Université de Liège
Faculté des Sciences Appliquées
Département d'Électricité, Électronique et Informatique
Institut Montefiore

Détection de personnes et de visages dans une séquence vidéo

Travail de fin d'études présenté en vue de l'obtention du grade d'ingénieur civil électricien (orientation électronique).

Olivier Barnich

Promoteur : Prof. Marc Van Droogenbroeck

Année académique 2003-2004

Remerciements

Je tiens à remercier Monsieur le Professeur Marc Van Droogenbroeck qui m'a permis d'aborder ce sujet. Ses conseil avisés m'ont toujours été d'une grande utilité.

Je remercie également mes parents et tout particulièrement ma mère qui a eu le courage de relire le présent document *plusieurs fois*.

Merci à Aurélie et Adrien pour l'arrière-plan.

Mes remerciements vont également à (sans ordre particulier) : Julien, Renaud Dardenne, Cédric Demoulin et Olivier.

Table des matières

1 Introduction **1**
- 1.1 Généralités . . . 1
- 1.2 Objectifs . . . 2
- 1.3 Aperçu . . . 2

2 Étude bibliographique **3**
- 2.1 Techniques basées sur la connaissance . . . 4
- 2.2 Techniques basées sur la recherche de traits invariants . . . 7
  - 2.2.1 Détection des traits du visage . . . 7
  - 2.2.2 Détection de la texture du visage . . . 10
  - 2.2.3 Détection de la couleur de la peau . . . 10
  - 2.2.4 Recherche de caractéristiques multiples . . . 12
- 2.3 Techniques basées sur la correspondance avec des modèles . . . 18
  - 2.3.1 Utilisation de modèles prédéfinis . . . 18
  - 2.3.2 Utilisation de modèles déformables . . . 20
- 2.4 Techniques utilisant des modèles obtenus par apprentissage . . . 22
  - 2.4.1 Choix d'une règle de classification . . . 23
  - 2.4.2 Recherche des pixels correspondant à la couleur de la peau en utilisant un classificateur . . . 24
  - 2.4.3 L'analyse en composantes principales . . . 25
  - 2.4.4 Les machines à vecteurs de support . . . 32
  - 2.4.5 Les réseaux de neurones . . . 38
  - 2.4.6 Autres approches statistiques . . . 42

3 Technique de détection proposée et implémentée **47**
- 3.1 Architecture du détecteur . . . 47
- 3.2 Sélection de zones d'intérêt . . . 49
  - 3.2.1 Génération et mise à jour de l'arrière-plan . . . 51
  - 3.2.2 Détection des pixels de la couleur de la peau . . . 53
- 3.3 Détection de visages vus de face . . . 54
  - 3.3.1 Regroupement des pixels d'intérêt en surfaces connexes . . . 55
  - 3.3.2 Sélection des surfaces répondant à une série de critères simples . . . 56
  - 3.3.3 Sélection des surfaces elliptiques d'orientations correctes . . . 57
  - 3.3.4 Recherche de la zone yeux-sourcils . . . 57

3.4 Suivi des visages déjà détectés une première fois 59
3.4.1 Définition du voisinage du visage suivi 61
3.4.2 Détection de zones de peau elliptiques dans le voisinage du visage suivi 61
3.4.3 Identification de la zone correspondant effectivement au visage suivi 62
3.4.4 Suivi par histogrammes de couleurs 63

4 Performances obtenues et améliorations possibles 66
4.1 Vitesse d'exécution 66
4.2 Qualité de la détection 67
4.3 Améliorations possibles 71
4.3.1 Une meilleure détection de la couleur de la peau 71
4.3.2 Une meilleure détection des traits du visage 72
4.3.3 Utilisation d'un classificateur 72

5 Conclusions 74

A Le classificateur idéal de Bayes 76

B Les machines à vecteurs de support : description théorique 77

C Les réseaux de neurones artificiels : description théorique 79

Chapitre 1

Introduction

1.1 Généralités

Depuis quelques années, les ordinateurs ont pris une place très importante dans la vie d'un grand nombre de personnes.

Afin de rendre leur utilisation toujours plus facile, naturelle et efficace, de plus en plus de techniques permettant une interaction homme-ordinateur plus *humaine* ont été et sont actuellement développées.

La baisse des prix permanente des systèmes d'acquisition d'images et la puissance sans cesse croissante des ordinateurs ont rendu les systèmes de vision par ordinateur accessibles pour une grande partie des utilisateurs. Il est ainsi devenu raisonnable d'envisager l'utilisation d'une caméra. Une partie de ces techniques y font donc appel.

Ces dernières font l'hypothèse que la présence, l'identité et les intentions de l'utilisateur peuvent être déduites à partir des images de cette caméra et que l'ordinateur peut alors réagir en fonction de celles-ci.

L'extraction de la plupart de ces informations nécessite au préalable la connaissance des positions des visages présents dans l'image. En effet, la présence et la position d'une personne peuvent être déduites de celles d'un visage. De plus, les systèmes de reconnaissance d'expressions faciales et d'identification de personnes font la plupart du temps l'hypothèse que les positions des différents visages contenus dans l'image sont connues.

Localiser des visages dans une image ou dans une séquence vidéo est une tâche loin d'être facile. La variabilité de l'apparence de ceux-ci en termes de taille, d'orientation, de position ou encore d'expression faciale fait de leur détection un véritable défi, surtout si on souhaite un fonctionnement correct du système quelles que soient les conditions d'illumination. La nécessité de pouvoir détecter des visages partiellement occultés peut rendre cette tâche encore plus compliquée.

Le développement de systèmes d'interaction entre humains et ordinateurs n'est qu'une des nombreuses applications faisant appel à la localisation de visages dans une image ou une séquence vidéo. Parmi celles-ci, on peut citer l'obtention d'informations biométriques dans le cadre de systèmes de sécurité, la vidéo-surveillance, le classement automatique d'images ou de séquences vidéo dans une base de données ou encore la vidéo-conférence.

1.2 Objectifs

L'objectif de ce travail est la conception d'un système de détection et de localisation de personnes et de visages dans une séquence vidéo destiné à être utilisé dans le cadre du développement d'un système d'interaction entre un être humain et un ordinateur.

Le système développé devra être suffisamment rapide pour pouvoir être utilisé dans le cadre d'une application interactive. Il devra donc être capable de traiter un nombre suffisant d'images par seconde (idéalement 25).

Il devra également être capable d'effectuer un suivi des personnes détectées. Il ne pourra donc pas se contenter de détecter les visages indépendamment dans chaque image mais devra pouvoir suivre le parcours d'une personne image après image.

Une séquence typique correspondant à l'utilisation prévue du détecteur consiste en une personne entrant dans la pièce, éventuellement par une porte située dans le champ de la caméra, et venant s'asseoir en face de celle-ci. Le système devra donc permettre la détection et le suivi d'une personne quelle que soit l'occlusion, partielle ou totale, de son corps, à l'exception du visage.

Cette dernière contrainte nous pousse à nous tourner vers un détecteur de visages uniquement. Nous ferons l'hypothèse réaliste que la détection de la présence d'une personne équivaut à celle de la présence d'un visage : une personne se trouve en dessous de chaque visage détecté.

1.3 Aperçu

Le présent document est organisé de la façon suivante :

- Le chapitre 2 comporte une étude bibliographique essentiellement consacrée à la détection de visages. L'objectif est de trouver, au sein des nombreuses techniques existantes, une série d'outils pouvant nous être utiles pour le développement de notre technique de détection.
- Dans le chapitre 3, nous décrirons en détail le système que nous avons mis au point et justifierons les choix effectués lors de sa conception.
- Les performances obtenues par celui-ci sont exposées et discutées au chapitre 4. Nous tenterons également d'y pointer une série de pistes à explorer en vue de les améliorer.
- Nous terminerons par le chapitre 5, dans lequel nous tirerons les conclusions du travail effectué.

Chapitre 2

Étude bibliographique

Dans ce chapitre, nous nous intéressons aux nombreuses techniques existantes décrites dans la littérature. Puisque, comme expliqué précédemment, notre objectif est avant tout de concevoir un détecteur de visages, nous nous intéressons ici essentiellement aux techniques de détection de visages. Nous aborderons parfois la détection de personnes mais la quasi-totalité de cette étude bibliographique est consacrée à celle des visages.

Il existe un très grand nombre de telles techniques (certains auteurs en dénombrent plus de 150). Notre objectif ici n'est pas de faire une énumération exhaustive de celles-ci mais plutôt de donner un aperçu de l'ensemble des différents types de méthodes existantes.

Nous classons les différentes techniques en quatre catégories [60] :
Les techniques basées sur la connaissance (*knowledge based*). Elles sont basées sur la définition de règles qui ne sont rien d'autre que la traduction plus ou moins fidèle de la connaissance qu'a un être humain de ce qu'*est* un visage. Cette connaissance est en général faite de relations liant les différents traits du visage (*features*). Ces techniques sont l'objet de la section 2.1.
Les techniques basées sur la recherche de traits invariants (*feature invariant*). Elles partent du principe que les visages comportent certains traits, certaines caractéristiques, dont l'existence et les propriétés ne dépendent pas de paramètres habituellement supposés inconnus tels que le point de vue, les conditions d'éclairage ou l'orientation du visage. Elles sont décrites à la section 2.2.
Les techniques basées sur la correspondance avec des modèles (*template matching*). Celles-ci recherchent dans l'image les zones correspondant le mieux à des modèles préalablement définis (éventuellement paramétriques ou déformables) sensés capturer au mieux l'ensemble des apparences que peut prendre un visage (*template matching*). Ces méthodes sont abordées à la section 2.3.
Les techniques utilisant des modèles obtenus par *apprentissage*. Dans ces dernières, le modèle recherché dans l'image n'est pas défini a priori mais est obtenu en *entraînant* le programme sur une base de données d'images contenant ou ne contenant pas de visages. La section 2.4 est consacrée à la description de ces techniques.

FIG. 2.1 – Sous-échantillonnages successifs de l'image. Chaque cellule carrée de taille $n \times n$ se voit attribuée la valeur moyenne des pixels lui correspondant dans l'image originale.

2.1 Techniques basées sur la connaissance

Cette approche consiste à déduire les règles de détection à partir de la connaissance qu'a un être humain de ce qu'est un visage. Il n'est pas compliqué de trouver des règles simples permettant de caractériser les différents traits du visage et les relations qui les lient. Par exemple, un visage vu de face présente souvent dans une image deux yeux symétriques l'un envers l'autre, un nez et une bouche. Les relations liant ces différents traits peuvent tout simplement être constituées de leurs distances et positions relatives.

Le principal problème de ce type d'approche est d'arriver à trouver un ensemble de règles bien définies traduisant fidèlement la connaissance humaine de ce qui caractérise l'*objet visage*. Il est également difficile d'arriver à définir un ensemble de règles réalisant un bon compromis entre le nombre de fausses détections et la quantité de visages non-détectés. Un ensemble de règles trop général produira un grand nombre de fausses détections mais rendre ces règles plus strictes risque d'empêcher certains visages d'être détectés. Trouver ce compromis est en fait un défi commun à toutes les techniques de détection.

Souvent, ces approches se limitent à la détection de visages vus de face puisqu'il est très difficile, voir quasi impossible, de produire un ensemble de relations liant les différents traits qui soit valable quel que soit le point de vue sous lequel le visage est observé.

La technique la plus représentative de ce type d'approche est celle proposée par Huang et Yang dans [57]. Ils y proposent un système de détection de visages hiérarchique basé sur trois niveaux de règles : une région étant considérée comme étant un visage si elle passe successivement à travers chacun de ces trois niveaux. Ils emploient une série de sous-échantillonnages de l'image (voir figure 2.1). Les règles de niveau 1 sont appliquées à toutes les fenêtres de l'image de plus basse résolution. Les fenêtres sélectionnées lors de cette première étape sont ensuite examinées à des résolutions de plus en plus importantes lors de l'application des règles de niveaux 2 et 3.

Les règles de niveau 1 permettant de localiser des zones candidates à basse résolution sont du type (voir figure 2.2) *"la partie centrale du visage contient quatre cellules d'intensités relativement uniformes"*, ou bien *"la partie supérieure du visage est d'intensité relativement uniforme et est significativement plus claire que la partie centrale"*.

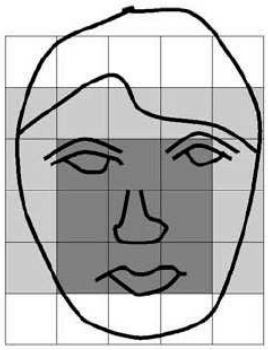

FIG. 2.2 – Exemple de modèle de visage utilisé pour l'obtention des règles de niveau 1 de la technique de détection hiérarchique de Yang et Huang [57].

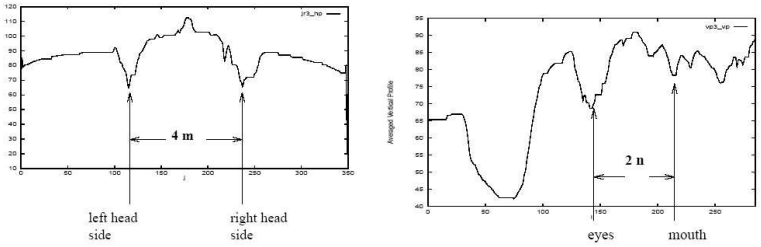

FIG. 2.3 – Projections horizontale et verticale des niveaux de gris d'une image de visage. On voit que les minima locaux de ces projections permettent de déterminer l'emplacement du visage dans l'image ainsi que les coordonnées verticales de certains de ses traits. [22]

Lors de l'étape 2, ils effectuent notamment une égalisation locale des histogrammes suivie d'une détection de bords sur les régions sélectionnées lors de l'étape 1.

L'étape 3 comporte un ensemble de règles permettant de localiser les différents traits du visage (yeux, nez, bouche...).

Les taux de détection obtenus par cette technique ne sont pas très élevés. On retiendra surtout l'approche intéressante consistant à examiner l'image à des résolutions de plus en plus élevées en éliminant à chaque étape une série de régions de celle-ci qui permet de ne devoir examiner en pleine résolution qu'une faible partie de l'image. Ceci permet de diminuer fortement le temps de calcul nécessaire au traitement d'une image.

Dans [22], Kotropoulos et Pitas présentent une extension de ce système. Leur objectif est d'une part d'éviter de devoir inspecter toutes les fenêtres possibles lors de la première étape (afin de réduire le temps de calcul) et d'autre part d'améliorer le taux de détection.

Pour ce faire, ils inspectent les projections horizontale et verticale des niveaux de gris $I(x, y)$ des images. Ces projections sont définies par $\sum_{y=1}^{n} I(x, y)$ et $\sum_{x=1}^{m} I(x, y)$. Ils font ainsi l'hypothèse que les deux minima locaux (caractérisés par des variations abruptes) de

FIG. 2.4 – Illustration des limites de la technique de localisation de Kotropoulos et Pitas [22]. A **gauche**, il est possible de déterminer la position du visage et de ses traits par inspection des minima locaux. Au **centre**, on voit qu'un arrière-plan complexe compromet cette localisation. L'image de **droite** montre l'échec du système si plusieurs visages sont présents dans l'image.

la projection horizontale de l'image correspondent aux bords gauche et droit du visage. Ceux de la projection verticale de l'image correspondent à la bouche, au bord du nez et aux yeux (voir figure 2.1). C'est seulement une fois que la région candidate est ainsi sélectionnée qu'ils sous-échantillonnent l'image et appliquent une méthode similaire à celle proposée par Yang et Huang. La connaissance préalable de la zone à examiner leur permet d'effectuer ce sous-échantillonnage en utilisant des cellules non plus carrées mais rectangulaires.

Ce sous-échantillonnage mieux adapté à la région candidate leur permet d'obtenir un gain significatif au niveau de la qualité de la détection par rapport au système précédent. Ils rapportent un taux de détection correcte de $86,5\%$. Cependant, leur système présuppose que le visage occupe une partie importante du centre de l'image et que le fond de celle-ci n'est pas trop complexe, sans cela il montre rapidement ses limites (voir figure 2.4).

2.2 Techniques basées sur la recherche de traits invariants

Les techniques basées sur la connaissance de la section précédente sont dites *top-down* en ce sens qu'elles détectent l'objet visage dans son entièreté avant d'investiguer plus en avant pour éventuellement détecter à l'intérieur de celui-ci les différents traits qui le composent. A l'inverse, les techniques basées sur la recherche de traits invariants présentées dans cette section sont dites, pour la plupart, *bottom-up*. Généralement, elles consistent à détecter indépendamment dans l'image différents éléments constitutifs d'un visage et décider ensuite si leur arrangement spatial permet de conclure à la présence d'un visage.

L'hypothèse sous-jacente est que, si les humains peuvent détecter des visages sans effort quels que soient les conditions d'éclairage ou le point de vue sous lequel ceux-ci sont observés, il doit exister dans les visages une série de caractéristiques dont la présence et la forme particulière ne sont pas ou peu influencées par ces conditions.

Souvent, on détecte dans une première étape les différents traits du visage tels que le nez, la bouche, les sourcils et les yeux. On utilise lors de la seconde étape un modèle probabiliste afin de décider si leur arrangement spatial permet de conclure à la présence d'un visage.

Le problème avec ces techniques basées sur la détection de traits est que ceux-ci sont peu indépendants du bruit ou des conditions d'illumination : une ombre portée sur un visage peut compromettre facilement une détection de traits classique basée sur une inspection des bords (dérivée, gradient, ...) de l'image.

Nous classons ces techniques de détection en quatre types que nous allons aborder successivement : celles basées sur la détection des traits du visage, celles utilisant la couleur de la peau, celles exploitant la texture du visage et enfin celles combinant ces différents types de caractéristiques.

2.2.1 Détection des traits du visage

Leung, Burl et Perona ont conçu dans [24] un modèle probabiliste capable de détecter des visages sur un fond complexe (beaucoup de méthodes de détection présupposent que l'arrière-plan est relativement uniforme) basé sur la détection locale de 5 traits du visage : les deux yeux, les deux narines et la jonction entre le nez et la bouche.

Ces traits sont détectés en filtrant l'image avec une série de filtres dérivatifs Gaussiens d'échelles et d'orientations variées. Ensuite, pour chaque pixel de l'image, ils comparent les réponses des différents filtres avec une série de réponses types (une par trait à détecter). Lors de cette première étape, ils retiennent les deux positions dont la réponse est la plus forte, c'est-à-dire les deux points dont la réponse est la plus proche d'un des modèles de réponse. Ils se servent ensuite de ces deux positions afin d'estimer les positions probables des autres traits. Par exemple, s'ils ont retenu lors de la première étape deux points supposés correspondre aux deux yeux, les narines et la jonction nez-bouche ne peuvent

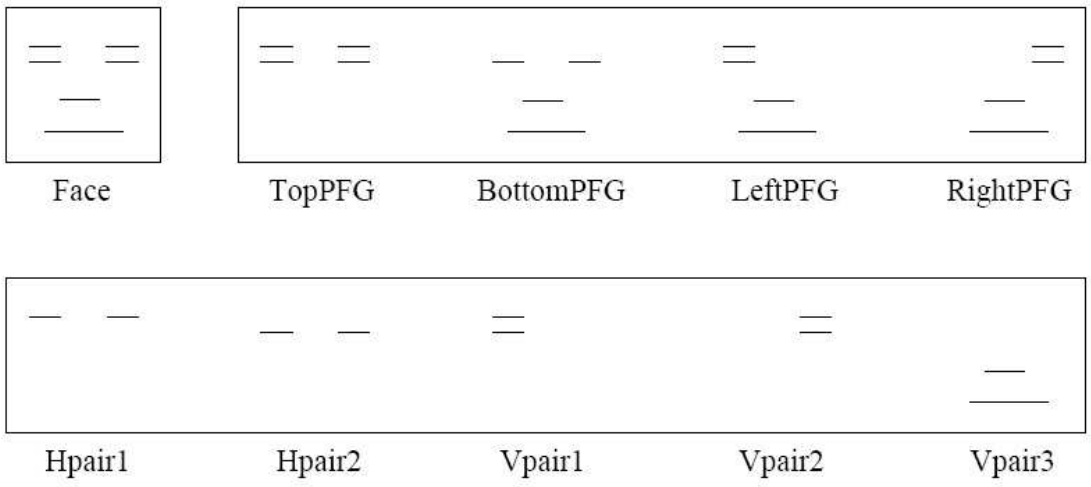

FIG. 2.5 – Modèle du visage utilisé par Yow et Cipolla dans [61]. Le modèle initial peut être subdivisé en 4 modèles partiels eux-mêmes constitués d'un sous-ensemble des 5 traits initialement recherchés dans l'image.

alors se trouver que dans une région bien déterminée de l'image.

Une fois ainsi déterminée une zone à l'intérieur de laquelle doivent se trouver les différents traits, il faut inspecter les différentes constellations de traits détectés *à l'intérieur de celle-ci*. Le problème de former les constellations les plus susceptibles d'être un visage est traité ici comme un problème d'assortissage aléatoire de graphes (*random graph matching*) dans lequel les noeuds du graphe sont les traits du visage et les arcs les distances entre ceux-ci.

Ils construisent un modèle probabiliste leur donnant la probabilité qu'a une constellation soit d'être un visage, soit d'avoir été générée par un autre processus. C'est cette dernière qui leur permet de choisir entre les différentes constellations candidates.

Ils testent leur détecteur sur une base de données de 150 images. Une détection est considérée comme correcte si trois des cinq traits sont situés au bon endroit. Avec ce critère, ils obtiennent 86% de détections correctes.

Yow et Cipolla décrivent dans [61] un système de recherche de visages basé sur la détection et le regroupement de traits du visage. Ces traits sont détectés par inspection de l'application d'un filtre dérivatif Gaussien du second ordre à l'image originale.

Ils modélisent le visage par une série de bords correspondant aux yeux, aux sourcils, au nez et à la bouche. Ils décomposent ce modèle en 4 modèles partiels qui peuvent eux-mêmes être subdivisés en 5 types de traits. Ceci est illustré à la figure 2.5.

La première étape consiste donc à détecter ces différents traits élémentaires. Pour cela, ils s'intéressent aux maxima locaux de la sortie du filtre dérivatif Gaussien du second ordre. Ils les considèrent comme les positions possibles des différents traits. Ils inspectent ensuite, toujours dans l'image filtrée, les bords situés aux alentours de ceux-ci et les regroupent en

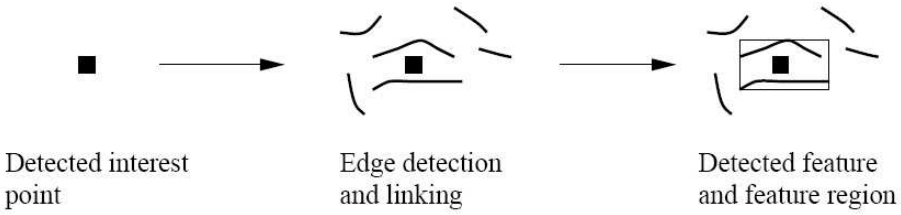

FIG. 2.6 – Illustration du processus de sélection des zones susceptibles de contenir un trait du visage décrit par Yow et Cipolla dans [61].

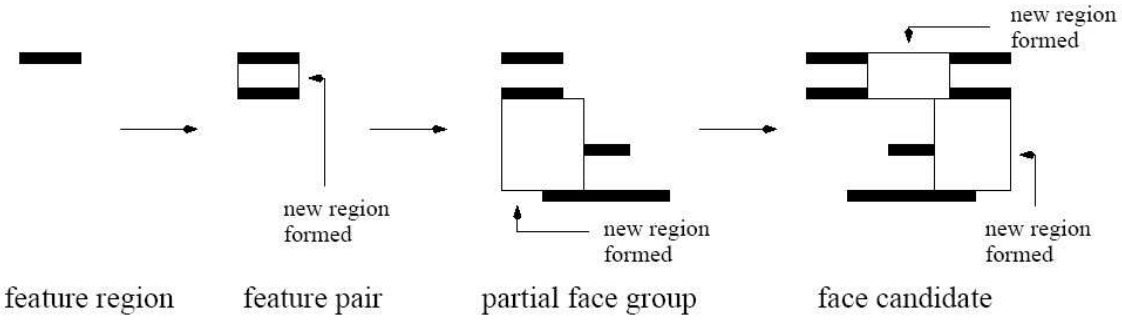

FIG. 2.7 – Regroupement des traits en modèles partiels et des modèles partiels en modèles complets dans la technique de Yow et Cipolla décrite dans [61].

différentes régions en utilisant des critères de proximité et de ressemblance en orientation et en intensité (voir figure 2.6).

Pour chaque région ainsi sélectionnée, ils effectuent une série de mesures (telles que les longueurs des différents bords et leurs intensités) et les placent dans un vecteur de caractéristiques. A partir d'une base de données d'images de visages, ils calculent la moyenne et la matrice de covariance des vecteurs de caractéristiques de chacun des cinq traits recherchés. Ils décident qu'une région contient l'un des traits si la distance de Mahalanobis[1] de son vecteur de caractéristiques est en dessous d'un certain seuil.

Une fois les traits détectés, ils sont regroupés en modèles de visages partiels en ne retenant que ceux donnant lieu à un arrangement spatial cohérent. Les modèles partiels sont ensuite regroupés en modèles complets de la même façon (voir figure 2.7).

Chaque modèle complet ainsi formé est ensuite soumis à un modèle probabiliste Gaussien afin de réduire le nombre de fausses détections.

Un des principaux avantages de cette technique de détection est qu'elle permet une détection correcte de visages ayant des orientations et des positions variées. Néanmoins, Yow et Cipolla rapportent un taux de fausses détections du système de 28% et une chute

[1] La distance de Mahalanobis mesure l'éloignement d'un objet $\mathbf{x}$ par rapport à une distribution $\mathcal{N}(\overline{\mathbf{x}}, \mathbf{\Sigma})$ en tenant compte de la matrice de covariance. Un même écartement Euclidien de la moyenne donnera lieu à une distance de Mahalanobis plus petite ou plus grande suivant que cet écartement est orienté selon une direction de variance élevée ou basse. Sa définition mathématique est $d_M = (\mathbf{x} - \overline{\mathbf{x}})\Sigma^{-1}(\mathbf{x} - \overline{\mathbf{x}})$.

des performances lorsque la taille des visages descend en dessous de 60×60 pixels.

2.2.2 Détection de la texture du visage

Les visages humains ont une texture particulière qui peut être employée afin de les distinguer des autres objets.

C'est ce que font Dai et Nakano dans [11]. Ils se servent de la matrice de dépendance spatiale des niveaux de gris (*space grey-level dependance matrix*) pour obtenir une série de grandeurs caractérisant les textures.

Cette matrice est définie par

$$P_{ab}(m,n) = \#\{((i,j),(i+m,j+n)) \in (L_x \times L_y) \mid I(i,j) = a, I(i+m,j+n) = b\},$$

où L_x et L_y sont les dimensions de l'image et $\#$ désigne la cardinalité d'un ensemble. La matrice est donc fonction des deux paramètres m et n.

Ils se servent de ces grandeurs caractéristiques pour rechercher les zones dont la texture correspond à celle du visage. Ils incorporent également l'information de couleur dans leur système afin de présélectionner les zones dont la teinte se situe autour de l'orange. Cette teinte correspond à celle de la peau des asiatiques, les peaux blanches ayant une teinte se situant plutôt aux alentours du rouge. Pour cette détection de teinte, ils utilisent la composante I de l'espace de couleur YIQ. L'incorporation de l'information de couleur leur a permis d'augmenter les performances, déjà correctes, de leur système basé sur la texture. Ils rapportent un taux de détection de 100% sur un ensemble de 30 images de tests comportant 60 visages.

Un avantage d'une telle technique est qu'elle peut détecter des visages relativement indépendamment de leur pose ou de leur orientation ou du fait qu'ils comportent une barbe ou des lunettes.

2.2.3 Détection de la couleur de la peau

La couleur de la peau est une des caractéristiques les plus saillantes et les plus facilement détectables d'un visage. Beaucoup de méthodes de détection y font appel de manière plus ou moins fondamentale.

Il existe un très grand nombre de techniques de détection de la peau. Les méthodes les plus simples consistent en un simple seuillage sur une ou plusieurs composantes d'un espace de couleur particulier. Les plus complexes font appel à des classificateurs. Deux de celles-ci sont décrites à la section 2.4.2.

Bien que la plupart des personnes possèdent une couleur de peau qui leur est propre, il a plusieurs fois été montré que ces différences résident beaucoup plus dans la luminance que dans la chrominance. C'est la raison pour laquelle la plupart des techniques de détection de la peau font appel à des espaces de couleurs à l'intérieur desquels la luminance et la chrominance sont découplées. Les espaces les plus souvent utilisés sont les espaces RGB normalisé, HSV, YIQ et $YCrCb$. Il existe des techniques faisant appel au classique espace RGB bien qu'à l'intérieur de celui-ci, la luminance et la chrominance ne sont pas

découplées.

La technique la plus simple consiste, en se plaçant par exemple dans l'espace $YCrCb$, à choisir deux tranches $[Cr_1, Cr_2]$ et $[Cb_1, Cb_2]$ et considérer qu'un pixel est de la couleur de la peau si $Cr_1 \leq Cr \leq Cr_2$ et $Cb_1 \leq Cb \leq Cb_2$. On utilise généralement des histogrammes obtenus à partir d'une série d'images témoins comportant de la peau afin de déterminer au mieux les valeurs de ces seuils. Bien que fort simple, une telle approche permet déjà d'obtenir des résultats suffisants pour certaines applications. De plus, une telle technique présente l'avantage souvent crucial de n'être que très peu coûteuse en temps de calcul. En effet, la seule opération y nécessitant une quantité significative de calculs est l'éventuel changement d'espace de couleurs.

Il peut donc être tout à fait justifié de choisir une approche aussi simple. La question qui se pose alors est de savoir quel espace de couleurs utiliser. Dans [52], Terraillon fait une étude comparative de différents espaces de couleurs pour la détection de peau dans le cadre de la détection de visages. Il y montre qu'en général, la distribution de la couleur de la peau peut être modélisée correctement par une simple Gaussienne à l'intérieur d'un espace de couleurs normalisé et qu'il n'est nécessaire de recourir à une somme de Gaussiennes que lors de l'utilisation d'espaces non normalisés. Lors de leurs tests de détection de visages, c'est l'espace TSL (Teinte, Saturation, Luminance) normalisé qui donne les meilleurs résultats. Mais la conclusion générale qu'il tire est que le critère le plus important pour la détection de visages est le degré de recouvrement entre les distributions de la peau et de la "non-peau" dans l'espace considéré, recouvrement fortement dépendant du nombre d'exemples disponibles pour l'entraînement du système.

Saxe et Foulds proposent dans [44] une technique de détection de peau itérative faisant appel à l'intersection d'histogrammes à l'intérieur de l'espace HSV. Un ensemble initial de pixels de couleur peau est choisi par l'utilisateur pour permettre l'initialisation de l'algorithme itératif. Afin d'identifier les pixels de couleur peau, leur technique balaie l'entièreté de l'image, ensemble de pixels par ensemble de pixels. Pour chacun de ces ensembles, ils comparent l'histogramme de l'ensemble considéré avec un histogramme de référence. Ils utilisent l'intersection des histogrammes comme critère de comparaison (distance). Si la mesure de distance entre l'histogramme de la zone inspectée et l'histogramme de contrôle est en dessous d'un certain seuil, la zone est classifiée comme étant de couleur peau.

McKenna choisit dans [28] de modéliser la densité de probabilités qu'a un pixel d'être de la couleur de la peau par une somme de Gaussiennes à l'intérieur de l'espace HS (teinte, saturation). Sa technique est similaire à celle développée par Jebara et Pentland décrite à la section 2.4.2. Jones et Regh ont construit dans [18] un classificateur en utilisant une base de données d'entraînement d'un milliard de pixels faite à partir d'images collectées sur internet. Cette technique est également décrite à la section 2.4.2.

L'information de couleur est un outil efficace pour identifier les régions susceptibles

d'être un visage. Cependant, il est difficile d'obtenir une modélisation de la couleur de la peau qui soit indépendante des conditions d'illumination.

Pour résoudre ce problème, McKenna et Raja proposent dans [29] une modélisation de la couleur de la peau par un modèle Gaussien *adaptatif* permettant de suivre des visages sous des conditions d'illumination changeantes. A l'hypothèse forte que le modèle probabiliste de la couleur de peau reste inchangé au cours du temps, ils substituent l'hypothèse plus légère que celui-ci ne peut changer du tout au tout entre deux images successives. Ils mettent donc à jour les paramètres de leur modèle Gaussien à chaque nouvelle image. Les résultats qu'ils obtiennent montrent que leur système est bien capable de suivre un visage sous des conditions d'illumination changeantes. Cependant, il ne peut être utilisé pour détecter les visages une première fois. Il faut donc lui adjoindre un autre détecteur afin d'initialiser le suivi des visages.

C'est également l'approche adoptée par Yang et Waibel dans [58]. Ils modélisent la couleur de la peau par une distribution Gaussienne dans les composantes r et g de l'espace de couleurs RGB normalisé. Les paramètres de cette distribution Gaussienne (moyennes de r et g et matrice de covariance) sont mis à jour à chaque image selon les règles suivantes :

$$\widehat{\overline{r}}_k = \sum_{i=0}^{N-1} \alpha_{k-i} \widehat{\overline{r}}_{k-i}$$

$$\widehat{\overline{g}}_k = \sum_{i=0}^{N-1} \beta_{k-i} \widehat{\overline{g}}_{k-i}$$

$$\widehat{\overline{\Sigma}}_k = \sum_{i=0}^{N-1} \gamma_{k-i} \widehat{\overline{\Sigma}}_{k-i}$$

N est la taille de la fenêtre temporelle considérée, $\widehat{\overline{r}}_k$, $\widehat{\overline{g}}_k$ et $\widehat{\overline{\Sigma}}_k$ sont les estimations à l'instant k des deux moyennes et de la matrice de covariance et les poids α_k, β_k et γ_k servent à définir l'influence qu'ont les différentes estimations précédentes sur les nouvelles. Comme dans le système précédent, il est également nécessaire d'initialiser le détecteur soit à la main, soit grâce à un second détecteur.

L'utilisation de la couleur caractéristique de la peau peut donc s'avérer être un outil précieux, voir essentiel, dans le cadre de la détection de visages. Cependant, cet outil ne peut constituer à lui seul un détecteur de visages fiable. Il existe un certain nombre de méthodes combinant la détection de peau et, entre autres, la détection de traits du visage ainsi que l'exploitation de l'information temporelle. Ces méthodes sont l'objet de la prochaine section.

2.2.4 Recherche de caractéristiques multiples

Les méthodes présentées dans cette section combinent un certain nombre d'aspects des techniques présentées précédemment. La plupart d'entre elles utilisent des caractéristiques globales telles que la couleur de la peau et la forme afin de sélectionner des zones

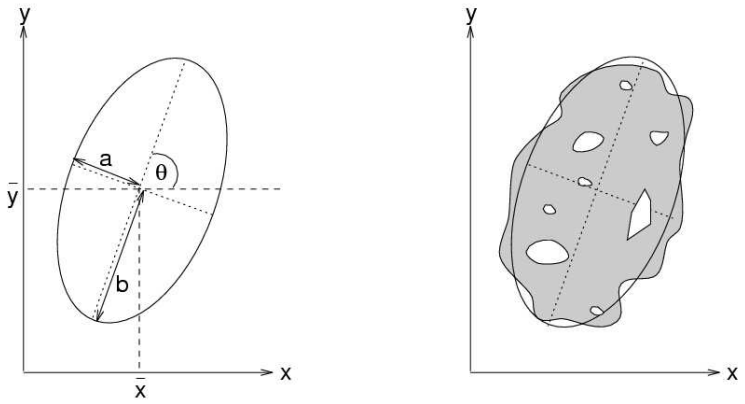

FIG. 2.8 – Meilleure ellipse pour une zone de peau connexe [48].

FIG. 2.9 – Détection des pixels de peau et génération des meilleures ellipses pour les zones connexes de taille suffisante.

candidates. Ces zones candidates sont ensuite inspectées à l'aide de détecteurs de traits tels que les yeux, le nez ou la bouche afin de vérifier si elles comportent bien un visage.

La schéma de détection le plus fréquent consiste à commencer par une détection de peau. On retient ensuite les zones de peau connexes de forme elliptique. On cherche alors à l'intérieur de ces zones elliptiques les différents traits du visage afin de décider si cette région est un visage ou pas.

C'est l'approche adoptée par Sobottka et Pitas dans [48].

Ils commencent par effectuer une détection de peau grâce à un seuillage dans l'espace de couleur HSV. Ensuite, pour chaque zone connexe de peau, l'ellipse lui correspondant le mieux est générée au moyen des moments géométriques de l'image (voir figures 2.8 et 2.9).

Ils sélectionnent ensuite les zones qui sont correctement approximées par leur meilleure ellipse et rejettent celles dont le grand axe est orienté trop horizontalement. Ils vérifient

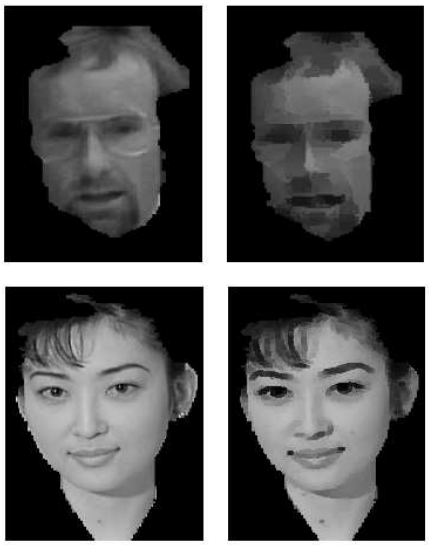

FIG. 2.10 – Opérations morphologiques effectuées par Sobottka et Pitas dans [48] afin de faire ressortir les traits du visage.

ensuite que ces surfaces elliptiques sont bien des visages en effectuant une recherche de traits à l'intérieur de celles-ci. Cette recherche est classiquement effectuée en exploitant le fait que le nez, les yeux et la bouche sont des zones globalement plus sombres et donnent donc lieu à des minima locaux sur les projections horizontale et verticale de l'image. Ils effectuent au préalable différents traitements morphologiques afin de mieux faire ressortir la bouche, les yeux et les sourcils (voir figure 2.10).

Ils rapportent un taux de détection correcte des traits du visage de 86%.

La recherche de peau et le calcul des ellipses correspondant le mieux aux régions de peau connexes sont deux opérations également effectuées dans [59], [49] et [53].

Ces ellipses sont obtenues à partir des moments d'inertie de l'image. Si $g(x, y)$ représente les niveaux de gris de l'image, ces moments sont définis par [19] :

$$\mu_{p,q} = \int (x_1 - \overline{x_1})^p \, (x_2 - \overline{x_2})^q \, g(x_1, x_2) \, dx \, dy$$

où

$$\overline{x_i} = \frac{\int x_i \, g(x, y) \, dx \, dy}{\int g(x, y) \, dx \, dy}.$$

Pour une image discrète binaire, le calcul des moments se réduit à

$$\mu_{p,q} = \sum (x_1 - \overline{x_1})^p (x_2 - \overline{x_2})^q,$$

la sommation s'effectuant sur tous les pixels de l'image.

FIG. 2.11 – Meilleures ellipses de zones de peau connexes [53].

L'angle θ définissant l'orientation de l'ellipse (voir figure 2.8) est donné par

$$\theta = \frac{1}{2} \arctan\left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right). \tag{2.1}$$

Si on définit (C désignant la zone connexe de pixels de peau)

$$I_{min} = \sum_{(x,y)\in C} [(x - \overline{x}) \cos\theta - (y - \overline{y}) \sin\theta]^2 \tag{2.2}$$

$$I_{max} = \sum_{(x,y)\in C} [(x - \overline{x}) \sin\theta - (y - \overline{y}) \cos\theta]^2, \tag{2.3}$$

la longueur du grand axe a et celle du petit axe b sont alors données par

$$a = \left(\frac{4}{\pi}\right)^{1/4} \left[\frac{(I_{max})^3}{I_{min}}\right]^{1/8} \qquad b = \left(\frac{4}{\pi}\right)^{1/4} \left[\frac{(I_{min})^3}{I_{max}}\right]^{1/8}. \tag{2.4}$$

Afin de déterminer à quel point la forme de la zone connexe se rapproche de celle d'une ellipse, Sobottka et Pitas proposent dans [48] la mesure de distance suivante (où E est la meilleure ellipse pour la zone connexe C) :

$$V = \frac{\sum_{(x,y)\in E}(1 - b(x,y)) + \sum_{(x,y)\in C\setminus E} b(x,y)}{\sum_{(x,y)\in E}} \tag{2.5}$$

avec

$$b(x,y) = \begin{cases} 1 & \text{si } (x,y) \in C \\ 0 & \text{sinon.} \end{cases}$$

V détermine la distance entre la zone connexe et l'ellipse en comptant les trous à l'intérieur de celle-ci et les points de la zone connexe qui se trouvent à l'extérieur de celle-ci.

Comme on peut le voir sur la figure 2.11, les ellipses ainsi obtenues correspondent remarquablement bien aux visages auquels elles appartiennent. C'est une technique efficace qui présente l'avantage de rester relativement simple à implémenter et, surtout, intuitive.

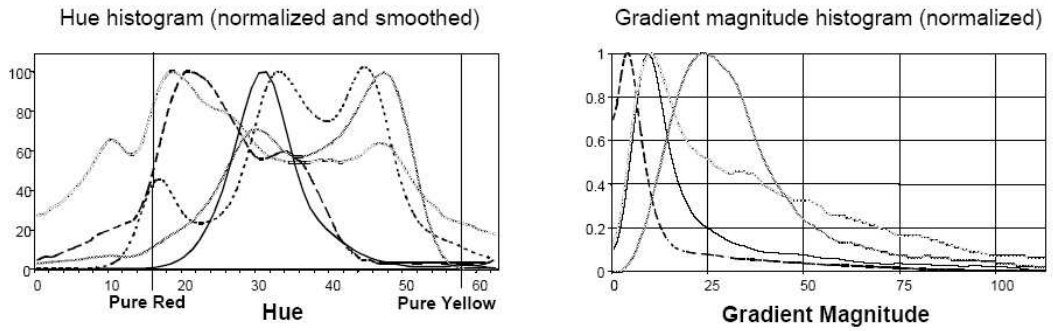

FIG. 2.12 – Histogrammes de la teinte et de la *texture* de différentes images de peau. On voit qu'une détection optimale nécessite une adaptation des seuils selon l'image traitée. [27]

De plus, elle est relativement peu coûteuse en temps de calcul. Cependant, il reste nécessaire d'inspecter les zones ainsi détectées (recherche de traits ...) afin d'éliminer les nombreuses fausses détections produites par une simple recherche de zones de peau elliptiques : par exemple, une simple main en constitue souvent une.

Mateos et Chicote ont développé dans [27] une technique relativement similaire. Cependant, ils détectent les zones de peau à l'intérieur d'un espace particulier qu'ils appellent l'espace HIT (*Hue, Intensity, Texture*). La troisième composante, la texture, est constituée du module du gradient de Sobel multi-spectral effectué dans l'espace RGB. Ils détectent la peau par de simples seuillages à l'intérieur de ces trois composantes. Un pixel n'est classifié comme étant de la peau que s'il respecte chacun des trois seuillages. Avec des seuils adéquats, cette technique fonctionne remarquablement bien mais ces seuils ne peuvent être fixés a priori si on souhaite obtenir des performances optimales pour tous types d'images. Ceci est illustré à la figure 2.12.

Ils effectuent ensuite une recherche de traits à l'intérieur des zones candidates. Ils choisissent de subdiviser successivement la zone en plusieurs morceaux en utilisant leur connaissance de l'aspect général d'un visage. Ceci leur permet d'effectuer la plupart de leurs recherches de minima locaux des projections horizontale et verticale de l'image à l'intérieur de ces zones plus petites et donc moins susceptibles d'être bruitées. Leur démarche est illustrée à la figure 2.13.

Birchfield fait également usage de l'hypothèse de forme elliptique du visage et de l'information de couleur dans [4]. Son détecteur de visage comporte deux modules distincts : le premier est basé sur le gradient et le second sur la couleur.

Le module basé sur le gradient est chargé de vérifier si une zone comporte un contour elliptique. Pour cela, dans chaque zone considérée, il intègre le long d'une ellipse la composante du gradient de l'image qui lui est perpendiculaire. Le résultat de cette intégration est d'autant plus élevé que la zone considérée comporte un contour dont la forme est elliptique.

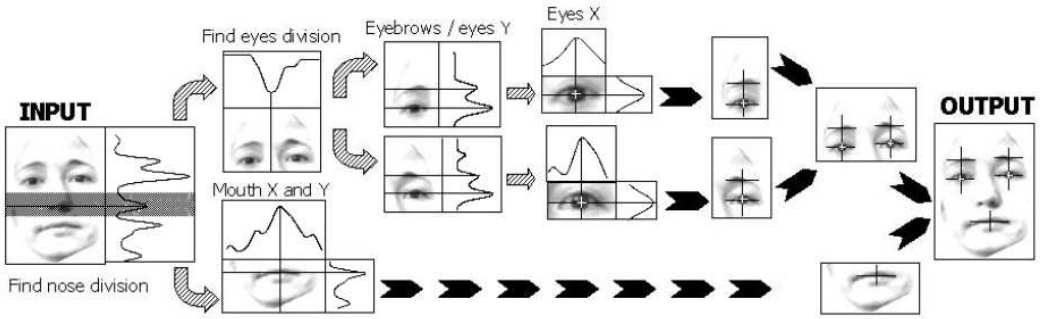

FIG. 2.13 – Illustration de la subdivision de la zone candidate effectuée par Mateos et Chicote dans [27] afin d'optimiser les recherches de minima locaux des projections horizontale et verticale de la zone.

Le module basé sur la couleur fait usage d'histogrammes dans l'espace $B-G, G-R, R+G+B$. Sa particularité est de permettre le suivi de personnes même lorsqu'elles tournent le dos à la caméra. L'aspect négatif de la technique est que le système doit être initialisé pour chaque nouvelle personne à suivre. Le sujet doit en effet présenter sa tête vue de trois-quarts afin que le système puisse générer les histogrammes caractéristiques de cette personne. Le sujet doit se présenter de trois-quarts afin de montrer à la caméra une surface suffisamment importante de cheveux. Une fois que le système est lancé, il calcule pour chaque zone l'intersection entre les histogrammes de celle-ci et les histogrammes témoins mémorisés lors de l'initialisation. Si N est le nombre de tranches que comporte un histogramme, M est l'histogramme témoin et $I_{\mathbf{s}}$ est l'histogramme de la tranche inspectée, cette intersection est définie par

$$\phi(\mathbf{s}) = \frac{\sum_{i=1}^{N} \min(I_{\mathbf{s}}(i), M(i))}{\sum_{i=1}^{N} I_{\mathbf{s}}(i)}.$$

La force de cette intersection réside dans la fonction min() qui permet de ne jamais comptabiliser plus de pixels d'une certaine couleur qu'il n'y en a dans l'histogramme témoin. Ainsi, le score d'une région comportant de la peau et des cheveux est plus élevé que celui d'une région ne comportant que de la peau. L'autre grande force des histogrammes est qu'ils ne comportent aucune information de géométrie et sont donc insensibles aux mouvements et aux changements de forme ou de taille des zones inspectées.

Ce système permet un suivi de visages robuste tout en étant relativement peu coûteux en temps de calcul. Son principal défaut est de nécessiter d'être initialisé pour chaque personne à suivre.

2.3 Techniques basées sur la correspondance avec des modèles

Ces techniques de détection font appel à des modèles de visage standards (*templates*) soit prédéfinis, soit paramétriques. L'examen d'une image consiste à évaluer la corrélation entre ses différents points et les modèles standards du contour du visage, des yeux, du nez ou encore de la bouche. La présence de visages à certains endroits est confirmée ou rejetée selon les résultats de ces corrélations.

L'avantage de telles méthodes est le fait qu'elles soient relativement simples à implémenter. Cependant, il a été prouvé qu'elles sont peu efficaces puisqu'elles ne peuvent prendre en considération les nombreuses variations d'apparence du visage.

C'est pour cette raison que des techniques de détection utilisant des modèles multi-résolutions, des modèles multi-échelles, des sous-modèles ou encore des modèles déformables ont été proposées.

2.3.1 Utilisation de modèles prédéfinis

Craw et ses associés présentent dans [9] une technique de localisation de visages basée sur un modèle prédéfini de visage vu de face. Un gradient de Sobel y est tout d'abord utilisé afin de détecter les bords de l'image. Ces bords sont ensuite regroupés afin d'arriver à retrouver le modèle de visage. Ces regroupements sont soumis à toute une série de contraintes afin d'éviter une reconstruction artificielle du modèle.

Une fois que le contour du visage a été détecté, le même processus est appliqué afin de détecter à l'intérieur de celui-ci une série de traits tels que les yeux, les sourcils ou les lèvres.

Craw est ses associés proposent plus tard dans [10] une technique de localisation basée sur un ensemble de 40 modèles permettant la recherche de 40 points particuliers du visage (voir figure 2.14). Leur modèle de visage a été généré à partir d'une base de données de 1000 visages. Sur chacun d'entre eux, les positions respectives de chacun des 40 points ont été mesurées et enregistrées afin de servir de base à l'établissement du modèle de visage.

Leur système de détection comporte deux parties distinctes :

- Un certain nombre de détecteurs indépendants chargés de localiser une série de parties du visage.
- Un système de contrôle chargé de valider ou d'invalider l'hypothèse de présence d'un visage dans une zone donnée en se basant sur les résultats que lui fournissent les détecteurs indépendants.

La figure 2.15 montre les modèles utilisés pour détecter le contour du visage et les yeux. Ceux-ci sont détectés en utilisant à la fois les niveaux de gris de l'image et ses bords détectés grâce à un filtre de Canny.

Le contour du visage est détecté en considérant toute une série de déformations, mises à l'échelle et rotations du modèle original obtenu en moyennant les contours des 1000 visages témoins. Les limites en dessous desquelles une déformation du modèle est considérée admissible sont également obtenues statistiquement.

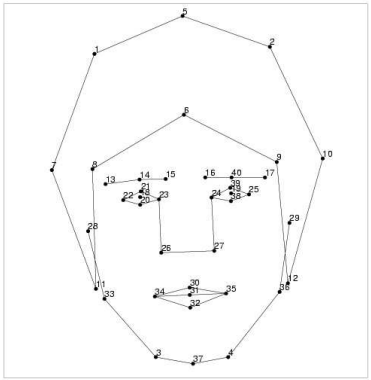

FIG. 2.14 – Les 40 points du visage recherchés par Craw dans [10].

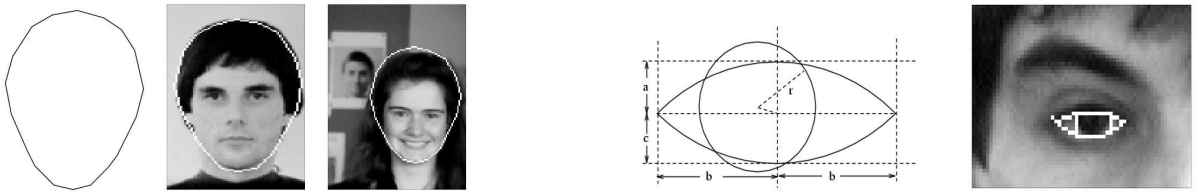

FIG. 2.15 – A **gauche** : modèle utilisé pour le contour du visage et deux exemples de déformation, rotation et mise à l'échelle de celui-ci afin de le faire correspondre avec le visage présent dans l'image. A **droite** : modèle paramétrique utilisé pour la détection des yeux et exemple de détection.

La détection des yeux est, elle, effectuée en considérant un ensemble de valeurs possibles pour chacun des 9 paramètres intervenant dans le modèle d'œil représenté à la figure 2.15.

Sinha fait usage dans [46] d'un petit nombre d'invariants spatiaux afin de décrire l'espace des visages.

L'idée permettant de générer ces invariants est que, alors que les variations d'illumination changent les intensités lumineuses des différentes parties du visage, les intensités relatives de ces parties y sont relativement insensibles (voir figure 2.16).

On peut alors trouver des invariants en déterminant les rapports entre les luminosités moyennes d'un certain nombre de régions du visage et en ne retenant que la "direction" de ces rapports. On obtient alors des invariants particulièrement robustes. Chacun de ces invariants s'intéresse à deux des régions du visage et stipule laquelle des deux est la plus claire.

On obtient ainsi un modèle relativement grossier du visage constitué d'un ensemble de régions bien choisies qui correspondent approximativement aux traits du visage tels que les yeux, la bouche ou encore le front. Une image est classifiée comme étant un visage

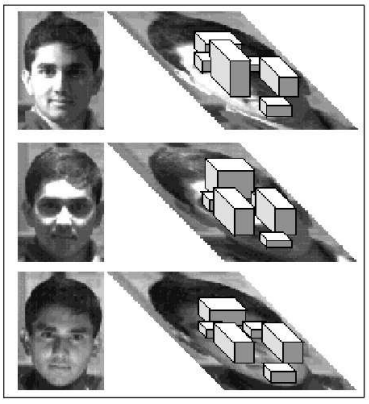

FIG. 2.16 – On voit ici que les luminosités relatives des différentes régions du visage sont globalement indépendantes des conditions d'illumination (la hauteur d'un bloc est proportionnelle à la luminosité moyenne de la zone de l'image qu'il couvre).[45]

si elle respecte toutes les contraintes sur les luminosités relatives des différentes paires de régions.

2.3.2 Utilisation de modèles déformables

Yuille et ses associés utilisent dans [63] des modèles déformables afin de détecter les traits du visage. Ceux-ci sont ici associés à des modèles paramétriques. Une fonction énergie est définie pour chacun d'entre eux. Les variables de cette fonction sont les différents paramètres intervenant dans le modèle. Le modèle déformable s'adapte aux contours du trait recherché en essayant de minimiser cette fonction énergie. Cette technique permet à Yuille et ses associés d'obtenir des bonnes performances pour le suivi de traits même lorsque ceux-ci changent de forme (ce qui peut par exemple arriver lors d'un changement de point de vue). Le gros défaut de cette technique est qu'elle nécessite que le modèle déformable soit initialement déposé dans le voisinage du trait recherché.

On trouve dans [23] une technique basée sur les contours actifs ou *snakes*.

Un contour actif sert à approximer le contour d'un objet. Il nécessite tout d'abord d'être déposé au voisinage de cet objet. Ensuite, durant un processus itératif, le contour est attiré vers le contour de l'objet par une série de forces contrôlant sa forme et sa position. La détermination de ces forces est de nouveau faite en cherchant à minimiser une fonction énergie. Cette énergie comporte deux termes : un terme d'énergie interne et un terme d'énergie externe. L'énergie interne sert à imposer que le contour actif soit plus ou moins régulier et continu. Cette énergie interne est contrebalancée par l'énergie externe qui impose au contour actif de se "coller" aux contours de l'objet. La position finale du contour actif correspond à un équilibre entre ces deux forces.

La première étape de la technique de détection décrite dans [23] consiste à filtrer l'image avec un filtre moyenneur afin d'éliminer une partie du bruit. On utilise ensuite des

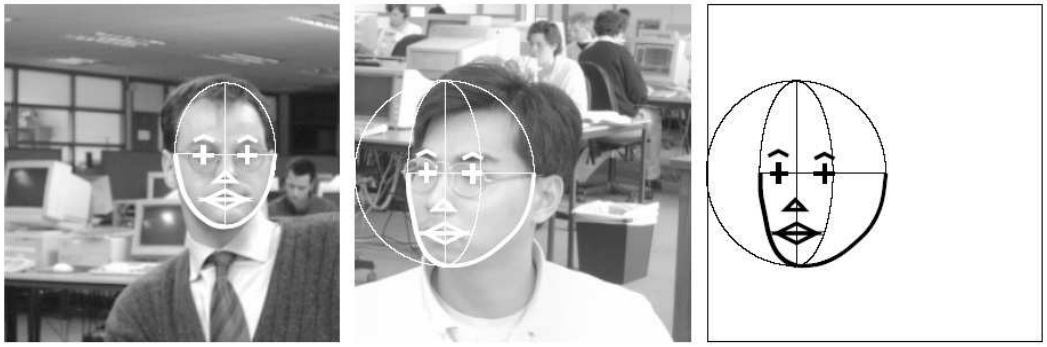

FIG. 2.17 – Modèle déformable utilisé pour détecter la moitié inférieure du visage par Yow et Cipolla dans [62].

opérations morphologiques pour accroître la taille des bords de celle-ci. Des petits contours actifs sont alors utilisés pour trouver et éliminer les petits bords courbes parasites. Une transformée de Hough[2] est ensuite utilisée pour détecter l'ellipse dominante parmi les contours qui ont survécu à l'étape précédente. Le contour du visage est donc ici une fois de plus approximé par une ellipse. Les ellipses dominantes ainsi détectées constituent les zones candidates pour la détection de visages.

On utilise ensuite une technique similaire à celle développée par Yuille et ses associés dans [63], que nous avons décrite un peu plus haut, pour détecter, ou pas, les différents traits du visage à l'intérieur de ces zones. Si un nombre suffisant de traits sont détectés à l'intérieur d'une zone et que les positions de ceux-ci sont cohérentes, on considère qu'un visage y a été détecté.

Yow et Cipolla font également usage de contours actifs afin de détecter des visages [62]. Ils choisissent classiquement de modéliser le contour du visage par une ellipse. Ils utilisent ainsi deux contours actifs en forme de quarts d'ellipses afin de rechercher le contour de la moitié inférieure du visage. Les yeux et la bouche étant, dans leur modèle, situés sur les grands et petits axes des deux quarts d'ellipses. Ceci est illustré à la figure 2.17.

Un contour actif finit toujours par converger vers une position. Il n'est donc pas possible de valider ou invalider la présence d'un visage grâce à l'éventuelle absence de convergence des contours actifs. Afin de pouvoir décider si un visage est présent, Yow et Cipolla font l'hypothèse qu'en cas d'absence de visage, les contours actifs convergeront vers des positions incompatibles avec l'hypothèse de présence d'un visage. Ils examinent donc les tailles, formes et positions de leurs contours actifs après convergence de ceux-ci afin de décider si un visage est présent ou pas.

[2]La transformée de Hough est un outil permettant de détecter des contours dont la forme est décrite par une équation mathématique.

2.4 Techniques utilisant des modèles obtenus par apprentissage

Les techniques présentées dans la section précédente faisaient appel à des modèles définis de manière arbitraire et cherchaient dans l'image les zones leur ressemblant le plus. Le problème est bien entendu qu'il est difficile de déterminer un modèle, ou un ensemble de ceux-ci, capturant toute la variabilité d'un objet tel qu'un visage humain. Nous allons ici nous intéresser à des techniques permettant de construire un test déterminant si une zone de l'image correspond à l'objet recherché ou pas à partir d'une base de données d'images étiquetées.

Les techniques présentées dans cette section ont pour principale caractéristique de faire appel à un ou plusieurs *classificateurs*.

Un classificateur peut être défini comme tout outil prenant en entrée un ensemble de caractéristiques appartenant à un objet et délivrant à sa sortie une étiquette indiquant la classe à laquelle appartient cet objet.

Dans le cas de la détection de visages, la démarche consiste typiquement à présenter au classificateur l'ensemble des fenêtres de taille déterminée présentes dans l'image. Celui-ci possède deux étiquettes de sortie possibles : soit la fenêtre contient un visage, soit elle n'en contient pas. Afin de pouvoir détecter des visages de différentes tailles, on répète ce processus sur plusieurs rééchantillonnages de l'image.

On se rend tout de suite compte que telle quelle, une telle technique a très peu de chances de donner lieu à des temps de calcul permettant l'utilisation du détecteur dans le cadre d'une applicaion interactive.

Les classificateurs nécessitent dans une première phase d'être entraînés. Ils le sont à partir d'une base de données d'images étiquetées. Dans le cas de la détection de visages, cette base de données contient une série d'images de visages de taille standard associées à l'étiquette "visage" et une série d'images, toujours de taille standard, ne contenant pas de visages et comportant l'étiquette "non-visage".

L'obtention d'une bonne base de données d'entraînement n'est pas triviale. Le risque d'introduire, par le choix des visages contenus dans celle-ci, un biais dans le modèle construit par le classificateur est toujours présent. De plus, s'il n'est déjà pas facile de produire un ensemble d'exemples représentant correctement le sous-espace constitué des images de visages, la taille de celui-ci est très petite comparée à celle de son complémentaire : le sous-espace des images ne comportant pas de visage ! Il est quasi impossible pour un opérateur humain de sélectionner un ensemble d'images représentant fidèlement l'ensemble des images ne comportant pas de visage.

Une solution souvent employée pour pallier ce problème est la technique dite du *bootstrapping*[3]. Celle-ci consiste à entraîner le programme une première fois en utilisant comme ensemble d'images représentant tout ce qui n'est pas un visage soit une première sélection

[3]bootstrap = programme d'amorçage.

d'exemples, soit des images constituées intégralement de bruit. On fait ensuite tourner le programme dans des conditions réelles d'utilisation en collectant les fenêtres détectées faussement comme étant un visage. Ces fenêtres sont ensuite introduites dans la partie "non-visage" de la base de données et le programme est à nouveau entraîné sur cette base de données augmentée. On peut, bien entendu, répéter le processus plusieurs fois jusqu'à ce que le nombre de fausses détections soit suffisamment bas, ou alors jusqu'à ce que celui-ci ne soit malheureusement plus diminué par les agrandissements successifs de la base de données. Il est, en règle générale, toujours préférable d'employer la technique du bootstrapping puisque ce faisant, on a beaucoup de chances d'introduire dans la base de données des exemples situés tout près de la frontière de décision qui sont ceux qui représentent le mieux la classe des motifs les plus susceptibles de provoquer une erreur de classification. Avoir une série d'exemples représentatifs des cas limites dans la base de données d'entraînement permet de faire pencher la balance du bon côté dans un grand nombre de cas limites.

Un recensement de bases de données de visages utilisées pour la détection de ceux-ci fait en 2001 peut être trouvé dans [3].

2.4.1 Choix d'une règle de classification

Lorsqu'on décide de considérer la recherche de visages comme un problème de classification, il faut choisir un règle de classification. Il en existe de nombreux types.

La règle de classification choisie doit tenir compte du coût induit par une erreur. A partir de ce coût, on peut définir le risque associé à une règle de classification particulière.

Si $L(i \to j)$ représente le coût d'attribuer la classe i à un objet de classe j, le risque total associé à la règle de classification r est

$$R(r) = P(1 \to 2|\textit{utilisant } r)L(1 \to 2) + P(2 \to 1|\textit{utilisant } r)L(2 \to 1).$$

Il existe un classificateur appelé *classificateur idéal de Bayes* qui minimise ce risque. Il est décrit dans l'annexe A.

Celui-ci fait intervenir les probabilités a posteriori $P(k|\mathbf{x})$ qui représentent la probabilité d'attribuer la classe k à l'objet $\mathbf{x}$. Le problème est que nous ne connaissons généralement pas exactement ces probabilités.

Nous devons donc en général construire notre classificateur à partir d'une base de données d'exemples. Il existe deux grandes approches :

- La **première** consiste à faire l'hypothèse que ces lois de probabilités suivent des modèles paramétriques connus (dont la loi normale, également dite Gaussienne, est l'exemple le plus connu) et à se servir de la base de données pour déterminer les valeurs de leurs paramètres.
- La **seconde** consiste à déterminer directement la frontière de décision (frontière située dans l'espace des caractéristiques présentées à l'entrée du classificateur) sans tenir compte de modèles de probabilités. Un cas particulier important pour les classificateurs binaires est celui où les données sont linéairement séparables, c'est-à-dire

qu'il existe un hyper-plan séparant parfaitement les points positifs des points négatifs. Nous ferons cette hypothèse lorsque nous introduirons les machines à vecteurs de support (*Support Vector Machines*, SVM) à la section 2.4.4.

2.4.2 Recherche des pixels correspondant à la couleur de la peau en utilisant un classificateur

Comme nous l'avons déjà vu, la peau possède une couleur extrêmement caractéristique et peut s'avérer être un outil précieux pour la détection de personnes. Nous allons explorer deux approches de conception d'un classificateur permettant de détecter les pixels correspondant à la couleur de la peau. La première technique utilise un modèle basé sur une mixture de Gaussiennes tandis que la seconde adopte une approche plus originale en se basant sur des histogrammes.

Classificateur basé sur une mixture de Gaussiennes

Jebara et Pentland proposent dans [17] de modéliser la densité de probabilités d'un pixel d'être de la couleur de la peau par une somme de Gaussiennes. La base de données d'entraînement sert ici à déterminer les valeurs des différents paramètres intervenant dans le modèle (moyennes, covariances et poids attribués à chaque distribution). Un tel modèle présente l'avantage de pouvoir être entraîné sur une base de données relativement restreinte tout en gardant des performances acceptables. Ils ne font pas mention de performances de ce classificateur dans leur article mais ce système a été repris et réentraîné par Jones et Regh [18] pour le comparer à leur système basé sur des histogrammes.

Classificateur basé sur des histogrammes

En plus de la technique classique décrite précédemment, on peut construire un classificateur relativement simple basé sur des histogrammes. C'est l'approche adoptée par Jones et Regh dans [18].

Ils proposent de construire des histogrammes dans l'espace RGB et de s'en servir comme estimateurs des densités de probabilité conditionnelles.

Pour générer ces histogrammes, ils subdivisent l'espace *RGB* en boîtes et comptent le pourcentage de pixels de peau appartenant à chaque boîte. Ils obtiennent ainsi un histogramme représentant $P(\mathbf{x}|\text{pixel de peau})$ où $\mathbf{x}$ est un vecteur contenant les trois coordonnées *RGB* d'un pixel. Ils obtiennent $P(\mathbf{x}|\text{pixel de non-peau})$ de manière similaire. La loi de probabilité recherchée est

$$P(\text{pixel de peau}|\mathbf{x})$$

qui est obtenue facilement par la loi de Bayes. Il reste maintenant à déterminer un paramètre θ permettant de choisir si un pixel correspond à de la peau ou pas de la manière suivante :

– Si $P(\text{peau}|\mathbf{x}) > \theta$, on a un pixel correspondant à de la peau.

– Si $P(\text{peau}|\mathbf{x}) < \theta$, le pixel ne correspond pas à de la peau.
– Si $P(\text{peau}|\mathbf{x}) = \theta$, choisir au hasard.

Ils entraînent leur classificateur sur une base de données d'images collectées sur internet de plus d'un milliard de pixels. Ils obtiennent une taux de détection de 80% avec un taux de fausses détections de 8.5%.

Ils comparent leur système avec le modèle basé sur une mixture de Gaussiennes décrit dans [17] en l'entraînant sur la même base de données. Les résultats obtenus par leur technique sont systématiquement meilleurs que ceux obtenus avec le système de Jebara et Pentland.

Comparaison

L'avantage du système de Pentland est de nécessiter une base de données d'entraînement plus restreinte que le système basé sur des histogrammes pour obtenir des résultats corrects. Son désavantage est, outre le fait que le système soit globalement moins performant, de nécessiter un temps de calcul beaucoup plus long d'une part lors de l'entraînement et d'autre part lors de l'utilisation. A moins de ne disposer que d'une base de données d'entraînement fort restreinte, l'emploi du système de Jones et Regh semble donc préférable à celui de Jebara et Pentland.

2.4.3 L'analyse en composantes principales

Lors de la conception d'un classificateur, une importante question est de savoir quelles caractéristiques vont être fournies à l'entrée de celui-ci. L'approche la plus simple consiste à lui fournir l'ensemble des pixels constitutifs de l'image à classifier. Le problème est qu'avec cette technique, les objets à classifier sont de très grandes dimensions. Ceci est un inconvénient majeur pour deux raisons : d'une part, cela augmente le temps de calcul et d'autre part cela élargit la taille de la base de données nécessaire pour représenter correctement l'ensemble des variations possibles dans la classe que l'on cherche à identifier.

L'analyse en composantes principales (*Principal Component Analysis, PCA*) est un outil permettant de déterminer, à partir d'un ensemble de caractéristiques, un nouvel ensemble de caractéristiques de dimensions plus réduites représentant au mieux la distribution de ces caractéristiques initiales autour de leurs moyennes. Ces nouvelles caractéristiques sont des fonctions linéaires des anciennes. Cette technique est également connue sous le nom de transformée de Karhunen-Loéve.

Description théorique de l'outil

Plaçons-nous dans un espace $\mathbb{R}^d$ et supposons que nous possédons un ensemble de n vecteurs $\mathbf{x}_i$ de celui-ci[4]. La moyenne de ceux-ci est $\boldsymbol{\mu}$ et leur matrice de covariance est $\boldsymbol{\Sigma}$. Le but est ici de déterminer une direction $\mathbf{v}$ sur laquelle projeter notre ensemble de vecteurs. Nous voulons que cette projection conserve un maximum de la variance de

[4]Cette section est largement inspirée par [13].

l'ensemble de vecteurs considéré. Pour ce faire, nous étudions les écarts de ceux-ci par rapport à leur moyenne. Si on appelle cette projection $c(\mathbf{x}_i)$, on a

$$c(\mathbf{x}_i) = \mathbf{v}^T (\mathbf{x_i} - \boldsymbol{\mu}).$$

Le but est alors de trouver le vecteur unitaire $\mathbf{v}$ qui maximise la variance de $c(\mathbf{x}_i)$ (dont la moyenne est nulle). Calculons celle-ci :

$$\begin{aligned} s\left(c(\mathbf{x}_i)\right) &= \frac{1}{n-1} \sum_{i=1}^{n} c(\mathbf{x}_i)\, c(\mathbf{x}_i)^T \\ &= \frac{1}{n-1} \sum_{i=1}^{n} \mathbf{v}^T (\mathbf{x_i} - \boldsymbol{\mu}) \left(\mathbf{v}^T (\mathbf{x_i} - \boldsymbol{\mu})\right)^T \\ &= \frac{1}{n-1} \mathbf{v}^T \left(\sum_{i=1}^{n} (\mathbf{x_i} - \boldsymbol{\mu}) (\mathbf{x_i} - \boldsymbol{\mu})^T \right) \mathbf{v} \\ &= \frac{1}{n-1} \mathbf{v} \Sigma \mathbf{v}^T. \end{aligned}$$

Il nous faut donc maximiser $\mathbf{v}\Sigma\mathbf{v}^T$ sous la contrainte $\mathbf{v}\mathbf{v}^T = 1$. C'est un problème aux valeurs et vecteurs propres. La solution est donnée par le vecteur propre de Σ correspondant à la plus grande valeur propre.

Les $k \leq n$ directions capturant au mieux la variance de l'ensemble de vecteurs initial sont donc les k vecteurs propres correspondant aux k plus grandes valeurs propres de la matrice de covariance Σ de ceux-ci. On les appelle *composantes principales*.

L'application typique de cet outil à la classification consiste à considérer que $\mathbb{R}^d$ est l'espace des caractéristiques initiales et que les k directions recherchées sont les nouvelles caractéristiques que nous enverrons à l'entrée du classificateur. L'ensemble de vecteurs $\mathbf{x_i}$ est constitué d'une base de données d'exemples d'objets appartenant une classe particulière.

La PCA nous permet donc de réduire la dimension de notre espace de caractéristiques tout en perdant un minimum de variance.

L'espace des visages propres (*eigenfaces*)

L'application la plus directe de l'analyse en composantes principales dans le cadre de la détection de visages consiste à construire, à partir d'une base de données de visages, ce que Turk et Pentland appellent dans [54] l'espace des visages propres (*eigenface space*). Il suffit ensuite de mesurer à quelle distance de cet espace se situe l'objet à tester pour avoir une estimation de la probabilité qu'a l'objet en question d'être un visage.

La construction de l'espace des visages propres est proposée pour la première fois par Sirovich et Kirby dans [47] (et plus tard dans [20]). Pour construire cet espace, on applique la technique décrite dans le paragraphe précédent en considérant que :

- L'espace $\mathbb{R}^d$ est celui des images en niveaux de gris de, par exemple, $d = 128^2$ pixels. Cet espace contient donc des vecteurs de $d = 128^2$ éléments obtenus en mettant bout à bout l'ensemble des lignes des images.

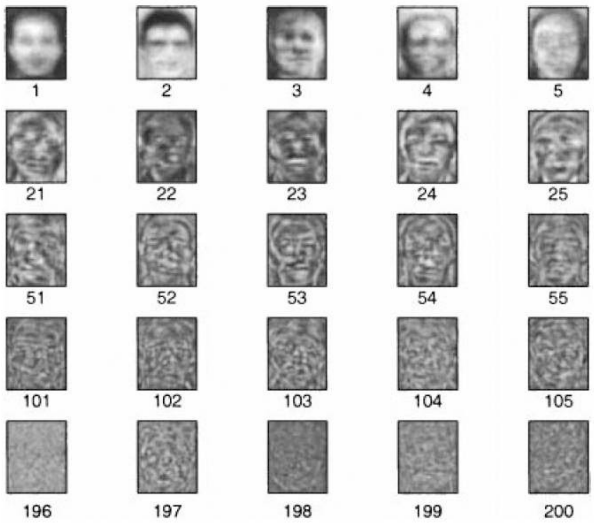

FIG. 2.18 – Quelques exemples de visages propres tirés de [14] : les numéros indiqués en dessous des images indiquent l'ordre de la valeur propre à laquelle ils correspondent.

- L'ensemble des n vecteurs $\mathbf{x}_i$ est constitué par la base de données de visages d'entraînement. La taille de ces visages est normalisée et on leur a appliqué une série de traitements de correction des variations d'illumination, de sensibilité de la caméra,... tels qu'une égalisation des histogrammes par exemple.
- Les $k < d$ vecteurs propres obtenus sont k visages propres tels que ceux de la figure 2.18.

Une fois l'espace construit, le principe du classificateur (décrit par Turk et Pentland dans [54], ainsi que par Liou dans [26]) consiste à :

1. Calculer la projection $\mathbf{y}$ de l'image à tester $\mathbf{x}$ sur l'espace des visages propres :

$$\mathbf{y} = \mathbf{P_M^T} (\mathbf{x} - \overline{\mathbf{x}})$$

où $\overline{\mathbf{x}}$ est la moyenne des images d'entraînement $\mathbf{x_i}$ et $\mathbf{P_M}$ la matrice de projection dont les colonnes sont les visages propres constituant les vecteurs de base du sous-espace linéaire sur lequel les images sont projetées.

2. On reconstruit ensuite l'image à partir des k visages propres :

$$\widehat{\mathbf{x}} = \mathbf{P_M}\, \mathbf{y} + \overline{\mathbf{x}}.$$

3. Il reste à calculer l'erreur de reconstruction ϵ^2, aussi appelée éloignement de l'espace des visages (*Distance From Face Space, DFFS*) :

$$\epsilon^2 = ||(\mathbf{x} - \widehat{\mathbf{x}})||.$$

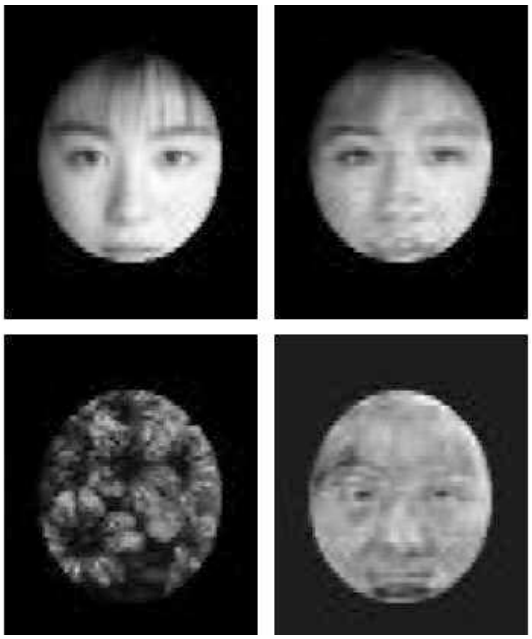

FIG. 2.19 – Sur cette figure tirée de [26], les images de départ sont à gauche et les images reconstruites à droite : on constate que l'image reconstruite diffère peu de l'image de départ si celle-ci est un visage et inversément.

4. Un simple seuillage sur la DFFS permet de décider si l'image à tester est un visage ou pas. En effet, la différence entre l'image de départ et l'image reconstuite est très petite si l'image est un visage et est, en général, grande si elle n'en est pas un, comme nous pouvons le voir sur la figure 2.19.

Sirovich et Kirby montrent dans [47] que, pour une erreur de reconstruction déterminée, le nombre de visages propres à considérer est moindre dans le cas de visages découpés ne présentant que les yeux, le nez et la bouche que dans le cas d'un visage complet incluant les cheveux et un arrière-plan. Cette constatation n'a rien d'étonnant puisque les cheveux et l'arrière-plan sont des éléments présentant une grande variance, il est donc normal que le sous-espace les contenant soit plus grand.

L'espace des traits propres (*eigenfeatures*)

Plutôt que de rechercher des visages entiers, peut-être est-il plus efficace de ne rechercher que des morceaux du visage tels que les yeux, la bouche ou le nez. C'est ce que proposent Pentland et Moghaddam dans [37] (voir figure 2.20). Ils rapportent un taux de détection des yeux allant jusqu'à 94%. Ils comparent également les performances de détecteurs basés sur la recherche de visages, sur la recherche de traits de visages ou sur les deux. Le détecteur basé sur la recherche de traits est toujours meilleur que celui basé sur la recherche de visages entiers. Ceci est probablement dû au fait qu'il est d'autant plus facile de capturer la variabilité d'un motif, au moyen d'une base de données d'entraînement, que

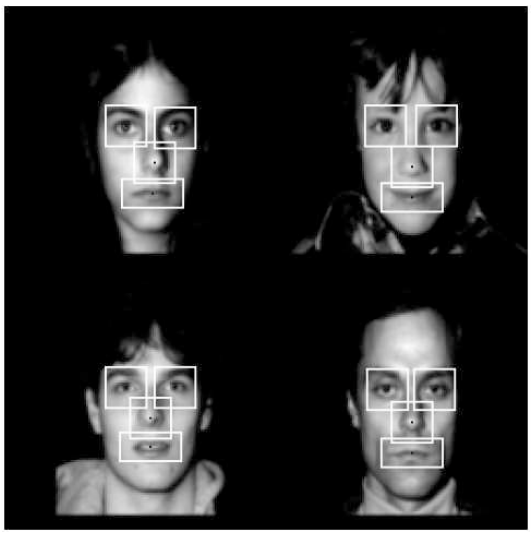

FIG. 2.20 – Exemples de détection des yeux, du nez et de la bouche tiré de [37].

ce motif est petit. Cependant, la technique obtenant les meilleures performances est celle qui consiste à combiner les deux approches. Spors et Rabenstein proposent dans [50] un système de suivi de visages basé sur une combinaison de détection de peau et de détection des yeux par une PCA.

Dans [12], De Campos compare les performances de systèmes de reconnaissance de visages basés d'une part sur une recherche de visages entiers et d'autre part sur une recherche des yeux et ce pour différents nombres k de composantes principales retenues. Le taux de reconnaissance est systématiquement meilleur pour le système basé sur la recherche des yeux si le nombre de composantes principales est plus grand que 4. En dessous de 4 composantes retenues, c'est le système basé sur la recherche de visages entiers qui l'emporte. Cependant, un bon taux de détection nécessite un nombre de composantes au moins égal à 10. D'après eux, on aurait donc systématiquement intérêt à rechercher des yeux plutôt que des visages. Ils constatent également une saturation des performances pour 15 composantes retenues et plus. Ils expliquent les meilleures performances du système basé sur les yeux par le fait qu'un tel système s'affranchit de la variabilité supplémentaire introduite par les différentes expressions du visage. De plus, comme déjà mentionné, moins le motif recherché est complexe, moins grand est le sous-espace linéaire capturant une grande partie de sa variance. On peut donc le caractériser avec un nombre de vecteurs propres plus restreint obtenus à partir d'une base de données d'entraînement moins étendue. De plus, les motifs étant plus petits, le temps de calcul nécessaire pour leur recherche est moindre. Il faut cependant garder à l'esprit qu'une recherche d'yeux nécessite que les sujets soient relativement proches de la caméra afin que leurs yeux soient suffisamment détaillés pour permettre une détection correcte.

Combinaison avec d'autres types de classificateurs

Une analyse en composantes principales peut être combinée avec d'autres types de classificateurs. On pourra par exemple l'utiliser en amont afin d'extraire un ensemble pertinent de caractéristiques à présenter à l'entrée du classificateur. C'est ce que font Popovici et Thiran dans [38]. Ils présentent à l'entrée d'une machine à vecteurs de support (*Support Vector Machine, SVM*), qui est le prochain type de classificateurs auxquels nous allons nous intéresser, la projection de l'image à tester sur l'espace des visages propres. L'utilisation d'un tel classificateur en lieu et place de la classique DFFS permet une modélisation plus fine de la frontière de décision à l'intérieur du sous-espace linéaire. Ils comparent les deux systèmes et obtiennent, comme attendu, de bien meilleurs résultats lors de l'utilisation de la machine à vecteurs de support : pour un espace formé de 36 visages propres, le taux de détection passe de 77% pour la DFFS à 97% en utilisant une SVM. Ils testent également leur système pour toute une série de nombres de visages propres. Ils constatent que s'il n'en faut pas un très grand nombre pour obtenir des performances acceptables (une quinzaine), il faut ensuite fortement augmenter celui-ci pour obtenir une amélioration significative des performances. Ils constatent même parfois, pour des nombres déjà relativement élevés de visages propres, des diminutions de celles-ci lorsqu'ils passent à des nombres plus élevés.

Méthodes apparentées

L'analyse en composantes principales est une méthode relativement intuitive et efficace pour construire un sous-espace représentant une famille d'objets. En revanche, elle n'est pas nécessairement optimale.

Plutôt que de chercher à construire un sous-espace monolithique, un autre type d'approche consiste à considérer que l'espace des visages pourrait être plus fidèlement représenté en le divisant en une série de sous-espaces et en représentant chacun de ceux-ci par un modèle approprié. Plusieurs techniques ont été proposées. La plupart de celles-ci utilisent une combinaison de distributions Gaussiennes à plusieurs dimensions pour le représenter. La première, la plus célèbre, est celle proposée par Sung et Poggio dans [51].

Sung et Poggio modélisent l'espace des images de visages de 19 sur 19 pixels par 12 distributions Gaussiennes multi-dimensionnelles caractérisées par une moyenne et une matrice de covariance. Six Gaussiennes représentent l'espace des visages. Les six distributions restantes servent à représenter l'espace des "non-visages" pouvant facilement être confondus avec des visages. Ils permettent ainsi de réduire le nombre de fausses détections. Ils servent en fait à creuser des concavités dans les distributions Gaussiennes qui sont de formes ellipsoïdales dans l'espace à 19^2 dimensions des images présentées à l'entrée du classificateur. Une illustration de l'intérêt d'une telle approche est donnée à la figure 2.21. L'algorithme générant ces distributions est basé sur une version modifiée de la *distance de Mahalanobis*.

Lorsqu'on présente une image à classifier au système, celui-ci commence par mesurer deux distances par rapport à chacune des 12 distributions. La première est une distance de Mahalanobis modifiée à l'intérieur du sous-espace des 75 plus grands vecteurs propres de

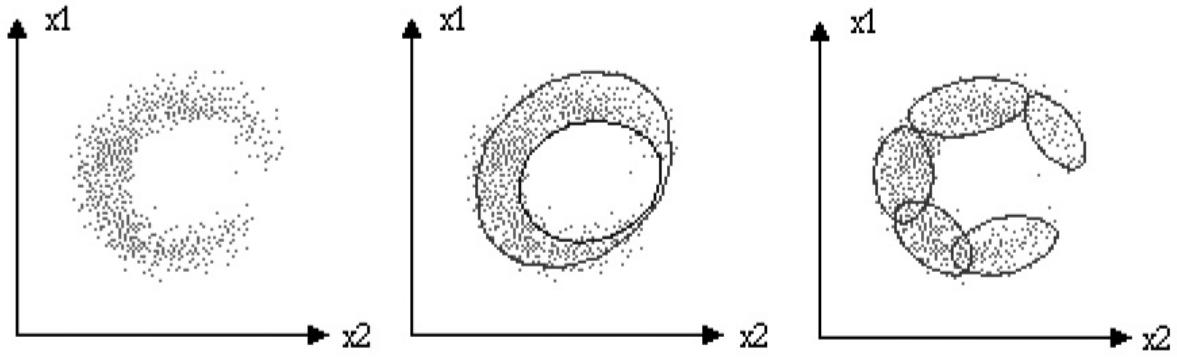

FIG. 2.21 – Illustration de l'intérêt de l'utilisation de distributions "négatives" pour modéliser des distributions concaves [51]. A **gauche** on trouve la distribution à modéliser. Au **centre** se trouve une modélisation par une Gaussienne positive entourant la distribution et une Gaussienne négative servant à creuser une concavité dans celle-ci. A **droite**, nous voyons que la modélisation par des distributions exclusivement positives entraîne l'utilisation d'un plus grand nombre de celles-ci.

la matrice de covariance de la distribution considérée entre l'image $\mathbf{x}$ et la moyenne $\overline{\mathbf{x}}$ de la distribution considérée (on retrouve donc ici quelque chose de fort proche de la DFFS classique de l'analyse en composantes principales). La seconde est la distance Euclidienne entre l'image et sa projection sur ce sous-espace. On génère ainsi pour cette image un vecteur de 24 caractéristiques que l'on présente à l'entrée d'un réseau de neurones (les réseaux de neurones artificiels sont abordés plus loin et décrits dans l'annexe C) dont la sortie indique si finalement l'image est un visage ou pas.

Yang, Abuja et Kriegman proposent dans [31] deux autres techniques de modélisation de l'espace des visages utilisant une combinaison de distributions Gaussiennes.

La première est basée sur une technique intitulée *Factor Analysis (FA)* relativement similaire à l'analyse en composantes principales mais utilisant un modèle différent. Le vecteur $\mathbf{x}$ de dimension d est ici représenté par un vecteur $\mathbf{z}$ de dimension $k < d$ en considérant que $\mathbf{x}$ obéit au modèle suivant :

$$\mathbf{x} = \mathbf{\Lambda}\,\mathbf{z} + \mathbf{u} + \boldsymbol{\mu}$$

où $\mathbf{\Lambda}$ est une matrice dont les colonnes jouent le même rôle que les composantes principales, $\mathbf{z}$ est un vecteur répondant à une distribution $\mathcal{N}(0, \mathbf{I})$, $\mathbf{u}$ est un vecteur obéissant à $\mathcal{N}(0, \Psi)$ où Ψ est une matrice diagonale et $\boldsymbol{\mu}$ un vecteur contenant les moyennes des composantes de $\mathbf{x}$. On peut retrouver le modèle utilisé dans l'analyse classique en composantes principales en forçant $\mathbf{u}$ à zéro. Dans ce modèle, $\mathbf{z}$ modélise les corrélations entre les éléments de $\mathbf{x}$ tandis que $\mathbf{u}$ capture les variations dues au bruit dans les éléments de $\mathbf{x}$. C'est un avantage de ce modèle sur l'analyse en composantes principales puisque celle-ci, en voulant garder le maximum de variance, est susceptible de capturer les variations néfastes dues au bruit. La technique de détection de visages proposée utilise une combinaison de

tels modèles pour construire l'expression mathématique donnant la probabilité $P(\mathbf{x})$ de rencontrer le vecteur $\mathbf{x}$ à l'intérieur du sous-espace des visages. Un simple seuillage sur les valeurs de $P(\mathbf{x})$ permet de déterminer quelles fenêtres correspondent à des visages.

La seconde technique décrite dans [31] consiste à diviser la base de données d'entraînement en 25 classes de visages et 25 classes de non-visages[5]. On utilise ensuite la base de données pour générer une matrice de projection sur un espace de dimension $c-1$, où c est le nombre de classes présentes dans la base de données (ici 50). Cette matrice de transformation est déterminée en cherchant à maximiser le rapport entre la dispersion entre les classes et la dispersion à l'intérieur des classes plutôt qu'en cherchant à représenter au mieux l'intégralité de la base de données de visages comme dans l'analyse en composantes principales. Cette technique est appelée *Linear Discriminant Analysis (LDA)* et les vecteurs propres de la matrice de projection correspondant aux plus grandes valeurs propres sont appelés les *Fisher's Liner Discriminant*. Les éléments de la base de données d'entraînement sont donc projetés sur un espace à 49 dimensions. On se sert ensuite de leurs projections z pour construire à l'intérieur de cet espace, un modèle Gaussien $P(z|X_i)$ pour chacune des 50 classes X_i. La règle de décision permettant de déterminer si une image est un visage ou pas est basée sur le principe du maximum de vraisemblance

$$X^* = \arg\max_{X_i} P(z|X_i).$$

Les résultats obtenus par les deux techniques sont systématiquement meilleurs que ceux obtenus avec une simple analyse en composantes principales.

Une dernière technique apparentée dont nous parlerons est la *Kernel Principal Component Analysis, (KPCA)* décrite dans [25]. On y considère que l'espace des visages vus sous différentes illuminations et différents points de vue est trop complexe pour être représenté par un simple sous-espace linéaire. La technique proposée est de commencer par projeter les données sur un espace non-linéaire avant d'y effectuer une analyse en composantes principales. On espère que dans ce sous-espace non-linéaire, les données prendront une configuration simple permettant une bonne analyse en composantes principales. Si c'est bien le cas, celle-ci y donnera de bons résultats, meilleurs que si elle avait été effectuée à l'intérieur de l'espace initial. Ceci est confirmé par leurs résultats.

2.4.4 Les machines à vecteurs de support

Les machines à vecteurs de support (*Support Vector Machines, SVM*) sont, comme nous l'avons déjà mentionné, des classificateurs appartenant à la seconde catégorie. L'objectif est donc ici de construire directement la frontière de décision à l'intérieur de l'espace des caractéristiques. On cherche plus particulièrement l'hyper-plan séparant au mieux les instances négatives des instances positives (les SVM sont des classificateurs binaires).

Une fois la surface de décision générée, la règle de décision pour un point à classifier consiste tout simplement à regarder de quel côté de celle-ci se trouve le point.

[5]Yang et ses associés disent utiliser les 'Self Organizing Map' de Kohonen décrites dans [21] pour réaliser cette opération.

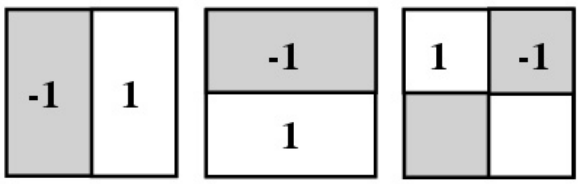

FIG. 2.22 – Les trois types d'ondelettes de Haar utilisés dans [34, 36].

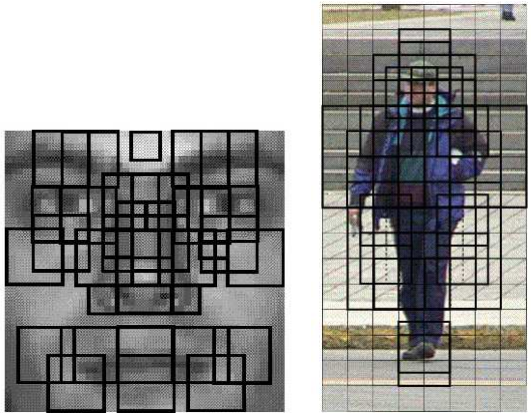

FIG. 2.23 – Emplacements des coefficients retenus pour la détection de visages et de piétons [36].

Une description théorique des SVM's est faite dans l'annexe B.

Applications

Les premiers à avoir proposé l'utilisation de SVM pour la détection de visages sont Papageorgiou, Oren et Poggio dans [36].

Ils proposent de générer un ensemble de caractéristiques à partir d'une base de données de visages de 19 par 19 pixels en utilisant une transformée en ondelettes de Haar sur-complète (quadruple densité, telle que décrite dans [34]) et de sélectionner les coefficients de plus grandes valeurs absolues. Ces coefficients sont ceux qui encodent les frontières entre deux régions. Ils obtiennent ainsi un ensemble de 37 caractéristiques par visage.

Pour la détection de piétons, la même procédure est appliquée sur des images de piétons de 128×64 pixels. Cette fois-ci, 29 coefficients sont retenus.

Une fois les coefficients à considérer choisis, ils entraînent une SVM sur la base de données, en prenant soin de faire du bootstrapping pour diminuer le nombre de fausses détections.

Pour la détection de visages, en permettant une fausse détection toutes les 7500 fenêtres testées, ils obtiennent un taux de détection de 75%. Une fausse détection toutes les 15000 fenêtres permet d'obtenir, pour la détection de piétons, un taux de détection de

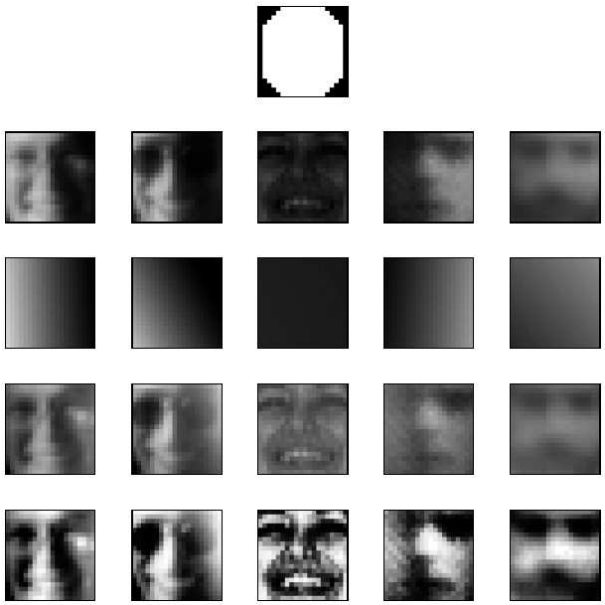

FIG. 2.24 – Traitements appliqués à chaque fenêtre de 19×19 pixels. Du haut vers le bas : masque binaire, image originale, meilleur plan de luminosité, image dont on a soustrait ce plan, image finale après égalisation des histogrammes. [41]

70%.

Ils proposent ensuite une amélioration du système de détection de piétons qui se sert du mouvement de ceux-ci pour définir des zones d'intérêt à l'intérieur desquelles on peut permettre au classificateur de détecter les piétons de manière moins restrictive. Les performances du système passent de 43.5% de détection pour une fausse détection toutes les 235.000 fenêtres à 53.8% de détection tout en maintenant un nombre de fausses détections relativement bas (une toute les 90.000 fenêtres).

Osuna, Freund et Girosi sont les suivants à proposer l'utilisation d'une SVM pour la recherche de visages [35]. Cependant, ils n'utilisent pas de transformée en ondelettes. Ils utilisent comme caractéristiques directement les pixels de l'image. Ils considèrent des images en niveaux de gris de 19×19 pixels. Avant d'être employée, chaque fenêtre subit les traitements, inspirés de [51], suivants (voir figure 2.24) :

- **Masquage** : un masque binaire est appliqué afin d'éviter de considérer trop de pixels de l'arrière-plan et d'introduire ainsi trop de bruit dans les données.
- **Correction du gradient d'illumination** : le plan de luminosité collant le mieux à l'image est soustrait de celle-ci afin de réduire les variations d'illumination à l'intérieur de la fenêtre dues à l'éclairage et aux ombres.
- **Égalisation des histogrammes** : afin de réduire les différences d'illumination, de calibration de la caméra, ... entre les différentes images.

Le masquage permet de réduire le nombre de caractéristiques de $19^2 = 361$ à 283.

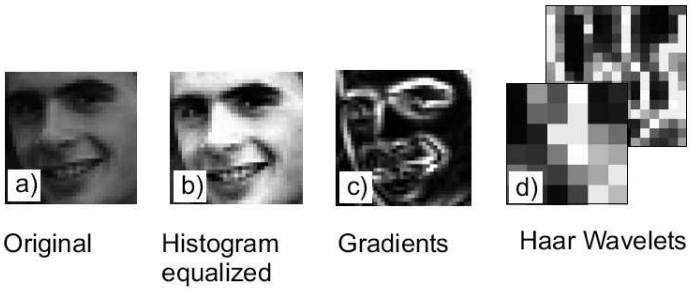

FIG. 2.25 – Les trois types de caractéristiques utilisés par Poggio dans [1].

Ce nombre de caractéristiques reste quand même très important si on le compare aux 37 et 29 utilisées dans le système précédent. Ce système nécessite donc une plus grande base de données d'entraînement et est plus coûteux en calcul. Ses performances sont heureusement bien supérieures à celles de la technique de Papageorgiou. Testé sur deux ensembles d'images, il obtient tantôt un taux de détection de 97,1% avec une fausse détection par million de fenêtres inspectées, tantôt 74,2% de détections correctes (les images du deuxième ensemble sont de moins bonne qualité) avec une fausse détection environ toutes les 250.000 fenêtres considérées. Ce qui est dans les deux cas supérieur au système précédent et de beaucoup si on ne considère que l'ensemble de données comportant des images de bonne qualité. Ce système obtient des performances similaires au système de Sung et Poggio [51] déjà décrit tout en étant près de trente fois plus rapide.

On voit ici qu'un classificateur employant quasi directement (après quelques traitements) les pixels de l'image donne de meilleurs résultats qu'un classificateur basé sur une transformée en ondelettes. Cependant, tous les coefficients de la transformée n'étaient pas retenus dans la méthode décrite et aussi bien la base de données d'entraînement que le type particulier de SVM choisi n'étaient pas identiques dans les deux techniques. Dans [1], Poggio, Heisele et Pontil comparent les performances de trois systèmes exploitant une même SVM à noyau quadratique et une même base de données mais trois types différents de caractéristiques (voir figure 2.25) :

1. Des fenêtres de 19×19 pixels en niveaux de gris sur lesquelles sont appliqués les trois traitements décrits ci-dessus.
2. Les gradients de Sobel de ces mêmes fenêtres.
3. Les transformées en ondelettes de Haar de celles-ci.

Les *Receiver Operating Curves (ROC)*[6] des trois systèmes sont reprises à la figure 2.26. Nous y constatons que le système basé sur les images en niveaux de gris est toujours le meilleur. Il est suivi par celui exploitant la transformée en ondelettes de Haar. C'est le système basé sur les gradients qui ferme la marche.

[6]Les ROC nous montrent le taux de détections correctes en fonction du taux de fausses détections. Un classificateur idéal aurait une aire unitaire sous sa ROC.

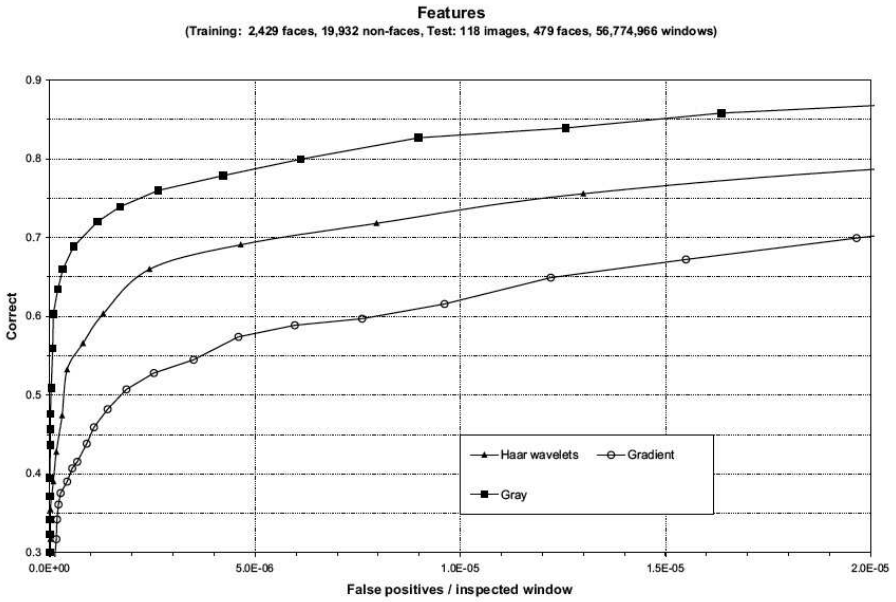

FIG. 2.26 – ROC des trois systèmes comparés par Poggio dans [1].

Ils cherchent ensuite à réduire le nombre de caractéristiques envoyées au classificateur sans pour autant réduire les performances. Ils proposent deux techniques.

La première est l'analyse en composantes principales déjà largement évoquée. La seconde est appelée Classification Linéaire Itérative *(Iterative Linear Classification, ILC)*. Pour une base de données d'entraînement, elle consiste à :

1. Déterminer la direction séparant le mieux les données au moyen d'un classificateur linéaire (une SVM linéaire par exemple).
2. Projeter les données sur le sous-espace orthogonal à cette direction et recommencer au point 1.

Les N premières directions ainsi déterminées forment le nouvel espace de caractéristiques sur lequel projeter l'ancien.

Ne retenir que trois directions donne de meilleurs résultats si on emploie l'ILC (résultats bien entendu largement inférieurs à ceux obtenus lors de l'emploi de l'entièreté des caractéristiques). En revanche, l'augmentation du nombre de directions retenues n'accroît quasi pas les performances. Par contre, une telle augmentation améliore sensiblement les performances d'un système utilisant l'analyse en composantes principales. L'emploi des 20 premières composantes principales donne même parfois de meilleurs résultats que l'emploi de l'entièreté des caractéristiques.

Ce système possède des performances tout à fait honorables. Cependant, il est très peu robuste face à des changements d'orientation du visage : aussi bien face à des rotations dans le plan de l'image que face à des visages pris plus ou moins de profil. Pour tenter de remédier à ces problèmes, ils proposent un nouveau système basé sur non pas une mais plusieurs SVM's (voir figure 2.27). Trois premières SVM's sont chargées de détecter

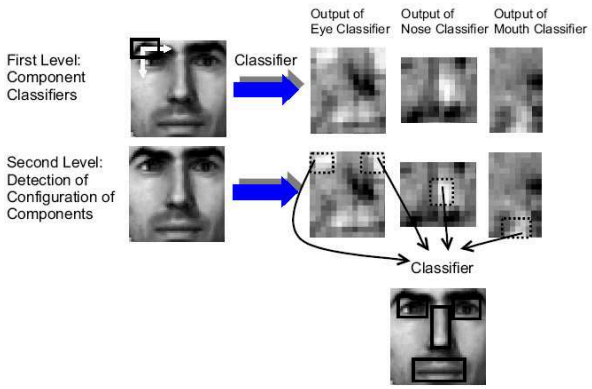

FIG. 2.27 – Vue d'ensemble du système décrit dans [1] basé sur une détection séparée des yeux, du nez et de la bouche. Un classificateur géométrique vérifie ensuite la bonne configuration des différents éléments détectés.

séparément les yeux, le nez et la bouche tandis qu'une quatrième est chargée de vérifier que ceux-ci se trouvent dans une configuration géométrique acceptable. Cette approche est motivée par le fait que la forme des yeux, du nez et de la bouche est moins affectée par des rotations que ne l'est celle du visage entier. On peut également détecter indépendamment des visages entiers et les considérer comme quatrième caractéristique présentée en entrée du classificateur géométrique.

Dans le cas de la détection de visages vus de face, c'est le système basé sur la combinaison yeux, nez, bouche, visage qui l'emporte. Le système uniquement basé sur les yeux, le nez et la bouche est le moins performant, à moins d'autoriser un nombre relativement élevé de fausses détections. Par contre, celui-ci est bien plus robuste que les deux autres face à des rotations des visages dans le plan et détecte également mieux les visages vus plus ou moins de profil. Il serait donc finalement le système le plus adapté à une utilisation en conditions réelles bien qu'il présente de relativement moins bonnes performances pour la détection de visages parfaitement vus de face.

Papageorgiou et Poggio (en collaboration avec Mohan) proposent dans [32] une amélioration de leur système de détection de piétons (basé sur une transformée en ondelettes de Haar et une SVM) utilisant le même principe : ils recherchent indépendamment des visages, des jambes, des bras gauches et des bras droits. Un classificateur géométrique sert ensuite à vérifier si ceux-ci respectent un agencement crédible. Ce dernier classificateur est conçu de manière à permettre la détection correcte de personnes en partie cachées (voir figure 2.28). Ils proposent plusieurs techniques permettant de combiner les résultats obtenus par les détecteurs de parties de corps. Chacune d'entre elles obtient des résultats supérieurs à ceux d'un système détectant des personnes entières directement.

On peut trouver dans [25, 38] des techniques utilisant des SVM dans l'espace des visages propres. Les SVM's y permettent une modélisation plus fine de la frontière de

FIG. 2.28 – Illustration de la robustesse du système décrit dans [32] face à des personnes dont certaines parties du corps sont occultées (images de gauche et de droite) ou dont certaines parties du corps ne sont pas détectées parce qu'elles ressortent trop peu de l'arrière-plan (image du milieu).

décision que la classique DFFS. Comme attendu, les résultats obtenus par ces systèmes sont meilleurs que ceux exploitant la DFFS (20% de détections correctes en plus dans [38] pour un même nombre de visages propres).

Buciu et Pitas proposent, dans [5], de combiner des SVM's entraînées sur la même base de données mais utilisant des noyaux différents. Chacune des SVM's vote indépendamment et la classe finalement retenue est celle obtenant le plus grand suffrage. Ceci permet d'améliorer quelque peu les performances en diminuant le nombre de fausses détections. Ils tentent également d'améliorer les performances en faisant ce qu'ils appellent du *bagging*. Ceci consiste à, au départ d'une base de données d'entraînement, en créer un nombre arbitraire en remplaçant certains de ses éléments. On entraîne ensuite une SVM sur chacune de ces nouvelles bases de données. La classification se fait ensuite de nouveau sur base d'un vote à la majorité. Cette technique ne permet pas d'améliorer les performances du classificateur.

2.4.5 Les réseaux de neurones

Jusqu'à présent, nous avons vu que nous pouvions tenter d'approximer la densité de probabilités qu'a un objet d'appartenir à une classe en utilisant une base de données d'entraînement soit par un histogramme, soit en considérant que cette densité de probabilités est une instance d'un modèle paramétrique connu dont nous n'avons qu'à estimer les différents paramètres. Il n'est malheureusement pas toujours possible d'obtenir un modèle efficace par l'un de ces deux moyens. Dans ce cas, il faut soit recourir à une technique permettant de modéliser directement la frontière de décision telle que les SVM's décrites ci-dessus, soit utiliser des moyens plus sophistiqués pour modéliser la densité de probabilités tels que les réseaux de neurones artificiels (décrits en annexe C).

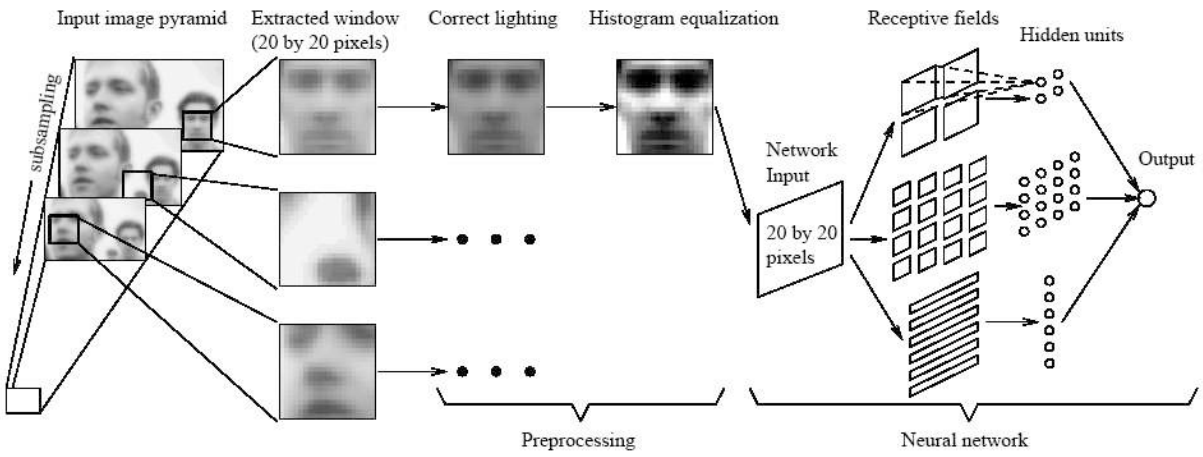

FIG. 2.29 – Réseau de neurones utilisé pour la détection de visages vus de face par Rowley dans [41, 42].

Détection de visages par un réseau de neurones

Les travaux les plus significatifs dans ce domaine sont ceux de Rowley, Baluja et Kanade. Ils ont été les premiers à proposer une technique de détection de visages basée sur l'utilisation d'un réseau de neurones artificiels.

Ils décrivent dans [41, 42] leur technique permettant de détecter des visages vus de face. Leur détecteur considère toutes les fenêtres de 20×20 pixels. La première étape consiste à leur appliquer les trois prétraitements classiques (voir section 2.4.4). Les pixels de la fenêtre sont ensuite fournis directement au réseau de neurones représenté sur la figure 2.29. Il comporte 3 couches cachées parallèles de types différents. La première et la seconde inspectent respectivement 4 zones de 10×10 pixels et 16 zones de 5×5 pixels. Leur rôle est de détecter les divers traits du visage tels que chacun des deux yeux, le nez ou les coins de la bouche. La troisième couche s'intéresse à des zones horizontales de 20×5 pixels qui se recouvrent. Elle s'occupe de détecter la bouche et les yeux par paires. Le réseau ne possède qu'une sortie à valeur réelle dont la valeur indique si la fenêtre considérée est un visage ou non. Il est entraîné pour que celle-ci ait la valeur 1 si la fenêtre est un visage et -1 si non. La règle de décision la plus évidente est donc de considérer que nous avons un visage si la valeur de celle-ci est positive. Une autre possibilité est de se servir de la valeur de seuil afin d'ajuster le compromis entre les taux de bonnes et de fausses détections.

Afin de limiter le nombre de fausses détections, ils proposent d'entraîner plusieurs réseaux identiques sur la même base de données mais avec des distributions initiales de poids attribués aux neurones différentes. En effet, les techniques classiques de minimisation de l'erreur convergent vers un minimum *local* de celle-ci. Plusieurs entraînements à partir de distributions initiales différentes sont donc susceptibles de converger vers des minima locaux différents et se comporteront donc de manière différente lors de leur utilisation. De plus, il a été constaté que ces réseaux ont tendance à s'accorder sur les détections positives

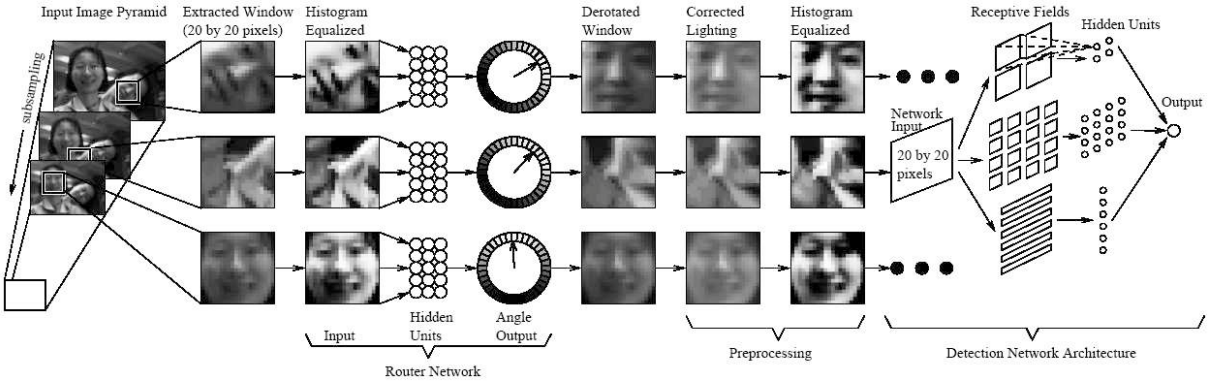

FIG. 2.30 – Réseau de neurones utilisé pour la détection de visages avec rotation dans le plan de l'image par Rowley dans [40].

de fenêtres représentant effectivement un visage. Par contre, si l'un des réseaux fait une fausse détection, il est rare que le second la fasse également. Un simple "AND" logique ou un vote à la majorité peuvent donc conduire à une forte réduction du nombre de fausses détections sans trop diminuer le taux de détections correctes. Ils ont également pensé à entraîner un second type de réseau de neurones dont le rôle est de faire l'arbitrage entre les différents réseaux de détection. Les résultats obtenus avec cette dernière technique sont bons mais ne sont pas de beaucoup meilleurs que ceux obtenus par les techniques plus simples, et surtout moins coûteuses en calculs, décrites ci-dessus.

Les performances obtenues par ce système sont, dans le cas de visages vus de face, excellentes. Le meilleur résultat obtenu lors des tests est un taux de détection de 95,1% pour un nombre de fausses détections très faible (1/51.368.003). Cependant, ses performances s'écroulent lorsqu'on lui présente des visages dont la rotation dans le plan de l'image dépasse 10°.

Rowley, Baluja et Kanade proposent dans [40] un raffinement de la méthode développée par Baluja dans [2] permettant de rendre robuste à des rotations dans le plan de l'image leur système de détection de visages.

Leur système consiste à reprendre le réseau de neurones décrit précédemment mais de lui fournir en entrée la sortie d'un deuxième réseau dont le rôle est de remettre *droites* les images de visages (voir figure 2.30).

Ce réseau de "dé-rotation" prend en entrée les pixels de l'image pré-traités. Il possède 36 neurones de sortie. Chacun d'entre eux représente une tranche de 10°. Le neurone i représente donc un angle de $i * 10°$. Ce réseau est entraîné pour que, si on lui présente un visage orienté sous un angle θ, les valeurs de ses sorties soient $\cos(\theta - (i * 10°))$. Lorsqu'on lui présente un nouveau visage, la valeur de l'angle sous lequel celui-ci est tourné est donné par celui de la direction du vecteur

$$\left(\sum_{i=0}^{35} \text{sortie}_i * \cos(i * 10°), \sum_{i=0}^{35} \text{sortie}_i * \sin(i * 10°) \right).$$

Notons que cette "dé-rotation" est appliquée à toutes les fenêtres de 20×20 pixels de l'image, qu'elles contiennent un visage ou pas. Celles qui ne contiennent pas de visage sont donc présentées au réseau de détection après application d'une rotation d'angle aléatoire. Ceci peut expliquer le fait que le nombre de fausses détections de ce système est plus important que celui du système précédent. En effet, le système de détection est entraîné sur des images présentées selon leur orientation initiale et, lors de son utilisation, il inspecte des images auxquelles une rotation, incohérente pour les fenêtres sans visage, a été appliquée. Un moyen de remédier à ce problème est d'employer la technique maintenant bien connue du bootstrapping. En effet, celui-ci ajoutera à la base de données d'entraînement des images collectées lors de l'utilisation et donc auxquelles a été appliquée une rotation.

De nouveau, les performances sont au rendez-vous. Si ce système présente de légèrement moins bonnes performances pour la détection de visages droits que le précédent, il présente une très bonne robustesse face à des rotations dans le plan de l'image. Il arrive à conserver un taux de détections correctes d'environ 80% pour des visages présentés sous n'importe quelle orientation tout en gardant un taux de fausses détections suffisamment bas.

On peut trouver dans [64] un système de détection de piétons utilisant de manière similaire un réseau de neurones. Cependant, celui-ci prend en entrée non pas les pixels de l'image mais une image dont les niveaux de gris sont proportionnels à l'intensité du gradient de celle-ci. Ceci présente l'avantage d'être robuste face aux changements d'illumination. De plus, par rapport aux bords de l'image, les intensités du gradient ont l'avantage de ne pas dépendre d'une valeur de seuil ajustée à la main. Le taux de détections correctes est de 85% avec un taux de fausses détections de 3%.

Sahbi et Boujeema décrivent dans [43] une toute autre manière d'utiliser un réseau de neurones dans le cadre de la détection de visages. En effet, ils entraînent leur réseau afin de lui faire détecter les pixels qui sont de la couleur de la peau pour de définir des zones d'intérêt où chercher des visages. Mais le réseau de neurones n'est qu'une partie du système chargé de détecter les pixels de couleur peau. Celui-ci ne sert en fait qu'à définir des premières zones de couleur peau. Une fois ces zones définies, le système se sert de celles-ci pour construire un modèle Gaussien définissant la probabilité qu'a un pixel d'être de la couleur de la peau quand sa couleur est connue. L'image est ensuite réinspectée avec ce classificateur qui lui est spécifique en afin de détecter plus précisément les pixels de la couleur de la peau. Le détecteur s'affranchit ainsi des problèmes liés à la différence entre les images employées lors de l'entraînement et celles qui sont effectivement présentées au système lors de son utilisation. Les résultats obtenus attestent de l'intérêt de l'utilisation d'un modèle local à chaque image pour la détection de la peau (voir figure 2.31).

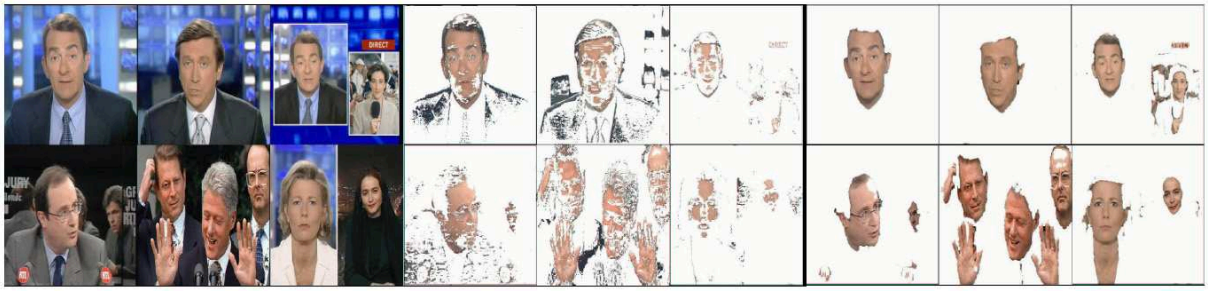

FIG. 2.31 – Détection de la peau en deux étapes telle que décrite dans [43]. On trouve à gauche l'image originale, au centre le résultat d'une première segmentation par un réseau de neurones et à droite le résultat de la classification effectuée sur base d'un modèle Gaussien construit à partir de cette première segmentation.

2.4.6 Autres approches statistiques

On trouve dans [56] une technique basée sur la théorie de l'information et en particulier sur l'information mutuelle de plusieurs variables aléatoires

$$I(X_1; X_2; \ldots; X_N) = \int_{x_1} \ldots \int_{x_n} f(X_1; X_2; \ldots; X_N) \ \log \frac{f(X_1; X_2; \ldots; X_N)}{f(X_1) f(X_2) \ldots f(X_N)}.$$

L'objectif est de construire les distributions de probabilités a posteriori

$$P(\text{visage}|X_1, \ldots, X_N)$$

et

$$P(\text{non-visage}|X_1, \ldots, X_N)$$

où les $X_1, \ldots, X_N$ sont les pixels d'un fenêtre de 12×12 pixels de l'image. Nous souhaitons ici ne pas faire appel à des modèles paramétriques tels que les distributions Gaussiennes. Afin de rendre l'estimation de ces distributions simple, on pourrait considérer que les pixels de l'image sont indépendants. Il serait alors facile d'estimer ces distributions à partir d'histogrammes générés grâce à une base de données d'entraînement. Cependant, il semble plus naturel de considérer qu'il existe des relations liant les pixels du visage entre eux. On pourrait alors considérer que la valeur de chaque pixel dépend des valeurs de l'ensemble des autres pixels mais nous aurions alors affaire à un modèle très complexe, donc coûteux en temps de calcul et nécessitant une énorme base de données d'entraînement. Une approche intermédiaire consiste à représenter la fenêtre de 12×12 pixels par un arbre et considérer que chaque fils est indépendant des autres variables lorsque la valeur de son père est connue. Un arbre est donc défini par un ensemble de N noeuds, $N-1$ densités de probabilités conditionnelles $f(c_i|p_i)$ (où les p_i sont les pères et les c_i les fils) et la densité de probabilités du noeud racine $f(r)$. On a alors

$$P(\text{visage}|X_1, \ldots, X_N) = f(r) * \prod_{i=1}^{N-1} f(c_i|p_i).$$

Cette densité de probabilités peut être estimée au moyen d'un histogramme et de $N-1$ histogrammes conjoints.

Nous n'avons cependant pas encore précisé comment choisir l'arbre qui va modéliser nos deux distributions. C'est ici que nous allons utiliser l'information mutuelle

$$I(X_1; X_2; \ldots; X_N) = \sum_{i=1}^{N-1} I(c_i; p_i).$$

En effet, l'arbre qui maximise celle-ci est l'arbre qui capture au mieux les relations liant les différents pixels. C'est donc cet arbre qui maximise l'information mutuelle que nous allons utiliser.

On utilise ensuite les *Fisher linear discriminants* afin de projeter chaque paire (c_i, p_i) sur un espace unidimensionnel. L'estimation de

$$P(\text{visage}|X_1, \ldots, X_N)$$

est donc ramenée à la construction de $N-1$ histogrammes unidimensionnels. Il en va de même pour

$$P(\text{non-visage}|X_1, \ldots, X_N).$$

On peut constater que les deux distributions générées se recouvrent très peu. Ce qui laisse présager de bonnes performances.

La classification est tout simplement basée sur le maximum de vraisemblance. Une fenêtre étant considérée comme étant un visage si

$$P(\text{visage}|X_1, \ldots, X_N) > P(\text{non-visage}|X_1, \ldots, X_N).$$

Parmi les techniques de détection de visages basées sur la théorie de l'information, les plus célèbres sont celles de Colmenarez et Huang [7, 6, 8].

Dans [6], ils décrivent une première méthode utilisant comme critère de détection ce qu'ils appellent la distance de vraisemblance (*Likelihood Distance*) définie par

$$D(O, \lambda) = -\log(P(O, \lambda))$$

afin d'estimer à quel point l'observation O est proche de la classe λ (en l'occurrence celle des visages).

Ils se servent de la base de données d'entraînement afin d'estimer les densités de probabilités des observations et des classes. Comme dans l'approche précédemment décrite, ils n'utilisent pas de modèles paramétriques. Ils représentent les fenêtres de 30×30 pixels binaires par un modèle très proche de l'arbre utilisé ci-dessus. La seule différence est qu'on permet ici à certains pixels d'être complètement indépendants. Le critère utilisé pour le choix de la structure représentant au mieux l'objet décrit est la minimisation de l'entropie totale du modèle. On tente de minimiser ainsi l'information apportée par une nouvelle

observation et donc d'inclure le plus possible d'information pertinente *à l'intérieur* du modèle.

Les résultats obtenus sont dits excellents mais aucune statistique ne vient étayer cette affirmation dans [6].

La technique que nous venons de présenter ne modélise pas explicitement la classe des non-visages. Colmenarez décrit dans [7] une nouvelle méthode de détection également basée sur les résultats de la théorie de l'information. Il s'agit ici de modéliser d'une part la classe des visages et d'autre part la classe des non-visages. Afin de limiter le nombre d'erreurs de classification, on souhaite maximiser la distance entre les deux distributions de probabilités. Comme mesure de distance, on utilise l'*information relative de Kullback* définie par

$$H_{P||M} = \sum_{X^n} P_{X^n} \ln \frac{P_{X^n}}{M_{X^n}}$$

où P et M sont deux distributions de probabilités définies pour le même processus aléatoire X^n.

La structure choisie ici pour modéliser les fenêtres de 11×11 pixels est une chaîne de Markov d'ordre 1. Un processus aléatoire X^n est une chaîne de Markov d'ordre 1 si

$$P(X^n|X^{n-1}, \ldots, X^1) = P(X^n|X^{n-1}).$$

On cherche donc l'ordre à attribuer aux variables aléatoires, constituées des pixels de l'image, permettant de maximiser $H_{P||M}$ où P_{X^n} est la distribution de probabilités de la classe des visages et M_{X^n} celle des non-visages. Cet ordre sera donc celui qui maximisera le rapport de vraisemblance

$$L(X^n) = \frac{P(X^n)}{M(X^n)}.$$

Ce problème de maximisation ne peut être résolu exactement. Ce n'est en fait rien d'autre que le bien connu problème du voyageur de commerce pour lequel il n'existe que des algorithmes fournissant des solutions sous-optimales.

Cette technique obtient des performances comparables à celles du détecteur de visages par réseau de neurones de Rowley [41, 42] tout en étant deux fois moins coûteuse en temps de calcul.

Colmenarez décrit dans [8] une troisième technique de détection cette fois-ci capable de localiser 9 traits différents par visage. La méthode employée est fort similaire à celle que nous venons de décrire. La plus grosse différence est qu'elle détecte chacun des 9 traits indépendamment. On utilise ensuite des contraintes géométriques obtenues à partir de la base de données d'entraînement afin de décider quels ensembles de traits constituent bien des visages.

Afin de pouvoir considérer des visages ayant subi des rotations dans le plan, on commence par estimer les positions des visages et leurs orientations. On les remet ensuite en position droite avant d'effectuer la détection de traits et la classification géométrique.

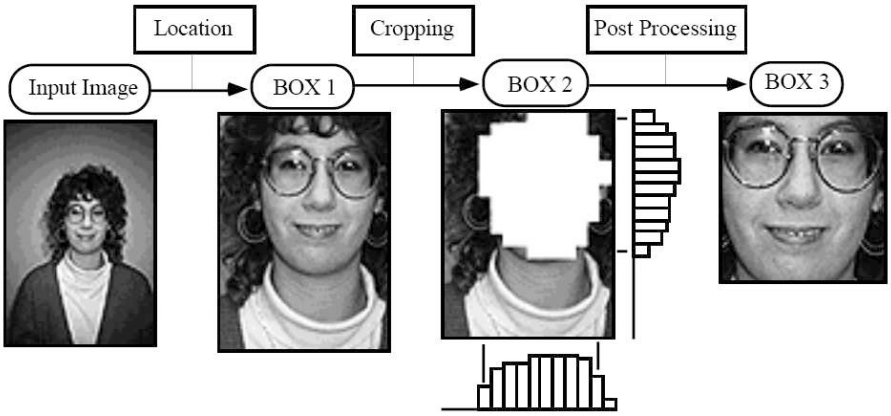

FIG. 2.32 – Illustration des 3 étapes principales de l' algorithme décrit par Huang dans [16].

Cependant, aucune description de la technique permettant d'estimer les positions et les orientations des visages n'est donnée dans [8].

Huang décrit dans [16] une technique basée sur les arbres de décision. Un arbre de décision est composé de noeuds correspondant à des tests à effectuer et de feuilles correspondant aux décisions d'attribuer une certaine classe à un objet. Cet arbre de décision est généré à partir d'une base de données d'exemples positifs et négatifs. Leur algorithme, illustré à la figure 2.32, comporte trois étapes :

1. La première consiste à localiser une fenêtre susceptible de contenir un visage. C'est une détection grossière basée sur une détection de bords et une analyse de leurs projections horizontale et verticale.
2. La seconde inspecte chaque zone de 8×8 pixels de l'image avec un arbre de décision afin de déterminer si elle appartient à un visage ou pas.
3. La dernière consiste à générer une boite de dimensions standards contenant le visage éventuellement détecté.

Une particularité de cet algorithme est de ne pas nécessiter l'examen d'une pyramide d'images. Le système présente le désavantage de n'être capable de détecter qu'un seul visage par image. Il donne une réponse correcte dans 96% des cas.

Roth, Yang et Ahuja décrivent dans [39] un système de détection de visages frontaux basé sur l'architecture d'apprentissage SNOW (*Sparse Network Of Winnows*[7]). Ce système possède un grand nombre de caractéristiques binaires. Ses sorties sont constituées de sommes pondérées des valeurs des caractéristiques. Il s'agit donc ici des sommes des poids des caractéristiques détectées. Il possède deux sorties, chacune possédant son propre réseau indépendant de détection. La première correspond à la détection d'un visage, la seconde à la détection d'un non-visage. La règle de décision consiste à attribuer une classe à un

[7] "To winnow" signifie "vanner" comme dans "vanner le grain".

objet si la valeur de la sortie correspondante dépasse un certain seuil. Si les deux sorties dépassent leur valeur de seuil, on choisit la classe correspondant à la valeur de sortie la plus grande. L'apprentissage se fait par actualisation des poids lors des mauvaises détections. On utilise pour cela les paramètres $\alpha > 1$ et $\beta < 1$. Si le système détecte faussement un visage, les poids des caractéristiques activées lors de cette fausse détection sont multipliés par β. Si le système manque une détection, les poids des caractéristiques activées lors de l'examen de la fenêtre de l'image contenant le visage non détecté sont multipliés par α. Les autres poids restent inchangés. L'avantage de ce système est que le nombre d'exemples requis pour l'entraîner ne croît que linéairement en fonction du nombre de caractéristiques. Un avantage supplémentaire est le fait que le système ne considère que les caractéristiques activées. Il n'est donc pas pénalisant du point de vue temps de calcul d'avoir un très grand nombre de caractéristiques dont seulement un petit sous-ensemble est activé à la fois.

Les caractéristiques considérées sont :

- Les valeurs en niveaux de gris des pixels de la fenêtre inspectée (20×20 pixels). Le pixel de position (x, y) d'intensité $I(x, y)$ (comprise entre 0 et 255) correspond à la caractéristique numéro $256 * (20 * y + x) + I(x, y)$ qui vaut 1 si l'intensité du pixel (x, y) vaut $I(x, y)$ et zéro si non.
- Les moyennes et les variances des niveaux de gris de zones de 2×2, 4×4 et 10×10 pixels de l'image. Leurs intervalles de valeurs sont discrétisés en une série de classes afin de pouvoir appliquer le même type d'attribution de numéro de caractéristique que précédemment.

Le nombre de caractéristiques actives est de $400 + 100 + 25 + 4$ par fenêtre. Le nombre total de caractéristiques est bien plus grand puisque les valeurs en niveaux de gris des pixels en génèrent à elles seules $400 \times 256 = 102.400$.

Les résultats obtenus sont excellents. Si on en croit les concepteurs du système, ils sont meilleurs que ceux de l'ensemble des autres techniques basées sur la classification décrites dans cette section.

Chapitre 3

Technique de détection proposée et implémentée

Ce chapitre est consacré à la description de la technique de détection que nous avons implémentée. Le chapitre suivant exposera les performances obtenues par celle-ci et proposera une série de modifications susceptibles de les améliorer.

Nous allons tout d'abord décrire l'architecture générale du détecteur. Nous détaillerons successivement plus en détail chacun des modules qu'il comporte dans les sections ultérieures du chapitre.

3.1 Architecture du détecteur

Le flux d'images qui constitue l'entrée du système passe successivement à travers les trois modules distincts qu'il comporte (voir figure 3.1) :

- Le premier module est chargé de sélectionner une série de zones d'intérêt à l'intérieur desquelles nous effectuerons la quasi-totalité des opérations ultérieures.
- Le second est un module de suivi des visages. Celui-ci est chargé de suivre ceux qui ont déjà été détectés une première fois par le programme lors d'une image précédente. Ce suivi ne se limite pas aux visages vus de face, il permet, dans une certaine mesure, de les suivre même lorsqu'ils ne sont plus face à la caméra.
- Le dernier module est chargé de détecter les visages qui n'ont pas encore été détectés une première fois par le programme. Ce module n'est capable de détecter les visages que lorsque ceux-ci sont vus de face.

Dans cette description très peu détaillée, apparaît déjà une caractéristique importante de notre système : s'il est capable de suivre des visages qui ne sont pas nécessairement vus de face, il ne peut les détecter une première fois que lorsque c'est le cas. Cependant, comme nous le verrons plus loin, lorsque les visages ne font pas face à la caméra, ce suivi est d'une part moins précis et, surtout, de durée limitée.

De plus, le programme ne détecte les visages que s'ils apparaissent droits dans l'image. La figure 3.2 illustre ce que nous entendons par là en termes d'orientation. Il n'est donc, par exemple, pas capable de détecter le visage orienté à 90° d'une personne couchée.

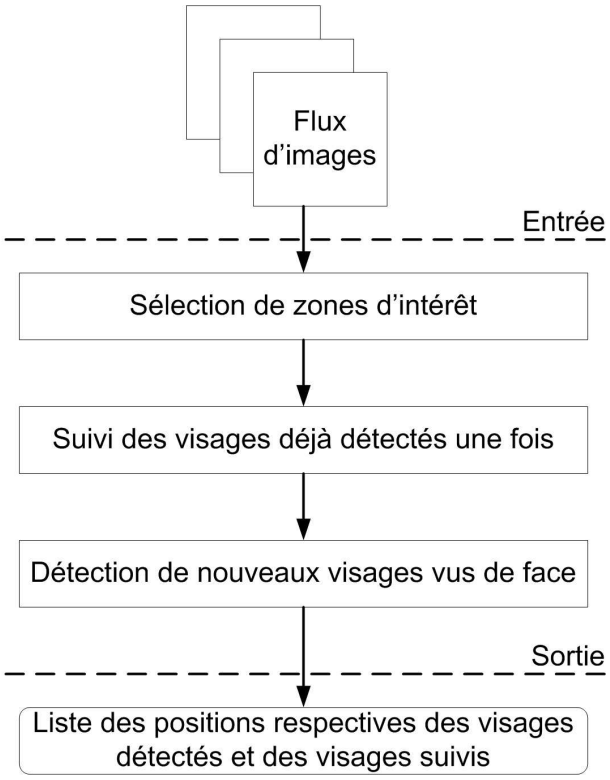

FIG. 3.1 – Schéma de l'architecture générale du détecteur.

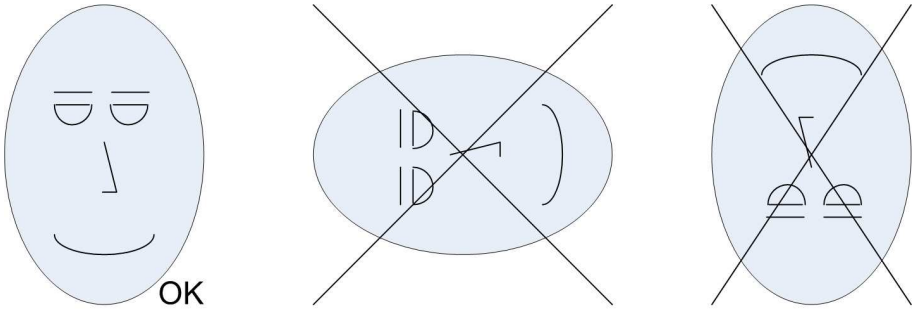

FIG. 3.2 – Illustration de l'orientation que doit avoir un visage pour être détecté et suivi.

Si le module de suivi est capable de suivre, dans une certaine mesure, les visages même lorsque ceux-ci ne sont plus vus de face, il n'est cependant pas capable de suivre un visage dont l'orientation s'éloignerait trop de la normale.

La description détaillée des trois modules qui constituent le détecteur fait l'objet des sections suivantes.

3.2 Sélection de zones d'intérêt

Cette section est consacrée à la description du premier des trois modules de la figure 3.1. Son rôle est de sélectionner dans l'image les pixels susceptibles de faire partie d'un visage.

Notre objectif étant la conception d'un détecteur capable de traiter un nombre significatif d'images par seconde, il est indispensable de minimiser la quantité de traitements à effectuer sur l'image.

La sélection de pixels d'intérêt en guise d'étape initiale est un premier moyen que nous utilisons pour réduire cette quantité de traitements.

En effet, la quasi-totalité des traitements effectués par les deux autres modules (suivi et détection frontale) ne sera faite qu'à l'intérieur de ces pixels.

Cette sélection comporte deux étapes (voir figure 3.4) : la première est la génération et la mise à jour d'un arrière-plan tandis que la seconde est la détection des pixels qui sont en superposition sur cet arrière-plan et qui sont de la couleur de la peau.

Nous faisons l'hypothèse que la caméra est fixe (hypothèse justifiée dans le cadre de l'application visée). L'image est donc constituée d'un arrière-plan fixe au-dessus duquel vient se superposer une série d'objets dont les personnes et les visages font partie (voir figure 3.3). La première étape consiste donc à essayer d'obtenir une image contenant cet arrière-plan, ou, au moins, une bonne approximation de celui-ci. Cette approximation de l'arrière-plan est mise à jour pour chaque nouvelle image.

Comme nous l'avons vu dans notre étude bibliographique du chapitre 2, la détection de la couleur de la peau est un moyen efficace souvent utilisé pour définir une série de zones candidates susceptibles de contenir un visage. De plus, celle-ci peut être effectuée de manière assez simple grâce à des seuillages dans un espace de couleurs approprié. Une telle détection est alors relativement peu coûteuse en temps de calcul et donc particulièrement adaptée à un détecteur devant traiter un nombre significatif d'images par seconde. Notre deuxième étape consiste donc à détecter les pixels qui sont de la couleur de la peau.

Ces deux étapes nous permettent de définir quels pixels sont des pixels d'intérêt susceptibles de contenir un visage et quels pixels ne le sont pas.

Pour être considéré comme étant d'intérêt, un pixel devra donc d'une part être en superposition sur l'arrière-plan et d'autre part être de la couleur de la peau.

FIG. 3.3 – A gauche : image à traiter. A droite : arrière-plan de celle-ci

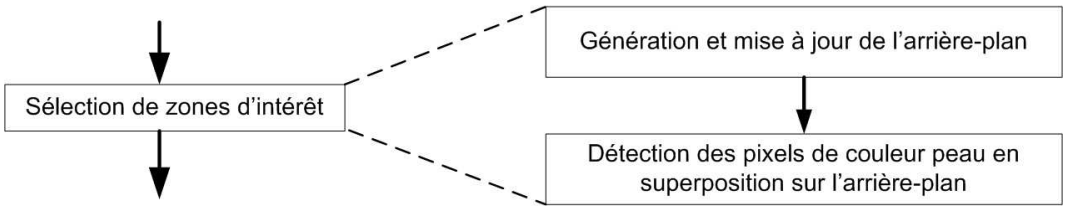

FIG. 3.4 – Détection des pixels d'intérêt en deux étapes.

Nous allons maintenant décrire en détail ces deux étapes. La section 3.2.1 est consacrée à la génération et à la mise à jour de l'arrière-plan et la section 3.2.2 à la détection des pixels de couleur peau.

3.2.1 Génération et mise à jour de l'arrière-plan

La génération et la mise à jour de l'arrière-plan constituent l'étape qui nous permet d'exploiter l'information de nature temporelle contenue dans le flux d'images. C'est en effet en grande partie grâce à elle que nous prenons en compte le fait que les images successivement présentées au détecteur sont fortement corrélées puisqu'elles représentent toutes l'évolution temporelle d'une même scène.

Nous employons une technique inspirée par [15].

Elle consiste à considérer la première image présentée au système comme approximation initiale de l'arrière-plan. Cette approximation est ensuite mise à jour pour chaque nouvelle image.

La technique de mise à jour de l'arrière-plan consiste à attribuer une des trois classes décrites ci-dessous à chaque pixel de la nouvelle image. A chacune de ces trois classes correspond une action à effectuer pour mettre à jour le pixel correspondant de l'arrière-plan.

Chaque pixel de l'image est donc classé dans une des trois classes suivantes :

- La première classe est celle des pixels faisant partie de l'arrière-plan (classe BKG).
- La seconde classe contient les pixels appartenant à un objet en superposition sur celui-ci (classe SUP).
- Les pixels appartenant à la troisième classe sont ceux faisant partie d'un *fantôme* (classe $GHOST$).

La troisième classe mérite quelques explications. Un fantôme est un ensemble de points détectés comme étant en superposition sur l'arrière-plan mais n'appartenant à aucun objet physiquement en superposition sur celui-ci.

Cette situation se présente lorsqu'un objet appartenant à l'arrière-plan se met en mouvement.

Dans notre contexte, ce cas de figure se présente, par exemple, systématiquement lorsqu'une personne en mouvement est présente lors de la première image. En effet, lors de l'initialisation du système, cette première image est prise comme première approximation de l'arrière-plan. Celle-ci étant en mouvement, elle constitue dès la seconde image un objet de l'arrière-plan qui se met en mouvement.

Ce phénomène est illustré à la figure 3.5. L'arrière-plan y contient deux objets : A et B. Sur la nouvelle image, l'objet A se met en mouvement, laissant un *vide* derrière lui. La détection d'objets en superposition sur l'arrière-plan se fait en comparant celui-ci avec la nouvelle image. On voit sur la partie de droite de la figure 3.5 que deux zones de la nouvelle image diffèrent de l'arrière-plan : d'une part la zone correspondant à la nouvelle place de l'objet A et d'autre part l'*ancien* emplacement de celui-ci. Cette deuxième zone est un fantôme : c'est un objet en superposition sur l'arrière-plan qui ne correspond à

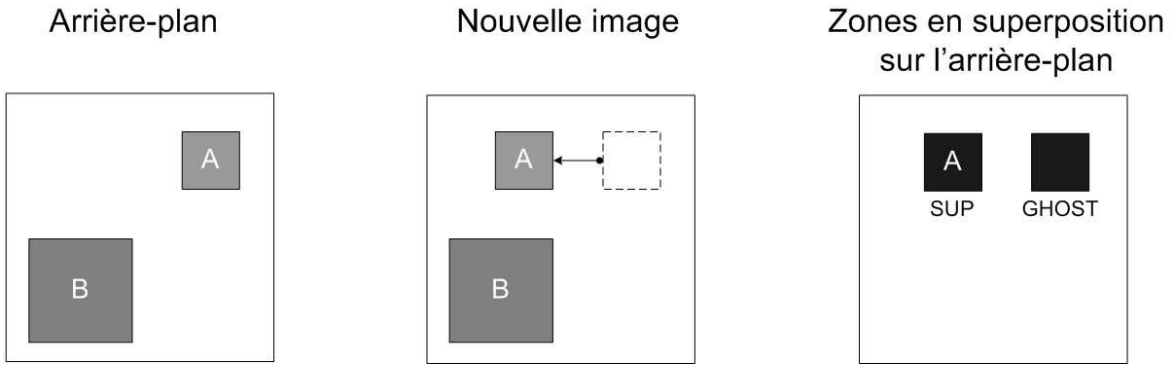

FIG. 3.5 – Apparition d'un fantôme lorsque l'objet A appartenant à l'arrière-plan se met en mouvement.

aucun objet physiquement en superposition sur celui-ci.

Lorsqu'un fantôme est détecté, il est bien entendu nécessaire de mettre à jour l'arrière-plan afin de l'y inclure puisque cet objet ne correspond à aucun objet physiquement en superposition sur l'arrière-plan.

Chaque pixel de la nouvelle image fait donc partie soit de l'arrière-plan, soit d'un objet en superposition sur celui-ci, soit d'un fantôme.

Afin de mettre à jour l'arrière-plan, pour chaque pixel de chaque nouvelle image, on effectue l'une des actions suivantes :

- Si le pixel fait partie de l'arrière-plan, il n'est pas nécessaire d'actualiser sa valeur. Cependant, afin d'éliminer une partie du bruit, on remplace le pixel correspondant de l'arrière-plan par une somme pondérée de sa valeur et de la valeur du pixel de la nouvelle image.
- Si le pixel fait partie d'un objet en superposition sur l'arrière-plan, la valeur du pixel correspondant de l'arrière-plan doit rester inchangée.
- Si le pixel fait partie d'un fantôme, il faut inclure le fantôme dans l'arrière-plan. On remplace donc la valeur du pixel de l'arrière-plan par celle du pixel de la nouvelle image.

Il nous faut maintenant préciser quelles règles nous permettent de déterminer à quelle classe appartient un pixel.

La première étape consiste à comparer l'arrière-plan et la nouvelle image.

Si la valeur d'un pixel est la même dans chacune des deux images, ce pixel est considéré comme faisant partie de l'arrière-plan.

Si sa valeur dans la nouvelle image est différente de sa valeur dans l'arrière-plan, le pixel fait partie soit d'un objet en superposition sur celui-ci, soit d'un fantôme.

Nous devons alors trouver un critère permettant de distinguer ces deux classes. Un objet en superposition sur l'arrière-plan est en général en mouvement. Un fantôme, par contre, est toujours immobile. C'est cette différence qui va nous permettre de faire la

distinction entre les deux classes : les pixels d'un fantôme sont indéfiniment détectés en superposition sur l'arrière-plan.

Un pixel détecté en superposition sur l'arrière-plan un certain nombre de fois consécutives (il s'agit là d'un seuil à fixer) sera considéré comme faisant partie d'un fantôme et sera donc alors inclus dans l'arrière-plan. Nous choisissons de fixer ce seuil à 30, nombre d'images correspondant à une durée d'un peu plus d'une seconde.

Un pixel détecté en superposition sur l'arrière-plan mais qui ne l'a pas encore été un nombre suffisant de fois est considéré comme faisant partie d'un objet en superposition sur celui-ci.

Remarque importante

Dans le cadre de notre détecteur de visages, employée telle quelle, cette technique de génération et de mise à jour de l'arrière-plan comporte un gros défaut : un visage immobile durant un trop grand nombre d'images peut être inclus dans l'arrière-plan puisque ses pixels seront détectés en superposition sur celui-ci un grand nombre de fois consécutives. Une fois inclus, celui-ci ne peut plus être détecté jusqu'à ce qu'il se remette en mouvement.

Pour résoudre ce problème, chaque fois qu'un pixel est détecté comme faisant partie d'un visage, nous remettons à zéro la variable comptabilisant le nombre de fois consécutives où il a été détecté en superposition sur l'arrière-plan. Nous évitons ainsi qu'un visage soit inclus dans l'arrière-plan.

Parmi les pixels faisant partie d'objets en superposition sur l'arrière-plan, il nous faut maintenant détecter ceux qui sont de la couleur de la peau. Cette détection est décrite dans la section suivante.

3.2.2 Détection des pixels de la couleur de la peau

Comme nous l'avons vu au chapitre 2, il existe un grand nombre de techniques permettant de classer des pixels comme étant, ou n'étant pas, de couleur peau. Ces techniques peuvent être très simples ou très sophistiquées.

Ayant pour objectif la conception d'un système rapide, nous choisissons volontairement une technique très peu complexe faisant appel à de simples seuillages.

Pour cela, il nous faut choisir un espace de couleurs approprié. Comme expliqué au chapitre 2, les espaces convenant le mieux sont ceux qui décorrèlent l'information de luminance de l'information de chrominance. Il existe toute une série d'espaces répondant à ce critère.

Parmi ceux-ci, nous choisissons l'espace *HSV* (*Hue* (Teinte), *Saturation* (Saturation), *Value* (Brillance), notamment parce que celui-ci est l'un des plus connus et que ses trois composantes ont une signification physique claire.

Nous devons maintenant décider dans quelle(s) composante(s) effectuer nos seuillages. Nous choisissons la teinte et la saturation. Nous excluons la brillance puisqu'elle contient l'information de luminance dont nous cherchons justement à nous affranchir.

La couleur de la peau est contenue dans une section continue relativement étroite de la teinte. Cette section se situe aux alentours de la teinte rouge. De plus, la couleur de la peau n'est que peu saturée.

Nous classons donc un pixel i comme étant de couleur peau si

$$\text{teinte}_{\min} \leq \text{teinte}(i) \leq \text{teinte}_{\max} \qquad \text{et} \qquad \text{saturation}(i) \leq \text{saturation}_{\max}$$

avec

$$\begin{aligned} \text{teinte}_{\min} &= 0° \\ \text{teinte}_{\max} &= 43° \\ \text{saturation}_{\max} &= 50\%. \end{aligned}$$

Nous effectuons volontairement un seuillage relativement large afin d'éviter au maximum de classer des pixels de peau comme n'étant pas de couleur peau. En effet, puisque la plupart de nos traitements ultérieurs ne se font qu'à l'intérieur des zones sélectionnées ici, il faut nous assurer de ne pas éliminer erronément un trop grand nombre de pixels de peau.

Le calcul des valeurs de la teinte et de la saturation est une opération coûteuse en calculs si on considère le nombre de fois qu'elle doit être faite pour chaque image traitée. Afin de limiter le temps pris par cette transformation, nous stockons dans deux tableaux l'ensemble des relations $(R, G, B) \rightarrow H$ et $(R, G, B) \rightarrow S$. Chacun de ces deux tableaux occupe près de 16 mégas de mémoire vive mais le gain de vitesse obtenu justifie pleinement leur utilisation.

Comme nous l'avons déjà mentionné, nous ne retenons ici que les pixels de couleur peau qui font partie d'un objet en superposition sur l'arrière-plan. Les pixels répondant à ces deux critères constituent les pixels d'intérêt qui forment les zones à l'intérieur desquelles nous allons effectuer notre recherche de visages frontaux et une grande partie de notre suivi de visages.

La section suivante décrit notre système de détection de visages vus de face.

3.3 Détection de visages vus de face

La description du troisième module de la figure 3 fait l'objet de cette section. Nous décrivons le troisième module avant le second parce qu'il nous paraît plus naturel de commencer par décrire la technique permettant la première détection d'un visage avant d'évoquer les opérations permettant son suivi.

Le rôle de ce module est donc, à partir de l'ensemble des pixels d'intérêt qui lui sont fournis en entrée, de produire une liste des positions des nouveaux visages frontaux apparus dans la scène vidéo.

Elle est constituée de quatre étapes principales (voir figure 3.6).

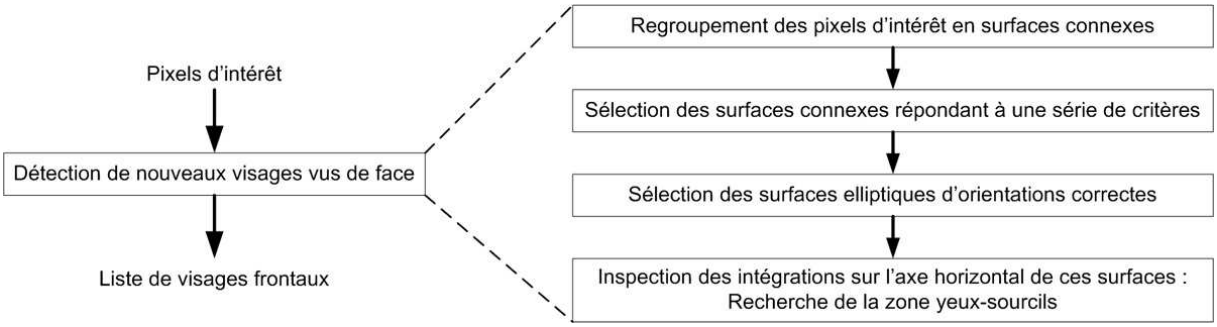

FIG. 3.6 – Les quatre étapes successives permettant la détection de visages vus de face.

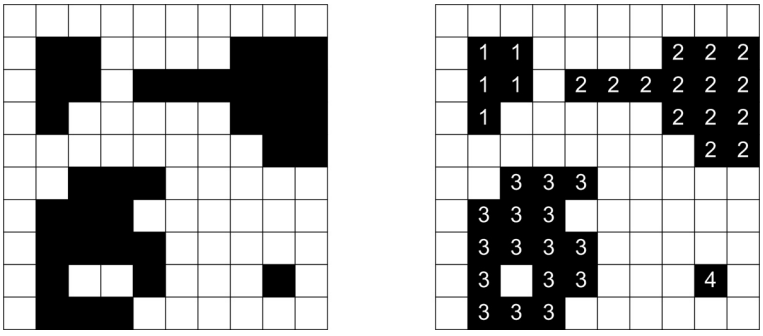

FIG. 3.7 – Exemple de génération de surfaces de pixels connexes.

- La première étape consiste à regrouper les pixels d'intérêt en surfaces connexes.
- La seconde consiste à, parmi ces dernières, sélectionner celles qui répondent à toute une série de critères simples concernant leur taille, leurs proportions,...
- La troisième sélectionne les surfaces dont la forme est proche d'une ellipse.
- La dernière consiste à rechercher la zone yeux-sourcils en inspectant les intégrations le long de l'axe horizontal des surfaces ayant réussi à passer au travers des trois étapes précédentes. Les surfaces pour lesquelles on arrive à localiser cette zone sont alors considérées comme étant des visages.

Nous allons maintenant décrire plus en détail chacune de ces quatre étapes.

3.3.1 Regroupement des pixels d'intérêt en surfaces connexes

Il s'agit ici d'obtenir à partir de l'ensemble des pixels d'intérêt, un ensemble de surfaces connexes reprenant l'intégralité de ceux-ci. Chaque surface se voit attribuer une étiquette unique.

On considère ici que chaque pixel possède 8 voisins (à l'exception des pixels situés sur les bords de l'image).

La figure 3.7 comporte un exemple de génération de surfaces de pixels connexes.

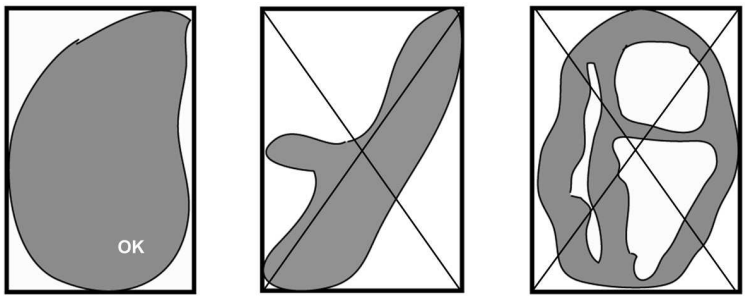

FIG. 3.8 – Illustration de la sélection des surfaces en fonction de leur compacité.

3.3.2 Sélection des surfaces répondant à une série de critères simples

Il s'agit ici d'éliminer déjà une partie de ces surfaces en appliquant trois critères discriminants relativement simples. Cette première sélection, très peu coûteuse en temps de calcul, nous permet d'économiser un grand nombre d'opérations. En effet, nous n'effectuons ainsi les traitements ultérieurs plus coûteux en temps de calcul que sur un nombre restreint de surfaces.

Les trois critères auxquels une surface doit répondre sont :

- Elle doit être de taille suffisante. La recherche de la zone yeux-sourcils de la dernière étape ne peut se faire que sur une surface suffisamment étendue. Nous excluons ici d'emblée les surfaces de moins de 1600 pixels (40×40). Les visages de tailles inférieures ne seront donc pas détectés.
- Elle doit avoir des proportions correctes. Nous calculons le rapport entre la hauteur et la largeur du plus petit rectangle contenant la surface et ne retenons que les surfaces pour lesquelles ce rapport se situe dans un certain intervalle. Après une série de tests, nous décidons que ce rapport doit être situé entre 1 et 2. Nous excluons donc ici tous les visages dont l'orientation s'éloigne trop de la normale. Par exemple, le rapport entre la hauteur et la largeur d'une surface correspondant à un visage orienté à 90° sera, selon toute vraisemblance, plus petit que 1. Une telle surface ne sera donc pas retenue ici.
- Enfin nous imposons que ces surfaces soient suffisamment compactes. Pour cela, nous calculons pour chacune d'entre elles le rapport entre la surface du plus petit rectangle qui la contient et la surface de celle-ci. Nous ne retenons que les surfaces pour lesquelles ce rapport est suffisamment élevé (40% au minimum). Ceci est illustré à la figure 3.8.

Nous allons maintenant vérifier que ces surfaces ont une forme elliptique.

3.3.3 Sélection des surfaces elliptiques d'orientations correctes

Comme nous l'avons vu à la section 2.2, il est fréquent de faire l'hypothèse que la forme du contour du visage est elliptique.

Le recherche de zones de peau elliptiques est une technique plusieurs fois utilisée avec succès dans des systèmes de détection de visages. Plusieurs détecteurs présentés à la section 2.2 y font appel.

De plus, si cette recherche est plus coûteuse en calculs que les critères simples de la section précédente, nous ne l'appliquons qu'à un nombre déjà fort réduit de surfaces. Elle ne devrait donc pas nécessiter un nombre trop important de calculs.

Nous calculons donc, pour chaque surface ayant passé les tests de l'étape précédente, les coordonnées de l'ellipse qui lui correspond le mieux (voir figure 2.8).

Le centre de l'ellipse est le centre de masse de la surface considérée. Les tailles de ses axes ainsi que son orientation sont données par les formules 2.1 à 2.4 mentionnées à la section 2.2 et expliquées dans [19].

Nous sélectionnons ensuite les surfaces dont la forme se rapproche suffisamment de celle de sa meilleure ellipse. Pour cela, nous utilisons la mesure de distance proposée dans [48]. Celle-ci correspond à la formule 2.5 mentionnée à la section 2.2.

Cette mesure de distance nous indique à quel point la surface correspond à sa meilleure ellipse en comptant les trous à l'intérieur de cette dernière ainsi que les points de la surface situés à l'extérieur de celle-ci.

Puisque cette mesure de distance est normalisée, nous pouvons choisir une valeur de seuil au-delà de laquelle une surface n'est pas reprise.

Après une série de tests, nous avons décidé de fixer ce seuil à $0,5$. Puisque nous ne sommes pas sensés détecter des visages dont l'orientation s'éloigne trop de la normale, nous fixons également une condition sur l'angle θ définissant l'orientation de l'ellipse (voir figure 2.8). Pour que la surface à laquelle il correspond puisse être retenue, il doit satisfaire à la condition suivante :

$$\frac{\pi}{4} \leq \theta \leq \frac{3\,\pi}{4}.$$

Une détection de visages frontaux s'arrêtant ici donne déjà des résultats encourageants. Le nombre de fausses détections, s'il n'est pas énorme, reste cependant significatif. Il nous faut maintenant vérifier que ces surfaces de couleur peau elliptiques correspondent bien à des visages. Pour cela, nous recherchons à l'intérieur de celles-ci la zone yeux-sourcils.

3.3.4 Recherche de la zone yeux-sourcils

La recherche des traits du visage en guise d'ultime vérification est une approche classique. Ses avantages sont nombreux. Le principal étant que, si le nombre de surfaces à examiner est relativement petit, ce qui est la cas ici, elle est peu coûteuse en temps de calcul. De plus, l'apparence relativement sombre vis à vis du reste du visage d'une zone telle que celle regroupant les yeux et les sourcils est une des constatations évidentes tirées

FIG. 3.9 – Sur cette figure, l'intégration des niveaux de gris sur l'axe horizontal correspond à la courbe bleue. On y observe un minimum local fort marqué à hauteur des yeux. La ligne verticale rouge représente la valeur moyenne de la courbe bleue.

de l'observation attentive d'une image en niveaux de gris d'un visage. On peut donc supposer qu'une technique de détection de celle-ci basée sur cet aspect plus sombre qu'elle présente a beaucoup de chances d'être robuste.

Nous allons donc nous en servir pour éliminer les quelques fausses détections encore présentes à ce stade.

Cette zone peut être détectée en inspectant l'intégration des niveaux de gris de la surface sur l'axe horizontal. On y observe à hauteur des yeux un minimum local très marqué (voir figure 3.9).

Nous pensions initialement détecter de manière similaire un plus grand nombre de traits du visage. Cependant, une série de tests nous ont permis de constater que la seule information que nous pouvions détecter avec un taux de confiance suffisant était la hauteur des yeux.

Ce minimum est très marqué car d'une part, les yeux sont une des parties les plus sombres du visage et d'autre part, ceux-ci sont situés entre le front et les joues qui sont deux parties fort claires de celui-ci.

Afin de relever les contrastes et d'ainsi marquer encore plus ce minimum local, nous effectuons dans le plus petit rectangle contenant la surface inspectée une **égalisation de l'histogramme** des niveaux de gris.

Nous obtenons ansi un minimum suffisamment marqué pour être détecté de manière fiable.

La détection de ce dernier consiste à déterminer la valeur moyenne de l'intégration sur l'axe vertical des niveaux de gris de la surface (en bleu sur la figure 3.9) et à parcourir ensuite cette dernière en partant du haut jusqu'à trouver le premier point dont la valeur est supérieure à la valeur moyenne. Cette première étape nous permet de ne pas détecter le mimimum local correspondant aux cheveux (plus sombres eux aussi) lorsque ceux-ci

sont inclus dans la surface. A partir de ce point, nous cherchons ensuite la première zone située *en-dessous* de la moyenne et nous en recherchons le minimum. Si la surface inspectée est un visage, ce minimum correspond alors à la zone yeux-sourcils.

La sélection des surfaces consiste à ne retenir que celles pour lesquelles ce minimum est détecté dans une zone jugée acceptable : la distance d entre celui-ci et le bord supérieur du plus petit rectangle contenant la surface doit répondre à la condition suivante :

$$0,2\,h \leq d \leq 0,5\,h,$$

où h est la hauteur du plus petit rectangle contenant la surface.

Si cette première tentative de sélection de la surface échoue, nous réitérons le processus en ajoutant 10% à la valeur de la moyenne. Une série de tests nous a conduits à procéder de la sorte afin de n'éliminer erronément qu'un nombre limité de sufaces. Si la surface n'est pas sélectionnée après cette seconde tentative, elle est définitivement éliminée.

Ceci constitue la dernière étape de notre détection de visages vus de face. Les surfaces répondant à l'ensemble des critères mentionnés jusqu'ici sont considérées comme étant des visages. La sortie de ce module de détection frontale est constituée d'une liste des positions respectives des différents visages détectés. Cette dernière sera fournie au module de suivi de visages lors de la prochaine image.

Ce module de suivi des visages déjà détectés une première fois est décrit dans la section suivante.

3.4 Suivi des visages déjà détectés une première fois

Une fois qu'un visage a été détecté, nous le suivons au fil des images au moyen d'un module séparé (voir figure 3.1).

Incorporer un module de suivi de visages en plus du module de détection est intéressant pour plusieurs raisons :

- Nous pouvons grâce à lui connaître le parcours d'une personne dans l'image ou encore le laps de temps durant lequel cette personne est restée devant la caméra.
- Nous pouvons nous servir du fait qu'un personne ne peut ni disparaître subitement d'une image à l'autre, ni changer radicalement de position ou d'apparence pour rendre plus robuste la détection de cette personne lors du suivi.
- Nous pouvons, dans une certaine mesure, suivre des visages alors qu'ils ne sont plus face à la caméra.

Le schéma général du suivi de visages est représenté à la figure 3.10.

Il consiste à, pour chaque visage détecté ou suivi dans l'image précédente, rechercher un visage dans le voisinage de celui-ci.

Cette recherche est d'abord effectuée en utilisant les mêmes outils que pour la détection de visages vus de face : nous recherchons des zones de peau de formes elliptiques. Suivant

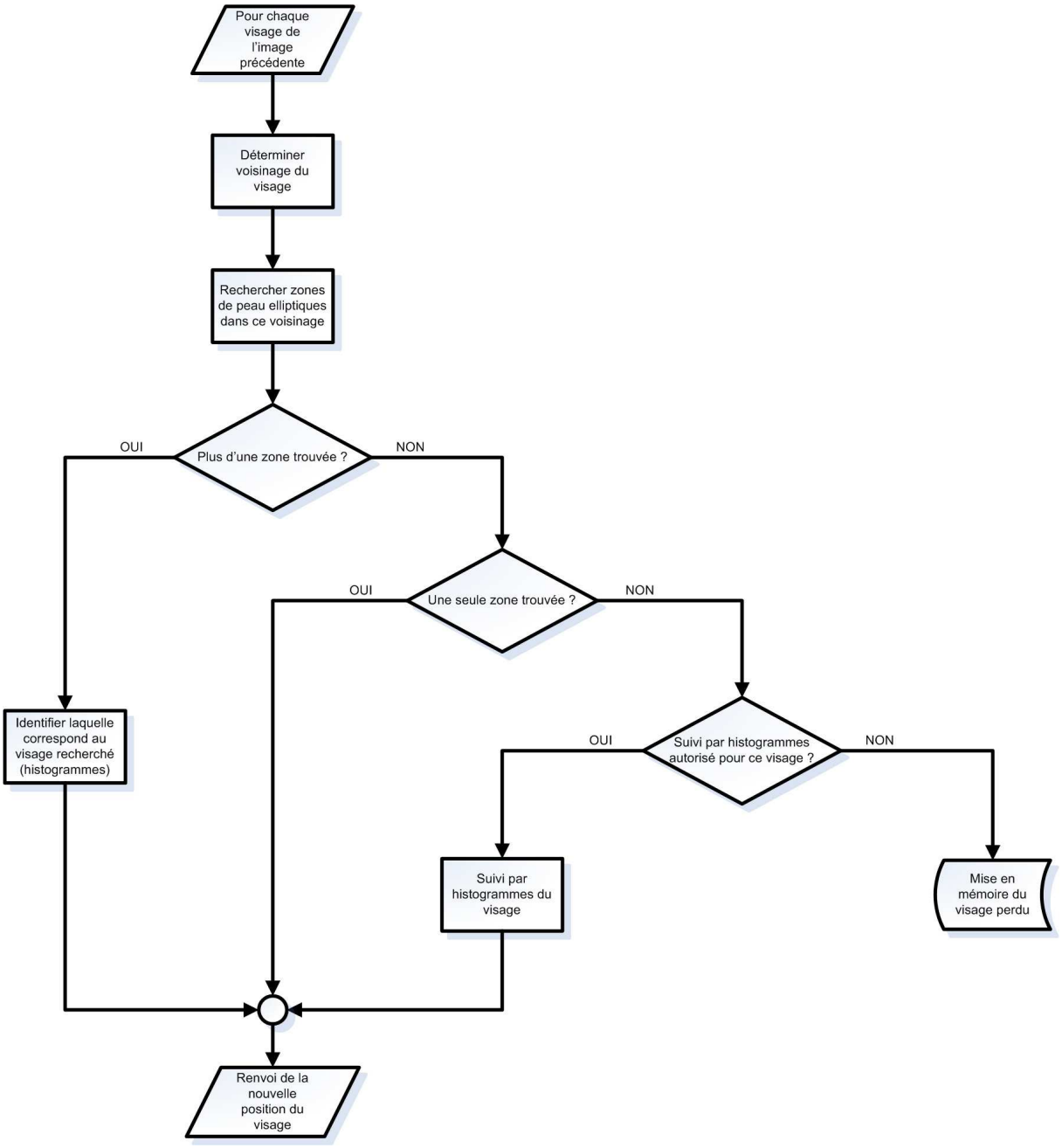

FIG. 3.10 – Schéma général du système de suivi de visages.

le nombre de zones candidates ainsi trouvées *dans le voisinage d'un visage de l'image précédente*, on effectue l'une des actions suivantes :

- Si plusieurs zones sont trouvées, on utilise un critère de sélection basé sur l'utilisation d'histogrammes de couleurs pour choisir laquelle parmi ces zones correspond au visage que nous suivons.
- Si une seule est trouvée, on considère que celle-ci correspond au visage que nous suivons.
- Si aucune zone candidate n'est ainsi trouvée, on recherche alors éventuellement le visage suivi à l'aide d'histogrammes de couleurs. Un visage ne peut être suivi à l'aide d'histogrammes de couleurs que s'il est déjà suivi avec succès depuis un certain nombre d'images. Cette technique de suivi trouve **toujours** une zone supposée contenir le motif recherché, quel que soit ce dernier. Nous imposons donc cette restriction afin de minimiser le risque de suivre une zone faussement détectée comme contenant un visage.
- Si pour un visage nous ne trouvons pas de zone de peau elliptique et que le suivi par histogrammes de couleurs n'est pas encore autorisé pour celui-ci, nous gardons en mémoire la dernière position connue de ce visage et retentons de le suivre à partir de cette position pendant encore un certain nombre d'images avant de le considérer comme définitivement perdu. S'il n'a pas quitté le champ de la caméra, il sera probablement redétecté par le module de détection frontale.

Les sections suivantes décrivent en détail les différentes étapes décrites ci-dessus. La section 3.4.1 précise ce que nous entendons par "voisinage". La section 3.4.2 décrit la recherche de zones de peau élliptiques dans ce voisinage. La section 3.4.3 décrit comment, parmi plusieurs de ces zones, nous déterminons laquelle correspond au visage suivi. Et enfin la section 3.4.4 détaille notre suivi par histogrammes de couleurs.

3.4.1 Définition du voisinage du visage suivi

Pour chaque visage détecté ou suivi dans l'image précédente, ainsi que chacun des visages éventuellement *perdus* depuis peu, nous définissons un voisinage à l'intérieur duquel nous le cherchons. Ce voisinage est tout simplement une zone rectangulaire centrée sur la dernière position connue du visage et dont la taille est supérieure et proportionnelle à celle du plus petit rectangle contenant le visage lors de sa dernière détection. Ceci est illustré à la figure 3.11.

3.4.2 Détection de zones de peau elliptiques dans le voisinage du visage suivi

Maitenant que notre zone de recherche est définie, nous tentons d'y trouver des zones candidates de peau elliptiques. Cette détection est faite de manière similaire à celle effectuée dans le cadre de la détection de visages vus de face expliquée à la section 3.3.

Nous n'effectuons cependant pas la dernière étape de validation constituée par la

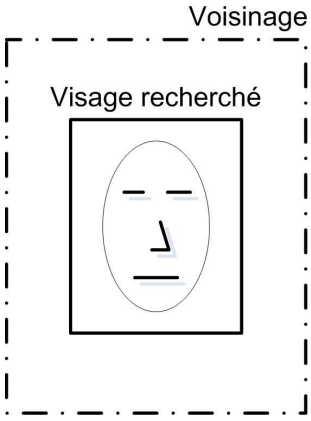

FIG. 3.11 – Voisinage du visage recherché.

recherche de la zone yeux-sourcils. En effet, nous avons remarqué lors de tests que si cette dernière étape permettait de réduire fortement le nombre de fausses détections, elle empêchait parfois la détection de certains visages durant quelques images. Nous ne l'incorporons donc pas au suivi.

C'est notamment ici que le module de suivi montre tout son intérêt. Il nous permet de garder l'étape de recherche de la zone yeux-sourcils dans le module de première détection tout en minimisant l'effet néfaste de celle-ci. En effet, une fois qu'un visage a été détecté une première fois, ce qui finit quasi toujours par arriver, même après quelques éliminations erronnées, le module de suivi le détecte au cours des images suivantes sans imposer cette dernière étape et sans donc risquer de l'éliminer faussement. Le nombre d'images dans lequel nous manquons de détecter ce visage à cause d'elle est donc très réduit, une seule validation suffisant à initialiser le processus de suivi.

La suite des opérations dépend maintenant du nombre de zones candidates détectées ici.

Si aucune zone n'est détectée, on passe alors éventuellement au suivi par histogrammes de couleurs décrit à la section 3.4.4.

Si une seule zone est trouvée, on considère qu'elle correspond au visage suivi. Le suivi de ce visage est alors terminé ici pour cette image.

Si plusieurs zones sont détectées, il nous faut alors identifier laquelle correspond au visage que nous suivons actuellement.

3.4.3 Identification de la zone correspondant effectivement au visage suivi

Le problème à résoudre ici est le suivant : nous avons un visage détecté ou suivi lors d'une image précédente (généralement la dernière) et une série de zones candidates susceptibles de lui correspondre dans l'image courante. Nous devons identifier laquelle lui

correspond effectivement.

Pour cela, nous comparons les histogrammes de couleurs du visage recherché avec ceux des zones candidates. Par zone nous entendons le plus petit rectangle contenant le visage ou la zone de peau elliptique détectée.

Comme mesure de similarité entre deux histogrammes, nous utilisons l'intersection (déjà mentionnée à la section 2.2.4) dont l'expression mathématique est :

$$\phi(\mathbf{s}) = \frac{\sum_{i=1}^{N} \min(I_{\mathbf{s}}(i), M(i))}{\sum_{i=1}^{N} I_{\mathbf{s}}(i)} \tag{3.1}$$

où N est le nombre de tranches que comporte un histogramme, M est un histogramme du visage recherché et $I_{\mathbf{s}}$ est un histogramme d'une des zones candidates.

Afin d'éviter une transformation vers un autre espace de couleurs coûteuse en calculs, nous générons nos histogrammes dans l'espace RGB. Nous exploitons chacune de ses trois composantes. Nous générons donc trois histogrammes pour le visage recherché et trois histogrammes pour chacune des zones candidates.

Les zones de l'image considérées ici ne sont pas de tailles suffisantes pour travailler avec des histogrammes complets (comportant 256 valeurs chacun). Nous utilisons des histogrammes comportant chacun 8 tranches.

Afin de déterminer quelle zone candidate correspond au visage suivi, nous commençons par générer les histogrammes de ce dernier ainsi que ceux de toutes les zones candidates. Pour chaque composante, nous calculons ensuite les intersections entre l'histogramme du visage recherché et les histogrammes des zones candidates. Pour chaque zone candidate, nous sommons les valeurs des trois intersections.

Nous choisissons alors la zone candidate pour laquelle cette somme est la plus élevée.

Il arrive parfois que nous ne trouvions aucune zone candidate lors de la recherche de zones de peau elliptiques.

Si le visage n'est pas suivi avec succès depuis un nombre suffisant d'images consécutives, le suivi échoue pour cette image. Nous continuons alors à essayer de le retrouver durant un nombre déterminé d'images. Si nous ne le retrouvons pas à temps, il est considéré comme perdu. Si celui-ci est en fait toujours dans le champ de la caméra, il sera redétecté une "première" fois lorsqu'il se positionnera face à celle-ci à nouveau.

Dans le cas où le visage est bien suivi avec succès depuis un nombre suffisant d'images consécutives, nous utilisons alors un autre type de suivi pour le retrouver dans l'image courante.

3.4.4 Suivi par histogrammes de couleurs

La recherche de zones de peau elliptiques à l'intérieur du voisinage du visage recherché n'ayant pas abouti, nous recherchons ici le visage, toujours à l'intérieur du voisinage de sa dernière position connue, en utilisant exclusivement des histogrammes de couleurs.

Sa dernière position connue est toujours ici située dans l'image précédant immédiatement l'image courante. En effet, le suivi par histogrammes n'est activé que si le visage est

correctement suivi depuis un certain nombre d'images consécutives.

Nous faisons donc ici l'hypothèse qu'entre deux images, le visage n'a pas le temps de changer suffisamment d'apparence pour affecter grandement ses histogrammes de couleurs.

Nous recherchons ici la zone, de même taille que la zone contenant le visage dans l'image précédente, dont les histogrammes de couleurs sont les plus proches de ceux du visage.

Nous utilisons une méthode similaire à celle décrite dans la section 3.4.3 si ce n'est que l'ensemble de zones candidates est constitué de l'ensemble des zones du voisinage dont la taille est celle de la zone de l'image précédente contenant le visage recherché.

Nous travaillons donc dans l'espace RGB avec des histogrammes comportant chacun 8 tranches.

Le choix de l'espace RGB nous permet d'économiser les calculs correspondant à un changement d'espace de couleurs. Cependant, la plupart des opérations effectuées jusqu'ici ont été faites à l'intérieur de l'espace HSV. Nous nous sommes donc posé la question de savoir si le suivi par histogrammes de couleurs ne serait pas plus robuste s'il était effectué dans l'espace HSV.

Après une série de tests, nous avons constaté une légère amélioration du suivi lors de l'utilisation de cet espace de couleurs. Cependant, celle-ci se faisait au détriment du nombres d'images traitées par seconde. Nous avons en effet constaté un net allongement du temps nécessaire à l'accomplissement des opérations correspondant au suivi par histogrammes de couleurs lors de l'utilisation de cet espace.

Nous considérons que le léger gain en qualité de suivi n'est pas justifié face à l'allongement singificatif du temps de calcul. Nous continuons donc à faire notre suivi dans l'espace RGB.

Nous avons égalemment vérifié que les performances n'étaient pas meilleures lors de l'utilisation d'histogrammes comportant un plus grand nombre de tranches. Le passage à 16 tranches n'ayant pas augmenté la qualité du suivi, nous continuons à utiliser des histogrammes à 8 tranches puisque le nombre d'opérations nécessaires pour calculer leurs intersections est moins élevé.

Nous travaillons avec des zones rectangulaires. Il aurait cependant paru plus naturel de considérer des zones elliptiques. Cependant, le code correspondant à l'utilisation de zones rectangulaires est facilement optimisable : on peut obtenir les histogrammes d'une zone décalée d'une colonne par rapport à une autre en reprenant ceux de cette dernière et en effectuant un nombre réduit d'opérations : pour chaque histogramme, on décompte une colonne et on en compte une nouvelle.

C'est la raison pour laquelle nous utilisons des zones rectangulaires.

Cette technique de suivi renvoit donc **toujours** une zone supposée correspondre au "visage" recherché, même si celui-ci n'en était en fait pas un. C'est la raison pour laquelle nous ne l'utilisons que si le visage est correctement suivi depuis un nombre suffisant

d'images consécutives en utilisant la recherche de zones de peau elliptiques. Nous choisissons de fixer ce nombre à 5. Nous considérons alors avoir une conviction suffisante que l'objet suivi est un visage pour pouvoir initialiser le suivi par histogrammes de couleurs.

Ceci termine la description du détecteur de visages que nous avons implémenté. Le chapitre suivant est consacré à l'exposition de ses performances et tente de donner une série de pistes à suivre pour les améliorer.

Chapitre 4

Performances obtenues et améliorations possibles

Nous analysons dans ce chapitre les performances obtenues par notre détecteur. Nous donnons également quelques pistes à explorer en vue de les améliorer.

La section 4.1 est consacrée aux performances en termes de vitesse d'exécution tandis que la section 4.2 s'intéresse à la qualité de détection obtenue. Nous proposerons quelques pistes à explorer en vue d'améliorer cette dernière à la section 4.3.

4.1 Vitesse d'exécution

Le programme a été développé sous Linux sur un Pentium 4 cadencé à 1.7Ghz et équipé de 512 mégas de mémoire vive. Le langage de programmation utilisé est le C. Les images à traiter sont stockées sur le disque-dur. Elles ont été prises avec une caméra mini-DV. Leur résolution est de 720×576 pixels.

Dans ces conditions, le nombre moyen d'images traitées par seconde oscille entre 5 et 10 suivant la séquence vidéo traitée. Une analyse des performances au moyen de `gprof` révèle que les trois fonctions permettant la mise à jour de l'arrière-plan, la détection de pixels de couleur peau et le suivi par histogrammes de couleurs consomment à elles seules près de 70% du temps d'exécution du programme. Nous avons donc particulièrement veillé à ce que le code de ces trois fonctions soit le plus rapide possible.

Le nombre d'images par seconde est fonction du taux d'utilisation du suivi par histogrammes de couleurs. Il tourne aux alentours de 10 pour les séquences dans lesquelles on y fait peu appel et aux alentours de 5 dans celles où il est beaucoup utilisé.

Le programme a également été testé sur un bi-Xeon cadencé à 3.4Ghz. Des images de 640×480 pixels sont ici directement obtenues à partir d'une caméra FireWire. Le nombre d'images traitées par seconde tourne ici, en moyenne, aux alentours de 13.

Notre détecteur est donc, comme nous le souhaitions, suffisamment rapide pour être utilisé dans le cadre d'une application interactive.

4.2 Qualité de la détection

Les nombres de visages manqués et de fausses détections varient fortement en fonction de la séquence vidéo traitée mais sont en moyenne relativement bas.

Afin d'analyser les performances en termes de qualité de détection, nous commençons par donner dans le tableau 4.1 les taux de détections correctes et de fausses détections pour trois séquences video.

Ces trois séquences sont situées dans le répertoire `/videos` du DVD fourni avec ce travail[1]. Nous invitons le lecteur à les visionner afin de se rendre compte des performances de notre détecteur.

Il nous faut cependant préciser ce que nous entendons exactement par taux de détections correctes et taux de fausses détections :

- Le taux de détections correctes est égal au rapport entre le nombre moyen de visages correctement détectés dans chaque image et le nombre moyen de visages présents dans chaque image.
- Le taux de fausses détections est égal au nombre moyen de visages faussement détectés dans chaque image.

Nom	Taux de détections correctes	Taux de fausses détections
Séquence 1	90%	3,4%
Séquence 2	95%	33%
Séquence 3	75%	0%

TAB. 4.1 – Indicateurs de performances pour 3 séquences vidéo.

Nous allons maintenant, pour chacune des trois séquences, revenir sur les bonnes et les fausses détections afin d'exposer les forces et les faiblesses de notre technique de détection.

Séquence 1

Les performances de détection obtenues dans cette séquence sont très bonnes. Neuf visages sur dix y sont détectés correctement et le nombre de fausses détections y est très faible.

La plupart des visages non-détectés le sont à cause de leur taille trop petite. Dans cette séquence, cette situation se présente lorsque des personnes entrent par le fond de la scène (voir figure 4.2).

Les fausses détections sont ici dues quasi exclusivement au suivi par histogrammes de couleurs. En effet, lorsqu'un visage quitte le champ de la caméra alors qu'il est suivi de cette façon, le détecteur n'a pas la possibilité de savoir que le visage n'est plus dans l'image puisque cette technique de suivi recherche la nouvelle position d'un motif d'une image par rapport à l'autre quel que soit celui-ci. Lorsque le visage sort de l'image, le système suit donc la partie du décor située au bord de celle-ci durant un certain nombre d'images

[1]Le DVD contient également le code source du programme ainsi qu'une séquence sur laquelle le tester.

(voir figure 4.1). Si nous n'avions pas imposé que ce suivi ne puisse être fait que durant un nombre limité d'images consécutives, il y aurait eu une fausse détection permanente à chaque bord de l'image par lequel est sorti un visage suivi par histogrammes de couleurs.

Le suivi par histogrammes de couleurs montre tout son intérêt dans cette séquence. Il permet, comme prévu, de ne par perdre des visages qui ne sont momentanément plus face à la caméra. Ceci est illustré à la figure 4.3.

L'exactitude de l'emplacement du visage décroît cependant lorsque le nombre d'images consécutives où le suivi est effectué de la sorte augmente ou lorsque la taille du visage varie (voir figure 4.4). Ceci montre également l'intérêt de limiter le nombre d'images consécutives au cours duquel celui-ci peut être effectué.

Une dernier problème survenant dans cette séquence se présente lors de l'arrivée d'une personne portant des vêtements de ton beige clair que notre système considère de la couleur de la peau (voir figure 4.5). Ils forment avec le visage une large zone de peau elliptique. Dans ce cas de figure, il existe deux possibilités : soit la recherche de la zone yeux-sourcils échoue (à raison) et le visage n'est pas du tout détecté, soit elle réussit (à tort) et le visage se voit associé à une zone elliptique bien trop grande mais dont il fait quand même partie.

La recherche de la zone yeux-sourcils, si elle permet d'éliminer un nombre important de fausses détections, n'élimine pas l'ensemble de celles-ci puisque la zone elliptique comprenant les vêtements et le visage est sélectionnée par le système alors qu'elle n'est normalement pas supposée l'être.

Séquence 2

Dans cette séquence, le taux de détections correctes est également très élevé. La quasi totalité des visages présents dans l'image sont détectés.

Le taux de fausses détections est, par contre, plus significatif : un tiers des images en comporte une.

Celles-ci sont quasi toutes dues à la même zone de l'image. C'est de nouveau un objet de couleur beige (un meuble en bas à gauche de l'image sur la figure 4.6) qui pose problème.

Celui-ci est "mis en mouvement" par une occlusion momentanée (la personne passe devant lui) et est détecté comme visage lorsqu'il réapparaît dans l'image. Le processus de suivi s'amorce alors et empêche le système d'inclure cet objet dans l'arrière-plan conduisant ainsi à une fausse détection permanente de l'objet.

Tous les objets de couleur beige ne donnent cependant pas lieu à des fausses détections. Ainsi, dans cette séquence, la porte de couleur beige située dans l'arrière-plan est deux fois mise en mouvement et ne donne pas lieu à une seule fausse détection.

Les autres fausses détections sont dues à la détection d'une zone de peau elliptique d'un bras (voir figure 4.6). Mais cette fausse détection ne dure que quelques images puisque le bras ne reste dans la position particulière donnant lieu à sa détection que quelques instants.

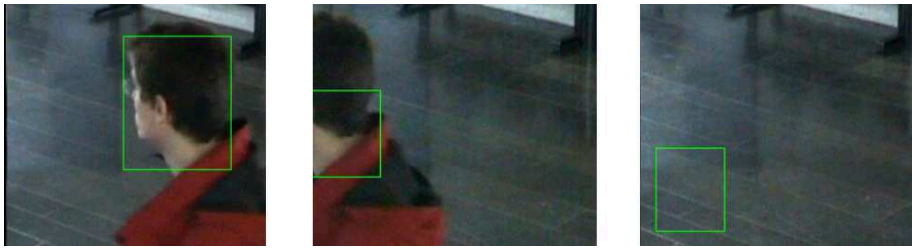

FIG. 4.1 – Suivi par histogrammes de couleurs alors que le visage est sorti de l'image.

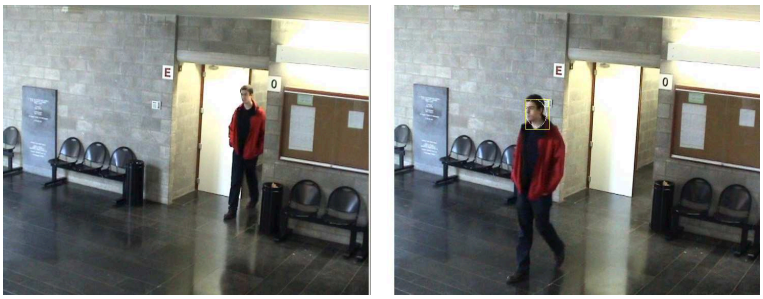

FIG. 4.2 – Non-détection du visage tant que la taille de celui-ci n'est pas assez grande.

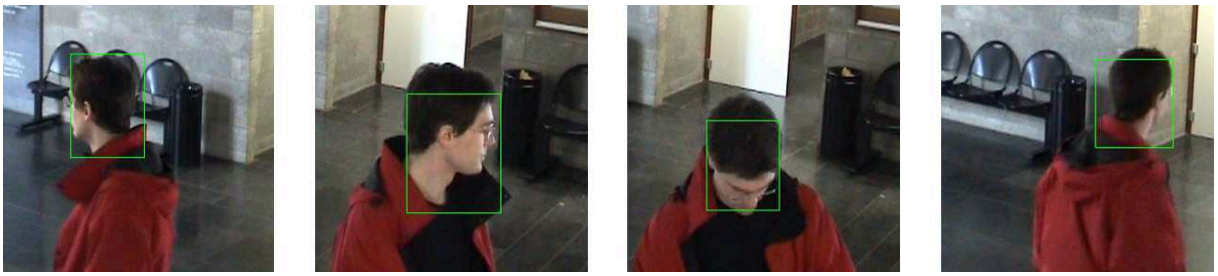

FIG. 4.3 – Suivi correct par histogrammes de couleurs d'un visage qui n'est plus face à la caméra.

FIG. 4.4 – Baisse de l'exactitude du suivi par histogrammes lorsque la taille du visage suivi varie.

FIG. 4.5 – Détection erronée de vêtements comme étant de la couleur de la peau.

FIG. 4.6 – A gauche : fonctionnement correct du système. Au centre : fausse détection d'un bras. A droite : fausse détection d'un objet beige mis en mouvement par une occlusion momentanée (les pixels verts sont ceux détectés comme étant de la couleur de la peau et en superposition sur l'arrière-plan).

Séquence 3

Cette séquence ne comporte aucune fausse détection. Par contre, le taux de détections correctes est plus bas que dans les deux autres : près de 25% des visages ne sont pas détectés.

Ceci est à nouveau dû à notre détection de pixels de la couleur de la peau. Le problème est cette fois-ci que certains pixels du visage ne sont pas détectés comme étant de la couleur de la peau (voir figure 4.7). A un moment, le système finit par "perdre" le visage et, celui-ci étant relativement peu mobile, il se retrouve inclus dans l'arrière-plan. Il faut alors attendre que celui-ci se remette en mouvement pour qu'il soit à nouveau détecté et suivi. Ceci explique la non-détection du visage durant un quart de la séquence.

Le suivi par histogrammes de couleurs montre à nouveau ici son utilité. Il permet en effet de continuer à suivre le visage durant une vingtaine d'images après que le nombre de pixels de couleur peau soit devenu insuffisant pour permettre le suivi du visage en les utilisant. De plus, sur un visage peu mobile comme ici, la qualité de son suivi ne se dégrade pas au cours du temps comme précédemment.

Appréciation générale

Comme nous venons de le voir, si la détection et le suivi effectués par notre système ne sont pas exempts de quelques défauts (comme toutes les techniques de détection), ils sont, en général, fiables.

En effet, la quasi totalité des visages sont détectés, parfois après quelques images seulement, et leur suivi est globalement robuste.

4.3 Améliorations possibles

Nous proposons dans cette section trois types d'améliorations possibles permettant d'augmenter les performances de notre détecteur en termes de taux de détections correctes et de fausses détections.

4.3.1 Une meilleure détection de la couleur de la peau

Nous avons volontairement choisi la technique de détection de la couleur de la peau la plus simple afin de minimiser le nombre de calculs nécessaires à son accomplissement.

Les résultats obtenus par celle-ci ne sont cependant pas toujours à la hauteur de nos espérances. Un grand nombre de pixels sont en effet faussement classifiés comme étant de la couleur de la peau. De plus, il arrive également que certains pixels de peau ne soient pas détectés.

Il serait donc intéressant d'utiliser une technique plus sophistiquée permettant une détection plus précise des pixels de la couleur de la peau. Ceci se ferait sans doute au prix d'une baisse du nombre d'images traitées par seconde mais pourrait permettre une amélioration significative de la qualité de la détection.

4.3.2 Une meilleure détection des traits du visage

Notre détection de la zone yeux-sourcils permet l'élimination d'un grand nombre de fausses détections. Elles ne sont cependant pas toutes éliminées.

On pourrait augmenter ce nombre en recherchant également d'autres traits tels que le nez ou la bouche. Une série de tests nous a conduits à la conclusion que ceux-ci ne donnaient pas lieu à des minima locaux suffisamment marqués pour être détectés de manière fiable. Une série d'opérations morphologiques, telles que celles décrites dans [48], effectuées sur la zone candidate devraient permettre l'accentuation de ces minima et rendre ainsi possible la détection fiable de ces autres traits.

Ne retenir que les zones candidates comportant un nombre suffisant de traits détectés diminuerait certainement le nombre de fausses détections. Ceci se ferait de nouveau au prix d'un accroissement du nombre de calculs à effectuer. Il ne devrait cependant pas être prohibitif puisque le nombre de zones candidates encore présentes à ce stade est relativement faible.

4.3.3 Utilisation d'un classificateur

Notre détecteur ne fait appel à aucun des outils introduits dans la section 2.4.

Si leur application brutale à toutes les fenêtres de l'image donnerait lieu à des temps de calcul extrêmement longs, leur utilisation en guise d'étape finale de validation pourrait éliminer les dernières fausses détections. Ils ne seraient alors appliqués qu'à un nombre très limité de zones candidates et ne devraient donc pas trop affecter le nombre d'images traitées par seconde.

Nous pensons qu'une analyse en composantes principales conviendrait particulièrement bien. Il n'est en effet pas nécessaire de recourir à un classificateur plus sophistiqué puisque la majeure partie du travail de sélection a déjà été effectuée.

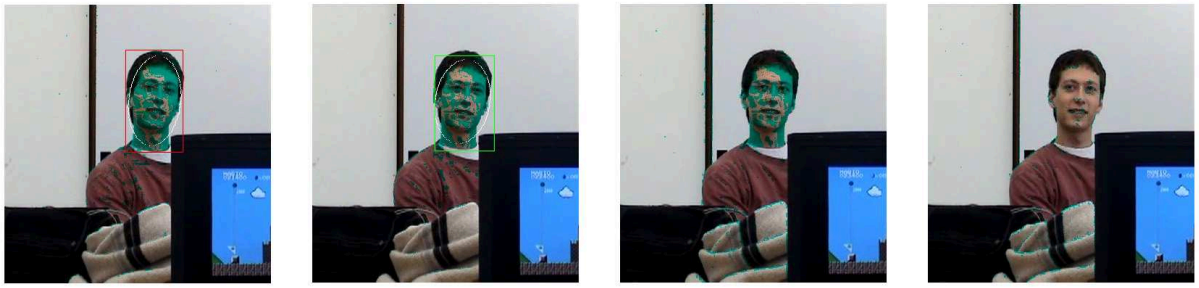

FIG. 4.7 – Perte d'un visage suite à la non-détection de pixels de la couleur de la peau : le système commence par suivre le visage grâce aux histogrammes de couleurs avant de le perdre, il est alors inclus dans l'arrière-plan (les pixels verts sont ceux détectés comme étant de la couleur de la peau et en superposition sur l'arrière-plan).

Chapitre 5

Conclusions

La détection de personnes et de visages dans une séquence vidéo constitue une étape fondamentale dans un grand nombre d'applications. Parmi celles-ci on trouve le développement d'interfaces plus naturelles et efficaces entre l'être humain et l'ordinateur.

Le but de ce travail de fin d'études était la conception d'un détecteur de personnes et de visages destiné à être employé dans ce cadre. Il devait donc être capable de traiter un nombre suffisant d'images par seconde pour permettre son utilisation dans une application interactive.

L'application visée par notre système nous a fait nous tourner vers un détecteur de visages uniquement.

Nous avons commencé par une recherche bibliographique essentiellement consacrée aux nombreux types de techniques de détection de visages existants. Nous y avons sélectionné une série d'outils que nous avons ensuite utilisés pour mettre au point notre propre détecteur. La plupart de ces outils ont été choisis pour le nombre relativement faible de calculs qu'ils impliquent en regard de leurs performances.

Nous avons ensuite mis au point notre méthode de détection. Celle-ci comporte trois modules : un module de sélection de zones d'intérêt, un module de détection de visages vus de face et un module de suivi de visages.

Le module de sélection de zones d'intérêt ne retient que les pixels qui sont d'une part de la couleur de la peau et d'autre part en superposition sur un arrière-plan que nous actualisons à chaque image.

Le module de détection frontale consiste principalement à retenir les zones d'intérêt de forme elliptique à l'intérieur desquelles la zone yeux-sourcils est détectées à la bonne place.

Le module de suivi de visages fonctionne soit en utilisant les mêmes outils que le module de détection frontale, soit en utilisant exclusivement des histogrammes de couleurs.

Ce deuxième type de suivi est utilisé en cas d'échec du premier.

Nous avons ensuite estimé les performances de notre détecteur. Il est capable de traiter en moyenne jusqu'à 13 images par seconde. Testé sur trois séquences videos, le taux de détections correctes varie entre 75 et 95% pour des taux de fausses détections allant de 0 à 33%. Le choix judicieux de traitements peu coûteux en temps de calcul nous a donc permis de conjuguer vitesse d'exécution et qualité de détection.

Nous terminons en donnant trois moyens permettant l'amélioration éventuelle de ses performances : une meilleure détection de la peau, une recherche plus évoluée de traits du visage ainsi que l'emploi d'un classificateur en guise d'étape finale.

Annexe A

Le classificateur idéal de Bayes

On introduit la fonction de coût $L(i \to j)$ qui représente le coût d'attribuer la classe i à un objet de classe j. Pour un classificateur binaire, le risque total associé à la règle de classification r est

$$R(r) = P(1 \to 2|utilisant\, r)L(1 \to 2) + P(2 \to 1|utilisant\, r)L(2 \to 1).$$

La meilleure stratégie possible est celle qui minimise R [13].

Si pour $L(i \to j)$ on choisit :

$$L(i \to j) = \begin{cases} 1 & i \neq j \\ 0 \quad si & i = j \\ d < 1 & \text{pas de décision} \end{cases}$$

Le *classificateur de Bayes* est le classificateur présentant le plus petit risque associé possible. Il est défini par l'algorithme suivant dans lequel $P(k|\mathbf{x})$ représente la probabilité d'attribuer la classe k à l'objet $\mathbf{x}$:

- Si $P(k|\mathsf{x}) > P(i|\mathsf{x})$ pour tout $i \neq k$ et si $P(k|\mathsf{x}) > 1 - d$ alors attribuer le type k à l'objet testé.
- S'il existe plusieurs classes $k_1, k_2, \ldots, k_j$ pour lesquelles $P(k_1|\mathsf{x}) = P(k_2|\mathsf{x}) = \ldots = P(k_j|\mathsf{x}) > P(i|\mathsf{x})$ pour tout i différent de $k_1, k_2, \ldots, k_j$, choisir uniformément et au hasard entre $k_1, k_2, \ldots, k_j$.
- Si $\forall k$ on a $P(k|\mathsf{x}) \leq 1 - d$, refuser de choisir.

Annexe B

Les machines à vecteurs de support : description théorique

On construit ici un classificateur binaire. L'objectif est de générer, à l'intérieur de l'espace des caractéristiques, l'hyper-plan séparant le mieux les instances positives des instances négatives.

Commençons par faire l'hypothèse que les données sont linéairement séparables et supposons que nous disposons, en guise de base de données, d'un ensemble de N points $\mathbf{x_i}$ appartenant à chacune des deux classes[1]. On dispose donc de N doublets $(\mathbf{x_i}, l_i)$, avec l_i valant 1 ou -1 suivant la classe à laquelle appartient le point.

Si les données sont linéairement séparables, cela signifie qu'il existe un hyper-plan séparant les deux classes. On a donc

$$l_i \left(\mathbf{w} \cdot \mathbf{x_i}\right) + b > 0$$

pour chacun des N points. L'hyper-plan optimal est obtenu en joignant $\mathbf{x_a}$ et $\mathbf{x_b}$, les deux points de classes différentes qui sont les plus proches et en choisissant l'hyper-plan perpendiculaire au segment qui les relie. On peut également, sans perte de généralité, considérer que

$$l_i \left(\mathbf{w} \cdot \mathbf{x_i}\right) + b \geq 1$$

avec l'égalité réalisée pour $\mathbf{x_a}$ et $\mathbf{x_b}$ (il suffit de multiplier $\mathbf{w}$ et b par une constante adéquate). On a donc, par exemple :

$$\left(\mathbf{w} \cdot \mathbf{x_a}\right) + b = 1 \qquad\qquad \left(\mathbf{w} \cdot \mathbf{x_b}\right) + b = -1$$

et donc

$$\mathbf{w} \cdot \left(\mathbf{x_a} - \mathbf{x_b}\right) = 2.$$

[1] Cette section est largement inspirée par [13].

Nous cherchons à maximiser

$$\begin{aligned}\text{dist}(\mathbf{x_a}, \text{hyper-plan}) + \text{dist}(\mathbf{x_b}, \text{hyper-plan}) &= \left(\frac{\mathbf{w}}{|\mathbf{w}|} \cdot \mathbf{x_a} + \frac{b}{|\mathbf{w}|}\right) - \left(\frac{\mathbf{w}}{|\mathbf{w}|} \cdot \mathbf{x_b} + \frac{b}{|\mathbf{w}|}\right) \\ &= \frac{\mathbf{w}}{|\mathbf{w}|} \cdot (\mathbf{x_a} - \mathbf{x_b}) \\ &= \frac{2}{|\mathbf{w}|,}\end{aligned}$$

ce qui revient à minimiser $\frac{1}{2}\,\mathbf{w} \cdot \mathbf{w}$.

Le problème de trouver un hyper-plan séparant un ensemble de données linéairement séparables peut donc être résumé par :

$$\textit{Minimiser } \tfrac{1}{2}\,\mathbf{w} \cdot \mathbf{w} \textit{ sous la contrainte } l_i\,(\mathbf{w} \cdot \mathbf{x_i}) + b \geq 1.$$

Ce problème peut être résolu par l'introduction de multiplicateurs de Lagrange et être ramené à un problème de *programmation quadratique*. Les bibliothèques standards de résolution de ce type de problèmes peuvent être employées mais la configuration particulière prise par les données dans le cas des SVM a conduit au développement de techniques plus efficaces et moins gourmandes en mémoire vive [35].

Se limiter à des données linéairement séparables est bien évidemment trop restrictif. On peut généraliser la méthode à des données non linéairement séparables en introduisant des variables de relâchement $\xi_i \geq 0$ (slack variables) qui quantifient à quel point la contrainte est violée. Le problème devient :

$$\textit{Minimiser } \tfrac{1}{2}\,\mathbf{w} \cdot \mathbf{w} + C\,\textstyle\sum_{i=1}^{N} \xi_i \textit{ sous la contrainte } l_i\,(\mathbf{w} \cdot \mathbf{x_i}) + b \geq 1 - \xi_i$$

où C est un paramètre qui permet d'ajuster le poids accordé aux violations des contraintes.

Il est également fréquent de considérer qu'il est préférable de commencer par projeter les données sur un sous-espace non-linéaire de plus grandes dimensions et d'effectuer la séparation au sein de ce sous-espace. Si la projection est définie par $\mathbf{x} \rightarrow \Phi(\mathbf{x})$ et qu'il existe une fonction noyau K telle que $K(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})$, alors la machine à vecteurs de support non-linéaire peut être obtenue en appliquant les règle énoncées précédemment en remplaçant le produit scalaire $\mathbf{x} \cdot \mathbf{y}$ par la fonction noyau $K(\mathbf{x}, \mathbf{y})$.

Une fois la surface de décision générée, la règle de décision pour un point à classifier $\mathbf{x}$ est alors tout simplement

$$y(\mathbf{x}) = \text{sign}\,((\mathbf{w} \cdot \mathbf{x}) + b)\,.$$

Annexe C

Les réseaux de neurones artificiels : description théorique

Un réseau de neurones sert à approcher une fonction vectorielle $\mathbf{f}(\mathbf{x})$ par une série de *couches* de neurones. Chaque couche produit un vecteur de sortie obtenu en appliquant la même fonction non-linéaire, que nous nommerons ϕ, à différentes fonctions affines de ses entrées. Nous ajouterons systématiquement un élément additionnel aux entrées que nous fixerons à 1. Ceci nous permettra de considérer que nous obtenons en sortie de chaque couche une fonction *linéaire*, et non plus affine, de ses entrées. Ce qui signifie que si une couche possède un vecteur d'entrée (auquel nous avons rajouté un élément unitaire) $\mathbf{u}$ et un vecteur de sortie $\mathbf{v}$ (voir figure C.1), nous pouvons écrire

$$\mathbf{v} = [\phi(\mathbf{w_1} \cdot \mathbf{u}), \phi(\mathbf{w_2} \cdot \mathbf{u}), \ldots \phi(\mathbf{w_n} \cdot \mathbf{u})],$$

où les $\mathbf{w_i}$ sont les paramètres qui peuvent être ajustés afin d'effectuer l'approximation.

En règle générale, un réseau de neurones comporte plusieurs couches pour approcher une fonction. Chaque couche utilise un vecteur d'entrée adjoint d'un élément unitaire. Si par exemple, nous approchons la fonction $\mathbf{g}(\mathbf{x})$ avec un réseau possédant deux couches (voir figure C.2), nous avons :

$$\mathbf{g}(\mathbf{x}) \approx \mathbf{f}(\mathbf{x}) = [\phi(\mathbf{w_{21}} \cdot \mathbf{y}), \phi(\mathbf{w_{22}} \cdot \mathbf{y}), \ldots \phi(\mathbf{w_{2n}} \cdot \mathbf{y})],$$

où

$$\mathbf{y}(\mathbf{z}) = [\phi(\mathbf{w_{11}} \cdot \mathbf{z}), \phi(\mathbf{w_{12}} \cdot \mathbf{z}), \ldots \phi(\mathbf{w_{1m}} \cdot \mathbf{z})]$$

et

$$\mathbf{z}(\mathbf{x}) = [x_1, x_2, \ldots, x_p, 1].$$

Le réseau de la figure C.2 est un réseau *totalement connecté*. Il existe également des réseaux *partiellement connectés* où toutes les connections inter-neuronales ne sont pas présentes. Mathématiquement, certains éléments des w_{ij} sont alors nuls. Par exemple, si le deuxième élément de w_{13} est nul, cela signifie que la connection reliant x_2 à y_3 est absente.

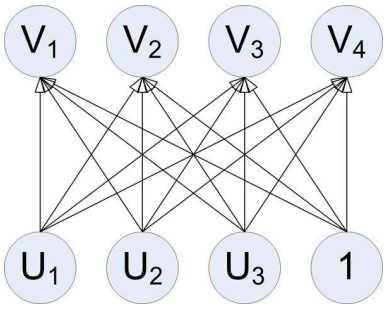

FIG. C.1 – Exemple pour $n = 4$ et un vecteur d'entrée à trois dimensions. On a, par exemple, pour V_1 : $V_1 = \phi(a\,U_1 + b\,U_2 + c\,U_3 + d)$.

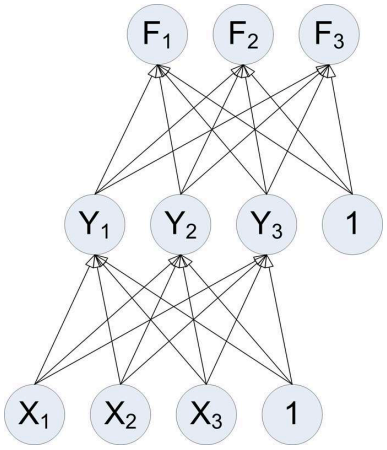

FIG. C.2 – Exemple de réseau à deux couches *totalement connecté* avec $m = n = p = 3$.

Il existe également une construction similaire pour les réseaux à trois couches. Les réseaux à plus de trois couches sont peu communs.

Il existe un grand choix de non-linéarité ϕ. On peut par exemple utiliser une fonction échelon sign(x) mais il est plus courant d'utiliser une fonction passant progressivement (mais relativement rapidement) de 0 à 1. Une fonction souvent choisie est la tangente hyperbolique $\phi(x; v\,A) = A\,\tanh(vx)$.

Afin de produire un réseau approchant une fonction $\mathbf{g}(\mathbf{x})$, nous utilisons un ensemble de paires comportant chacune un exemple d'entrée et le vecteur de sortie du réseau désiré pour cette entrée $(\mathbf{x^e}, \mathbf{o^e})$ qui constitue la base de données à partir de laquelle nous entraînons notre réseau. Typiquement, $\mathbf{o^e} = \mathbf{g}(\mathbf{x^e})$. Nous désignons notre réseau par $\mathbf{n}(\mathbf{x^e}; \mathbf{p})$ où $\mathbf{p}$ est un vecteur contenant l'ensemble des $\mathbf{w_{ij}}$.

Nous recherchons $\widehat{\mathbf{p}}$ qui minimise

$$\text{Erreur}(\mathbf{p}) = \sum_e |\mathbf{n}(\mathbf{x^e}; \mathbf{p}) - \mathbf{o^e}|^2.$$

Dans le cadre de la classification, nous cherchons le plus souvent à approcher la probabilité a posteriori de la classe quand on connaît les caractéristiques. Nous ne connaissons pas la valeur de cette probabilité et ne pouvons donc pas la fournir à l'algorithme d'entraînement. Afin de remédier à ce problème, nous choisissons, en guise de sortie désirée, un vecteur contenant autant d'éléments qu'il y a de classes et ne comportant que des zéros à l'exception d'un élément qui vaut 1. C'est bien évidemment la position de ce 1 dans le vecteur qui nous renseigne sur la classe de l'objet considéré. Une fois le réseau entraîné, un nouvel élément $\mathbf{x}$ sera classifié en calculant $\mathbf{n}(\mathbf{x}; \widehat{\mathbf{p}})$ et en choisissant la classe correspondant à l'élément le plus grand de ce vecteur.

La minimisation de Erreur$(\mathbf{p})$ est effectuée itérativement en descendant le long du gradient de celle-ci.

$$\mathbf{p_{i+1}} = \mathbf{p_i} - \epsilon(\nabla \text{Erreur})$$

Afin de ne pas avoir à calculer ce gradient à chaque itération, nous pouvons nous contenter de choisir, à chaque itération, un exemple différent, calculer le gradient *pour cet exemple* (ce qui est beaucoup moins coûteux en calculs) et descendre le long de celui-ci. L'erreur ne diminue pas nécessairement à chaque itération mais en itérant un nombre suffisant de fois et en choisissant ϵ suffisamment petit, nous minimiserons finalement quand même Erreur$(\mathbf{p})$. Une technique efficace de calcul de ce gradient est la technique dite de 'rétro-propagation' (*backpropagation*) décrite dans la section 22.9 de [13].

Bibliographie

[1] M. Pontil B. Heisele, T. Poggio. Face detection in still gray images. AI Memo 1687, Center for Biological and Computational Learning, MIT, Cambridge, Mai 2000.

[2] Shumeet Baluja. Face detection with in-plane rotation : early concepts and preliminary results. Technical report, Justsystem Pittsburgh Research Center, 1997.

[3] Petr Bílek. Face localization from discriminative regions. *Research Reports of CMP, Czech Technical University in Prague*, 24, 2001.

[4] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 232–237, Santa Barbara, California, june 1998.

[5] I. Buciu, C. Kotropoulos, and I. Pitas. Combining support vector machines for accurate face detection. In *2001 IEEE International Conference on Image Processing*, pages 1054–1057, Thessaloniki, Greece, Octobre 2001.

[6] A. Colmenarez and T. Huang. Maximum likelihood face detection. In *Proceedings of 2nd International Conference on Automatic Face and Gesture Recognition (FG '96)*, pages 307–311, Killington, Vermont, 1996.

[7] A. Colmenarez and T. Huang. Face detection with informationbased maximum discrimination. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pages 782–787, Puerto Rico, June 1997.

[8] Antonio Colmenarez, Brendan Frey, and Thomas S. Huang. Detection and tracking of face and facial features. In *International Conference on Image Processing*, pages 657–661, Kobe, Japan, 99.

[9] I. Craw, H. Elis, and J. R. Lishman. Automatic extraction of face-features. *Pattern Recognition Letters*, 5(2) :183–187, 1987.

[10] Ian Craw, David Tock, and Alan Bennett. Finding face features. In *European Conference on Computer Vision*, pages 92–96, 1992.

[11] Ying Dai and Yasuaki Nakano. Face-texture model based on sgld and its application in face detection in a color scene. *Pattern Recognition*, 29(6) :1007–1017, 1996.

[12] Teofilo Emidio de Campos, Rogerio Schmidt Feris, and Roberto Marcondes Cesar Junior. Eigenfaces versus eigeneyes : First steps toward performace assessment of representations for face recognition. In *Lecture Notes in Artificial Intelligence*, volume 1793, pages 193–201, April 2000.

[13] David A. Forsyth and Jean Ponce. *Computer Vision : A Modern Approach*. Prentice Hall, 2002.

[14] Erik Hjelmåsa and Boon Kee Low. Face detection : A survey. *Computer Vision and Image Understanding*, 83(3) :236–274, september 2001.

[15] T. Horprasert, D. Harwood, and L.S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *Frame-Rate99*, 1999.

[16] J. Huang, S. Gutta, and H. Wechsler. Detection of human faces using decision trees. In *Proceedings of IEEE Second International Conference on Automatic Face and Gesture Recognition*, pages pages 248–252, October 1996.

[17] Tony Jebara and Alex Pentland. Parameterized structure from motion for 3d adaptative feedback tracking of faces. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 144–150, San Juan, Puerto Rico, juin 1997.

[18] Michael J. Jones and James M. Rehg. Statistical color models with application to skin detection. *Int. J. Comput. Vision*, 46(1) :81–96, 2002.

[19] Bernd Jähne. *Digital Image Processing*. Springer-Verlag, 2002.

[20] M. Kirby and L. Sirovich. Application of the karjunen-loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1) :103–108, 1990.

[21] T. Kohonen. *Self Organizing Map*. Springer, 1996.

[22] C. Kotropoulos and I. Pitas. Rule-based face detection in frontal views. In *Proceedings of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP 97)*, volume IV, pages 2367–2540, Munich, Germany, 1997.

[23] Young Ho Kwon and Niels da Victoria Lobo. Face detection using templates. In *Proceedings of International Conference on Pattern Recognition*, pages 764–767, 1994.

[24] T.K. Leung, M. C. Burl, and P. Perona. Finding faces in cluttered scenes using random labeled graph matching. In *Fifth Intl. Conf. on Comp. Vision*, June 1995.

[25] S. Li, Q. Fu, L. Gu, B. Scholkopf, Y. Cheng, and H. Zhang. Kernel machine based learning for Multi-View face detection and pose estimation. In *Proceedings of 8th IEEE International Conference on Computer Vision.*, pages 674–679, Vancouver, Canada, July 2001.

[26] Cherng Jye Liou. A real time face recognition system. Master's thesis, National Taiwan University, Department of Electrical Engineering, DSP/IC Design Lab, June 1997.

[27] G.G. Mateos and C.V. Chicote. Face detection on still images using hit maps. In *Proceedings of Audio- and Video-Based Biometric Person Authentication, Third International Conference*, page 102, Halmstad, Sweden, 2001.

[28] Stephen J. McKenna, Shaogang Gong, and Yogesh Raja. Modelling facial colour and identity with gaussian mixtures. *Pattern Recognition*, 31(12) :1883–1892, December 1998.

[29] Stephen J. McKenna, Yogesh Raja, and Shaogang Gong. Tracking colour objects using adaptive mixture models. *Image and Vision Computing*, 17(3-4) :225–231, march 1999.

[30] B. Menser and F. Müller. Face detection in color images using principal components analasys, 1999.

[31] Hsuan Yang Ming, N. Abuja, and D. Kriegman. Face detection using mixtures of linear subspaces. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition. (FG 2000)*, pages 70–76, Grenoble, France, March 2000.

[32] Anuj Mohan, Constantine Papageorgiou, and Tomaso Poggio. Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4) :349–361, 2001.

[33] C. Nakajima, M. Pontil, and T. Poggio. People recognition and pose estimation in image sequences. In *Proceedings of IEEE International Joint Conference on Neural Networks*, pages 4189–4196, July 2000.

[34] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pages 193–199, Puerto Rico, June 1997.

[35] E. Osuna, R. Freund, and F. Girosi. Training support vector machines :an application to face detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pages 130–136, Puerto Rico, June 1997.

[36] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *International Conference on Computer Vision (ICCV'98)*, pages 555–562, Bombay, India, Janvier 1998.

[37] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, WA, June 1994.

[38] Vlad Popovici and J.P. Thiran. Face detection using an svm trained in eigenfaces. In *Audio- and Video-Based Biometric Person Authentication*, pages 190–198, 2003.

[39] D. Roth, M. Yang, and N. Ahuja. A snow-based face detector. In *Advances in Neural Information Processing Systems 12 (NIPS 12)*, Cambridge, May 2000.

[40] Henry Rowley, Shumeet Baluja, and Takeo Kanade. Rotation invariant neural network-based face detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 1998.

[41] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Human face detection in visual scenes. In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 875–881. The MIT Press, 1996.

[42] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1) :23–38, 1998.

[43] H. Sahbi and N. Boujemaa. Accurate face detection based on coarse segmentation and fine skin color adaption. In *International Conference on Image and Signal Processing*, Agadir, Morocco, 2001.

[44] D. Saxe and R. Foulds. Toward robust skin identification in video images. In *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pages 379–384, 1996.

[45] Pawan Sinha. Object recognition via image-invariants. http ://www.ai.mit.edu/projects/cbcl/web-pawan/src/paper.ps.

[46] Pawan Sinha. Object recognition via image invariants : A case study. *Investigative Ophthalmology and Visual Science*, 35 :1735–1740, May 1994.

[47] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America*, 4(3) :519–524, 1987.

[48] K. Sobottka and I. Pitas. Extraction of facial regions and features using color and shape information. In *Int. Conf. on Pattern Recognition (ICIP)*, Vienna, Austria, August 1996.

[49] K. Sobottka and I. Pitas. A novel method for automatic face segmentation, facial feature extraction and tracking. *Signal Processing : Image Communication*, 12(3) :263–281, June 1998.

[50] S. Spors and R. Rabenstein. A real-time facetracker for color video. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001.

[51] Kah Kay Sung and Tomaso Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1) :39–51, 1998.

[52] J. Terrillon and S. Akamatsu. Comparative performance of different chrominance spaces for color segmentation and detection of human faces in complex scene images. In *Proceedings of International Conf on Face and Gesture Recognition*, pages 54–61, 2000.

[53] J. Terrillon, M. David, and S. Akamatsu. Automatic detection of human faces in natural scene images by use of a skin color model and of invariant moments. In *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition*, pages 112–117, Nara, Japan, 1998.

[54] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1) :71–86, 1991.

[55] Marc Van Droogenbroeck. Traitement numérique des images. Notes de cours, Université de Liège, 225 pages, Septembre 2001. PDF available on request, HTML version available on line.

[56] R.R. Wang, T. Huang, and J. Zhong. Generative and discriminative face modelling for detection. In *Proceedings of IEEE Fifth International Conference on Automatic Face and Gesture Recognition*, pages 266–271, May 2002.

[57] G. Yang and T.S. Huang. Human face detection in complex background. *Pattern recognition*, 27(1) :53–63, 1994.

[58] Jie Yang and Alex Waibel. A real-time face tracker. In *Proceedings. 3rd IEEE Workshop on Applications of Computer Vision*, pages 142–147, 1996.

[59] M.-H. Yang and N. Ahuja. Detecting human faces in color images. In *Proceedings of the 1998 IEEE International Conference on Image Processing (ICIP 98)*, pages 127–139, Chicago, 1998.

[60] Ming-Hsuan Yang, David J. Kriegman, and Narendra Ahuja. Detecting faces in images : A survey. *IEEE Transactions on pattern analysis and machine intelligence*, 24(1) :34–58, 2002.

[61] K. Yow and R. Cipolla. Feature-based human face detection. Technical report, University of Cambridge, Department of Engineering, England, 1996.

[62] Kin Choong Yow and Roberto Cipolla. Enhancing human face detection using motion and active contours. In *Proceedings of the Third Asian Conference on Computer Vision*, volume 1, pages 515–522, 1998.

[63] A.L. Yuille, D.S. Cohen, and P. Hallinan. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2) :99–112, 1992.

[64] Liang Zhao and Charles Thorpe. Stereo and neural network-based pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems*, 1(3) :148–154, September 2000.