ALEXANDRE LEMIEUX

# SYSTÈME D'IDENTIFICATION DE PERSONNES PAR VISION NUMÉRIQUE

Mémoire
présenté
à la Faculté des études supérieures
de l'Université Laval
pour l'obtention
du grade de maître ès sciences (M.Sc.)

Département de génie électrique et de génie informatique
FACULTÉ DES SCIENCES ET DE GÉNIE
UNIVERSITÉ LAVAL
QUÉBEC

DÉCEMBRE 2003

# Résumé

Ces travaux de recherche présentent l'intégration complète d'un système d'identification de personnes par vision numérique. Conçu pour opérer dans un contexte de surveillance, le système a pour tâche d'identifier les personnes circulant dans une zone d'intérêt, et ce, à partir des informations contenues dans une base de données préalablement construite. Les techniques de reconnaissance sélectionnées utilisent uniquement le visage comme caractéristique discriminante. Le système réalise par ailleurs l'identification en quatre phases principales, soient l'acquisition, la détection du mouvement, la détection des visages et l'identification des personnes. Les images couleurs, acquisent à l'aide d'une caméra *web* abordable, sont prétraitées par un algorithme de soustraction de l'arrière-plan et soumisent à une méthode hybride de détection du visage. Les résultats obtenus sont ensuite acheminés à un module de reconnaissance multi-classifieurs utilisant des notions de patrons de design orientés-objets. Ceux-ci permettent entre autres une gestion efficace des classifieurs ainsi qu'une simplification du processus d'expérimentation des différentes combinaisons et des fonctions de décision. Finalement, la validation des techniques sélectionnées est réalisée à l'aide des banques d'images FERET et AR-face contenant respectivement les photos de 1 196 et 135 individus. Les configurations multi-classifieurs procurent des améliorations substantielles du taux de reconnaissance par rapport aux classifieurs individuels notamment dans le cas de la FERET.

Alexandre Lemieux

Marc Parizeau

# Avant-propos

Je tiens tout d'abord à remercier mon superviseur de maîtrise, Marc Parizeau, qui a toujours su me prodiguer des conseils judicieux depuis mon projet à l'automne 1998 et lors de mes stages ultérieurs. Ces études graduées m'ont permis d'acquérir de nombreuses connaissances et représentent sans aucun doute une expérience enrichissante.

Par la suite, je tiens à remercier Ingrid pour ses conseils et le soutien qu'elle m'a apporté tout au long de cette maîtrise. J'aimerais également souligner le support de mes parents ainsi que de ma famille. Je tiens à remercier tout particulièrement ma grand-mère Lemieux, qui restera toujours pour moi une très grande source de motivation.

Finalement, j'aimerais remercier les professeurs, le personnel de soutien ainsi que mes collègues du LVSN, notamment Christian Gagné et Jérôme Vignola, pour ces belles années au laboratoire !

# Table des matières

# Liste des tableaux

# Liste des figures

# Introduction

Depuis quelques années, on observe un besoin croissant pour des systèmes automatiques d'identification de personnes. On a qu'à penser aux besoins relatifs aux contrôles des frontières ainsi qu'à la surveillance des lieux publics tels que les banques, les aéroports, les centres commerciaux, *etc.* À cet égard, il est intéressant de noter que l'aéroport Pearson de Toronto possède déjà son système de reconnaissance d'individus [27].

Également, le gouvernement canadien se préoccupe de plus en plus de la sécurité nationale en lui consacrant 7,7 milliards de dollars sur 5 ans dans le cadre de son budget 2002 [6]. À ceci s'ajoute plus d'un milliard de dollars provenant du budget 2003 dédié aux forces armées canadiennes [7]. Il y a donc, d'un côté, des besoins importants de sécurité et, d'un autre, des ressources financières importantes.

L'identification d'une personne peut être réalisée à partir d'une image de l'individu, plus particulièrement de son visage. La vision numérique vise ainsi l'acquisition, le traitement et l'interprétation de ces images pour réaliser la reconnaissance des personnes. Ces systèmes sont particulièrement intéressants car ils permettent la surveillance silencieuse d'un endroit, sans requérir la coopération des individus.

**Vision numérique, automatisation et reconnaissance des formes** La vision numérique, ou vision artificielle, est un domaine visant la reproduction de la capacité de perception visuelle de l'œil humain. Ainsi, des caméras sont utilisées pour observer des scènes qu'elles reproduisent sous la forme d'un signal vidéo ou d'une séquence d'images. Ce domaine inclut également le traitement et l'analyse qui est effectué sur les données brutes.

Connue des milieux scientifiques depuis de nombreuses années, la vision numérique gagne maintenant le grand public. Pour ce faire, la plupart des ordinateurs vendus aujourd'hui sont équipés d'une caméra *web* permettant l'acquisition d'images, de films ainsi que l'exécution d'applications interactives. Citons entre autres les programmes de communication et les jeux.

Ces technologies sont également utilisées à des fins industrielles et commerciales. Mentionnons notamment l'industrie du bois (p. ex. : optimisation de coupe) et alimentaire (p. ex. : contrôle de la qualité pour les biscuits, coupe automatique de viande). Plusieurs avantages résultent de son utilisation, mais la vison artificielle permet avant tout la réalisation de tâches complexes à un coût abordable.

Des caméras peuvent également être utilisées dans des endroits dangereux ou difficiles d'accès ainsi que pour des applications nocturnes (thermographie infrarouge). Outre la vision 2D conventionnelle, l'utilisation d'informations tridimensionnelles permet la création d'applications mobiles, comme par exemple la manipulation précise d'objets ou le déplacement d'un robot.

Certains modèles de caméra permettent également l'exécution d'algorithmes par de l'électronique embarqué, ce qui réduit les coûts nécessaires. Pour les applications plus exigeantes, un ordinateur doit être utilisé, mais ceux-ci sont de plus en plus abordables et offrent des performances accrues.

Cela étant dit, une quantité phénoménale d'informations peut être traitée par un système de vision ; bien plus qu'un être humain effectuant le même travail. Par exemple, pour l'optimisation de la coupe des billots de bois, des milliers de solutions possibles peuvent être explorées en un court laps de temps. La vision ne fait cependant pas des miracles et lorsqu'un jugement est nécessaire, cela peut être très difficile à programmer et/ou à réaliser.

**Description du projet** La reconnaissance d'individus étant un sujet d'actualité, jumelé avec les défis passionnants de la vision numérique, le développement d'un système complet d'identification de personnes représente alors un projet très intéressant.

L'objectif principal concerne la conception d'une application d'identification de personnes en temps réel, en utilisant un montage simple et peu coûteux.

Parmi les objectifs supplémentaires, il y a entre autres le choix des algorithmes et méthodes nécessaires pour effectuer les tâches de détection et de reconnaissance. De plus, différentes expérimentations doivent présenter les niveaux de précision, de robustesse et d'efficacité des techniques sélectionnées.

En résumé, ce projet vise l'exploration des différentes facettes de la vision numérique à partir de l'acquisition des images et de leur traitement, jusqu'à l'interprétation et la reconnaissance des personnes. Il s'agit donc en fait de la conception d'un système complet de reconnaissance de personnes.

**Contraintes du projet** Pour réaliser ce projet, plusieurs contraintes ont été établies dès le départ afin de préciser les conditions réelles pour lesquelles l'application devait être conçue. Ces conditions ont donc permis le design d'un montage convenable ainsi que la sélection des différents algorithmes nécessaires au fonctionnement du système.

Tout d'abord, le système d'identification doit pouvoir observer une scène très complexe. Il n'y a donc aucune contrainte sur les objets pouvant s'y trouver, tant sur leur nature que sur leur quantité.

En plus d'être silencieux et discret, le système doit pouvoir fonctionner en temps réel en réalisant plusieurs identifications par seconde. Étant donné qu'il n'y a pas de contrainte sur le nombre d'objets, il est donc possible que plusieurs personnes se trouvent simultanément dans la zone de reconnaissance.

Comme contrainte supplémentaire, l'identification doit être réalisée seulement si les deux yeux de la personne sont visibles par les capteurs. Le système de reconnaissance se concentre donc sur l'identification frontale mais certaines tolérances à des variations en rotation sont par ailleurs souhaitées.

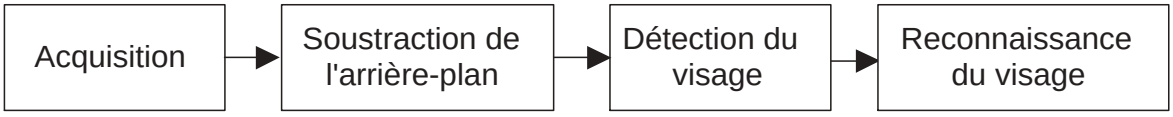



**Fig.** ***1:*** *Modules composant le système d'identification de personnes.*

| Caractéristiques | Valeurs |
|---|---|
| **Modèle** | Logitech® QuickCam® Pro 3000 |
| **Interface** | USB 1.0 |
| **Type de capteur** | CCD |
| **Résolution maximale** | 640×480 |
| **Couleurs** | 24 bits |
| **Débit d'images** | jusqu'à 30 FPS |
| **Prix** | environ 125$ |

**Tab.** ***1:*** *Spécifications techniques de la caméra utilisée pour le système d'identification de personnes.*

Concernant le montage physique, il se doit d'être simple et peu coûteux tout en offrant performance, robustesse et efficacité. Cette étape concerne donc le choix de l'équipement informatique et des capteurs à utiliser. Aucune contrainte monétaire n'est réellement appliqué au système mais il se doit d'utiliser des composantes standards et non spécialisées à des coûts raisonnables.

Cela étant dit, le système peut être fixe, c'est-à-dire qu'il peut être installé à un endroit sans aucun déplacement ultérieur. Cette condition est favorable à certains algorithmes, dont entre autres la soustraction de l'arrière-plan par modélisation statistique.

**Description de la solution** Le système d'identification de personnes développé comporte plusieurs modules effectuant chacun des tâches bien précises. La figure 1 illustre un diagramme représentant ces différentes parties.

Tout d'abord, l'acquisition est réalisée à l'aide d'une caméra abordable, la Logitech® QuickCam® Pro 3000, dont les spécifications techniques sont illustrées dans le tableau 1. Avec son coût approximatif de 125$, cette caméra satisfait l'exigence de coût abordable associée au projet.

| Composantes | Valeurs |
|---|---|
| **Processeurs** | 2 × Intel® Pentium® III (*dual*) |
| **Vitesse** | 550 Mhz |
| **Mémoire vive** | 512 Mo |
| **Carte graphique** | ASUS® AGP-V6800 32 Megs DDR SGRAM |
| **Système d'exploitation** | Windows® 2000 Professional |

**Tab.** ***2:*** *Spécifications techniques de l'ordinateur utilisé pour les expérimentations.*

Par la suite, un module de détection du mouvement traite les images brutes fournies au système. Ce module, utilisant une technique simple mais efficace de soustraction de l'arrière-plan, génère des images ne contenant que les pixels représentant des changements d'activités. Ce premier filtrage des données vise donc à éviter une recherche exhaustive sur des régions sans intérêt.

Ces pixels sont ensuite regroupés pour former des ensembles qui sont analysés par le module de détection du visage. Ce dernier utilise une technique hybride ayant pour but de déterminer les coordonnées des yeux avec le plus d'exactitude possible. Ces informations sont nécessaires à la normalisation de l'image représentant le visage, afin de la rendre compatible à celles utilisées lors de l'apprentissage.

Finalement, cette image normalisée est présentée au module de reconnaissance du visage afin d'identifier la personne. Pour davantage de robustesse, ce module d'identification est basé sur une architecture multi-classifieurs utilisant des méthodes de reconnaissance variées.

Toutes les opérations et calculs nécessaires sont réalisées à partir d'un ordinateur standard munis de composantes courantes et non spécialisées. Les spécifications de cet équipement sont résumées au tableau 2.

Ce mémoire est donc organisé comme suit. Tout d'abord, le chapitre 1 abordera en détails la problématique de la détection du mouvement. Le chapitre 2 portera ensuite sur la détection du visage alors que la reconnaissance d'individus sera traitée au chapitre 3. Finalement, le chapitre 4 présentera le fruit de nos expérimentations, notamment sur des banques d'images connues.

# Chapitre 1

# Détection du mouvement

Afin de limiter la somme importante de calculs et de traitements nécessaires à la segmentation, à la détection et à la reconnaissance de visages, il est avantageux d'utiliser des techniques de préfiltrage qui restreindront l'espace de recherche. Parmi ces méthodes, la détection du mouvement demeure l'une des plus efficaces.

## 1.1 Introduction

La détection du mouvement, réalisée immédiatement après l'acquisition d'une image, représente une étape non essentielle, mais très avantageuse pour un système de vision numérique. En effet, un gain de performance considérable peut être réalisé lorsque des

zones sans intérêt sont éliminées avant les phases d'analyses. Cette amélioration dépend cependant de la complexité des algorithmes de détection et de reconnaissance utilisés.

Par ailleurs, il est important de bien définir le terme *détection du mouvement* afin d'éliminer les ambiguïtés qui pourraient survenir. Nous en distinguons ici deux types : celui qui signale la présence de mouvement physique dans la scène ainsi que celui qui extrait et classifie les zones précises où le mouvement a lieu.

Le premier type de détection du mouvement peut s'effectuer tout d'abord à l'aide d'équipements spécialisés (p. ex. : détecteur infrarouge ou photosensible) ainsi que par vision numérique. L'objectif visé est seulement d'annoncer la présence de mouvement dans la scène afin de démarrer les acquisitions et le traitement des données.

La deuxième phase de détection consiste, quant à elle, à identifier les zones de mouvement dans l'image, rendant alors possible un raffinement de l'espace de recherche pour les opérations ultérieures. Parmi les techniques envisageables, la soustraction de l'arrière-plan (SAP) forme la plus grande famille de méthodes de détection du mouvement. Celles-ci sont par ailleurs assez répandues et ont été utilisées dans de nombreux systèmes [12, 42, 66, 9, 24, 44, 61].

Cela étant dit, l'utilisation de la vision numérique est nettement avantageuse car elle permet la réalisation des deux étapes de détection du mouvement à l'aide d'un seul module.

L'organisation du présent chapitre est la suivante. Tout d'abord, une recension des écrits sera abordée à la section 1.2 et couvrira la majorité des techniques de détection du mouvement. La section 1.3 portera ensuite sur l'approche retenue alors que la section 1.4 présentera certains résultats expérimentaux.

## 1.2 Recension des écrits

Dans cette section, plusieurs méthodes de détection du mouvement par vision numérique seront présentées. Pour celles-ci, la performance varie, autant en ce qui à trait aux temps de traitement, qu'à la qualité des résultats produits.

### 1.2.1 Soustraction de l'arrière-plan (SAP) par modélisation statistique

La présente technique est une des plus utilisées, probablement grâce à sa simplicité théorique ainsi qu'à sa faible complexité algorithmique. Le principe fondamental repose sur une estimation statistique de la scène observée. Le mouvement est détecté en comparant une image test avec le modèle d'arrière-plan calculé auparavant.

Certaines hypothèses de base doivent par contre être respectées pour un fonctionnement adéquat de cette méthode. Tout d'abord, la caméra utilisée est fixe et ne doit bouger à aucun moment. Une caméra à l'épaule est un bon exemple de situation non applicable à la SAP. Pour ce qui est de la scène observée, elle doit être relativement constante et conserver la même apparence. Un paysage observée à partir d'un train est donc une bonne représentation d'une scène non statique.

Il est important de noter qu'aucune limite n'est utilisée pour la quantité d'objets en mouvement. De plus, des variations de luminosité sont tolérées en autant qu'elles ne soient pas trop brusques.

Le modèle statistique calculé lors de la phase d'initialisation est constamment mis à jour, lui permettant ainsi de s'adapter aux changements qui peuvent se produire dans la scène observée (p. ex. : soleil levant). Cette capacité d'adaptation est commune à toutes les techniques de SAP par modélisation statistique et leur confère un atout majeur qui sera abordé en détails à la section 1.3. Par ailleurs, cette méthode connaît plusieurs implantations différentes qui varient principalement selon le type de capteur utilisé.

**Visible (2D)** La première catégorie de méthodes de soustraction d'arrière-plan regroupe les techniques basées sur l'utilisation d'images 2D dans le spectre visible.

Un des modèles de couleurs le plus fréquemment utilisé pour la modélisation statistique est le RGB. Celui-ci est modélisé par un cube où chacun des axes du référentiel représente une composante de base, soient le rouge (R), le vert (G) et le bleu (B). Une couleur particulière est donc un point à l'intérieur (ou aux frontières) du cube et codée par trois nombres pouvant prendre une valeur comprise dans l'intervalle $[0, 255]$. La

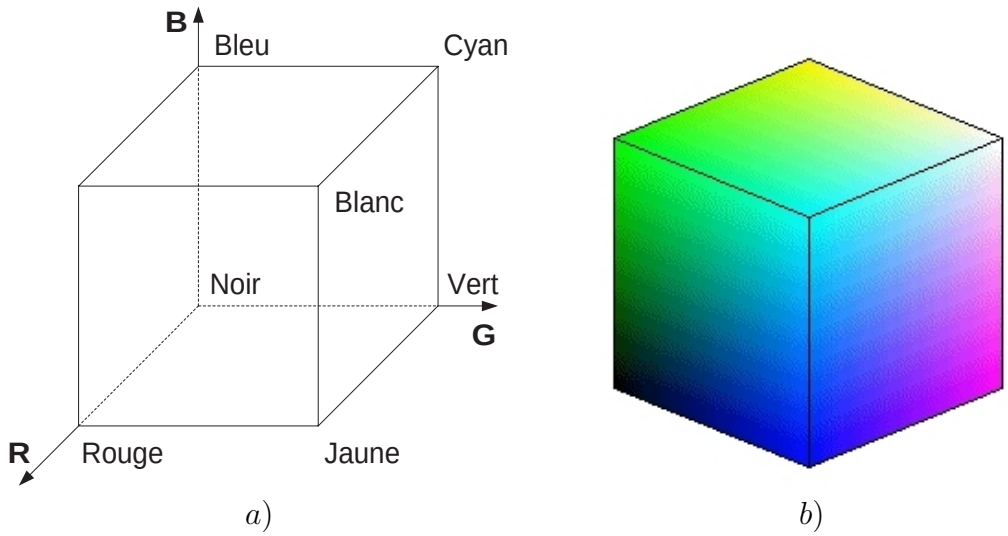


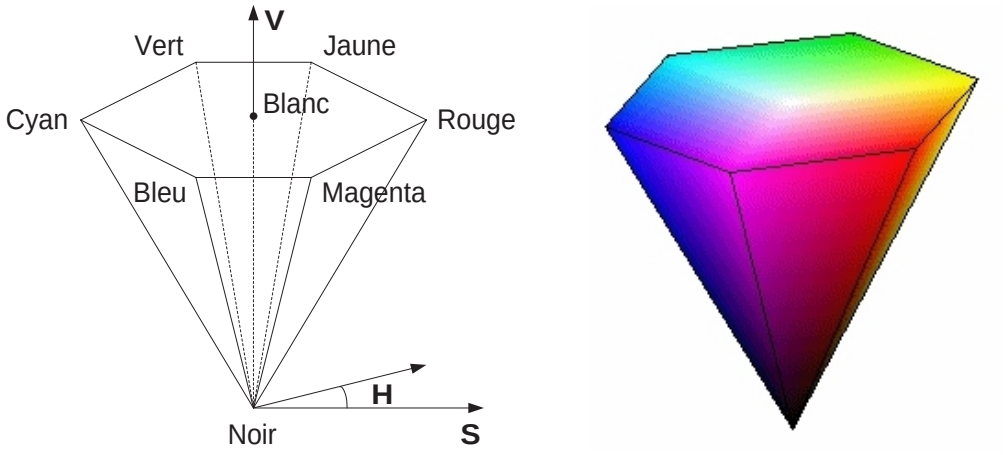

**Fig. *1.1:*** *Représentation graphique de l'espace de couleurs RGB. a) Avec un référentiel et b) Exemple de distribution des couleurs. Il est important de noter que les figures a et b ne représentent pas la même vue du cube RGB.*

figure 1.1 illustre une représentation graphique de cet espace de couleurs.

La technique de base consiste à modéliser l'arrière-plan à partir de plusieurs images acquises séquentiellement. Pour chaque pixel de l'image, ainsi que pour chacun des canaux (R, G et B), une moyenne et une variance sont calculées. Lorsqu'un pixel test doit être classifié, il faut tout d'abord lui soustraire la moyenne correspondante dans le modèle statistique. Il sera alors étiqueté comme un pixel contenant du mouvement seulement si la valeur absolue du résultat dépasse un certain multiple de l'écart-type correspondant.

Par ailleurs, d'autres modèles de couleurs ont été utilisés auparavant pour la modélisation statistique, comme par exemple le YUV [66] et le HSV [9]. Ce dernier, particulièrement utilisé en reconnaissance d'objets pour son invariance à la luminosité, est représenté par un cône hexagonal (*hexcone*) et est illustré à la figure 1.2. Alors que son axe principal représente la luminance (*Value* V), l'angle procure la couleur pure (*Hue* H) et la distance par rapport à l'axe central fournie la saturation (S).

Horprasert et al. [24] ont proposé un nouveau modèle de couleurs basé sur le RGB. Leur technique permet la classification des pixels en quatre catégories, soient l'arrière-plan original, illuminé, ombré et un pixel en mouvement. Pour ce faire, deux mesures sont ajoutées à la méthode de base en RGB, soient la distorsion chromatique ($\alpha$) et de




*a)* *b)*

**Fig.** ***1.2:*** *Représentation graphique de l'espace de couleurs HSV. a) Avec un référentiel et b) Exemple de distribution des couleurs. Il est important de noter que les figures a) et b) ne représentent pas la même vue du cône hexagonal HSV.*

luminosité ($CD$). Les points faibles de cette approche résident surtout dans la somme d'opérations supplémentaires nécessaires pour calculer ces deux mesures ainsi que les seuils associés. En pratique, certaines erreurs de classifications peuvent également se produirent entraînant, par exemple, l'identification d'un objet en mouvement comme étant de l'ombre.

Il y a finalement un très grand nombre de méthodes de SAP par modélisation statistique non abordées ici [42], notamment pour des raisons de complexité et pour lesquelles les gains en performance sur la technique de base sont relativement négligeables.

**Thermographie infrarouge (2D)** La deuxième méthode de SAP utilise quant à elle des images en tons de gris (256 niveaux de quantification) comme données d'entrées. Les valeurs des pixels représentent des températures observées dans la scène et obtenues grâce au procédé de thermographie infrarouge. Pour ce qui est des algorithmes de modélisation et de détection, ils sont identiques à ceux employés pour le visible à l'exception de l'espace de couleurs utilisé, ce qui simplifie légèrement le traitement.

L'atout majeur de l'infrarouge est sans contredit sa forte invariance aux illuminations. En effet, comme les observations sont basées sur la température, un brusque changement d'éclairage ne peut modifier suffisamment la température d'une zone pour que du mouvement y soit détecté. De plus, il est également possible de poursuivre

les acquisitions lorsque l'éclairage ambiant diminue ou devient nul, ce qui permet des utilisations nocturnes (p. ex. : opérations militaires ou de surveillance).

Les principaux défauts de cette technique reposent sur le coût élevé de l'équipement nécessaire ainsi que sur la faible résolution des images obtenues. Par contre, de plus en plus d'équipements abordables voient le jour et permettront de concevoir (à moyen terme) des systèmes avec thermographie infrarouge à un coût relativement bas.

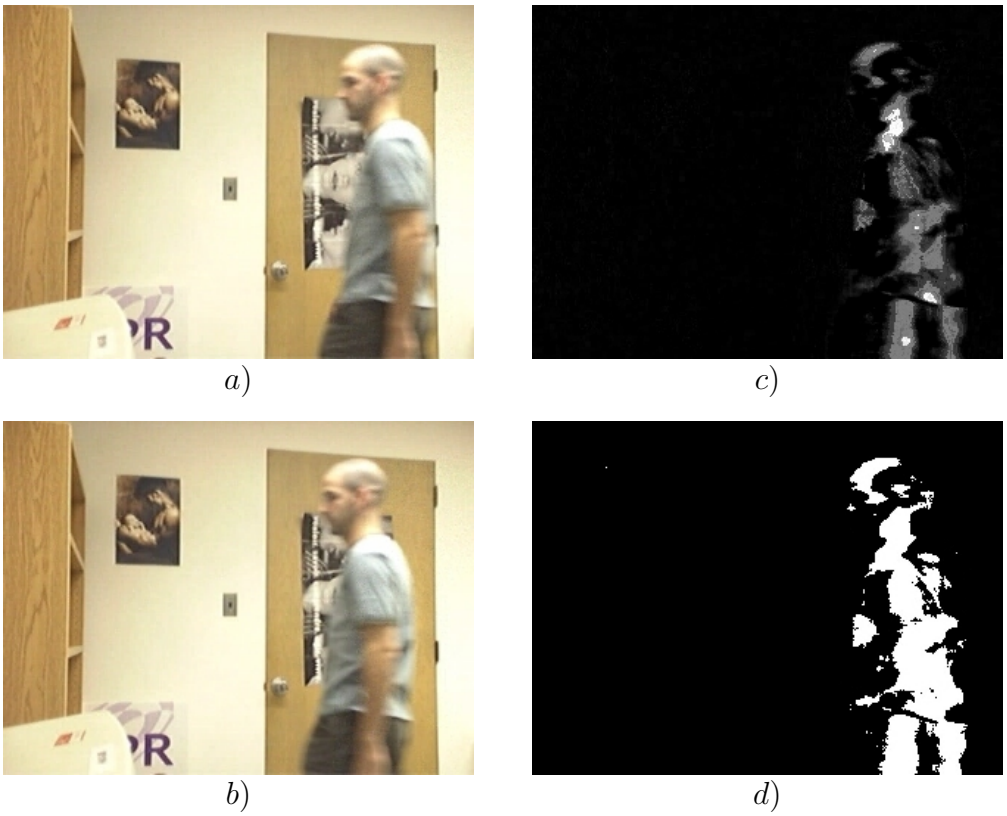
**Informations de profondeur (3D)** L'utilisation d'informations tridimensionnelles (p. ex. : capteur stéréoscopique) permet la réalisation d'une soustraction d'arrière-plan très efficace. Tout comme les méthodes précédentes, cette technique nécessite un modèle statistique de l'arrière-plan. Mais contrairement aux autres, il renferme des valeurs de distances entre la caméra et les différentes composantes de la scène. Le mouvement sera donc détecté lorsque des points seront à des distances différentes de celles retrouvées dans le modèle statistique. Une implantation en temps réel de cet algorithme requiert cependant énormément de puissance de calcul[1] ou de l'équipement matériel spécialisé [32].

Par ailleurs, Ivanov et al. [30] ont proposé un modèle hybride utilisant la couleur et les informations 3D pour accomplir la SAP. Dans ce cas, les informations de profondeurs sont modélisés et calculés hors-ligne en générant un modèle de disparité de l'arrière-plan nécessaire à la validation. Il est ensuite utilisé pour étiqueté un pixel qui ne respecte pas la couleur d'une image de référence comme étant de l'ombre ou un objet en mouvement. Les avantages majeurs de cette méthode repose sur sa robustesse à l'illumination et sa capacité à éliminer les ombres.

### 1.2.2 Différences entre deux images consécutives

Étant peu complexe, la différence entre deux images consécutives représente une solution très intéressante. Comme son nom l'indique, elle consiste à soustraire une image acquise au temps $t_n$ d'une autre au temps $t_{n+k}$, où $k$ est habituellement égal à 1. Ainsi, l'image résultante sera vide si aucun mouvement ne s'est produit pendant l'intervalle

[1] L'application de cette méthode requiert le calcul de la position tridimensionnelle de chacun des pixels de l'image test afin de les comparer avec le modèle d'arrière-plan.



*a)* *c)*

*b)* *d)*

**Fig.** ***1.3:*** *Exemple de détection du mouvement avec la méthode de soustraction d'images consécutives. a) Image $t_0$, b) Image $t_1$, c) Détection du mouvement non seuillée et d) Détection du mouvement.*

de temps observé car l'intensité et la couleur des pixels seront presque identiques. Par contre, si du mouvement a lieu dans le champ de vue, les pixels frontières des objets en déplacement devraient changer drastiquement de valeurs, révélant alors la présence d'activité dans la scène.

Cette technique nécessite très peu de ressources, car aucun modèle n'est nécessaire. Cela implique donc qu'il n'y a pas de phase d'initialisation obligatoire avec une scène statique, ce qui procure une très grande flexibilité d'utilisation. De plus, une opération de soustraction d'images requiert très peu de puissance de calcul, lui conférant un avantage supplémentaire.

Par ailleurs, les résultats obtenus avec cette méthode ne sont pas aussi éloquents que ceux générés en utilisant un modèle statistique de l'arrière-plan. En effet, certains traitements supplémentaires sont nécessaires afin de déterminer la zone en mouvement, car l'information disponible ne concerne que les contours des régions en déplacement (ce qui inclus également les zones intérieures d'un objet).

La figure 1.3 illustre certains résultats expérimentaux. Alors que les images 1.3a et b représentent les deux images consécutives utilisées, les images 1.3c et d contiennent les résultats de la détection du mouvement. La différence entre celles-ci reposent sur un seuillage appliqué aux données pour binariser les résultats et ainsi faciliter la visualisation.

Il est par ailleurs intéressant de remarquer les nombreuses irrégularités de l'image 1.3d, ce qui complique l'étape d'extraction et surtout, qui nuit à une segmentation robuste.

### 1.2.3 Flux optique

Similaire à l'approche précédente, l'utilisation du flux optique procure une information de mouvement pour chaque pixel de l'image. Ainsi, il mesure les vecteurs de déplacement à partir de l'intensité des pixels de deux images consécutives ou temporellement rapprochées. Dans un contexte de détection de mouvement, les pixels inactifs posséderont alors une vélocité nulle contrairement aux pixels appartenant à des objets dynamiques. Une classification sous forme de regroupement est donc nécessaire afin d'isoler et de localiser les zones représentant du mouvement. Cette technique a notamment été utilisée pour la détection de piétons [10]. Il y a finalement plusieurs méthodes pour calculer le flux optique, mentionnons entre autres celle de Lucas et Kanade [38] ainsi que celle de Horn et Schunck [23].

L'inconvénient majeur de l'utilisation du flux optique est la somme importante de calculs à réaliser pour l'estimation du mouvement. Par ailleurs, une variante utilisant le *block matching*[2] peut bénéficier de certaines instructions optimisées MMX$^{TM}$, ce qui peut accélérer le traitement global. Néanmoins, une tâche supplémentaire de classification et d'interprétation est nécessaire.

De plus, si certaines parties d'un objet ne sont pas en mouvement, elles seront complètement ignorées par cette méthode. Ce pourrait être le cas par exemple d'une séquence vidéo contenant une personne assise par terre et agitant les bras. Dans cette

[2]Le *block matching* est utilisé dans plusieurs algorithmes de compression vidéo pour la prédiction du mouvement. Il a donc fait l'objet de recherches intensives afin d'optimiser son exécution.

situation bien précise, le corps de la personne ne serait pas détecté contrairement à ses bras.

### 1.2.4 Élimination des ombres

Des ombres peuvent parfois être générées par l'effet de sources de lumière sur des objets en mouvement. Ces cas spéciaux peuvent alors produirent des désagréments, par exemple pour l'appariement des parties entre deux régions. Cependant, les ombres ne sont pas trop dérangeantes dans un cadre de reconnaissance de visage, car elles peuvent être éliminées facilement par une opération de masquage.

Par ailleurs, les méthodes de reconnaissance d'individu basées sur le corps en entier souffriront énormément de ces effets secondaires. Certaines techniques peuvent toutefois solutionner en majorité ou en partie ce problème. Il y a notamment la méthode d'Horprasert et al. [24] abordée à la section 1.2.1, de Mikiæ et al. [44] et celle de Cucchiara et al. [9]. Cette dernière technique utilise par exemple l'information de couleur obtenue à l'aide du HSV afin d'éliminer les ombres. En effet, un arrière-plan ombré devrait posséder en principe une couleur identique avec une luminosité plus faible.

Outre les avantages incontestables de l'élimination des ombres, il ne faut pas oublier que des fausses détections peuvent se produirent lors de cette phase supplémentaire. Or, dans le cadre actuel du projet, mieux vaut conserver des zones inutiles que de les éliminer trop rapidement et ainsi ignorer un visage.

## 1.3 Approche retenue

Tout d'abord, il est impératif de comparer les avantages et les inconvénients de chacune des approches envisagées. Le tableau 1.1 résume les principales remarques. Parmi toutes ces méthodes, peu d'entre elles respectent l'ensemble des exigences et des besoins du projet.

Premièrement, les contraintes matérielles (c.-à-d. : coût) excluent *de facto* l'utili-

| Type | Avantages | Inconvénients |
|---|---|---|
| **SAP (visible 2D)** | - Algorithme peu complexe<br>- Classification simple<br>- Résultats clairs | - Initialisation/scène statique<br>- Ombres non rejetées |
| **SAP (infrarouge 2D)** | - Idem (visible 2D)<br>- Robustesse à l'éclairage<br>- Éclairage faible ou nul<br>- Robustesse aux ombres | - Initialisation/scène statique<br>- Coût élevé du matériel<br>- Faible résolution des images |
| **SAP (visible 3D)** | - Robustesse aux ombres<br>- Informations de profondeur | - Initialisation/scène statique<br>- Complexité et calculs<br>- Plusieurs caméras |
| **Différences d'images consécutives** | - Flexibilité d'utilisation<br>- Faible complexité de l'algorithme de base<br>- Souplesse d'initialisation | - Mouvement obligatoire<br>- Ombres non rejetées<br>- Détection incomplète |
| **Flux optique** | - Informations précises sur le mouvement<br>- Suivi/prédiction possible | - Complexité et calculs<br>- Ombres non rejetées<br>- Mouvement obligatoire<br>- Interprétation difficile |

**Tab.** ***1.1:*** *Tableau comparatif des différentes méthodes de détection du mouvement.*

sation d'informations tridimensionnelles ou d'équipement d'imagerie infrarouge. Pour ce qui est du flux optique, l'importante somme de calculs nécessaire ainsi que l'interprétation difficile des résultats générés nuisent à sa sélection. Ensuite, grâce à leur faible complexité et leur temps de traitement raisonnable, deux approches différentes sont finalement sélectionnées, soient la SAP par modélisation statistique (2D visible) et la détection de mouvement par différence d'images consécutives. Cette dernière méthode, quoique possédant un avantage certain sur le plan de l'initialisation, souffre de certaines limitations du côté de la classification, favorisant finalement la SAP par modélisation statistique 2D dans le spectre visible.

La sous-section 1.3.1 abordera donc l'algorithme utilisé par la technique sélectionnée. La sous-section 1.3.2 traitera ensuite de l'implantation logicielle réalisée dans le cadre du projet.

### 1.3.1 SAP par modélisation statistique : algorithme

L'algorithme utilisé pour la soustraction de l'arrière-plan par modélisation statistique comporte trois étapes importantes : l'initialisation, l'extraction du mouvement (avant-plan) et la mise à jour du modèle.

**Initialisation** La première étape consiste à modéliser l'arrière-plan à partir des $N$ premières images ($N \approx 30$) d'une séquence vidéo. Une moyenne d'intensité est donc calculée à partir de ces images pour chaque pixel et pour chacun des canaux (R, G et B). La moyenne d'intensité d'un pixel donné se résume alors à l'équation suivante :

$$\mu_c(x, y) = \frac{1}{N} \sum_{i=0}^{N} I_{i,c}(x, y) \tag{1.1}$$

où $I_i$ est la $i$ème image d'initialisation, $N$ la quantité d'images utilisées et $c$ le canal sélectionné.

L'étape suivante consiste à calculer un écart-type $\sigma$ pour chaque pixel (et pour chaque canal) afin d'être utilisé comme seuil de détection. Cette opération nécessite habituellement le stockage des $N$ premières images. Or, une équation modifiée permet de contourner cette contrainte de façon incrémentale et ainsi réduire la consommation d'espace mémoire. Pour ce faire, deux accumulateurs sont utilisés, soient $S(x, y)$ pour stocker la somme des intensités des pixels et $SC(x, y)$ pour emmagasiner la somme des carrés. Les écarts-types peuvent alors être calculés à l'aide de l'équation 1.2. Par ailleurs, il est intéressant de remarquer que $S(x, y)$ peut être réutilisé pour le calcul de la moyenne, ce qui évite des opérations supplémentaires et superflues.

$$\sigma_c(x, y) = \sqrt{\left(\frac{SC_c(x, y)}{N}\right) - \left(\frac{S_c(x, y)}{N}\right)^2} \tag{1.2}$$

**Extraction de l'avant-plan** Afin d'extraire le mouvement dans une image, le modèle de l'arrière-plan doit tout d'abord lui être soustrait. Chaque pixel, dont la différence en valeur absolue dépasse la valeur $\alpha \times \sigma$, est ensuite classifié comme étant un pixel en mouvement. Dans l'expression précédente, la variable $\alpha$ représente une certaine fraction de l'écart-type $\sigma$. En pratique, ce paramètre se situe dans l'intervalle $[2.0, 4.0]$ et dépend

du niveau d'exclusion désiré. Un masque binaire de mouvement peut alors être généré pour chaque canal à l'aide de l'équation 1.3 :

$$m_c(x,y) = \begin{cases} 1 & \text{si } |I_c(x,y) - \mu_c(x,y)| > \alpha\sigma_c(x,y) \\ 0 & \text{autrement} \end{cases} \tag{1.3}$$

où $m_c(x,y)$ représente le masque de mouvement pour un canal $c$ et $I_c(x,y)$ l'image d'entrée à analyser.

L'équation 1.3 représente le calcul du masque de mouvement pour un seul canal. Pour utiliser cet algorithme avec les 3 canaux (RGB) des images utilisées, les masques individuels doivent tout d'abord être générés indépendamment et combinés par la suite à l'aide d'un opérateur $OU$ logique. Par conséquent, si un mouvement est détecté pour un pixel dans un seul canal, cela sera suffisant pour en modifier l'état. L'équation suivante représente cette combinaison produisant ainsi un masque de mouvement à un seul canal :

$$M(x,y) = m_r(x,y) \cup m_g(x,y) \cup m_b(x,y) \tag{1.4}$$

Une fois cette opération complétée, certaines opérations de morphologie mathématique [17] doivent être appliquées afin d'éliminer le bruit et les fausses détections. Pour ce faire, 2 *érosions* et 2 *dilatations* sont appliquées respectivement dans cet ordre sur le masque de mouvement. Finalement, l'image d'entrée est combinée avec le masque pour produire une image à 3 canaux (avant-plan) contenant seulement les pixels représentant du mouvement. Cette opération peut se résumer à l'équation suivante :

$$F(x,y) = M(x,y)I(x,y) \tag{1.5}$$

où $F(x,y)$ représente l'image d'avant-plan (mouvement ou *foreground*) et $I(x,y)$ l'image d'entrée. Les deux images sont combinées grâce à une multiplication pixel à pixel pour chacun des canaux.

**Mise à jour du modèle** Au cours de la période d'acquisition, certaines régions de la scène peuvent subir des modifications d'éclairage, ce qui rend la mise à jour du modèle statistique de l'arrière-plan primordiale. Ainsi, un changement graduel de luminosité (p. ex. : lever du soleil) sera donc intégré au modèle et ne sera pas considéré comme du mouvement. Pour ce faire, l'extraction de l'avant-plan est réalisée avec l'image courante, ce qui génère un masque de mouvement $M$. Le modèle de l'arrière-plan est ensuite mis à jour à partir du complément de $M$, c'est-à-dire en utilisant tous les pixels qui sont

étiquetés comme faisant partie de l'arrière-plan. Les changements brusques dans l'image ne sont donc pas ajoutés au modèle. L'équation 1.6 illustre ce processus de mise à jour :

$$\mu'_c(x,y) = (1-\eta)\mu_c(x,y) + \eta I_c(x,y)\overline{M}(x,y) \tag{1.6}$$

où $\mu'(x,y)$ représente un pixel de l'arrière-plan moyen mis à jour et $\eta$ le taux d'apprentissage. L'expression $I_c(x,y)\overline{M}(x,y)$ représente les pixels statiques de l'image courante, c'est-à-dire ceux pour lesquels aucun changement n'est associé.

Afin de ne pas modifier radicalement le modèle d'arrière-plan, seulement une fraction $\eta$ de l'image temporaire $I_c(x,y)\overline{M}(x,y)$ est utilisée. En pratique, ce taux d'apprentissage peut prendre des valeurs comprises dans l'intervalle $[0.05, 0.25]$. Plus la valeur de ce paramètre est élevée, plus les changements s'intégreront rapidement. Cela revient alors à oublier rapidement le modèle construit lors de la phase d'initialisation. Il est conseillé d'utiliser des valeurs relativement faibles (p. ex. : 0.05).

Finalement, l'écart-type n'est pas ajusté ou mis à jour pendant l'exécution de l'algorithme (c.-à-d.. : une fois l'initialisation effectuée) afin de réduire la somme de calculs nécessaire. Certaines expérimentations supplémentaires devraient cependant être réalisées pour vérifier l'utilité et l'impact de cette mise à jour sur les résultats.

### 1.3.2 Implantation logicielle

Afin de réaliser la soustraction de l'arrière-plan, une classe en C++ fût créée en utilisant certaines fonctions utilitaires de la librairie OpenCV d'Intel® [28]. Cette classe nommée *BackgroundSub* possède les fonctions nécessaires pour initialiser et mettre à jour le modèle statistique ainsi que pour extraire les images de mouvemement et de l'arrière-plan.

À l'usage, certaines particularités de la méthode de base ont pu être simplifiées, notamment pour les seuils de détection. En effet, il est habituellement courant d'utiliser une valeur d'écart-type pour chaque pixel et pour chaque canal de l'image. Cependant, des expérimentations ont démontrées que l'utilisation d'un écart-type global pour chacun des canaux ne dégradait nullement les résultats obtenus. En pratique ces valeurs sont calculées à l'aide de la moyenne des écarts-types de chaque canal correspondant.

Cette simplification est utilisée dans notre implantation, représentant donc une légère économie d'espace et d'accès mémoire lors de la phase d'extraction du mouvement.

**Interface** Afin d'interagir avec la classe, plusieurs fonctions d'interface ont été créées. La liste suivante résume les principales fonctions publiques accompagnées d'une courte description pour chacune d'elle :

- *init* : Initialisation des différents compteurs, accumulateurs et modèles nécessaires à l'exécution des algorithmes ;
- *update* : Mise à jour du modèle statistique. Cette fonction dépend essentiellement de l'état du modèle statistique, c'est-à-dire du niveau d'avancement de celui-ci. Le traitement exécuté dépend du nombre d'images d'initialisation ajouté au modèle et est déterminé automatiquement lors de l'appel de la fonction. Les deux modes sont :
  1. Modèle non complet : l'image d'entrée est ajoutée aux différents accumulateurs et le compteur d'images d'initialisation est incrémenté ;
  2. Modèle initialisé : le modèle statistique de l'arrière-plan est modifié en ajoutant une fraction des pixels statiques de l'image d'entrée (équation 1.6) ;
- *getMask* : Extraction du masque des pixels en mouvement à l'aide de l'équation 1.4 ;
- *getForeground* : Extraction de l'avant-plan (ou pixels en mouvement). Cette opération implique un calcul du masque du mouvement et réprésente l'application de l'équation 1.5 ;
- *getBackground* : Extraction de l'arrière-plan. En pratique, le modèle statistique est simplement converti et copié dans l'image passée par référence à la fonction ;
- *reset* : Réinitialisation des variables membres, compteurs, modèles, *etc.* afin de démarrer une nouvelle soustraction de l'arrière-plan.

**Optimisations** Finalement, plusieurs optimisations logicielles peuvent contribuer à des gains de performance intéressants. Tout d'abord, une modification à l'algorithme de base concernant le masque de mouvement est déjà implantée dans la classe développée. Lorsque plusieurs accès au masque sont requis (c.-à-d. : appel séquentiel des fonctions *getForeground*, *getMask* et *update*), il est possible de le conserver en mémoire afin de le réutiliser dans les fonctions subséquentes. Cela évite alors le calcul du même masque plusieurs fois de suite. Le seul inconvénient avec cette façon de procéder est

que l'utilisateur doit prendre les précautions nécessaires pour ne pas utiliser un masque obsolète.

Dans le cas où la plateforme de calculs utilisée dispose de plusieurs processeurs, il est possible de distribuer le traitement sur chacun d'entre eux. En effet, ceci peut être réalisé en utilisant simplement un nombre de *thread* égal à la quantité de processeurs disponibles. Le système d'exploitation[3] s'occupera ensuite de partager les processus entre ses différentes unités de calcul. Chaque processus pourra alors effectuer des traitements sur une portion seulement de l'image (c.-à-d. : moitié pour deux processeurs), ce qui réduira substantiellement le temps nécessaire au traitement d'une image. L'implantation est par ailleurs simplifiée, car aucune communication n'est requise entre les processeurs. Cela est dû au fait que les traitements ne dépendent pas des résultats des autres régions de l'image.

Il est également envisageable de programmer certaines parties du traitement à l'aide du jeu d'instructions MMX™. Ceci pourrait améliorer grandement la performance des nombreuses opérations d'addition et de multiplication de vecteurs réalisées dans l'algorithme. Par ailleurs, certaines opérations de base des librairies OpenCV, IPL (*Image Processing Library*) et IPP (*Integrated Performance Primitives*) sont déjà optimisées en totalité ou en partie. Il est cependant difficile de déterminer le niveau d'optimisation réalisé puisque ces modules sont pré-compilés et non disponibles à des fins de consultation[4].

Finalement, l'optimisation la plus facilement réalisable est l'ajustement de la fréquence de mise à jour du modèle statistique. En effet, il est inutile d'effectuer cette opération à chaque acquisition d'image considérant que dans la majorité des cas, la différence entre deux images consécutives est minime (p. ex. : mise à jour d'une image sur deux). Certaines expérimentations seront notamment présentées à la section 1.4. Toutefois, il serait également envisageable, lorsque les conditions sont contrôlées, d'éliminer totalement la phase de mise à jour.

[3] Il est important de noter que les systèmes d'exploitation ne sont pas tous conçus pour supporter des applications multi-processeurs.

[4] La librairie OpenCV est à code source libre (*open source*) sauf les *dll* (*Dynamic Loading Libraries*) optimisées selon les modèles de processeur de marque Intel®. Pour ce qui est des librairies IPL et IPP également développées par Intel®, aucune partie n'est à code ouvert.

*a)* *b)* *c)*

**Fig. *1.4:*** *Exemple de soustraction d'arrière-plan : personne. a) Scène statique, b) Image test et c) Mouvement détecté.*

## 1.4 Résultats expérimentaux

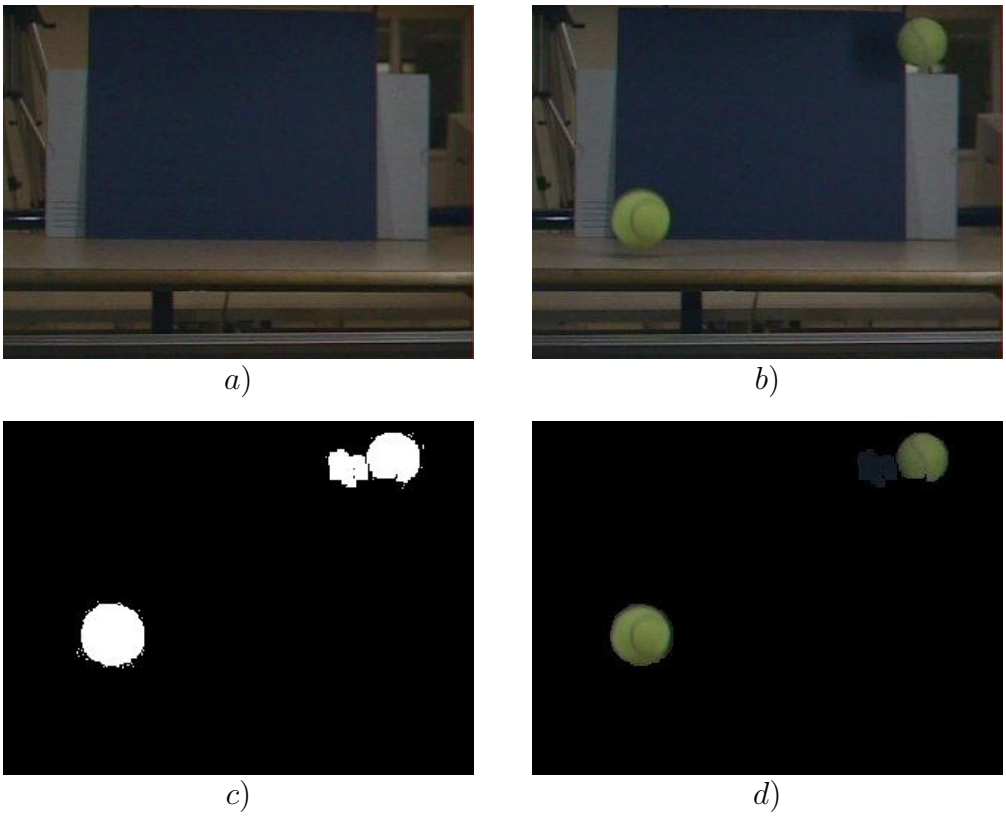
La présente section illustre certains résultats obtenus avec la méthode de SAP par modélisation statistique ainsi que certaines expérimentations avec différents paramètres de l'algorithme.

Il est à noter que pour toutes les expérimentations réalisées, la fonction d'ajustement automatique (*auto white balance*) de la caméra était désactivée. De plus, les valeurs utilisées pour le temps d'intégration ainsi que le gain du capteur étaient fixes et non modifiées au cours d'une même acquisition.

**Détection de mouvement** Tout d'abord, un exemple de détection de mouvement appliqué à une séquence vidéo contenant un être humain est illustré à la figure 1.4. Alors que l'image 1.4a représente l'arrière-plan modélisé, l'image 1.4c illustre quant à elle le mouvement détecté dans l'image test en 1.4b.

Il est intéressant de remarquer l'ombre projetée sur le mur (située au bas de la main gauche). Ce type de fausse détection pourrait notamment nuire à une segmentation en parties du corps humain ou à une squelettisation efficace. L'interrupteur mural est également détecté, ce qui est probablement dû à une réflexion sur le métal de la plaque. Par ailleurs, la soustraction de l'arrière-plan demeure très efficace, conservant la quasi intégralité de la personne sur toute sa superficie.

La figure 1.5 illustre quant à elle les résultats obtenus à partir d'une séquence vidéo contenant deux balles bondissantes. Une ombre est une fois de plus projetée par la



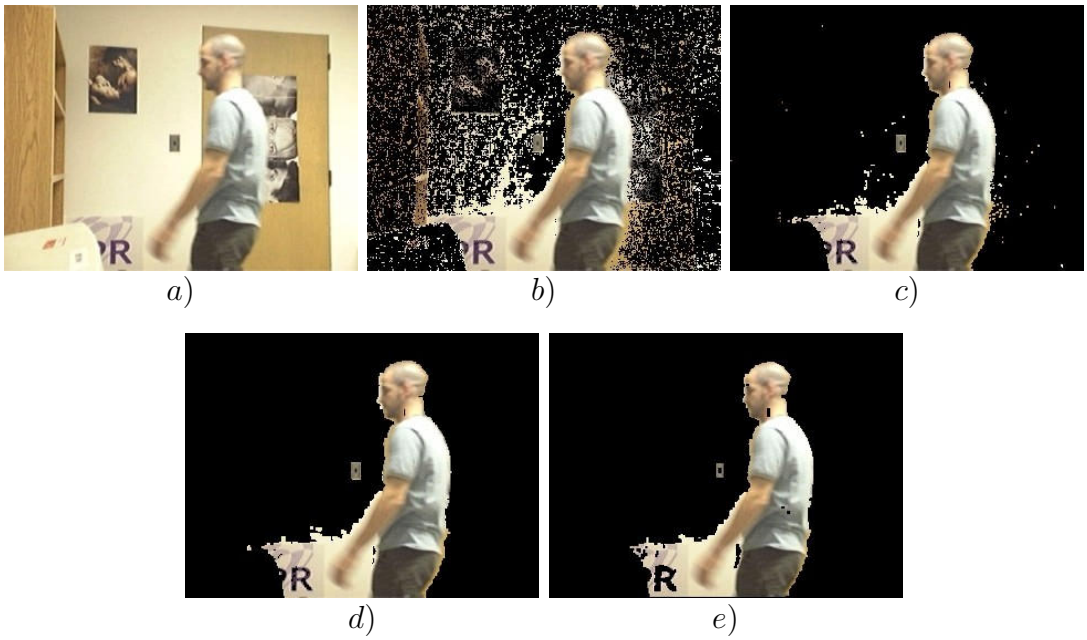
*a)* *b)*

*c)* *d)*

**Fig.** ***1.5:*** *Exemple de soustraction d'arrière-plan : balles. a) Scène statique, b) Image test, c) Masque de mouvement et d) Avant-plan (mouvement détecté).*

balle de droite (clairement visible à l'image 1.5c), pouvant nuire à sa localisation et à son suivi. Celle-ci peut également occasionner une fausse détection gênante en étant identifiée comme une balle supplémentaire.

**Opérations de morphologie mathématique** Il est également intéressant d'observer l'impact des opérations de morphologie mathématique sur la soustraction de l'arrière-plan. La figure 1.6 illustre les résultats obtenus avec certaines variations de paramètres. Tout d'abord, l'image 1.6a représente une image test alors que les résultats obtenus sont illustrés en b, c, d et e pour différentes combinaisons d'opérations. Le seuil $\alpha$ utilisé pour la détection a été fixé à 4.0 pour ces expérimentations. Les résultats obtenus suggèrent l'utilisation d'au moins 1 paire d'opérations (1 érosion suivie de 1 dilatation) pour éliminer le bruit (voir figure 1.6b *vs* 1.6c). En pratique, la combinaison de 2 érosions suivies de 2 dilatations (illustrée en 1.6d) procure également de bons résultats.



*a)* *b)* *c)* *d)* *e)*

**Fig. *1.6:*** *Influence des opérations de morphologie mathématique sur la détection de mouvement : personne. a) Image test, b) Aucune opération, c) 1 érosion suivie de 1 dilatation, d) 2 érosions suivies de 2 dilatations et e) 3 érosions suivies de 1 dilatation.*

**Génération du masque de mouvement** Certaines expérimentations supplémentaires ont été réalisées afin d'évaluer l'importance de l'opérateur *OU* logique dans la combinaison des masques individuels. La figure 1.7 illustre les résultats obtenus en utilisant un opérateur *ET* logique. Alors que l'image test est illustrée en 1.7a, les images b et c représentent quant à elles le mouvement détecté avec respectivement aucune opération morphologique et une combinaison de 2 érosions suivies de 2 dilatations.

Il est intéressant de remarquer que l'image 1.7b contient beaucoup moins de bruit que celle obtenue avec l'opérateur *OU* (image 1.6b) lorsqu'aucune opération de morphologie mathématique n'est employée. Cependant, l'image 1.7b contient des zones de mouvement non détectées et ceci s'aggrave lorsque l'image est filtrée pour éliminer le bruit.

Bien que l'algorithme utilise tous les masques de mouvement, il serait également envisageable de ne conserver qu'un seul masque à titre d'approximation ou de combiner les 3 canaux RGB avant la phase d'extraction de l'avant-plan.

*a)* *b)* *c)*

**Fig.** ***1.7:*** *Effets de l'utilisation d'un opérateur logique* ET *pour la création du masque de mouvement. a) Image test, b) Aucune opération et c) 2 érosions suivies de 2 dilatations.*

**Performance** Finalement, le tableau 1.2 résume la capacité de traitement de l'algorithme à différentes résolutions et pour certaines fréquences de mise à jour. Il est important de noter qu'aucun affichage n'a été réalisé pendant ces expérimentations.

Ces résultats ont été obtenus en utilisant la plateforme décrite dans l'introduction. Donc, pour des images de dimensions 320×240, le système peut traiter jusqu'à 24 images par seconde (*FPS* ou *Frame Per Second*).

Par ailleurs, lorsque la fréquence de mise à jour est abaissée de moitié, le nombre d'images traitées par seconde augmente drastiquement (surtout à basses résolutions) avec, par exemple, un gain de 48 FPS pour une résolution de 160×120 (c.-à-d. : passe de 124.0 à 172.0 FPS).

Notons cependant que la quantité d'images traitées ne double pas lorsque la fréquence de mise à jour est abaissée de moitié. Ceci s'explique simplement par le fait que les opérations nécessaires à la mise à jour du modèle ne constituent pas l'ensemble des étapes à effectuer à chaque itération de l'algorithme.

## 1.5 Conclusion

Ce chapitre a présenté, dans un premier temps, la majorité des techniques de détection du mouvement. Après plusieurs comparaisons, la méthode de soustraction de l'arrière-plan par modélisation statistique (visible 2D) a été retenue pour la réalisation du projet. L'algorithme utilisé est une simplification de la technique de base, notamment par rapport au calcul des seuils de détection.

| Résolution | Fréquence de mise à jour | FPS |
|---|---|---|
| 160×120 | 1 : 1 | 124.0 |
| | 1 : 2 | 172.0 |
| 320×240 | 1 : 1 | 24.0 |
| | 1 : 2 | 31.2 |
| 640×480 | 1 : 1 | 3.9 |
| | 1 : 2 | 4.8 |

**Tab.** ***1.2:*** *Tableau comparatif de performance pour la soustraction de l'arrière-plan à différentes résolutions d'images et fréquences de mise à jour du modèle.*

La méthode sélectionnée procure de très bons résultats. De plus, un grand nombre d'optimisations est envisageable pour accélérer son traitement.

Le désavantage majeur relié à son utilisation repose sur les ombres qui peuvent parfois être générées par les objets en mouvement. Par contre, dans un contexte de reconnaissance du visage, ce problème ne cause pas de désagréments majeurs.

Finalement, l'usage de la thermographie infrarouge pourrait être un atout majeur pour un système de surveillance, procurant une invariance à l'éclairage et aux ombres.

# Chapitre 2

# Détection et normalisation du visage

De nombreux algorithmes de reconnaissance du visage ont été développés au cours des dernières années et plusieurs se révèlent très performants. Cependant, le succès de ces méthodes dépend largement de la qualité des résultats de détection et de normalisation des visages. En effet, plus la précision obtenue est élevée, plus les conditions se rapprocheront de celles de la phase d'apprentissage, ce qui augmentera les probabilités d'une identification efficace.

## 2.1 Introduction

La performance d'un système de reconnaissance de personnes est évidemment tributaire de la qualité et de l'efficacité du module d'identification. Une mauvaise localisation et/ou normalisation du visage entraîne cependant une chute drastique du taux de reconnaissance. Dans ce cas bien précis, l'extraction de la représentation du visage sera erronée et difficilement comparable aux prototypes d'apprentissage.

Il est donc impératif que ce module de prétraitement soit le plus précis possible afin d'obtenir les meilleures conditions initiales pour la suite des opérations.

Cela étant dit, le processus global de détection et de normalisation peut se diviser comme suit :

1. Détection des visages dans l'image d'entrée ;
2. Positionnement et identification des composantes du visage (c.-à-d. : yeux, nez, *etc.*) ;
3. Validation des résultats ;
4. Normalisation du visage.

La première étape consiste donc à procéder à la détection des visages dans l'image d'entrée. Étant donné que les individus observés peuvent se trouver à une distance variable de la caméra, une approche multi-échelle s'avère essentielle afin d'obtenir un taux de détection élevé, indépendamment de la taille des visages. Pour réaliser cette tâche, de nombreuses techniques peuvent être appliquées [21, 72].

Les différentes composantes du visage devront ensuite être localisées, tout particulièrement les yeux qui orienteront le processus de normalisation. L'intérêt porté envers ceux-ci repose essentiellement sur leur invariance pour un même individu. Ainsi, les yeux seront toujours espacés d'une certaine distance, peu importe l'expression faciale de la personne. Les positions des caractéristiques du visage permettent également l'estimation de la pose de la tête.

La prochaine étape consiste à valider le ou les visages détecté(s) lors de la phase précédente afin d'exclure les fausses détections. Finalement, le visage est normalisé en géométrie et en intensité, ce qui rend l'image compatible avec celles utilisées lors de

l'apprentissage.

L'organisation du présent chapitre est la suivante. La section 2.2 portera sur les différentes techniques de détection du visage développées jusqu'à présent. La section 2.3 traitera ensuite de l'approche retenue ainsi que de l'implantation logicielle réalisée. La section 2.4 présentera quant à elle certains résultats expérimentaux. Une discussion à propos des avantages et des inconvénients de la méthode développée suivra finalement à la section 2.5.

## 2.2 Recension des écrits

Une grande variété de méthodes de détection du visage a été proposée dans les dernières années [21, 72]. La plupart d'entre elles s'attardent par contre à répondre aux questions "Y a-t-il des visages dans cette image ? Si oui, où sont-ils ?", sans toutefois extraire les coordonnées des caractéristiques du visage. Or, il est possible dans plusieurs cas de dériver ou de spécialiser la technique afin d'obtenir ces informations supplémentaires. Les prochaines sous-sections résumeront donc les principales méthodes de détection appuyées par des résultats intermédiaires.

### 2.2.1 Couleurs

Lorsque les images d'entrées sont en couleurs, il est avantageux d'utiliser cette information supplémentaire pour isoler les régions susceptibles de contenir des visages. En effet, plusieurs auteurs ont développé et utilisé ce que l'on pourrait qualifier de *détecteur de peau* [25, 55, 56]. Dans la majorité des cas, la peau est représentée par une portion d'un espace de couleurs particulier. En utilisant les frontières de cette région comme valeurs de seuillage sur une image, il est possible d'extraire les pixels dont la couleur peut s'apparenter à celle de la peau.

Il y a plusieurs modèles de couleurs pouvant s'appliquer à la détection de la peau [60]. L'un des espaces le plus couramment utilisé pour effectuer cette tâche est le HSV [55, 56], introduit préalablement au chapitre 1. L'avantage du HSV pour la détection

des couleurs réside dans le fait qu'un des canaux représente la luminance (*Value* V). Cette particularité permet d'exprimer adéquatement les couleurs sans se soucier des variations de luminosité. Ainsi, les pixels peau seront extraits en observant seulement la teinte (*Hue* H) et la saturation (S) des pixels. La figure 2.1 illustre notamment certains exemples d'extraction. Plus particulièrement, les figures 2.1a à 2.1c représentent les images originales alors que les images 2.1d à 2.1f illustrent les résultats d'extraction.

Il est intéressant de remarquer les fausses détections pour les cheveux ainsi que les régions non détectées (p. ex. : front). Cela peut s'expliquer par le fait que certaines zones reflètent davantage la lumière et semblent ainsi plus éclairées, ce qui modifie la couleur de la région dans l'espace HSV. Dans ce cas bien précis, la saturation tend vers 0 et la valeur vers 1, ce qui est représenté sur le cône hexagonal par une région située au centre de sa base. Autrement dit, lorsque la saturation est basse et que la valeur est élevée, la couleur tend vers le blanc, échappant donc aux seuils de détection. Pour ce qui est des cheveux, ils peuvent correspondre dans certains cas aux couleurs de la peau.

Certains auteurs ont également proposé un apprentissage automatique des couleurs représentant la peau à l'aide de réseaux neuronaux [29, 49]. Ceux-ci peuvent être entraînés à partir d'échantillons de pixels représentant la peau (et de contre-exemples), préalablement converti dans l'espace de couleur $YC_rC_b$. Par ailleurs, d'autres espaces de couleurs peuvent être utilisés dans ce contexte.

Cela étant dit, les fonctions de distributions gaussiennes ont aussi été employées dans plusieurs travaux [71] pour la modélisation de la couleur peau. D'autres auteurs ont également proposé l'utilisation d'un modèle mixte de gaussiennes (*Gaussian Mixture Model*), qui semble mieux représenter et modéliser la portion d'un espace de couleurs associée à la couleur peau [70].

La détection des composantes du visage peut se faire à partir des résultats de l'extraction de la peau. Une des méthodes envisageable consiste à analyser les trous contenus dans la zone de peau, comme par exemple ceux générés par les yeux. La validation peut alors être réalisée en respectant différentes règles de positionnement (des trous entre eux et par rapport au visage) et de taille. Ces règles peuvent être ajustées dynamiquement en analysant les moments[1] [59] de la zone de peau et semblent de plus

[1] L'analyse des moments fournie entre autres les orientations maximales de la zone de peau, informations utiles entre autres pour l'estimation de l'inclinaison de la tête et la validation de la position

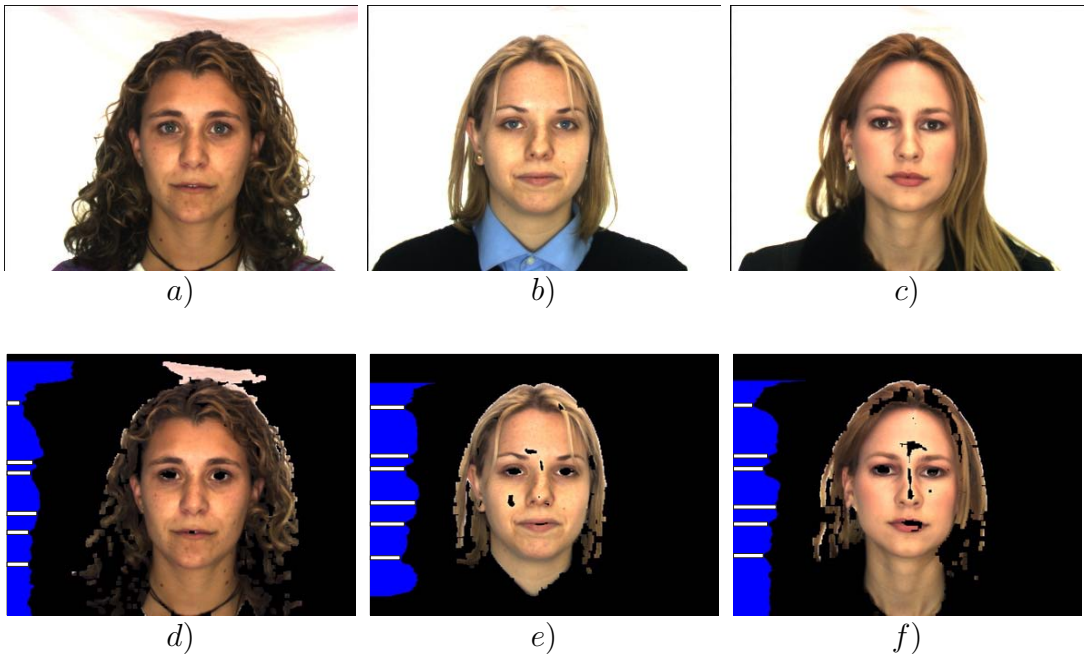



**Fig.** ***2.1:*** *Exemple d'extraction des pixels représentant la peau (HSV). a) b) c) Images originales, d) e) et f) Extraction de la peau et reliefs de projections horizontales des intensités moyennes. Les lignes blanches représentent les minimums globaux correspondants à certaines caractéristiques du visage.*

particulièrement adaptées à l'utilisation de logique floue.

Une autre technique possible consiste quant à elle à effectuer des projections (horizontales et verticales) afin de produire des reliefs d'intensités moyennes [55]. Les images 2.1d à 2.1f illustrent notamment les reliefs horizontaux correspondants aux images 2.1a à 2.1c. Dans ces exemples, les zones sombres du visage (c.-à-d. : yeux, bouche, nez, menton) sont caractérisées par des moyennes d'intensités plus basses, produisant des creux dans les reliefs. Les yeux peuvent alors être localisés à l'aide du relief vertical généré par la ligne horizontale correspondante. Cette méthode a été utilisée avec succès dans différents travaux [55, 56].

### 2.2.2 Appariement de gabarits (*Template matching*)

L'appariement de gabarit, ou *template matching*, est certainement une des techniques de détection du visage la plus simple qui soit. Elle consiste en effet à comparer

des yeux.

l'intensité des pixels entre un gabarit prédéfini et plusieurs sous-régions de l'image à analyser. Ce processus correspond en pratique à effectuer plusieurs balayages couvrant toute la superficie de l'image. Les endroits les plus propices à la présence de visages seront donc facilement identifiés par des minimums de distance entre le gabarit et l'image sous-jacente.

Parmi les distances envisageables, il y a notamment la $L_1$ (*city-block*) et la $L_2$ (distance euclidienne), la cross-corrélation, ainsi que plusieurs autres[2]. Il est à noter que cette technique peut également être appliquée sur une carte d'arêtes, ce qui peut simplifier le calcul.

Les plus grands désavantages de cette méthode résident sans contredit au niveau de la pose du visage et de son échelle. Pour résoudre ce problème, plusieurs gabarits peuvent être définis pour différentes poses, mais ceci rend la gestion des résultats assez complexe. Par ailleurs, cette technique n'étant pas intrinsèquement multi-échelle, plusieurs balayages à différents niveaux de redimensionnement du gabarit sont nécessaires. Ceci implique donc la réalisation d'une étape de décision supplémentaire dans le but de filtrer les multiples détections appartenant aux mêmes visages. Finalement, la construction d'un gabarit efficace représente un défi en elle-même. En effet, l'utilisation d'un gabarit plus ou moins adapté au type d'objet recherché peut nuire à une détection efficace et diminuer la précision des résultats.

Pour ce qui est de la localisation des différentes caractéristiques du visage, leurs positions peuvent être déduites à partir des positions correspondantes sur le gabarit. Celles-ci doivent être préalablement déterminées manuellement en positions relatives par rapport aux dimensions du gabarit. Il n'est cependant pas garanti que le gabarit soit parfaitement positionné en translation, en échelle et en rotation sur le visage à détecter, ce qui produira des coordonnées légèrement erronées.

Par ailleurs, certains travaux [4, 72] ont utilisé une détection des caractéristiques du visage à l'aide de gabarits plus spécialisés (p. ex. : yeux, bouche, *etc.*). Cette méthode implique par contre une recherche intensive dans un vaste espace de solutions possibles (c.-à-d. : rotation, échelle et translation).

[2] La librairie OpenCV [28] d'Intel® offre par ailleurs un grand choix de distances utilisables.

### 2.2.3 Arêtes

La prochaine famille de méthodes de détection du visage porte sur l'utilisation des arêtes. Celles-ci sont décrites comme étant formées des "points de discontinuités dans la fonction d'illuminance (intensité) de l'image" [3]. Ces informations utiles sont notamment employées pour l'interprétation de scène et la reconnaissance d'objets.

Le principe de base consiste à reconnaître des objets dans une image à partir de modèles de contours connus au préalable [3]. Pour réaliser cette tâche, deux méthodes seront présentées : la transformée de Hough et la distance de Hausdorff.

**Transformée de Hough** La transformée de Hough est une méthode permettant d'extraire et de localiser des groupes de points respectant certaines caractéristiques. Par exemple, les particularités recherchées peuvent être des droites, des arcs de cercle, des formes quelconques, *etc.* Dans un contexte de détection de visage, ce dernier est représenté par une ellipse dans la carte d'arêtes. L'application de la transformée de Hough circulaire produirait donc une liste de tous les candidats étant des cercles ou des dérivés [3, 39].

L'algorithme de base a également été modifié pour voir ainsi apparaître plusieurs variantes, dont la *Randomized Hough Transform*, qui peut être appliqué à la recherche de formes quelconques tout comme des cercles (p. ex. : visages) [67, 68, 73].

Finalement, la transformée de Hough peut être utilisée pour détecter les yeux et les iris. Par contre, cette méthode échouera lorsque l'image est trop petite ou lorsque les yeux ne sont pas clairement visibles.

**Distance de Hausdorff** La prochaine méthode utilise quant à elle les arêtes comme données de base ainsi qu'un algorithme spécial de *template matching*. En effet, la distance de Hausdorff [26] vise à mesurer la distance entre deux ensembles de points, qui sont la plupart du temps une carte d'arêtes (image de recherche) et un modèle. L'algorithme de base effectue la recherche des meilleurs endroits de correspondance partout dans l'image (translation) et ce, pour différentes rotations. Cette recherche peut également inclure un facteur d'échelle afin de détecter des variations du modèle

original.

Cette méthode a été utilisée avec succès pour la détection de visages avec vues frontales [31]. L'adaptation de celle-ci pour pallier à différentes poses (rotation axiale de la tête) amène cependant certains problèmes, car différents modèles devraient être utilisés. De plus, une étape complexe de décision aurait pour mission de départager les fausses détections ainsi que les détections multiples.

### 2.2.4 *EigenObjects* et *EigenFaces*

Les techniques des *EigenObjects* (EO) [45] et des *EigenFaces* (EF) [62] ressemblent légèrement à la méthode d'appariement de gabarit, notamment en ce qui a trait aux inconvénients inhérents à leur utilisation. Tout d'abord, étant donné l'importance de cette technique pour la reconnaissance du visage, davantage de détails seront fournis aux sections ultérieures (3.2.3.2 et 3.3.1.1).

Cela étant dit, l'élément essentiel de ces méthodes repose sur la réalisation d'une analyse en composantes principales (ACP) sur une série d'images représentant les objets à détecter. Cette opération de réduction de dimensionnalité fournie les premiers vecteurs propres (d'où le terme *eigen*) représentant les plus fortes différences entre les objets d'intérêt (visages, yeux, nez, *etc.*). Celles-ci peuvent également être vues comme les directions, dans l'espace des images, où la variance est la plus élevée.

Tout comme dans le cas du *template matching*, l'étape de détection consiste à effectuer un balayage de la zone à traiter. Ensuite, une reconstruction avec les premiers vecteurs propres est réalisée pour chacune des sous-images à analyser. Si l'image reconstruite est suffisamment fidèle à l'imagette d'origine, celle-ci possède alors de fortes ressemblances avec la catégorie d'objet à reconnaître. En d'autres mots, la sous-image possède les caractéristiques discriminantes de la classe visée et peut être exprimée efficacement avec la base générée par les vecteurs propres. Cette ressemblance est mesurée en calculant la distance entre les deux images à l'aide d'une métrique particulière comme celles utilisées pour l'appariement de gabarit.

Dans le cas des *EigenFaces*, les visages sont les objets recherchés alors que pour les

*EigenObjects*, ce sont les différentes composantes du visage qui sont d'intérêt (yeux, nez, bouche, oreilles, *etc.*). Évidemment, la technique des *EigenObjects* peut bénéficier de quelques hypothèses de base pour valider les résultats, comme par exemple les positions relatives des différentes composantes du visage. Il est par ailleurs envisageable de combiner les deux méthodes pour diminuer le temps de recherche et raffiner les résultats. En effet, une première recherche de visages pourrait être réalisée à l'aide des EF suivie d'une étape de raffinement utilisant les EO et ce, exclusivement dans les régions potentielles.

Cette technique souffre cependant des mêmes inconvénients que ceux inhérents au *template matching*, ce qui rend son utilisation difficile. Les techniques des EF et des EO ont par contre l'avantage d'être moins sensibles au choix du gabarit et de l'éclairage. Afin d'assurer une bonne détection, la construction des images moyennes et des vecteurs propres doit se faire à partir d'images suffisamment compatibles avec celles qui seront présentées lors de la détection.

### 2.2.5 Réseau de neurones

La détection du visage à l'aide de réseaux neuronaux [13, 52] se résume à l'utilisation d'un classifieur à deux sorties[3] représentant la présence ou l'absence de l'objet recherché dans une sous-région de l'image. Le principe de base, identique à certaines techniques précédentes, consiste à balayer l'image avec une fenêtre d'attention de dimensions fixes et de réaliser la détection sur les sous-images. Néanmoins, il est encore une fois nécessaire d'effectuer plusieurs balayages à différentes résolutions pour ainsi réaliser une détection suffisamment robuste.

Différents types de réseaux de neurones peuvent être utilisés [72], mais pour la plupart d'entre eux, les données d'entrées sont l'intensité des pixels de l'image (après un prétraitement adéquat). Par ailleurs, le réseau peut utiliser d'autres caractéristiques extraites des sous-images, comme par exemple des arêtes ou des informations fréquentielles.

[3] Le classifieur peut également ne posséder qu'une seule sortie qui s'activera lors de la présence d'un visage.

Afin de réaliser l'apprentissage du réseau de neurones artificiel, une banque d'images contenant des visages est nécessaire. Avant tout, ces images devront être redimensionnées pour être compatibles avec les dimensions requises par le nombre d'entrées du réseau. Par exemple, la couche d'entrée possédera 875 neurones pour des imagettes de dimensions 25×35 pixels.

Pour pallier aux effets produits par les rotations possibles de la tête, il est possible d'ajouter un réseau de neurones routeur qui prénormalise la sous-image en rotation [51]. Ce filtrage ne corrige cependant pas une pose de la tête résultant d'une rotation axiale. Pour remédier à ce problème, il est envisageable d'ajouter des images d'entraînement contenant différentes poses.

Comme dans plusieurs autres techniques multi-échelles, une étape supplémentaire de prise de décision est nécessaire afin de réduire les réponses multiples pour un même visage. Finalement, les coordonnées des caractéristiques peuvent être déduites à partir des positions moyennes dans les images d'apprentissage. Ce processus de détection, tout comme les précédents, peut cependant manquer de précision lors de cette étape de localisation.

## 2.3 Approche retenue

Parmi toutes les techniques abordées à la section précédente, certaines s'avèrent plus efficaces, tant au niveau du temps d'exécution que de la précision des résultats. Le tableau 2.1 résume les différents avantages et inconvénients pour chacune de ces méthodes.

Des techniques comparées au tableau 2.1, il est intéressant de remarquer que la majorité des méthodes ne sont pas adéquates à une détection précise des yeux. Par contre, plusieurs d'entre elles excellent à la détection du visage. Une étape de raffinement est donc à prévoir pour améliorer la précision de la localisation des caractéristiques du visage.

L'utilisation d'une image d'entrée ne contenant que des pixels en mouvements favorise inévitablement l'ensemble de ces techniques. En effet, les balayages à différentes

| Méthode | Avantages | Inconvénients |
|---|---|---|
| **Couleurs** | - Rapidité<br>- Détection de la peau efficace | - Détection des yeux peu robuste<br>- Conflits avec l'arrière-plan |
| ***Template matching*** | - Conceptuellement simple<br>- Mesure de similarité | - Recherche multi-échelle<br>- Filtrage des multiples détections<br>- Faible précision<br>- Modèle représentatif |
| ***EigenObjects*** | - Gabarit moins crucial<br>- Moins sensible à l'éclairage<br>- Mesure de similarité | - Idem au *template matching* |
| **Arêtes (Hausdorff)** | - Estimation de la rotation<br>- Implicitement multi-échelle | - Modèle représentatif |
| **Arêtes (Transformée de Hough)** | - Estimation de la rotation (ellipse)<br>- Implicitement multi-échelle<br>- Invariance aux rotations | - Faible précision (détection des yeux)<br>- Arêtes bien visibles<br>- Estimation de la rotation (cercle) |
| **Réseaux de neurones** | - Apprentissage automatique<br>- Capacité de généralisation | - Faible précision<br>- Recherche multi-échelle |

**Tab. *2.1:*** *Tableau comparatif des différentes méthodes de détection du visage.*

résolutions requis par certaines méthodes sont évités grâce à une focalisation sur les zones d'intérêt, c'est-à-dire les régions qui contiennent du mouvement.

Après avoir comparé les différentes méthodes, l'utilisation d'une méthode hybride alliant les forces de plusieurs techniques a été retenue. Cette solution est composée des étapes suivantes :

1. Détection des pixels représentant la peau à partir de l'image de mouvement ;
2. Extraction et filtrage des groupes de pixels ;
3. Détection du visage par *template matching* à l'intérieur des régions d'intérêt (un seul gabarit) ;
4. Raffinement du positionnement des caractéristiques du visage par *template mat-*

| Canal | Seuil inférieur | Seuil supérieur |
|---|---|---|
| **H (*Hue*)** | 0° | 27° |
| **S (*Saturation*)** | 0.2 | 0.8 |
| **V (*Value*)** | 0 | 1 |

**Tab. *2.2:*** *Seuils utilisés lors de la détection de la peau.*

*ching* (trois gabarits).

Les sous-sections 2.3.1 et 2.3.2 aborderont donc en détails les différentes étapes de la solution retenue, suivies de la présentation de résultats expérimentaux à la section 2.4.

### 2.3.1 Détection du visage

Après avoir effectué la modélisation de l'arrière-plan et la détection du mouvement, les images résultantes sont présentées au module de détection du visage. Ainsi, il est inutile de poursuivre le traitement si aucun ou peu de pixels sont en mouvement.

Dans le cas contraire, quatre étapes de traitement sont appliquées sur les nuées de pixels restantes, en débutant tout d'abord par la détection de la peau.

**Détection de la peau** La première étape consiste à détecter et à identifier les pixels de l'image qui représentent les couleurs pouvant correspondre à de la peau. Suivant la méthode abordée précédemment (2.2.1), l'image est avant tout convertie de l'espace de couleurs RGB vers le HSV. Un filtrage est effectué par la suite afin de conserver uniquement les pixels contenus dans une portion bien définie de l'espace HSV.

Certains auteurs ont proposés [55] des seuils appropriés pour ce type d'opération. Ceux utilisés lors des expérimentations sont illustrés au tableau 2.2. Il est important de noter que les seuils utilisés sur le canal $V$ sont à titre indicatif seulement car ce canal est ignoré[4] lors du seuillage.

[4]C'est donc dire que les valeurs de $V$ n'influencent nullement la détection de la peau.

Cette opération de seuillage, appliquée sur chacun des canaux, produit une image binaire contenant trois plans. Tous les pixels sont initialisés à 0 et chaque pixel retenu se voit attribuer la valeur 1.

Un masque de pixels à un seul canal est ensuite généré en combinant les trois plans via une opération de *ET* logique. Pour qu'un pixel soit étiqueté comme étant de la peau, il doit donc se situer à l'intérieur de tous les intervalles précités.

Le dernier traitement consiste à utiliser des opérations de morphologie mathématique tout comme pour la soustraction d'arrière-plan. En pratique, 3 érosions suivies de 3 dilatations sont appliquées sur le masque. Celui-ci est finalement utilisé pour modifié l'image RGB de base, qui ne contiendra alors que les pixels représentant de la peau.

**Extraction des nuées de pixels** La prochaine étape consiste à isoler et à extraire les différents groupes de pixels connectés, communément appelés *blobs* ou nuées de pixels. Cette segmentation est réalisée à l'aide de fonctions présentes dans la librairie OpenCV [28] d'Intel®. La fonction *cvFindContours*, basée sur une connectivité à 8 voisins, utilise notamment des algorithmes de suivi de contours [58] pour modéliser et extraire les frontières des différentes régions. Ceux-ci fournissent, en une seule passe[5] ligne par ligne dans l'image, différentes structures représentant les contours des objets présents.

L'un de ces algorithmes est particulièrement intéressant car il génère un arbre hiérarchique des contours internes et externes des objets. Il est ainsi possible d'identifier et de localiser les différentes nuées de pixels dans l'image pour les analyses ultérieures.

Dans le cas où les *blobs* détectés possèdent une aire trop faible, ils sont automatiquement rejetés. Cette opération peut également éliminer les amas de pixels douteux, comme par exemple ceux possédant des ratios hauteur/largeur non conformes (p. ex. : rectangle de dimensions 100×10 pixels). Un ratio acceptable peut être situé entre 1 :1 et 4 :3 (hauteur/largeur). Il est par contre inutile de filtrer trop rapidement des régions, mais plutôt d'exclure seulement celles dont le ratio semble invraisemblable.

Avant de poursuivre les traitements, les amas de pixels restants sont complétés à

[5] Certains cas rares peuvent requérir des passes supplémentaires.

l'aide des pixels originaux. C'est-à-dire que toutes les régions vides à l'intérieur des *blobs* sont remplacées par les zones correspondantes de l'image originale. La nuée de pixels est ensuite emmagasinée dans un objet de type *Blob* accompagnée de l'image originale contenue dans le rectangle englobant[6].

La classe *Blob* contient les informations relatives à un amas de pixels, comme entre autres :

- Rectangle englobant (*bounding box*) ;
- Zone de l'image originale correspondante ;
- Zone de l'image de mouvement correspondante ;
- Moments ;
- Contours internes et externes ;
- Nombre de pixels ;
- *etc.*

**Détection du visage par appariement de gabarit** Une fois les prétraitements complétés, la détection du visage proprement dite peut être effectuée sur la zone de recherche grandement réduite. La méthode sélectionnée, le *template matching*, requiert un gabarit efficace avant toute chose. La problématique de la conception d'un modèle représentatif des objets à repérer est donc à résoudre. Celui-ci se doit d'être assez général pour bien apparier n'importe quel objet de la classe visée, mais sans toutefois effectuer trop de fausses détections.

Afin de solutionner ce problème, certaines méthodes d'intelligence artificielle pourraient être employées. Il serait par exemple envisageable d'utiliser la programmation génétique pour trouver des solutions, en espérant converger vers un gabarit idéal au cours des générations. Il existe par contre d'autres approches plus simples et surtout moins voraces en temps *cpu*.

En effet, le gabarit peut être généré à partir d'un visage moyen calculé avec une banque d'images représentatives. La technique de reconnaissance de visage *EigenFaces* requiert justement le calcul d'un visage moyen ; celui-ci peut donc être réutilisé ici à des fins de détection. Le gabarit utilisé lors des expérimentations est illustré à la figure 2.2a.

[6] Le rectangle englobant, ou *bounding box*, représente la zone minimum servant à contenir l'ensemble de l'amas de pixels.

La méthode de *template matching* conventionnelle consiste à balayer entièrement l'image afin de trouver le meilleur appariement, et ce à plusieurs échelles et inclinaisons. Cette approche souffre par contre de plusieurs désavantages qui ont été mentionnés auparavant. Or, grâce à la soustraction de l'arrière-plan et à la détection de la peau, l'espace de recherche peut être considérablement diminué car une majorité de pixels sont habituellement inintéressants. Ceci consiste donc en pratique à réduire drastiquement le nombre et l'étendue des balayages.

L'algorithme se poursuit avec une analyse des moments [28, 59] pour chacun des *blobs* afin d'évaluer certains paramètres. Cette analyse fournie entre autres des renseignements précieux sur la géométrie et l'état du groupe de pixels. Ceux d'entre eux qui représentent des visages devraient occupées des zones relativement ovales, un fait qui peut être vérifié grâce à leur degré d'excentricité[7].

Parmi les informations obtenues par l'analyse des moments, les axes de variations maximales permettent notamment l'estimation de l'orientation de chacun des amas de pixels. Ainsi, les balayages avec des degrés de rotation variés, qui sont habituellement effectués afin de détecter les visages sous différentes rotations, peuvent être éliminés ou largement simplifiés.

Un angle $\alpha$ est estimé à partir de l'inclinaison de l'orientation maximale de la région et utilisé afin de prénormaliser l'amas de pixels en rotation. Étant donné que ce n'est qu'une estimation et que les probabilités d'erreurs sont relativement élevées (à cause des fausses détections comme les cheveux par exemple), il est raisonnable d'utiliser une fraction seulement de l'angle estimé et ainsi réduire les effets d'une trop grande rotation. Certains résultats vérifiant cette affirmation seront présentés à la section 2.4.

Il est par ailleurs très peu probable que le gabarit soit de la même taille que l'objet à détecter, ce qui rend une normalisation d'échelle incontournable. Il est donc redimensionné en respectant son ratio d'origine afin d'obtenir une largeur égale à une fraction de la largeur moyenne du *blob* traité. Cette valeur est fixée à 90% et dépend notamment du gabarit utilisé[8].

[7] L'excentricité est calculée à partir des deux axes d'une ellipse, soit $\frac{\sqrt{a^2-b^2}}{a}$ où $a$ est l'axe principal et $b$ l'axe secondaire.

[8] Le visage moyen illustré à la figure 2.2a est légèrement plus étroit que ceux qui seront détectés (c.-à-d. : à cause des oreilles, du cou et des cheveux), ce qui justifie l'utilisation d'une plus faible largeur


**Fig.** ***2.2:*** *Gabarits utilisés pour la détection du visage par* template matching. *a) Visage moyen, b) Paire d'yeux, c) Oeil gauche et d) Oeil droit. Ces gabarits ont été générés à partir d'une banque de 40 images de 10 personnes oeuvrant au Laboratoire de Vision et de Systèmes Numériques (LVSN).*

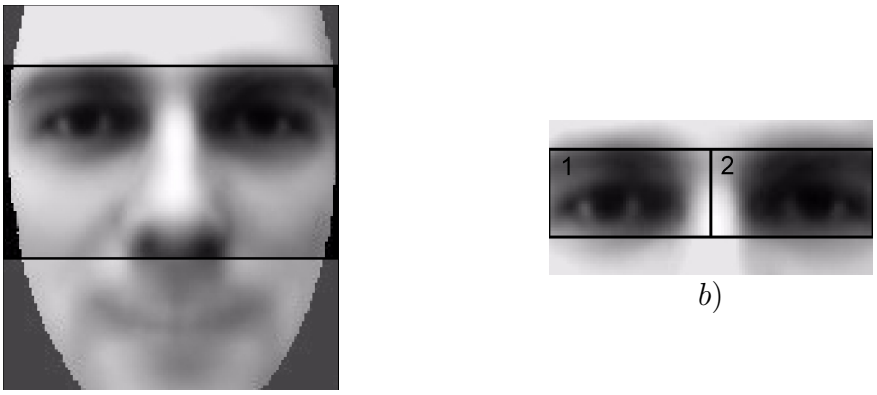
Ainsi, après cette normalisation, le gabarit devrait posséder une taille relativement semblable à celle du visage à détecter. L'algorithme de balayage peut alors débuter afin de déterminer les endroits les plus propices à un appariement. La méthode sélectionnée effectue donc un seul balayage dans l'image au lieu de plusieurs à différentes échelles et rotations. Ceci élimine également l'étape de filtrage des multiples détections.

Finalement, les coordonnées des caractéristiques du visage peuvent être déduites grâce à la position du meilleur appariement de gabarit dans l'image ainsi qu'à partir de leurs positions relatives dans ce gabarit. Cependant, cette technique peut donner de très mauvais résultats, surtout en présence de poses variées. Une méthode de raffinement de la position des yeux est donc proposée afin d'améliorer la précision de cette étape de localisation.

**Raffinement de la position des caractéristiques du visage** Une fois le gabarit du visage positionné (figure 2.2a), une étape de raffinement de la position des yeux en deux phases est réalisée. Pour débuter, un gabarit représentant une paire d'yeux est utilisé (illustré à la figure 2.2b) afin de raffiner la position, notamment en largeur pour ainsi pallier à une légère rotation axiale de la tête.

L'espace de recherche est cependant restreint à une portion du visage trouvé à l'étape précédente. En effet, comme les yeux ne peuvent se trouver au bas du visage,

moyenne.




*a)* *b)*

**Fig.** ***2.3:*** *Zones de recherche utilisées pour le raffinement de la détection des caractéristiques du visage. a) Paire d'yeux et b) Yeux indépendants. Les coordonnées nécessaires au positionnement de ces régions de recherche sont définies de façon relative.*

la zone de recherche est limitée à l'intervalle $[0.15, 0.65]$ de la hauteur et sur toute la largeur de la région. Cette zone de recherche est illustrée à la figure 2.3a et est représentée par un rectangle foncé défini par des coordonnées relatives.

La raison pour laquelle la largeur est balayée en entier repose sur le fait que la tête peut être légèrement tournée, ce qui implique que le gabarit peut se retrouver aux limites de la nuée de pixels. De plus, la tête peut être inclinée vers l'avant (ou vers l'arrière), ce qui justifie l'utilisation d'un rectangle possédant une hauteur plus élevée. Il est important de noter que le gabarit est préalablement redimensionné afin de respecter les dimensions de la région d'intérêt.

La détection du visage peut se terminer après cette première étape de raffinement, processus qui sera dénoté par l'expression "version à 2 gabarits".

Toutefois, une dernière phase de raffinement peut être ajoutée. Ce double processus de raffinement est représenté par la notation "version à 4 gabarits". La position du gabarit précédent est utilisée comme zone de recherche de base. Cette dernière est restreinte à l'intervalle $[0.2, 0.75]$ de la hauteur et sur toute la largeur de la région.

Un oeil est donc recherché dans la moitié gauche du cadre couvert par le deuxième gabarit (2.3b zone 1) et un autre dans la partie de droite (2.3b zone 2). La position verticale est également ajustée grâce à cette dernière étape, palliant ainsi à certaines erreurs de rotation. Les gabarits utilisés sont par ailleurs illustrés aux figures 2.2c et d.

**Transformée inverse des coordonnées** La toute dernière étape de la détection du visage consiste à déterminer les coordonnées des yeux à partir des résultats obtenus dans les phases précédentes. Les positions relatives trouvées lors de l'étape de raffinement sont donc d'abord converties dans le référentiel de la paire d'yeux et ensuite dans celui du gabarit du visage.

Ces coordonnées subissent alors une transformation inverse pour corriger l'effet de la prénormalisation en rotation. Pour terminer, les coordonnées du point d'ancrage de la nuée de pixels sont utilisées afin de convertir la position des yeux dans le référentiel global de l'image.

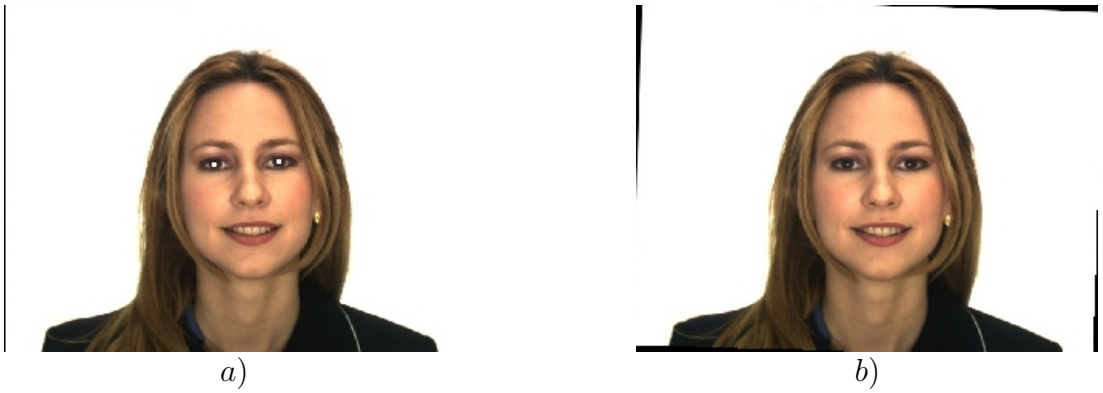
### 2.3.2 Normalisation

La dernière tâche accomplie par le module de détection du visage consiste à normaliser le visage détecté afin de le rendre compatible au format utilisé lors de l'apprentissage. Cette phase de normalisation est principalement basée sur les coordonnées des yeux obtenues à l'étape précédente. Le choix des yeux est justifié par leur invariance en position pour une même personne[9].

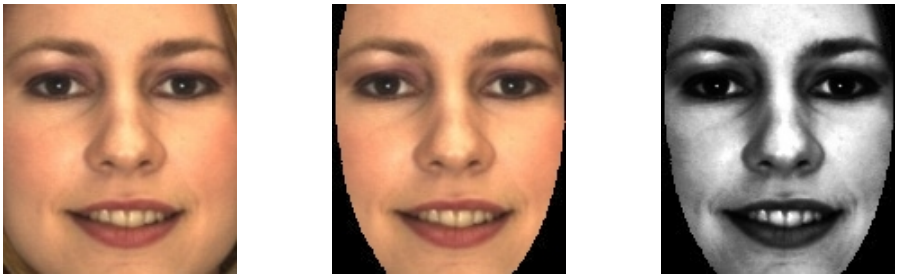
Cette normalisation vise à corriger la rotation, l'échelle, l'illumination ainsi qu'à retirer l'arrière-plan et les cheveux de la personne. Cela étant dit, toutes ces opérations sont réalisées directement sur l'image originale et seront décrites dans les paragraphes suivants. La figure 2.4 illustre par ailleurs le processus complet de normalisation à l'aide d'une image[10] illustrée en 2.4a. Cette dernière représente une image non normalisée ainsi que les coordonnées des yeux qui sont illustrées par des points blanc.

**Rotation** La première opération vise à corriger l'effet produit par une rotation de la tête (c.-à-d. : avec un axe de rotation perpendiculaire à l'image). Le traitement consiste alors à effectuer une rotation de l'image pour aligner les centres des yeux sur une même rangée de pixels. Pour ce faire l'angle de rotation est calculé et l'image est tournée par référence à l'oeil gauche. Il est à noter que certains pixels sont perdus lors de la

[9] Cette valeur perd cependant de son intérêt lorsqu'en présence d'une trop grande rotation axiale.
[10] Cette image provient de la banque d'images AR-face [41].



*a)* *b)*



*c)* *d)* *e)*

**Fig.** ***2.4:*** *Normalisation du visage. a) Image originale, b) Rotation, c) Redimensionnement, d) Masquage et e) Égalisation d'histogramme. Les images a et b sont de dimensions 768×576 pixels contre 130×150 pixels pour les images c, d et e.*

rotation car ils se retrouvent à l'extérieur de l'image. Un exemple de résultat pour cette première phase de normalisation est illustré à la figure 2.4b.

**Mise à l'échelle (*scaling*)** La distance entre les yeux est un paramètre établi à l'avance qui permet le calcul du ratio de mise à l'échelle nécessaire. Dans les expérimentations réalisées, cette distance est fixée à 70 pixels exactement. La largeur entre les yeux détectés est donc mesurée afin de calculer un facteur d'agrandissement (celui-ci peut être inférieur à 1, ce qui représente une réduction de la taille de l'image).

L'image est par la suite redimensionnée à l'aide de cette valeur selon un algorithme d'interpolation cubique tout en respectant son ratio hauteur/largeur d'origine. Un exemple de résultat pour cette opération est illustrée à l'image 2.4c. Il est important de noter que les dimensions de l'image normalisée sont beaucoup plus faibles que celle d'origine, soient 130×150 pixels contre 768×576 pixels.

**Masquage (*cropping*)** L'avant-dernière étape, dont un exemple est illustré à la figure 2.4d, consiste à appliquer un masque ovale sur le visage. Cette opération élimine les cheveux, les oreilles, les vêtements de la personne ainsi que l'arrière-plan de l'image.

Le masque est centré horizontalement selon le point milieu entre les yeux et verticalement afin d'obtenir les yeux à une rangée spécifique. Dans les expérimentations réalisées, la rangée utilisée est située à 45 pixels de hauteur, et ce, pour une image de dimensions 130×150 pixels.

**Illumination** La dernière opération de normalisation consiste à corriger l'intensité de l'image préalablement convertie en tons de gris. Ceci est accomplie à l'aide d'un algorithme d'égalisation d'histogramme [18]. L'objectif principal est d'utiliser toute la plage dynamique disponible (donc tous les tons de gris) afin d'améliorer le contraste de l'image.

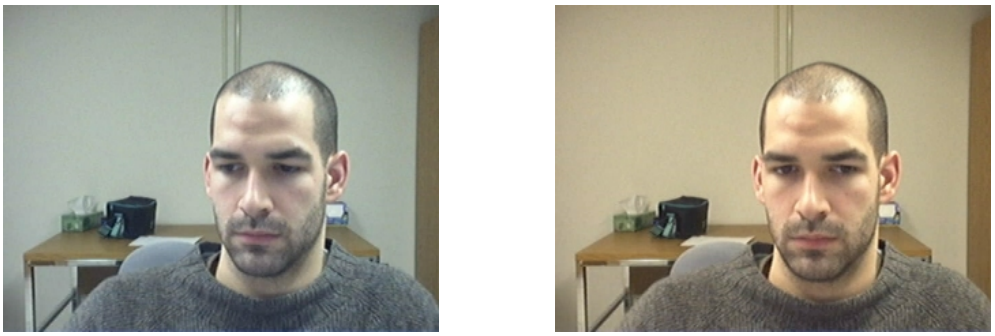
L'image 2.4e représente finalement un exemple de résultat généré par cette dernière phase de normalisation. Il est important de noter que les pixels noirs situés à l'extérieur du masque ne sont pas comptabilisés dans le calcul de l'histogramme.

## 2.4 Résultats expérimentaux

La présente section porte sur différentes expérimentations réalisées dans le but d'évaluer la performance et la robustesse de la méthode développée. Les premiers résultats portent sur l'importance du choix des différents paramètres de la détection de la peau, étape cruciale pour le bon fonctionnement de la méthode.

Certains exemples de détection du visage seront ensuite présentés. Ceux-ci ont été réalisés à l'aide d'images sources représentant des cas particuliers susceptibles d'occasionner des erreurs de localisation.

Cela étant dit, dû à sa forte influence sur la phase d'identification, il est primordial d'effectuer une analyse détaillée de la précision du module de détection du visage. Celle-ci sera donc présentée à la toute fin de la présente section.



*a*) *b*)

**Fig.** ***2.5:*** *Exemple d'ajustement automatique des couleurs : éclairage fluorescent. Correction de type a)* auto white balance *et b)* fluorescent.

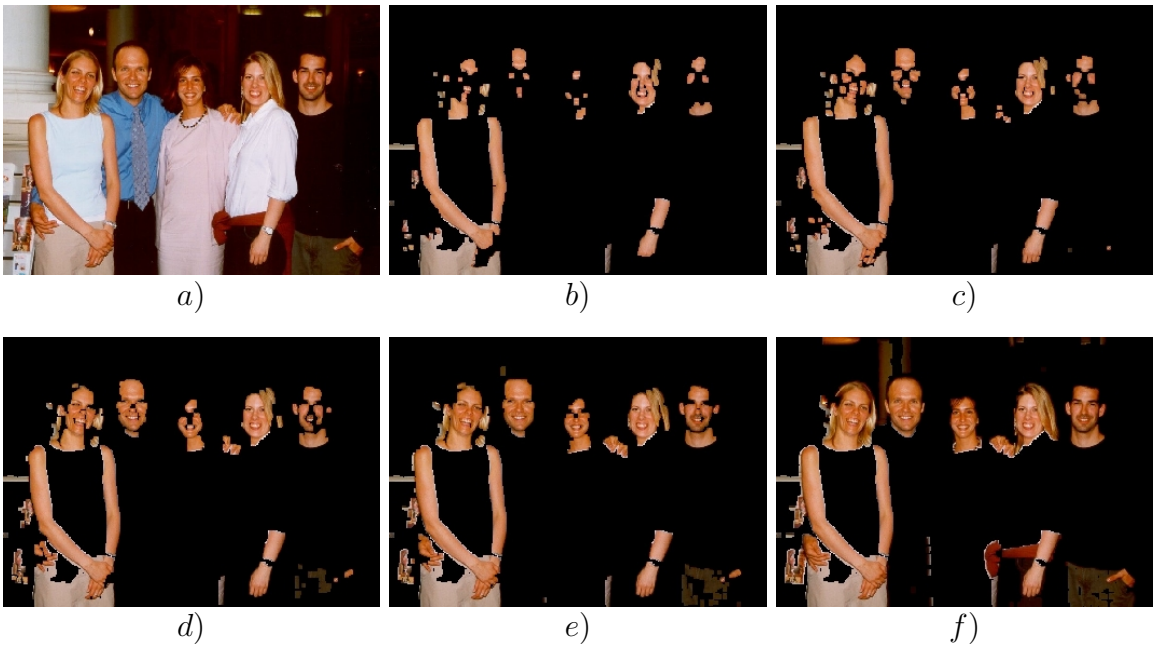
**Détection de la peau** L'extraction des pixels représentant la peau par HSV demeure une technique très efficace, mais qui dépend toutefois des seuils utilisés ainsi que de la qualité des images d'entrées. En effet, pour différentes images, les seuils optimums ne seront pas nécessairement les mêmes, entre autre pour la saturation.

La caméra utilisée pour l'acquisition influence également les résultats. En effet, les caméras n'offrent pas des couleurs parfaitement identiques de l'une à l'autre et d'un modèle à l'autre. Il ne faut pas oublier de plus que les conditions ambiantes lors de l'acquisition peuvent altérer les couleurs (p. ex. : fluorescent *vs* incandescent).

Certaines caméras (p. ex. : *webcam* Logitech® QuickCam® Pro 3000) offrent cependant des ajustements de couleurs (*white balance*) selon les conditions ambiantes (p. ex. : scène ensoleillée ou ennuagée, sous une lumière incandescente ou fluorescente, *etc.*). Ces paramètres peuvent ainsi assurer une certaine répétabilité des résultats peu importe les conditions et permet également d'obtenir des couleurs plus fidèles.

La figure 2.5 illustre notamment un exemple de correction[11] automatique de couleurs appliquée sur des images acquisent dans une scène contenant un éclairage fluorescent. Alors que l'image 2.5a représente un ajustement de type *auto white balance*, l'image 2.5b contient l'image corrigée avec le type *fluorescent*, qui correspond aux conditions ambiantes de la scène observée. Cette dernière reflète adéquatement la couleur peau qui semble moins terne et verdâtre que dans la première image.

[11] Ces corrections sont disponibles grâce aux pilotes fournis avec la caméra utilisée pour les expérimentations.



*a)* *b)* *c)*

*d)* *e)* *f)*

**Fig. *2.6:*** *Détection de la peau : impacts du seuil utilisé pour la saturation. a) Image originale, b) Extraction de la peau avec saturation* $[0.2, 0.68]$*, c) saturation* $[0.2, 0.78]$*, d) saturation* $[0.2, 0.88]$*, e) saturation* $[0.2, 0.99]$ *et f) saturation* $[0.2, 1]$*.*

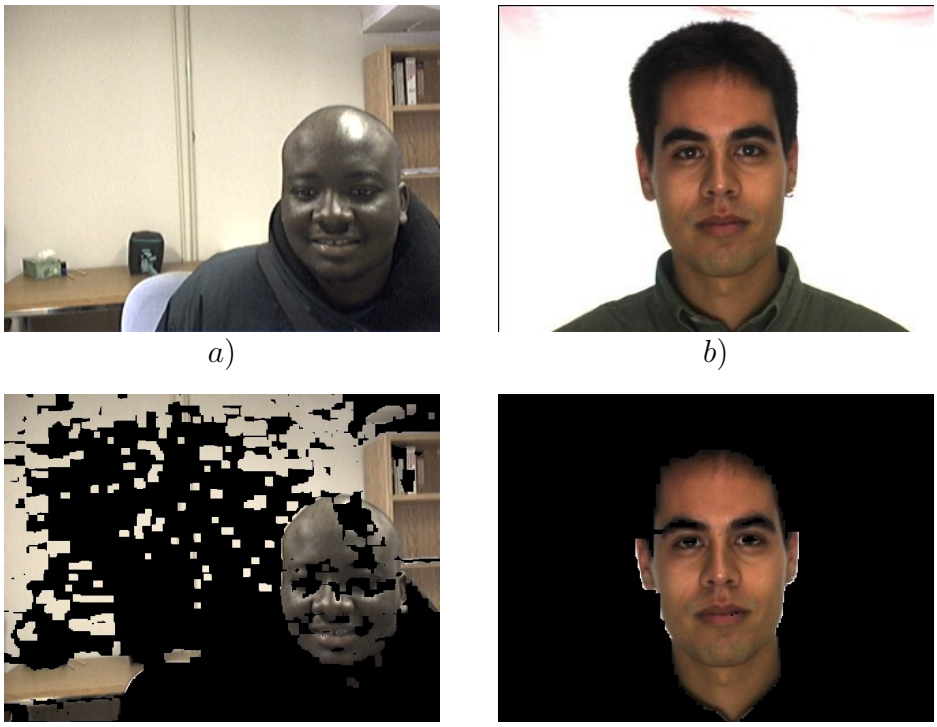
Cela étant dit, la figure 2.6 illustre[12] l'impact de différents seuils de saturation utilisés sur les résultats de la détection de la peau. Les seuils associés au canal *Hue* sont les mêmes pour toutes les images, soient l'intervalle $[0°, 27°]$.

Les résultats semblent démontrer que plus le seuil sur la saturation augmente, plus l'extraction est efficace (c.-à-d. : le nombre de pixels représentant la peau correctement détectée est plus élevé). Cette amélioration se fait cependant au détriment d'un nombre de fausses détections plus élevé.

L'image 2.6f illustre par exemple les effets négatifs produits par l'absence de seuil supérieur pour la saturation. Il y a effectivement une quantité élevé de fausses détections supplémentaires par rapport à l'image 2.6e, notamment pour les vêtements, les cheveux ainsi que pour l'arrière-plan.

Il est important de noter que cette technique de détection de la peau fonctionne également avec des peaux de couleur foncée. La figure 2.7 en illustre notamment deux exemples. Alors que les images 2.7a et b représentent les versions originales, celles en

[12] L'image de base de cette figure provient d'une numérisation de photo prise à l'aide d'une caméra 35mm conventionnelle.



*a)* *b)* *c)* *d)*

**Fig.** ***2.7:*** *Exemple d'extraction des pixels représentant la peau : peaux de couleur foncée. a) b) Images originales, c) et d) Extraction de la peau. Seuils utilisés : c)* $H\ [0°, 30°]$ $S\ [0.12, 0.59]$ *et d)* $H\ [0°, 27°]$ $S\ [0.20, 0.78]$

2.7c et d illustrent quant à elles les résultats de la détection de la peau. Ces images démontrent bien l'efficacité de la méthode pour détecter la peau malgré les couleurs foncées.

Les seuils doivent être cependant légèrement plus permissifs[13] pour obtenir de tels résultats (surtout pour l'image 2.7a), ce qui occasionne plusieurs fausses détections. C'est ce qui se produit dans l'image 2.7c pour une grande majorité du mobilier ainsi que pour une bonne partie du mur. Ces erreurs de détection viennent justifier la présence d'un module de soustraction de l'arrière-plan qui élimine les fausses détections sur des zones sans intérêt.

**Détection du visage** La figure 2.8 illustre certains résultats de détection du visage. Les images originales ont été extraites à partir de séquences vidéos, ce qui permettait une soustraction de l'arrière-plan. Alors que les images 2.8a, c, e et g représentent

[13]Figure 2.7c : Augmentation du seuil supérieur de 3° pour le *Hue* et abaissement de 0.08 du seuil inférieur de la saturation. Figure 2.7d : Augmentation du seuil supérieur de 0.19 pour la saturation.

les images originales, les images 2.8b, d, f et h illustrent les résultats de la détection. Dans tous les cas, les images résultats contiennent plusieurs informations à propos de la détection et illustrées sous forme de rectangles, de points et de lignes.

Tout d'abord, l'image ne contient que les pixels étiquetés comme étant de la peau. Les rectangles bleus de plus grande dimension représentent ensuite l'espace de recherche généré par la première phase de *template matching*. Le second rectangle, quant à lui, représente le meilleur emplacement du gabarit contenant la paire d'yeux. Les points centraux des yeux détectés lors de la phase de raffinement sont illustrés par deux cercles pleins de couleur blanche. Un dernier point blanc représente finalement le centre de masse de la nuée de pixels alors que les deux lignes bleues associées représentent ses orientations maximales.

Parmi les exemples de la figure 2.8, certains représentent des conditions très particulières. L'image 2.8a contient premièrement une personne portant un casque d'écoute qui pourrait nuire à la détection du visage. Or, cet objet est facilement éliminé grâce à l'étape d'extraction de la peau.

Les images 2.8d et f illustrent quant à elles des cas où la détection de peau est flouée par l'éclairage (au nord-ouest de la tête). Malgré ces mauvaises conditions, le processus de détection du visage réussi quand même à bien localiser les yeux dans les deux images.

Pour terminer, l'image 2.8g contient une personne avec les yeux fermés, qui possède une moustache ainsi que des cheveux mi-longs et portant des lunettes. Toutes ces particularités pourraient fort bien nuire au processus de détection. Or, l'image 2.8h démontre que la méthode hybride permet une détection adéquate du visage et de ses composantes malgré des éléments manquants et/ou gênants.

Ces résultats viennent appuyer et démontrer la propriété de recherche multi-échelle de l'approche retenue. En effet, les différents exemples de la figure 2.8 contiennent des visages de tailles variées, ce qui n'influence pas les résultats de détection.

**Détection du visage : invariance aux rotations** Les prochains résultats expérimentaux concernent la robustesse de la méthode sélectionnée quant aux différentes

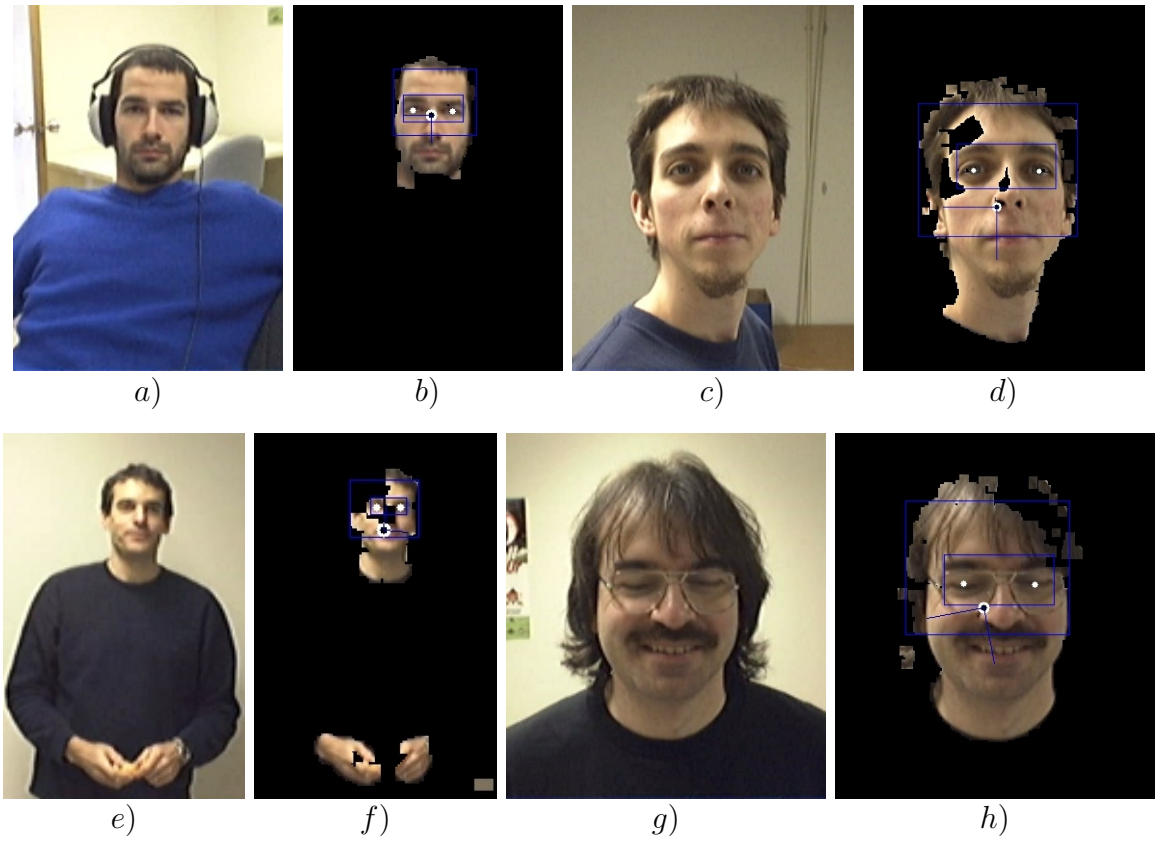
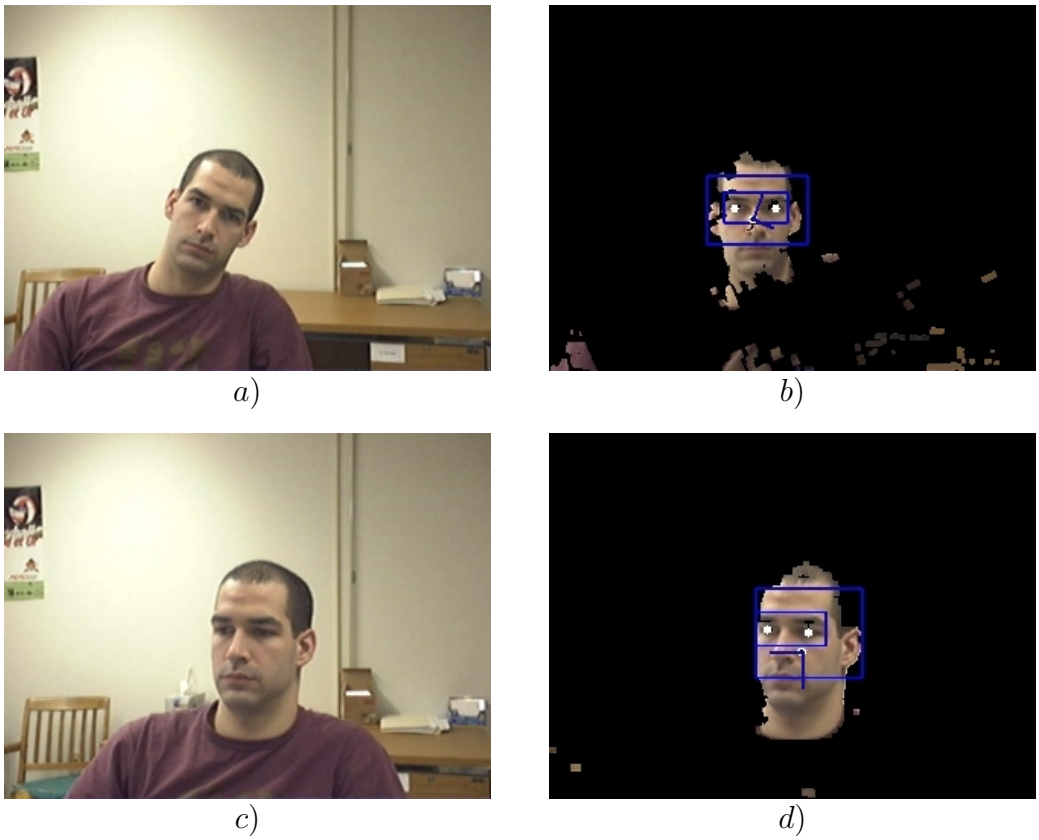



**Fig. *2.8:*** *Détection du visage : résultats expérimentaux. a) c) e) et g) Image originales, b) d) f) et h) Résultats de la détection du visage. Les rectangles bleus représentent les zones de recherche des gabarits alors que les points blancs illustrent les coordonnées des yeux détectés. Le point blanc et les deux lignes bleues représentent respectivement le centre de masse et les orientations maximales de la nuée de pixels.*



*a)* *b)*

*c)* *d)*

**Fig.** ***2.9:*** *Exemple de détection du visage : invariance aux rotations. a) Image originale contenant une rotation dans le plan image, b) Résultat de détection du visage, c) Image originale contenant une rotation axiale formée par la tête et le corps, et d) Résultat de détection du visage. Il est important de noter que l'image b est prénormalisée en rotation, ce qui explique la différence d'angle avec l'image a.*

rotations possibles de la tête. En effet, deux types de rotations doivent être envisagées, soient la rotation de la tête dans le plan image (illustrée à l'image 2.9a) et la rotation axiale (illustrée à l'image 2.9c).

Tout d'abord, l'image 2.9b présente les résultats obtenus à partir du premier type de rotation, soit celle réalisée dans le plan image. Il est important de noter que cette image ne représente pas l'angle de rotation original de la tête, mais plutôt le résultat de la prénormalisation selon l'angle estimé. Les vecteurs d'orientations (illustrés par deux lignes bleues) indiquent clairement l'orientation de la nuée de pixels avant la prénormalisation. Cela étant dit, les yeux sont correctement détectés malgré l'angle de rotation élevé de la tête.

Pour ce qui est du deuxième type de rotation, soit celle relative à l'axe formé par la tête et le corps, les résultats sont présentés à l'image 2.9d. Cette figure illustre le degré

| Particularités | Valeurs |
|---|---|
| **Taille de la banque** | 416 images |
| **Dimensions des images** | 768×576 pixels |
| **Format** | Couleurs 24 bits |
| **Dimensions approximatives des visages** | 300×400 pixels |
| **Dimensions approximatives d'un oeil** | 60×30 pixels |
| **Proportion de femmes** | 36% |

**Tab.** ***2.3:*** *Caractéristiques du sous-ensemble d'images de la banque AR-face utilisé pour évaluer la précision du processus de détection du visage.*

de rotation maximal toléré par la technique de détection du visage. Au-delà de cette limite, les yeux ne sont plus clairement visibles.

**Détection du visage : précision** Afin d'évaluer la précision de l'approche hybride sélectionnée, certaines expérimentations ont été réalisées à l'aide d'une partie de la banque d'images AR-face [41]. Celle-ci sera décrite au chapitre 3 avec moult détails. Par ailleurs, pour favoriser la compréhension, le tableau 2.3 résume les principales caractéristiques du sous-ensemble d'images sélectionnées pour l'expérience.

Ce sous-ensemble d'images contient 416 images couleurs de dimensions 768×576 pixels. Ces images contiennent des hommes et des femmes avec un ratio de 36% pour la gente féminine. Il est à noter que les images ne possèdent pas d'arrière-plan (c.-à-d. : rideau blanc).

Le principe général de cette analyse de précision repose essentiellement sur une comparaison des résultats obtenus d'une part avec la méthode automatique développée et, d'autre part, avec la technique manuelle (c.-à-d. : caractéristiques étiquetées à la main par un être humain). Le protocole expérimental utilisé se résume alors comme suit :

1. Identifier manuellement les coordonnées des yeux pour toutes les images de la banque ;
2. Détecter les visages automatiquement afin de générer les coordonnées d'emplacements des yeux ;
3. Calculer la distance euclidienne (erreur) entre les coordonnées détectées par le

| Méthodes | | Oeil gauche | Oeil droit | Nb. Filtrées |
|---|---|---|---|---|
| **Méthode de base (2 gabarits)** | Sans rotation | 15.2 | 13.9 | 26 (6.2%) |
| | Rotation ($\frac{\alpha}{2}$) | 14.2 | 14.6 | 31 (7.4%) |
| | Rotation ($\alpha$) | 14.2 | 16.2 | 38 (9.1%) |
| **Avec raffinements (4 gabarits)** | Sans rotation | 13.7 | 12.4 | 29 (6.9%) |
| | Rotation ($\frac{\alpha}{2}$) | 13.1 | 12.6 | 32 (7.7%) |
| | Rotation ($\alpha$) | 13.0 | 13.3 | 39 (9.3%) |

**Tab.** ***2.4:*** *Précision de la méthode de détection du visage. Les résultats sont rapportés pour deux variantes de la technique ainsi que pour trois types de prénormalisation en rotation. Les erreurs sont mesurées en pixels alors que le nombre d'images filtrées représente la quantité d'images rejetées par rapport à la taille totale de la banque (c.-à-d. : 416 images).*

processus automatique et celles qui sont connues ;

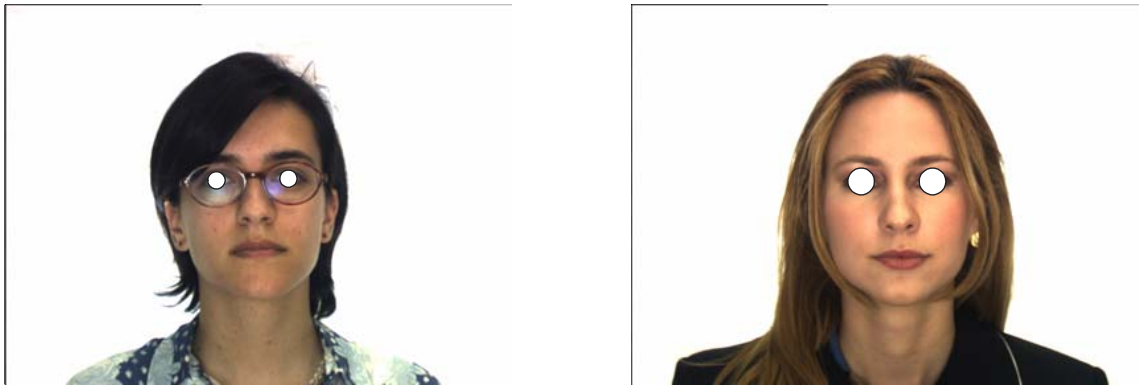
4. Calculer la moyenne des erreurs de localisation pour chacune des caractéristiques d'intérêt.

Afin d'évaluer l'efficacité relative des différentes variantes de l'approche sélectionnée, quatre type d'expérimentations ont été réalisées. Tout d'abord, nous distinguons deux groupes de tests, soient ceux utilisant un raffinement de la position (versions à 4 gabarits) et ceux n'en utilisant pas (versions à 2 gabarits). Dans cette dernière catégorie, les coordonnées des yeux sont déduites à partir des positions relatives des composantes sur le gabarit contenant la paire d'yeux.

L'efficacité de l'estimation de la prénormalisation en rotation est ensuite évaluée pour chacune de ces familles et ce, pour deux angles différents $\alpha$ et $\frac{\alpha}{2}$ (où $\alpha$ est l'angle estimé à partir de l'analyse des moments). Pour toutes ces expérimentations, les détections très erronées (erreurs de plus de 65 pixels pour l'oeil gauche *et* l'oeil droit) sont filtrées et ne sont pas comptabilisées dans la moyenne.

Les taux d'erreurs expérimentaux sont illustrés au tableau 2.4. Les meilleures performances sont obtenues en utilisant la méthode avec raffinements et prénormalisation de la rotation selon un angle de $\frac{\alpha}{2}$ ; une combinaison qui atteint une erreur moyenne légèrement inférieure à 13 pixels pour les deux yeux.

Cela étant dit, la figure 2.10 illustre les zones d'erreurs engendrées par la technique choisie. L'image 2.10a contient tout d'abord deux cercles de 13 pixels de rayon cor-



*a)* *b)*

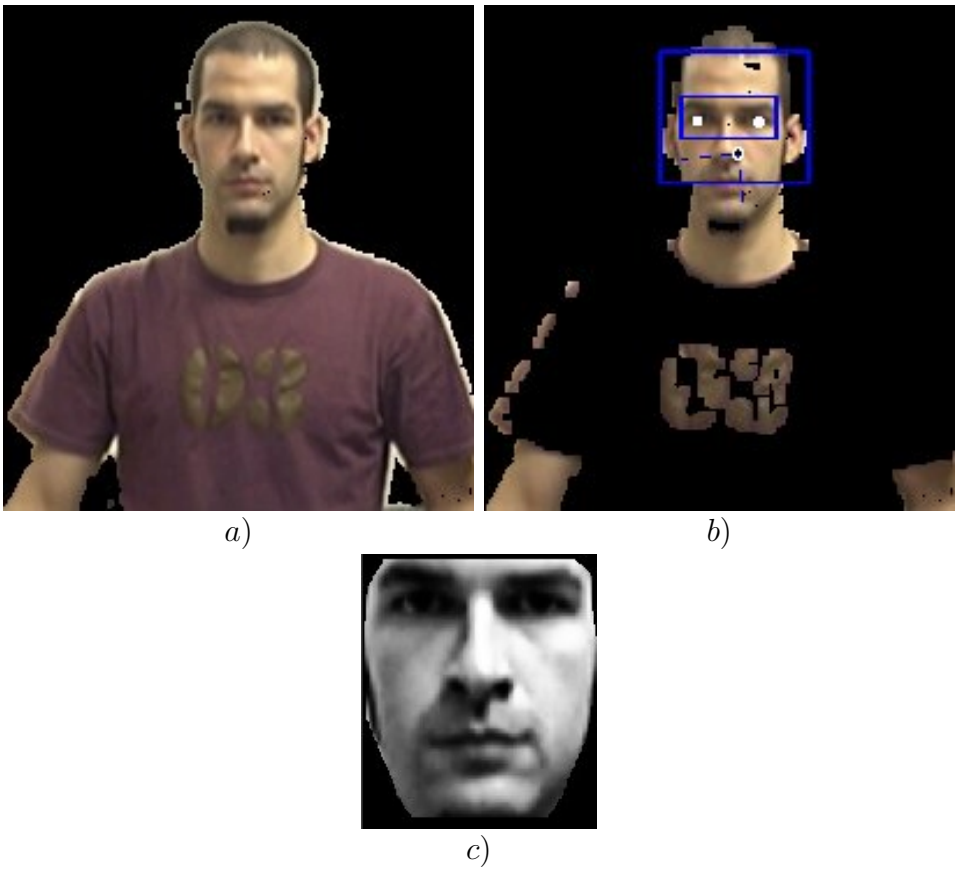
**Fig.** ***2.10:*** *Limites de précision du processus de détection du visage. Zones d'erreurs possédant un rayon de a) 13 pixels et b) 20 pixels.*

respondant à l'erreur moyenne obtenue. À titre indicatif, l'image 2.10b représente des cercles de 20 pixels de rayon.

Un point important à considérer pour l'analyse de ces résultats provient du pourcentage d'images de femmes contenues dans la banque (36%). En effet, relativement aux hommes, une plus forte proportion des femmes possèdent des cheveux longs qui, dépendant de leur couleur, peuvent être considérer comme de la peau et ainsi fausser le redimensionnement des gabarits. L'estimation de l'orientation des visages peut en être aussi fortement affectée.

Il est également intéressant de remarquer la légère supériorité des méthodes utilisant une étape de raffinement supplémentaire. Une prénormalisation de la rotation avec l'angle $\alpha$ génère par contre un plus haut taux d'images incorrectement identifiées. Pour terminer, la précision obtenue peut être jugée suffisante si l'on considère que les dimensions des yeux dans l'image sont d'approximativement de 60×30 pixels.

**Détection du visage : normalisation** La figure 2.11 illustre un processus de détection et de normalisation complet à partir d'une séquence vidéo. En premier lieu, la soustraction de l'arrière-plan est appliquée sur une image et illustrée à la figure 2.11a. Le visage est ensuite détecté (2.11b) à partir de l'image contenant les pixels peau pour être finalement normalisé selon les coordonnées des yeux en rotation, en échelle et en illumination (2.11c). Cette toute dernière image est alors fin prête pour la phase d'identification.



*a*) *b*)

*c*)

**Fig.** ***2.11:*** *Exemple complet de détection du visage. a) Détection du mouvement par soustraction de l'arrière-plan, b) Détection de la peau et du visage par appariement de gabarits et c) Visage normalisé.*

## 2.5 Discussion

La prochaine section porte sur une énumération des différentes forces et limitations de l'approche développée. Certaines pistes d'amélioration seront également abordées à la sous-section 2.5.3.

### 2.5.1 Forces et avantages

**Réduction de l'espace de recherche** Une des contraintes majeures du système est la conception d'une application fonctionnant en temps réel, c'est-à-dire d'avoir la capacité d'effectuer tout le traitement en-ligne sans opérations réalisées hors-ligne. Pour atteindre cet objectif, chaque module doit être optimisé pour ainsi réduire le temps d'exécution au maximum.

La méthode développée possède plusieurs étapes qui visent à faire des économies d'opérations, notamment en réduisant l'espace de recherche utilisé pour la détection du visage. En effet, une recherche des zones susceptibles de contenir un visage est réalisée avant d'effectuer la détection du visage proprement dite.

Dans un premier temps, la procédure de détection de la couleur peau contribue à filtrer et à rejeter plusieurs pixels de l'image d'avant-plan qui ne font pas partie d'un visage. L'utilisation de plusieurs étapes de raffinement permet ensuite d'éliminer plusieurs opérations d'appariement de gabarits qui, à plusieurs échelles et pour différentes poses, représentent un temps d'exécution cumulatif non négligeable.

**Invariance à l'échelle** L'étape précédente de réduction de l'espace de recherche vient contribuer indirectement à un autre objectif visé par le système, soit la capacité de localisation de visage de taille variable.

Les conditions d'acquisition ne permettent pas la connaissance à priori des dimensions des visages recherchés. L'étape d'identification de la couleur peau procure cependant certaines informations qui permettent d'orienter la détection du visage. Notons entre autres l'estimation de la taille et de l'orientation du visage obtenues grâce aux

dimensions et aux orientations maximales des nuées de pixels.

Un souci de traitement multi-échelle est également apporté tout au long du processus de détection, notamment par les nombreux redimensionnements visant à améliorer la localisation des caractéristiques du visage.

**Localisation des caractéristiques** L'objectif premier visé par la phase de détection du visage est bien entendu la localisation de visages dans une image d'entrée. Cependant, cette information ne suffit pas à une identification adéquate des individus et ne peut orienter directement le processus de normalisation.

Pour remédier à ce problème, des caractéristiques plus précises doivent être utilisées afin d'accomplir une normalisation juste et efficace du visage. Les yeux représentent donc ces points d'intérêt.

La technique développée utilise des phases de raffinement qui visent à améliorer la précision de la localisation des yeux. Ceci est accomplie à l'aide de plusieurs gabarits utilisés par des phases incrémentales de *template matching*. Leur utilisation permet ainsi la réduction de l'erreur réalisée sur les coordonnées des yeux.

**Pose du visage** Les propriétés du projet impliquent que le système d'identification ne nécessite pas des conditions particulièrement contrôlées (p. ex : individu devant se placer face à un appareil de reconnaissance). Cela étant dit, la pose du visage est limitée à une pose frontale qui est valide tant que les yeux sont visibles par la caméra.

Certains exemples présentées à la figure 2.9 illustre notamment la robustesse de la méthode choisie face à des rotations variées, tant axiale que dans le plan image.

### 2.5.2 Limitations

Il serait utopique de penser qu'une méthode soit parfaite dans toutes les situations. Ainsi, certains cas particuliers peuvent parfois ne pas être traités adéquatement et mener à une détection erronée.

**Détection de la peau** Le processus de détection de la peau présente à lui seul deux limitations importantes. La première repose sur l'utilisation d'images couleurs, qui sont essentielles au bon fonctionnement de tout le module. Ce faisant, la méthode développée ne peut fonctionner avec des images en tons de gris.

Les fausses détections représentent évidemment la principale faiblesse du processus d'extraction des pixels associés à la couleur peau. En effet, les images 2.1d à f illustrent certains problèmes de détection, notamment en ce qui à trait aux cheveux ou à des vêtements de couleur inclue dans la portion de l'espace HSV représentant la peau.

Ces erreurs de détections influencent directement le redimensionnement des gabarits utilisés dans les phases ultérieurs, ce qui occasionne des localisations peu précises. Il est évidemment envisageable de restreindre la portion de l'espace HSV afin de réduire la quantité de couleurs permises. Cette solution n'est cependant pas viable car plusieurs types de peau seraient ignorés par le détecteur.

Pour terminer, la phase de détection de peau requiert des images contenant des couleurs relativement fidèles à la réalité. La figure 2.5a présentée auparavant, illustre notamment des couleurs peu représentatives qui nuisent considérablement à la qualité des résultats générés.

**Pose du visage** Il a été mentionné précédemment que la technique de détection du visage est robuste à des rotations variées. Les images contenues à la figure 2.9 représentent cependant les conditions limites supportées par le module de détection.

Une fois ces limites dépassées, la localisation et l'identification du visage n'est pas garantie. En effet, la normalisation du visage repose entièrement sur les coordonnées des yeux et échouera donc lamentablement si l'un d'entre eux est absent. Il est cependant important de noter que la détection des yeux peut quand même réussir lorsque ceux-ci sont fermés (figure 2.8h).

### 2.5.3 Améliorations possibles

**Détection de la couleur peau** L'étape d'extraction des pixels pourrait subir quelques améliorations concernant les fausses détections. Les cheveux et les vêtements représentent les sources principales d'erreurs qui influencent l'estimation de la taille du visage.

Pour remédier à cette problématique, il serait intéressant d'utiliser des techniques de segmentation en régions[14] pour identifier le visage et les parties gênantes. Cette segmentation permettrait alors d'éliminer complètement les cheveux et les vêtements, toujours en autant que les parties soient séparables.

**Localisation des caractéristiques : raffinement** Certaines améliorations peuvent être apportées à la toute dernière étape de détection des caractéristiques. Il serait effectivement envisageable d'effectuer deux appariements de gabarits centrés sur les coordonnées des yeux détectés. Cela pourrait être réalisé pour différentes échelles (c.-à-d. : $\pm 5\%$) et orientations (c.-à-d. : $\pm 5°$) dans une zone légèrement plus grande que les gabarits (c.-à-d. : $\pm 15$ pixels). Le meilleur endroit serait celui qui, parmi toutes les possibilités générées, possède la distance minimum par rapport à l'image. L'avantage de cette amélioration réside dans l'exploration de solutions différentes de celles envisagées par les estimations des étapes précédentes.

## 2.6 Conclusion

La détection du visage, étape cruciale d'un système de reconnaissance, représente un défi particulièrement intéressant. Ce module se doit notamment d'être robuste à la pose des visages et aux expressions faciales.

Plusieurs techniques de détection ont été abordées au cours de ce chapitre, chacune possédant ses forces et ses faiblesses. Compte tenu des différentes particularités du projet, une méthode hybride fût développée alliant les atouts de certaines techniques.

[14] La segmentation en régions peut être réalisée en analysant la couleur et la texture de différentes parties de l'image. Le *region growing* est couramment utilisé pour regrouper les pixels en régions.

Pour débuter, l'image de mouvement est prétraitée à l'aide d'un algorithme d'extraction des pixels représentant la peau dans l'espace de couleurs HSV. Les nuées de pixels sont ensuite localisées et analysées, afin d'estimer les différents paramètres des visages potentiels.

Une approche d'appariement de gabarits, ajustée selon les paramètres estimés auparavant, est ensuite utilisée. Ce faisant, un raffinement de la localisation des yeux est réalisé afin d'améliorer la précision des coordonnées détectées. La position des yeux oriente par ailleurs directement le processus de normalisation.

Des résultats expérimentaux ont également été présentés, révélant une performance intéressante dans plusieurs cas spéciaux (p. ex. : yeux fermés, casque d'écoute, éclairage, *etc.*) ainsi que pour différentes rotations du visage.

Cela étant dit, une analyse de la précision de la méthode hybride a été réalisée à l'aide d'un sous-ensemble de la banque d'images AR-face [41] contenant 416 images. Les résultats obtenus suggèrent une performance suffisante dans le cadre du projet.

Pour terminer, l'impact de la détection automatique sera évaluée quantitativement au chapitre 4. Ceci permettra d'évaluer l'influence de la méthode sur le taux de reconnaissance par rapport à une localisation quasi-parfaite (c.-à-d. : manuelle).

# Chapitre 3

# Reconnaissance de l'individu

Une fois la détection de la personne et de son visage complétée, l'image résultante ainsi que les informations pertinentes sont fournies à l'engin de reconnaissance. Celui-ci, composé de plusieurs sous-modules, est basé sur une architecture orientée-objets facilitant la gestion et l'ajout de techniques supplémentaires. Utilisant une banque d'informations centralisée, l'engin de reconnaissance doit tout d'abord effectuer l'apprentissage des sous-modules pour les différentes personnes à reconnaître.

## 3.1 Introduction

Plusieurs techniques de reconnaissance d'individus ont été développées au cours des dernières années. La plupart d'entre elles ont le visage comme zone d'intérêt ; une

tâche qui est par ailleurs un problème de reconnaissance des formes assez complexe. En effet, contrairement à certaines problématiques comme la reconnaissance de caractères manuscrits, le nombre de classes à distinguer est très élevé et chacune d'elle ne possède qu'un nombre restreint d'exemples. Ces conditions particulières nuisent donc à certaines techniques d'apprentissage automatique qui nécessitent un grand nombre de données afin d'apprendre efficacement.

Les techniques de reconnaissance d'individus peuvent essentiellement se diviser en deux grandes catégories, soient celles dites intrusives et celles qui ne le sont pas. Ce faisant, une méthode intrusive requiert la coopération de l'individu pour l'identifier. Ainsi, les empreintes digitales sont un exemple de ce type de classification. Étant donné que ces méthodes débordent du cadre du projet, une très brève énumération des techniques sera présenté à la section 3.2.1.

Pour ce qui est des méthodes non-intrusives, ce sont celles qui peuvent êtres appliquées à distance en observant les individus avec des capteurs, mais sans toutefois requérir leur coopération. À l'intérieur de cette catégorie, nous retrouvons principalement les techniques utilisant la vision numérique (c.-à-d. : spectre visible et infrarouge) en deux et en trois dimensions.

Cela étant dit, les méthodes peuvent être davantage distinguées selon les régions d'intérêt utilisées. Alors que certaines utilisent des informations provenant du corps en entier, plusieurs s'intéressent uniquement au visage. Cette dernière catégorie forme la majorité des techniques d'identification de personnes par vision numérique développées à ce jour.

Malgré le degré de performance satisfaisant atteint par les différents algorithmes de reconnaissance, il demeure néanmoins que des conditions spécifiques sont plus favorables à certaines méthodes, et vice versa. L'utilisation d'un multi-classifieur (MC) alliant les forces de plusieurs techniques semble alors être une solution particulièrement intéressante.

Plusieurs techniques seront donc décrites à la section 3.2, comme entre autres les *EigenFaces* et les réseaux de neurones. Ensuite, la section 3.3 abordera les méthodes privilégiées dans le cadre du projet ainsi que certaines caractéristiques importantes (p. ex. : architecture logicielle du système).

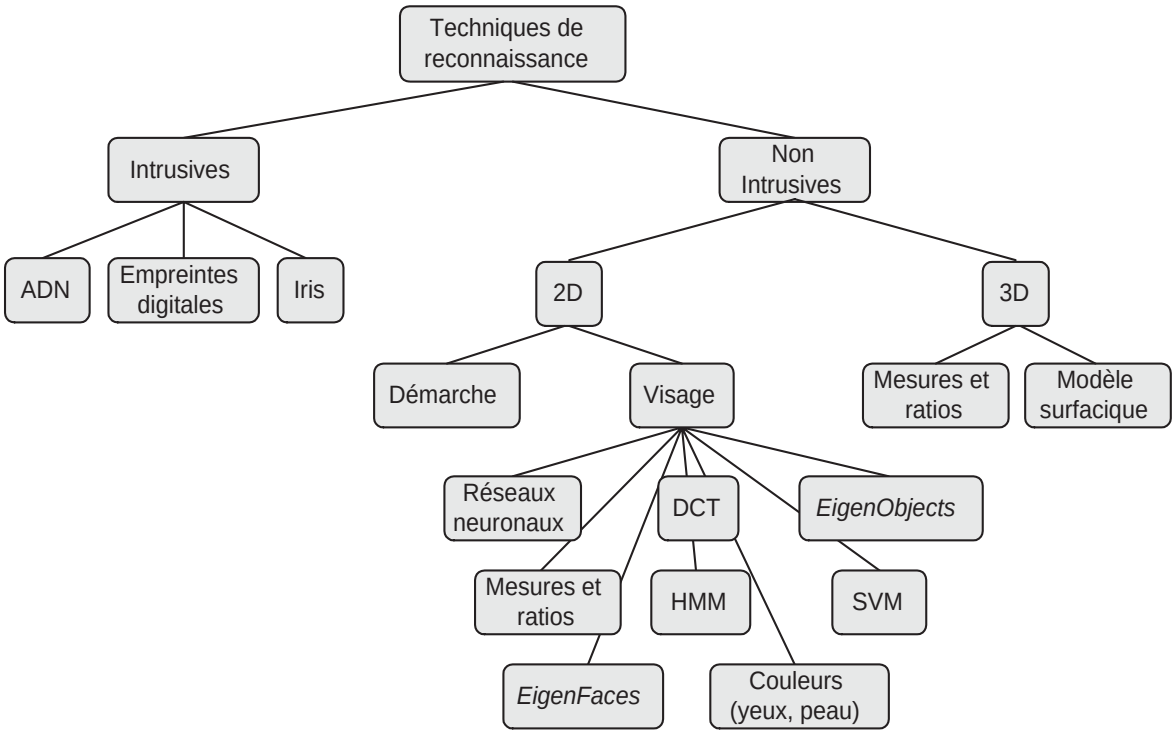



**Fig.** ***3.1:*** *Principales techniques de reconnaissance d'individu.*

Il est important de noter que dû à l'importance cruciale de l'étape de reconnaissance sur la performance du système, toutes les expérimentations y étant reliées seront présentées en détails au chapitre suivant.

## 3.2 Recension des écrits

La reconnaissance de personne peut être accomplie par une très grande variété de méthodes. La figure 3.1 illustre un diagramme représentant une partie[1] de ces techniques. La catégorie contenant les méthodes non-intrusives est la plus volumineuse, dû entre autres au sous-groupe des techniques de reconnaissance du visage. Il est intéressant de remarquer que la plupart de ces méthodes utilisent la vision numérique comme source d'information.

La figure 3.1 contient évidemment un grand nombre de méthodes non appropriées

[1]Cette figure n'a pas la prétention d'illustrer toutes les techniques existantes, mais seulement de représenter les plus importantes ainsi que les principales branches existantes.

au projet et dépassant largement les contraintes fixées au départ. La recension des écrits couvre donc principalement les techniques de reconnaissance du visage par vision numérique. Qui plus est, les méthodes tridimensionnelles et infrarouges sont exclues dû à l'équipement utilisé. De brèves descriptions seront tout de même fournies pour toutes les techniques rejetées.

### 3.2.1 Méthodes de reconnaissance d'individu : intrusives

Parmi les techniques de reconnaissance d'individu disponibles à ce jour, les plus performantes appartiennent sans contredit à la catégorie des méthodes intrusives. Nous y retrouvons entre autres la comparaison d'ADN (*DNA matching*) ainsi que l'identification à partir d'informations biométriques provenant d'empreintes digitales, d'iris, de mains, *etc.*

Malgré leurs taux de reconnaissance impressionnants, ces méthodes restent avant tout intrusives, ce qui implique un contact direct avec les individus à identifier. Cette caractéristique importante ne peut malheureusement pas être acceptable pour tous les projets. En effet, dans un contexte de surveillance où la scène est observée silencieusement et discrètement par un "policier mécanique", les méthodes intrusives ne peuvent être utilisées sans trahir la présence du système ou transgresser davantage la vie privée.

Ces méthodes ne doivent cependant pas être exclues (au contraire) pour le design d'applications de haute sécurité, comme par exemple des accès privilégiés à des bâtiments protégés ou à des ressources particulières.

### 3.2.2 Méthodes de reconnaissance : corps

Lorsqu'une vue du corps en entier est disponible, il est avantageux d'utiliser cette information supplémentaire pour identifier l'individu ou pour tout simplement raffiner le processus d'identification. En effet, lorsque plusieurs techniques d'identification sont simultanément possibles, bon nombre d'individus peuvent être éliminés uniquement à la vue du corps.

Par exemple, si une personne observée est d'une taille de $1.95m$, il est inutile de tenter la reconnaissance sur les individus de plus petite taille présents dans la base de données.

Parmi les méthodes envisageables, il y a entre autres les mesures morphologiques (3D) ainsi que l'analyse de la démarche des personnes.

#### 3.2.2.1 Mesures morphologiques (3D)

Lorsqu'une acquisition tridimensionnelle est possible et que le sujet à reconnaître est couvert totalement ou en partie par le champ de vue de la caméra, plusieurs mesures morphologiques peuvent être prélevées pour des fins d'identification.

Tout d'abord, il est impératif d'effectuer une segmentation robuste et adéquate du corps humain, c'est-à-dire de bien isoler, identifier et positionner chacun des membres de la personne.

Il est par la suite possible d'extraire certaines mesures comme la longueur des bras, des jambes ainsi que la taille de la personne. Toutes ces valeurs peuvent alors être comparées à celles contenues dans la base de données et ainsi aider à éliminer certains candidats.

Cela étant dit, une précision suffisante doit être atteinte pour que ces valeurs soient vraiment discriminantes et pour que des candidats ne soient pas faussement identifiés et/ou exclus. De plus, certains éléments peuvent nuire au bon fonctionnement de cette technique comme par exemple les manteaux amples, les chapeaux, *etc.*

#### 3.2.2.2 Analyse de la démarche (*Gait analysis*)

Il est également envisageable d'observer une personne pendant plusieurs secondes afin de modéliser sa démarche. Plusieurs techniques sont ainsi utilisables, tant au niveau de l'extraction des caractéristiques qu'au niveau de la reconnaissance.

Certains auteurs ont d'ailleurs utilisés des *EigenGait* afin de représenter la démarche des individus [2]. Ces informations sont par la suite jumelées à un classifieur de type *K-ppv* pour la phase d'identification. Des travaux ont aussi été réalisés afin d'intégrer la reconnaissance du visage avec celle de la démarche [54].

### 3.2.3 Méthodes globales de reconnaissance du visage

La première grande famille de méthodes de reconnaissance concernent celles qui utilisent le visage au complet comme source d'information et ce, sans segmentation de ses parties.

Dans la majorité des cas, les images sont représentées par un vecteur de pixels généré par la concaténation de toutes les colonnes de l'image. Ainsi, une image en niveaux de gris de dimensions de 130×150 pixels possédera une représentation vectorielle de 19 500 éléments. Finalement, les couleurs ne sont habituellement pas utilisées par les méthodes globales de reconnaissance, ce qui simplifie un grand nombre d'opérations.

#### 3.2.3.1 Corrélation

La technique de corrélation est basée sur une comparaison simple entre une image test et les visages d'apprentissage. Celui d'entre eux se trouvant à la plus faible distance du visage test sera sélectionné comme premier choix.

Plusieurs métriques peuvent être utilisées afin d'évaluer cette valeur comme par exemple les distances $L_1$ (*city-block*) et $L_2$ (euclidienne), la cross-corrélation, la distance de Mahalanobis, *etc.* Ce processus de décision est communément appelé algorithme du $K$ plus proche voisin ($K$-ppv) et sera présenté avec davantage de détails à la sous-section 3.3.1.5).

Malgré sa grande simplicité, cette méthode n'offre cependant pas d'avantages particulièrement intéressants. En effet, n'utilisant pas d'informations de plus haut niveau, la technique de corrélation offre peu de robustesse face aux expressions faciales, aux variations d'éclairage et aux changements physiques (p. ex. : barbe).

**Fig.** ***3.2:*** *EigenFaces : Image moyenne ainsi que les 5 premiers visages propres. Ces images ont été générées à l'aide d'une banque d'images contenant 10 personnes oeuvrant au LVSN.*

#### 3.2.3.2 *EigenFaces* (EF)

L'utilisation de méthodes statistiques appliquées à la modélisation et à la reconnaissance de visage est largement répandue. Kirby et Sirovich ont d'ailleurs [33] utilisé la transformée K-L (c.-à-d. : Karhunen-Loève) afin de coder des visages et ainsi réduire la dimensionnalité de leur représentation.

En 1991, Turk et Pentland [62] introduisirent le concept d'*EigenFaces* à des fins de reconnaissance. Basée sur une analyse en composantes principales (ACP), la méthode des EF repose sur une utilisation des premiers vecteurs propres comme *visages propres*, d'où le terme *EigenFaces*.

La base formée par ces vecteurs génère alors un espace utilisé pour représenter les images des visages. Les personnes se voient donc attribuer un vecteur d'appartenance pour chacune de leur image.

Cela étant dit, la reconnaissance est réalisée en comparant les coefficients de projection d'un visage test avec ceux appartenant aux visages d'entraînement. La méthode est relativement rapide en phase de reconnaissance et peut également bénéficier de plusieurs optimisations algorithmiques [62].

Des résultats satisfaisants ont par ailleurs été obtenus sur des banques d'images d'envergure [48] ainsi que sur des images infrarouges [57]. Cette méthode faisant partie de l'approche choisie, davantage de détails seront fournies à la section 3.3.1.1.

La figure 3.2 illustre finalement l'image moyenne ainsi que les cinq premiers visages propres associés à la banque d'images LVSN contenant 10 personnes.

#### 3.2.3.3 DCT

L'utilisation de la transformée de cosinus discrète (*Discrete Cosine Transform* ou DCT) [50] à des fins de reconnaissance de visage est assez récente [19]. Similaire aux *EigenFaces* d'un point de vue mathématique, elle est par contre beaucoup plus rapide, tant en phase d'apprentissage qu'en phase de reconnaissance.

Cela étant dit, chaque image de visage est représentée par un vecteur composé des premiers coefficients de la transformée. Lorsqu'un visage est présenté au module, sa transformée est calculée et un certain nombre de coefficients est retenu pour comparaison avec ceux de la banque de données. Cette dernière étape est réalisée à l'aide de la distance $L_1$ ou avec d'autres métriques pertinentes.

L'utilisation de plus en plus massive de techniques de compression multimédia a également favorisé une optimisation des algorithmes de DCT, procurant donc à cette méthode un atout certain. Grâce à ses nombreux avantages, cette technique de reconnaissance a été sélectionnée pour le projet et sera donc abordée en détails à la section 3.3.1.3.

#### 3.2.3.4 Réseaux de neurones

La prochaine technique envisagée utilise des réseaux de neurones comme engin d'apprentissage et de reconnaissance. Pour débuter, une image brute (ou prétraitée) de dimensions fixes constitue habituellement la source d'entrée des réseaux. Les dimensions doivent être établies au préalable car le nombre de neurones sur la couche d'entrée en dépend.

Cela étant dit, plus les dimensions de l'image sont élevées, plus la complexité et le temps d'apprentissage augmentent. En effet, pour une image de dimensions 130×150 pixels, 19 500 neurones seront requis sur la couche d'entrée, ce qui est énorme. L'apprentissage efficace (c.-à-d. : la convergence) d'un tel réseau est également douteux.

Le nombre de sorties du réseau dépend par ailleurs directement de la quantité d'individus à discriminer. Il est donc évident qu'un apprentissage incrémental (avec

de nouveaux individus et non de nouveaux exemples) sera difficile et requerra des ajustements directs à l'architecture du réseau.

Certains auteurs ont par ailleurs utilisé des variantes de la technique de base en modifiant les données d'entrée. Les coefficients de projections d'images dans un espace des visages (*EigenFaces*) peuvent par exemple être utilisés comme source d'information [62]. Cette méthode peut évidemment être étendue aux coefficients de DCT, de Fourier, *etc.*

Les informations peuvent également être fusionnées ensemble avant d'être acheminées au réseau : c'est le cas par exemple de la concaténation d'une image prétraitée (dimensions 30×40 pixels) et de l'échantillonnage d'histogrammes de couleurs RGB [40].

#### 3.2.3.5 Modèle surfacique du visage (3D)

La prochaine méthode de reconnaissance repose sur l'utilisation d'un modèle tridimensionnel du visage. Pour que cette technique soit réellement efficace, une vue rapprochée du visage est nécessaire pour chacune des caméras impliquées dans l'acquisition[2].

Dans certains cas, il est possible de réaliser de la stéréo dense, c'est-à-dire d'extraire un grand nombre de points dans une zone relativement restreinte. Celle-ci garantie alors de meilleures précisions sur les mesures ainsi qu'une résolution accrue. Une fois l'appariement des points réalisé, le modèle peut être normalisé et stocké dans la base de données.

Lorsqu'un individu se présente devant les caméras, la même procédure s'applique, mais suivie d'une étape de comparaison. En effet, le modèle à reconnaître doit être comparé à tous les modèles de la base de données, ce qui représente un travail colossal. Une réduction de calculs est donc impérative pour minimiser la complexité de ce problème d'optimisation.

[2]L'utilisation d'une caméra combinée avec un laser de faible puissance est également envisageable pour réaliser une acquisition tridimensionnelle.

Ceci peut être réalisé en alignant les centres des yeux des deux modèles. Il ne reste alors qu'à mesurer l'erreur entre les deux surfaces. Certains auteurs [1] ont d'ailleurs proposé l'utilisation d'une distance d'Hausdorff modifiée pour réaliser ce calcul.

Il est finalement envisageable de prélever certaines mesures sur le modèle du visage, comme la distance réelle entre les composantes du visage (p. ex. : distance entre les yeux) ou leurs dimensions. Ces informations pourraient être utilisées ensuite pour la reconnaissance, tout comme dans la technique précédente de prises de mesures morphologiques (sous-section 3.2.2.1).

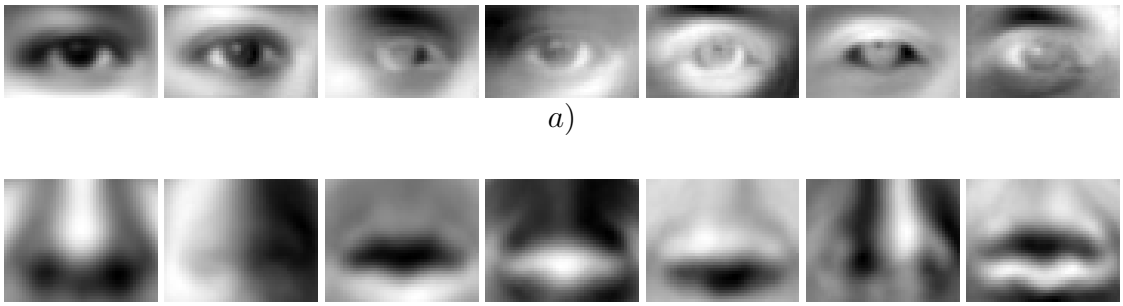
### 3.2.4 Méthodes locales de reconnaissance du visage

Le principal désavantage des méthodes globales réside au niveau de détails utilisé. En effet, lorsqu'une technique s'attarde aux variations dans toute une image, elle tentera de limiter l'impact des changements locaux et concentrera le maximum d'énergie pour représenter adéquatement l'ensemble de l'image (p. ex. : *EigenFaces*). Par contre, il arrive parfois que des personnes possèdent une physionomie faciale très semblable, mais que certains petits détails diffèrent grandement. Ce serait le cas par exemple d'une personne possédant un nez imposant.

En utilisant une méthode locale, davantage d'énergie sera accordée aux fins détails locaux, évitant ainsi le bruit causé par les cheveux, les chapeaux, la barbe, *etc.* De plus, certaines parties du visage sont relativement invariantes pour une même personne malgré ses expressions faciales ; c'est le cas notamment des yeux et du nez. Ceci demeure vrai tant que ces caractéristiques du visage ne sont pas en occultation. Les prochains paragraphes porteront donc sur les principales techniques de reconnaissance locales.

#### 3.2.4.1 *EigenObjects* (EO)

Basés sur les mêmes principes théoriques que la méthode des *EigenFaces* abordée à la section 3.2.3.2, les *EigenObjects* visent cette fois certaines parties bien précises du visage. La personne peut par exemple être reconnue uniquement grâce à ses yeux.



*a)*



*b)*

**Fig.** ***3.3:*** *EigenObjects : Image moyenne ainsi que les 6 premiers vecteurs propres pour a) l'oeil gauche et b) le nez.*

Pour réaliser l'apprentissage, un module de ce type doit tout d'abord procéder à une ACP des yeux contenus dans la banque de visages. L'espace des yeux (*eye space*) ainsi construit pourra alors servir au processus de reconnaissance qui est identique à celui utilisé pour les *EigenFaces*. Des résultats intéressants ont notamment été obtenus par Moghaddam et Pentland [47] sur une portion de la banque d'images FERET [48].

La figure 3.3a illustre un exemple d'oeil gauche moyen ainsi que les 6 premiers *eigeneyes* associés a la banque FERET, alors que la figure 3.3b illustre les mêmes informations pour le nez.

#### 3.2.4.2 HMM (Hidden Markov Models)

Les modèles de Markov cachés (HMM) sont utilisés depuis plusieurs années pour la détection et la reconnaissance du visage [53]. Différentes variantes ont également été proposées mais celle des (*Embedded HMM*) génère des résultats supérieurs [46] aux méthodes HMM de base.

Les *Embedded HMM* sont caractérisés par l'utilisation d'un HMM 1D de base, modélisant l'apparence du visage de haut en bas. Ensuite, chacun des états de ce modèle général contient un autre HMM 1D, dénommé *embedded* (c.-à-d. : incorporé). Ceux-ci modélisent cette fois l'apparence du visage de la gauche vers la droite.

Reposant sur certains coefficients de la transformée en cosinus discrète (*DCT*) comme source d'observations, les *Embedded HMM* constituent un algorithme de reconnaissance très performant. Or, les temps d'exécution des phases d'apprentissage

et de test sont relativement élevés, nuisant donc à son utilisation en temps réel sur d'immenses banques d'images.

Finalement, il est important de noter que le terme HMM sera utilisé tout au long de ce mémoire pour représenter la variante des *Embedded HMM*.

#### 3.2.4.3 Mesures et ratios

Lorsque la localisation des différentes parties[3] du visage est complétée, certaines mesures en pixels peuvent être prélevées à des fins de reconnaissance [4, 5]. Ces différentes valeurs peuvent être regroupées en deux catégories importantes, soient les dimensions des parties du visage et leurs distances relatives. Les mesures prélevées peuvent par exemple être les particularités suivantes :

- Dimensions de la tête, du nez, de la bouche, *etc.* ;
- Épaisseurs des sourcils, de la bouche, *etc.* ;
- Forme du menton (représentée par des distances relatives au centre de la bouche) [5] ;
- Positions relatives[4] du nez, des sourcils, de la bouche, *etc.*

Afin que cette technique soit efficace, l'image doit être préalablement normalisée sans altérer son ratio original. La technique de normalisation décrite au chapitre précédent convient adéquatement à cette prise de mesures.

Cela étant dit, la pose du visage doit être semblable à celle observée lors de l'apprentissage. Cette limitation étant difficilement respectée en pratique, l'utilisation de cette approche ne peut être efficace que dans un environnement contrôlé (p. ex. : une personne devant se présenter à une station d'identification).

#### 3.2.4.4 Couleurs

Une des caractéristiques les plus discriminantes entre les personnes repose sur la couleur. En effet, on peut identifier rapidement une personne de notre entourage selon

[3] Yeux, nez, oreilles, bouche, menton, début des cheveux, *etc.*

[4] Positions relatives par référence aux coordonnées des yeux.

la couleur de ses cheveux (p. ex. : rouge ou blanc). Il est donc possible d'intégrer plusieurs mesures de couleurs pour la reconnaissance, notamment pour les yeux, les cheveux et la peau [40].

Cela étant dit, les mesures dépendent énormément de l'éclairage et sont par ailleurs assez bien contournables (c.-à-d. : verres de contact, bronzage, perruque, *etc.*) par un imposteur. Cette méthode devrait donc être utilisée conjointement avec d'autres techniques pour améliorer sa robustesse.

Pour terminer, les caméras utilisées lors des acquisitions influencent également le niveau de performance de cette méthode. En effet, des couleurs différentes peuvent être obtenues avec une même caméra, dépendant des paramètres sélectionnés. Cette conclusion s'applique également à des caméras de marques différentes.

### 3.2.5 Combinaison de classifieurs

Plusieurs techniques peuvent parfois s'appliquer afin de résoudre un problème de reconnaissance des formes. Chacune d'entre elles possède évidemment ses forces et ses faiblesses qui, dans la majorité des cas, dépendent des situations (c.-à-d. : pose, éclairage, expressions faciales, *etc.*).

Il est par ailleurs possible d'utiliser une combinaison de classifieurs basés sur des techniques variées dans le but d'unir les forces de chacun et ainsi pallier à leurs faiblesses. Cette approche n'est cependant pas triviale, ni miraculeuse et certaines erreurs de classification peuvent parfois survenir même lorsqu'un des classifieurs est correct.

Trois problématiques importantes surgissent donc au moment de l'implantation :

1. Gestion logicielle de création, d'apprentissage et de communication efficace des différents modules de reconnaissance ;
2. Utilisation d'une base de données unique pour la représentation des objets à reconnaître ;
3. Configuration du système multi-classifeurs et fusion des résultats.

Ces différentes problématiques seront présentées dans les sous-sections suivantes et la sous-section 3.2.5.4 abordera une approche intéressante de sélection dynamique de

classifieurs.

#### 3.2.5.1 Architecture logicielle

Pour débuter, l'utilisation de techniques de reconnaissance différentes amène souvent de nombreuses complications. En effet, plusieurs questions doivent être solutionnées :

- Comment représenter les données et les entrées ?
- Comment réaliser l'apprentissage initial et en-ligne de la banque d'entraînement ?
- Que doit-on faire pour ajouter une personne à reconnaître ?
- Quels sont les résultats produits par chacun des modules ? Ces réponses sont-elles compatibles ?
- Comment doit-on accomplir la fusion des résultats ?
- *etc.*

Ces problématiques ne sont pas triviales, mais certaines notions de design orienté-objets peuvent aider à les résoudre en partie. Les *design pattern* [15], ou patrons de design, permettent en effet la réutilisation de principes d'implémentation résolvant des problématiques bien spécifiques.

La section 3.3.3 exposera les détails de l'architecture logicielle réalisée et répondant aux divers critères cités précédemment.

#### 3.2.5.2 Base de données

Dans un contexte de reconnaissance d'individu, la quantité d'informations requise peut être considérable. En effet, un système de surveillance et de reconnaissance doit en pratique identifier un grand nombre de gens, ce qui nécessite plusieurs images pour chaque personne.

Par exemple, pour reconnaître les 35 564 étudiants inscrits à l'Université Laval en 2001 [63] ainsi que le personnel professoral (1 542) et administratif (1 877), la base d'images[5] suivante serait nécessaire :

[5]Le nombre d'images par personne est fixé à 5, ce qui semble être une quantité suffisante pour représenter différentes vues frontales d'un individu.

- Nombre d'individus total : $35\,564 + 1\,542 + 1\,877 = 38\,983$ individus
- 5 images originales (/individu) : $5 \times (640 \times 480) \times 3$ octets $= 4\,608\,000$ octets
- 5 images normalisées (/individu) : $5 \times (130 \times 150) \times 3$ octets $= 292\,500$ octets

Dans le cas où cette base d'images serait conservée en mémoire pour des fins d'affichage par exemple, la quantité d'espace nécessaire serait de :

$$38\,983 \times (4\,608\,000 + 292\,500) = 190\,889\,176\,500 \approx 191 \text{ Gigaoctets}$$

ce qui est difficilement réalisable en pratique. Par contre, si on ne garde que les images normalisées en mémoire[6], un espace plus modeste est nécessaire :

$$38\,983 \times 292\,500 = 11\,402\,527\,500 \approx 11.4 \text{ Gigaoctets}$$

ce qui est quand même relativement élevé compte tenu de la quantité d'espace mémoire typique des ordinateurs qui est de 256 ou 512 Mégaoctets.

Il est évidemment possible d'utiliser une compression d'images (p. ex. : JPG), ce qui ferait chuter drastiquement la quantité d'espace mémoire nécessaire. Cependant, l'utilisation de la compression complique le traitement car le décodage (et l'encodage) des images est désormais obligatoire.

À la vue de ces nombres colossaux, il est évident que le système de reconnaissance ne doit contenir qu'une seule version de la base de données et ce, peu importe le nombre de modules de reconnaissance utilisé.

Pour ce faire, une certaine uniformité est nécessaire entre les différents modules. La base de données devra donc être accessible de façon standard par tous les modules.

#### 3.2.5.3 Fusion des résultats

Pour réaliser la fusion des résultats et ainsi prendre une décision, deux méthodes sont envisageables. La première, dite hiérarchique, élimine les candidats d'un classifieur à l'autre afin de sélectionner le meilleur individu.

[6] Certaines méthodes d'identification (p. ex. : corrélation) utilisent directement les images normalisées pour la reconnaissance.

La deuxième technique, dénommée ensemble de classifieurs ou *late fusion*, effectue une prise de décision à partir des réponses fournies par différents classifieurs configurés en parallèle. Cette dernière peut être réalisée de plusieurs façons différentes qui sont regroupées en deux catégories, soient celles incluant les méthodes de votes et deuxièmement, celles reposant sur un apprentissage automatique des données.

**Méthodes de votes** Les méthodes de votes, étant plus simples et plus rapides, sont très couramment utilisées comme fonctions de décision. Le principe de base consiste à utiliser directement les sorties des différents classifieurs comme source d'information pour ainsi appliquer une fonction spécifique de décision.

Ces sorties prennent la forme de listes ordonnées de choix représentant les identités les plus vraisemblables selon les classifieurs. Ces différentes listes (c.-à-d. : une pour chaque classifieur du système) sont ensuite fusionnées à l'aide d'une fonction particulière afin de générer une liste ultime des identités les plus probables.

Parmi les fonctions de décisions disponibles, notons entre autres la règle de la somme (*sum rule*) [65], du produit (*product rule*) [34] et le compte de Borda (*Borda count*) [11, 22].

**Apprentissage automatique** Il est également intéressant d'ajouter une couche «intelligente» au système multi-classifieur. En effet, un réseau de neurones de type perceptron multi-couches (PMC) [20] peut être utilisé pour discriminer les différentes classes. La performance du PMC dépendra essentiellement de la séparabilité des classes, du nombre d'exemples d'entraînement et de la persistance des données entraînement/réel.

D'autres types de réseaux de neurones peuvent être utilisés, mais les SVM [8] semblent les plus prometteurs. En effet, ceux-ci sont particulièrement efficace lorsque le nombre d'observations par classe est faible.

#### 3.2.5.4 Sélection dynamique de classifieur (DSC)

La sélection dynamique de classifieur (*Dynamic selection of classifiers* ou DSC) est une approche relativement récente [16] visant à pallier à certaines lacunes des systèmes multi-classifieurs (MC).

L'inconvénient majeur des techniques MC provient du fait que pour un certain prototype à identifier, un classifieur possédant une réponse exacte peut être noyé parmi plusieurs autres ayant tort. Cette situation provoque évidemment une erreur de reconnaissance avec n'importe qu'elle méthode de vote.

Cela étant dit, la technique de DSC consiste à sélectionner le classifieur le plus adéquat pour l'identification d'un prototype. C'est un processus qui peut être comparé à une phase de classification préliminaire.

Outre les mécanismes internes utilisés pour effectuer cette prédiction, la DSC requiert tout particulièrement une architecture versatile pouvant facilement (et efficacement) modifier sa configuration, peu importe le sous-module de classification. Il sera démontré ultérieurement que l'architecture logicielle développée rencontre les prérequis nécessaires à l'implémentation de cette technique.

## 3.3 Approche retenue

Pour la conception d'un système d'identification de personnes, plusieurs avenues sont possibles et méritent d'être envisagées. Certaines méthodes peuvent cependant se révéler irréalisables, soit à cause du matériel disponible ou des conditions spécifiques de l'environnement visé.

Les techniques de reconnaissance basées sur la thermographie infrarouge ainsi que celles utilisant des données tridimensionnelles ont donc été rejetées. Ces méthodes, quoique particulièrement intéressantes, nécessitent des équipements spécialisés, transgressant ainsi les contraintes du projet.

L'approche retenue vise tout d'abord le choix du type d'engin de reconnaissance. Pour des raisons de robustesse et de variété, l'utilisation d'un module d'identification multi-classifieur est donc retenue.

Quatre techniques de reconnaissance ont donc été sélectionnées, soient les *Eigen-Faces* [62], les *EigenObjects* [47], la DCT [19] et les HMM [46]. La sous-section 3.3.1 présentera ces méthodes en détails, notamment en ce qui a trait à l'apprentissage, la reconnaissance et aux ressources requises par chacune d'elle.

Pour ce qui est du type de multi-classifieur utilisé ainsi que de la fonction de décison retenue, un MC de type parallèle utilisant le compte de Borda comme méthode de vote a été sélectionné. Ces choix seront présentés avec davantage de détails à la sous-section 3.3.2.

L'architecture logicielle développée sera par la suite décrite à la sous-section 3.3.3, suivie d'une description détaillée des classes C++ créées.

Finalement, une image représentant l'interface de l'application de reconnaissance multi-classifieur est illustrée à la figure 3.4.

### 3.3.1 Méthodes de reconnaissance

Parmi toutes les méthodes présentées à la section 3.2, certaines demeurent plus avantageuses que d'autres. C'est ainsi que quatre techniques de reconnaissance ont été sélectionnées pour la réalisation de l'application finale. Les critères utilisés pour cette sélection reposent notamment sur les temps d'exécution et les taux de reconnaissance.

Cela étant dit, les différentes techniques seront abordées avec moult détails dans les sections suivantes et divisées en quatre rubriques principales : l'apprentissage, la reconnaissance, l'ajout de personnes et les ressources utilisées.


**Fig.** ***3.4:*** *Interface de l'application de reconnaissance multi-classifieur d'individu. L'interface principale contient trois fenêtres représentant les résultats de la soustraction de l'arrière-plan (nord-ouest), de la détection du visage (nord-est) et de l'identification (sud). Cette dernière contient notamment le visage détecté normalisé à l'extrême gauche suivi des résultats pour les méthodes individuelles (DCT,* EigenFaces*, HMM) et par la fonction de décision (*Fusion*).*

#### 3.3.1.1 *EigenFaces*

Il est pratiquement impossible de concevoir un système de reconnaissance du visage sans envisager la très populaire technique des *EigenFaces*, introduite en 1991 par Turk et Pentland [62].

Cette méthode est tout d'abord basée sur une analyse en composantes principales (ACP) appliquée sur l'ensemble des visages d'une banque d'entraînement. Elle consiste essentiellement à effectuer un réduction de dimensionnalité en codant les visages dans une nouvelle base formée par les premiers vecteurs propres (c.-à-d. : *EigenFaces*) provenant du calcul de l'ACP.

Les *EigenFaces* associés aux plus fortes valeurs propres représentent donc, dans l'espace des images, les directions dans lesquelles les variations sont les plus marquées. C'est ainsi que les premiers visages propres représentent habituellement les différences d'éclairage ainsi que les personnes portant des lunettes ou une barbe.

**Apprentissage** La phase d'apprentissage (ou de modélisation) des *EigenFaces* se déroule comme suit :

1. Un visage moyen $\Psi$ est calculé à partir des $N$ images d'entraînement $I_i$ de dimensions $L \times H$ :
$$\Psi = \frac{1}{N}\sum_{i=1}^{N} I_i \tag{3.1}$$
2. Ce visage moyen est soustrait des images d'apprentissage (on élimine donc les ressemblances pour se concentrer sur les différences), ce qui génère les vecteurs de différences $\Phi_i$ associés à chacune des images :
$$\Phi_i = I_i - \Psi \tag{3.2}$$
3. La matrice de covariance $C$ est construite (approche inter-pixels) :
$$C = \frac{1}{N}\sum_{i=1}^{N} \Phi_i \Phi_i^T = AA^T \tag{3.3}$$
où chacune des colonnes de $A$ représente un vecteur de différences, soit $A = [\Phi_1, \Phi_2, ..., \Phi_N]$ ;

4. Étant donné les dimensions élevées de $C$ ($LH \times LH$), une approche inter-images est privilégiée[7]. Le calcul se limite [62] donc à une matrice $L = A^T A$ dépendant du nombre d'images dans la banque d'apprentissage (c.-à-d. : dimensions $N \times N$) ;
5. Calcul des valeurs et vecteurs propres de la matrice $L$ ;
6. Le visage propre $u_i$ associé à la $i$ème valeur propre est formé en utilisant les vecteurs propres $v_i$ de la matrice $L$ :

$$u_i = \sum_{k=1}^{N} v_{ik} \Phi_k \tag{3.4}$$

7. Les $M$ premiers vecteurs propres (EF) (c.-à-d. : associés aux plus fortes valeurs propres) sont conservés. Ils définissent ainsi l'espace des visages (*face space*) ;
8. Les images originales sont projetées dans l'espace des visages pour former une suite de coefficients d'appartenance, ce qui donne pour une l'image $I_i$ :

$$\omega_k = u_k^T \Phi_k \tag{3.5}$$

où $k = 1, ..., M$ ;

9. Ces coefficients forment alors un vecteur représentant l'image $I_i$ :

$$\Omega_i = [\omega_1, \omega_2, ..., \omega_M] \tag{3.6}$$

Une fois l'apprentissage complétée, les différentes représentations d'un individu peuvent être regroupées afin de former une classe. Ceci peut être réalisé notamment en calculant une moyenne des différents vecteurs $\Omega_i$ correspondants à la personne [62].

Il est également possible de considérer les représentations individuellement et ainsi les utiliser directement avec l'algorithme du $K$-ppv. C'est d'ailleurs cette approche qui a été retenue dans l'implémentation du système.

**Reconnaissance** Lorsqu'un visage est présenté au système, la procédure d'identification consiste à :

1. Projeter l'image d'entrée $I$ dans l'espace des visages, ce qui engendre un coefficient d'appartenance $\omega_k$ à un *EigenFace* $u_k$ en utilisant l'équation 3.5 ;

[7] Il serait par ailleurs presque irréalisable d'effectuer l'inversion d'une matrice 19 500×19 500 correspondant à des images de dimensions 130×150 pixels.

2. Les coefficients d'appartenance forment alors un vecteur de représentation $\Omega$ de taille $M$ ;
3. Ce dernier est comparé avec ceux obtenus lors de la phase d'apprentissage à l'aide de l'algorithme $K$-ppv (voir sous-section 3.3.1.5) en appliquant une métrique de distance particulière.

**Ajout d'une personne** Lorsqu'une nouvelle personne[8] est ajoutée à la base de données, la méthode classique consiste à refaire l'apprentissage complet (c.-à-d. : ACP pour déterminer les nouveaux visages propres). Il existe cependant deux alternatives à cette solution.

Premièrement, lorsque la banque d'apprentissage est relativement grande et que les visages qu'on y retrouve sont représentatifs, il est possible d'utiliser directement les EF existants afin de calculer les coefficients de projection des nouvelles images.

Il serait par contre intéressant à long terme (et après plusieurs ajouts de personnes) de réaliser un ré-apprentissage complet afin d'obtenir des visages propres plus représentatifs de la base de données.

La deuxième méthode est relativement récente et repose sur une fusion d'espaces de visages [14]. Il est en effet possible de fusionner deux *face space* sans toutefois nuire au processus de reconnaissance (p. ex. : altération des visages propres). Donc en pratique, un espace temporaire est généré à partir des nouvelles images pour être ensuite fusionné avec l'espace principal.

Cette opération est également avantageuse en terme de temps de calcul [14] comparativement à un recalcul complet. Par ailleurs, comme l'espace des visages est modifié, il est primordial de re-projeter tous les visages d'entraînement afin de reconstruire les représentations.

**Ressources requises** La méthode des EF nécessite plusieurs structures devant être chargées en mémoire pour une performance accrue (c.-à-d. : pour un temps de réponse rapide).

[8] Le même processus est nécessaire lorsque de nouvelles images sont ajoutées à des personnes déjà présentes dans la banque.

Tout d'abord, c'est le cas des $M$ premiers visages propres. Ceux-ci sont primordiaux dû à leur importance en phase de reconnaissance et d'apprentissage. Pour une base de données contenant 1 000 images de dimensions 130×150 pixels et la conservation des 200 premiers EF, ceci consiste en pratique à charger une quantité de 200×(130×150) = 3.9 Megaoctets de mémoire.

Une représentation d'un visage d'apprentissage consiste quant à elle à un vecteur de 200 éléments[9]. Compte tenu qu'une valeur de type *float* occupe 4 octets, 1 000×200×4 = 800 Kilooctets sont nécessaires pour emmagasiner les prototypes de la banque d'entraînement.

En conclusion, les ressources requises par la technique des EF sont donc relativement faibles. Le processus d'ACP demande par ailleurs une plus grande quantité de mémoire qui dépend directement du nombre d'images utilisées. Pour des banques d'images très volumineuses, il est conseillé d'utiliser un algorithme d'ACP qui n'est pas exclusivement en-ligne.

#### 3.3.1.2 *EigenObjects*

La méthode des *EigenObjects* est avant tout une application plus ciblée des *EigenFaces* ayant des zones spécifiques du visage comme régions d'intérêt. Étant donné que certaines parties du visages sont moins affectées par les expressions faciales, il est intéressant de s'y attarder pour extraire de l'information. C'est le cas notamment des yeux et du nez, qui demeurent presque inchangés pour une même personne et ce, quelle que soit son expression faciale (pour autant que les yeux soient ouverts).

La toute première étape de prétraitement consiste, tant en phase d'apprentissage qu'en phase de reconnaissance, à localiser les parties importantes à l'intérieur du visage. La précision du module de détection du visage est donc cruciale.

Contrairement aux visages, les yeux et le nez se ressemblent davantage entre eux, ce qui rend les fausses identifications plus fréquentes. Par contre, grâce à la concaténation des représentations individuelles, certaines ressemblances peuvent être éliminées.

[9] Cette valeur est égale à $M$, soit le nombre de visages propres conservés.

Évidemment, la performance de cette technique dépend fortement de l'efficacité de la segmentation. Les cas d'occultations (p. ex. : port de lunettes fumées ou yeux fermés) nuisent également aux *EigenObjects* en y ajoutant du bruit, causant alors de fausses identifications.

Cette particularité peut donc fortement brouiller les cartes lors de son utilisation à l'intérieur d'un système multi-classifieur, nuisant ainsi aux autres modules de reconnaissance.

**Apprentissage** Afin d'appliquer la technique des *EigenFaces* sur les parties du visages, les sous-images doivent avant tout être extraites et regroupées en trois ensembles. Ceux-ci sont utilisés pour calculer les ACP correspondantes, procédure qui génère de nouvelles bases associées à chacun des *EigenObjects*.

La phase d'apprentissage utilisée est identique à celle présentée auparavant à la sous-section 3.3.1.1. La seule différence réside au niveau du nombre d'opérations à réaliser. Celui-ci dépend en effet de la quantité de caractéristiques à reconnaître qui est de trois composantes dans le cas présent.

Les coefficients de projection de chaque caractéristique sont donc calculés et concaténés ensemble pour former un seul et unique vecteur pour chaque image. Cette représentation unifiée facilite légèrement la gestion et le nombre d'opérations à réaliser lors de la phase d'identification.

**Reconnaissance** Lors de la phase d'identification, les sous-images représentant les caractéristiques du visage sont extraites et utilisées indépendamment selon la procédure de reconnaissance des *EigenFaces*.

Les représentations individuelles sont ensuite concaténées pour former le vecteur unifié qui est comparé à ceux de la banque d'apprentissage en utilisant l'algorithme $K$-ppv.

**Ajout d'une personne** Les solutions à la problématique d'ajout d'images ou de personnes à la base de données sont identiques à celles envisagées pour la méthode des *EigenFaces*.

Par ailleurs, il est possible d'émettre l'hypothèse que les caractéristiques du visages se ressemblent davantage entre elles que les visages entre eux. Les vecteurs propres ainsi calculés lors des ACP seraient donc suffisants et adéquats pour représenter de nouveaux visages, ce qui évite un ré-apprentissage complet. Cette solution demeure cependant à vérifier mais semble à priori très avantageuse.

**Ressources requises** Les ressources nécessaires à l'utilisation de la méthode des *EigenObjects* représentent une très faible quantité de mémoire. En effet, les sous-images utilisées sont de dimensions 50×30 et 50×40 pixels respectivement, pour les yeux et le nez.

Ceci représente donc pour une banque de 1 000 images et 75 vecteurs propres conservés, $75 \times (2\times(50\times30) + (50\times40)) = 375$ Kilooctets de mémoire, ce qui est relativement très peu.

Les représentations vectorielles des individus constituent également une très faible consommation de ressources, étant encore une fois inférieure à la quantité requise pour la technique des EF.

#### 3.3.1.3 DCT

La prochaine technique de reconnaissance retenue utilise la transformée de cosinus discrète [50] pour extraire et modéliser l'information des visages [19]. Les premiers coefficients de la DCT sont ainsi conservés lors de l'apprentissage et utilisés directement pour la phase d'identification. Ceux-ci correspondent aux basses et moyennes fréquences contenues dans les images.

Il est important de noter que cette méthode possède certaines similitudes mathématiques avec les *EigenFaces*. L'utilisation combinée de celles-ci est donc à valider.

**Apprentissage** La phase d'apprentissage de la méthode basée sur l'utilisation de la DCT se révèle en pratique fort simple. Elle est composée de trois étapes à réaliser pour chacune des images de la banque :

1. Calcul de la transformée en cosinus discrète de l'image normalisée ;
2. Extraction et concaténation des premiers coefficients de la DCT afin de former un vecteur unifié ;
3. Sauvegarde des représentations.

Le processus d'apprentissage est donc réalisé sur chaque image indépendamment (contrairement aux techniques EF et EO).

Dans un autre ordre d'idée, le nombre de coefficients conservé varient en pratique de 7×7 (49) à 14×14 (196). Cette quantité dépend essentiellement de la taille du problème à résoudre (c.-à-d. : le nombre d'images dans la banque).

**Reconnaissance** Lorsqu'une image test est soumise à des fins d'identification, la procédure consiste tout d'abord à extraire sa représentation, ce qui est effectué par le calcul de la transformée. Ensuite, tout comme les EF et les EO, l'identification est réalisée à l'aide de l'algorithme $K$-ppv entre le vecteur unifié et les représentations connues.

Le temps d'exécution dépend alors de deux opérations importantes, soient le calcul de la transformée et l'application de l'algorithme $K$-ppv.

**Ajout d'une personne** Un des avantages de cette méthode repose sur sa grande flexibilité en cas d'ajouts d'images ou de personnes. En effet, cette opération consiste tout simplement à effectuer les étapes 1 et 2 de la phase d'apprentissage pour chaque image supplémentaire et à ajouter les représentations à la liste actuelle.

Cette tâche n'implique donc aucun ré-apprentissage complet, contrairement aux méthodes *EigenFaces* et *EigenObjects*.

**Ressources requises** Les ressources requises par cette méthode ne concernent que la liste des représentations vectorielles, ce qui résulte en une très faible consommation

de mémoire. Typiquement, les vecteurs comportent de 49 à 196 éléments (de type *float*) et constituent les seules données nécessaires à l'exécution de la méthode.

#### 3.3.1.4 Hidden Markov Models (HMM)

Le dernier algorithme de reconnaissance du visage retenu pour le système est basé sur l'utilisation de modèles de Markov cachés (*HMM*) [53]. La variante des HMM incorporés (*Embedded HMM*) [46] tente par ailleurs de modéliser l'apparence bidimensionnelle du visage, c'est-à-dire l'ordre d'apparition des caractéristiques du visage de haut en bas et de gauche à droite.

Pour ce faire, le modèle est tout d'abord composé d'un HMM 1D dont les différents états représentent les états primaires (*super states*) du système. Plus précisément, ceux-ci modélisent les diverses tranches du visage de haut en bas (c.-à-d. : front, yeux, nez, bouche, menton, *etc.*). À l'intérieur de chacun de ces états se retrouve un HMM 1D secondaire, modélisant cette fois les caractéristiques de la gauche vers la droite.

Les observations nécessaires à la modélisation des HMM incorporés sont basées sur les coefficients de basses fréquences de la transformée en cosinus discrète (DCT) des sous-régions de l'image.

**Apprentissage** La phase d'apprentissage des HMM consiste essentiellement à la conception d'un modèle pour chacun des individus de la banque. Ce processus itératif peut cependant être très long à réaliser.

Les différentes étapes de l'apprentissage sont plus précisément :

1. Segmentation initiale uniforme des images de l'individu. Celles-ci sont divisées en $C$ rangées (états primaires) composées de $N_c$ régions (états incorporés ou secondaires). Deux exemples sont notamment illustrés à la figure 3.5 et représentent exactement le nombre d'états utilisés pour les expérimentations ;
2. Étape itérative :
   (a) Une segmentation de Viterbi doublement incorporée [35] est utilisée afin de raffiner la séparation des différentes régions ;

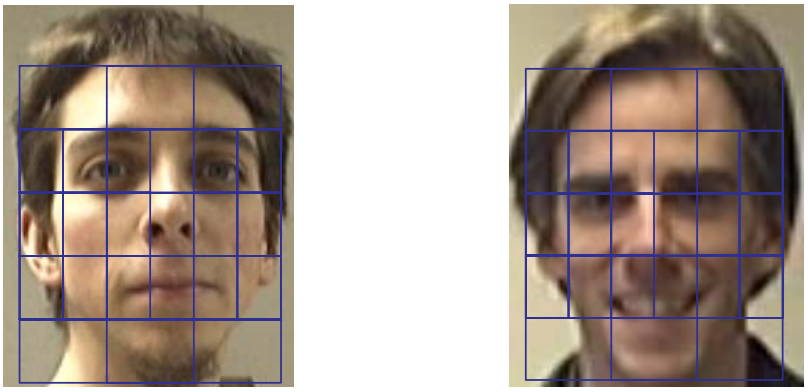


**Fig. *3.5:*** *HMM : exemples de segmentation initiale. Le nombre d'états primaires et secondaires illustré est identique à celui utilisé pour les expérimentations qui seront présentées au chapitre suivant, soit 5 rangées composées respectivement de 3, 6, 6, 6 et 3 états.*

(b) Les paramètres du modèle sont estimés à l'aide d'une extension en 2D de l'algorithme *k-means* segmentaire [35] ;

3. L'apprentissage se termine lorsqu'un seuil appliqué à la segmentation de Viterbi est atteint.

**Reconnaissance** Afin d'identifier un individu à l'aide des HMM, tous les modèles sont utilisés et celui qui possède le maximum de vraisemblance est retenu comme étant celui représentant la personne à reconnaître.

**Ajout d'une personne** Contrairement aux méthodes EF et EO, les HMM ne requièrent pas l'utilisation simultanée de toutes les images des individus pour réaliser l'apprentissage. L'ajout d'une personne peut donc être réalisé facilement sans avoir à fusionner des données ou reprendre l'apprentissage au complet. Pour ce faire, le modèle représentant l'individu est généré et simplement ajouté à la liste existante.

Cette facette des HMM leur confèrent bien entendu un atout certain dans un contexte de système temps réel.

**Ressources requises** La somme des ressources requises par cette technique peut être divisée en deux parties, soient celles nécessaires lors de l'apprentissage et celles utilisées pour le stockage des modèles.

Tout d'abord, l'espace mémoire nécessaire pour l'apprentissage est relativement

faible étant donné que peu d'images sont utilisées simultanément pour réaliser cette tâche.

Pour ce qui est du stockage des modèles, les paramètres choisis influencent grandement la taille des fichiers de sauvegarde. Plus le nombre d'états est élevé, plus il y aura de données à sauvegarder (p. ex. : matrices de transition).

Il demeure néanmoins que la technique des HMM requiert un espace disque beaucoup plus élevé que les autres algorithmes. À titre d'exemple, la taille du fichier de sauvegarde des HMM pour la banque d'images FERET occupe environ 25 Mo d'espace disque contrairement à moins de 2 Mo pour la méthode utilisant la DCT.

#### 3.3.1.5 *K*-ppv et métriques de distance

Pour bon nombre de techniques, l'algorithme du $K$ plus proche voisin ($K$-ppv) est utilisé lors de la phase de reconnaissance. C'est le cas notamment des techniques *EigenFaces*, *EigenObjects* ainsi que DCT abordées précédemment. Le $K$-ppv se base tout simplement sur une liste ordonnée des voisins les plus près d'une image test.

Pour ce faire, le vecteur test est comparé avec chacun des prototypes constituant la banque d'apprentissage. Pour chacune de ces comparaisons, une distance est calculée et agit comme taux de ressemblance avec un certain prototype. Cette distance peut être mesurée par différentes métriques, dont voici une liste partielle :

**City-block ($L_1$)** La distance $L_1$ consiste à calculer la somme des différences absolues entre les éléments des vecteurs, soit la fonction suivante :

$$L_1 = |U - V| = \sum_{i=1}^{N} |U_i - V_i|$$

où U et V représentent les vecteurs à comparer et $N$ la taille des vecteurs.

**Euclidienne ($L_2$)** La distance $L_2$ ou distance euclidienne entre deux vecteurs consiste à calculer la racine de la somme des différences au carré, soit :

$$L_2 = \|U - V\| = \sqrt{\sum_{i=1}^{N}(U_i - V_i)^2}$$

où U et V représentent les vecteurs à comparer et $N$ la taille des vecteurs.

**Angle** L'angle négatif entre les deux vecteurs [69] équivaut à l'équation :

$$Angle = -\frac{U.V}{\|U\|\|V\|} = -\frac{\sum_{i=1}^{N} U_i V_i}{\sqrt{\sum_{i=1}^{N} U_i^2 \sum_{i=1}^{N} V_i^2}}$$

où U et V représentent les vecteurs à comparer et $N$ la taille des vecteurs.

Une fois toutes les distances mesurées, une liste ordonnée croissante est générée afin de départager les candidats. Habituellement, $K$ se voit assigner une valeur de 1, ce qui signifie que le prototype de la banque d'apprentissage le plus proche est sélectionné. Si $K$ est supérieur à 1, le prototype qui possède la majorité de votes parmi les $K$ premiers voisins sera choisi.

### 3.3.2 Multi-classifieur

Afin de combiner les différentes techniques de reconnaissance, le système d'identification utilise un MC de type parallèle couplé à une fonction de décision basée sur les votes.

Le choix de cette approche repose tout d'abord sur la simplicité et la flexibilité de son implémentation. En effet, n'importe quel classifieur peut générer une liste d'identités potentielles. De plus, dû à l'homogénéité des sorties générées par les différents classifieurs, ceux-ci peuvent être retirés ou ajoutés en cours d'exécution et ce, sans problème.

Pour ce qui est de la fonction de décision utilisée, l'approche développée utilise principalement le compte de Borda [11, 22]. Il est notamment un des meilleurs algorithmes basés sur les votes selon des expérimentations récentes [65].

Son fonctionnement est par ailleurs fort simple. Son utilisation implique en effet seulement le calcul des rangs moyens de chacune des classes d'intérêt. Ces valeurs permettent ensuite la création d'une nouvelle liste ordonnée. L'individu possédant le plus faible[10] rang moyen est sélectionné pour l'identification.

De plus, différentes variantes du compte de Borda sont possibles afin de limiter certains désagréments, comme entre autres l'effet des votes très bas [64]. Si par exemple une classe obtient 3 votes en premières positions ainsi qu'un vote à la $444^e$ position, elle sera fort probablement exclu à cause de son médiocre rang moyen.

La solution apportée dans la cadre du système de reconnaissance repose sur un plafond maximum pour l'attribution des rangs. Ainsi, une classe ne peut se voir décerner un vote en $444^e$ position mais plutôt une valeur fixe pour un rang dépassant un certain seuil. Celui-ci peut être fixé à 10 ou 15 par exemple.

### 3.3.3 Architecture logicielle

L'architecture logicielle a été conçue avant tout afin de simplifier l'intégration de méthodes de reconnaissance différentes, tout en demeurant la plus flexible et versatile possible. Pour ce faire, plusieurs classes codées en C++ furent créées, chacune d'elle représentant une partie clé du système d'identification.

L'utilisation du patron de design de stratégie comportementale (*behavioral strategy*) d'abstraction [15] représente une solution ingénieuse et particulièrement adaptée à ce type de problème. Il permet en effet la sélection dynamique d'un algorithme parmi plusieurs et ce, peu importe son implémentation. Ceci est vrai tant qu'il respecte la structure et l'interface de base.

Cela étant dit, ce patron de design est réalisé à l'aide du langage de programmation C++, qui procure des notions de polymorphisme et d'héritage simplifiant l'organisation et le fonctionnement des modules. En utilisant des classes abstraites comme patrons de base, il est possible de créer plusieurs classes dérivées héritant des particularités de la

[10] Un rang moyen de 1 pour une classe $C$ signifie que chacun des classifieurs a voté pour la classe $C$ en premier choix.

classe mère.

Par ailleurs, un des objectif de cette architecture demeure la flexibilité logicielle en temps réel, c'est-à-dire la faculté de modifier dynamiquement les composantes et les particularités du système. Cette souplesse accrue facilite de plus l'expérimentation de différentes configurations multi-classifieurs.

La section 3.3.3.2 présentera donc les différentes classes réalisées alors que la section 3.3.3.3 portera sur les forces et les faiblesses de cette architecture.

#### 3.3.3.1 Principes généraux

Pour débuter, le système est composé d'un moteur de reconnaissance et d'une base de données contenant des informations sur les personnes à identifier. Cette banque contient des éléments clés comme par exemple les images et le nom des individus.

Ce moteur de reconnaissance est composé de plusieurs modules d'identification spécifiques implémentant différents algorithmes. Chacun d'entre eux génère alors en phase de reconnaissance une liste de choix qui sont combinées grâce aux fonctions disponibles dans l'engin principal.

#### 3.3.3.2 Classes

La figure 3.6 illustre l'architecture logicielle du système avec ses différentes classes et leurs interrelations. Les fonctions les plus importantes y sont également illustrées, citons notamment les fonctions abstraites de la classe *RecognitionModule*.

**HumanDatabase** Le but d'un système de reconnaissance est évidemment d'identifier un certain type d'objets à partir de classes connues, qui sont des visages dans le cas présent. Toute l'information disponible lors de l'apprentissage doit donc être stockée efficacement dans une structure de données. Pour ce faire, les classes *HumanDatabase* et *Human* ont été créées.

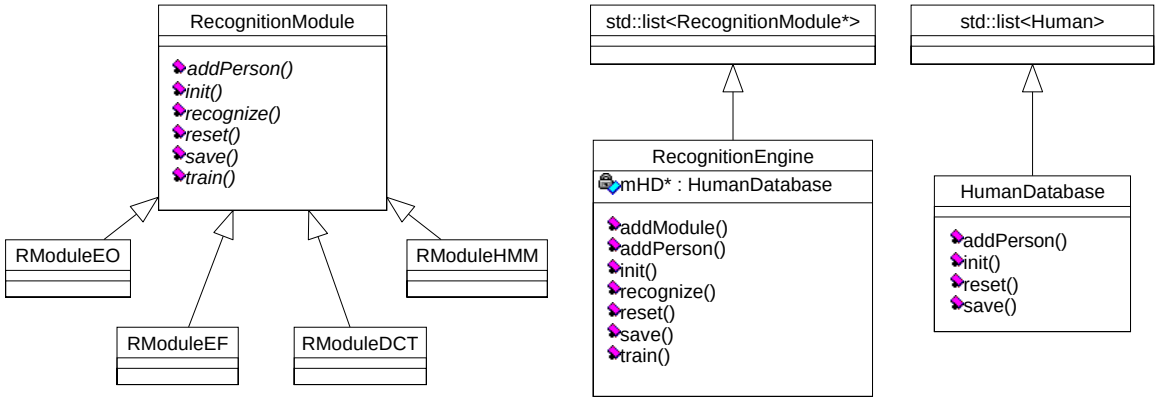



**Fig.** ***3.6:*** *Architecture logicielle du système multi-classifieur.*

Pour faciliter certaines opérations, *HumanDatabase* hérite directement de la classe *list* provenant de la librairie STL (*Standard Template Library*). Celle-ci contient également certaines informations importantes comme par exemple la quantité totale d'images ou le nombre d'individus.

Par ailleurs, la grande majorité des données (et les plus importantes) se retrouve dans chacun des objets *Human* contenus dans la liste. Cette classe de base contient :

- Nom et numéro d'identification de la personne ;
- Date d'entrée dans le système ;
- Images originales ;
- Images normalisées ;
- Chemins d'accès des différentes images sur le support matériel de stockage ;
- Informations diverses[11].

La classe *Human* possède également les fonctions *read/write* qui permettent de lire et de sauvegarder l'objet.

**RecognitionModule** Dans un but d'uniformité, une classe nommée *RecognitionModule* agit comme structure de base pour tous les modules de reconnaissance spécifiques. Cette classe abstraite possède des fonctions virtuelles pures devant absolument être surdéfinies dans les classes dérivées, ce qui consiste en pratique à fournir une implantation pour chacune d'elle.

[11] Il est envisageable d'inclure d'autres types d'informations comme entre autres des modèles 3D, des empreintes digitales, *etc.*

Ce mécanisme assure alors aux usagers d'une classe dérivée que chacune des fonctions sera disponible et utilisable[12]. Les différentes fonctions virtuelles pures sont :

- *addPerson* : Ajouter une personne aux données d'apprentissage (implique un réapprentissage total ou partiel) ;
- *init* : Initialiser le module (par défaut ou à partir d'un fichier) ;
- *recognize* : Reconnaître une personne à partir d'une image ;
- *retrain* : Redémarrer le processus d'apprentissage ;
- *save* : Sauvegarder les données d'apprentissage et certaines variables membres du module.

De nombreuses fonctions utilitaires sont également disponibles dans la classe *RecognitionModule*, ce qui a pour but de simplifier le code et de minimiser les chances d'erreurs dues à des versions différentes de la même fonction. Il y a notamment :

- *findKNearestNeighbors* : Appliquer l'algorithme $K$-ppv entre les données d'apprentissage et les résultats d'une recherche ;
- *computeDistanceVectorsL1, L2 et Angle* : Calculer une distance spécifique entre deux vecteurs (*city-block*, distance euclidienne et angle) ;
- *writeOnlineResults* : Écrire la liste ordonnée des résultats dans un fichier.

**RModuleDCT** Ce premier module spécifique repose sur l'utilisation de la DCT pour la formation des représentations des visages. Tout d'abord, comme cette classe est dérivée de *RecognitionModule*, toutes les fonctions citées précédemment doivent être implantées.

Voici donc davantage de détails sur ces différentes fonctions :

- *addPerson* : Calculer les coefficients de la DCT à partir du visage d'entrée et ajouter les $C$ premiers coefficients à la liste des représentations. L'ajout d'une personne est donc une opération demandant très peu d'opérations et qui ne dépend que du temps nécessaire pour compléter la transformée ;
- *retrain* : Répéter l'opération *addPerson* pour chacune des personnes de la banque ;
- *recognize* : Extraire la représentation du visage d'entrée et trouver les $K$ plus proches voisins parmi les prototypes de la banque d'apprentissage.

[12] Ceci ne garantie évidemment pas son efficacité mais sa disponibilité.

**RModuleEF** Le prochain module de reconnaissance implanté concerne l'utilisation de la méthode des *EigenFaces*. En plus d'hériter directement de *RecognitionModule*, ce module hérite également de la classe *EigenObjects* appliqué à des visages. Il est important de noter que la classe nommée *EigenObjects* n'implémente pas la technique des *EigenObjects* mentionnée auparavant, mais est plutôt une classe générale *Eigen* pouvant être utilisée avec n'importe quel type d'objet à reconnaître.

Cette façon de procéder permet alors la séparation adéquate de la mécanique propre aux *EigenFaces* des fonctions d'interface et de gestion. En effet, toutes les fonctions de sauvegarde, de lecture, de projection, *etc.* sont conservées dans la classe mère. Voici par ailleurs davantage de détails sur les fonctions de base de *RModuleEF* :

- *retrain* : Appliquer l'algorithme décrit à la section 3.3.1.1 et sauvegarder les coefficients de projection dans la liste des représentations ;
- *recognize* : Extraire la représentation du visage d'entrée en le projetant dans l'espace des visages (*face space*) et trouver les $K$ plus proches voisins parmi les prototypes de la banque d'apprentissage.

La fonction *addPerson* représente le désavantage majeur de l'utilisation des *EigenFaces* car un ré-apprentissage total est habituellement obligatoire. Plusieurs solutions envisageables ont cependant été décrites précédemment à la sous-section 3.3.1.1.

**RModuleEO** Le module utilisant la technique des *EigenObjects* ressemble largement au module précédent. Tout d'abord, la classe hérite directement de *RecognitionModule* mais contrairement au *RModuleEF*, elle dérive également d'un vecteur de trois pointeurs à des objets *EigenObjects*. Ceux-ci représentent les différents espaces associés aux caractéristiques du visage qui sont dans ce cas-ci, les yeux et le nez.

Pour ce qui est des fonctions, les différences avec la classe *RModuleEF* sont les suivantes :

- *retrain* : Appliquer l'algorithme détaillé à la section 3.3.1.1 sur chacun des ensembles d'images caractéristiques. Concaténer les coefficients de projection des images d'entraînement sur les premiers vecteurs propres de chaque ensemble et les sauvegarder dans la liste des représentations ;
- *recognize* : Extraire la représentation du visage d'entrée en projetant chacune des caractéristiques du visage dans l'espace associé et trouver les $K$ plus proches

voisins parmi les prototypes de la banque d'apprentissage.

**RModuleHMM** Ce dernier module de reconnaissance spécifique repose sur l'utilisation des HMM présentés auparavant à la section 3.2.4.2 et 3.3.1.4. Une classe nommée *ContEHMM* est utilisée comme structure de base et représente un objet de base de type *Embedded* HMM. Chacun des individus présent dans la base de données possède, après la phase d'apprentissage, un modèle HMM qui sera sauvegardé par la suite.

Les fonctions membres de cette classe sont les suivantes :

- *addPerson* : Appliquer l'algorithme d'apprentissage abordé à la section 3.3.1.4 et ajouter la représentation de l'individu à la liste des modèles ;
- *retrain* : Répéter l'opération *addPerson* pour chacune des personnes de la banque ;
- *recognize* : Extraire la représentation du visage d'entrée et calculer les probabilités d'appartenance à chacun des modèles avec l'algorithme de Viterbi. La liste de probabilités est ensuite triée et les $K$ plus proches voisins parmi les représentations de la banque d'apprentissage sont sélectionnés.

Le module *RModuleHMM* possède un avantage certain lors de l'ajout d'une nouvelle personne car un ré-apprentissage complet est totalement inutile. Par ailleurs, les phases d'apprentissage et de reconnaissance sont particulièrement longues, nuisant donc à une utilisation en temps réel de ce module.

**RecognitionEngine** La dernière classe inclue dans le système de reconnaissance représente également la partie la plus importante. Comme son nom l'indique, le *RecognitionEngine* est le moteur qui permet au système de fonctionner. En effet, cette composante contient les modules de reconnaissance spécifiques ainsi qu'un lien avec la base de données. C'est finalement dans ce module que la décision ultime sur l'identité des personnes se réalisera.

Cette classe hérite tout d'abord d'une liste de pointeurs à des *RecognitionModule* initialement vide. Ceux-ci sont alloués dynamiquement et insérés en temps voulu. Il est à noter que le mécanisme propre aux classes abstraites fait ici tout le travail : l'engin de reconnaissance ne s'intéresse pas au type de module inséré mais plutôt au fait qu'il soit dérivé de la classe *RecognitionModule*.

Lorsque le *RecognitionEngine* devra ordonner des actions à chacun des modules, les fonctions virtuelles pures de la classe mère seront appelées et chacun procédera avec son implantation particulière. L'utilisation d'une classe abstraite garantie au *RecognitionEngine* que les modules spécifiques utilisent les mêmes entrées et sorties, ce qui rend la gestion plus agréable. De plus, les fonctions virtuelles pures garantissent également que chacune des fonctions est implémentée.

Les différentes méthodes particulières à cette classe sont les suivantes :

- *addModule* : Ajouter un module de reconnaissance spécifique au système. Le module peut être initialisé lors de sa création grâce à un fichier de configuration passé en arguments (si l'apprentissage a eu lieu) ;
- *recognize* : Appeler la fonction *recognize* de chacun des modules du système et appliquer une fonction de décision sur les listes ordonnées de votes générées par les différents algorithmes ;
- *reset* : Ré-initialiser les variables membres du système et détruire les modules de reconnaissance ;
- *retrain* : Commander un ré-apprentissage à chacun des modules de reconnaissance du système ;
- *save* : Ordonner la sauvegarde des données de chacun des modules.

Il est intéressant de noter que les fonctions *recognize* et *retrain* sont particulièrement propices à une parallélisation des calculs. En effet, l'exécution peut être distribuée efficacement sur les différentes unités de calculs d'un ordinateur de type multi-processeurs ou sur les nœuds de traitements d'une ferme d'ordinateurs (*cluster*). Pour ce faire, des modifications devraient être apportées aux fonctions pour y inclure la communication avec les différents modules distants.

Ensuite, plusieurs fonctions de décision pourraient également être disponibles à l'engin de reconnaissance. Ces fonctions seraient ainsi sélectionnées dynamiquement et utilisées pour générer la décision finale. Ce choix dynamique permettrait également la réalisation d'expérimentations et de comparaisons plus poussées sur les différentes méthodes de vote à utiliser.

#### 3.3.3.3 Forces et faiblesses

Un concept étant rarement parfait, plusieurs particularités du design peuvent être analysées et évaluées. La présente sous-section se veut donc être un bilan résumant les différentes forces et faiblesses de l'architecture développée.

**Forces** Voici dans un premier temps les différentes qualités de l'architecture logicielle :

- *Dynamique* : La plus grande force de l'architecture réside dans sa capacité à modifier dynamiquement la configuration et les paramètres du système. Ainsi, des modules de reconnaissance peuvent être ajoutés ou retirés sans toutefois requérir des changements à la fonction de décision. Des modifications peuvent également être apportées en temps réel à la base de données des personnes ;
- *Hétérogénéité* : L'utilisation du patron de design *strategy* couplé à une fonction de décision basée sur les votes permet d'utiliser n'importe quelle méthode de reconnaissance du visage et ce, peu importe son implémentation. Ainsi, le système développé utilise 4 algorithmes de bases différentes (2 de type global et 2 de type local). Cela étant dit, un module à base d'informations tridimensionnelles pourrait facilement être ajouté au système sans aucune modification[13] ;
- *Facilité d'emploi* : L'architecture logicielle est par ailleurs facile d'emploi, tant pour l'implémentation de nouvelles techniques que pour la gestion du système en cours d'utilisation. Pour l'ajout de modules de reconnaissance, de nouvelles classes héritant de *RecognitionModule* doivent être créées et chacune des méthodes virtuelles pures doit être surdéfinie. Pour ce qui est de la gestion du système, les différentes fonctions d'interface réalisent les tâches les plus importantes du système. Les entrées et sorties étant communes d'un module à l'autre, leur utilisation et leur interprétation sont donc simplifiées ;
- *Modularité* : La séparation des composantes en différentes classes favorise tout d'abord l'organisation générale du système. Cette modularité est par ailleurs particulièrement propice à une implémentation sous forme de librairies dynamiques (*DLL : Dynamic Loading Libraries*). L'utilisation de tels mécanismes facilitent le chargement d'un certain module en cours d'exécution ou à l'initialisation, un peu comme les plugiciels (c.-à-d. : *plug-ins*) dans certains logiciels populaires ;

[13] La classe *Human* devrait par ailleurs subir quelques modifications pour inclure une structure 3D.

- *Calcul distribué* : L'architecture proposée est également adéquate à une parallélisation des différentes phases du système. Ainsi, les modules de reconnaissance spécifiques pourraient être localisés sur divers nœuds de calculs (ou processeurs) d'un système distribué. Des fonctions de communication devraient alors être ajoutées dans le *RecognitionEngine* et le *RecognitionModule*. Cette dernière classe permettrait alors à toutes ses descendantes d'utiliser ces nouvelles fonctions. Le calcul distribué s'avère un atout majeur pour l'implémentation d'un système de reconnaissance utilisant une base de données volumineuse ;

- *Améliorations des fonctions de décision* : Les fonctions de décision basées sur les votes pourraient être remplacées par des algorithmes d'intelligence artificielle plus évolués. En effet, l'utilisation de réseaux de neurones serait notamment une avenue intéressante à explorer. Pour ce faire, la fonction *RecognitionEngine : :recognize* devrait être modifiée pour inclure le réseau (ou autre algorithme) et ainsi utiliser directement les données provenant des divers modules de reconnaissance. L'architecture peut donc être modifiée légèrement pour permettre des expérimentations supplémentaires sur les fonctions de décision ;

- *DSC* : L'architecture développée fournie les bases nécessaires à l'implémentation de la technique de sélection dynamique de classifieur (*DSC*). Il est effectivement possible de sélectionner n'importe quel module de classification parmi un ensemble donné afin de réaliser la reconnaissance d'un certain prototype. Ainsi, le système peut changer l'algorithme d'identification sélectionné à chacun des prototypes qui lui est présenté ;

- *Expérimentations* : Étant donné que la configuration du système peut être modifiée dynamiquement sans problème, il est possible de réaliser automatiquement un très grand nombre d'expérimentations. Par exemple, une comparaison de la performance pourrait être réalisée séquentiellement pour toutes les configurations possibles d'un multi-classifieur composé de $N$ modules. À ceci peut s'ajouter l'expérimentation de différentes fonctions de décision, de fonctions de classement (p. ex. : distance $L_1$), la persistance des résultats sur diverses banques de tests, *etc.* ;

- *Base de données* : Par souci d'économie de ressources informatiques, toutes les informations reliées aux personnes à reconnaître sont regroupées dans une seule et unique base de données. Celle-ci est couplée à l'engin de reconnaissance principal et est accessible aux modules spécifiques via ce dernier. Dans le cas d'une

application distribuée, cette base de données serait également située sur un seul ordinateur et accessible via le *RecognitionEngine*. Cette centralisation des données élimine donc les redondances et permet une économie substancielle d'espace de stockage ;

**Faiblesses et limitations** Les points suivants représentent cette fois-ci les différentes faiblesses et limitations de l'architecture logicielle :

- *Type de MC* : Tout d'abord, un seul type de multi-classifieur est supporté par l'architecture, soit ceux de type parallèle. Les classifieurs en cascades et hiérarchiques ne sont donc pas supportés directement. L'architecture devrait subir des modifications majeures pour permettre l'utilisation de d'autres types de MC ;
- *Phase de test* : Lorsque des expérimentations séquentielles sont effectuées sur différentes configurations, plusieurs étapes inutiles sont réalisées. En effet, pour une configuration particulière, tous les prototypes d'une banque test sont analysés pour fins d'identification. Cependant, les résultats seront identiques pour la configuration suivante. Des modifications doivent donc être apportées pour amoindrir ce problème. Ainsi, chacun des modules peut générer une liste contenant toutes les listes ordonnées de votes correspondant à une certaine banque test. Par la suite, cette banque peut être lue et utilisée directement pour mesurer la performance d'une configuration spécifique. Cette solution est implantée dans le système mais requiert par ailleurs certaines précautions liées à son utilisation ;
- *Asymétrie des listes de votes* : Les listes de votes peuvent parfois être asymétriques. C'est le cas par exemple de celles générées par les HMM qui ne contiennent que $N$ votes[14] versus celles produites par les DCT, EF et EO qui en possèdent $\sum_{i=1}^{N} Q_i$, où N est le nombre d'individus et $Q_i$ le nombre d'images pour la personne $i$. Lorsque ces listes sont fusionnées par la fonction de décision, les votes provenant du module *HMM* sont moins significatifs ou dérangeants. Par exemple, supposons qu'un individu possédant 5 images au total est identifié correctement dans les 5 premières positions des modules DCT, EF et EO, et en première position du *HMM*. Les 4 autres positions de ce dernier ne pourront pas être pour le même individu, ce qui viendra troublé la fonction de décision. Pour l'instant, aucune

[14]Les *HMM* utilisent en effet plusieurs images d'un même individu pour générer son modèle. Lors de la phase d'identification, les modèles sont testés et ordonnés du plus au moins probable, produisant donc une liste de $N$ votes.

solution n'est proposée ou implémentée dans le système. Par ailleurs, il pourrait être envisageable d'ajouter des votes fantômes pour un individu possédant une bonne longueur d'avance (c.-à-d. : un écart plus élevé que la moyenne) sur son plus proche rival.

## 3.4 Conclusion

Plusieurs algorithmes de reconnaissance ont été présentés tout au long de ce chapitre. Parmi ceux-ci, certains ne pouvaient être appliqués pour des raisons matérielles ou parce qu'ils transgressaient certaines conditions du projet. Notons tout particulièrement toutes les techniques intrusives ainsi que celles utilisant des informations tridimensionnelles.

C'est après maintes réflexions que quatre techniques ont été sélectionnées pour faire partie d'un système multi-classifieur, soient les *EigenFaces*, les *EigenObjects*, DCT et les *Embedded HMM*. Les raisons justifiant ces décisions reposent sur un compromis entre les taux de reconnaissance et les temps d'exécution des différentes méthodes.

Pour ce qui est du multi-classifieur utilisé, il est de type parallèle et il est jumelé à une fonction de décision basée sur les votes. Le compte de Borda s'avère être une de fonction de votes parmi les plus performantes.

Afin d'implémenter efficacement le système, plusieurs classes en C++ ont été créées. Basées sur le patron de design comportemental d'abstraction, ces classes permettent notamment une gestion flexible des différents modules de reconnaissance. Qui plus est, une base de données centralisée regroupe toutes les informations nécessaires sur les individus à identifier, économisant donc les ressources de stockage.

Finalement, cette architecture logicielle favorise les expérimentations de différentes configurations multi-classifieurs grâce à ses capacités de modification dynamique. Les résultats expérimentaux reliés à la reconnaissance du visage seront présentés au chapitre suivant.

# Chapitre 4

# Résultats expérimentaux

Plusieurs expérimentations ont été réalisées afin d'évaluer la performance relative des différents algorithmes sélectionnés, tant sur le plan de la détection que de la reconnaissance. En effet, il est intéressant d'évaluer l'impact de la qualité de la détection du visage sur la performance globale du système. De plus, diverses configurations multi-classifieurs furent testées à l'aide de deux banques d'images : la FERET et la AR-face.

## 4.1 Introduction

La reconnaissance du visage par vision numérique est, comme démontrée précédemment, très complexe et très variée. Les différentes méthodes envisageables possèdent des avantages et des inconvénients qui doivent être considérés lors du design d'un

système complet d'identification. Pour ce faire, il est primordial de valider les techniques choisies sur des ensembles de données relativement volumineux. Même si de telles images ne représentent pas exactement les conditions réelles d'utilisation, elles procurent néanmoins une idée fiable du comportement des différents modules dans un environnement contrôlé.

Ainsi, plusieurs banques d'images ont été créées afin de comparer les différentes méthodes entre elles selon diverses conditions (c.-à-d. : éclairage, pose, occultations, *etc.*). Parmi celles-ci, il y a notamment la FERET [48], AR-face [41], AT&T (appelée auparavant Olivetti), X2MVTS [43], Yale, MIT, Achermann ainsi que plusieurs autres. Chacune d'entre elles possède évidemment ses particularités spécifiques ainsi que ses qualités et défauts[1].

Ce dernier chapitre exposera donc à la section 4.2 les différentes bases d'images retenues pour les expérimentations, soient la FERET et la AR-face. Ensuite, la section 4.3 abordera le protocole expérimental utilisé lors des tests et finalement, la section 4.4 présentera de nombreux résultats expérimentaux.

## 4.2 Banque d'images

Peu importe le problème de reconnaissance des formes, un point commun demeure toujours présent : la nécessité d'utiliser un ensemble de données volumineux, représentatif et standardisé. Cette particularité est effectivement primordiale pour la comparaison de techniques ou d'algorithmes, permettant ainsi une évaluation relative des performances.

Cela étant dit, plusieurs points importants sont à considérer lors de la création ou de la sélection d'une banque d'images. Voici donc les particularités majeures à tenir en compte :

– *Nombre de personnes* : La quantité d'individus dans une banque d'images est un des points le plus important. En effet, ce nombre influence directement le niveau de difficulté de la banque : plus la quantité est élevée, plus la tâche de reconnaissance sera difficile. De surcroît, la banque représentera davantage les

[1] Certaines sont également très difficiles à obtenir alors que d'autres sont gratuites.

tâches d'identification en situations réelles, qui contiennent au minimum plusieurs milliers de personnes à identifier ;

- *Nombre d'images par individu* : Une certaine quantité d'images est habituellement disponible pour chaque personne de la base de données. Un nombre élevé procure généralement un meilleur apprentissage du module d'identification. Certaines banques d'images n'offrent cependant qu'une seule image d'entraînement par individu, ce qui complexifie énormément le problème ;
- *Hommes/femmes* : Le ratio d'hommes et de femmes représente une particularité intéressante. Étant donné que certaines différences relatives au genre peuvent être modélisées efficacement[2], une banque ne contenant que des hommes ne pourra être de difficulté égale à une autre contenant 50% de femmes. Finalement, il y a habituellement un plus grand nombre de femmes portant les cheveux longs, ce qui peut influencer certains algorithmes de reconnaissance ;
- *Arrière-plan* : La plupart des banques d'images contiennent des photos avec un arrière-plan neutre ou de couleur blanche. Les conditions d'acquisition ne sont par contre pas toujours idéales, occasionnant parfois la présence d'objets nuisibles ou d'arrière-plans complexes ;
- *Dimension des images* : La taille en pixels des images n'a généralement pas beaucoup d'influence sur les algorithmes de reconnaissance. Il existe cependant des dimensions minimales nécessaires à une représentation fidèle et unique de l'individu ;
- *Couleurs/tons de gris* : L'utilisation de couleurs dans les techniques d'identification est peu répandue. Celle-ci peu par contre s'avérer forte utile pour une détection des pixels représentant la peau ou pour la pré-classification d'individus de races différentes ;
- *Coordonnées cartésiennes des composantes du visage* : Ces informations supplémentaires s'avèrent particulièrement pratiques pour la comparaison de méthodes de reconnaissance. En effet, les résultats obtenus ne dépendant pas de la qualité de la détection du visage, des analyses plus robustes et plus représentatives peuvent être réalisées ;

[2]Un exemple de ce type de différence réside dans la taille de la tête. Une fois normalisé à partir des yeux, le visage d'un homme est en moyenne plus grand que celui d'une femme, aidant donc à discriminer certains individus.

- *Cas particuliers ou difficiles* : Des conditions spéciales peuvent également être présentes dans les bases d'images. Citons notamment les cas d'occultations (p. ex. : lunettes fumées, chapeau, bandeau, cigares, *etc.*), d'expressions faciales variées (p. ex. : sourire, grimace, yeux fermés, *etc.*), de changements corporels (p. ex. : barbe, moustache, maquillage, verres de contact de couleurs, couleurs de cheveux, cheveux détachés, *etc.*) et d'éclairage (p. ex. : incandescent, directionnel, *etc.*) ;
- *Pose* : La pose de la tête de l'individu représente finalement un autre point important. En effet, la reconnaissance d'un visage de profil sera différente d'un visage orienté à 45 degrés et nécessitera un ajustement des techniques d'apprentissage.

Il y a donc plusieurs propriétés importantes à vérifier lors de la sélection d'une banque d'images pour fins d'expérimentations. Ces particularités s'appliquent également lors de la création d'une banque d'images.

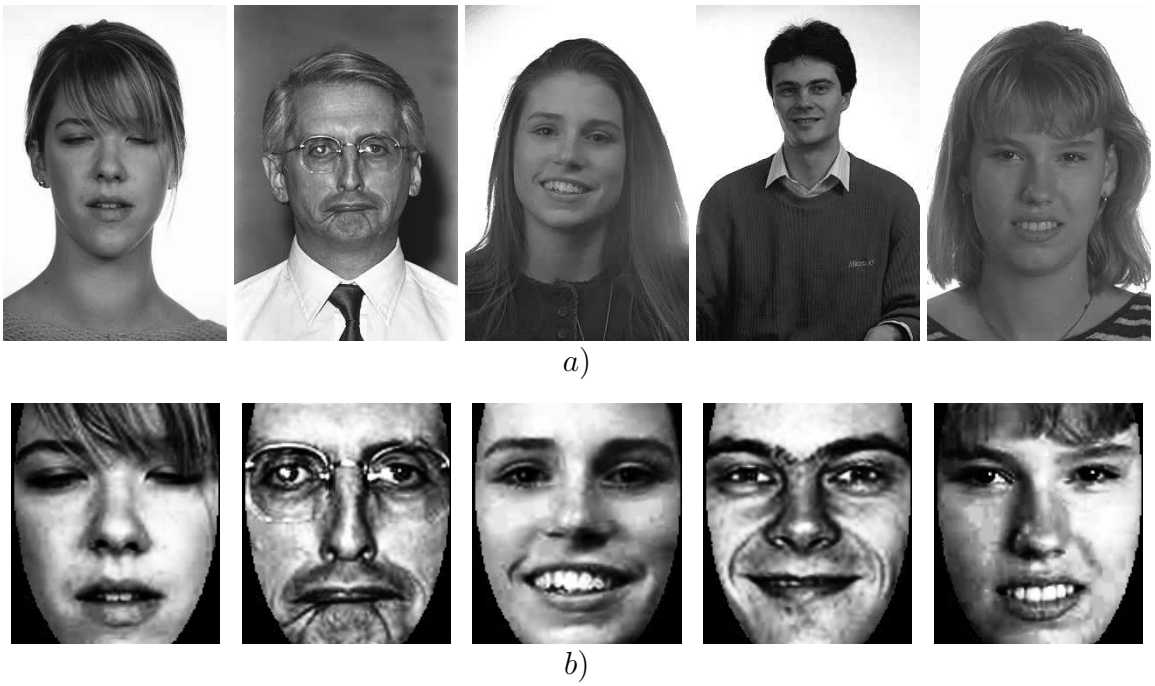
Dans le cadre de ce projet, trois banques d'images ont été utilisées, soient la FERET, la AR-face et la LVSN. Cette dernière est la seule des trois à avoir été conçue spécialement pour le projet et était destinée aux expérimentations en temps réel de l'application.

Les facteurs ayant favorisés leur sélection résident entre autres dans la grande quantité d'images disponibles, du degré de complexité et de la présence de couleurs dans les photos. Les sous-sections suivantes présenteront chacune de ses banques d'images avec moult détails.

### 4.2.1 FERET

Le programme FERET [48] fût démarré en 1993 dans le but de comparer les différents algorithmes de reconnaissance disponibles à ce moment-là. Une série de compétitions entre différentes institutions (universités) fût également instaurée. Un protocole expérimental a alors été développé conjointement avec la création d'une banque d'images impressionnante pour uniformiser ces expérimentations.

Tout d'abord, la FERET contient à elle seule 14 126 images de 1 199 hommes et femmes de toutes races confondues. Les photos qu'elle renferme sont de faibles dimen-



*a)*

*b)*

**Fig. *4.1:*** *Banque d'images FERET : exemples. a) Images originales (dim. :* $256 \times 384$*) et b) Images normalisées (dim. :* $130 \times 150$*).*

sions (256×384) et quantifiées en tons de gris sur 8 bits (256 niveaux). La figure 4.1 illustre notamment quelques images extraites de la FERET.

Chaque individu possède évidemment plusieurs images représentant différentes caractéristiques :

- Séances différentes et temporellement espacées (c.-à-d. : plus d'un an) ;
- Variation de la pose : rotation de la tête selon des angles prédéfinis ;
- Acquisition d'images avec des caméras et un éclairage différents ;
- Expressions faciales variées ;
- Altération numérique des photos ;
- Ajout ou élimination d'objets (p. ex. : lunettes), *etc.*

Pour faciliter la comparaison des différentes méthodes, un protocole expérimental fût élaboré. Des sections standards contenant des centaines d'images furent également définies et se divisant en deux catégories : les *gallery* et les *probe*.

Alors que le sous-groupe *gallery* contient les images d'apprentissage, le sous-groupe *probe* regroupe quant à lui les images utilisées lors de la vérification. Un résumé des différentes sections définies est illustré au tableau 4.2.1.

| Catégorie | Taille banque test | Taille banque d'apprentissage |
|---|---|---|
| FB | 1195 | 1196 |
| duplicate I | 722 | 1196 |
| fc | 194 | 1196 |
| duplicate II | 234 | 864 |

**Tab. *4.1:*** *Taille des sections de la banque d'images FERET.*

Il est intéressant de noter que les trois premières banques test utilisent la même base d'apprentissage. Chacune d'entre elles regroupe des images possédant des caractéristiques similaires mais différentes de la base d'entraînement : expressions faciales différentes (FB), images frontales dupliquées (duplicate I), caméra et éclairage variés (fc) et images frontales dupliquées acquises au moins un an plus tard (duplicate II) [48].

Dans tous les cas, les sections de vérification peuvent contenir plusieurs images par personne[3] contre une seule image d'apprentissage ; l'objectif est donc de déterminer l'image correspondante parmi plusieurs. La catégorie de tests utilisant la section FB contient par exemple 1 195 images test pour 1 196 images d'apprentissage (c.-à-d. : 1 image par personne donc 1 196 individus).

Cela représente en somme un problème très difficile car contrairement à certains domaines de reconnaissance des formes (p. ex. : reconnaissance de caractères manuscrits) qui possèdent peu de classes et beaucoup d'exemples, la reconnaissance des visages sur la banque FERET repose sur un grand nombre de classes comptant très peu de prototypes.

Dans un autre ordre d'idée, les coordonnées des yeux, du nez et de la bouche (coins gauche et droit) sont fournies pour 3 816 images de la banque, ce qui représente en pratique l'ensemble des images contenues dans les sections prédéfinies.

Finalement, les raisons pour lesquelles la banque d'images FERET fût retenue reposent essentiellement sur sa taille et sa diversité. En effet, dû au nombre élevé d'individus contenus dans la banque, celle-ci représente une tâche complexe d'identification et un bon défi pour tout système de reconnaissance.

[3] Certains individus ne sont pas représentés dans la banque d'images test.

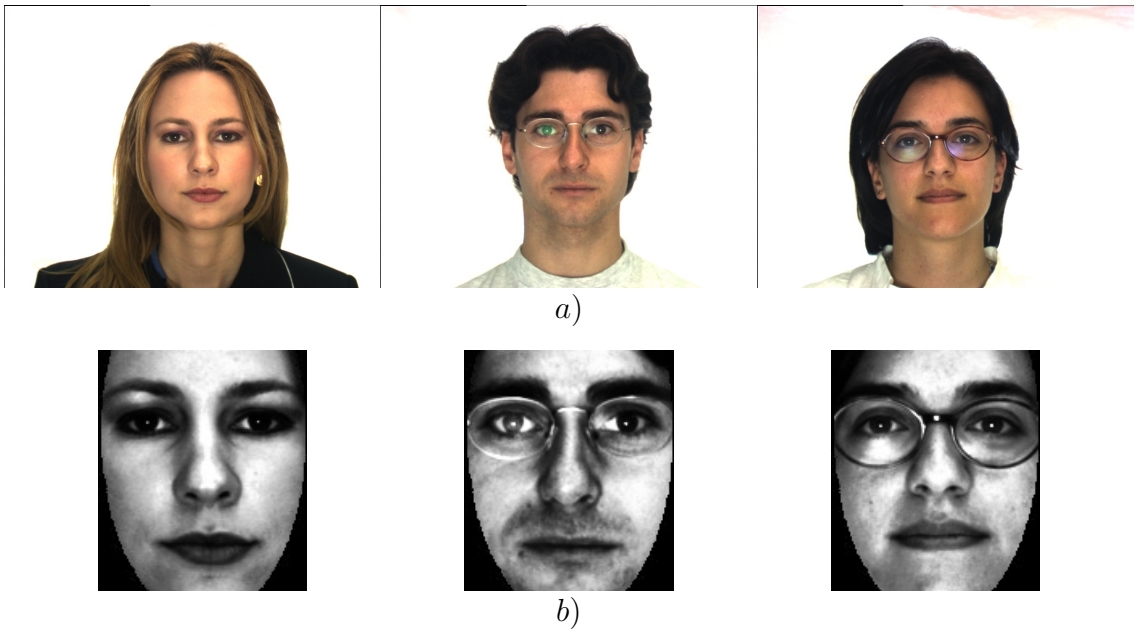


**Fig. *4.2:*** *Banque d'images AR-face : exemples. a) Images originales (dim. : 768×576) et b) Images normalisées (dim. : 130×150).*

Certains désavantages incombent cependant à son utilisation. Tout d'abord, les algorithmes utilisant la couleur doivent être rejetés car les images sont en tons de gris. De plus, l'identification de personnes avec une seule image d'apprentissage par personne est une pâle imitation de la réalité et semble peu approprié pour tester un système devant par exemple reconnaître des dizaines de milliers d'individus.

### 4.2.2 AR-face

La banque d'images AR-face [41] fût quant à elle créée au Computer Vision Center (CVC) de l'UAB (Universitat Autònoma de Barcelona) en Espagne. Elle contient plus de 4 000 images frontales de 135 personnes[4], soit 77 hommes et 58 femmes. La figure 4.2 contient 3 photos provenant de cette banque d'images. Pour celles-ci, les résultats du processus de normalisation du visage sont également illustrés.

Les images ont été acquises en deux sessions espacées de 14 jours. Chaque personne

[4]La documentation officielle ne mentionne que 126 personnes alors qu'elle en contient 135 différentes. L'écart entre les deux nombres provient du fait que certaines personnes n'étaient pas présentes aux deux sessions de photographies.

possède alors jusqu'à 26 images (2×13) contenant des expressions faciales variées (c.-à-d. : sourire, neutre et colère), des occultations (c.-à-d. : lunettes fumées et foulard), des variations d'éclairage (c.-à-d. : gauche, droite et uniforme frontal) ainsi que des combinaisons des conditions particulières précédentes. Ces différentes particularités sont bien identifiées dans les noms de fichiers, ce qui permet des expérimentations et/ou validation de l'impact d'un seul effet (p. ex. : l'influence d'un sourire sur l'identification d'une personne).

Pour ce qui est du côté plus technique de la banque, les images sont en couleurs (24 bits) et de relativement grandes dimensions (768×576). L'arrière-plan est généralement neutre sauf quelques exceptions possédant un reflet rosé. Il est par ailleurs important de noter qu'aucune coordonnée des caractéristiques du visage n'est fournie avec cette banque d'images.

Pour terminer, l'intérêt porté envers cette banque est justifié en partie par la présence d'images couleurs. Cette caractéristique permet la vérification de l'efficacité de certains algorithmes qui l'utilisent. Qui plus est, la couleur procure également l'information supplémentaire[5] nécessaire à l'estimation de la précision du module de détection du visage ainsi que de son impact sur le taux de reconnaissance.

### 4.2.3 LVSN

Une dernière banque d'images a été utilisée pour les expérimentations en temps réel du système, soit un ensemble d'images de 10 personnes membres du LVSN. La caméra utilisée est une Logitech® QuickCam® Pro 3000 qui génère des images couleurs de 640×480 pixels. De plus, une interface conviviale permet de configurer les différents paramètres de la caméra, comme entre autres un ajustement du *white balance* qui permet de corriger les couleurs dans des environnements particuliers (p. ex. : lumière incandescente, fluorescente, *etc.*).

Pour chaque acquisition, l'arrière-plan est retiré automatiquement et les images sont sauvegardées (originale, visage détecté et normalisé). Il est important de mentionner

[5] L'algorithme de détection du visage détaillé au chapitre 2 utilise la couleur pour localiser les pixels de l'image représentant la peau.

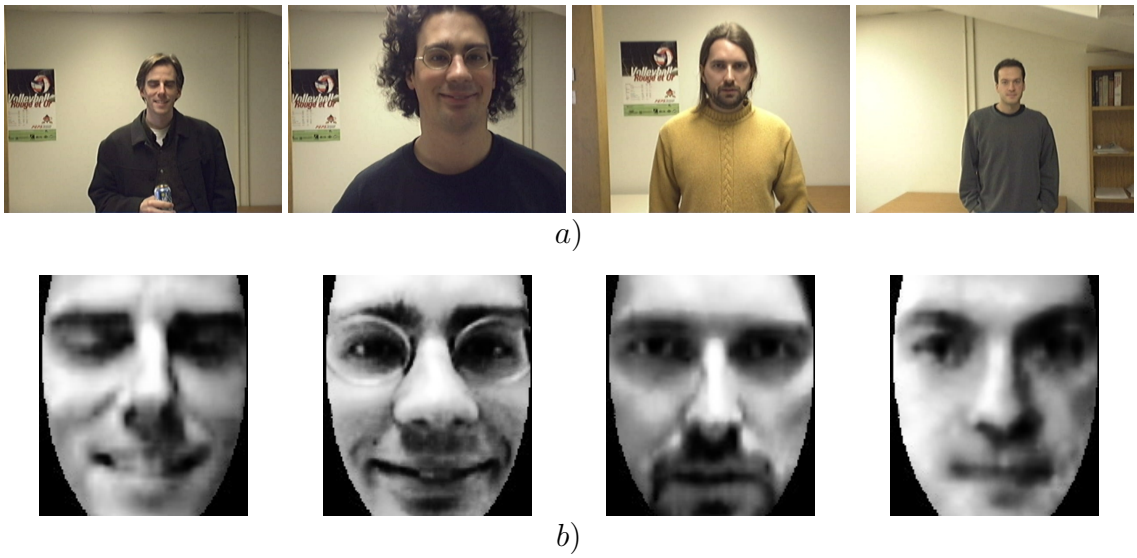


**Fig. *4.3:*** *Banque d'images LVSN : exemples. a) Images originales (dim. : 640×480) et b) Images normalisées (dim. : 130×150).*

que les coordonnées des yeux ont été ajustées manuellement pour éviter des erreurs lors de l'apprentissage. La figure 4.3 illustre finalement quelques images provenant de cette banque.

## 4.3 Protocole expérimental

Un protocole expérimental rigoureux permet avant tout la structuration des étapes nécessaires à l'exécution d'une certaine tâche. Dans le cas présent, l'objectif est de préparer les images brutes pour les expérimentations.

Les étapes suivantes forment donc le protocole expérimental utilisé :

1. Vérification des coordonnées des caractéristiques du visage fournies avec la banque d'images ;
2. Normalisation des images à partir des coordonnées des yeux selon la procédure décrite au chapitre 2 ;
3. Apprentissage des différentes méthodes de reconnaissance ;
4. Réalisation des expérimentations.

Un protocole rigoureux permet alors la répétabilité des expérimentations. Ainsi,

quiconque désirant reproduire les résultats pour fins de vérification ou de comparaison pourra alors suivre la recette fournie par le protocole. De cette façon, la procédure utilisée est standard et assure qu'aucune modification ou altération n'est réalisée dans le processus.

Pour ce qui est de l'expérience portant sur l'impact de la détection automatique du visage, la première étape fût tout simplement réalisée par le module approprié présenté précédemment au chapitre 2. Ce dernier génère les coordonnées cartésiennes des centres des yeux pour chacune des images.

**AR-face database** Dans le cas de la banque d'images AR-face, la première étape concernant la vérification des coordonnées ne peut avoir lieu car ces informations n'accompagne pas la banque d'images. Cette tâche fût donc réalisée manuellement afin de localiser les centres des yeux pour les 679 images utilisées.

De plus, un partitionnement de la banque est nécessaire pour former les sections d'apprentissage et de test. Pour ce faire, les images ont été triées selon leur caractéristique principale et regroupées selon les règles suivantes :

- *Entraînement* : Cette première section contient 263 images regroupant les photos "-1" (c.-à-d. : session 1 et expression faciale neutre) et "-2" (c.-à-d. : session 1 et sourire) ;
- *Vérification* : Pour ce qui est de la banque test, elle est formée de 416 images regroupant les photos "-3" (c.-à-d. : session 1 et grimace ou colère), "-14" (c.-à-d. : session 2 et expression faciale neutre), "-15" (c.-à-d. : session 2 et sourire) et "-16" (c.-à-d. : session 2 et grimace ou colère).

Selon cette segmentation, aucune image ne peut se trouver dans les deux banques simultanément. Finalement, les images couleurs sont converties en tons de gris car cette information n'est pas nécessaire aux algorithmes de reconnaissance utilisés.

## 4.4 Résultats expérimentaux

La présente section regroupe le fruit des expérimentations réalisées dans le cadre du projet de reconnaissance d'individus par vision numérique. Les sous-sections suivantes présenteront donc différents aspects ayant été évalués.

Tout d'abord, la sous-section 4.4.1 présentera des expériences sur la robustesse de la méthode *EigenFaces*. La sous-section 4.4.2 abordera ensuite les performances individuelles des quatre techniques de reconnaissance du visage retenues.

La sous-section 4.4.3 traitera de l'impact des métriques sur les algorithmes de reconnaissance suivie de leurs temps d'exécution à la sous-section 4.4.4. La sous-section 4.4.5 présentera quant à elle les résultats multi-classifieurs pour terminer à la sous-section 4.4.6 avec les impacts de la détection automatique du visage sur l'identification.

### 4.4.1 Robustesse de la méthode *EigenFaces*

Certaines expérimentations ont démontrées [36] que la méthode des *EigenFaces* demeure relativement robuste à certaines erreurs de détection et/ou de normalisation. Cette étude avait pour objectif principal de mesurer l'impact de transformations contrôlées sur le taux de reconnaissance afin d'établir les limites acceptables d'un module de détection du visage.

Les différents effets étudiés tendent à simuler les nombreuses erreurs qui peuvent survenir lors de la détection et de la normalisation des visages. Ainsi, les transformations suivantes, illustrées à la figure 4.4, ont été évaluées :

- *Translation horizontale* : Ce premier paramètre correspond en pratique à une erreur de détection des yeux. Ceci génère alors un visage normalisé légèrement décalé par rapport aux visages d'apprentissage ;
- *Translation verticale* : Idem à la translation horizontale ;
- *Downsampling* : Ce troisième paramètre vise à simuler l'effet d'un agrandissement d'images (c.-à-d. : un *zoom*) sur un visage de petite taille. Cette transformation se produit lorsqu'une personne est loin de la caméra et qu'une identification est tout

**Fig. *4.4:*** *Exemples de transformations. De gauche à droite : image originale, après* downsampling *(90%), après changement d'échelle (−17.5%), après une rotation (20°), après morphing (30%) et finalement, après un changement de luminance (×1.4).*

de même tentée. L'image utilisée pour l'identification est donc de mauvaise qualité dû à l'interpolation nécessaire à son agrandissement. Cette simulation est réalisée par deux redimensionnements consécutifs de l'image et appliquées respectivement selon un algorithme d'interpolation linéaire et cubique. Un *downsampling* de 80% signifie donc qu'une image de dimensions 200×220 est tout d'abord redimensionnée à 40×44 pour revenir ensuite à sa dimension d'origine. De cette manière, l'image perd largement en qualité et crée alors un effet de brume ;

– *Facteur d'échelle* : Pendant la normalisation du visage, un facteur d'échelle est appliqué sur l'image afin d'aligner les yeux à une position bien précise. Mais lorsque cette étape de localisation est erronée, le visage généré est trop grand ou trop petit par rapport aux prototypes d'entraînement, ce qui, à une certaine limite, engendre de mauvaises classifications. Un facteur d'échelle de -40% signifie que l'image tranformée possèdera une taille 40% plus petite ;

– *Rotation* : La rotation correspond encore une fois à une mauvaise détection des yeux, engendrant cette fois une rotation parallèle au plan image ou encore autour d'un axe normal à l'image et passant par le nez ;

– *Morphing* : Le *morphing* est un paramètre visant à recréer l'effet d'une rotation axiale de la tête par rapport à la colonne vertébrale (c.-à-d : rotation de la tête

| Transformations | Intervalles utilisés |
|---|---|
| **Translation horizontale** | 0 à 40% de la largeur |
| **Translation verticale** | 0 à 40% de la hauteur |
| **Downsampling** | 50 à 90% |
| **Facteur d'échelle** | ±40% |
| **Rotation** | ±40 degrés |
| **Morphing** | 0 à 75% |
| **Luminance** | 0 à 1.5 |

**Tab.** ***4.2:*** *Intervalles utilisés pour les paramètres des transformations étudiées.*

vers la gauche ou la droite). Ceci est réaliser en pratique en compressant une moitié de l'image et en étirant la seconde (p. ex. : un facteur de 30% pourrait par exemple être appliqué de chaque côté). Ce paramètre ne vise en rien à reproduire une transformation affine, mais vise plutôt à générer une image similaire qui serait produite par ce genre de rotation ;

– *Luminance* : L'éclairage demeure l'un des effets extérieurs influençant l'efficacité des différents modules de reconnaissance. Afin de simuler une variation globale de l'intensité, l'image normalisée est convertie en HLS et un facteur multiplicatif est appliqué sur le canal L (c.-à-d. : luminance) ;

– *Combinaison* : Finalement, tous les paramètres ont graduellement été combinés ensemble afin de simuler le *pire des cas*.

Pour réaliser ces expériences, un protocole expérimental similaire à celui présenté à la section 4.3 a été développé. Parmi les nombreuses images de la banque d'images AR-face [41], une image fût sélectionnée pour chacune des 126 personnes[6].

Après avoir effectué la normalisation des images, chacun des paramètres est utilisé indépendamment pour altérer l'image synthétiquement. Les intervalles utilisés pour chacune des transformations sont illustrés au tableau 4.2. Les images modifiées sont ensuite utilisées avec l'algorithme des *EigenFaces* pour vérifier l'impact sur le taux de reconnaissance. Lorsqu'aucune modification n'est appliquée sur les images, une performance de 100% est atteinte. Cela revient à dire que toutes les images ont été correcte-

[6]Ces images correspondent à un des deux clichés contenant une vue frontale où la personne arbore une expression faciale neutre.

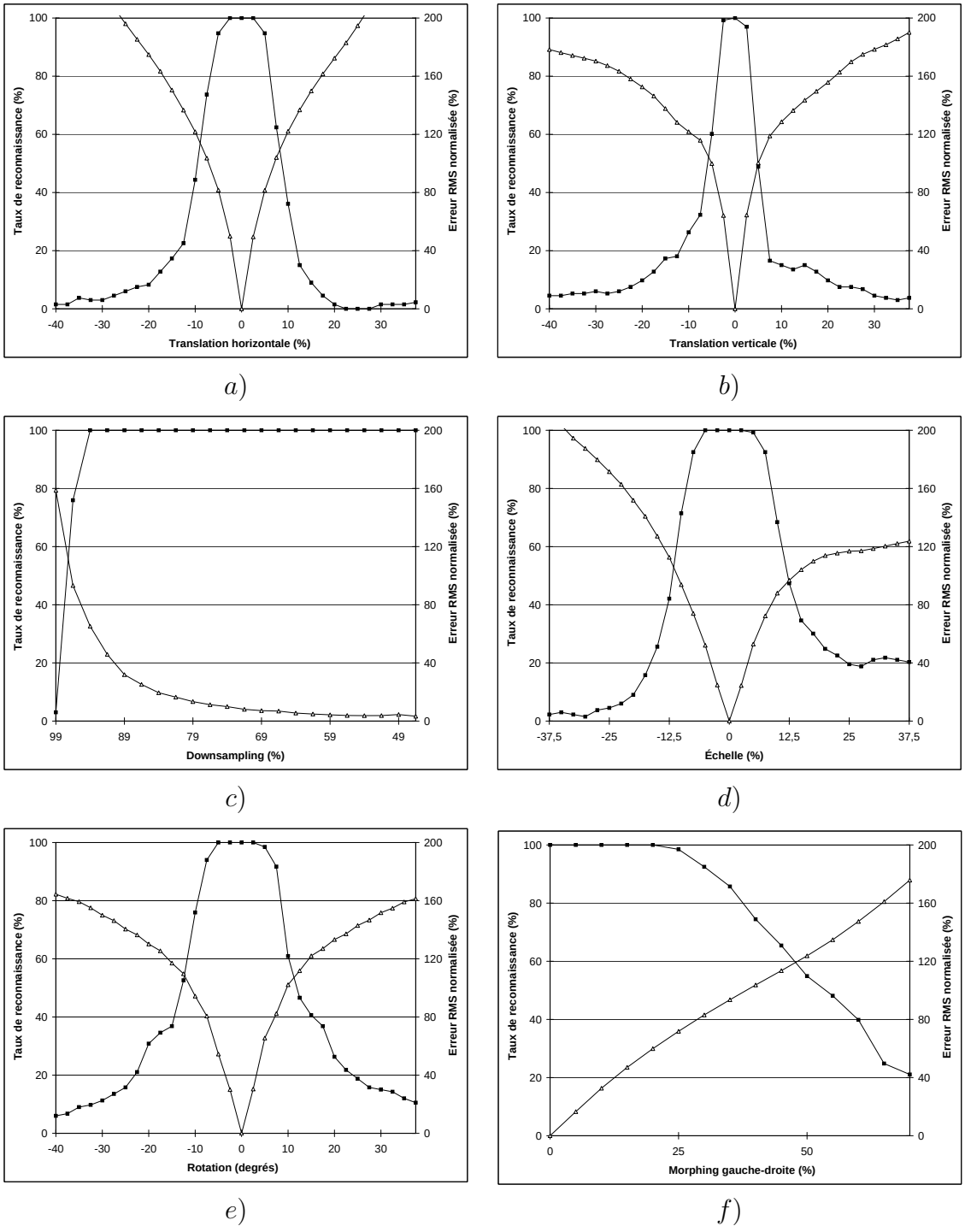
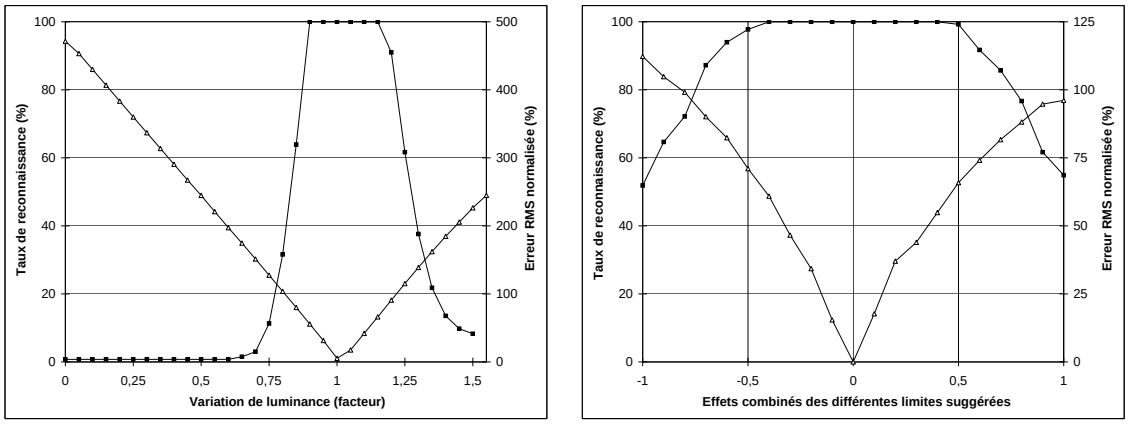



**Fig.** ***4.5:*** *Effets des paramètres sur la méthode des* EigenFaces *: a) translation horizontale, b) translation verticale, c)* downsampling*, d) facteur d'échelle, e) rotation et f)* morphing *gauche-droite. Légende :* ■ *— Taux de reconnaissance (échelle de gauche);* △ *— Erreur RMS normalisée (échelle de droite).*




*g)* *h)*

**Fig.** ***4.6:*** *Effets des paramètres sur la méthode des* EigenFaces *(suite) : a) luminance et b) combinaison des différents paramètres. Légende :* ■ *— Taux de reconnaissance (échelle de gauche) ;* △ *— Erreur RMS normalisée (échelle de droite).*

| **Transformations** | **Limites suggérées** |
|---|---|
| **Translation horizontale** | ±5% de la largeur |
| **Translation verticale** | ±3% de la hauteur |
| **Downsampling** | $< 90\%$ |
| **Facteur d'échelle** | ±5% |
| **Rotation** | ±5 degrés |
| **Morphing** | $< 20\%$ |

**Tab.** ***4.3:*** *Limites suggérées pour les transformations étudiées.*

ment identifiées.

Les figures 4.5 et 4.6 illustrent les différents résultats obtenus. Il est intéressant de remarquer que la méthode est assez robuste aux variations, surtout lorsqu'elles sont combinées simultanément.

Finalement, les différents graphiques obtenus ont permis d'établir les limites acceptables et nécessaires pour que l'algorithme des *EigenFaces* soit efficace. Celles-ci sont résumées au tableau 4.3.

### 4.4.2 Performances individuelles des modules de reconnaissance

Afin d'évaluer l'efficacité du système de reconnaissance multi-classifieur, il faut tout d'abord évaluer la performance individuelle des différents modules d'identification. Pour ce faire, chacun d'entre eux est entraîné à partir des sections d'apprentissage des banques d'images et vérifié à l'aide de la section test correspondante.

Les résultats expérimentaux pour les banques FERET et AR-face sont illustrés aux figures 4.7 et 4.8. Pour chacun de ces graphiques, les taux de reconnaissance cumulatifs pour les trois premières positions sont rapportés pour chacune des techniques.

Le taux de reconnaissance cumulatif représente le pourcentage de bonnes identifications réalisées à un certain niveau $N$ de tolérance (c.-à-d. : *Top N* où N = 1,2 ou 3). Au niveau 3 par exemple, le taux de reconnaissance sera composé du pourcentage de bonnes identifications réalisées au premier niveau plus celles qui sont correctes aux niveaux 2 et 3 (p. ex. : *Top 3* $= 80\% + 5\% + 2\% = 87\%$).

Concernant la métrique utilisée pour générer les résultats de cette sous-section, la distance $L_1$ fût utilisée exclusivement. Les raisons justifiant cette décision seront abordées à la sous-section 4.4.3.

**FERET** Pour cette première banque d'images, la section test FB a été utilisée pour réaliser les expérimentations. Celle-ci contient 1 195 images qui doivent être associées à 1 196 personnes.

Pour ce qui est des paramètres utilisés par chacun des modules d'identification, le tableau 4.4 résume les principales valeurs employées.

À partir des résultats illustrés à la figure 4.7, les analyses suivantes peuvent être réalisées :

– La méthode de reconnaissance des *EigenFaces* obtient le meilleur taux de reconnaissance de premier niveau (c.-à-d. : *Top 1*) avec plus de 81% suivie par les HMM avec un peu moins de 80% ;

| Algorithmes | Paramètres |
|---|---|
| ***EigenFaces*** | 200 premiers vecteurs propres |
| ***EigenObjects*** | 25 premiers vecteurs propres par caractéristique (yeux+nez) pour un total de 75 valeurs |
| **DCT** | 192 coefficients |
| **HMM** | 5 états primaires<br>(3 6 6 6 3) états incorporés |

**Tab.** ***4.4:*** *Paramètres utilisés par les modules d'identification.*

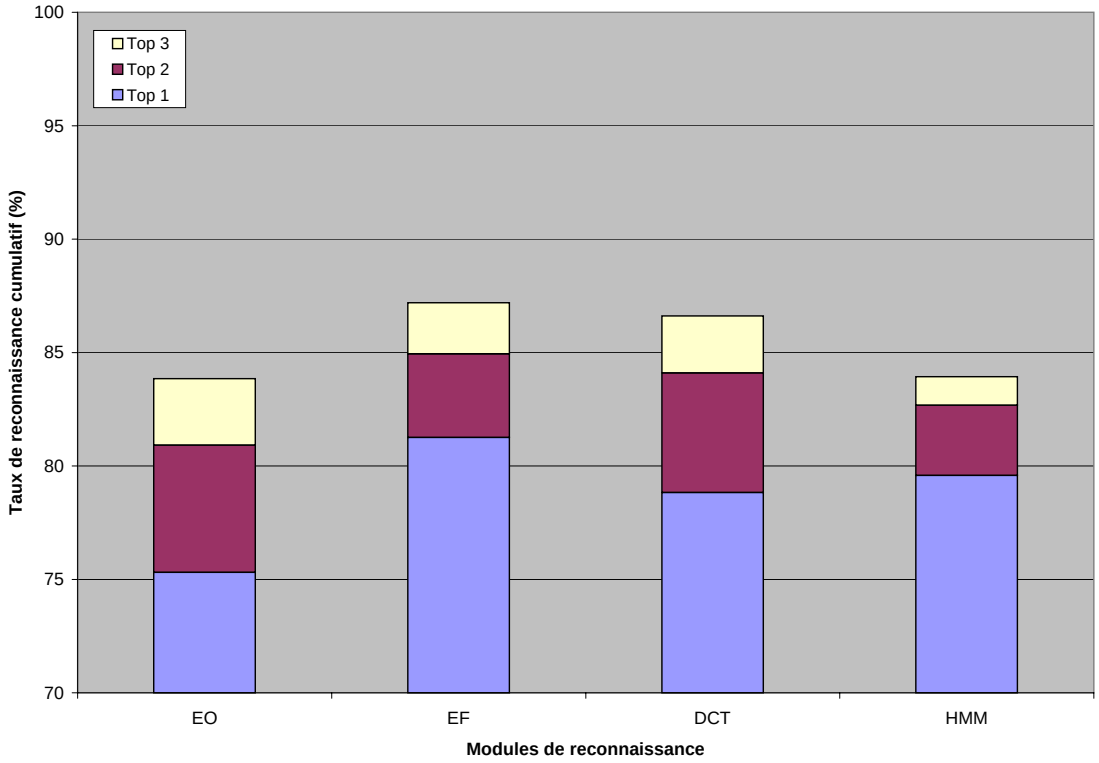



**Fig.** ***4.7:*** *Résultats expérimentaux sur la base d'images FERET pour différentes méthodes de reconnaissance. La section test utilisée est la FB et contient 1 195 images.*

- Au deuxième niveau cumulatif, le portrait change légèrement et c'est l'algorithme utilisant la DCT qui est en deuxième position avec un peu moins de 85% ;
- Dans tous les cas, les *EigenObjects* obtiennent la dernière position avec un peu plus de 75% pour le *Top 1* ;
- Finalement, les techniques EF et DCT génèrent des taux de reconnaissance similaires lorsque l'on tient compte du *Top 3* uniquement.

Les HMM offrent ici une performance un peu décevante face aux autres méthodes d'identification. Il est intéressant de remarquer que la banque d'apprentissage correspondante ne contient qu'une seule image par personne, ce qui ne favorise pas les HMM qui sont particulièrement adaptés à réaliser leur apprentissage avec plusieurs images par individu (c.-à-d. : 2 images et plus).

**AR-face** Pour ce qui est de la banque d'images AR-face, le portrait est légèrement différent. En effet, les individus disposent cette fois de deux images d'entraînement, seulement quelques uns n'en possèdent qu'une seule.

Les résultats expérimentaux sont illustrés à la figure 4.8 et ont permis la formulation des commentaires suivants :

- Les HMM obtiennent sans l'ombre d'un doute les meilleurs taux de reconnaissance à n'importe quel niveau de tolérance : plus de 92%. La présence de deux images d'entraînement par individu semble faire toute la différence, favorisant l'apprentissage efficace de cette méthode ;
- Les HMM n'affichent cependant aucun gain supplémentaire lorsque les trois premières positions sont utilisées (*Top 3*) ;
- La méthode utilisant la DCT offre le second meilleur taux de reconnaissance avec presque 86%, c'est davantage que les EF et les EO. Il est donc intéressant de remarquer que malgré toutes les similitudes mathématiques entre ces méthodes, des performances relativement différentes sont obtenues ;
- Pour terminer, le *Top 2* procure aux *EigenObjects* un gain appréciable de plus de 6%. Ce comportement peu également être observé pour la FERET.

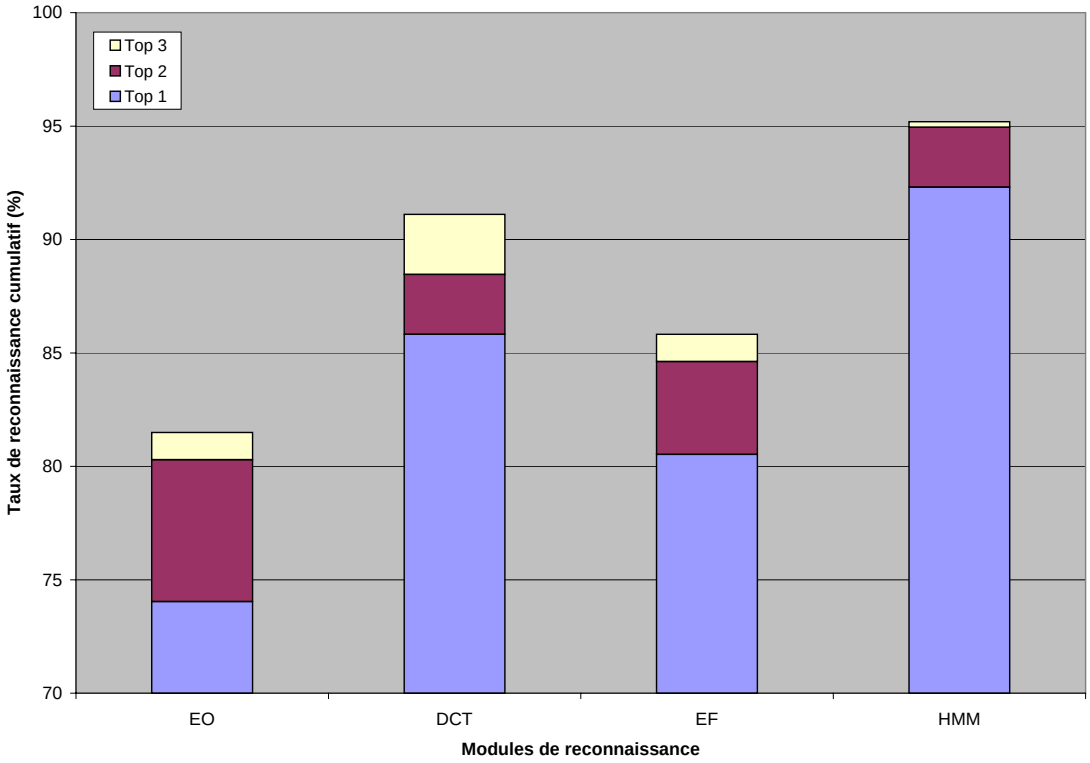



**Fig.** ***4.8:*** *Résultats expérimentaux sur la base d'images AR-face pour différentes méthodes de reconnaissance.*

**Discussion** Les résultats expérimentaux pour les modules individuels soulèvent plusieurs points intéressants demandant réflexion. Tout d'abord, la présence de 2 images d'apprentissage par personnes semble procurer un avantage incontestable à la méthode des HMM qui les utilisent pour modéliser chacune des classes présentes (c.-à-d. : chacun des individus). Ce traitement intrinsèque propre à cette technique n'est évidemment pas utilisé directement dans les autres méthodes.

Il serait cependant envisageable de former des classes moyennes pour chacun des individus tel que proposé dans la littérature pour les EF [62]. Cette modification risquerait d'améliorer les résultats des méthodes EF, DCT et EO.

Cela étant dit, il serait également intéressant de reproduire ces expérimentations sur des sections modifiées de la FERET contenant plusieurs images d'apprentissage par individu, et ainsi observer les différences de performance par rapport aux sections

prédéfinies.

Dans un autre ordre idée, les erreurs d'identifications effectuées par chacune des méthodes n'ont pas été comparées entres elles. Il y a fort à parier que les techniques de reconnaissance réalisent des classifications différentes et qu'elles varient selon des conditions particulières. L'expérimentation de ces méthodes dans un système multi-classifieur sur les mêmes banques d'images fournira donc davantage de pistes de solutions.

### 4.4.3 Impact des métriques utilisées

La prochaine série d'expérimentations porte sur les différentes métriques utilisables par certaines méthodes (c.-à-d. : EF, DCT et EO) lors de l'application de l'algorithme $K$-ppv. Ce dernier utilise en effet une métrique particulière pour déterminer l'ordre de proximité de la représentation test avec les différents prototypes d'apprentissage.

Certaines des métriques envisageables ont été présentées au chapitre précédent (3.3.1.5), citons notamment la distance $L_1$, $L_2$ et la différence d'angle entre deux vecteurs. La distance de Mahalanobis n'est pas envisagée car elle ne peut être appliquée à la méthode utilisant la DCT[7].

Il est à noter que la technique des HMM n'utilisent pas de métriques pour la classification des individus et ne sera donc pas visée par les expériences courantes. En effet, celle-ci génère directement des probabilités qui peuvent être triées à l'aide d'un algorithme de tri conventionnel.

**Discussion** La figure 4.9 illustre les différents taux de reconnaissance obtenus sur la banque d'images FERET pour les trois méthodes d'identification concernées. Voici donc certains points d'analyses :

- La distance $L_1$, ou *city-block*, procure les meilleurs taux de reconnaissance ;
- Dans le cas de la DCT, la distance $L_1$ est préférable à la $L_2$, contrairement à ce qui est utilisé par certains auteurs [19] ;

[7] La distance de Mahalanobis est basée sur l'utilisation des valeurs propres de la matrice de covariance correspondante.

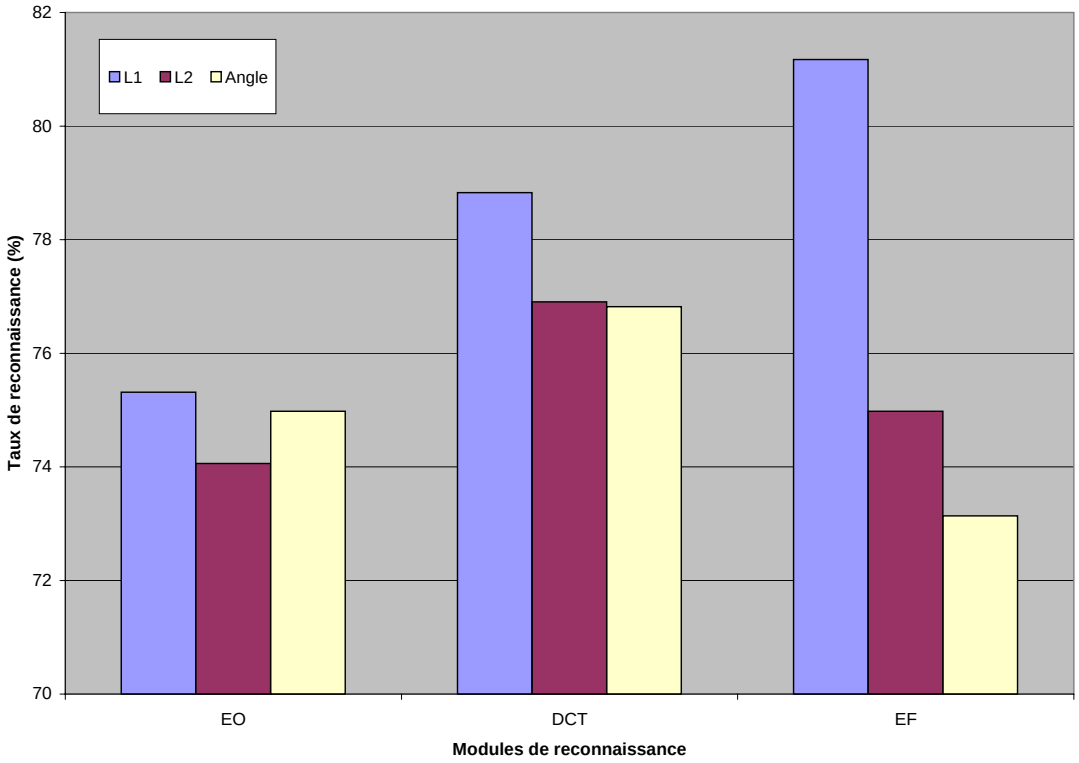



**Fig.** ***4.9:*** *Impact des métriques utilisées sur le taux de reconnaissance.*

– Les taux de reconnaissance de la distance $L_2$ et de la différence d'angle ne permettent pas d'établir clairement la métrique la plus performante. La distance euclidienne semble cependant être en moyenne légèrement plus appropriée ;

– Les résultats obtenus pour la méthode EF sont conformes à ceux publiés dans une étude [69] sur la sélection de vecteur propre et de métriques de distance pour les techniques *PCA*. En d'autres mots, le classement final des métriques (c.-à-d. : $L_1$, $L_2$ et angle) est identique à celui de l'étude.

Suite à ces conclusions, toutes les expérimentations ont été réalisées uniquement à l'aide de la distance $L_1$. Il serait cependant intéressant de vérifier l'impact de l'utilisation de différentes métriques dans un système multi-classifieur. Qui plus est, les erreurs de classifications commises en utilisant une certaine métrique ne sont peut-être pas identiques à celles réalisées par les autres.

| Méthodes | Apprentissage | Reconnaissance | Reconnaissance 1 ind. |
|---|---|---|---|
| **EF** | 96 | 122 | 0.25 |
| **EO** | 36 | 57 | 0.09 |
| **DCT** | 21 | 77 | 0.13 |
| **HMM** | 28 | 2 027 | 5.21 |

**Tab. *4.5:*** *Tableau comparatif des temps de traitement des algorithmes de reconnaissance. Ces valeurs sont exprimées en secondes et représentent les temps nécessaires pour réaliser les phases d'apprentissage et de reconnaissance sur la base de données AR-face qui contient 263 images d'apprentissage. La phase de reconnaissance regroupe les temps nécessaires pour l'identification des 476 images tests.*

### 4.4.4 Temps de traitement

Dans un contexte temps réel, le temps d'exécution des méthodes de reconnaissance occupe évidemment une place prépondérante dans le design global de l'application. En effet, afin de permettre une intervention rapide et pour traiter une masse phénoménale d'information[8], les algorithmes doivent être optimisés au maximum.

Ainsi, les temps de traitement des quatre modules de reconnaissance ont été mesurés pour les tâches d'apprentissage et d'identification. Le tableau 4.5 résume le fruit de ces expérimentations.

**Discussion** La banque AR-face a été utilisée pour évaluer le temps nécessaire aux différents modules d'identification pour compléter les phases d'apprentissage et de reconnaissance. La première tâche consiste à extraire la représentation des 263 images d'entraînement et à réaliser l'apprentissage adéquat. Ensuite, les 476 photos de la banque test sont traitées une à la fois pour déterminer le bon candidat. Voici quelques analyses sommaires :

- La méthode des *EigenFaces* nécessite la plus grande quantité de temps pour compléter la phase d'apprentissage. Pour ce qui est des autres techniques, elles requièrent de 21 à 36 secondes pour effectuer la même tâche ;
- L'étape de reconnaissance dresse par contre un bilan très différent. La technique des HMM utilise en effet plus de 2 000 secondes (c.-à-d. : plus de 33 minutes) pour

[8] Réaliser par exemple l'identification simultanée de plusieurs individus à partir d'une banque contenant des milliers de personnes.

identifier les 476 individus. ;

- L'identification d'un seul individu requiert par ailleurs un temps raisonnable de 90 à 250ms pour les méthodes EF, DCT et EO. Les HMM demandent cependant de 5 à 7 secondes pour réaliser la reconnaissance d'une seule personne ;
- Les résultats obtenus confirment également l'affirmation [19] soutenant l'avantage du temps de calcul de la DCT envers les EF.

Les temps d'exécution nécessaires aux différentes méthodes pour compléter les tâches d'apprentissage et de reconnaissance procurent de bons indices pour le design du système complet. En effet, les méthodes sont utilisables en temps réel excepté les HMM qui ne pourront identifier plusieurs individus à la seconde. Cette méthode devrait donc faire l'objet de toutes les optimisations possibles.

Une autre option envisageable serait d'utiliser un système distribué pour paralléliser les calculs. Cette solution doit cependant être utilisée en dernier recours car plusieurs autres tentatives d'accélération peuvent être réalisées auparavant.

Finalement, un système d'identification devant reconnaître des milliers de personnes aurait probablement à utiliser plusieurs unités de calculs pour conserver un temps de réponse acceptable. Cependant, un module efficace de suivi (c.-à-d. : *tracking*) de personnes pourrait être ajouté au système, ce qui éviterait de réaliser autant d'identifications par seconde.

### 4.4.5 Multi-classifieur

L'une des expériences les plus importantes est abordée dans la présente sous-section et concerne directement la performance du système multi-classifieur. Pour réaliser cette évaluation, les deux banques d'images sélectionnées sont une fois de plus utilisées indépendamment pour l'apprentissage et la vérification.

Pour faciliter les comparaisons avec les méthodes individuelles, les paramètres des modules de reconnaissance sont identiques à ceux utilisés pour l'évaluation des taux d'identification relatifs présentés précédemment à la sous-section 4.4.2.

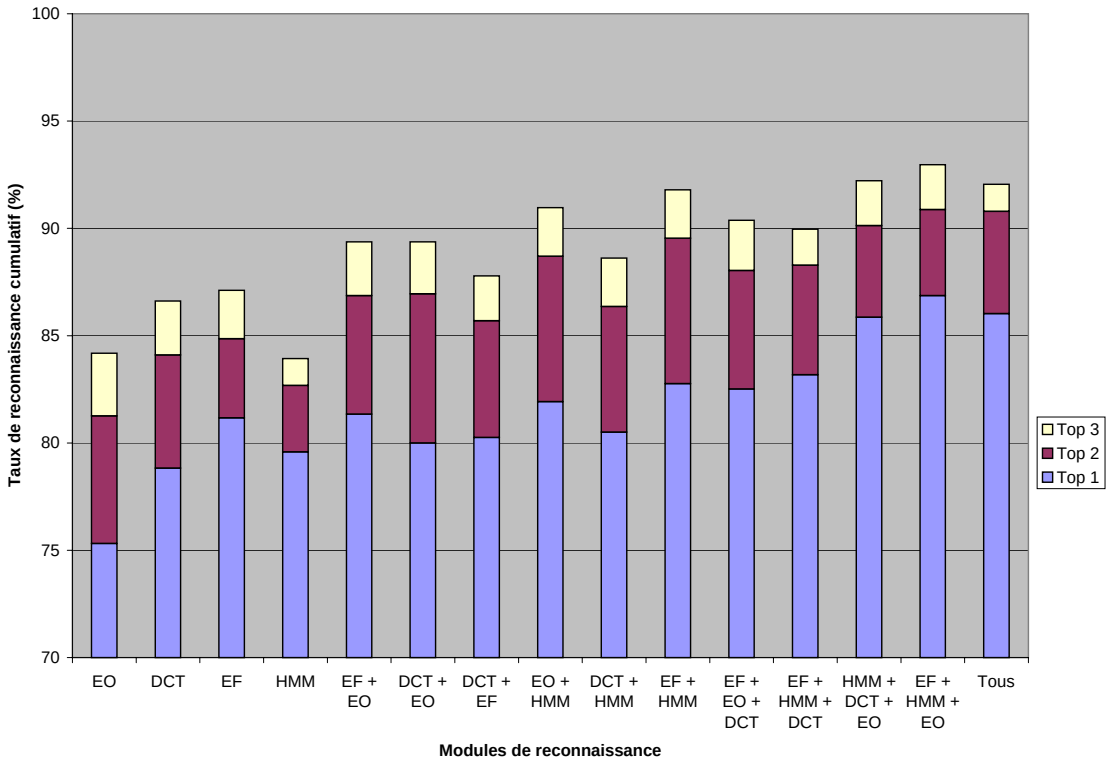



**Fig. *4.10:*** *Résultats expérimentaux sur la section FB (expressions) de la banque d'images FERET pour différents agencements multi-classifieurs.*

Pour ce qui est de la méthode de vote utilisée, le compte de Borda a été privilégié pour toutes les expérimentations. Ainsi, il serait intéressant de réaliser d'autres séries d'expériences portant sur l'impact du choix de la méthode de décision sur les taux de reconnaissance.

Les figures 4.10 et 4.11 illustrent les résultats expérimentaux obtenus. Pour chacune d'entre elles, les taux d'identification individuels sont affichés à la gauche du graphique et représentent les valeurs étalons. Les combinaisons possibles de 2 modules sont ensuite illustrées à la droite suivies de celles composées de 3 modules. La dernière colonne dénommée *Tous* représente le multi-classifieur complet contenant les 4 modules disponibles.

**FERET** Les résultats pour cette première banque sont représentés à la figure 4.10 et font partie intégrale d'un article publié à l'été 2003 [37]. Il est intéressant de remarquer la progression croissante vers la droite des taux de reconnaissance pour les différents agencements MC, ce qui démontre une augmentation proportionnelle de la performance par rapport au nombre de classifieurs utilisé.

L'analyse de ce graphique soulève plusieurs remarques intéressantes :

- Les HMM contribuent grandement à l'amélioration du taux de reconnaissance. En effet, les groupes possédant deux classifieurs (2-classifieurs) et qui contiennent un HMM performent mieux que les autres agencements du même niveau. Par exemple, la paire *EF+HMM* obtient un taux d'identification supérieur aux groupes *EF+EO* et *EF+DCT* ;
- La combinaison *EF+HMM* performe mieux que le groupe de trois éléments *EF+EO+DCT*. Ceci peut être expliqué d'une part par de fausses identifications influencées par un classifieur supplémentaire erroné ou d'autre part, à cause d'une complémentarité plus adéquate entre les HMM et les EF ;
- Le groupe *EF+DCT* obtient de très mauvais résultats pour sa catégorie ainsi que par rapport aux classifieurs individuels. Il est vrai que la combinaison *EO+DCT* n'offre pas une bien meilleure performance, mais elle engendre néanmoins une amélioration intéressante des méthodes EO et DCT individuelles. Bref, l'union des EF et des DCT n'offre aucun avantage et semble suggérer que ces classifieurs réalisent les mêmes erreurs ou qu'ils ne peuvent bénéficier des forces de l'autre ;
- Dans une lignée semblable, le groupe *EF+EO+DCT* offre la pire performance des 3-classifieurs (*Top-1* et *Top-2*) ;
- Les résultats suggèrent également que l'utilisation d'une méthode basée sur une ACP avec les HMM procurent de bons résultats ;
- La combinaison formée des quatre modules d'identification obtient des résultats légèrement inférieurs au groupe *EF+HMM+EO*, ce dernier procurant les meilleurs taux de reconnaissance ;
- Finalement, un agencement composé de méthodes locales et globales semblent fournir les meilleurs résultats (p. ex. : *EF+HMM+EO* et *HMM+DCT+EO*).

Les résultats obtenus pour cette première banque d'images sont assez concluants et démontrent bien l'avantage des systèmes multi-classifieurs.

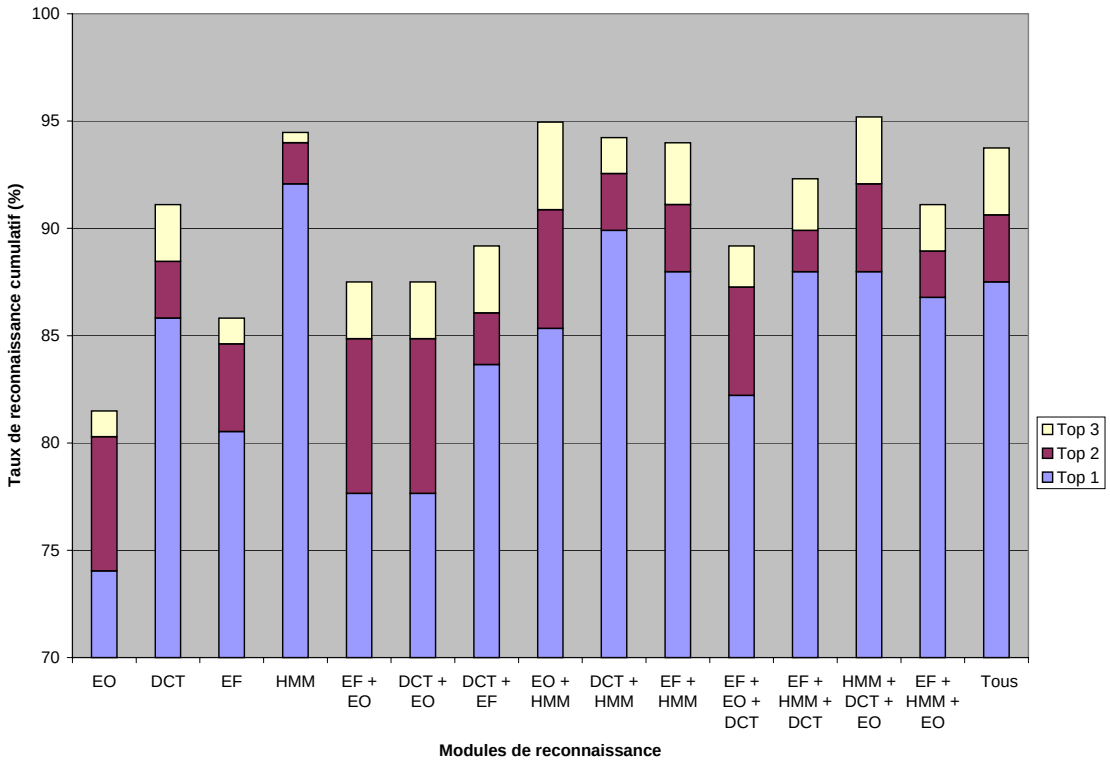



**Fig.** ***4.11:*** *Résultats expérimentaux sur la base d'images AR-face pour différents agencements multi-classifieurs.*

**AR-face** La deuxième banque d'images utilisée, la AR-face, dresse par contre un portrait très différent de celui généré par la FERET. Ces résultats sont illustrés à la figure 4.11.

Avant de poursuivre avec l'analyse et la comparaison des résultats, il est important de résumer les différences entre les deux banques d'images employées. Les sections sélectionnées de la FERET contiennent tout d'abord un plus grand nombre de personnes, c'est-à-dire 1196 contre 135 pour la AR-face. Ces informations sont critiques : des quantités de classes plus faibles diminuent habituellement les chances d'erreurs.

Qui plus est, un plus grand nombre d'images d'entraînement par individu favorisent généralement leur apprentissage. La AR-face contient deux images par individu dans la majorité des cas contrairement à la FERET qui n'en contient qu'une seule par personne.

L'analyse des résultats illustrés à la figure 4.11 permet la formulation des points suivants :

- Ce qui se démarque le plus des résultats, c'est sans aucun doute l'écrasante supériorité des HMM face à n'importe quelle combinaison de classifieurs. Les ensembles *EO+HMM* et *HMM+DCT+EO* parviennent cependant à de meilleurs taux de reconnaissance au troisième niveau (c.-à-d. : *Top-3*) ;
- Les *EigenObjects* font par contre très mauvaise figure. En effet, tous les agencements qui en contiennent baissent en performance. Par exemple, DCT performe mieux que *DCT+EO*, *DCT+HMM* mieux que *DCT+HMM+EO*, *DCT+EF* mieux que *DCT+EF+EO*, *etc.* Ce module nuit dans tous les cas au système multi-classifieur ; ce qui est un comportement différent de celui observé avec la FERET.

- Cette dernière affirmation peut être nuancée car au deuxième niveau (*Top-2*, la majorité des ensembles contenant des EO gagnent plus de 5% sur les taux de reconnaissance du premier niveau ;
- Le module utilisant la DCT procure des résultats largement supérieurs à ceux obtenus avec les EF.

**Discussion** Les résultats obtenus pour la banque d'images AR-face sont plutôt contradictoires à ceux produits par les expérimentations utilisant la FERET. En effet, ceux-ci ne permettent nullement de conclure sur l'utilité d'un système multi-classifieur, qui semble même très peu efficace dans le cas de la AR-face.

### 4.4.6 Détection automatique du visage : Impacts

La dernière expérience réalisée dans le cadre du projet concerne les effets de la détection automatique du visage sur les taux de reconnaissance des différents agencements multi-classifieurs.

Le protocole expérimental utilisé est légèrement différent de ce qui est employé dans les sous-sections précédentes. La seule étape modifiée concerne les coordonnées des caractéristiques du visage ayant été manuellement étiquetées. Cette fois-ci, le module de détection du visage présenté au chapitre 2 est utilisé pour accomplir cette tâche.

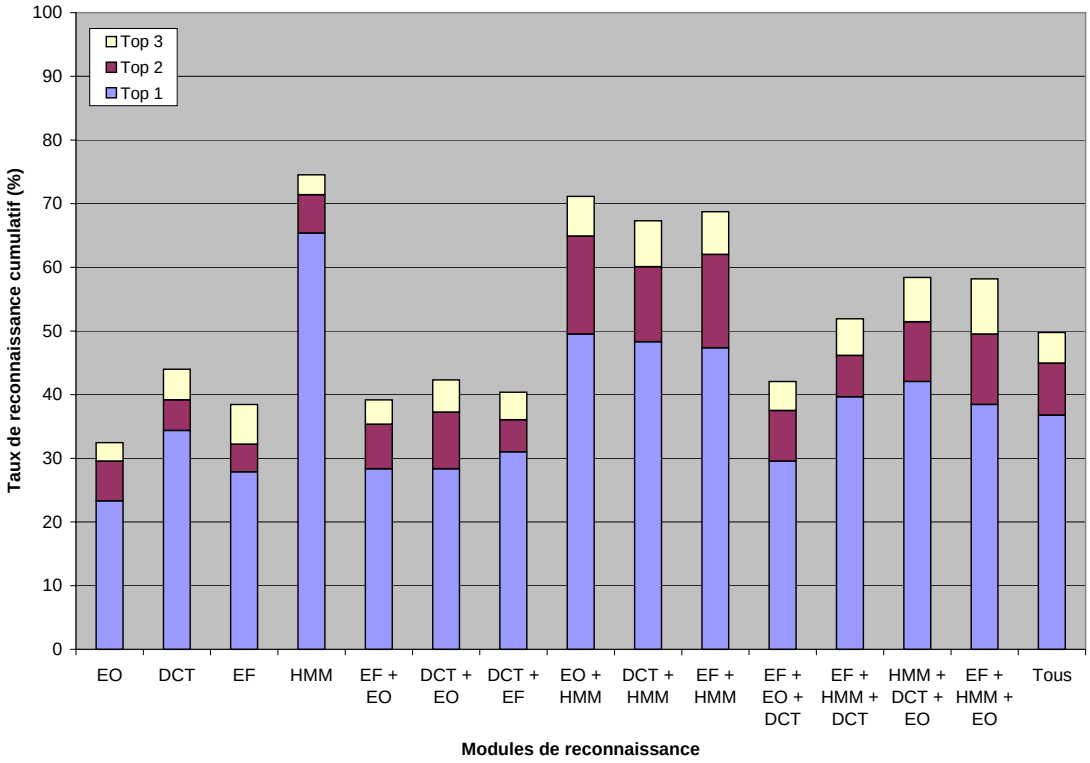



**Fig.** ***4.12:*** *Impact de la détection automatique du visage sur le taux de reconnaissance de différents agencements multi-classifieurs. La base d'images utilisée est la AR-face.*

Les coordonnées produites sont employées directement sans aucune modification pour normaliser les images d'apprentissage et de vérification. Il n'y a donc aucun rejet ni filtrage des données fortement erronées.

Étant donné que la technique de détection du visage nécessite la présence de couleurs pour segmenter les pixels représentant la peau, la banque FERET n'a pu être utilisée. Les résultats illustrés à la figure 4.12 ont donc été obtenus uniquement à partir de la banque d'images AR-face.

L'analyse de ces résultats suggère avant tout que les modules de reconnaissance (ainsi que n'importe quelle combinaison) subissent une perte marquée de leur taux de reconnaissance respectif. Ainsi, une chute de performance de près de 50% est observée.

Les HMM représentent la seule technique d'identification offrant malgré tout une

performance acceptable, soit tout de même une baisse de plus de 27% de son taux de reconnaissance. Cette situation plus favorable s'explique en partie par l'étape itérative de segmentation du visage propre aux HMM. C'est ainsi que l'algorithme positionne et raffine chacune des régions sur les visages, minimisant alors les désagréments causés par les erreurs de translation, d'échelle et de rotation.

Pour ce qui est des agencements de classifieurs, ils n'aident nullement à amoindrir l'effet négatif de la détection automatique. Cette situation serait différente dans le cas où les modules de reconnaissance commettraient des erreurs dans des cas différents (p. ex. : HMM robuste aux translations, EF robuste aux rotations, *etc.*).

À ce stade-ci, il est intéressant de faire le lien entre les résultats de reconnaissance, ceux obtenus lors de l'étude sur la robustesse des EF ainsi que ceux provenant de la précision de la détection du visage (sous-section 2.4). L'étude sur la robustesse des EF suggère des limites de translation de 5% sur la largeur et 3% sur la hauteur. Or, les résultats obtenus au chapitre 2 révèlent une erreur moyenne de 13 pixels ; erreur qui peut être horizontale, verticale ou diagonale. Cette valeur représente dans le cas d'un visage de dimensions $300 \times 400$, respectivement 4.3% et 3.25% pour la largeur et la hauteur. Les figures 4.5a et b démontrent bien que ces valeurs sont situées tout précisément aux points de cassure des courbes.

Ces informations suggèrent donc que le module de détection produit en moyenne des visages normalisés au seuil de la reconnaissance des EF. Qui plus est, les affirmations précédentes ne tiennent pas compte des erreurs de rotation et de mise à l'échelle. Les modules de reconnaissance auraient alors besoin d'une détection du visage plus précise.

Finalement, il est important de noter que la banque AR-face contient certaines images nuisant clairement au module de détection. En effet, toutes les personnes possédant des cheveux blonds détachés ou des chemises roses biaiseront la prénormalisation du visage en rotation et en échelle, ce qui occasionnera des visages normalisés complètement erronés. Une grande majorité des mauvaises classifications s'explique par ces cas particuliers non supportés par l'algorithme de détection du visage.

## 4.5 Conclusion

Trois banques d'images ont été présentées au cours de ce dernier chapitre, soient la FERET, la AR-face et la LVSN. Alors que la LVSN n'est utilisée que pour l'application en temps réel du système, les deux autres ont été employées pour réaliser plusieurs expérimentations visant à vérifier des particularités du système.

Tout d'abord, une étude portant sur la méthode des *EigenFaces* a démontré la robustesse de cette dernière à certaines transformations pouvant survenir lors de la détection et de la normalisation du visage. Cette technique peut même tolérer la présence de plusieurs effets négatifs simultanément, en autant que des limites spécifiques ne sont pas transgressées.

Après avoir présenté les performances individuelles des modules de reconnaissance, l'impact des métriques utilisées lors de l'identification a été abordé. En effet, le choix de cette métrique est crucial et agit directement sur le taux de reconnaissance des méthodes *EigenFaces*, *EigenObjects* et *DCT*. Parmi les distances testées, la $L_1$ a permis l'obtention des meilleurs résultats.

Les différents temps d'exécution des techniques retenues ont également été présentés. Alors que la plupart d'entres elles performent relativement rapidement en phase d'identification, les HMM démontrent un sérieux problème quant au laps de temps nécessaire pour compléter cette opération. Cet algorithme devrait donc être optimisé davantage pour permettre son utilisation en temps réel sur une banque d'images volumineuse.

L'utilisation de multi-classifieur a également été vérifiée sur la FERET et la AR-face. Chacune d'entre elles généra par contre un constat fort différent. Alors que l'emploi de MC sur la FERET améliore de près de 6% la meilleure performance individuelle, les HMM dominent clairement sur les MC dans le cas de la AR-face. Le fait de posséder deux images d'apprentissage par individu semble favoriser les HMM face aux autres méthodes qui ne sont pas adaptées à cette situation dans le cadre du projet.

Afin d'étendre les conclusions obtenues, des expérimentations supplémentaires pourraient être réalisées sur la banque d'images FERET. En effet, pour valider l'efficacité des MC et la performance des HMM, de nouvelles sections devraient être créées pour

obtenir une banque d'apprentissage contenant plusieurs images par individu. Cette nouvelle condition jumelée au nombre élevé de classes à discerner de la FERET permettrait d'établir des analyses plus robustes sur l'utilité des MC.

Qui plus est, des métriques adaptées devraient également être utilisées pour les méthodes DCT, EO et EF afin de supporter un nombre de prototypes supérieur à un. Cette amélioration permettrait sans doute à ces techniques de mieux performer face aux HMM.

Il serait également intéressant de déterminer le gain théorique maximal possible pour un agencement MC. Pour ce faire, le nombre total de détections correctes des méthodes de reconnaissance individuelles indiquerait le plafond atteignable lorsque la fonction de décision effectue toujours le bon choix. Il serait ainsi envisageable d'effectuer de meilleures sélections d'algorithmes en maximisant leur complémentarité.

Finalement, la détection automatique du visage influence fortement les taux de reconnaissance des différentes méthodes et agencements multi-classifieurs qui voient leur performance subir une perte de plus de 50%. Les HMM sont les seuls épargnés avec une chute de 27% de leur taux de reconnaissance. Le module de détection nécessiterait donc quelques ajustements, notamment pour les cas spéciaux qui nuisent et biaisent l'identification.

# Conclusion

Malgré tous les travaux réalisés au cours des dernières années, la reconnaissance d'individus demeure un problème complexe et non parfaitement résolu. Plusieurs sous-problèmes incombent à cette tâche d'identification et chacun d'eux n'est pas trivial. Il y a également de nombreuses conditions réelles influençant la performance d'un système ; conditions qui ne sont pas toujours tenues en compte en laboratoire.

Cela étant dit, le système complet de détection et d'identification proposé dans ce mémoire n'a pas la prétention d'être le meilleur de tous ou de résoudre toutes les situations problématiques. Il représente néanmoins une solution efficace respectant les contraintes initiales et accomplissant les différentes tâches lui étant demandé.

De nombreuses techniques ont été présentées tout au long de ce mémoire, certaines furent adaptées ou tout simplement rejetées. Les prochains paragraphes résument brièvement les méthodes retenues ainsi que les principales conclusions qui ressortent de ce travail.

**Résumé des modules spécifiques** Le système de reconnaissance développé contient quatre phases principales accomplissant les tâches d'acquisition des images, de détection du mouvement, de détection du visage et, finalement, d'identification des personnes. Ces différentes étapes sont accomplies par des modules spécifiques.

L'acquisition est tout d'abord réalisée à l'aide d'une caméra *web* abordable capable de générer des images couleurs. Utilisant le port USB, ce type de capteur ne nécessite aucun équipement spécialisé comme par exemple une carte d'acquisition (*frame grabber*).

La détection du mouvement est réalisée ensuite par un module utilisant la soustraction de l'arrière-plan par modélisation statistique. Ce dernier nécessite un champ de vue fixe afin de bien modéliser l'arrière-plan et utilise directement les images brutes fournies par la caméra. La technique est relativement efficace et comble largement les besoins nécessaires à l'identification des zones de mouvement dans l'image.

Les images contenant les pixels en mouvement sont ensuite acheminées au module de détection du visage. Certains prétraitements sont effectués, dont notamment la détection des pixels représentant la peau dans l'espace des couleurs HSV. Cette étape importante vise à réduire davantage l'espace de recherche des visages et à orienter directement le processus de détection.

La détection du visage est réalisée par une méthode hybride alliant deux étapes d'appariement de gabarits. Cette technique tolère de légères rotations ainsi que certains cas spéciaux comme par exemple des yeux fermés.

Les coordonnées du visage générées par ce module sont utilisées directement pour la normalisation de l'image du visage. Ainsi, ce dernier possédera des dimensions et des conditions (c.-à-d. : yeux alignés parfaitement) similaires à celles de l'apprentissage.

Selon certaines expérimentations, l'erreur moyenne de la localisation des yeux se chiffre à environ 13 pixels sur des photos contenant des visages de dimensions approximatives de $300 \times 400$.

Le dernier module du système réalise la tâche d'identification proprement dite. Ainsi, une architecture multi-classifieur a été développée permettant l'utilisation de n'importe quelle technique de reconnaissance. Cette architecture logicielle peut également subir des modifications dynamiques, utiles pour l'expérimentation en série de plusieurs paramètres.

Quatre méthodes ont été retenues pour la reconnaissance des individus, soient les

*EigenFaces*, les *EigenObjects*, la DCT ainsi que les HMM.

Les résultats obtenus soulèvent plusieurs remarques intéressantes dont voici les plus importantes :

- Lorsqu'au moins 2 images d'apprentissage sont disponibles, les HMM obtiennent des résultats supérieurs aux autres méthodes ;
- Les résultats obtenus ne peuvent clairement départager les méthodes EF et DCT quant à leurs performances relatives ;
- L'utilisation de multi-classifieurs sur la base d'images FERET procure un gain de près de 6% sur le meilleur classifieur individuel. La section utilisée de la FERET contient 1 196 personnes à identifier ;
- La conclusion précédente ne s'applique pas à la base d'images AR-face où les HMM dominent sur tous les agencements MC ;
- La détection automatique des visages influencent énormément la performance du module d'identification. Une chute de près de 50% du taux de reconnaissance est observée pour l'ensemble des méthodes excepté les HMM avec une baisse de près de 27%. Cette conclusion suggère la nécessité d'un module de détection du visage plus précis.

**Atteintes des objectifs et respect des contraintes** Le système proposé dans ce mémoire satisfait toutes les contraintes établies au début du projet. Il réalise en effet, en temps réel, toutes les opérations nécessaires à l'identification d'une personne.

Le montage proposé, qui est composé d'un ordinateur standard et d'une caméra *web*, satisfait également les contraintes de coût associées au projet.

**Travaux futurs** La conception de ce projet a permis l'apprentissage d'une grande quantité d'informations. Qui plus est, plusieurs améliorations potentielles ont été observées à l'usage et pourraient faire l'objet de travaux futurs.

Tout d'abord, de nombreuses expérimentations supplémentaires pourraient être réalisées, notamment à propos des multi-classifieurs. Les fonctions de votes, les métri-

ques du $K$-ppv, la complémentarité des méthodes et la sélection dynamique de classifieur représentent des sujets intéressants à investiguer. Des techniques d'intelligence artificielle comme les réseaux de neurones pourraient également être expérimentées comme fonction de décision.

Des expérimentations plus poussées devraient également être réalisées sur la banque d'images FERET. Il serait intéressant d'étendre les conclusions MC de ce mémoire en créant de nouvelles sections de vérification possédant plusieurs images d'apprentissage pour chacune des 1 196 personnes de la banque d'images.

Les HMM auraient sans aucun doute besoin d'un travail d'optimisation afin de faciliter leur intégration dans un système devant identifier des milliers de personnes. Pour l'instant, son utilisation en temps réel est possible tant qu'une banque restreinte d'individus est employée.

Pour ce qui est de la détection du visage, certaines améliorations pourraient être apportées au processus de raffinement de la position des yeux. Des modifications importantes devraient cependant être réalisées pour la détection de visage de profil.

Finalement, ce système d'identification pourrait facilement être jumelé à d'autres projets du LVSN. Tout d'abord, la détection du visage profiterait énormément d'une segmentation en parties du corps humain. La tête serait donc utilisée directement comme zone de recherche, ce qui éviterait les normalisations erronées à cause des vêtements ou des cheveux.

De nouveaux modules de reconnaissance pourraient également être intégrés au système multi-classifieurs. Parmi ceux-ci, il y a notamment l'identification basée sur la couleur ainsi que les mesures morphologiques du corps humain provenant d'une acquisition tridimensionnelle.

**Conclusion** La biométrie, dont fait partie la reconnaissance du visage, est sans contredit un domaine d'avenir. De plus en plus de systèmes verront probablement le jour au cours des prochaines décennies afin de réaliser une surveillance accrue. Plusieurs aéroports songent déjà à installer des systèmes d'identification par le visage ou l'iris. C'est notamment le cas de l'aéroport Pearson de Toronto.

Mais alors que certaines personnes doutent de l'efficacité d'une identification par le visage (c.-à-d. : déguisement pour déjouer le système), d'autres protestent et affirment qu'il y a atteinte à la vie privée.

Outre ces questions d'éthique, il demeure néanmoins que ces technologies sont, d'un point de vue scientifique, très passionnantes et que beaucoup de recherches restent à faire. Mais heureusement, nous sommes encore loin des tatouages de codes zébrés pour fins d'identification !

# Bibliographie


[1] Bernard Achermann et H. Bunke : Classifying range images of human faces with hausdorff distance. Dans *International Conference on Pattern Recognition (ICPR)*, pages 813–817, 2000.

[2] Chiraz BenAbdelkader, Ross Cutler et Larry Davis : Motion-based recognition of people in EigenGait space. Dans *5th International Conference on Automatic Face and Gesture Recognition (FG)*, pages 254–259, May 2002.

[3] Robert Bergevin : Vision numérique : aspects cognitifs (notes de cours GEL-64793). Université Laval, Automne 2000.

[4] R. Brunelli et T. Poggio : Face recognition : features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15(10):1042–1052, October 1993.

[5] Roberto Brunelli et Tomaso Poggio : Face recognition through geometrical features. Dans *European Conference on Computer Vision (ECCV)*, pages 792–800, 1992.

[6] Budget fédéral 2002. http://www.radio-canada.ca/nouvelles/budget/federal_2002/bref.html.

[7] Budget fédéral 2003. http://www.fin.gc.ca.

[8] N. Cristianini et J. Shawe-Taylor : *An Introduction to Support Vector Machines*. Cambridge University Press, 2002.

[9] Rita Cucchiara, Costantino Grana, Massimo Piccardi, Andrea Pratti et Stefano Sirotti : Improving shadow suppression in moving object detection with hsv color information. Dans *Intelligent Transportation Systems*, pages 334– 339. IEEE, 2001.

[10] M. Kunert D. M. Gavrila et U. Lages : A multi-sensor approach for the protection of vulnerable traffic participants - the protector project. Dans *Proc. of the IEEE Instrumentation and Measurement Technology Conference*, volume 3, pages 2044–2048, 2001.

[11] Jean-Charles de Borda : Mémoire sur les Élections au scrutin. Histoire de l'Académie Royale des Sciences, Paris, 1781.

[12] Ahmed Elgammal, David Harwood et Larry Davis : Non-parametric model for background subtraction. *European Conference on Computer Vision*, pages 751–767, 2000.

[13] Raphaël Feraud, Olivier Bernier, Jean-Emmanuel Viallet et Michel Collobert : A fast and accurate face detector based on neural networks. Dans *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 23(1), pages 42–53, 2001.

[14] A. Franco, A. Lumini et D. Maio : Eigenspace merging for model updating. Dans *16th International Conference on Pattern Recognition (ICPR)*, volume 2, pages 156–159, Québec, Canada, August 2002.

[15] E. Gamma, R. Helm, R. Johnson et J. Vlissides : *Design Patterns : Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, MA, USA, 1994.

[16] G. Giacinto et F. Roli : Dynamic classifier selection based on multiple classifier behaviour. *Pattern Recognition*, 34(9):179–181, 2001.

[17] Rafael C. Gonzalez et Richard E. Woods : *Digital Image Processing (2nd Ed.)*, chapitre 10, Morphological Image Processing, pages 519–566. Prentice Hall, 2002.

[18] Rafael C. Gonzalez et Richard E. Woods : *Digital Image Processing (2nd Ed.)*, chapitre 3, Image Enhancement in the Spatial Domain, pages 75–146. Prentice Hall, 2002.

[19] Ziad M. Hafed et Martin D. Levine : Face recognition using discrete cosine transform. *International Journal of Computer Vision*, 43(3):167–188, July - August 2001.

[20] S. Haykin : *Neural Networks : A Comprehensive Foundation*, chapitre 6. IEEE Press, 1994.

[21] Erik Hjelmås et Boon Kee Low : Face detection : A survey. *Computer Vision and Image Understanding*, 83(3):236–274, September 2001.

[22] Tin Kam Ho, Jonathan J. Hull et Sargur N. Srihari : On multiple classifier systems for pattern recognition. Dans *11th International Conference on Pattern Recognition*, volume 2, pages 84–87, 1992.

[23] B. K. P. Horn et B. G. Schunck : Determining optical flow. Dans *Artificial Inteligence*, volume 17, pages 185–203, 1981.

[24] T. Horprasert, D. Harwood et L. Davis : A robust background subtraction and shadow detection. *In Proceedings of the ACCV*, 2000.

[25] Rein-Lien Hsu, Mohamed Abdel-Mottaleb et Anil K. Jain : Face detection in color images. *IEEE Trans. PAMI*, 24(5):696–706, 2002.

[26] Daniel P. Huttenlocher et William J. Rucklidge : A multi-resolution technique for comparing images using the hausdorff distance. Rapport technique CUCS TR 92-1321, Department of Computer Science, Cornell University, 1992.

[27] Imagis technologies inc. http://www.imagistechnologies.com.

[28] OpenCV : Open source computer vision library. http://www.intel.com/research/mrl/research/opencv/.

[29] H. Ishii, M. Fukumi et N. Akamatsu : Face detection based on skin color information in visual scenes by neural networks. Dans *IEEE International Conference on Systems, Man, and Cybernetics (SMC '99)*, volume 5, pages 350–355, 1999.

[30] Yuri A. Ivanov, Aaron F. Bobick et John Liu : Fast lighting independent background subtraction. *International Journal of Computer Vision*, 37(2):199–207, 2000.

[31] Oliver Jesorsky, Klaus J. Kirchberg et Robert W. Frischholz : Robust face detection using the hausdorff distance. Dans Josef Bigun et Fabrizio Smeraldi, éditeurs : *Audio- and Video-Based Person Authentication - AVBPA 2001*, volume 2091 de *Lecture Notes in Computer Science*, pages 90–95, Halmstad, Sweden, 2001. Springer.

[32] T. Kanade, A. Yoshida, K. Oda, H. Kano et M. Tanaka : A stereo machine for video-rate dense depth mapping and its new applications. Dans *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 196–202, 1996.

[33] M. Kirby et L. Sirovich : Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(1):103–108, 1990.

[34] Josef Kittler, Mohammad Hatef, Robert P.W. Duin et Jiri Matas : On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-20(3):226– 239, 1998.

[35] Shyh-Shiaw Kuo et Oscar E.Agazzi : Keyword spotting in poorly printed documents using pseudo 2-d hidden markov models. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 16(8):842–848, August 1994.

[36] Alexandre Lemieux et Marc Parizeau : Experiments on Eigenfaces Robustness. Dans *16th International Conference on Pattern Recognition (ICPR)*, volume 1, pages 421–424, Québec, Canada, August 2002.

[37] Alexandre Lemieux et Marc Parizeau : Flexible multi-classifier architecture for face recognition systems. *Vision Interface*, 2003.

[38] B.D. Lucas et T. Kanade : An iterative image registration technique with an application to stereo vision. Dans *IJCAI81*, pages 674–679, 1981.

[39] D. Maio et D. Maltoni : Real-time face location on gray scale static images. *Pattern Recognition*, 33(9):1525–1539, September 2000.

[40] Sebastien Marcel et Samy Bengio : Improving face verification using skin color information. Dans *16th International Conference on Pattern Recognition (ICPR)*, pages 378–381, Québec, Canada, August 2002.

[41] Aleix M. Martinez et R. Benavente : The AR-face database. Rapport technique, CVC Technical Report #24, June, 1998.

[42] Alan M. McIvor : Background subtraction techniques. *IVCNZ*, 2000.

[43] K. Messer, J. Matas, J. Kittler et K. Jonsson : Xm2vtsdb : The extended m2vts database. *Audio- and Video-based Biometric Person Authentication (AVBPA)*, pages 72–77, Mars 1999.

[44] Ivana Mikiæ, Pamela C. Cosman, Greg T. Kogut et Mohan M. Trivedi : Moving shadow and object detection in traffic scenes. Dans *International Conference on Pattern Recognition (ICPR)*, pages 321–324, 2000.

[45] B. Moghaddam et A. Pentland : Probabilistic visual learning for object representation. Dans S. Nayar et T. Poggio, éditeurs : *Early Visual Learning*, chapitre 5, pages 99–130. Oxford University Press, 1996.

[46] Ara V. Nefian et Monson H. Hayes III : An embedded hmm based approach for face detection and recognition. Dans *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume VI, pages 3553–3556, March 1999.

[47] A. Pentland, B. Moghaddam et T. Starner : View-based and modular eigenspaces for face recognition. Dans *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, WA, June 1994.

[48] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi et Patrick J. Rauss : The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[49] Son Lam Phung, D. Chai et A. Bouzerdoum : A universal and robust human skin color model using neural networks. Dans *International Joint Conference on Neural Networks (IJCNN '01)*, volume 4, pages 2844–2849, 1999.

[50] K. Rao et P. Yip : *Discrete Cosine Transform : Algorithms, Advantages, Applications*. Academic Press, 1990.

[51] H. Rowley, S. Baluja et T. Kanade : Rotation invariant neural network-based face detection. Dans *Proceedings of Computer Vision and Pattern Recognition*, 1998.

[52] Henry A. Rowley, Shumeet Baluja et Takeo Kanade : Human face detection in visual scenes. Dans David S. Touretzky, Michael C. Mozer et Michael E. Hasselmo, éditeurs : *Advances in Neural Information Processing Systems*, volume 8, pages 875–881. The MIT Press, 1996.

[53] Ferdinando Samaria : *Face Recognition Using Hidden Markov Models*. Thèse de doctorat, Engineering Department, Cambridge University, Trumpington Street, Cambridge CB2 1PZ, UK, October 1994.

[54] G. Shakhnarovich, L. Lee et T. Darrell : Integrated face and gait recognition from multiple views. Dans *Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 439–446, 2001.

[55] Karin Sobottka et Ioannis Pitas : Extraction of facial regions and features using color and shape information. *ICIP*, August 1996.

[56] Karin Sobottka et Ioannis Pitas : Segmentation and tracking of faces in color images. Dans *Proceedings of the second International Conference on Automatic Face and Gesture Recognition*, pages 236–241, 1996.

[57] Diego A. Socolinsky et Andrea Selinger : A comparative analysis of face recognition performance with visible and thermal infrared imagery. Dans *16th International Conference on Pattern Recognition (ICPR)*, volume 4, pages 217–222, August 2002.

[58] S. Suzuki et K. Abe : Topological structural analysis of digital binary images by border following. *CVGIP*, 30(1):32–46, 1985.

[59] M. R. Teague : Image analysis via the general theory of moments. *Journal of the Optical Society of America*, 70(8):920–930, 1980.

[60] Jean-Christophe Terrillon et Shigeru Akamatsu : Comparative performance of different chrominance spaces for color segmentation and detection of human faces in complex scene images. *VI*, 1999.

[61] Kentaro Toyama, John Krumm, Barry Brumitt et Brian Meyers : Wallflower : Principles and practice of background maintenance. Dans *ICCV (1)*, pages 255–261, 1999.

[62] Matthew Turk et Alex Pentland : Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[63] Rapport annuel 2000-2001 de l'Université Laval (version intégrale). http://www.ulaval.ca/sg/reg/Rapports.officiels.

[64] Merijn van Erp et Lambert Schomaker : Variants of the Borda count method for combining ranked classifier hypotheses. Dans *Seventh International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, pages 443–452, September 2000.

[65] Merijn van Erp, Louis Vuurpijl et Lambert Schomaker : An overview and comparison of voting methods for pattern recognition. Dans *Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, pages 195–200, August 2002.

[66] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell et Alex Pentland : Pfinder : Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[67] Lei Xu et Erkki Oja : Randomized hough transform (RHT) : Basic mechanisms, algorithms, and computational complexities. Dans *CVGIP : Image Understanding*, volume 57(2), pages 131–154, 1993.

[68] Lei Xu, Erkki Oja et Pekka Kultanen : A new curve detection method : Randomized hough transform (rht). *Pattern Recognition Letters*, 11:331–338, 1990.

[69] W. Yambor, B. Draper et R. Beveridge : Analyzing PCA-based Face Recognition Algorithms : Eigenvector Selection and Distance Measures. Dans H. Christensen et J. Phillips, éditeurs : *Empirical Evaluation Methods in Computer Vision*. World Scientific Press, Singapore, 2002.

[70] M. Yang et N. Ahuja : Gaussian mixture model for human skin color and its application in image and video databases. Dans *Conf. on Storage and Retrieval for Image and Video Databases (SPIE 99)*, volume 3656, pages 458–466, January 1999.

[71] Ming-Hsuan YANG et Narendra AHUJA : Detecting human faces in color images. Dans *International Conference on Image Processing (ICIP)*, volume 1, pages 127–130, 1998.

[72] Ming-Hsuan YANG, David J. KRIEGMAN et Narendra AHUJA : Detecting faces in images : A survey. Dans *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 24(1), pages 34–58, 2002.

[73] Qiang ZHU et Jiashi CHEN : A new approach for rotated face detection. Dans *Proceedings of the ninth ACM international conference on Multimedia*, pages 537–539, 2001.