 480 images des yeux pour chaque sujet et ces images des yeux correspondent aux regards du sujet vers les 16 points. Nous utilisons le Laplacian Eigenmaps sur cet ensemble des images de chaque sujet pour obtenir sa variété en 3D comme on peut le voir dans les colonnes (b, c) de la Figure 4.9. De la même façon que ce que nous avons présenté dans la section 4.3.4.2, les coordonnées 3D de chaque point de la variété sont les trois premiers vecteurs propres différents de zéro (y_1, y_2, y_3). Chaque point de la variété représente une image des yeux dans l'ensemble. Les colonnes (d, e) de la Figure 4.9 montrent les variétés d'un nouvel ensemble qui contient 120 images des yeux vers les quatre coins de l'écran. Nous pouvons distinguer quatre classes de points qui représentent les quatre groupes d'images des yeux vers ces quatre positions. Ces résultats montrent que, malgré la différence de la forme et d'illumination entre chaque sujet, le Laplacian Eigenmaps permet de projeter les images de haute dimension dans un espace de faible dimension en gardant la variation des mouvements oculaires. De plus, la structure topologique de la variété des images des yeux peut rendre compte des directions du regard. La Figure 4.10 illustre les 24 stimuli qu'on demande au sujet de suivre sur l'écran. Nous appliquons le Laplacian Eigenmaps sur l'ensemble d'images obtenu et les variétes générées avec différents ϵ sont montrées dans b) et c). Nous pouvons observer que la variété avec $\epsilon = 600$ possède une certaine similitude avec le plan des stimuli sur l'écran.

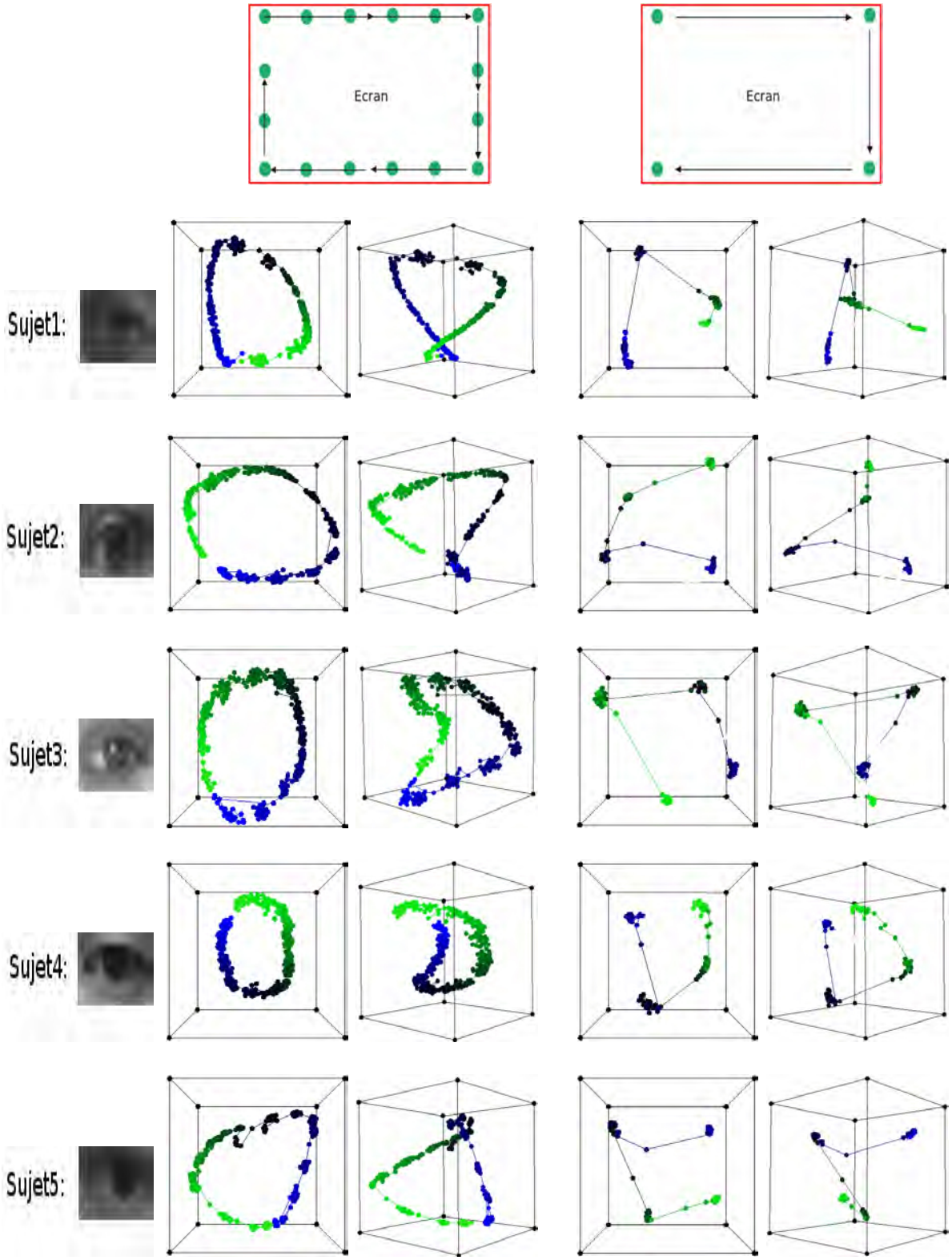
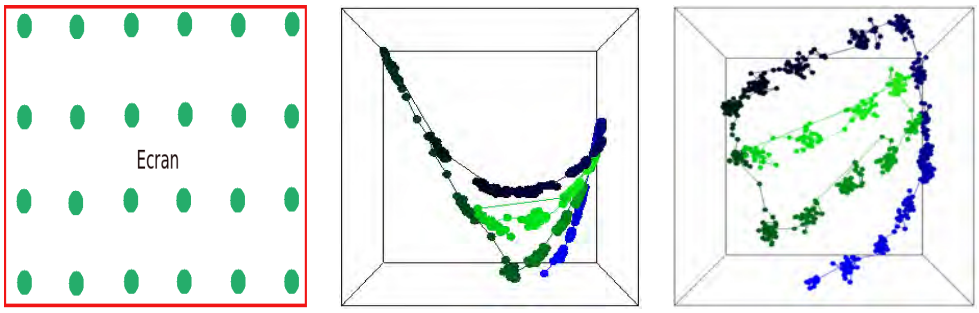


FIGURE 4.9 – La variété des yeux obtenue par Laplacien Eigenmaps ($\epsilon = 100$) de 5 sujets différents vers les 16 points et 4 coins à l'écran. a) l'exemple d'une image de l'œil pour chaque sujet. b)c) montrent 2 vues différentes respectivement de la variété 3D sur l'ensemble de 480 images des yeux vers les 16 points. d)e) montrent 2 vues différentes respectivement de la variété sur l'ensemble de 120 images vers les 4 coins sur l'écran.



(a) (b) (c)

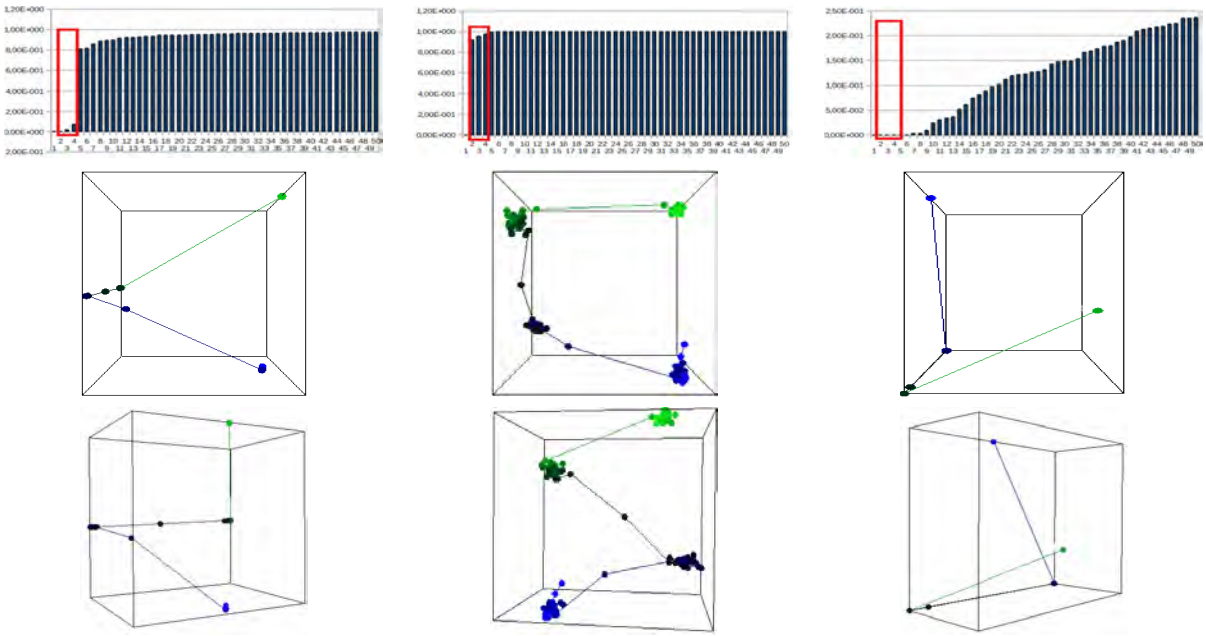
FIGURE 4.10 – a) Les positions de 24 points sur l'écran. b)c) montrent 2 variétés différentes obtenues par le Laplacian Eigenmaps, avec différents ϵ, sur les exemples d'images des yeux sur les 24 points. Le nombre des images capturées $n = 740$. (b) $\epsilon = 30$; (c) $\epsilon = 600$.

4.4.2 Comparaison des différentes techniques

1) Nous analysons le choix du paramètre ϵ pour le Laplacian Eigenmaps et le Diffusion Maps. Dans notre expérimentation, les deux techniques utilisent le noyau de chaleur pour définir la matrice de similarité W. Le choix de ϵ influence directement la variété générée. Pour le Laplacian Eigenmaps, les 3 premiers vecteurs propres non-zéros (y_1, y_2, y_3) définissent l'espace tri-dimensionnel où $0 = \lambda_0 \leqslant \lambda_1 \leqslant \lambda_2 \leqslant \lambda_3$. Pour le Diffusion Maps, les points de l'espace tri-dimensionnel sont définis par $(\lambda_1\psi_1, \lambda_2\psi_2, \lambda_3\psi_3)$, où $1 = \lambda_0 \geqslant |\lambda_1| \geqslant |\lambda_2| \geqslant |\lambda_3|$, λ_n est la valeur propre et ψ_n est le vecteur propre correspondant. Le choix de ϵ dépend de la dimension de chaque donnée et du nombre des données. Nous prenons un ensemble $\mathcal{I}$ de 110 images de l'œil vers les 4 coins de l'écran comme dans la Figure 4.9. Chaque image est divisée en 40 blocs et nous combinons les histogrammes CS-LBP de chaque bloc pour former un descripteur de l'image dont la dimension est $16 \times 40 = 640$. Un nouvel ensemble $\mathcal{I}'$ est donc obtenu et sa dimension est bien réduite comparativement à la dimension de l'ensemble $\mathcal{I}$. Les Figures 4.11 et 4.12 illustrent les variétés obtenues par le Laplacian Eigenmaps et le Diffusion Maps en utilisant différents ϵ sur l'ensemble $\mathcal{I}'$. De façon empirique, nous choisissons la valeur ϵ qui permet d'obtenir les 3 valeurs propres λ (rectangle rouge) représentant le plus grand changement, comme celles dans la Figure 4.11 a) pour le Laplacian Eigenmaps et la Figure 4.12 a) pour le Diffusion Maps. Dans notre expérimentation sur l'ensemble $\mathcal{I}'$, l'application du Laplacian Eigenmaps avec $\epsilon = 30$ permet de générer la variété plus fidèle aux données initiales que celles avec d'autres ϵ. Sur le même ensemble $\mathcal{I}'$, l'application du Diffusion Maps avec $\epsilon = 35$ donnent le meilleur résultat, comme le montre la Figure 4.12 a).

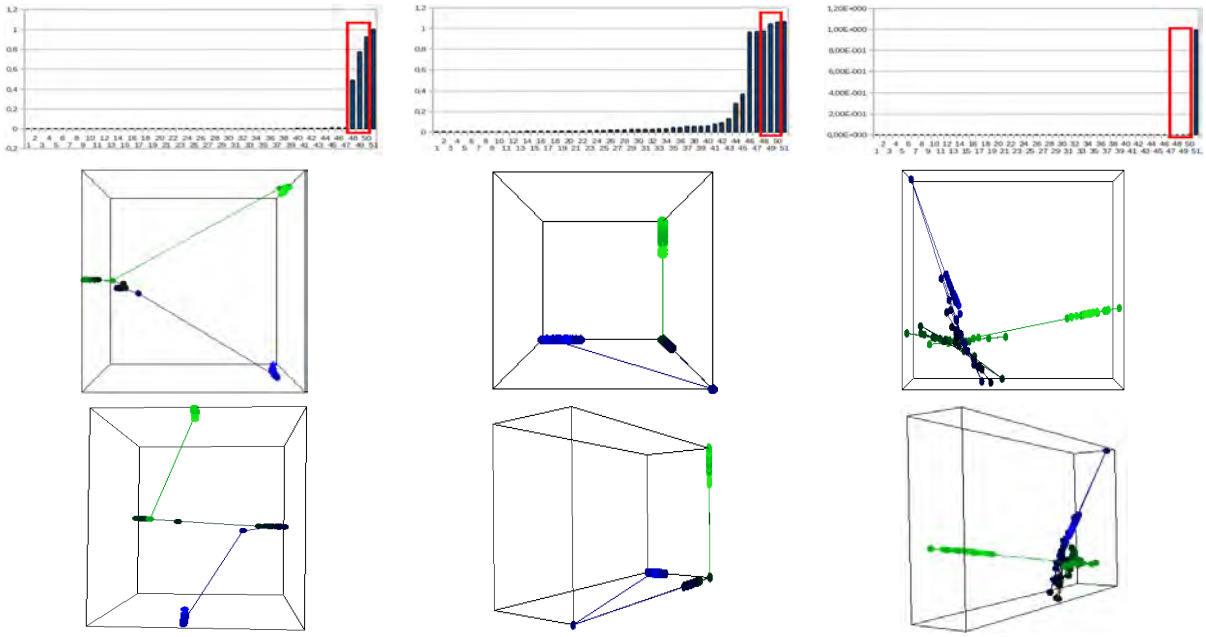
2) Nous étudions les trois techniques différentes : l'ACP, le Laplacian Eigenmaps et le Diffusion Maps et nous analysons les variétés générées par ces techniques sur les différents ensembles suivants :

- **L'ensemble $\mathcal{I}$ des images de l'œil** : Cet ensemble contient 110 images de l'œil et chaque image est en niveau de gris. La dimension de l'image est 3200 (80×40 pixels).



(a) (b) (c)

FIGURE 4.11 – Les variétés de 110 images des yeux par le Laplacian Eigenmaps avec différents ϵ : a)$\epsilon = 30$, b)$\epsilon = 180$, c)$\epsilon = 10$. La première ligne représente la distribution des valeurs propres λ_n, la 2ème et 3ème ligne représentent les différentes vues sur la variété 3D correspondante.



(a) (b) (c)

FIGURE 4.12 – Les variétés de 110 images des yeux par le Diffusion Maps avec différents ϵ : a)$\epsilon = 35$, b)$\epsilon = 15$, c)$\epsilon = 200$. La première ligne représente la distribution des valeurs propres λ_n, la 2ème et 3ème ligne représentent les différentes vues sur la variété 3D correspondante.

- **L'ensemble** $\mathcal{I}_{CS-LBP}$: Chaque image de l'ensemble $\mathcal{I}$ est divisée en 40 blocs et un descripteur est formé par la combinaison des histogrammes CS-LBP de chaque bloc. La dimension du descripteur est donc $16 \times 40 = 640$. Cet ensemble $\mathcal{I}_{CS-LBP}$ contient 110 descripteurs qui correspondent aux images de l'ensemble $\mathcal{I}$.

- **L'ensemble** $\mathcal{I}_{in}$: Nous divisons l'image en $N(N = 40)$ blocs, puis calculons la somme des intensités des pixels pour chaque bloc. On note S_j la somme des intensités dans le $j$ème bloc, le descripteur X de dimension N est représenté par :

$$X = \frac{(S_1, S_2, ..., S_j)}{\sum S_j} \qquad j \in N(N = 40)$$

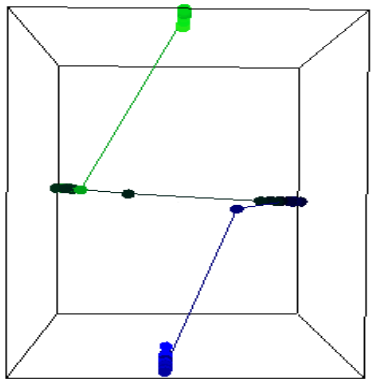
  L'ensemble $\mathcal{I}_{in}$ contient 110 descripteurs de ce type qui correspondent aux images de l'ensemble $\mathcal{I}$.

Nous appliquons les trois techniques (le Laplacian Eigenmaps, le Diffusion Maps et l'ACP) respectivement sur les trois ensembles présentés ci-dessus : $\mathcal{I}_{CS-LBP}$, $\mathcal{I}$ et $\mathcal{I}_{int}$. La Figure 4.13 montre les variétés générées pour ces ensembles. Pour comparer, nous faisons la même application sur 3 nouveaux ensembles : $\mathcal{I}^+_{CS-LBP}$, $\mathcal{I}^+$ et $\mathcal{I}^+_{in}$. L'ensemble $\mathcal{I}^+$ contient 86 images bruitées. Les bruits proviennent des perturbations sur la localisation de l'œil à cause de la condition expérimentale ou des mouvements du sujet. Les erreurs de localisation, soit la différence entre la position de l'œil localisé et la position réelle de l'œil, sont environ ± 1 pixel. Les variétés de ces 3 ensembles sont représentées dans la Figure 4.14. Nous pouvons observer que :

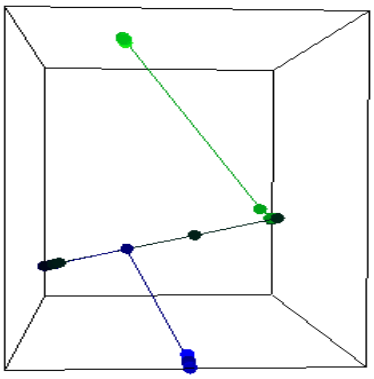
- Le descripteur CS-LBP permet de mieux représenter l'image d'origine que le descripteur d'intensités des pixels en préservant les caractéristiques locales de l'image.

- L'utilisation du descripteur CS-LBP permet de réduire la dimension de l'image, ce qui est avantageux pour le calcul.

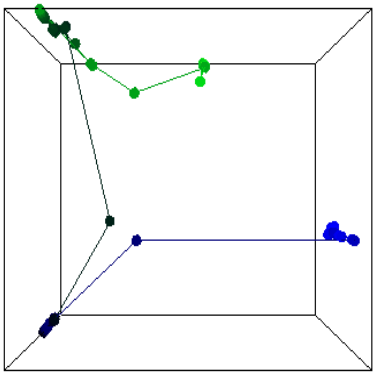
- La structure topologique de la variété générée par le Laplacian Eigenmaps sur un ensemble d'images (nettes ou bruitées) de l'œil permet de révéler la différence entre les mouvements oculaires.



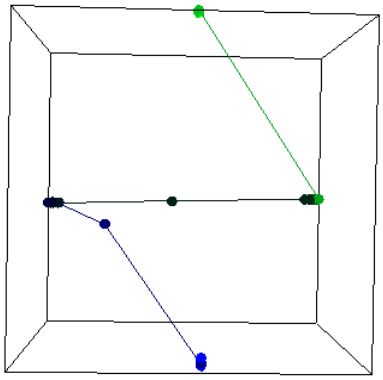
Diffusion Maps
($\epsilon = 35$) sur $\mathcal{I}_{CS-LBP}$
(a)



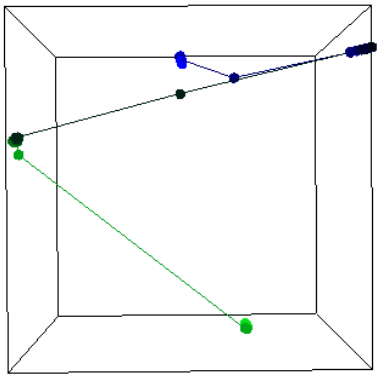
Diffusion Maps
($\epsilon = 250$) sur $\mathcal{I}$
(b)



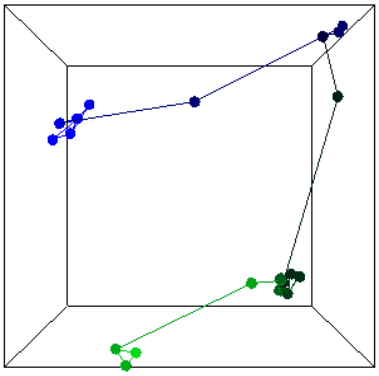
Diffusion Maps
($\epsilon = 1$) sur $\mathcal{I}_{int}$
(c)



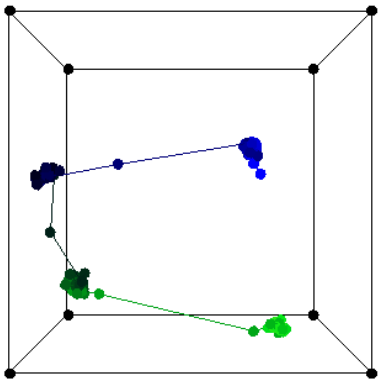
Laplacian Eigenmaps
($\epsilon = 30$) sur $\mathcal{I}_{CS-LBP}$
(d)



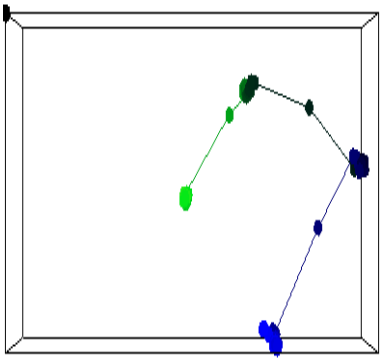
Laplacian Eigenmaps
($\epsilon = 250$) sur $\mathcal{I}$
(e)



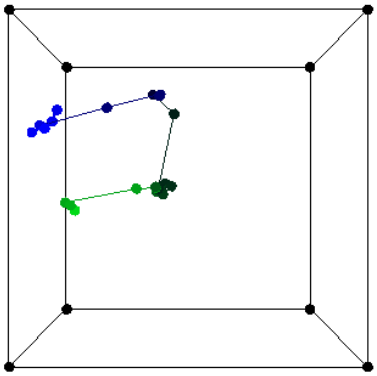
Laplacian Eigenmaps
($\epsilon = 2$) sur $\mathcal{I}_{in}$
(f)



ACP sur
$\mathcal{I}_{CS-LBP}$
(g)

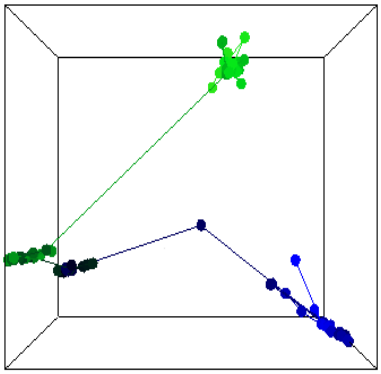


ACP sur
$\mathcal{I}$
(h)

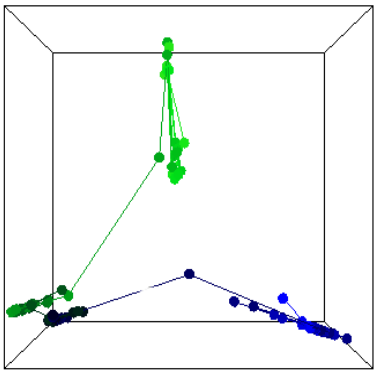


ACP sur
$\mathcal{I}_{in}$
(i)

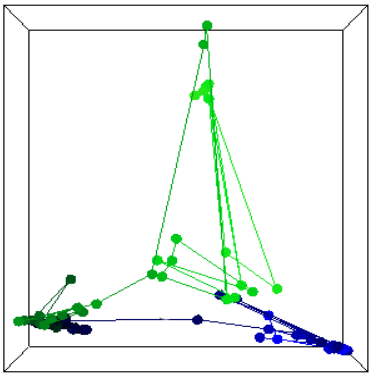
FIGURE 4.13 – Les variétés générées par les 3 techniques (le Laplacian Eigenmaps, le Diffusion Maps et l'ACP) sur les 3 ensembles : $\mathcal{I}$, $\mathcal{I}_{CS-LBP}$ et $\mathcal{I}_{in}$. Les descripteurs de chaque ensemble représentent les images de l'œil vers les 4 coins de l'écran.



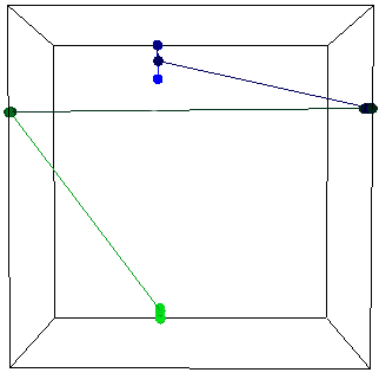
Diffusion Maps
($\epsilon = 45$) sur $\mathcal{I}^+_{CS-LBP}$
(a)



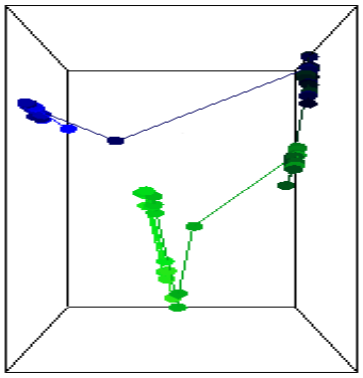
Diffusion Maps
($\epsilon = 450$) sur $\mathcal{I}^+$
(b)



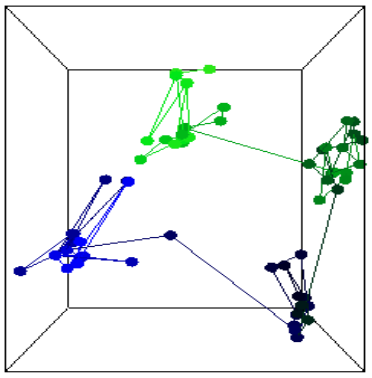
Diffusion Maps
($\epsilon = 1.5$) sur $\mathcal{I}^+_{in}$
(c)



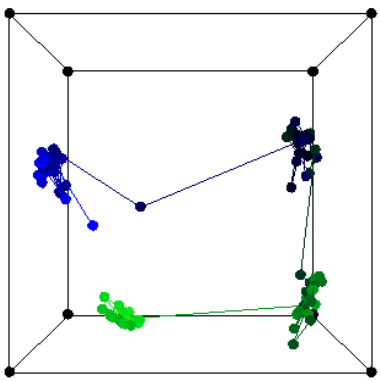
Laplacian Eigenmaps
($\epsilon = 28$) sur $\mathcal{I}^+_{CS-LBP}$
(d)



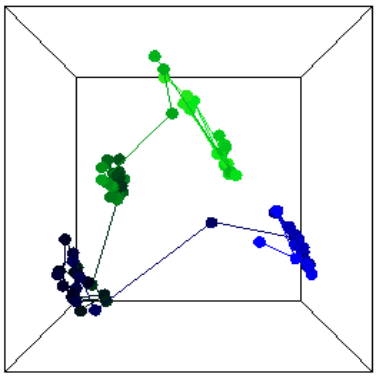
Laplacian Eigenmaps
($\epsilon = 600$) sur $\mathcal{I}^+$
(e)



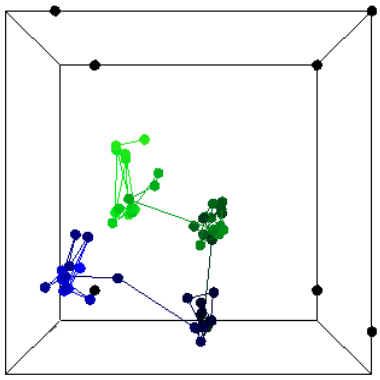
Laplacian Eigenmaps
($\epsilon = 5$) sur $\mathcal{I}^+_{in}$
(f)



ACP sur
$\mathcal{I}^+_{CS-LBP}$
(g)



ACP sur
$\mathcal{I}^+$
(h)



ACP sur
$\mathcal{I}^+_{in}$
(i)

FIGURE 4.14 – Les variétés générées par les 3 techniques (le Laplacian Eigenmaps, le Diffusion Maps et l'ACP) sur les 3 ensembles : $\mathcal{I}^+$, $\mathcal{I}^+_{CS-LBP}$ et $\mathcal{I}^+_{in}$. L'ensemble des images de l'œil $\mathcal{I}^+$ se compose de 86 images des yeux bruitées des regards vers les 4 coins de l'écran.

4.5 Conclusion

Dans ce chapitre nous avons présenté les différentes techniques linéaires et non-linéaires de réduction de la dimensionnalité. Ces techniques visent à trouver la structure intrinsèque de dimension réduite pour un ensemble de données de haute dimension, par exemple, un ensemble des images des yeux qui représentent les différents mouvements oculaires. L'objectif de l'utilisation de ces techniques sur nos images des yeux est d'analyser au point de vue global la variation des mouvements oculaires par la variété de faible dimension d ($d << D$) sur cet ensemble d'images. L'étude des variétés permet de découvrir des invariants et l'information géométrique associée à la distribution des données dans l'espace original.

Dans un premier temps chaque individu de l'ensemble des images de l'œil est traité par la méthode d'extraction des caractéristiques présentée dans le chapitre 3. Ce processus est considéré comme une réduction de la dimensionnalité de l'image du point de vue local. Ensuite nous utilisons le Laplacian Eigenmaps pour apprendre au point de vue global la variété en 3D sur cet ensemble de dimension réduite. La représentation de la variété en 3D de ces images peut fournir des informations utiles sur la nature et l'organisation des données et être exploitée pour les tâches de classification ou de regroupement des mouvements oculaires dans le module d'estimation du regard qui sera présenté dans le chapitre suivant.

Cinquième partie

Estimation du regard

Sommaire

5.1 Introduction

Dans ce chapitre, nous allons présenter notre méthode d'estimation de la position du regard. L'objectif est de trouver la relation entre le vecteur du descripteur de l'œil et la position du regard à l'écran. Ce processus d'estimation du regard peut être considéré comme une méthode d'apprentissage supervisé, c'est-à-dire un mécanisme d'induction qui utilise des cas particuliers pour expliquer le cas général. Les "cas particuliers" signifient un ensemble d'exemples observés, étiquetés ou classés. Dans notre cas, nous allons nous intéresser à l'apprentissage supervisé, pour lequel on dispose d'un ensemble d'apprentissage constitué d'exemples d'observations de type entrée-sortie : $\mathcal{D} = \{(x_i, y_i)|i = 1, 2, ..., n\}$ avec $x_i \in \mathcal{X}$ et $y_i \in \mathcal{Y}$. L'entrée est le vecteur du descripteur de l'œil noté $\mathcal{X} \in \mathbb{R}^d$ et la sortie est la position du regard $\mathcal{Y}$. L'objectif est de construire, à partir de cet ensemble d'apprentissage, une fonction $f : \mathcal{Y} = f(\mathcal{X})$ qui nous permette de prévoir la sortie y associée à chaque nouvelle entrée x.

Nous distinguons 2 types de problèmes selon la propriété de la sortie $\mathcal{Y}$:

- la *régression* quand la sortie $\mathcal{Y}$ est quantitative, soit une valeur dans un ensemble continu de réels $\mathbb{R}^m$. Par exemple, le prix d'un stock, la courbe de consommation électrique ou la position du regard en coordonnées 2D sur l'écran où $\mathcal{Y} \in \mathbb{R}^2$ dans notre cas (Figure 5.1 gauche).
- la *catégorisation* ou d'autres termes le *classement* et la *discrimination*[15] lorsque la sortie $\mathcal{Y}$ est qualitative, soit une valeur dans un ensemble de cardinaux finis, par exemple la reconnaissance de chiffres, la survenue d'une maladie. On parle de la discrimination binaire lorsque $\mathcal{Y} =$ {-1, 1}. Dans notre expérimentation, nous proposons un modèle qui permet de classer le regard dans les 5 classes selon les activités oculomotrices différentes : {fixation vers la région de droite en haut de l'écran (DH), de droite en bas (DB), de gauche en haut (GH), de gauche en bas (GB), et le clignement (CL)} (Figure 5.1 droite).

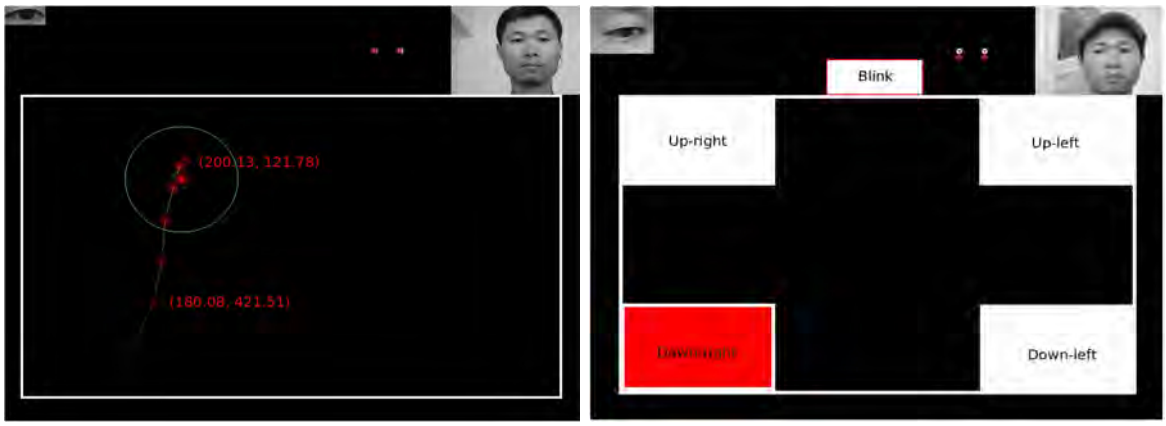


FIGURE 5.1 – Les deux manières d'estimation du regard. A gauche : le regard (le point rouge) est estimé sous forme de coordonnées 2D, exprimées en nombres réels, qui indiquent la position du regard du sujet sur l'écran. A droite, les activités oculomotrices sont classifiés en 5 classes : DH, DB, GH, GB, et CL. La région rouge signifie que la direction du regard actuel est vers la région de droite en bas.

15. L'anglais "classification" se traduit plutôt en français par catégorisation, discrimination ou classement (apprentissage supervisé). Le terme français "classification" désigne le problème de la recherche de classes, soit en anglais "clustering" (apprentissage non-supervisé).[Besse and Laurent, 2014]

Pour la source des exemples $\mathcal{X}$, nous utilisons le vecteur du descripteur au lieu de l'image initiale, parce que l'image de l'œil est souvent bruitée à cause de la qualité de la webcam. Dans la section (3.3), l'histogramme CS-LBP a prouvé sa capacité à caractériser la texture de l'image. L'avantage d'utiliser ce vecteur est non seulement d'être plus representatif pour l'apparence de l'image, mais aussi de réduire la dimension pour optimiser le calcul.

La formation d'un ensemble de données d'apprentissage est cruciale pour la précision de l'estimation de la position du regard, surtout pour une approche fondée sur un modèle d'apparence de l'image. Nous pouvons appeler cette phase *la calibration* : l'objectif est de recueillir tous les attributs nécessaires pour établir un lien entre le modèle d'apparence des yeux et la position du regard. Elle existe dans tous les modèles d'oculomètre et nous avons présenté quatre types de calibration (section 2.3.1.3), soit automatique, soit manuelle. L'avantage de notre méthode est qu'elle est simple et n'a pas besoin de la calibration des matériels, ni de la calibration individuelle, qui sont souvent exigées dans la technique basée sur les caractéristiques des yeux. Par contre, la calibration des marqueurs reste le moyen principal pour recueillir les données d'apprentissage. Généralement cette calibration demande au sujet de fixer successivement plusieurs marqueurs sur l'écran avant l'expérimentation.

Il existe des techniques variées de calibration des marqueurs dans les systèms différents, soit l'utilisateur clique sur chaque marqueur sur l'écran au moment où il le regarde [Tan et al., 2002], soit le système prend l'image des yeux juste à la fin de l'apparition de chaque point sans vérifier la correspondance entre l'image et ce point de la calibration [Nguyen et al., 2009]. Dans notre système, la calibration est réalisée de manière automatique et intelligente. Nous utilisons l'apprentissage par variété sur l'ensemble des images de l'œil capturées pendant la phase de la calibration pour analyser la représentation topologique de faible dimension de cet ensemble. Cette représentation permet de sélectionner les meilleurs échantillons pour former un ensemble de données d'apprentissage pour l'estimation du regard.

Après la calibration, nous pouvons estimer le regard par une méthode de l'apprentissage. Dans la section 5.2, nous allons présenter l'utilisation du processus gaussien pour estimer les coordonnées 2D du regard (x, y), $x \in \mathbb{R}$ et $y \in \mathbb{R}$. Dans la phase de calibration, nous proposons une sélection semi-supervisée pour former un ensemble de données à partir d'un nombre réduit de points de calibration. Dans la section 5.3, nous présenterons un modèle qui permet d'estimer la classe d'un regard par la classification des variétés des exemples pendant la calibration. Nous pouvons utiliser ce modèle pour détecter certains types d'activités oculomotrices pour des applications d'IHM. Enfin dans la section 5.4 nous allons présenter les applications réalisées avec notre méthode.

5.2 Régression par processus gaussien

Dans cette section nous présentons la méthode de la régression par processus gaussien utilisée pour l'estimation du regard. L'objectif de la *régression* est de déduire la valeur $y^* \in \mathcal{Y}$ pour une nouvelle entrée $x^* \in \mathcal{X}$, à partir d'un ensemble d'exemples $\mathcal{D} = \{(x_i, y_i)|i = 1, 2, ..., n\}$, où $x_i \in \mathcal{X}$ et $y_i \in \mathcal{Y}$, soit déduire la fonction f, $\mathcal{Y} = f(\mathcal{X})$. Les *processus gaussiens* permettent de modéliser des fonctions et ils assignent des pro-

babilités à toutes les fonctions f, de la même manière qu'une loi gaussienne assigne des probabilités à tous les réels. Ils ont été appliqués pour estimer le regard comme dans le travail de Ba Linh NGUYEN *et al.*[Nguyen et al., 2009] et celui de Williams *et al.*[Williams et al., 2006]. D'abord nous présenterons la théorie des processus gaussiens, ensuite nous parlerons de la sélection des exemples pour constuire l'ensemble d'apprentissage $\mathcal{D}$. Ensuite, nous présenterons les résultats des expérimentations.

5.2.1 Processus gaussien

De nombreuses méthodes ont été proposées dans le but de modéliser la structure de la fonction f. Une première solution est de supposer qu'elle appartient à une famille de fonctions qui est connue. Par exemple, on peut supposer que la fonction étudiée est linéaire, exponentielle ou trigonométrique si son domaine de définition le permet. Dans ce cas, prédire sa valeur en n'importe quel point devient équivalent à déterminer à partir des observations les paramètres qui la caractérisent. Dans de nombreux cas, un modèle paramétrique est efficace pour modéliser un phénomène, surtout quand certaines considérations physiques viennent supporter le choix du modèle. Cependant, il arrive dans certains cas que le modèle choisi ne corresponde pas aux données, ou bien même que rien ne permette de privilégier un modèle plutôt qu'un autre. Dans ces cas, l'approche paramétrique touche à ses limites et il devient essentiel d'utiliser un modèle qui permette une plus grande souplesse. Une autre solution, l'approche *non paramétrique* ne fait pas de telles hypothèses sur la nature globale de la fonction étudiée. Son principe est d'assigner une probabilité à toutes les fonctions possibles. Bien que toutes les fonctions se voient attribuer une probabilité, certaines seront considérées comme plus plausibles. Les *hyperparamètres* $\mathcal{H}$[Rasmussen and Williams, 2006] permettent par exemple, de favoriser les fonctions lisses. Dans l'approche non paramétrique, rien n'oblige à considérer une quelconque fonction comme impossible, seule compte sa probabilité. Ainsi, un modèle non paramétrique ne contraindra pas la fonction étudiée à être une droite ou une exponentielle, caractérisée par un ensemble de *paramètres*. Par contre, toutes les fonctions possibles ne sont pas équiprobables et les *hyperparamètres* permettent de préciser lesquelles sont plus plausibles que d'autres.

La grande force du modèle *non paramétrique* est qu'après l'observation des données (exemples), les probabilités de toutes les fonctions sont mises à jour pour devenir une probabilité *a posteriori* qui favorise celles qui rendent mieux compte des données. Dans la Figure 5.2(a), nous présentons trois fonctions d'échantillons à partir de la distribution *a priori*. En disposant trois données d'observation $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$, nous pouvons remarquer que les fonctions passent par les points observés (Figure 5.2(b)). La ligne en pointillés représente la fonction moyenne, et la zone grise représentant la variation des fonctions est réduite à proximité des observations.

La combinaison de la distribution *a priori* et des données conduit à la distribution *a posteriori* sur des fonctions. Une telle approche probabiliste permet d'envisager tous les problèmes de régression d'une manière naturelle. La principale difficulté pour la mettre en œuvre est de pouvoir attribuer une probabilité à chaque élément d'un ensemble très grand. C'est sur ce point que les *processus gaussiens* interviennent comme des distributions sur des espaces de fonctions. Ici, une fonction peut être considérée comme un vecteur de longueur infinie, qui donne une valeur pour chacun des points de $\mathcal{X}$. Un

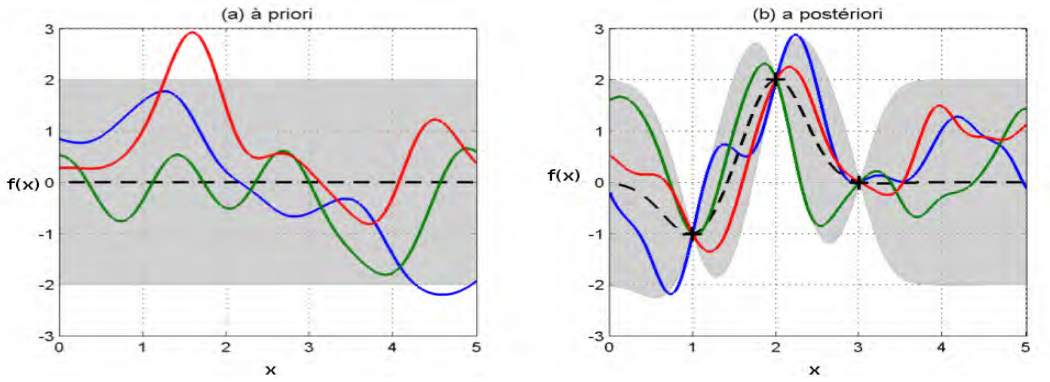


FIGURE 5.2 – Réalisation de processus gaussiens en utilisant a) une distribution *a priori* sur les fonctions et b) la distribution *a posteriori*, après les 3 points d'observation {(1,-1), (2,2), (3,0)}. La ligne en pointillé correspond à la fonction moyenne et la zone grise correspond à 2 fois l'écart type en chacun des points x.

processus gaussien établit une distribution sur de telles données, de la même manière qu'une distribution gaussienne considère le cas de vecteurs d'une longueur finie.

Definition 1. Un processus gaussien est une collection de variables aléatoires, dont tout ensemble fini a une distribution jointe gaussienne multivariée.

Considérons un ensemble $\mathcal{X}$ et n points $X = [x_1, ..., x_n]$ de cet ensemble. Une fonction f de $\mathcal{X}$ est un processus gaussien si pour tout X

$$f(X) = [f(x_1), ..., f(x_n)]^T$$

est distribuée selon une loi gaussienne multivariée :

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(x_1) \\ \vdots \\ \mu(x_n) \end{bmatrix}, \begin{bmatrix} k(x_1,x_1) & \cdots & k(x_1,x_n) \\ \vdots & k(x_l,x_s) & \vdots \\ k(t_n,t_1) & \cdots & k(x_n,x_n) \end{bmatrix} \right)$$

également notée

$$f(X) \sim \mathcal{N}(\mu(X), K(f(X), f(X))),$$

où $\mu(x)$ désigne la moyenne du processus en $x \in \mathcal{X}$:

$$\forall x \in \mathcal{X}, \mu(x) = \mathbb{E}[f(x)]$$

et $k(x, x')$ désigne la covariance de $f(x)$ et $f(x')$:

$$\forall (x, x') \in \mathcal{X} \times \mathcal{X}, k(x, x') = \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))]$$

et le processus gaussien s'écrit alors :

$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x'))$$

Si la fonction moyenne peut être comprise simplement comme la valeur autour de laquelle on s'attend à trouver la valeur de la fonction, pour chaque point de l'espace, la

fonction de covariance, plus subtile, donne la corrélation qu'on s'attend à trouver entre ses valeurs en deux points de l'espace. Par exemple, si la fonction s est supposée lisse, cela signifie qu'on s'attend à une forte corrélation entre deux points qui sont proches dans $\mathcal{X}$. Dans ce cas,

$$k(x, x^{'}) = \sigma^2 exp(-\frac{(x - x^{'})^2}{2l^2}) \qquad (E5.1)$$

où la longueur caractéristique l et la variance σ font partie de l'ensemble de *hyperparamètres* $\mathcal{H}$[Rasmussen and Williams, 2006].

5.2.2 Régression

Disposant d'un ensemble de données d'observation $\mathcal{D} = \{(x_i, y_i)|i = 1, 2, ..., n\}$, où $x_i \in X$ et $y_i \in Y$, nous voulons intégrer à la fonction les connaissances que ces données fournissent. Dans les situations plus réalistes, les valeurs de la fonction f sont souvent bruitées, $Y = f(X) + \epsilon$. En supposant que le bruit additif indépendant est identiquement distribué selon le gaussien ϵ avec une variance σ_n^2, la distribution a priori sur les observations bruitées devient

$$cov(y, y^{'}) = k(x, x^{'}) + \sigma_n^2 \delta_{xx'}$$

où $\delta_{xx'}$ est le symbole de Kronecker, qui vaut 1 si et seulement si $x = x^{'}$ et 0 sinon.

Considérons un ensemble de points de test X^*, et les sorties de test selon l'*a priori* $Y^* = f^*(X^*) = P(Y^*|X^*, \mathcal{D})$, la distribution conjointe de f et f^* est :

$$\begin{bmatrix} f(X) \\ f^*(X^*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(X) \\ \mu(X^*) \end{bmatrix}, \begin{bmatrix} K(f(X), f(X)) & K(f(X), f^*(X^*)) \\ K(f^*(X^*), f(X)) & K(f^*(X^*), f^*(X^*)) \end{bmatrix} \right)$$

La particularité de la distribution gaussienne est que la distribution *a posteriori* se calcule facilement à partir de la distribution jointe et qu'elle est gaussienne.

Alors nous calculons la distribution $f^*(X^*)|f(X)$:

$$f^*(X^*)|f(X) \sim \mathcal{N}(\mu_{post}, K_{post})$$

avec

$$\begin{cases} \mu_{post} = \mu(X^*) + K(f^*(X^*), f(X))K(f(X), f(X))^{-1}(f(X) - \mu(X)) \\ K_{post} = K(f^*(X^*), f^*(X^*)) - K(f^*(X^*), f(X))K(f(X), f(X))^{-1}K(f(X), f^*(X^*)) \end{cases}$$

Cette distribution a posteriori nous donne à la fois une indication sur les valeurs les plus probables de f^* en chacun des points de X^* par le biais de sa moyenne, mais également une indication précieuse sur notre incertitude par le biais de sa matrice de covariance *a posteriori*. Chaque élément de la diagonale de K_{post} nous donne la variance de notre estimée en un point de X^* et représente donc notre incertitude pour la valeur de f^* autour de la moyenne estimée.

5.2.3 Expérimentation

L'objectif de notre expérimentation est d'estimer la position du regard sur un plan devant le sujet, comme par exemple, un écran d'ordinateur. Dans les chapitres précédents, nous avons présenté nos méthodes de localisation de la région de l'œil et d'extraction des caractéristiques de l'apparence des images. Ici nous expliquons l'implémentation de la régression par processus gaussien en utilisant les caractéristiques de l'œil pour estimer les coordonnées du regard. D'abord nous réalisons une procédure de calibration en 5 points sur l'écran, ensuite nous proposons une sélection semi-supervisée pour former un ensemble d'exemples d'apprentissage, suffisant pour atteindre la précision nécessaire pour estimer la position du regard.

5.2.3.1 Calibration

Durant la phase de calibration, on demande au sujet de fixer les points de calibrations un par un pendant un certain temps. C'est une procédure qui permet au système de recueillir les exemples pour l'apprentissage. Théoriquement plus de points de calibration peuvent générer plus d'exemples et en conséquence une fonction f plus pertinente est obtenue pour estimer la position du regard. Pourtant dans la réalisation concrète du système, plus de points de calibration nécessitent que le sujet fixe plus longtemps et provoquent des erreurs humaines à cause de la fatigue ou la déconcentration, etc. Cette tâche devient alors plus complexe et délicate, parce que si le système recueille un mauvais exemple à cause des erreurs humaines, le résultat devient imprécis. Pour éviter ce risque, le nombre des points de la calibration doit donc être réduit, mais doit être néanmoins suffisant pour estimer la position du regard avec une précision correcte.

La Figure 5.3 illustre 3 conditions différentes où il y a respectivement 4 points, 5 points et 8 points de calibration. Chaque point s'affiche à l'écran pendant une seconde et on demande au sujet de fixer ce point. A ce moment-là, le système va capturer les images exemples de l'œil, qui sont ensuite étiquetées en fonction des coordonnées de ce point de calibration.

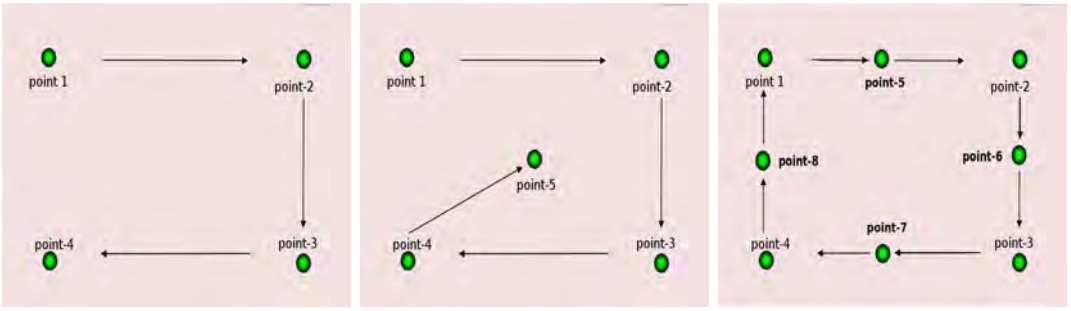


FIGURE 5.3 – 3 conditions de calibration : 4 points, 5 points et 8 points.

Par exemple, la caméra de notre expérimentation permet de capturer 30 images par seconde. Nous pouvons donc obtenir un ensemble d'origine qui contient au maximum 150 images de l'œil pendant la phase de calibration sur 5 points. L'objectif de la calibration est de construire un ensemble d'apprentissage en sélectionnant des exemples parmi les

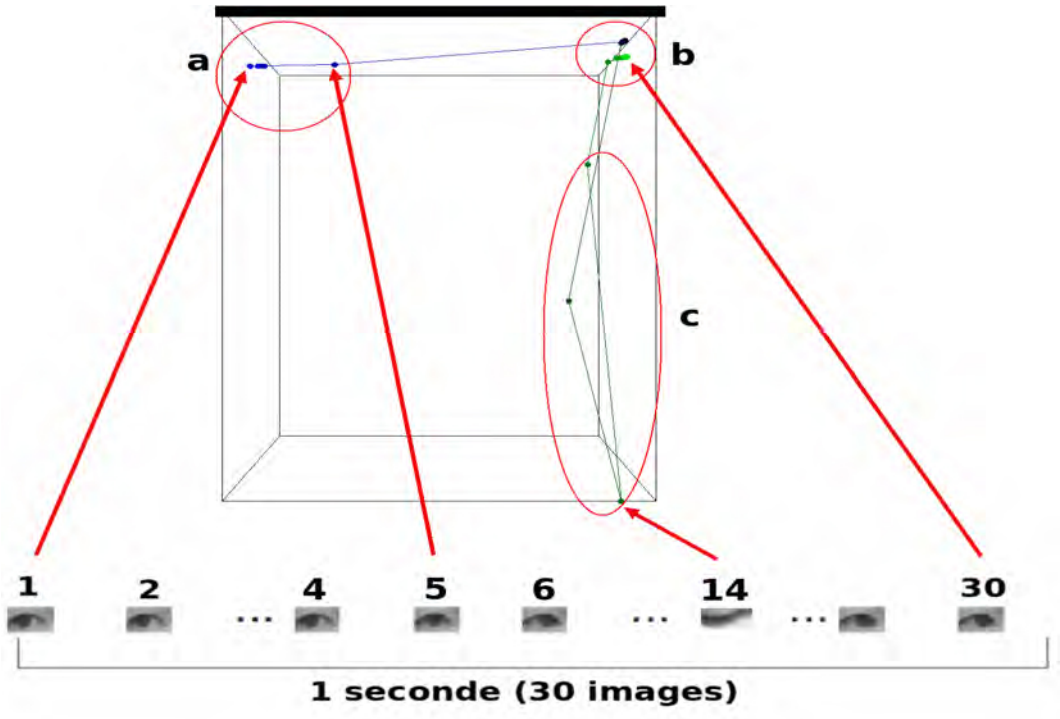


FIGURE 5.4 – Classification par la variété de l'ensemble de 30 images de l'œil capturées pendant une seconde de l'apparition d'un point de calibration sur l'écran. La représentation topologique de la variété permet de distinguer trois classes (a, b, c) selon les différentes activités oculomotrices comme la saccade, la fixation ou le clignement.

150 images. Par rapport à la technique de la sélection, la calibration nécessite parfois une intervention manuelle comme la réalisation décrite dans le travail de [Tan et al., 2002] : la capture de l'image des yeux est effectuée au moment où le sujet clique lui-même sur le point de calibration avec la souris. Lorsque la technique est automatisée le système capture automatiquement la dernière image avant qu'un autre point de calibration s'affiche à l'écran [Nguyen et al., 2009]. Mais cette technique peut également recueillir un mauvais exemple si le sujet cligne les yeux à ce moment précis, et si le système n'est pas capable d'identifier cette erreur.

La calibration de notre système est effectuée de manière automatique et intelligente par l'analyse de la variété de l'ensemble d'origine. La représentation topologique des images en faible dimension permet au système d'identifier et de sélectionner les meilleurs exemples qui correspondent le mieux aux fixations du sujet sur les points de calibration. Lorsque le sujet passe d'un point à un autre point de calibration sur l'écran, on peut observer les différentes activités oculomotrices : les saccades quand le sujet modifie son regard vers un point nouvellement apparu, les fixations quand le sujet fixe son attention sur le point et les autres mouvements comme les clignements, etc.

Par exemple, pendant une seconde de présence d'un point de calibration sur l'écran, nous pouvons obtenir 30 exemples d'images de l'œil (Figure 5.4). Les activités oculomotrices du sujet dans cet exemple sont : la fixation sur le point précédent, puis la saccade pour diriger son regard vers le nouveau point de calibration, ensuite la fixation sur ce point. Pendant la fixation, le sujet a cligné une fois des yeux. En analysant la variété

en 3D de cet ensemble de 30 images, nous pouvons obtenir trois classes d'images, soit les trois cercles rouges indiqués dans la Figure 5.4. Généralement le sujet prend plus de temps à fixer le point de calibration. La classe représentant la fixation sur le point actuel doit avoir un nombre plus important d'images. Chaque classe est associée à un poids ω en fonction du nombre des individus dans la classe. Par exemple, les poids des trois classes dans la Figure 5.4 sont respectivemnt : $\omega_a = 5$, $\omega_b = 22$ et $\omega_c = 3$. Parmi toutes les images de la classe b, qui a un poids plus important, seule l'image la plus près du centre de la classe est étiquetée avec les coordonnées du point de calibration. Les autres images sont considérées non-étiquetées.

Pendant la phase de la calibration, seuls certains exemples d'image sont étiquetés. Le nombre d'exemples étiquetés est trop faible pour apporter suffisamment d'informations nécessaires pour estimer la position du regard. Par contre dans notre cas, nous pouvons obtenir un ensemble d'exemples non-étiquetés plus nombreux. Il est donc préférable que les exemples non-étiquetés et étiquetés soient utilisés ensemble.

5.2.3.2 Apprentissage semi-supervisé

L'objectif de l'apprentissage semi-supervisé est, de même qu'en apprentissage supervisé, de proposer un modèle permettant de prédire une valeur d'une entrée (*régression*), ou de représenter l'appartenance de l'entrée à différentes classes prédéterminées (*catégorisation*), mais en se servant également des exemples non-étiquetés qui sont susceptibles d'apporter de l'information quant à la distribution sous-jacente des exemples dans leur espace de description.

Après avoir fait une calibration de n points, nous disposons d'un ensemble d'exemples étiquetés $\mathcal{D} = \{(x_i, y_i)|i = 1, 2, ..., n\}$ et d'un ensemble d'exemples non-étiquetés $\mathcal{D}^* = \{x_j^*|j = 1, 2, ..., m\}$. L'entrée x est le vecteur du descripteur extrait par le modèle d'apparence : la combinaison des histogrammes CS-LBP de chaque bloc dans l'image. Par exemple, chaque image de l'œil de 60×40 pixels est divisée en 20 blocs. Pour tous les exemples $x, x^* \in \mathcal{D} \cup \mathcal{D}^*$, nous appliquons une classification spectrale (section 5.3.1) pour obtenir l classes $\mathcal{U} = \{\mathcal{U}_1, ...\mathcal{U}_l\}$, où $n \leq l < n + m$, associés aux poids $W = \{w_1, ...w_l\}$.

Nous voulons obtenir un ensemble actif $\mathcal{A}$ par une manière semi-supervisée en quatre étapes et le schéma explicatif est présenté dans la Figure 5.5 (Gauche) :

1) Prendre les n classes les plus importantes par rapport aux poids w les plus importants pour former une classe $\mathcal{U}_{labelled} = \{\mathcal{U}_1, ...\mathcal{U}_n\}$, $\mathcal{U}_{labelled} \subset \mathcal{U}$.

2) Former la classe $\mathcal{D} = \{(x_1, y_1), ..., (x_n, y_n)\}$ où $||\mathcal{X}_i - \mu_i|| < T$, $\mathcal{X}_i \in \mathcal{U}_{labelled}$, $i \in n$. μ_i est le point centrique de la classe $U_{labelled}$ et T est le seuil, $T \in \mathbb{R}$.

3) Former $\mathcal{D}^* = \{x_1^*, ..., x_j^*\}$, où $x_{j,k} \in \mathcal{U} - \mathcal{U}_{labelled}$ et $||x_j - x_k|| > T_2$, $T_2 \in \mathbb{R}$. Pour chaque exemple non-étiqueté x_j^*, prédire $y_j^* = P(y_j^*|x_j, \mathcal{D})$ par le processus gaussien et former un nouvel ensemble $\mathcal{U}_{semi} = \{(x_1, y_1), ..., (x_j, y_j)\}$.

4) Obtenir l'ensemble actif $\mathcal{A} = \mathcal{U}_{labelled} \cup \mathcal{U}_{semi}$ pour estimer de nouveau le point du regard par le processus gaussien.

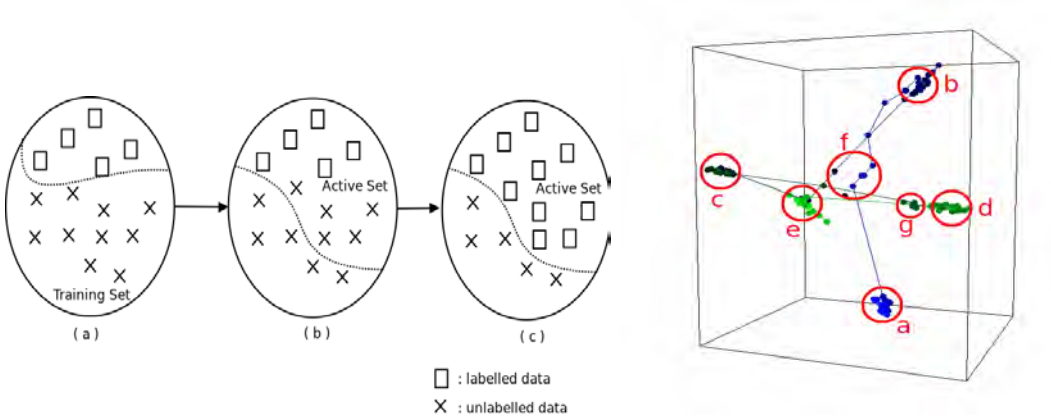


FIGURE 5.5 – Gauche : Génération d'un ensemble actif $\mathcal{A}$: les carrés représentent les exemples étiquetés et les croix réprésentent les exemples non-étiquetés. Droite : Classification d'images selon la variété de l'ensemble d'images de l'œil capturées pendant une calibration en 5 points. Les classes (cercles rouges) a,b,c,d,e représentent les 5 ensembles des images de l'œil qui correspondent aux fixations sur les 5 points de calibration. Les deux autres classes sont les différents comportements oculo-moteurs comme les saccades ou les fixations sur d'autres endroits de l'écran.

La Figure 5.5 (Droite) illustre la classification d'exemples selon la variété de l'ensemble d'exemples obtenus pendant une calibration sur cinq points. Selon les poids des classes, nous pouvons distinguer les 5 classes les plus importantes (a, b, c, d, e) qui correspondent respectivement aux fixations du sujet sur cinq points de calibration. L'exemple le plus proche du centre de chaque classe est étiqueté. Les exemples dans les 2 classes (f, g) représentent des variations des activités oculomotrices comme le changement de la direction du regard (les saccades), ou une attention visuelle temporaire sur un autre endroit de l'écran. Ils sont non-étiquetés, mais ils peuvent apporter de l'information sur l'exploration visuelle de l'écran en analysant la structure topologique de la variété de l'ensemble d'apprentissage semi-supervisé $\mathcal{A}$.

Le principe original et novateur de la formation de cet ensemble $\mathcal{A}$ est d'utiliser d'abord la classification des exemples par la variété générée pour choisir automatiquement le meilleur exemple correspondant à chaque point de calibration, et d'utiliser ensuite la distribution des exemples non-étiquetés dans la variété pour compléter et renforcer l'ensemble de données étiquetées $\mathcal{D}$.

5.2.3.3 Résultats

Nous présentons les résultats de l'utilisation de la méthode du processus gaussien et de la formation de l'ensemble d'apprentissage semi-supervisé pour estimer la position du regard en 2D (x, y). Les matériels du système sont un portable Macbook Pro 8.1 et la webcam Microsoft LifeCam HD-5000 dont la fréquence est de 30 Hz pour une résolution de 640×480 pixels. La distance entre le sujet et l'écran correspond environ à la longueur du bras du sujet.

Pour évaluer la précision de l'estimation par notre méthode, on demande d'abord au sujet d'effectuer une phase de calibration (Figure 5.3). Chaque point de calibration

s'affiche pendant une seconde. Le sujet fixe ainsi 24 points dans une région de taille 800×600 pixels à l'écran, comme les points verts dans la Figure 5.6.

Le système proposé se compose de 3 modules :

1. Nous utilisons la méthode présentée dans le chapitre 3 pour localiser la région de l'œil au cours de l'expérimentation. Nous utilisons également le modèle d'apparence pour extraire les caractéristiques de l'image de l'œil obtenue. Une image de 60×40 pixels sur la région de l'œil est obtenue et cette image est divisée en 20 blocs. La concaténation des histogrammes CS-LBP de chaque bloc est utilisée en tant que vecteur du descripteur qui représente cette image de l'œil. Pour simplifier, l'image de l'œil mentionnée désigne le vecteur du descripteur traité.

2. Pour la calibration, nous proposons 3 conditions différentes (4, 5 et 8 points) afin de comparer les résultats. Pour une seconde d'affichage du point de calibration à l'écran, nous obtenons généralement 30 images de l'œil et nous effectuons la classification spectrale pour classifier ces images. Chaque classe $\mathcal{U}$ est associée à un poids ω en fonction du nombre des images classifiées dans la même classe. La Figure 5.5 (à droite) illustre la représentation topologique de la variété de l'ensemble d'exemples obtenu pendant une calibration de 5 points. Nous distinguons non seulement les classes avec les poids les plus importants comme (a,b,c,d,e) qui représentent les fixations du sujet vers les cinq points de calibration, mais aussi les classes (f,g) qui représentent les différents comportements oculo-moteurs comme les saccades ou les fixations sur d'autres endroits de l'écran. Pour former l'ensemble d'apprentissage semi-supervisé, nous choisissons l'image près du centre de chaque classe (a,b,c,d,e) comme l'exemple étiqueté et les autres images des classes (f,g) comme les exemples non-étiquetés.

3. La position d'un point du regard est représentée par les coordonnées de deux valeurs réelles x et y. Techniquement nous appliquons la regression du processus gaussien indépendamment sur ces deux valeurs. La longueur caractéristique l et la variance σ du processus gaussien dans l'équation (E 5.1) sont respectivement de 500 et de 1.25. Par la phase de calibration, nous pouvons obtenir deux ensembles : l'ensemble d'exemples étiquetés ($\mathcal{D}$) et l'ensemble d'exemples non-étiquetés ($\mathcal{D}^*$). Pour estimer la position du regard, la construction de l'ensemble d'apprentissage est importante et nous proposons une "*double*" application de la régression par processus gaussien. La première application de la régression vise à estimer les sorties des exemples non-étiquetés de l'ensemble $\mathcal{D}^*$ à partir de l'ensemble étiqueté $\mathcal{D}$. La représentation topologique de la variété 3D de toutes les images de l'œil dans $\mathcal{D}$ et $\mathcal{D}^*$ permet d'estimer les sorties des exemples correctement et un nouvel ensemble d'apprentissage est formé par $\mathcal{D}$ et $\mathcal{D}^*$. Ensuite, la deuxième application de la régression permet d'estimer la position d'un nouveau regard en utilisant l'ensemble d'apprentissage $\mathcal{D} \cup \mathcal{D}^*$.

L'objectif de notre expérimentation est d'évaluer la précision du système proposé. Le sujet fixe 24 points verts sur l'écran. Pour chaque point, nous calculons la différence entre la position du regard estimée et la position réelle du point. La précision du système est calculée par la moyenne des différences sur les 24 points et elle est présentée en degré d'angle visuel (Annexe 1), qui est utilisé comme indice de la précision d'un système

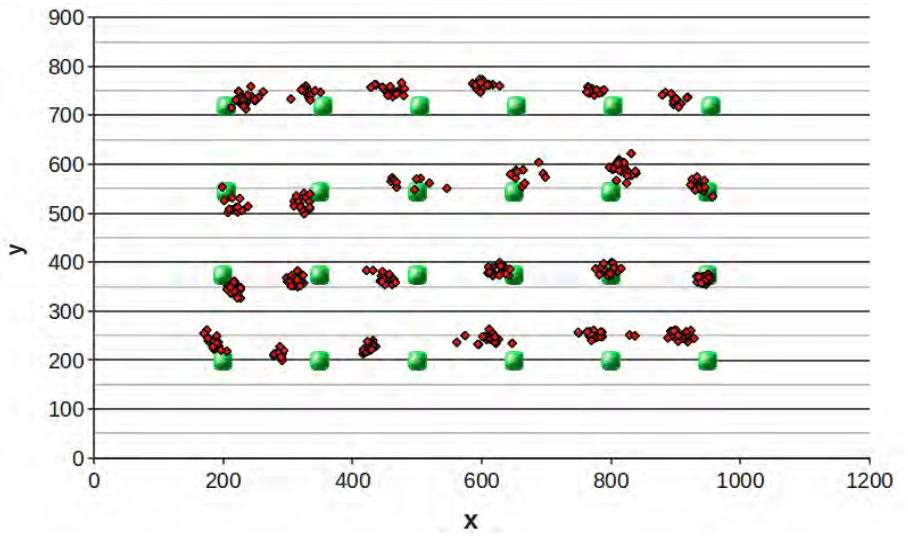


FIGURE 5.6 – L'estimation de la position du regard (point rouge) vers les 24 points (points verts) dans une région de taille 800×600 pixels. La précision obtenue par notre méthode est de 0.92° en moyenne, et l'écart-type est de 0.7°.

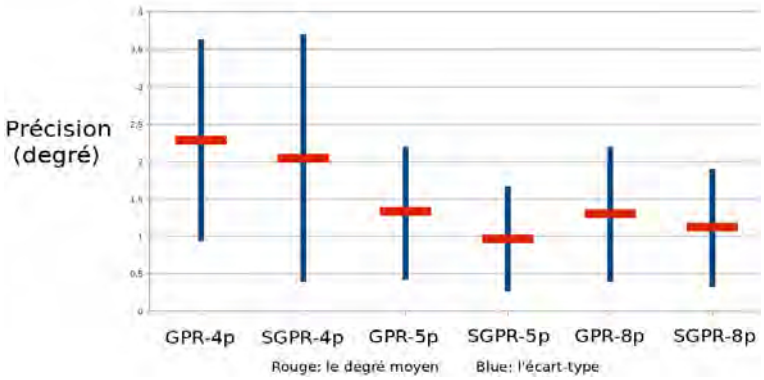


FIGURE 5.7 – Comparaison des résultats obtenus par les méthodes supervisée (GPR) et semi-supervisée (SGPR) en utilisant 3 conditions de calibration (4, 5 et 8 points).

oculométrique. Le degré θ est calculé par :

$$\theta = \frac{atan(\frac{d_r}{f}) * 180}{\pi}$$

où f est la distance réelle entre l'œil et l'écran et d_r est la distance réelle entre deux points de l'écran.

$$d_r = d_p \times \frac{T}{R}$$

d_p est la distance entre deux points de l'écran en pixels, R est la résolution dans la direction utilisée (horizontale ou verticale) en pixels et T est la taille de l'écran dans la direction utilisée en mm. Dans notre expérimentation, f est 600 mm. $\frac{T}{R}$, soit la taille réelle d'un pixel de l'écran, est 0.226 mm.

Pour comparer avec notre méthode semi-supervisée d'estimation, nous appliquons également la méthode supervisée de régression par processus gaussien en utilisant trois

Méthode	Précision	Exemples d'apprentissage
notre méthode	0.92°	5 étiquetés et 20∼50 non-étiquetés
S^3GP+ contour+filtre[Williams et al., 2006]	0.83°	16 étiquetés et 75 non-étiquetés
S^3GP [Williams et al., 2006]	1.32°	16 étiquetés et 75 non-étiquetés
Tan *et al.* [Tan et al., 2002]	0.5°	252 étiquetés
Baluja *et al.* [Baluja and Pomerleau, 1994]	1.5°	2000 étiquetés

Table 5.1 – Comparaison des résultats.

conditions de calibration (4, 5 et 8 points). La Figure 5.7 montre les résultats en précision obtenus avec les six techniques différentes : GPR-4p, GPR-5p, GPR-8p, SGPR-4p, SGPR-5p, SGPR-8p. La méthode supervisée GPR-4p utilise la regression par processus gaussien pour estimer la position du regard en utilisant quatre exemples étiquetés qui proviennent de la calibration sur quatre points. La technique SGPR-5p utilise notre méthode semi-supervisée avec une calibration sur cinq points. Parmi les positions des regards, nous avons traité les fixations sur chaque point vert et calculé au final le degré moyen θ_{moyen} et l'écart-type. Les résultats de la Figure 5.7 montrent que la méthode semi-supervisée est plus efficace en utilisant plus d'exemples non-étiquetés et que le système avec cinq points de calibration permet d'obtenir une précision correcte. La méthode de la régression par processus gaussien avec cinq points de calibration (SGPR-5p) peut atteindre une précision $\theta_{moyen} = 0.92°$ avec l'écart-type de 0.7°. En comparant avec d'autres travaux dans la Table 5.1, la méthode que nous proposons permet d'obtenir une précision très correcte avec très peu de points de calibration. A l'inverse, les autres travaux ont besoin d'utiliser plus d'exemples étiquetés pour atteindre une précision inférieure à 1°. La figure 5.6, où les points verts sont les points à fixer à l'écran et les points rouges correspondent à la position estimée du regard, illustre la précision de notre méthode.

5.3 Catégorisations des activités oculomotrices

L'objectif de la méthode de l'apprentissage supervisé est d'estimer la fonction f, où $\mathcal{Y} = f(\mathcal{X})$ en utilisant un ensemble de données d'exemple. Nous distinguons le problème de la *catégorisation* quand $\mathcal{Y}$ est une valeur qualitative dans un ensemble discret non-ordonné, par exemple la reconnaissance de lettres. Cette section s'intéresse à la méthode de classification spectrale (spectral clustering) et à sa mise en œuvre pour effectuer le partitionnement des mouvements oculaires. La classification spectrale utilise le laplacien du graphe normalisé. Dans le modèle de *régression*, le Laplacian Eigenmaps, fondé sur le laplacien du graphe, est utilisé dans la phase de calibration pour sélectionner les exemples d'apprentissage par la structure topologique de la variété de l'ensemble des exemples. Ici la variété issue du Laplacian Eigenmaps est utilisée directement pour le problème de la *catégorisation* en appliquant la méthode des k-moyennes.

5.3.1 Classification spectrale

[Donath and Hoffman, 1973] ont proposé la classification spectrale (spectral clustering), dont l'idée est de construire les partitions du graphe à la base des vecteurs propres de la matrice d'adjacence. La classification spectrale a été beaucoup développée, et utilisée dans différentes communautés de recherche. Une vue d'ensemble sur la classification spectrale est présentée dans l'article de [Spielman and Teng, 1996]. Le succès de la classification spectrale est notamment lié au fait qu'elle ne fait pas d'hypothèse sur la forme des partitionnements des données. En plus, elle peut être implémentée efficacement pour les données à grande échelle.

La méthode de la classification spectrale consiste à extraire les vecteurs propres associés aux plus grandes valeurs propres d'une matrice d'affinité normalisée. Ces vecteurs propres constituent un espace de dimension réduite dans lequel les données transformées seront linéairement séparables. Il existe de nombreuses versions d'algorithmes de classification spectrale. Ici on s'intéresse à la classification spectrale qui utilise le graphe du Laplacien normalisé, comme celle proposée par [Shi and Malik, 2000] (c.f. Algorithme 1). Cette version est fondée sur un partitionnement bipartite récursif à partir du vecteur propre associé à la seconde plus grande valeur propre du laplacien du graphe normalisé.

Pour un ensemble de données $X = \{x_1, ..., x_n\}$, L'objectif est de partitionner cet ensemble de points $X = x_1, ..., x_n \subset \mathbb{R}^p$ en l classes où l est fixé. Dans l'algorithme 1, les notions comme le graphe de similarité G, la matrice de similarité W ont été présentées dans la section 4.3.3.1.

Algorithme 1 : spectral clustering normalisé d'après Shi et Malik (2000)

Input : Ensemble de données X, Nombre de classes l

- Construction du graphe de similarité $G = (X, E, \omega)$ à partir de l'ensemble X. La matrice de similarité W est définie par : $W = \omega(x_i, x_j) = (\omega_{ij})_{i,j=1,...,n}$.
- Construction du graphe de Laplacien normalisé

  $$L_{rw} = D^{-1}W$$

  où D est la matrice des degrés des sommets $D_{ii} = D(x_i, x_i) = \sum_{x_{i,j} \in X} \omega_{ij}$ et $D(x_i, x_j) = 0$ pour $x_i \neq x_j$.
- Construction de la matrice $V = [v_1, ..., v_l] \in \mathbb{R}^{n \times l}$ formée à partir des l plus grands vecteurs propres v_i, $i \in 1, ..., l$.
- Construction de la matrice $Y = [y_1, ..., y_n]$ où $y_i \in \mathbb{R}^l$ correspond à la i-ème ligne de V.
- Classer les points $y_i \in Y$ en l classes via la méthode k-moyennes et assigner le point original x_i à la classe j si et seulement si la ligne i de la matrice Y est assignée à la classe j.

La méthode des k-moyennes est une méthode de classification qui permet de mettre au jour une éventuelle structure de groupes dans un ensemble de données. Les premiers articles présentant des aspects théoriques remontent aux années cinquante, avec notamment les travaux de Cox et de Fisher. Mais la méthode est toujours d'actualité. Pour un nombre fixé k, la méthode des k-moyennes cherche à rassembler les observations les plus similaires autour de k centres $\{T_1 ..., T_k\} \subset \mathbb{R}^d$ définis. Une fois les centres définis, les classes peuvent être construites. La j-ème classe rassemble les observations plus proches du j-ème centre que des autres centres.

On note x_i $(1 \leq i \leq N)$ les N observations que l'on souhaite partionner. L'algorithme des k-moyennes se déroule comme suit :

1). On choisit k observations au hasard parmi les N observations de la population. Soit $(R_1, R_2, ..., R_k)$, la famille des k observations sélectionnées. Ces dernières sont les "représentants" des k classes $(C_1, C_2, ..., C_k)$. On les appelle aussi les centres des k classes.

2). On assigne chaque observation à l'une des classes en fonction du centre le plus proche selon un principe de distance ou de similarité : $argmin_{j,1\leq j\leq k} d(x_i, R_j)$, où d est une distance ou une similarité entre les observations.

3). On calcule les nouveaux centres pour les classes. Ces nouveaux représentants de classes correspondent à la moyenne des observations de la classe. Le calcul se fait comme suit :

$$\forall j, 1 \leq j \leq k, R_j = \frac{1}{|C_j|} \sum_{x \in C_j} x$$

4). On retourne à l'étape 2) jusqu'à ce que deux itérations successives conduisent à une même partition, c'est-à-dire que deux itérations successives donnent les mêmes représentants des classes.

5.3.2 Modèle prédictif

Parmi les observations (exemples), nous pouvons en avoir certaines déjà étiquetées et certaines dont la classe est inconnue. Nous utilisons la méthode des k-moyennes pour regrouper les observations selon la représentation topologique de la variété de cet ensemble d'observations. Nous pouvons donc assigner une observation non-étiquetée à la j-ème classe si elle est plus près du j-ème centre.

Dans notre expérimentation, nous proposons un modèle qui utilise la classification spectrale pour reconnaitre certains types des mouvements des yeux, par exemple les cinq types de mouvements spécifiques à la direction du regard : la fixation à droite en haut (DH), à droite en bas (DB), à gauche en haut (GH), à gauche en bas (GB) de la région (voir la Figure 5.1), ainsi que le clignement des yeux (CL) [16].

Le clignement des yeux est un mouvement spécial et la détection du clignement est appliquée de plus en plus dans les domaines comme l'IHM, la santé, la sécurité routière, etc. Les méthodes de détection du clignement sont généralement divisées en 2 catégories[Le et al., 2013] :

16. Nous clignons des yeux près de 30000 fois par jour. Le nombre de fois qu'une personne cligne des yeux par minute est en moyenne entre 15 et 20. Un clignement dure entre 100 et 150 millisecondes.

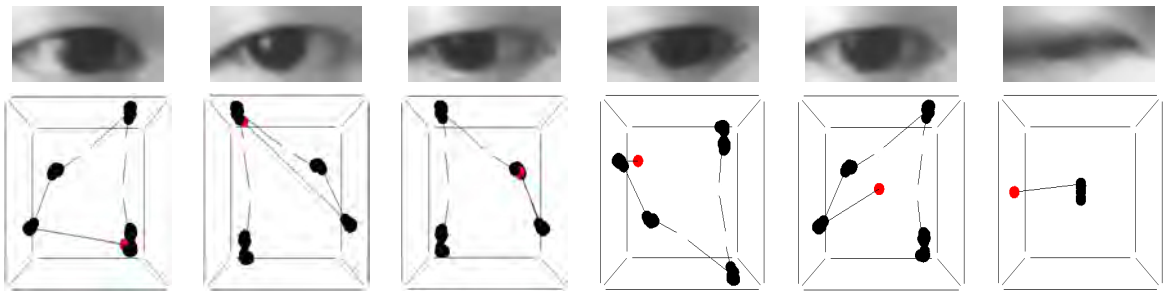
1. **Modèle du contour des yeux**. Cette approche nécessite souvent d'abord de détecter les yeux correctement, ensuite de mettre ce modèle autour des yeux afin de s'adapter au contour des yeux. Les défis résident dans le changement d'illumination et la distinction entre les yeux ouverts et les yeux fermés.
2. **Apprentissage d'images**. Par exemple, l'utilisation de l'analyse en composantes principales permet d'apprendre la forme des yeux à partir d'un ensemble de données d'apprentissage pour discriminer les yeux ouverts et les yeux fermés. Cette approche traite directement un ensemble d'images pour apprendre l'apparence de l'objet dans l'image.

Notre méthode est fondée sur un modèle d'apparence de l'œil et elle utilise la classification spectrale pour partitionner les images de l'œil par variété. Les différentes classes générées nous permettent de déduire le comportement oculo-moteur. Notre modèle est conçu pour reconnaitre les cinq types de mouvements présentés ci-dessus : DH, DB, GH, GB et CL. Autrement dit, après avoir localisé l'image de l'œil, le système permet d'estimer la classe à laquelle cette image appartient.

Le principe d'estimation est expliqué dans la Figure 5.8. Dans cet exemple, nous obtenons un ensemble d'apprentissage qui comporte 32 images de l'œil étiquetées (points noirs dans la figure). Nous avons distingué quatre classes, qui représentent les regards du sujet vers les 4 régions de l'écran (DH, DB, GH, GB), par la structure topologique de la variété de cet ensemble. A partir de cet ensemble d'apprentissage, nous voulons connaitre la classe d'une nouvelle image de l'œil, comme celles présentées dans la première ligne de la Figure 5.8. Nous ajoutons alors cette nouvelle image à l'ensemble d'apprentissage et analysons la variété de ce nouvel ensemble. Nous pouvons ainsi connaitre la classe de la nouvelle image par son emplacement dans la variété générée. Dans la Figure 5.8, a)b)c)d) démontrent que chaque nouvelle donnée appartient respectivement aux 4 classes (DH, DB, GH, GB) en évaluant la distance entre le point rouge et le centre de chaque classe. Quand le regard est au centre de l'écran, le point rouge n'appartient à aucune classe (e). Nous remarquons également que le clignement (f) perturbe brutalement la structure des variétés.

A partir des caractéristiques observées, nous proposons un modèle pour reconnaitre ces cinq types de mouvements d'une manière efficace. Ce modèle peut être utilisé pour les applications d'IHM, où le déplacement du regard pourrait être utilisé pour sélectionner les éléments de l'interface et le clignement de l'œil utilisé pour valider le choix comme lorsqu'on clique sur le bouton d'une souris. Le modèle proposé se compose de 3 parties : la localisation de l'œil, la calibration et la prédiction. Le procédure de la localisation de l'œil a été présentée dans le chapitre 3. Ici nous parlons des phases de calibration et de prédiction.

- **La calibration** : Cette phase est légèrement différente de la calibration dans le modèle de *régression*, dont l'objectif est de chercher les exemples qui ont le plus de variations possibles. La calibration du modèle de *catégorisation* utilise également la classification spectrale pour sélectionner les exemples d'apprentissage, mais les exemples sélectionnés ne représentent que les quatre régions de l'écran. Dans ce cas, après avoir appliqué la classification spectrale sur les images de la calibration, les quatre classes principales de la variété sont distinguées et les exemples qui se situent près du centre de chaque classe sont sélectionnés. Nous pouvons donc obtenir quatre classes C_1 =DH, C_2 =DB, C_3 =GH, C_4 =GB.



(a) (b) (c) (d) (e) (f)

FIGURE 5.8 – La variété d'un ensemble qui contient 32 images de l'œil et une nouvelle image de classe inconnue (les images de la première ligne). Dans la structure topologique de la variété, les points noirs représentent les 32 images de l'œil qui représentent les regards du sujet vers les 4 régions de l'écran (DH, DB, GH, GB). Le point rouge représente la nouvelle image de l'œil. Par l'emplacement du point rouge, nous pouvons connaitre sa classe. a)b)c)d) : la classe du point rouge appartient à chaque classe (DH, DB, GH, GB). e) le point rouge représente le regard vers le centre de l'écran. f) le point rouge représente le clignement (CL).

- **La prédiction** : Pour estimer la classe d'une nouvelle entrée, celle-ci est d'abord ajoutée à l'ensemble d'exemples d'apprentissage $\mathcal{A}$ qui a été construit pendant la calibration. Ensuite, nous analysons la variété 3D de ce nouvel ensemble de données $\mathcal{A}^*$. Après nous appliquons la méthode des k-moyennes pour classifier tous les exemples de $\mathcal{A}^*$ selon leur position dans cette variété. La méthode des k-moyennes est une méthode de classification qui permet de mettre au jour une éventuelle structure de classes dans un ensemble de données. Avant de connaitre la vraie classe de la nouvelle entrée, nous assignons une classe C^* à cette nouvelle entrée. Après avoir appliqué la méthode des k-moyennes, les classes des données sont construites : soit la nouvelle entrée appartient aux quatre classes définies (C_1 =DH, C_2 =DB, C_3 =GH, C_4 =GB), soit la nouvelle entrée est classée dans une classe indépendante C_5, comme par exemple, les cas (e, f) dans la Figure 5.8.

Le schéma présenté dans la Figure 5.9 montre les composants de trois parties du modèle et leurs fonctionnements. Ce modèle proposé peut être utilisé pour catégoriser des mouvements des yeux en temps réel dans une séquence d'images capturée par la webcam.

5.3.3 Expérimentation

Dans cette section nous allons présenter la réalisation du modèle proposé pour reconnaitre les cinq types des mouvements des yeux : DH, DB, GH, GB, CL. Nous utilisons une webcam pour capturer des images. Le modèle de webcam est Microsoft LifeCam HD-5000 avec une fréquence de 30 Hz pour une résolution de 640×480 pixels. La phase de localisation de l'œil nous fournit une image de 60×40 pixels sur la région de l'œil. Nous divisons cette image en 20 blocs, et combinons les histogrammes CS-LBP de chaque bloc pour former le vecteur du descripteur représentant cette image.

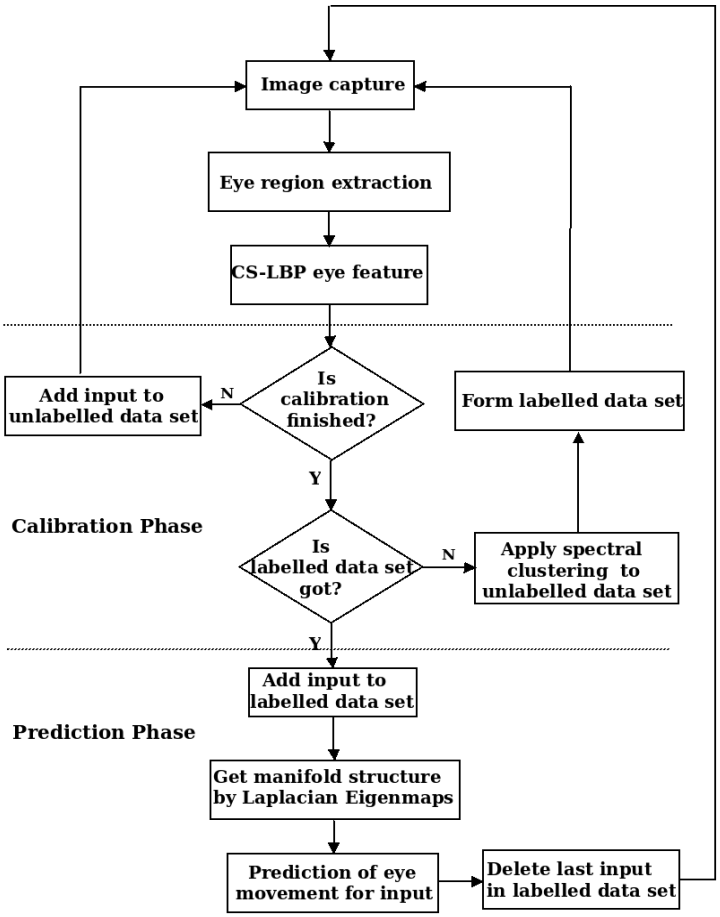


FIGURE 5.9 – Le schéma du modèle.

L'objectif de la procédure de la calibration est de former un ensemble d'images de l'œil qui représente les fixations vers les quatre régions de l'écran : DH, DB, GH, GB. Nous pouvons simplifier la procédure de la calibration en demandant au sujet de regarder lui-même les quatre coins de l'écran sans même afficher les points à l'écran. Cette calibration peut être relativement simple à faire par le sujet. Pendant une calibration de quatre secondes, nous demandons au sujet de fixer quatre régions de l'écran dans l'ordre suivant : DH→GH→GB→DB et nous pouvons obtenir environ 100 images de l'œil. Nous appliquons la classification spectrale pour trouver les quatre classes qui représentent les fixations vers les quatre régions. La technique a été présentée dans la section 5.2.3.1. Ensuite, nous sélectionnons, pour chaque classe, les trois exemples qui sont les plus proches du centre de la classe et nous pouvons donc former un ensemble d'apprentissage $\mathcal{A}$ avec 12 exemples d'images de l'œil.

Après la calibration, chaque fois qu'une nouvelle image de l'œil est obtenue par la procédure de la localisation des yeux, le modèle va estimer la catégorie de cette image. Nous ajoutons d'abord cette image à l'ensemble d'apprentissage pour former un nouvel ensemble $\mathcal{A}^*$ et appliquons le Laplacian Eigenmaps pour analyser la variété de l'ensemble $\mathcal{A}^*$ qui contient 13 exemples (12 exemples d'apprentissage + 1 exemple nouveau à estimer). Ensuite nous utilisons la méthode des k-moyennes pour classifier les exemples. Le nombre des classes k est fixé à 5. Dans l'ensemble $\mathcal{A}^*$, nous avons eu 12 exemples étiquetés de quatre classes : DH, DB, GH, GB. La première étape

de la méthode des k-moyennes est de choisir les centres des classes. Nous choisissons quatre exemples étiquetés en tant que centres des classes (C_1 =DH, C_2 =DB, C_3 =GH, C_4 =GB). L'exemple non-étiqueté, soit la nouvelle image de l'œil, est choisi comme le centre de la classe C_5.

Apès les itérations de la méthode des k-moyennes, nous pouvons obtenir trois résultats de regroupement des classes :

1). La classe C_5 est vide et l'exemple de la nouvelle image est classée dans une des quatre classes : DH, DB, GH, GB.

2). La classe C_5 contient seulement l'exemple non-étiqueté et les classes C_1, C_2, C_3, C_4 ne changent pas, comme dans la Figure 5.8 (e). Dans ce cas, l'exemple témoigne du fait que la direction du regard est vers la partie centrale de l'écran.

3). La classe C_5 contient un seul exemple et la structure topologique des autres classes change, comme le montre la Figure 5.8 (f). Dans ce cas, l'exemple de l'image représente un clignement des yeux. Pour distinguer le changement de la structure des classes, nous calculons un paramètre *ratio* :

$$ratio = \frac{\sum_{i \in k-1} d(T^*, T_i)}{\sum_{i,j \in k-1} d(T_i, T_j)}$$

où d est la distance, T est le centre d'une classe. T^* est le centre de la classe C_5. Si le paramètre *ratio* est supérieur à un seuil défini, on considère que la stucture topologique des classes est perturbée. Dans notre expérimentation, la valeur empirique pour ce seuil est fixé à 2.

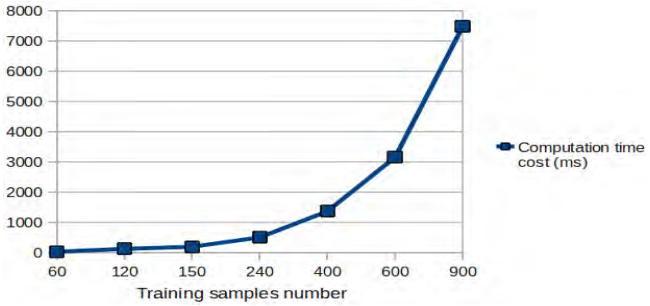


FIGURE 5.10 – Temps de calcul par la classification spectrale en fonction du nombre d'exemples.

Le temps de calcul pour la prédiction dépend du nombre d'exemples utilisés à chaque étape pour l'apprentissage de variété et du nombre d'itérations par la méthode des k-moyennes. La Figure 5.10 illustre le temps de calcul en millisecondes de l'application du Laplacian Eigenmaps sur un ensemble d'exemples dont le nombre varie. Pour 60 exemples, le temps de calcul est de 41ms et de 137ms pour 120 exemples. Dans notre expérimentation, notre modèle qui utilise au maximum 13 exemples peut donc être utilisé en temps réel.

5.4 Applications

Dans cette section, nous présentons les réalisations expérimentales qui ont été mises en oeuvre pour tester la validité de notre modèle oculométrique ONE. Les trois premiers

projets présentés dans les sections suivantes correspondent à l'application du modèle de *régression* pour estimer la position du regard en coordonnées, ce qui nous permet d'étudier l'attention visuelle du sujet sur l'écran. La dernière expérimentation concerne le développement d'un système de commande oculomotrice en utilisant la technique de *catégorisation* qui permet d'utiliser la fixation du regard sur une région donnée et le clignement pour contrôler un événement à l'écran.

5.4.1 Projet Tatihou

Le projet Tatihou est une recherche menée par le laboratoire CHArt et le musée maritime de l'île de Tatihou en Normandie à partir de mars 2014. Ce musée assure la conservation et la mise en valeur de collections d'archéologie sous-marine, d'ethnologie maritime et de beaux-arts et présente chaque année des expositions sur ces thèmes. Ce projet vise à étudier les caractéristiques de l'exploration visuelle des œuvres d'art, comme un tableau.

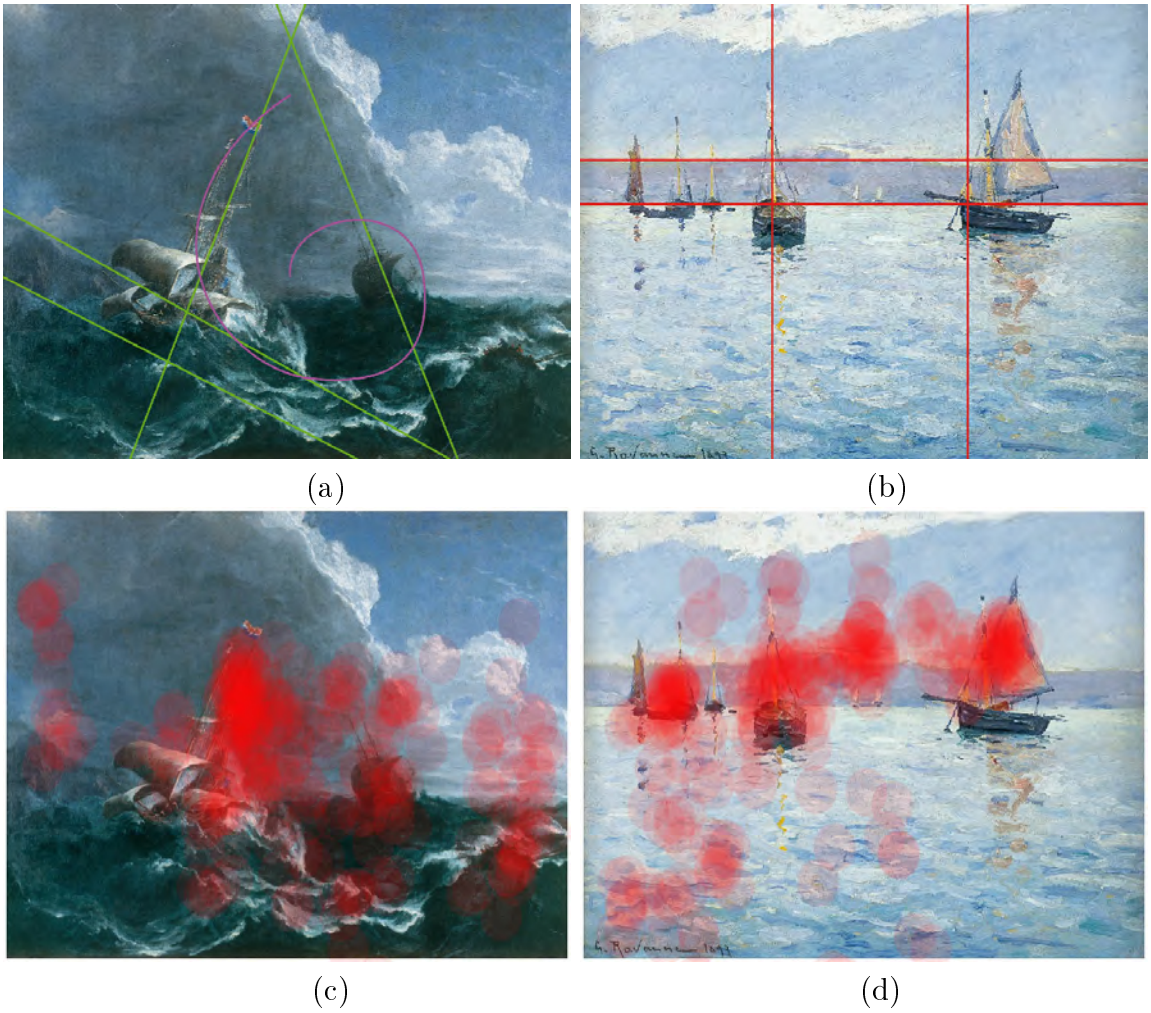
Dans notre expérimentation, l'image numérique du tableau est affichée sur l'écran. Pendant que le sujet regarde ce tableau, notre système enregistre les regards du sujet. Les résultats de l'exploration visuelle sont montrés au sujet à la fin de la session. Les deux tableaux retenus par les experts du musée pour l'expérimentation sont celui de Matthieu de Plattemotagne et celui de Léon-Gustave Ravanne, qui correspondent au thème de l'exposition "Marine".

Notre travail a comporté 2 phases :

1. Réalisation d'un oculographe numérique et économique qui permet d'enregistrer les regards du visiteur lorsqu'il est en train de regarder un tableau. Les matériels utilisés sont une webcam normale de 30 Hz de marque Logitech et un PC avec un microprocesseur Intel PIV. La distance entre le visiteur et l'écran est environ 65-75cm. L'expérimentation se déroule dans une cellule comme un photomaton (Figure 5.11). Nous avons implémenté notre système qui utilise les méthodes présentées dans les chapitres précédents pour localiser les yeux, extraire les caractéristiques d'apparence et estimer la position du regard en coordonnées 2D.

FIGURE 5.11 – Photomaton de l'expérimentation

2. Réalisation d'une interface du système. Nous avons programmé une interface pour effectuer la procédure de façon autonome pour le visiteur, sans intervention d'un expert. Une session dure environ deux minutes par sujet. Le système commence par enregistrer le profil du sujet (sexe, âge). Dès que les yeux du sujet sont bien localisés par la webcam, le système d'oculométrie et la présentation sont lancés en même temps. D'abord le sujet va suivre une phase de calibration des points. Ensuite, les images de deux tableaux s'afficheront successivement sur l'écran. Chaque tableau s'affiche pendant 15 secondes. Une fois que l'affichage des tableaux est terminé, le système va générer les résultats en images pour les montrer au sujet.



(a) (b)

(c) (d)

FIGURE 5.12 – a) Le tableau de Matthieu de Plattemotagne a une structure en trois tiers dans le sens vertical et horizontal. b) La structure du tableau de Léon-Gustave Ravanne en 3 bandes horizontales. c)d) : les résultats du regard du visiteur estimés par notre système. La partie rouge décrit la région explorée par le visiteur.

Un des objectifs est d'analyser l'attention visuelle sur les tableaux par rapport aux profils des sujets. Cette étude est encore au cours. Notre expérimentation a intéressé beaucoup de visiteurs du musée et nous avons obtenu beaucoup de données brutes sur des sujets différents.

La première analyse effectuée est de comparer les fixations du sujet avec la struc-

ture du tableau telle qu'elle décrite par les experts du musée. Pour les deux tableaux retenus dans notre expérimentation, chaque tableau possède sa propre composition, la manière dont l'œuvre est organisée et structurée. Le tableau de Plattemotagne a une structure en 3 tiers dans le sens vertical et horizontal (Figure 5.12 (a)). Les éléments principaux se trouvent sur ces lignes ou à leur intersection. La composition s'appuie sur les obliques formées par les mâts des navires et la masse nuageuse. Ces obliques créent des mouvements ascendants et descendants qui amplifie le mouvement et le déchaînement des eaux. La composition est différente dans le tableau de Ravanne, où l'espace est structuré en 3 bandes horizontales (Figure 5.12 (b)) : le ciel, la côte et la mer. Les seuls éléments verticaux sont les mâts des bateaux. L'impression est plus statique que celle de Plattemotagne. Les résultats (voir Figure 5.12 (c) et (d)) montrent que les explorations visuelles du sujet sur ces deux tableaux sont différentes. Le tableau de Plattemotagne attire le regard dans le creux de la vague, la partie la plus sombre. Dans le tableau de Ravanne, l'horizon stablise et accroche le regard avant qu'il ne plonge dans l'eau. Les explorations visuelles du sujet correspondent assez bien aux structures décrites. Les données seront exploitées à la fin de l'exposition.

5.4.2 Projet Ubiquiet

Ce projet industriel visait à intégrer un oculomètre dans un prototype communicant, destiné à la personne âgée, similaire à celui qui est montré dans la Figure 5.13 gauche. Ce prototype mesure 20cm × 15cm en largeur et hauteur. Ce prototype communicant, développé par la société Ubiquiet, est un produit de technologie gérontologique qui permet d'aider les personnes âgées dans leur vie quotidienne : téléphoner aux proches, écouter la radio, consulter un médecin, etc. Nous voulions intégrer un oculomètre basé sur une webcam pour enregistrer les mouvements oculaires du sujet devant le prototype. L'objectif est de pouvoir étudier les capacités d'attention visuelle chez des personnes âgées, notamment chez les personnes atteintes de la maladie d'Alzheimer.

FIGURE 5.13 – Gauche : Prototype Ubiquiet. Droite : l'installation pour l'expérimentation.

Pour analyser l'attention visuelle du sujet, nous avons intégré 20 LEDs à l'avant de la plaque du prototype pour réaliser les stimulations visuelles (voir la Figure 5.14). Chaque LED peut être programmée pour effectuer des actions différentes : s'allumer, clignoter avec différentes couleurs et pour un temps défini. Dans cette expérimentation

un scénario désigne un planning des séquences des LEDs : couleur et action (allumer, clignoter ou éteindre). Nous avons créé les différents scénari pour notre expérimentation. Par exemple le scénario d'anticipation est un planning qui commence par des séquences répétitives : LED1⟶ LED2⟶ LED3 et finit par un changement : LED1⟶ LED3. Ce qui nous intéresse est de savoir ce que le changement provoque chez les personnes âgées atteintes d'Alzheimer ou en bonne santé. L'idée de l'expérimentation des LEDs provient du résultat des travaux expérimentaux sur les mouvements oculaires chez les personne âgées[Crutcher et al., 2009] [Mosimann et al., 2004].

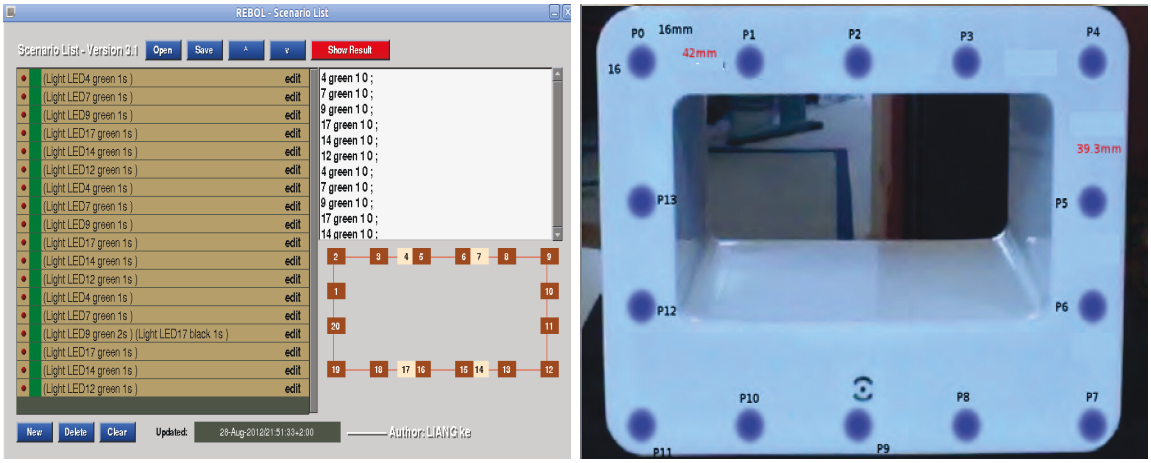


FIGURE 5.14 – Gauche : interface de création des scénari des LEDs. Droite : Les positions des LEDs intégrés dans le prototype.

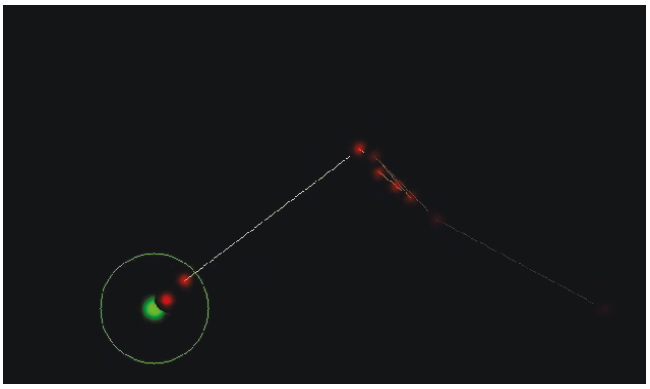


FIGURE 5.15 – Résultat d'estimation de la position du regard en situation de poursuite d'une LED (point vert). Les positions des regards sont indiquées par les points rouges.

L'installation du système est montrée dans la Figure 5.13 droite. Nous utilisons les ports USB pour connecter le prototype et la caméra à distance. L'expérimentation est réalisée dans une salle, en lumière naturelle et sans source infra-rouge. Le comportement du sujet n'est pas gêné pendant l'expérimentation. Les enregistrements du regard (Figure 5.15) nous permettent d'étudier les variables suivantes : le temps de fixation sur chaque LED et la latence qui représente le temps entre le début de l'allumage de LED et la première fixation sur cette LED.

A cause de l'arrêt d'activité de l'entreprise en 2013, nous n'avons pas pu continuer à développer le prototype et à faire des expérimentations sur ce sujet de recherche.

5.4.3 Expérience sur le raisonnement humain

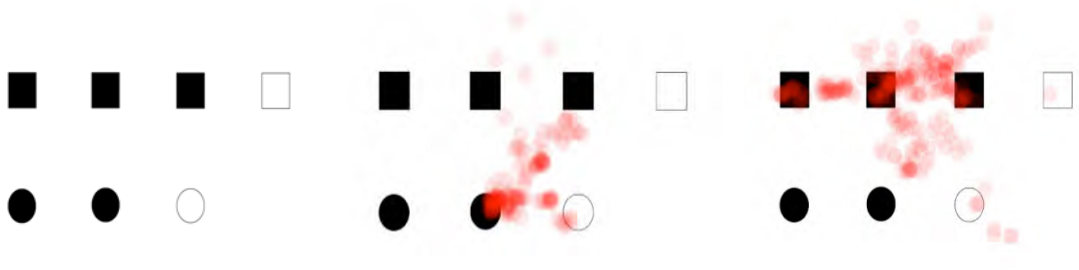
Les approches bayésiennes sont souvent utilisées pour tenter de comprendre le raisonnement humain. De nombreuse études ont ainsi montré que les individus considèrent le conditionnel indicatif du langage naturel "si A alors C" équivalent au conditionnel probabiliste $P(C|A)$[Baratgin and Politzer, 2010][Baratgin et al., 2013]. Politzer, Over et Baratgin[Politzer et al., 2010] ont utilisé le protocole suivant (Figure 5.16) :

1. Un ensemble de sept jetons de couleur blanche ou noire et de forme carrée ou ronde.
2. Le sujet est informé qu'un jeton est tiré au hasard et il doit répondre à la question : Quelle est la chance que la phrase « si le jeton est carré alors il est noir » soit vraie.

Les résultats montrent que une majorité des sujets donne la réponse 3/4 correspondant à la probabilité que le jeton soit noir sachant qu'il est carré. La seconde réponse la plus répandue est 3/7 correspondant à la probabilité que le jeton soit carré et noir.

Notre oculométrie a été utilisée comme outil pour analyser les différences stratégies visuelles correspondant aux différentes réponses à cette tâche.Le matériel utilisé pour l'expérience est : une webcam Microsoft HD (30Hz pour la capture d'image 640×480 pixels), un écran d'ordinateur pour afficher les questions et un ordinateur portable sous Linux Ubuntu 10.10.

L'expérience dure environ 2 minutes. Le sujet suit les instructions apparues sur l'écran et une calibration de 9 points est effectuée. Ensuite la question posée porte soit sur la conjonction « le jeton est carré et noir », ou soit sur la conditionnelle « si le jeton est carré alors il est noir ». Le sujet répond alors à la question. Le système détecte et suit les yeux du sujet tout au long de l'expérience. Dans la question conjonction, chez les sujets donnant la réponse conjonctive on observe, une stratégie de vision globale (Figure 5.16 b). Le sujet parcourt de manière très générale l'ensemble de jetons d'où un résultat d'un regard très centré sur l'image qui traduit un visu avec un champ élargi. En revanche, dans la question conditionnelle, pour les sujets donnant la réponse P(Noir/Carré), on observe une concentration du regard sur les jetons carrés (Figure 5.16 c).



(a) (b) (c)

FIGURE 5.16 – a) Les jetons de couleur blanche ou noire et de forme carrée ou ronde montrés dans l'expérience. b) le regard détecté pour ceux qui répondent 3/7 c)le regard détecté pour ceux qui répondent 3/4.

5.4.4 Système de commande par les yeux

Notre modèle de *catégorisation* permet de reconnaitre cinq types de mouvements des yeux : DH, DB, GH, GB et CL, qui sont utilisables pour les applications d'IHM. Nous présentons la réalisation d'un système de commande qui permet d'utiliser les cinq types de mouvements pour déclencher ou contrôler un événement. Dans cette expérimentation, nous utilisons les cinq types de mouvements en tant que signaux pour contrôler en temps réel les comportements d'un personnage de dessin animé comme la jeune fille dans la Figure 5.17. L'objectif est de créer un film sur ce personnage avec des comportements naturels comme : fermer les yeux, parler, sourire, être en colère et être triste.

FIGURE 5.17 – Démonstration des comportements d'un personnage (une jeune fille) : fermer les yeux, parler, sourire, être en colère et être triste. (*Source d'images : www.shutterstock.com*)

Cette expérimentation a été réalisée avec un portable Macbook Pro 8.1 dont le processeur est un Intel Core i5-2415M. Nous utilisons la webcam Microsoft LifeCam HD-5000 dont la fréquence est 30 Hz pour une résolution de 640×480 pixels. La phase de localisation de l'œil nous fournit une image de 60×40 pixels sur la région de l'œil. La calibration nécessite quatre points qui s'affichent aux quatre coins de l'écran. Nous pouvons simplifier la procédure de la calibration en demandant au sujet de regarder lui-même les quatre coins de l'écran sans afficher les points à l'écran.

FIGURE 5.18 – Démonstration du modèle pour créer une séquence des comportements du personnage en temps réel avec les cinq signaux : DH→parler, DB→être triste, GH→sourire, GB→être en colère et CL→fermer les yeux.

La Figure 5.18 démontre l'interface et le résultat de notre expérimentation pour créer

une séquence des comportements du personnage en temps réel avec les cinq signaux : DH, DB, GH, GB et CL. Le signal CL contrôle la fermeture des yeux du personnage. Le DH fait parler le personnage. Le GH fait sourire le personnage. Le GB et DB montrent que le personnage est en colère ou triste. Le résultat nous montre des réactions naturelles du personnage de dessin animé dans le film créé.

5.5 Conclusion

Dans ce chapitre nous avons présenté deux modèles qui estiment le regard soit en coordonnées 2D soit en classes. Pour estimer la position du regard dont la valeur est $\mathcal{Y} \in \mathbb{R}$, nous avons proposé une méthode qui met en jeu la régression du processus gaussien et qui utilise un ensemble de données d'apprentissage semi-supervisé. Les expérimentations nous ont montré que cette méthode permet de fournir une précision inférieure au degré pour l'estimation du regard en n'utilisant QUE cinq points de calibration.

Nous proposons également un modèle qui est conçu pour détecter les mouvements spécifiques des yeux comme le clignement et les fixations sur des régions précises. Ce modèle applique la classification spectrale pour classifier les points de la variété d'un ensemble d'images de l'œil. L'expérimentation a montré que l'efficacité de la méthode permet son utilisation pour les applications d'IHM en temps réel.

Sixième partie

Conclusion générale et perspectives

Durant cette thèse, nous avons réalisé un modèle d'oculométrie non-intrusif pour estimer le regard à partir d'une webcam sans lumière infra-rouge. Le défi de notre travail était de développer les méthodes efficaces et robustes pour non seulement localiser l'œil dans la séquence d'images capturées par la webcam, mais également estimer la position du regard sans extraire les caractéristiques explicites comme la pupille ou le reflet cornéen, qui sont utilisées dans les modèles conventionnels des oculomètres. Nous avons réalisé un système oculométrique économique et opérationnel. Nos contributions portent sur le développement des méthodes pour les quatre modules qui constituent notre système : l'extraction des caractéristiques des yeux, la détection et le suivi des yeux, l'analyse de variété d'un ensemble d'images des yeux et l'estimation de la position du regard.

Différente des autres techniques d'**extraction des caractéristiques des yeux**, cette oculométrie numérique à distance utilise un modèle d'apparence pour réduire la dimensionnalité de l'image de l'œil au niveau local en préservant les caractéristiques de l'apparence de l'image. La méthode proposée utilise les motifs binaires locaux centrés-symétriques (CS-LBP) pour caractériser le motif local autour de chaque pixel de l'image. Le vecteur de caractéristique obtenu est utilisé non seulement pour discriminer l'œil et d'autres objets pendant la phase de localisation des yeux, mais également pour distinguer les différents mouvements oculaires durant la phase d'estimation du regard. De plus, ce vecteur est de faible dimension, ce qui permet d'optimiser le calcul. Ces caractéristiques traitées par ce modèle d'apparence peuvent être utilisées pour la localisation de l'œil et l'estimation du regard.

Nous avons proposé une méthode hybride et efficace pour **la détection et le suivi des yeux** dans la séquence d'images capturées par la webcam. Nous avons appliqué un modèle à formes actives (ASM) et une carte des yeux (EyeMap) dans la première image pour localiser les yeux. Une fois la localisation faite, le filtre particulaire est utilisé pour suivre le déplacement de l'œil dans les images suivantes selon un algorithme stochastique. Cette méthode permet de détecter et suivre les yeux plus efficacement et de rélocaliser rapidement les yeux quand ils sont perdus à cause des mouvements du sujet. En outre, cette méthode probabiliste est plus efficace du point de vue temps de calcul, même si nous devions localiser l'œil dans une séquence d'images de haute résolution.

Nous avons également introduit **une technique de réduction de la dimensionnalité non-linéaire** pour analyser les mouvements oculaires selon les variétés (manifolds) d'un ensemble d'images de l'œil. Appliquer cette technique nous permet de déterminer la structure intrinsèque qui décrit la variation des mouvements oculaires. Notre expérimentation montre que la variété en 3D de ces images peut fournir des informations utiles sur la nature et l'organisation des données et être exploitée en tâches de classification ou de regroupement, ce que nous avons fait dans la phase de calibration pour estimer le regard. Concrètement l'analyse de la variété contribue à réaliser une calibration automatique qui est cruciale pour le fonctionnement du module d'estimation du regard.

Pour **l'estimation de la position regard**, nous avons proposé deux méthodes différentes d'apprentissage supervisé : une méthode de régression par processus gaussien pour estimer le regard en coordonnées 2D ; une méthode de catégorisation qui utilise la classification spectrale pour classifier le regard dans les classes définies correspondant à certains types des mouvement oculaires prédéfinis.

Les expérimentations nous ont montré que, avec une simple webcam, notre système oculométrique permet **d'obtenir une précision inférieure au degré en temps réel en n'utilisant que cinq points de calibration**. De plus, notre système peut être utilisé efficacement pour les applications d'IHM en temps réel, par exemple pour un système de commande oculaire. Durant cette thèse, nous avons réalisé plusieurs applications en utilisant notre modèle d'oculométrie numérique économique. Les résultats sont satisfaisants et encourageants. Par conséquent nous continuons d'améliorer le système et de l'utiliser pour les applications dans les domaines différents.

Notre système oculométrique va être utilisé dans un projet ambitieux avec l'université fédérale de São Paulo (UNIFESP) au Brésil : «ICM-eTrack». L'objectif du projet est de développer un système de navigation pour personne à mobilité réduite utilisant conjointement une interface cerveau machine (ICM) et un système d'oculométrie. Les dispositifs actuels permettant le déplacement de personnes à mobilité réduite de manière autonome sont réservés à celles conservant une mobilité des membres supérieurs suffisante pour pousser un fauteuil manuel ou activer une commande de fauteuil électrique. Par contre les personnes souffrant de pathologie tel qu'une tétraplégie haute ou un locked-in syndrome, ne sont pas en mesure d'utiliser la technologie aujourd'hui. C'est pourquoi nous souhaitons de développer de nouveaux systèmes de pilotage d'effecteur tel qu'un fauteuil roulant électrique. Le projet consiste à piloter un fauteuil roulant par le regard, avec un système de confirmation, par le bias d'une interface cerveau-machine (ICM) basé sur la mesure de signaux d'électroencéphalogramme (Figure[6.1]). Une analyse de l'espace environnant est réalisée grâce à l'image obtenue par caméra. L'utilisation de notre modèle d'oculométrie permet de déterminer le point cible où souhaite se rendre l'utilisateur. Ce dernier doit valider, via l'ICM, que la zone détectée est bien celle où il souhaite se rendre et que le chemin proposé pour s'y rendre est convenable.

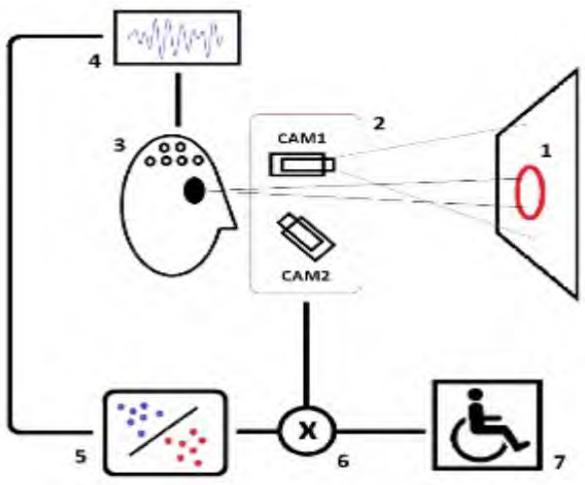


FIGURE 6.1 – Schéma général de l'interface cerveau-machine (ICM) en interaction avec le système oculométrique. 1) Point cible, destination souhaitée. 2) Acquisition de l'image de l'oculomètre. 3-4) Acquisition et prétraitement en temps réel du signal d'électroencéphalogramme. 5) Interprétation des modèles par classification binaire pour la prise de décision et l'actionnement du fauteuil roulant en direction du point cible. 6)Acceptation ou non via le système d'ICM. 7) Mise en action du fauteuil roulant électrique.

Notre modèle oculométrique sera également utilisé dans un projet novateur TBH (Tableau de Bord de l'Habitat) Alliance, dont les collaborateurs sont les entreprises ARCHOS, CGI, EcoCO2/QUARTUM, Fludia, SPLV Analytics et le laboratoire CHArt/Lutin Userlab. Le Projet TBH Alliance va recruter un panel de consommateurs pour tester la visualisation de la consommation d'électricité en temps réel sur une ARCHOS Smart Home (Figure 6.2) et fournir des analyses et conseils dans le but de mettre en évidence le potentiel d'économies d'énergie. Ce projet permettra l'étude du potentiel d'économie d'énergie de divers dispositifs d'affichage et d'accompagnement des ménages autour de leur consommation d'électricité sur un panel représentatif de 3200 consommateurs français. Dès le début de cette étude, la tablette ARCHOS Smart Home sera fournie à 3200 ménages. Elle sera accompagnée de capteurs de température, d'hygrométrie et d'un lecteur optique de compteur. Ainsi, les consommateurs pourront visualiser et analyser leurs consommations d'énergie. Notre modèle qui utilise la caméra de la tablette permet d'enregistrer en temps réel le regard du consommateur quand il visualise les interfaces de la tablette. Les informations oculaires nous permettent d'étudier l'ergonomie liée à l'utilisation des applications sur tablette.

FIGURE 6.2 – Tablette ARCHOS Smart Home

Par rapport aux perspectives du développement, notre système oculométrique peut être amélioré sur les deux parties : des composants du système et des algorithmes. La qualité des composants, les méthodes de traitement d'images et d'extraction des caractéristiques sont les grands enjeux pour le développement du système oculométrique qui est fondé sur un modèle d'apparence des images. Par rapport aux matériels, la caméra numérique est un composant important comme source des images. Elle continue de s'évoluer sans cesse et devient de plus en plus accessible en prix et en qualité. Par exemple, *Thorlabs* a développé des caméras CCD ou CMOS 1.4 megapixels (1392×1040) avec une fréquence de 20 à 40 fps adaptées au traitement d'images. Une telle caméra de haute résolution pourrait fournir à notre système plus d'informations sur la région de l'œil. De plus, la caméra peut être de plus en plus "intelligente" grâce au développement de DSP capables implémenter directement dans le processus les algorithmes que nous avons développés.

Pour des algorithmes, une des difficultés importantes est d'estimer le regard lorsque le sujet bouge la tête : l'apparence de l'œil change au moment où la tête bouge. Pour améliorer, nous souhaitons trouver la relation entre le degré du changement de la position

de l'œil et le degré du changement du regard. Ce problème est un défi pour les méthodes d'estimation de la position du regard fondées sur l'apparence d'image. En outre, nous allons chercher des méthodes plus efficaces que CS-LBP et SURF pour l'extraction des caractéristiques. La caractéristique FREAK[Ortiz, 2012] peut être une solution possible. Les autres variantes de LBP peuvent également contribuer à l'amélioration du système.

En résumé, nous avons réussi à réaliser un système oculométrique opérationnel avec une simple webcam, grâce à l'ensemble des méthodes que nous avons développées. Ce système économique est facile pour l'installation et la configuration, moins exigeant pour le choix des composants et plus accessible pour l'utilisateur. De plus, il permet d'estimer la position du regard avec une précision inférieure au degré en temps réel en n'utilisant que cinq points de calibration. Ces avantages lui ont permis d'être adapté et utilisé pour les applications diverses. Nous souhaitons que le système continue d'être amélioré pour rendre l'oculométrie numérique plus flexible, plus accessible et plus performante.

Annexes

Annexe 1 Angle visuel

La notion d'angle visuel est utilisée fréquemment en optique physiologique et en oculométrie. L'utilisation de l'angle visuel permet de s'affranchir de la distance entre l'œil et la cible observée : une pièce d'un diamètre de 1 cm vue à une distance d'1 mètre sera imagée sur la rétine avec une taille identique à celle d'une pièce d'un diamètre de 2 cm vue à une distance de 2 mètres.

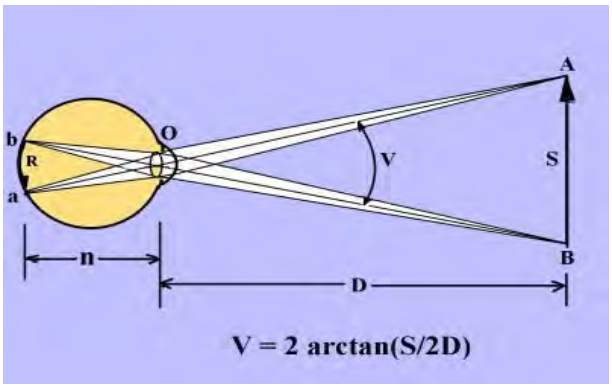


FIGURE 7.1 – L'angle visuel V. (source d'image : wikipédia)

L'angle visuel peut être exprimé dans des unités diverses : radians, degrés, ou minutes et secondes d'arc.

$$1^{\circ}(degree) = \frac{\pi}{180}(radians)$$
$$1^{\circ}(degree) = 60^{'}(minutes)$$
$$1^{\circ}(degree) = 3600^{''}(secondes)$$

L'angle visuel V (Figure 7.1) est calculé par [Holmqvist et al., 2011] :

$$V = 2 \times arctan(\frac{S}{2D})$$

où S est la distance entre A et B, D est la distance focale. Avec une longue distance D, l'angle étant petit, on peut approcher le résultat par :

$$V = arctan(\frac{S}{D})$$

Généralement 1° est équivalent à 1 cm (soit $S = 1$ cm) quand la distance $D \approx 57.3$ cm.

Annexe 2 Distance et produit scalaire

La distance euclidienne est un moyen intuitif et habituel pour mesurer la distance entre deux vecteurs de nombres. Soient x et y deux vecteurs de $\mathcal{R}^d$, le carré de la distance

euclidienne est simplement :

$$D^2_{eucl}(x,y) = \sum_{i=1}^{d}(x_i - y_i)^2$$

u_i désignant la i-ème composante du vecteur u. On le voit, il s'agit simplement de la somme des différences au carré. La distance euclidienne est intimement liée à la notion de produit scalaire. Le produit scalaire entre x et y, noté $< x, y >$ (ou parfois $x \cdot y$ ou encore $x^T y$ en considérant les vecteurs comme des matrices à une colonne) est défini par :

$$< x, y >= \sum_{i=1}^{d} x_i y_i$$

et il est facile de montrer que la distance euclidienne peut se réécrire :

$$D^2_{eucl}(x,y) =< x, x > + < y, y > -2 < x, y >$$

La norme (euclidienne) d'un vecteur $\| x \|$ est définie par :

$$\| x \|^2 =< x, x >$$

De nombreux algorithmes d'analyse de données ou d'apprentissage sont basés sur ces notions : les k-moyennes, les machines à vecteurs supports, la régression linéaire, l'analyse en composantes principales sont des exemples de méthodes reposant sur la notion de distance ou de produit scalaire.

Annexe 3 Interpolation bilinéaire

L'interpolation bilinéaire est une méthode d'interpolation pour les fonctions de 2 variables sur une grille régulière. C'est une méthode simple pour éliminer le phénomène d'aliasing. On utilise donc les 4 points les plus proches des coordonnées calculées dans l'image source en les pondérant par des coefficients inversement proportionnels à la distance et dont la somme vaut 1 (voir la Figure 7.2). Le poids affecté à chaque point

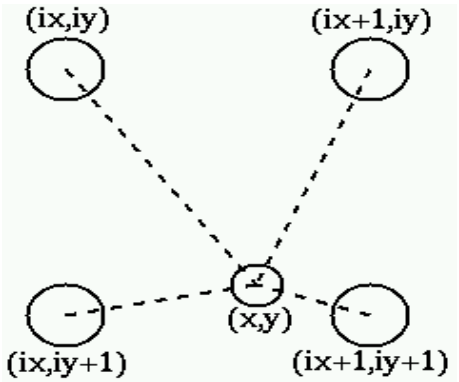


FIGURE 7.2 – Le calcul le point (x, y) autour des 4 points par l'interpolation bilinéaire

est :

$$p(x,y) = \frac{1}{\sqrt{(x-ix)^2 + (y-iy)^2}}$$

La valeur du point (x, y) par interpolation sur les 4 points les plus proches est :

$$v(x,y) = \frac{p(ix,iy)v(ix,iy) + p(ix+1,iy)v(ix+1,iy) + p(ix,iy+1)v(ix,iy+1) + p(ix+1,iy+1)v(ix+1,iy+1)}{p(ix,iy) + p(ix,iy+1) + p(ix+1,iy) + p(ix+1,iy+1)}$$

Annexe 4 — k plus proches voisins

L'algorithme des k plus proches voisins (k-ppv) est un algorithme non paramétrique utilisé pour la régression et la classification. Etant donnée une mesure de distance dans l'espace d'entré $\mathbb{R}^d$ (souvent prise comme la distance Euclidienne), la prédiction du modèle sur un exemple de test $x \in \mathcal{X}$ dépend uniquement des k plus proches voisins de x dans l'ensemble d'apprentissage $\mathcal{D}$. En notant $i_1(x), ..., i_k(x)$ les indices des k exemples de $\mathcal{D}$ les plus proches de x selon la distance choisie, la prédiction du modèle en régression est la moyenne des étiquettes observées ches ces k voisins :

$$f(x) = \frac{1}{k} \sum_{j=1}^{k} y_{i_j(x)}$$

et en classification il s'agit d'un vote parmi les k voisins :

$$f(x) = argmax_y \sum_{j=1}^{k} \mathbf{1}_{y=y_{i_j(x)}}$$

où en cas d'égalité parmi les votes le modèle choisit aléatoirement l'une des classes majoritaires. la classification par les k plus proches voisins est illustré en figure 7.3.

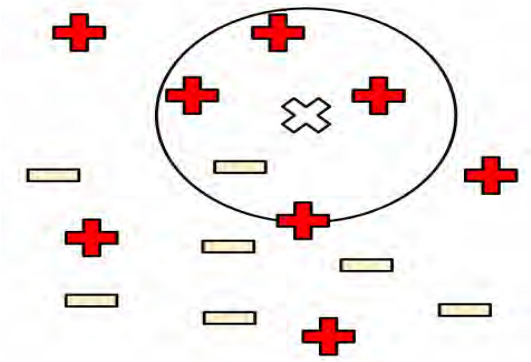


FIGURE 7.3 – k plus proches voisins (k=5, tâche de classification) : pour classifier un nouvel exemple (le $\times$ blanc) on cherche ses 5 plus proches voisins dans l'ensemble d'apprentissage (à l'intérieur du cercle), et on compte le nombre d'exemples de chaque classe. Il y a ici 4 exemples de la classe + (rouge) et 1 de la classe − (jaune) donc ce nouvel exemple $\times$ sera classifié comme +.

Annexe 5 — Méthode Monte-Carlo

Remarquons que si l'on cherche une représentation fidèle des phénomènes observés, on est rapidement confronté à des difficultés dues aux calculs non explicites. Les

techniques de simulation aléatoire vont nous permettre d'approcher numériquement ces calculs. La méthode de *Monte-Carlo* désigne toute méthode visant à calculer une valeur numérique par un procédé aléatoire. Cette méthode peut servir pour :

- le calcul d'intégrale ;
- la résolution d'équations aux dérivées partielles ;
- la résolution de système linéaire ;
- la résolution de problèmes d'optimisation.

Considérons par exemple le problème de l'intégration numérique. Il s'agit d'approcher

$$\mathcal{I} = \int_0^1 f(x)dx.$$

La méthode de Monte-Carlo consiste à écrire cette intégrale sous la forme :

$$\mathcal{I} = \mathbb{E}[f(U)]$$

où U est une variable aléatoire suivant une loi uniforme sur [0, 1] et à utiliser la loi des grands nombres : si $(U_i)_{i\in N}$ est une suite de variables aléatoires indépendantes et de loi uniforme sur [0, 1], alors

$$\frac{1}{n}\sum_{i=1}^{n} f(U_i) \quad \rightarrow \quad \mathbb{E}[f(U)]$$

En d'autres termes, si U_1, U_2,...,U_n sont des nombres tirés au hasard dans [0, 1], $\frac{1}{n}(f(U_1) + f(U_2) + ... + f(U_n))$ est une approximation de $\int_0^1 f(x)dx$.

Annexe 6 CS-LBP

Classe CS-LBP :

```
1.  template <typename _Tp>
2.  void feature : : CSLBP_ (const Mat& src, Mat& dst, int radius, int neighbors) {
3.
4.    neighbors = max( min( neighbors,31 ), 1) ;
5.    dst = Mat : :zeros(src.rows - 2*radius, src.cols - 2*radius, CV_32SC1) ;
6.    for(int n=0 ; n<neighbors/2 ; n++) {
7.      // premier point
8.      float x = static_cast<float>( radius )
9.                * cos( 2.0*M_PI*n/static_cast<floa t>(neighbors)) ;
10.     float y = static_cast<float>( radius )
11.               * -sin(2.0*M_PI*n/static_cast<float>(neighbors)) ;
12.
13.     int fx = static_cast<int>(floor(x)) ;
14.     int fy = static_cast<int>(floor(y)) ;
15.     int cx = static_cast<int>(ceil(x)) ;
16.     int cy = static_cast<int>(ceil(y)) ;
17.
18.     float ty = y - fy ;
```

```
19.         float tx = x - fx;
20.         // poids
21.         float w1 = (1 - tx) * (1 - ty);
22.         float w2 = tx * (1 - ty);
23.         float w3 = (1 - tx) * ty;
24.         float w4 = tx * ty;
25.         //deuxième point
26.         float x0 = static_cast<float>( radius )
27.                   * cos(2.0*M_PI*(n+4)/static_cast<float>(neighbors));
28.         float y0 = static_cast<float>( radius )
29.                   * -sin(2.0*M_PI*(n+4)/static_cast<float>(neighbors));
30.
31.         int fx0 = static_cast<int>(floor(x0));
32.         int fy0 = static_cast<int>(floor(y0));
33.         int cx0 = static_cast<int>(ceil(x0));
34.         int cy0 = static_cast<int>(ceil(y0));
35.
36.         float ty0 = y0 - fy0;
37.         float tx0 = x0 - fx0;
38.         // poids
39.         float w10 = (1 - tx0) * (1 - ty0);
40.         float w20 = tx0 * (1 - ty0);
41.         float w30 = (1 - tx0) * ty0;
42.         float w40 = tx0 * ty0;
43.         // traitement itératif
44.         for(int i=radius; i < src.rows-radius;i++) {
45.         for(int j=radius;j < src.cols-radius;j++) {
46.         float t = w1*src.at<_Tp>(i+fy, j+fx) + w2*src.at<_Tp>(i+fy, j+cx)
47.                   + w3*src.at<_Tp>(i+cy,j+fx) + w4*src.at<_Tp>(i+cy,j+cx);
48.         float t0 = w10*src.at<_Tp>(i+fy0,j+fx0) + w20*src.at<_Tp>(i+fy0,j+cx0)
49.              +w30*src.at<_Tp>(i+cy0, j+fx0)+w40*src.at<_Tp>(i+cy0, j+cx0);
50.
51.         dst.at<unsigned int>(i-radius,j-radius) +=
52.           ((t > t0) && (abs(t-t0) > std : :numeric_limits<float> : :epsilon())) « n;
53.           }
54.       }
55.   }
56. }
```

Annexe 7 EyeMap

Classe EyeMap :

```
1.    cv : :Point EyeMap(IplImage* source, int width, int height, int x0, int y0)
2.    {
3.      cv : :Point p;
```

```
4.    int x,y;
5.    float Y,Cr1,Cb,Cr,R, G, B, eye;
6.    CvScalar s, s1, s2;
7.    IplImage* Y_map = cvCreateImage(cvGetSize(source),IPL_DEPTH_32F, 1);
8.    IplImage* eyemap = cvCreateImage(cvGetSize(source),IPL_DEPTH_32F, 1);
9.    IplImage* Y_dilate = cvCreateImage(cvGetSize(source),IPL_DEPTH_32F, 1);
10.    IplImage* Y_erode =cvCreateImage(cvGetSize(source),IPL_DEPTH_32F,1);
11.    IplImage* final = cvCreateImage(cvGetSize(source),IPL_DEPTH_32F, 1);
12.    for( x = 0; x < source->width; x++)
13.      {for(y = 0; y < source->height; y++) {
14.        s = cvGet2D( source, y, x );
15.        R = (float)s.val[ 2 ];
16.        G = (float)s.val[ 1 ];
17.        B = (float)s.val[ 0 ];
18.        Y = 0.257 * R + 0.504 * G + 0.098 * B + 16;
19.        Cb = -0.148 * R - 0.291 * G + 0.439 * B + 128;
20.        Cr = 0.439 * R - 0.368 * G - 0.071 * B + 128;
21.        s.val[ 2 ] = Y;
22.        s.val[ 1 ] = Cb;
23.        s.val[ 0 ] = Cr;
24.        Cr1 = 255 - Cr;
25.        s1.val[ 0 ] = (float) 1/3*(( Cb * Cb ) + ( Cr1 * Cr1 )+( Cb/Cr ));
26.        cvSet2D( eyemap, y, x, s1 );
27.        s2.val[0] = Y;
28.        cvSet2D( Y_map, y, x, s2 ); }
29.      }
30.    cvNormalize( eyemap, eyemap, 0, 255, CV_MINMAX );
31.    cvNormalize( Y_map, Y_map, 0, 255, CV_MINMAX );
32.    IplConvKernel* element =
33.        cvCreateStructuringElementEx( 5, 5, 2, 2, CV_SHAPE_RECT, 0 );
34.    cvDilate( Y_map, Y_dilate, element, 1 );
35.    cvErode( Y_map, Y_erode, element, 1 );
36.    float sommeIris = 0;
37.    int xIris = 0, yIris = 0;
38.    for( int j = y0+RADIUS; j < (y0 + height-RADIUS); j++)
39.      for( int i = x0+RADIUS; i < (x0+width-RADIUS); i++) {
40.        float somme = 0;
41.        int number = 0;
42.        for(int jcircle = ( j - RADIUS ); jcircle < (j + RADIUS ); jcircle++)
43.          for(int icircle = ( i - RADIUS ); icircle < ( i + RADIUS ); icircle++) {
44.            if(InRadius(i,j,icircle,jcircle)) {
45.              number++;
46.              s2 = cvGet2D( final, jcircle, icircle);
47.          somme = somme + (float)s2.val[ 0 ]; }
48.      }
49.      somme = somme / number;
50.      if( somme > sommeIris )
51.      { sommeIris = somme; xIris = i, yIris = j;}
```

```
52.     p.x=xIris ;p.y=yIris ;
53.     return p ;
54.   }
```

Annexe 8 Laplacian Eigenmaps et Diffusion Maps

Classe Laplacian Eigenmaps :

```
1.   Matrix ObjetManifold : :laplacianEigenmaps( double sigma, double epsilon )
2.   {
3.     Matrix K( objetimages.size(), objetimages.size() ) ;
4.     for( int i = 0 ; i < objetimages.size() ; i++ )
5.       for( int j = 0 ; j < objetimages.size() ; j++ ) {
6.       double t = manifoldcovariance( objetimages[ i ],
7.                                   objetimages[ j ], sigma, epsilon ) ;
8.       K(i,j) =t ; }
9.     Matrix D( objetimages.size(), objetimages.size() ) ;
10.     Matrix I( objetimages.size(), objetimages.size() ) ;
11.     I.set_identity() ;
12.     D.fill(0) ;
13.     Matrix P( objetimages.size(), objetimages.size() ) ;
14.     for( int i = 0 ; i < objetimages.size() ; i++ )
15.       { double som = 0 ;
16.         for( int j = 0 ; j < objetimages.size() ; j++ )
17.           som = som + K( i, j ) ;
18.
19.         D( i, i ) = som ; }
20.     // calculer la matrice du Laplacien
21.     P=D-K ;
22.     // résolution du système P*y=lambda*D*y
23.     vnl_generalized_eigensystem test=vnl_generalized_eigensystem(P,D) ;
24.     return P ;
25.     }
```

Classe Diffusion Maps :

```
1.   Matrix ObjetManifold : :diffusionmaps( double sigma, double epsilon )
2.   {
3.     Matrix K( objetimages.size(), objetimages.size() ) ;
4.     for( int i = 0 ; i < objetimages.size() ; i++ )
5.       for( int j = 0 ; j < objetimages.size() ; j++ ) {
6.       double t = manifoldcovariance( objetimages[ i ],
7.                                   objetimages[ j ], sigma, epsilon ) ;
```

```
8.        K(i,j) =t ; }
9.      Matrix D( objetimages.size(), objetimages.size() ) ;
10.     Matrix I( objetimages.size(), objetimages.size() ) ;
11.     I.set_identity() ;
12.     D.fill(0) ;
13.     Matrix P( objetimages.size(), objetimages.size() ) ;
14.     for( int i = 0 ; i < objetimages.size() ; i++ )
15.       { double som = 0 ;
16.         for( int j = 0 ; j < objetimages.size() ; j++ )
17.           som = som + K( i, j ) ;
18.
19.         D( i, i ) = som ; }
20.     Matrix D2=vnl_matrix_inverse<double>(D) ;
21.     P=D2*K ;
22.     K=P*P*P*P ;
23.     Matrix K_new=K, V, X_ ;
24.     Vector Lambda ;
25.     // résolution du système K_new*y=lambda*y
26.     vnl_symmetric_eigensystem_compute(K_new,V,Lambda) ;
27.     for( int i = 0 ; i < objetimages.size() ; i++ )
28.       {for( int j = 1 ; j <4 ; j++ ) {
29.         double v=V(i,objetimages.size()-j-1)*
30.                     Lambda[objetimages.size()-j-1] ;
31.         myfile2«v«" " ; }
32.       myfile2«endl ; }
33.     return V ; }
```

Références

[Ahonen et al., 2006] Ahonen, T., Member, S., Hadid, A., Pietikäinen, M., and Member, S. (2006). Face description with local binary patterns : Application to face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28 :2037–2041.

[Arulampalam et al., 2002] Arulampalam, M. S., Maskell, S., et al. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE TRANSACTIONS ON SIGNAL PROCESSING.

[Baluja and Pomerleau, 1994] Baluja, S. and Pomerleau, D. (1994). Non-intrusive gaze tracking using artificial neural networks. Advances in Neural Information Processing Systems.

[Baratgin et al., 2013] Baratgin, J., Over, D., and Politzer, G. (2013). Uncertainty and the finetti tables. Thinking & Reasonning, pages 308–328.

[Baratgin and Politzer, 2010] Baratgin, J. and Politzer, G. (2010). Updating : A psychological basic situation of probability revision. Thinking & Reasonning, pages 253–287.

[Bay et al., 2006] Bay, H., Tuytelaars, T., and Gool, L. V. (2006). Surf : Speeded up robust features. In In ECCV, pages 404–417.

[Belkin and Niyogi, 2001] Belkin, M. and Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. NIPS, 15(6) :1373–1396.

[Belkin and Niyogi, 2003] Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput., 15(6) :1373–1396.

[Besse and Laurent, 2014] Besse, P. and Laurent, B. (2014). Apprentissage statistique : modélisation, prévision et data mining.

[Blake et al., 1998] Blake, A., ISARD, M., et al. (1998). Active contours. Springer.

[Bonnet et al., 1989] Bonnet, C., Ghiglione, R., and Richard, J.-F. (1989). Traité de psychologie cognitive 1. DUNOD.

[Cadavid et al., 2009] Cadavid, S., Mahoor, M. H., Messinger, D. S., and Cohn, J. F. (2009). Automated classification of gaze direction using spectral regression and support vector machine. International Conference on Affective Computing and Intelligent Interaction (ACII), pages 1–6.

[Chen et al., 2005] Chen, J., Wang, R., Yan, S., Shan, S., Chen, X., and Gao, W. (2005). How to train a classifier based on the huge face database. In IEEE International Workshop on AMFG2005, LNCS 3723, pages 84–95.

[Coifman and Lafon, 2006] Coifman, R. R. and Lafon, S. (2006). Diffusion maps. Applied and Computational Harmonic Analysis.

[Cornsweet et al., 1973] Cornsweet, T., H.D.Crane, et al. (1973). Accurate two-dimensional eye tracker using first and fourth purkinje images. Journal of the optical society of america.

[Crutcher et al., 2009] Crutcher, M. D., Calhoun-Haney, R., Manzanares, C. M., et al. (2009). Eye tracking during a visual paired comparison task as a predictor of early dementia. American Journal of Alzheimerâs Disease & Other Dementias.

[Delabarre, 1898] Delabarre, E. B. (1898). A method of recording eye-movements. American Journal of Psychology, pages 572–574.

[Desbois, 2005] Desbois, D. (2005). Une introduction au positionnement multidimensionnel. Modulad, pages 1–28.

[Djuric and Godsill, 2002] Djuric, P. M. and Godsill, S. J. (2002). Special issue on Monte Carlo Methods for Statistical Signal Processing. IEEE Transactions on Signal Processing.

[Dodge et al., 1901] Dodge, R., Cline, T. S., et al. (1901). The angle velocity of eye movements. Psychological Review, (8) :145–157.

[Donath and Hoffman, 1973] Donath, W. E. and Hoffman, A. J. (1973). Lower bounds for the partitioning of graphs. IBM J. Res. Dev., 17(5) :420–425.

[Dore et al., 2008] Dore, A., Beoldo, A., and Regazzoni, C. S. (2008). Multiple cue adaptive tracking of deformable objects with particle filter. ICIP(International Conference on Image Processing).

[Doucet and Johansen, 2008] Doucet, A. and Johansen, A. M. (2008). A tutorial on particle filtering and smoothing : Fifteen years later.

[Duchowski, 2006] Duchowski, A. (2006). Eye Tracking Methdology Theory and Practice. Springer.

[Fukuda et al., 2011] Fukuda, T., Morimoto, K., and Yamana, H. (2011). Model-based eye-tracking method for low-resolution eye-images. 2nd Workshop on Eye Gaze in Intelligent Human Machine Interaction.

[Fukunaga, 1990] Fukunaga, K. (1990). Introduction to statistical pattern recognition. Academic Press.

[Gordon et al., 1993] Gordon, N. J., Salmond, D. J., and Smith, A. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. IEEE TRANSACTIONS.

[Hammoud, 2008] Hammoud, R. I. (2008). Passive Eye Monitoring. Springer-Verlag.

[Hansen and Ji, 2010] Hansen, D. W. and Ji, Q. (2010). In the eye of the beholder : A survey of models for eyes and gaze. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 32(3) :478–500.

[Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In In Proc. of Fourth Alvey Vision Conference, pages 147–151.

[Heikkilä et al., 2009] Heikkilä, M., Pietikäinen, M., and Schmid, C. (2009). Description of interest regions with local binary patterns. Pattern Recogn., 42(3) :425–436.

[Hess and Fern, 2009] Hess, R. and Fern, A. (2009). Discriminatively trained particle filters for complex multi-object tracking. CVPR 09.

[Holmqvist et al., 2011] Holmqvist, K. et al. (2011). Eye tracking - A comprehensive guide to methods and measures. Hardback.

[Hong et al., 2007] Hong, H., Pansing, C., and Rolland, J. (2007). Modeling of an eye-imaging system for optimizing illumination schemes in an eye-tracked head-mounted display. Applied Optics.

[Hsu et al., 2002] Hsu, R.-L., Abdel-Mottaleb, M., and Jain, A. K. (2002). Face detection in color images. IEEE TRANSACTIONS.

[Huey, 1898] Huey, E. B. (1898). Preliminary experiments in the physiology and psychology of reading. American Journal of Psychology, 9 :575–586.

[Imbert, 2001] Imbert, M. (2001). La rétine et son fonctionnement. Science Et Vie, (216).

[Isard et al., 1998] Isard, M., Andrew, B., et al. (1998). Condensation - conditional density propagation for visual tracking. International Journal of Computer Visio.

[Jouen et al., 1995] Jouen, F., BAUDONNIERE, P.-M., et al. (1995). Dispositif de controle des mouvements oculaires. Brevet Internationale WO95/31927.

[Juan et al., 2010] Juan, L. et al. (2010). A comparison of sift, pca-sift and surf. In In IJIP, pages 404–417.

[Kalman, 1960] Kalman, R. (1960). A new approach to linear filtering and prediction problems. Transactions of the ASME - Journal of Basic Engineering Vol. 82.

[Kassner and Patera, 2012] Kassner, M. P. and Patera, W. R. (2012). Pupil : Constructing the space of visual attention. Master's thesis, MIT.

[Laganiere, 2011] Laganiere, R. (2011). OpenCV 2 Computer Vision Application Programming Cookbook. PACKT.

[Le et al., 2013] Le, H., Dang, T., and Liu, F. (2013). Eye blink detection for smart glasses. IEEE ISM 2013.

[Lienhart and Maydt, 2002] Lienhart, R. and Maydt, J. (2002). An extended set of haar-like features for rapid object detection. In IEEE ICIP 2002, pages 900–903.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. Journal of Computer Vision, pages 91–110.

[Maenpaa and Pietikäinen, 2005] Maenpaa, T. and Pietikäinen, M. (2005). Texture analysis with local binary patterns. Handbook of Pattern Recognition and Computer Vision.

[Martinez et al., 2012] Martinez, F., Carbonne, A., and Pissaloux, E. (2012). Gaze estimation using local features and non-linear regression. ICIP(International Conference on Image Processing).

[Matas et al., 2004] Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. Image Vision Comput., pages 761–767.

[Milborrow et al., 2010] Milborrow, S., Morkel, J., and Nicolls, F. (2010). The MUCT Landmarked Face Database. Pattern Recognition Association of South Africa. http ://www.milbo.org/muct.

[Moral et al., 1992] Moral, P. D., Rigal, G., and Salut, G. (1992). Estimation et commande optimale non-linéaire. technical report 2. Convention D.R.E.T.-DIGILOG-LAAS/CNRS.

[Mosimann et al., 2004] Mosimann, U. P., Felblinger, J., Ballinari, P., Hess, C. W., and Muri, R. M. (2004). Visual exploration behaviour during clock reading in alzheimer's disease. Brain, 127.

[Muja and Lowe, 2009] Muja, M. and Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In In VISAPP International Conference on Computer Vision Theory and Applications, pages 331–340.

[Nadler et al., 2005] Nadler, B., Lafon, S., Coifman, R. R., and Kevrekidis, I. (2005). Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. ArXiv Mathematics.

[Nakamura, 2005] Nakamura, J. (2005). Image sensors and signal processing for digital still cameras. Taylor & Francis.

[Nguyen et al., 2009] Nguyen, B. L., Chahir, Y., and Jouen, F. (2009). Eye gaze tracking. RIVF '09.

[Noris et al., 2008] Noris, B., Benmachiche, K., and Billard, A. (2008). Calibration-free eye gaze direction detection with gaussian processes. Proceedings of the International Conference on Computer Vision Theory and Application.

[Ojala et al., 1994] Ojala, T., Pietikäinen, M., , et al. (1994). Performance evaluation of texture measures with classification based on kullback discrimination of distributions. Proceedings of the International Conference on Pattern Recognition.

[Ojala et al., 1996] Ojala, T., Pietikäinen, M., et al. (1996). A comparative study of texture measures with classification based on feature distributions. Proceedings of the International Conference on Pattern Recognition.

[Ojala et al., 2002] Ojala, T., Pietikäinen, M., et al. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell.

[Ortiz, 2012] Ortiz, R. (2012). Freak : Fast retina keypoint. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), CVPR '12, pages 510–517, Washington, DC, USA. IEEE Computer Society.

[Pearson, 1901] Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. Philosophical Magazine, pages 559–572.

[Pentland et al., 1994] Pentland, A., Moghaddam, B., and Starner, T. (1994). View-based and modular eigenspaces for face recognition. In IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION & PATTERN RECOGNITION.

[Pérez et al., 2002] Pérez, P., Hue, C., et al. (2002). Color-based probabilistic tracking. ECCV '02, Part I.

[Pietikäinen et al., 2011] Pietikäinen, M., Zhao, G., Hadid, A., and Ahonen, T. (2011). Computer Vision Using Local Binary Patterns. Springer.

[Pless, 2003] Pless, R. (2003). Image spaces and video trajectories : Using isomap to explore video sequences. Proc. International Conference on Computer Vision (ICCV).

[Pless and Simon, 2002] Pless, R. and Simon, I. (2002). Using thousands of images of an object. CVPRIP.

[Politzer et al., 2010] Politzer, G., Baratgin, J., and Over, D. (2010). Betting on conditionals. Thinking & Reasonning, pages 172–197.

[Rahimi et al., 2005] Rahimi, A., Recht, B., and Darrell, T. (2005). Learning appearance manifolds from video. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01, CVPR '05, pages 868–875, Washington, DC, USA. IEEE Computer Society.

[Rane and Birchfield, 2007] Rane, N. and Birchfield, S. (2007). Isomap tracking with particle filter. ICIP(International Conference on Image Processing).

[Rasmussen and Williams, 2006] Rasmussen, C. E. and Williams, C. K. I. (2006). Gaussian processes for machine learning. MIT Press.

[Roweis and Saul, 2000] Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. SCIENCE, 290 :2323–2326.

[Scott et al., 1993] Scott, D., Findlay, J., et al. (1993). Visual search, eye movements and display units, human factors report. Technical report, University of Durham.

[Sheela and Vijaya, 2011] Sheela, S. and Vijaya, P. (2011). An appearance based method for eye gaze tracking. Journal of Computer Science.

[Shi and Malik, 2000] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.

[Silpa-Anan and Hartley, 2008] Silpa-Anan, C. and Hartley, R. (2008). Optimised kd-trees for fast image descriptor matching. In CVPR. IEEE Computer Society.

[Souvenir and Pless, 2007] Souvenir, R. and Pless, R. (2007). Image distance functions for manifold learning. Image Vision Comput., 25(3) :365–373.

[Sparks, 2002] Sparks, D. (2002). The brainstem control of saccadic eye movements. Nature Reviews : Neuroscience.

[Spearman, 1904] Spearman, C. (1904). "general intelligence", objectively dtermined and measured. American Journal of Psychology, pages 201–293.

[Spielman and Teng, 1996] Spielman, D. A. and Teng, S.-H. (1996). Spectral partitioning works : Planar graphs and finite element meshes. In In IEEE Symposium on Foundations of Computer Science, pages 96–105.

[Stiefelhagen et al., 1997] Stiefelhagen, R., Yang, J., and Waibel, A. (1997). Tracking eyes and monitoring eye gaze. Proc. Workshop Perceptual User Interfaces.

[Sugano et al., 2008] Sugano, Y., Matsushita, Y., Sato, Y., and Koike, H. (2008). An incremental learning method for unconstrained gaze estimation. In Proceedings of the 10th European Conference on Computer Vision : Part III, ECCV '08, pages 656–667, Berlin, Heidelberg. Springer-Verlag.

[Tan et al., 2002] Tan, K. H., Kriegman, D. J., and Ahuja, N. (2002). Appearance-based eye gaze estimation. Proc. Sixth IEEE Workshop Application of Computer Vision '02.

[Tenenbaum et al., 2000] Tenenbaum, J., de Silva, V., and Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. Science.

[T.F.Cootes et al., 1995] T.F.Cootes, C.J.Taylor, D.H.Cooper, and J.Graham (1995). Active shape models– their training and application. Computer vision and image understanding, 61(1) :38–59.

[Torgerson, 1952] Torgerson, W. (1952). Multidimensional scaling : I. theory and method. Psychometrica, pages 401–419.

[Viola and Jones, 2001] Viola, P. and Jones, M. (2001). Robust real-time object detection. Second International Workshop on Statistidal and Computational Theories of Vision- Modeling, Learning, Computing, and Sampling.

[Viola and Jones, 2004] Viola, P. and Jones, M. (2004). Robust real-time face detection. International Journal of Computer Vision, 57 :137–154.

[Vurpillot, 1991] Vurpillot, E. (1991). L'exploration oculaire. La perception de l'environnement, pages 161–177.

[Williams et al., 2006] Williams, O., Blake, A., and Cipolla, R. (2006). Sparse and semi-supervised visual mapping with the s3p. Proc. IEEE CS Conf. Computer Vision and Pattern Recognition.

[Wittman, 2005] Wittman, T. (2005). Manifold learning techniques : So which is the best ?

[Xu et al., 1998] Xu, L.-Q., Machin, D., and Sheppard, P. (1998). A novel approach to real-time non-intrusive gaze finding. Proc. British Machine Vision Conference.

[Yang, 2002] Yang, M.-H. (2002). Extended isomap for pattern classification. In Eighteenth national conference on Artificial intelligence, pages 224–229, Menlo Park, CA, USA. American Association for Artificial Intelligence.

[Yarbus, 1967] Yarbus, A. (1967). Eye movements and vision. New York : Plenum Press.

[Young and Householder, 1938] Young, G. and Householder, A. (1938). Discussion of a set of points in terms of their mutual distances. Psychometrica.

[Zhang et al., 2004] Zhang, J., Li, S. Z., and Wang, J. (2004). Manifold learning and applications in recognition. In in Intelligent Multimedia Processing with Soft Computing, pages 281–300. Springer-Verlag.

[Zhang et al., 2011] Zhang, Y., Bulling, A., and Gellersen, H. (2011). Discrimination of gaze directions using low-level eye image features. In Proceedings of the 1st international workshop on pervasive eye tracking &#38 ; mobile eye-based interaction, PETMEI '11, pages 9–14, New York, NY, USA. ACM.

Table des figures

Liste des tableaux

Résumé

L'oculométrie est un ensemble de techniques dédié à enregistrer et analyser les mouvements oculaires. Dans cette thèse, je présente l'étude, la conception et la mise en œuvre d'un système oculométrique numérique, non-intrusif permettant d'analyser les mouvements oculaires en temps réel avec une webcam à distance et sans lumière infra-rouge.

Dans le cadre de la réalisation, le système oculométrique proposé se compose de quatre modules : l'extraction des caractéristiques, la détection et le suivi des yeux, l'analyse de la variété des mouvements des yeux à partir des images et l'estimation du regard par l'apprentissage. Nos contributions reposent sur le développement des méthodes autour de ces quatre modules : la première réalise une méthode hybride pour détecter et suivre les yeux en temps réel à partir des techniques du filtre particulaire, du modèle à formes actives et des cartes des yeux (EyeMap) ; la seconde réalise l'extraction des caractéristiques à partir de l'image des yeux en utilisant les techniques des motifs binaires locaux ; la troisième méthode classifie les mouvements oculaires selon la variété générée par le Laplacian Eigenmaps et forme un ensemble de données d'apprentissage ; enfin, la quatrième méthode calcul la position du regard à partir de cet ensemble d'apprentissage. Nous proposons également deux méthodes d'estimation : une méthode de la régression par le processus gaussien et un apprentissage semi-supervisé et une méthode de la catégorisation par la classification spectrale (spectral clustering). Il en résulte un système complet, générique et économique pour les applications diverses dans le domaine de l'oculométrie.

Mots-clefs : Oculométrie, suivi des yeux, filtre particulaire, apprentissage par variétés, régression par processus gaussien, extraction des caractéristiques

.

Eye Tracking système: appearance based model and manifold learning

Abstract

Gaze tracker offers a powerful tool for diverse study fields, in particular eye movement analysis. In this thesis, we present a new appearance-based real-time gaze tracking system with only a remote webcam and without infra-red illumination.

Our proposed gaze tracking model has 4 components: eye localization, eye feature extraction, eye manifold learning and gaze estimation. Our research focuses on the developpement of methods on each component of the system. Firstly, we propose a hybrid method to localize in real time the eye region in the frames captured by the webcam. The eye can be detected by Active Shape Model and EyeMap in the first frame where eye occurs. Then the eye can be tracked through a stochastic method, particle filter. Secondly, we employ the Center-Symmetric Local Binary Patterns for the detected eye region, which has been divided into blocs, in order to get the eye features. Thirdly, we introduce manifold learning technique, such as Laplacian Eigenmaps, to learn different eye movements by a set of eye images collected. This unsupervised learning helps to construct an automatic and correct calibration phase. In the end, as for the gaze estimation, we propose 2 models: a semi-supervised Gaussian Process Regression prediction model to estimate the coordinates of eye direction; and a prediction model by spectral clustering to classify different eye movements. Our system with 5-points calibration can not only reduce the run-time cost, but also estimate the gaze accurately. Our experimental results show that our gaze tracking model has less constraints from the hardware settings and it can be applied efficiently in different real-time applications.

Keywords : Gaze tracking, eye localization and tracking, particle filter, manifold learning, gaussian process regression, feature extraction.