THESE

présentée à

L'UNIVERSITÉ PARIS 6

Pour obtenir le titre de

DOCTEUR EN SCIENCES

Spécialité

PROBABILITÉS ET APPLICATIONS

Par

François FLEURET

Sujet

DÉTECTION HIÉRARCHIQUE DE VISAGES PAR APPRENTISSAGE STATISTIQUE

Soutenue le 4 Janvier 2000 devant un jury composé de :

MM.	**Marc**	**Yor**	Président
	Laurent	**Younes**	Rapporteur
	Yali	**Amit**	
	Donald	**Geman**	Examinateur
	Olivier	**Catoni**	
	Stéphane	**Mallat**	
	Henri	**Maitre**	

À mes parents,
et à Dominique,

Remerciements

Je tiens à exprimer ma gratitude et mes plus vifs remerciements aux membres du jury :

- à Marc Yor, qui m'a fait l'honneur d'être le président du jury ;
- à Yali Amit et Laurent Younes, pour avoir accepté d'être rapporteurs, et dont les remarques et conseils ont été très enrichissants ;
- à Donald Geman, pour avoir été pendant trois ans un directeur de thèse idéal, humainement, et scientifiquement ;
- à Olivier Catoni, Stéphane Mallat et Henri Maitre, pour avoir accepté de faire partie de mon jury ;

Ainsi qu'à André Gagalowicz et à Chahab Nastar pour m'avoir accueilli dans leurs projets à l'INRIA de Rocquencourt.

Table des matières

Chapitre 1

Introduction

1.1 Problème de la détection

La détection d'objets génériques, et plus particulièrement la détection de visages, est une fonction cognitive que le cerveau réalise très efficacement aussi bien en terme de vitesse qu'en terme d'erreur. Cette tâche n'a jusqu'ici pas trouvé de réponse algorithmique comparable, même implémentée sur des ordinateurs rapides.

Par "détection d'objets", on entend la tâche consistant à indiquer sur une image toutes les occurences de l'objet recherché, ce dernier pouvant être une classe générique (visages, voitures, silhouettes, etc.). Ainsi définie, la détection peut se poser de multiples manières différentes, en fonction d'une part des capteurs dont sont issues les images, et d'autre part de la finalité de la détection.

Les capteurs peuvent fournir une image couleur ou noir et blanc, plusieurs images de la même scène (images stéréoscopiques par exemple), des séquences animées (avec une fréquence plus ou moins élevée), voir même des informations hors du spectre du visible (infra-rouge, etc.). La finalité quant à elle fixe les contraintes en terme de vitesse (quelle est la puissance de calcul de l'électronique disponible, et doit-on traiter

l'information en temps réel), et en terme de taux d'erreur.

L'approche proposée dans cette thèse est générique, et répond de manière assez globale au problème de la détection d'objets ; néanmoins, toutes nos expérimentations concernent spécifiquement la détection de visages, nous nous focaliserons donc sur ce cas particulier.

Le développement des bases de données d'images, et plus précisément du WEB, crée un besoin d'automatisation de requêtes par le contenu sur des bases importantes. De même on cherche de plus en plus à automatiser des tâches de recensement (comptage sur des sites touristiques) ou de surveillance. Le problème auquel nous nous intéressons spécifiquement ici est un cas particulier sans redondance dans l'information, la scène se réduisant à une unique image noir et blanc. Nous ne pourrons donc utiliser ni des critères de couleurs (Haiyuan et al. 1999, Sabert & Tekalp 1998, Ming-Hsuan & Ahuja 1999), ni la cohérence temporelle (Graf et al. 1996, Jacquin & Eleftheriadis 1995), ni la profondeur (Ming & Akatsuka 1998), ni enfin des techniques de reconstruction 3D (Jeng et al. 1996). Cette situation correspond à la pire configuration pour des bases de données (qui contiennent souvent des images en couleur), et à la qualité des images acquises par une caméra de vidéo-surveillance standard.

Une scène typique que nous voulons traiter est représentée sur la figure 1.1. Notre algorithme devra être capable, étant donnée une telle scène, de fournir la liste de tous les emplacements (x, y) dans le plan image, situés entre les yeux d'un visage.

La difficulté de cette tâche réside dans la grande variabilité de l'apparence des visages. Ces variations peuvent être regroupées en trois grandes familles :

- Pose : La position dans le plan image d'un visage, sa taille, et enfin son inclinaison peuvent être quelconques.

- Déformations intrinsèques : La forme du crâne, les expressions, la présence de lunettes, la coiffure, etc. peuvent modifier grandement l'apparence d'un visage.

FIG. 1.1: *Exemple de scène. Le but est de détecter tous les visages présents.*

– Illumination : Les variations dans l'éclairage entraînent des variations importantes de l'apparence. Les ombres peuvent créer des bords illusoires, et masquer des bords réels (selon (Ullman 1996), deux images de la même personne illuminée de manières différentes, peuvent être plus "éloignées" - pour de nombreuses métriques - que deux images de personnes différentes) .

1.2 Cadre de travail

Nous voulons, partant d'une base d'apprentissage contenant des images de visages vus de face, construire un algorithme efficace pour détecter et localiser tous les visages dans des images en niveaux de gris. Les exemples de la base sont des images de taille standardisée (par exemple 64×64 pixels) contenant une unique vue frontale d'un visage, à une échelle, une inclinaison, et un éclairage quelconque. Même si les fonds de ces images d'apprentissage sont très simples, l'algorithme doit pouvoir fonctionner dans des scènes naturelle très complexes. Nous nous intéressons à la première étape d'un processus complet de détection ; il s'agit ici de sélectionner très rapidement

FIG. 1.2: *Extrait de la base d'images utilisées pour l'apprentissage.*

un petit nombre d'emplacements sur la scène où devraient être utilisé ensuite un algorithme plus lourd. Le taux de fausses alarmes est donc moins important que l'efficacité algorithmique.

Les visages que nous cherchons à détecter dans les scènes sont ceux vus de face, de tailles telles que la distance entre les yeux soit comprise entre 10 et 80 pixels. Nous estimons les performances de notre algorithme en considérant son coût algorithmique, le taux de faux positifs et la taille de la base de données d'apprentissage nécessaire pour obtenir un taux de faux négatifs très faible.

La base de données que nous avons utilisée au cours de nos travaux contient 300 images extraites de la base de données Olivetti. Quelques exemples d'images de notre ensemble d'apprentissage sont représentés sur la figure (1.2).

Pour ce qui est du coût, l'algorithme devra pouvoir traiter une scène comme celle de la figure (1.1) en moins d'une seconde sur un PC standard. Les temps d'apprentissage ne constituent, pas quant à eux, une contrainte dure.

1.3 Forme hiérachique du détecteur

L'algorithme que nous proposons utilise un détecteur conçu pour traiter une imagette 64×64, admettant une grande tolérance : pour la position du visage dans le plan image (le point situé entre les yeux doit se trouver dans un carré de référence de 8×8 pixels) et pour sa taille (la distance entre les yeux doit être comprise entre 10 et 20 pixels). Il suffit de l'utiliser un nombre de fois suffisant pour que tous les points de l'image soient contenus dans la zone de tolérance 8×8 au moins une fois.

Nous appelons ce détecteur principal *détecteur global*. Il détermine si un visage est présent ou pas dans une imagette, et dans l'affirmative, il détermine sa pose. Cette pose est constituée de la donnée de la localisation dans le plan image du point situé entre les deux yeux, de l'inclinaison de la droite passant par la bouche et ce même point, et enfin de la distance entre les yeux. Ce détecteur global est obtenu en combinant plusieurs détecteurs plus élémentaires, que nous appelons *détecteurs dédiés*. Chacun est dédié à un sous-ensemble d'images de visages dont la pose est plus ou moins contrainte.

Nous ne cherchons donc pas à réaliser un seul détecteur par apprentissage, détecteur qui devrait être capable de gérer les multiples déformations de l'objet à reconnaître. Nous ne voulons pas inférer les invariants globaux de l'objet à partir de la base d'apprentissage. A la place, nous gérons explicitement les variations de la pose en utilisant l'information dont nous disposons sur le positionnement des visages des images de la base d'apprentissage pour construire des détecteurs dédiés à des sous-cas de plus en plus contraints.

Pour construire chacun de ces détecteurs dédiés, nous construisons par apprentissage une succession de tests de complexités croissantes. La détection se fait en s'assurant successivement qu'un nombre minimum de tests de chaque complexité est *vrai*. Ces fragments ont une localisation et une orientation approximatives. Leur définition est volontairement tolérante pour être invariante par des déformations géométriques lo-

cales. Les structures, que nous appellerons *arrangements*, n'ont pas de signification ou d'interprétation géométrique. A la place nous voulons simplement qu'elles soient le plus probables possible sur les visages. Leur spécificité seule les rend très peu probables sur le "fond" (c'est à dire sur les images ne représentant pas un visage).

Pour obtenir cette propriété de probabilité élevée sur les images de visages, nous introduisons la notion d'*arrangement décomposable* et un algorithme pour construire un grand nombre de tels arrangements à partir de l'ensemble d'apprentissage. Un arrangement est la conjonction de la présence d'une famille de fragments de bords. *Décomposable* signifie que l'arrangement peut être mis sous la forme d'une conjonction de deux sous-arrangements très corrélés, qui peuvent à leur tour être décomposés en plus petits arrangements très corrélés, etc. jusqu'aux fragments de bords seuls. L'algorithme de construction est une procédure qui permet de construire des arrangements de plus en plus complexes de manière récursive. La motivation est que la probabilité qu'un arrangement de taille k apparaisse sur un visage décroît très progressivement quand k croît, nous assurant donc que les tests seront très discriminants pour séparer les visages du fond. Cette hypothèse sera justifiée théoriquement et illustrée empiriquement.

Le processus de construction des détecteurs dédiés est le même pour tous les détecteurs. Seul change l'ensemble d'apprentissage qui sert à estimer l'ensemble d'arrangements décomposables. Les poses des visages représentés sur les images présentes dans un de ces ensembles sont contraintes de manière à dédier le détecteur ainsi construit à une sous-partie de l'ensemble complet des poses possibles. On construit ainsi des détecteurs plus ou moins performants et coûteux en fonction de leurs spécificités.

Finalement, le processus est hiérarchisé pour la détermination de la pose, puisque nous avons une série de détecteurs dédiés à des sous-espaces de poses de tailles décroissantes. Il l'est également dans la représentation de l'objet lui même, puisque nous construisons des familles d'arrangements de plus en plus denses et complexes.

En terme d'erreur, comme nous le verrons, le taux de faux positifs est réduit grâce à la

combinaison des détecteurs dédiés qui se comportent de manière quasi-indépendante sur les images de "fond", et rejettent donc successivement une proportion importante de ces images. Le taux de faux négatifs peut être, lui, réduit à une valeur arbitrairement faible en fixant le nombre minimum de tests qui doivent être présents à chaque étape à une valeur peu élevée.

Nous ne cherchons pas, avec cet algorithme, à réaliser le processus complet de détection. Notre critère fondamental, qui consiste en un comptage d'arrangements présents, ne permet pas de rejeter des candidats trop riches en fragments de bords, ce qui est problématique sur des images qui comportent de hautes fréquences spatiales (par exemple des textures). Il permet néanmoins de sélectionner très efficacement des endroits peu nombreux où un algorithme intensif (arbres de décision(Huang et al. 1996), réseau de neurones (Vaillant et al. 1994), etc.) pourrait être utilisé. Nous pourrions également appliquer des heuristiques simples, semblables à celles utilisées par Rowley dans (Rowley 1999), pour regrouper les détections proches les unes des autres.

Le processus que nous décrivons dans cette thèse correspond donc à la "sélection visuelle" décrite dans (Amit & Geman 1999), qui est un précurseur de notre méthode. Ses similitudes avec notre travail sont nombreuses. Tout d'abord les critères discriminants sont comme dans notre algorithme des arrangements particuliers de fragments de bords. Ces arrangements sont également sélectionnés sur des critères statistiques, sans géométrie, et les détecteurs de bords sont les mêmes que les nôtres. Enfin, les tests les plus fins admettent comme dans notre cas une tolérance locale en position.

Contrairement à notre méthode, l'algorithme proposé dans (Amit & Geman 1999) ne gère pas l'efficacité algorithmique par une hiérarchisation de la représentation, mais par une implémentation efficace de la recherche des arrangements de fragments de bords utilisant une transformation de Hough (Rojer & Schwartz 1992). De plus, l'apprentissage consiste à construire un détecteur sur une pose de référence, puis à relaxer le détecteur ainsi obtenu afin de le rendre tolérant à des déformations de cette pose. Comparativement, cette technique repose moins sur l'apprentissage que la nôtre.

1.4 Détection d'objet invariante

Une des principales difficultés réside dans la variation de l'apparence des visages due aux variations d'éclairage ; voir par exemple la discussion dans (Ullman 1996). Notre approche consiste à introduire une grande invariance photométrique dans la définition des fragments de bords en ne considérant que des comparaisons entre des différences d'intensités (cf. section 4.6.2). Une telle définition est invariante pour toute transformation linéaire croissante des niveaux de gris.

De même notre approche de l'invariance géométrique est assez explicite. Chaque arrangement est une conjonction de tests élémentaires, qui sont eux-même des disjonctions de morceaux de bords (cf. 4.6.3). Ainsi, chacun d'entre eux est invariant par des déformations locales de l'image. La manière dont ces arrangements sont combinés, sous la forme d'un comptage, nous assure d'une grande invariance à des dégradations de l'image ou à des occlusions. Quels arrangements sont présents sur l'image n'importe pas, seul leur nombre compte.

Les variations dans la pose globale du visage, enfin, sont gérées explicitement avec la construction de plusieurs détecteurs, chacun dédié à un sous-ensemble de l'espace complet de poses. Nous nous assurons ainsi d'avoir pour tout visage possible un détecteur précis qui lui est dédié.

Cette gestion très explicite de l'invariance géométrique est très différente de la majorité des autres approches ; voir (Amit & Geman 1999) pour une discussion plus complète.

1.5 Coût algorithmique, taille de la base de données

Notre approche n'est pas une implémentation d'un algorithme généraliste déja connu. Nous avons développé une nouvelle approche du problème générique de la détection, de manière à gérer explicitement le coût algorithmique et les nuisances dues à une petite base d'apprentisage.

L'organisation très hiérarchique du détecteur permet d'obtenir un algorithme très efficace en terme de coût. La propriété essentielle que nous utilisons est que le calcul nécessaire à l'évaluation d'une conjonction s'arrête dès que l'on sait que l'un des termes de la conjonction est faux. Ainsi, étant donnée une imagette extraite d'une scène, il y a une grande probabilité qu'il ne soit pas nécessaire d'aller loin dans la hiérarchie de tests pour déterminer qu'elle ne contient pas de visage. Par exemple, une imagette de couleur uniforme, sur laquelle n'apparaît aucun bord, sera rejetée par le premier test du premier détecteur dédié.

De plus, à aucun moment nous n'avons besoin d'un modèle de la loi des images de fond, modèle qui serait très complexe et dont l'estimation demanderait une base d'apprentissage gigantesque. A la place, nous n'utilisons que des estimations de corrélations entre des évènements pour la probabilité des images de visages. Ces estimations sont faites *séparément*, contrairement par exemple à l'estimation des poids dans un réseau de neurones. Une base de taille très raisonnable nous suffit pour éviter des phénomènes de sur-ajustement aux données.

Finalement, la forme originale de cet algorithme a théoriquement un coût optimal sous certaines hypothèses assez faibles, et nous montrerons en pratique que nous atteignons effectivement une vitesse de traitement très élevée. De même, nous obtenons de bons taux d'erreur dans nos expérimentations qui ont toutes été faites avec une base réduite pour l'apprentissage.

Ce travail s'inscrit dans un projet plus large sur la reconnaissance d'images vue comme un exemple du "jeu des vingt questions". Nous construisons les fonctions de l'image et la stratégie d'exploration en même temps, et d'une manière très hiérarchisée aussi bien pour la représentation du visage que pour son positionnement dans l'espace des poses. Ce paradigme a été analysé dans le contexte des arbres de décision et de la réduction de l'incertitude pas à pas dans (Amit & Geman 1997, Geman & Jedynak 1996, Jedynak & Fleuret 1996, Wilder 1998). Bien que l'approche de l'apprentissage soit différente ici, et qu'il n'y ait pas de construction d'arbres, l'algorithme final peut être mis sous la forme d'un énorme arbre binaire récursif, dont les questions sont basées sur des comparaisons entre les niveaux de gris des pixels.

1.6 Etat de l'art

La détection des instances d'un objet générique sans utiliser d'informations de couleur, profondeur ou mouvement a été énormément étudiée dans la littérature informatique. Dans le cas des visages, une multitude de méthodes ont été proposées, par exemple les réseaux de neurones artificiels (Rowley et al. 1998, Sung & Poggio 1998, Burel & Carel 1994, Vaillant et al. 1994), des modèles utilisant des centroïdes gaussiens (Sung & Poggio 1994), les "support vector machines" (Osuna et al. 1997), la mise en correspondance de graphes (Leung et al. 1995, Maurer & von der Malsburg 1996), l'inférence Bayésienne (Cootes & Taylor 1996), les modèles déformables (Yuille et al. 1992), le hachage géométrique (Lamdan et al. 1988), et les précurseurs de notre méthode qui ont déja été cités.

Les trois premiers algorithme partagent avec notre méthode le fait de parcourir la scène avec un détecteur construit sur une base d'images de visages. Pour chacune des positions, ils extraient une imagette, puis la normalisent en niveau de gris (La composante linéaire des niveaux de gris est soustraite, puis il y a égalisation d'histogramme, cela afin de corriger grossièrement les variations d'illumination).

Dans l'algorithme proposé par Rowley et al., l'imagette extraite est utilisée comme entrée d'un réseau de neurones à une couche cachée dont les connections sont contraintes. L'apprentissage de ce réseau se fait par rétropropagation sur une base de 1.000 images de visages originales à partir desquelles sont synthétisées 15.000 images de visages, et auxquelles sont rajoutées 8.000 images de "fond". Ces dernières sont sélectionnées par un processus de "bootstrap" qui consiste à choisir des images sur lesquelles le classificateur se trompe. Rowley annonce que l'apprentissage peut demander une journée complète sur une machine très puissante.

Un tel algorithme ayant un taux de faux positifs trop élevé, l'auteur combine plusieurs de ces réseaux et utilise des heuristiques pour réduire le nombre de fausses alarmes. Cet algorithme est relativement lent puisque son exécution demande 140s sur une machine comparable à un gros PC. Pour accélérer le traitement, Rowley propose de rajouter une première passe utilisant le même type de détecteur construit pour détecter un visage dans une sous-image 30×30, avec une tolérance de 10×10 en position. Ce réseau peut donc être utilisé de manière beaucoup moins intensive pour faire une pré-détection. Cette nouvelle procédure permet de réduire le temps de calcul de 140s à 2s, mais le taux de détection passe de 85.3% à 74%.

Dans (Sung & Poggio 1994), les sous-images extraites ont une taille de 19×19 pixels et sont vues comme des vecteur de l'espace $[0,1]^{19\times19}$. Le classificateur utilise un modèle statistique des images de visages et de fond dans cet espace sous la forme d'une famille de 6 centroïdes gaussiens pour les visages et 6 autres pour le "fond".

La classification utilise un réseau de neurones à une couche cachée de 24 neurones, et 12 neurones en entrée, correspondant aux 12 distances de Mahalanobis de l'image aux centres des gaussiennes (6 pour les visages, et 6 pour les fonds). Les paramètres de ces centroïdes sont estimés pendant une période d'apprentissage à partir des images d'une base d'exemples. Cette base contient 4.000 images, dont 1.000 images de visages, et 3.000 image de "fond" sélectionnées à nouveau par une méthode de bootstrap.

Sung traite spécifiquement le problème du sur-ajustement aux données en réduisant

la dimensionnalité de l'espace des images (initialement de dimension 19×19) par une technique d'analyse en composante principale. Ainsi la dimension de l'espace dans lequel est estimée la distance de Mahalanobis est réduite pour que le nombre d'images d'apprentissage disponibles soit suffisant. Les auteurs ne font pas allusion au coût algorithmique de leur technique, que l'on peut supposer élevé considérant les dimensions des espaces considérés (de l'ordre de 75).

Dans (Osuna et al. 1997) enfin, les auteurs proposent une application des "support vector machines" à la détection de visages. Cet algorithme est un classificateur générique à deux classes (Cortes & Vapnik 1995). Il consiste à plonger les données dans un espace de grande dimension à l'aide d'une application non linéaire, puis à déterminer dans cet espace un classificateur linéaire. Le sur-ajustement aux données est géré explicitement en sélectionnant un classificateur linéaire qui minimise le taux d'erreur sur la base d'apprentissage, et qui en plus maximise géométriquement la distance entre l'hyperplan de séparation et les points associés aux données. Malgré cette spécificité, la base d'apprentissage est ici très importante, puisqu'elle contient 50.000 imagettes, dont 1.000 exemples générés par "bootstrap".

Pour réduire le coût très important de cet algorithme, les auteurs utilisent une technique de simplification proposée par C. J. C Burges (Burges 1996) qui permet de réduire la complexité de la règle de décision.

Ainsi, ces techniques sont prévues essentiellement pour avoir un taux d'erreur peu élevé. L'efficacité algorithmique est gérée artificiellement en rajoutant une première passe utilisant un autre algorithme, ou bien en utilisant une approximation de l'algorithme le plus performant. Quant aux problèmes de sur-ajustement aux données (provoqués par des bases d'apprentissage trop réduites), ils sont traités en jouant sur les paramètres du classificateur : nombre de neurones cachés, dimension des espaces de représentation, etc. Néanmoins, dans tous les cas, les bases d'apprentissages sont importantes.

1.7 Plan du mémoire

Le chapitre 2 introduit une formalisation pour notre travail, et définit précisément quelle est la forme de notre détecteur.

Le chapitre 3 décrit les détecteur dédiés, introduit formellement l'idée d'arrangement décomposable, et énonce un théorème sur la décroissance des probabilités des tests associés à ces arrangements.

Le chapitre 4 présente l'algorithme de construction des détecteurs dédiés, qui permet d'approximer le détecteur théorique présenté dans le chapitre précédent.

Le chapitre 5 montre comment les détecteurs dédiés peuvent être combinés. Ce chapitre montre que sous certaines hypothèses le choix du critère de détection final est optimal en terme d'erreur, et que la stratégie utilisée pour implémenter ce critère est optimale en terme d'efficacité algorithmique.

Le chapitre 6 présente des résultats sur des images réelles, et illustre les propriétés d'optimalité présentées dans le chapitre précédent.

Le chapitre 7 enfin récapitule les résultats, propose des pistes pour les développements futurs de cet algorithme, et met en parallèles certains mécanismes biologiques avec la structure du détecteur que nous avons obtenu.

Chapitre 2

Formalisation

2.1 Pose

Comme nous l'avons dit dans l'introduction, nous nous intéressons ici à la conception d'un détecteur qui doit traiter une imagette 64×64 en 256 niveaux de gris, et doit être capable de détecter la présence éventuelle d'un visage.

Nous appellerons *pose* du visage l'ensemble des paramètres qui déterminent sa position. Elle est définie par la position des yeux. Nous considérons que la position de la bouche est grossièrement fixée par celle de ces derniers.

- La position (x, y) du point situé entre les yeux,
- L'angle θ que fait la droite qui passe par la bouche et entre les yeux avec la verticale,
- La distance s entre les yeux.

Nous considérons dans toute la suite de cette thèse que la pose est légèrement contrainte : le point situé entre les yeux se trouve dans un carré 8×8, la distance entre les yeux

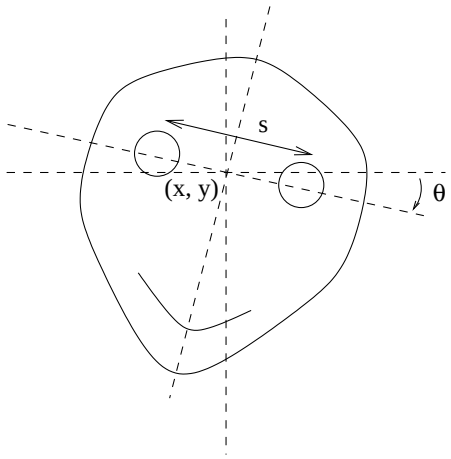


FIG. 2.1: *Nous faisons l'hypothèse que la pose d'un visage est déterminée par la position du point situé entre les yeux (x, y), par l'angle que fait la droite passant par les yeux avec l'horizontale, et enfin par la distance entre les yeux.*

s est comprise entre 10 et 20 pixels[1], et enfin l'inclinaison θ est comprise entre $-20°$ et $+20°$.

Nous noterons Θ l'ensemble des poses qui vérifient ces contraintes. Nous faisons l'hypothèse que cet ensemble a la forme $\Theta = \{\theta_1, \ldots, \theta_{2^M}\}$, qui correspond à la résolution la plus fine que nous considèrerons. Dans nos expériences $|\Theta| = 64$, c'est à dire $M = 6$. Cette précision correspond à 16 positions possibles en translations (la zone de tolérance 8×8 est subdivisée en 16 zones de 2×2 pixels), 2 tolérances possibles pour l'inclinaison θ ($\theta < 0$ et $\theta \geq 0$), et deux tolérances possibles pour la distance entre les yeux s ($s < 15$ et $s \geq 15$).

1. Les visages plus grands, tels que $s > 20$, sont traités au niveau global de la scène en utilisant plusieurs fois le détecteur aprés réduction de l'image d'un facteur 2^M

2.2 Lois de probabilités sur les images

Soit $G = \{1, \ldots, 64\}^2$ l'ensemble des positions des pixels d'une image 64×64. Soit $\mathcal{I}$ l'ensemble de telles images en 256 niveaux de gris. $I = \{I(x,y) : I(x,y) \in \{1, \ldots, 256\},\ (x,y) \in G\}$.

Nous noterons P la distribution sur $\mathcal{I}$ représentant les images 64×64 "qui existent dans le monde". Cette distribution pourrait être vue par exemple comme la distribution empirique obtenue en considérant toutes les imagettes 64×64 qui peuvent être extraites des images présentes sur le web.

Etant donnée une imagette, soit $Y \in \{0, 1, \ldots, 2^M\}$ sa classe. Elle est égale à 0 si il s'agit d'une image de "fond" (absence de visage), ou bien d'une image d'un visage dont la pose n'est pas dans Θ. Elle correspond à l'une des 2^M poses possibles dans le cas où elle représente un visage dont la pose est dans Θ. Nous faisons l'hypothèse que cette classe est complètement définie par la donnée des pixels, et nous la noterons donc $Y(I)$, ou plus simplement Y si nous faisons référence à la variable aléatoire.

Nous définissons plusieurs lois sur $\mathcal{I}$ correspondant à des sous-ensembles d'images

$$\begin{aligned} P_0 &= P(\ .\ |Y = 0) \\ P_1 &= P(\ .\ |Y > 0) \\ \forall \Gamma \subset \Theta,\ P_\Gamma &= P(\ .\ |Y \in \Gamma) \end{aligned}$$

P_0 représente donc la distribution sur les images de "fond", P_1 sur les images de visages, et enfin P_Γ sur les images de visages dont la pose est contrainte dans un sous-ensemble Γ de l'ensemble de poses complet Θ.

Soit enfin Q_i les probabilités conditionnellement à la pose θ_i

$$\forall i,\ 1 \leq i \leq 2^M,\ Q_i \ = \ P(\ .\ |Y = i) \ = \ P_{\{\theta_i\}}$$

Qui donne donc

$$\begin{aligned} P_\Gamma(\ .\ ) &= \sum_{i\in\Gamma} Q_i(\ .\ )P(Y = i|Y \in \Gamma) \\ &= \frac{1}{P(Y \in \Gamma)} \sum_{i\in\Gamma} Q_i(\ .\ )\pi_i \end{aligned}$$

en posant $\pi_i = P(Y = i)$, $P(Y \in \Gamma) = \sum_{i\in\Gamma} \pi_i$.

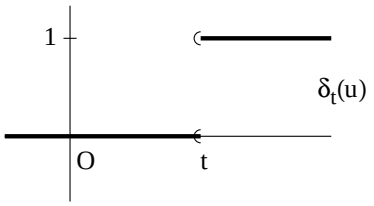
Nous ne cherchons pas dans cette thèse à modéliser les π_i. La seule hypothèse est que la présence d'un visage est un "évènement rare", c'est à dire $P(Y > 0) \ll P(Y = 0)$.

2.3 Structure du détecteur

Nous nous donnons une famille de fonctions booléennes de l'image. Nous appelons ces fonctions les *tests élémentaires*, et nous les notons $X_1, X_2, \ldots, X_N$. Comme nous le verrons dans la section 4.6.2 ces fonctions dépendent de la présence d'un bord d'une orientation donnée, à un endroit donné de l'image.

Le détecteur final $f : \mathcal{I} \to \{0, 1\}$ est une disjonction de détecteurs $f_\theta,\ \theta \in \Theta$ dédiés à des poses particulières. En notant

$$\begin{aligned} \delta_t(u) &= 0 \quad \text{si} \quad u < t \\ &= 1 \quad \text{si} \quad u \geq t \end{aligned}$$



on a

$$f(I) \;=\; \delta_1 \left(\sum_{\theta \in \Theta} f_\theta(I) \right)$$

Soit $\theta \in \Theta$. Le détecteur f_θ est une conjonction de plusieurs détecteurs dédiés à des ensembles de poses d'une séquence décroissante $\Gamma_1 \supset \Gamma_2 \supset \cdots \supset \{\theta\}$ dont l'intersection donne $\{\theta\}$. Pour indiquer que C parcourt une telle séquence, nous noterons $\Gamma \downarrow \theta$. Le détecteur f_θ est donc défini par :

$$f_\theta \;=\; \prod_{\Gamma \downarrow \theta} f_\Gamma$$

Chacun des f_Γ est dédié à un ensemble de poses Γ, et a la forme d'une conjonction de *tests cumulatifs* de plus en plus complexes

$$f_\Gamma \;=\; \prod_{k=1}^{K(\Gamma)} f_{\Gamma,k}$$

Chaque test cumulatif $f_{\Gamma,k}$ vaut 1 si, et seulement si, une somme de produits de tests élémentaires dépasse un seuil donné. Nous appellerons ces produits de tests élémentaires des *arrangements*. La taille de ces arrangements, c'est à dire le nombre de tests élémentaires impliqués dans le produit, est égale à j. On a donc l'expression suivante

$$f_{\Gamma,k} \;=\; \delta_{t(\Gamma,\ k)} \left(\sum_{A \in \mathcal{A}(\Gamma,\ k)} \prod_{i \in A} X_i \right)$$

Les $f_{\Gamma,k}$ seront construits de manière à avoir un taux de faux négatifs nul, c'est à dire $P_\Gamma(f_{\Gamma,k} = 0) = 0$, d'où $P_\Gamma(f_\Gamma = 0) = 0$, et $Q_i(f_{\theta_i} = 0) = 0$. Ce qui implique que le détecteur final a un taux de faux négatifs nul : $P_1(f = 0) = 0$.

Finalement, le détecteur dédié à la pose θ a donc la forme

$$f_\theta = \prod_{\Gamma \downarrow \theta} \underbrace{\prod_{k=1}^{K(\Gamma)} \overbrace{\delta_{t(\Gamma,\ k)} \left(\sum_{A \in \mathcal{A}(\Gamma,\ k)} \overbrace{\prod_{i \in A} X_i}^{\text{Arrangement}} \right)}^{\text{Test cumulatif de complexité } j}}_{\text{Détecteur dédié à } \Gamma}$$

Cette forme correspond à la hiérarchisation dont nous avons parlé dans la section 1.3. Le produit de détecteurs dédiés à des ensembles de poses de plus en plus précis implémente l'idée de hiérarchisation de l'espace des poses, et le produit des tests cumulatifs de complexités croissantes implémente la hiérarchisation de la complexité graphique d'un visage pour un ensemble de poses fixé.

Comme nous le verrons dans les chapitres qui viennent, le choix d'une telle forme permet d'obtenir un algorithme peu coûteux, avec un faible taux d'erreur, et qui nécessite une petite base de données d'apprentissage.

Chapitre 3

Détecteurs dédiés

3.1 Introduction

Pour détecter un visage dans une image 64×64, nous proposons donc une approche hiérarchisée qui consiste à utiliser une famille $f_\Gamma, \quad \Gamma \subset \Theta$, de *détecteurs dédiés* à des sous-ensembles particuliers de poses des visages.

En pratique tous les f_Γ sont construits de la même manière par apprentissage, et seules les bases d'exemples utilisées lors de leurs constructions les différencient les uns des autres. Ils ont tous un taux de faux négatifs ($P_\Gamma(f_\Gamma = 0)$) nul, et un taux de faux positifs ($P_0(f_\Gamma = 1)$) très peu élevé. Dans ce chapitre nous nous intéressons à la construction d'un de ces détecteurs, dédié à un sous-ensemble de poses Γ qui est fixé.

Au lieu d'utiliser un modèle non-paramétrique classique tel que des réseaux de neurones, des arbres de décision, ou autre, nous proposons un type original de détecteur qui a la forme d'une conjonction de tests de plus en plus complexes, comme introduite dans 2.3 :

$$f_\Gamma = \prod_{k=1}^{K} \delta_{t(k)} \left(\sum_{A \in \mathcal{A}(k)} \prod_{i \in A} X_i \right)$$

Où $K = K(\Gamma)$, $A(k) = A(k, \Gamma)$, etc. L'idée fondamentale qui motive la conception de ce nouvel algorithme est que toutes les structures géométriques propres à une classe d'objets, et en particulier propres aux visages, peuvent se décomposer sous la forme de morceaux de bords dont les présences sont statistiquement corrélées sur les images de l'objet. C'est en utilisant cette propriété que nous sélectionnons les éléments de $\mathcal{A}(k)$, puis que nous déterminons $t(k)$.

La plupart des structures physiques qui engendrent des bords sur les images de visages (sourcils, bouche, contour du crâne, etc.) sont relativement rigides, ou du moins se déplacent de manière cohérente. La présence d'un bord sur une de ces structures lorsqu'elle se trouve à une certaine position est donc un évènement corrélé avec la présence de n'importe quel autre bord sur la même structure lorsqu'elle est à la même position. Par exemple, le sourcil constitue une structure assez grande pour impliquer plusieurs fragments de bords, dont les présences sont donc corrélées. Sur la figure 3.1 sont représentés trois positions des sourcils et deux fragments de bords. Ces deux fragments appartenant à la même position du sourcil, leurs présences (aux positions et avec les orientations données), sont des événements statistiquement corrélés.

Une fois cette idée de corrélation entre deux fragments précisée, on l'étend naturellement à des structures impliquants plusieurs fragments en considérant des décompositions hiérarchiques en morceaux de plus en plus petits, corrélés entre eux, jusqu'à arriver aux fragments de bords les plus élémentaires. Par exemple, tous les fragments de bords situés sur la même position du sourcil sont corrélés. Ainsi, le sourcil en entier peut être mis sous la forme d'une réunion de fragments dont les présences sont corrélées.

Cette propriété de décomposabilité des structures en sous-structures dont les présences sont corrélées nous assure que la probabilité de présence de la structure globale

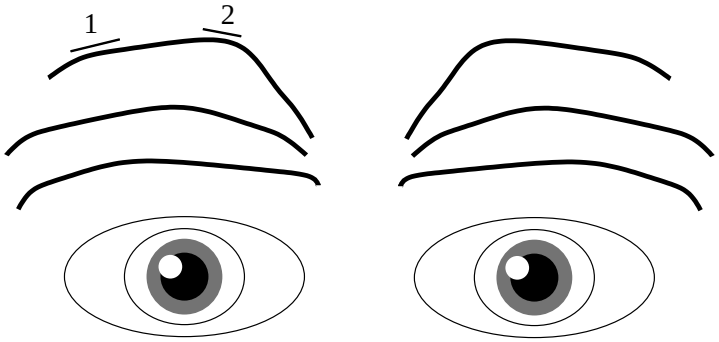


FIG. 3.1: *Représentation de trois positions des sourcils, et de deux fragments de bords. Ces deux fragments se trouvant sur la même position du sourcil, ils apparaissent simultanément et leurs présences sont des événements très corrélés. Le sourcil dans son entier peut être mis sous la forme d'une réunion de fragments de bords dont les présences sur l'image sont des événements corrélés entre eux.*

est plus grande que le produit des probabilités de présence des fragments de bords isolés (ce qui serait le cas si les présences de ces fragments étaient des événements indépendants), alors que les mêmes structures ont une probabilité beaucoup plus faible d'être présentes sur une image de fond.

Durant l'apprentissage, un grand nombre de telles structures de plus en plus complexes (c'est à dire réunissant un nombre de plus en plus grand de fragments de bords) sont construites. La corrélation est estimée à l'aide de la base d'images de visages. Le détecteur a finalement la forme d'une succession de comptages du nombre de ces structures présentes sur l'image. Si pour toutes les complexités k le nombre de structures présentes dépasse un certain seuil $t(k)$ (qui dépend donc de k, et est estimé pendant l'apprentissage), l'image est classée comme une image de visage, sinon comme une image de fond. Réduire la valeur du seuil permet de fixer le taux de faux négatifs à une valeur arbitrairement faible (tout en augmentant le nombre de fausses alarmes). Comme dans (Amit & Geman 1999), nous sélectionnons donc des arrangements discriminants en ne nous basant que sur des propriétés statistiques.

Cette démarche offre trois avantages essentiels. Tout d'abord nous n'utilisons que

des probabilités empiriques d'événements sur les images de visages. *Nous ne cherchons donc pas à modéliser la probabilité des images de fond*, car nous savons qu'un tel modèle est extrêmement complexe, et qu'il serait nécessaire de disposer d'une base d'apprentissage gigantesque pour réaliser une estimation réaliste. Ensuite, nous n'estimons que des corrélations, qui sont des grandeurs qui peuvent être calculées indépendamment les unes des autres.

Cette approche évite de souffrir d'une base d'apprentissage trop réduite en limitant les phénomènes de sur-ajustement. Dans le cas de modèles comme les réseaux de neurones, *un grand nombre de paramètres couplés sont estimés simultanément durant l'apprentissage*, et permettent d'adapter très finement le comportement du détecteur aux données. Cette souplesse peut devenir un handicap si les données sont trop peu nombreuses et pas assez représentatives des probabilités réelles sur les images de l'objet à détecter. Dans un tel cas, un biais de l'ensemble d'apprentissage (éclairage, couleur de peau, présence ou absence de lunettes) peut être utilisé à tort comme critère discriminant par le détecteur.

Ensuite, la forme finale du détecteur, organisé en une succession de tests, chacun comptant la présence d'un nombre minimum de structures d'une complexité donnée, est également intéressante d'un point de vue algorithmique. Cette architecture respecte l'idée de hiérarchisation de la représentation. L'algorithme poursuit le comptage de structures *tant que l'image n'a pas été rejetée*. De fait, si une image peut être rejetée en ne considérant que des structures peu complexes, elle le sera très rapidement. Par exemple, si une image est uniforme, elle ne présente aucun bord. Le premier test, basé sur un comptage des arrangements de complexité 1, n'atteindra pas le seuil et rejettera l'image.

Le calcul ne se poursuit vraiment que pour des images ambiguës. Il y a donc bien une organisation de la représentation de l'objet qui permet d'adapter le coût algorithmique à la similitude de l'image avec un visage.

Enfin, les tests successifs correspondent à des critères qualitativement différents, puis-

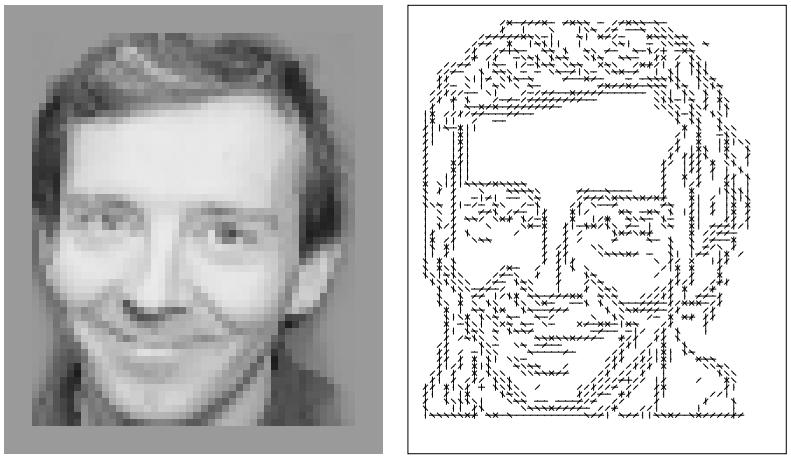


FIG. 3.2: *Gauche : Un visage provenant de la base d'apprentissage. Droite : Les fragments de bords détectés.*

qu'ils considèrent des structures de complexités différentes. De là vient une certaine indépendance statistique qui permet d'obtenir un taux de faux positifs inférieur au taux atteint avec le meilleur de ces tests pris isolément.

La présentation de cet algorithme se fait en trois parties. Nous décrivons dans la section 3.2 la notion d'arrangements ρ-décomposables, qui formalise précisément l'idée de *structure* que nous venons de décrire dans cette introduction, et qui constituent les $\mathcal{A}(k)$ des détecteurs $f_{\Gamma,k}$. Dans la section 3.3 nous prouvons que la probabilité de présence de tels arrangements ρ-décomposables décroît lentement quand leur complexité augmente. Enfin, dans la section 3.4 nous décrivons comment ces arrangements sont combinés sour la forme d'une succession de tests qui comptent chacun la présence de structures de plus en plus complexes.

3.2 Arrangements

3.2.1 Définition d'un arrangement

Nos *tests élémentaires* X_i sont des disjonctions de filtres locaux. Nous avons utilisé des filtres très grossiers, tolérants, et qui peuvent être implémentés efficacement. Néanmoins d'autres filtres, plus sophistiqués, pourraient être plus efficaces.

Chaque filtre est appliqué à toutes les positions de R, et a une orientation (horizontale, verticale, ou une des deux diagonales) et une polarité (positive ou négative). Sur la figure 3.2 nous montrons un visage provenant de la base d'apprentissage (à gauche) et les fragments de bords détectés (à droite). Nous reviendrons dans la section 4.6 sur la définition précise de ces détecteurs de bords.

Il y a un test élémentaire $X_i = X_i(I)$ pour chaque fragment de bord $i = 1, 2, \ldots, N$, où $N \approx 8|R|$. Le test $X_i = 1$ si le bord correspondant est présent dans un petit voisinage de x_i et $X_i = 0$ sinon. La taille du voisinage (le degré de "tolérance") dépend de l'orientation, de x_i, et du sous-ensemble de poses Γ. Il est choisi de manière à ce que la probabilité des tests élémentaires soit de l'ordre de $\frac{1}{2}$ (cf. section 4.6.4).

Nous formalisons l'idée de *structure* en considérant les indices des tests élémentaires qui détectent les fragments qui la composent. Plus précisément : nous appelons *arrangement* un sous-ensemble $A \subset \{1, \ldots, N\}$. La variable aléatoire correspondante

$$X_A(I) = \prod_{i \in A} X_i(I)$$

sur $\mathcal{I}$ est simplement la conjonction spatiale de tests élémentaires : $X_A = 1$ si et seulement si $X_i = 1$ pour tous les $i \in A$. Soit $supp\ X_i \subset R$ l'ensemble des positions de bords qui apparaissent dans la définition de X_i. Pour limiter la taille de la famille des arrangements, nous faisons l'hypothèse que $supp\ X_i \bigcap supp\ X_j = \emptyset$ pour tout

$i \in A, j \in A, i \neq j$. Cet ensemble constitue notre famille de tests ; le détecteur sera construit à l'aide d'un sous-ensemble de ces tests.

3.2.2 ρ-décomposabilité

Nous voulons trouver des arrangements A ayant des statistiques les plus différentes possibles pour P_0 et pour P_Γ. Parce que l'estimation de la probabilité sous P_0 est problématique, nous essayons d'obtenir cette disparité en construisant de grands arrangements vraisemblables sous P_Γ. La taille seule les rend rares sous P_0.

La construction est basée sur la corrélation. Soient U et V deux variables aléatoires de Ω dans $\{0,1\}$ telles que $0 < P_\Gamma(U=1), P_\Gamma(V=1) < 1$. Notons $\rho(U,V)$ le coefficient de corrélation :

$$\rho(U,V) = \frac{P_\Gamma(U=1,V=1) - P_\Gamma(U=1)\cdot P_\Gamma(V=1)}{(P_\Gamma(U=1)\cdot P_\Gamma(U=0)P_\Gamma(V=1)\cdot P_\Gamma(V=0))^{1/2}}$$

Soit $\{X_1,\ldots,X_N\}$ un ensemble de "tests élémentaires". Considérons les arrangements X_iX_j de taille deux. Nous pouvons filtrer de telles paires à l'aide de la contrainte

$$\rho(X_i,X_j) \geq \rho$$

pour un certain seuil ρ, $0 < \rho < 1$. Nous sélectionnons ainsi les paires de tests qui ont tendance à apparaître simultanément sur les visages dont la pose est dans Γ. De même $X_iX_jX_k$ serait un bon candidat pour un arrangement discriminant de taille trois, si $\rho(X_iX_j,X_k) \geq \rho$. En continuant ainsi, nous pouvons isoler de bons candidats de taille quatre en combinant deux "bonnes" paires de taille deux, qui vérifient $\rho(X_iX_j,X_kX_l) \geq \rho$. Etc.

Nous définissons une *décomposition* de A comme une succession de partitions bi-

naires emboitées, jusqu'à des singletons sous-ensembles de $\{1, \ldots, N\}$. Nous imposons également que chaque partition d'un sous-ensemble le divise en deux parties de même cardinaux si l'ensemble est de cardinal pair, et en deux parties de cardinaux différents de un si l'ensemble est de cardinal impair. Nous dirons qu'il s'agit d'une ρ-*décomposition* si l'inégalité sur la corrélation est vérifiée pour chaque partage en deux. Sur la figure 3.3 nous montrons une telle décomposition de $A = \{1, 2, 4, 5, 9\}$. C'est une ρ-décomposition si $\rho(X_1X_4,\ X_2X_5X_9) \geq \rho$, $\rho(X_1,\ X_4) \geq \rho$, $\rho(X_5X_9,\ X_2) \geq \rho$ et $\rho(X_5,\ X_9) \geq \rho$. Finalement, un arrangement A (ou la variable aléatoire correspondante X_A) sera dit ρ-*décomposable* s'il existe au moins une ρ-décomposition de A. Précisément :

Définition: *Un arrangement $A \subset \{1, \ldots, N\}$ est ρ-décomposable si c'est un arrangement de taille un (un test élémentaire soit $|A| = 1$) ou bien s'il existe deux arrangements B et C, eux-même ρ-décomposables, avec :*

- $A = B \cup C, \quad B \cap C = \emptyset$
- $||B| - |C|| \leq 1$
- $\rho(X_B, X_C) \geq \rho$

De plus, nous avons vu dans 3.2.1 que deux tests élémentaires qui appartiennent à un même arrangement doivent dépendre de bords situés à des positions différentes. Cette condition impose que les emplacements de l'image où se trouvent les bords dont dépendent B et C ne doivent pas se recouper. En posant

$$supp\ B = \bigcup_{i \in B} supp\ X_i \quad \text{et} \quad supp\ C = \bigcup_{i \in C} supp\ X_i$$

on doit donc avoir

$$supp\ B \bigcap supp\ C = \emptyset$$

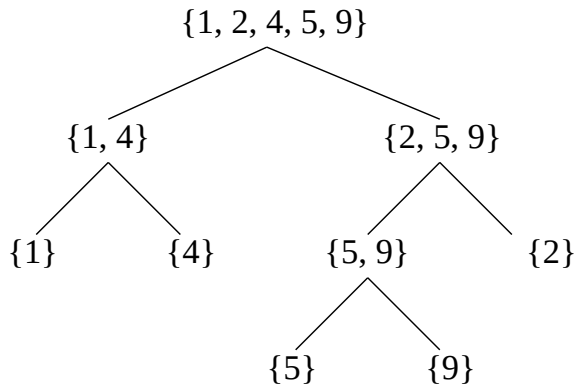


FIG. 3.3: *Un arrangement ρ-décomposable peut être décomposé récursivement en conjonctions d'arrangements corrélés et à peu près de même tailles.*

On notera $\mathcal{A}(k, \rho)$ l'ensemble des arrangements de taille k qui sont ρ-décomposables. La définition de cet ensemble dépend de la loi P_Γ, donc de Γ. Ces ensembles seront donc différents d'un f_Γ à l'autre.

3.3 Théorème sur la borne inférieure

En général $P_0(X_A = 1)$ et $P_\Gamma(X_A = 1)$ dépendent de A et décroissent quand $|A|$ augmente. Une hypothèse raisonnable sous P_0 est une décroissance exponentielle, et c'est ce que nous observons effectivement dans nos résultats expérimentaux. Par contre, si A est un arrangement ρ-décomposable, nous pouvons nous attendre à une décroissance moins rapide sous P_Γ. Nous observons expérimentalement ces comportements, comme représenté sur la figure 3.5.

La conséquence est que pour pour des valeurs "raisonnables" de ρ, on a $P_\Gamma(X_A = 1) \gg P_0(X_A = 1)$ pour de "grand" A. On ne peut néanmoins rien dire pour le rapport des vraisemblances puisque nous ne proposons aucun modèle pour P_0. Pour $P_\Gamma(X_A = 1)$ par contre nous pouvons calculer trois bornes inférieures précises. La première est évidente et de la forme $f(k) = \xi^{|A|}$, la seconde de la forme $g(k) = \xi \cdot \rho^{\log_2 |A|}$, et

enfin la dernière $U(k)$ est définie par une relation de récurence, et n'a pas de forme analytique évidente.

Proposition : *Pour tout* $k \geq 1,\ \rho > 0$ *et* $A \in \mathcal{A}(k,\rho)$,

$$P_\Gamma(X_A = 1) \geq \left(\min_{1 \leq i \leq N} P_\Gamma(X_i = 1) \right)^k \tag{3.1}$$

Démonstration : Le résultat est évident pour $k = 1$. De plus :

$$\frac{P_\Gamma(X_{A\cup B} = 1) - P_\Gamma(X_A = 1) \cdot P_\Gamma(X_B = 1)}{\sqrt{P_\Gamma(X_A = 1) \cdot (1 - P_\Gamma(X_A = 1)) \cdot P_\Gamma(X_B = 1) \cdot (1 - P_\Gamma(X_B = 1))}} \quad \geq \quad \rho$$

D'où :

$$\begin{aligned} P_\Gamma(X_{A\cup B} = 1) \quad &\geq \quad \rho \cdot \sqrt{P_\Gamma(X_A = 1) \cdot (1 - P_\Gamma(X_A = 1)) \cdot P_\Gamma(X_B = 1) \cdot (1 - P_\Gamma(X_B = 1))} \\ &\quad + P_\Gamma(X_A = 1) \cdot P_\Gamma(X_B = 1) \\ &\geq \quad P_\Gamma(X_A = 1) \cdot P_\Gamma(X_B = 1) \end{aligned}$$

Une récurence immédiate prouve le résultat. □

Théorème : *Pour tout* $k \geq 1,\ \rho > 0$ *et* $A \in \mathcal{A}(k,\rho)$,

$$P_\Gamma(X_A = 1) \geq \min_{1 \leq i \leq N} P_\Gamma(X_i = 1) \cdot \rho^{\log_2 k} \tag{3.2}$$

Démonstration : Le résultat est évident pour $k = 1$. Posons $\xi = \min_{1 \leq i \leq N} P_\Gamma(X_i = 1)$. Faisons l'hypothèse que (3.2) est vraie pour tout $k \leq n$. Alors, soit $i, j \leq n,\ i \leq j \leq i + 1,\ B \in \mathcal{A}(i,\rho),\ C \in \mathcal{A}(j,\rho)$ et enfin $B \cup C \in \mathcal{A}(i + j,\rho)$; on a :

$$\begin{aligned} P_\Gamma(X_{B\cup C}=1) &\geq \rho\cdot\sqrt{P_\Gamma(X_B=1)\cdot(1-P_\Gamma(X_B=1))\cdot P_\Gamma(X_C=1)\cdot(1-P_\Gamma(X_C=1))} \\ &\quad + P_\Gamma(X_B=1)\cdot P_\Gamma(X_C=1) \end{aligned}$$

Si l'on pose $\alpha=\log_2 i$ et $\beta=\log_2 j$ et que l'on fait l'hypothèse que $P_\Gamma(X_B=1)\leq\frac{1}{2}$ et $P_\Gamma(X_C=1)\leq\frac{1}{2}$, comme $x\mapsto x(1-x)$ est croissante sur $[0,\frac{1}{2}]$:

$$\begin{aligned} P_\Gamma(X_{B\cup C}=1) &\geq \rho\cdot\sqrt{\xi\cdot\rho^\alpha(1-\xi\cdot\rho^\alpha)\cdot\xi\cdot\rho^\beta(1-\xi\cdot\rho^\beta)}+\xi\cdot\rho^\alpha\cdot\xi\cdot\rho^\beta \\ &\geq \xi\cdot\rho^{\frac{\alpha+\beta}{2}+1}\cdot\sqrt{(1-\xi\cdot\rho^\alpha)\cdot(1-\xi\cdot\rho^\beta)}\ +\ \xi^2\cdot\rho^{\alpha+\beta} \end{aligned}$$

Or $\beta\geq\alpha$ d'où $1-\xi\rho^\beta\geq 1-\xi\rho^\alpha$, donc :

$$\begin{aligned} P_\Gamma(X_{B\cup C}=1) &\geq \xi\cdot\rho^{\frac{\alpha+\beta}{2}+1}\cdot\sqrt{(1-\xi\cdot\rho^\alpha)\cdot(1-\xi\cdot\rho^\alpha)}\ +\ \xi^2\cdot\rho^{\alpha+\beta} \\ &\geq \xi\cdot\rho^{\frac{\alpha+\beta}{2}+1}\cdot(1-\xi\cdot\rho^\alpha)\ +\ \xi^2\cdot\rho^{\alpha+\beta} \\ &= \xi\cdot\rho^{\frac{\alpha+\beta}{2}+1}\cdot\left(1-\xi\cdot\rho^\alpha+\xi\cdot\rho^{\frac{\alpha+\beta}{2}-1}\right) \\ &\geq \xi\cdot\rho^{\frac{\alpha+\beta}{2}+1}\cdot\left(1+\xi\cdot\left(\rho^{\frac{\alpha+\beta}{2}-1}-\rho^\alpha\right)\right) \end{aligned}$$

Or :

$$\begin{aligned} i\geq 1,\ j\leq p+1 &\Rightarrow j\leq 4i \\ &\Rightarrow \log_2 j\leq\log_2 i+2 \\ &\Rightarrow \beta\leq\alpha+2 \\ &\Rightarrow \alpha+\beta-2\leq 2\alpha \\ &\Rightarrow \frac{\alpha+\beta}{2}-1\leq\alpha \\ &\Rightarrow \rho^{\frac{\alpha+\beta}{2}-1}\geq\rho^\alpha \end{aligned}$$

finalement :

$$P_\Gamma(X_{B\cup C}=1) \;\geq\; \xi\cdot\rho^{\frac{\alpha+\beta}{2}+1}$$

Enfin, on a par concavité du $\log_2$:

$$\begin{aligned}\frac{\log_2 i+\log_2 j}{2}+1 &\leq \log_2\left(\frac{i+j}{2}\right)+1\\ &\leq \log_2(i+j)\end{aligned}$$

Donc :

$$P_\Gamma(X_{B\cup C}=1) \;\geq\; \xi\cdot\rho^{\log_2(i+j)}$$

Finalement, si (3.2) est vraie pour tous les k plus petits que n, et si $A\in\mathcal{A}(n+1,\rho)$, alors : si $n+1$ est pair, (respectivement impair) $\exists B\in\mathcal{A}(\frac{n+1}{2},\rho),\ C\in\mathcal{A}(\frac{n+1}{2},\rho)$ (respectivement $\exists B\in\mathcal{A}(\frac{n}{2},\rho),\ C\in\mathcal{A}(\frac{n}{2}+1,\rho)$), $A=B\cup C$ et $\rho(B,\ C)\geq\rho$. Et on a $P_\Gamma(X_A=1)=P_\Gamma(X_{B\cup C}=1)\geq\xi\cdot\rho^{\log_2(n+1)}$. □

La troisième borne enfin :

Proposition : *Soit :*

- $U(0)=\min_{1\leq i\leq N}P_\Gamma(X_i=1)$
- $U(2k)=\rho\cdot U(k)\cdot(1-U(k))+U(k)^2$
- $U(2k+1)=\rho\cdot\sqrt{U(k)\cdot(1-U(k))\cdot U(k+1)\cdot(1-U(k+1))}+U(k)\cdot U(k+1)$

Pour tout $k \geq 1$, $\rho > 0$ et $A \in \mathcal{A}(k, \rho)$, on a $P_\Gamma(X_A = 1) \geq U(k)$

La figure 3.4 correspond à $\rho = .2$ et $\xi = .5$ et permet de comparer sur une échelle logarithmique le comportement des trois bornes et celui des probabilités réelles d'échantillons d'arrangements de complexités croissantes. Comme on peut le voir sur cette figure $\xi \cdot \rho^{\log_2 k}$, $U(k)$, et les probabilités mesurées ont un comportement similaire asymptotiquement, à un facteur multiplicatif près.

3.4 Structure du détecteur dédié

Le détecteur f_Γ a la forme d'une succession de tests de plus en plus complexes. Chacun de ces tests consiste en un comptage du nombre d'arrangements ρ-décomposables d'une complexité donnée k présents sur l'image, nombre qui est ensuite comparé à un seuil $t(k)$. Si ce seuil est atteint pour tous les tests du détecteur, un visage est détecté.

Fixons ρ. Chaque test est basé sur le nombre $Z_{k,\rho}$ d'arrangements ρ-décomposables de taille k présents sur l'image I :

$$Z_{k,\rho}(I) = \sum_{A \in \mathcal{A}(k,\rho)} X_A(I)$$

Soit K le plus grand k tel que les arrangements de taille k "couvrent" la classe objet, c'est à dire $P_\Gamma(Z_{k,\rho} \geq 1) = 1$ (au cours de nos expériences, il n'est jamais arrivé que les arrangements de taille k couvrent la classe objet mais pas les arrangements de taille $j < k$). Etant donnés des seuils $\{t(1), \ldots, t(K)\}$, nous détectons un visage sur l'image I si elle contient plus de $t(k)$ arrangements ρ-décomposables pour tous les $k = 1, \ldots, K$. Ce qui revient à dire que le détecteur se met sous la forme :

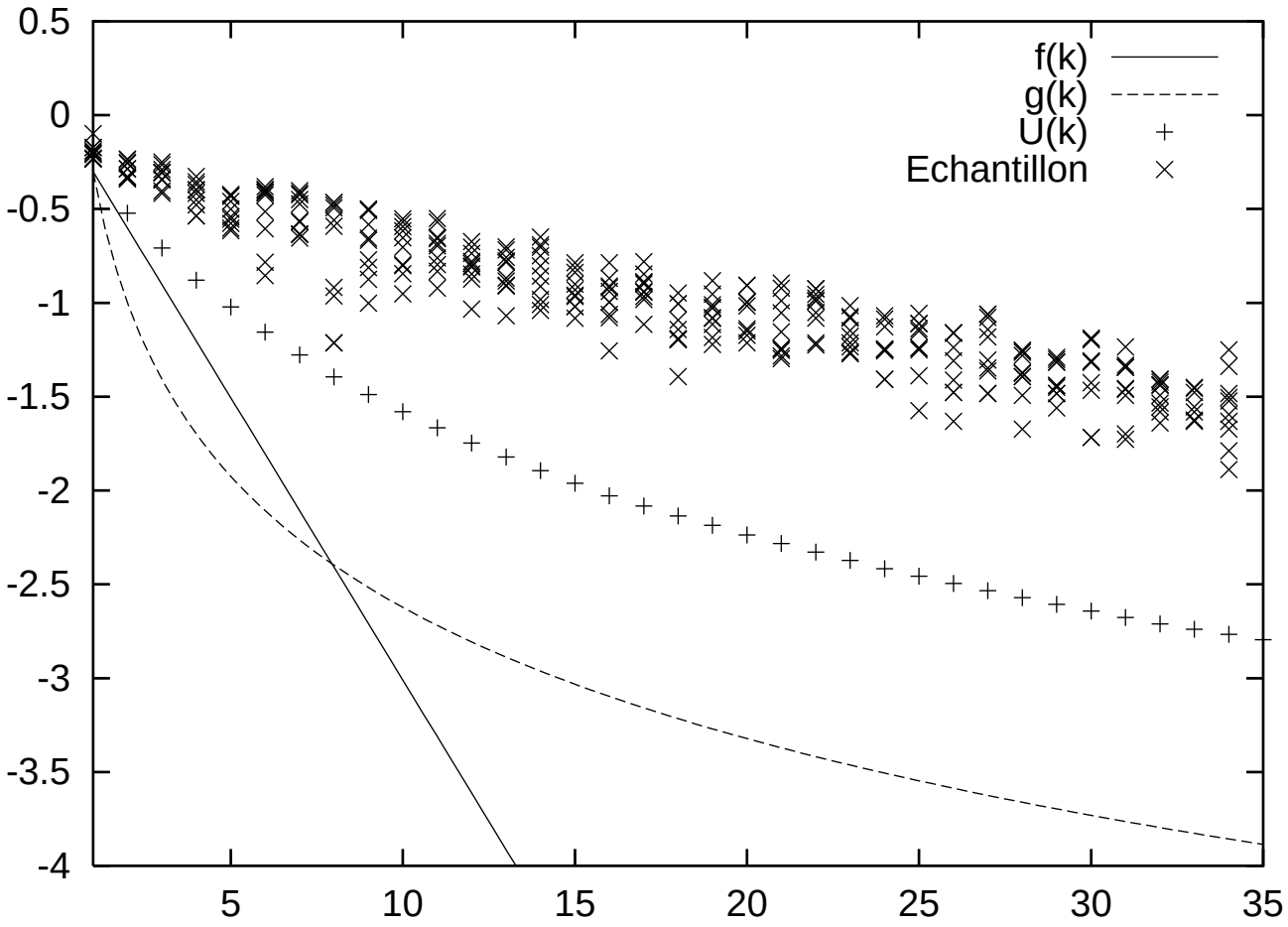


FIG. 3.4: *Représentations sur une échelle logarithmique des trois différentes bornes* $f(k) = \xi^k$, $g(k) = \xi \cdot \rho^{\log_2 k}$, *et* $U(k)$, *pour* $\rho = .2$ *et* $\xi = .5$, *ainsi que des probabilités d'un échantillon de dix arrangements de chaque complexité.*

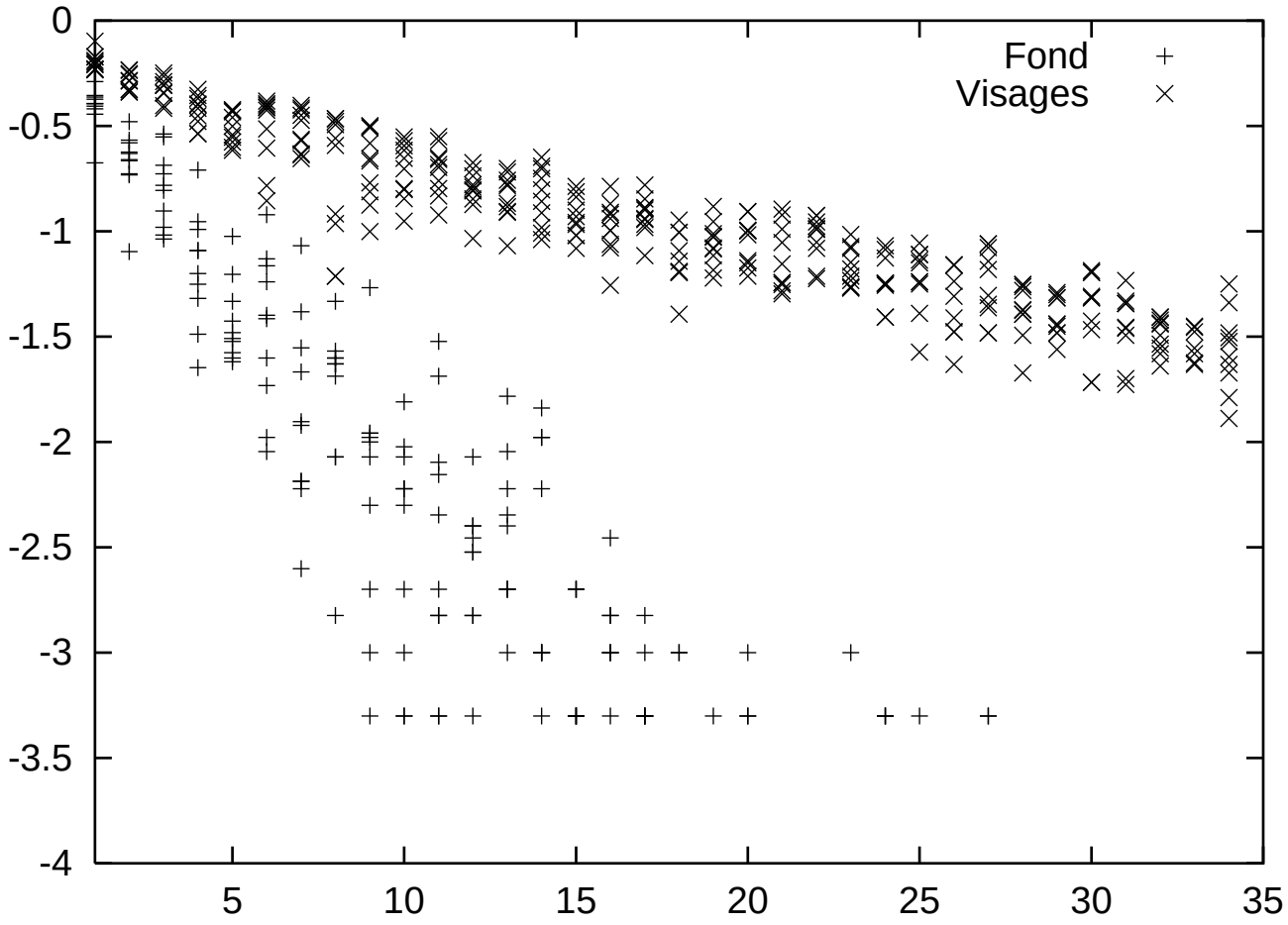


FIG. 3.5: *Représentations sur une échelle logarithmique de probabilités des arrangements pour P_0 (fond, symbole +) et P_Γ (visages, symbole ×).*

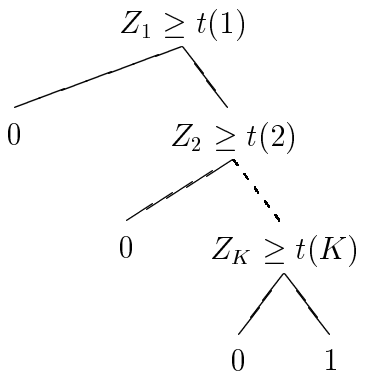


FIG. 3.6: *Le détecteur est une séquence hiérarchique de filtres*

$$f_\Gamma(I) = \prod_{k=1}^{K} \delta_{t(k)}\left(Z_{k,\rho}(I)\right)$$

Les seuils $t(1), \ldots, t(K)$ sont définis par

$$t(k) = \max\{t : P_\Gamma(Z_{k,\rho} \geq t) = 1\}$$

Ils correspondent donc aux valeurs maximum qui permettent d'obtenir $P_\Gamma(f_\Gamma = 0) = 0$.

L'invariance à l'occlusion, ou à la dégradation de l'image, est introduite dans la forme des tests $Z_k \geq t(k)$. Par définition, un tel test admet une grande tolérance dans le sous-ensemble d'arrangements effectivement présents puisque seul leur nombre importe.

Nous implémentons f_Γ sous la forme d'une succession de tests comme représenté sur la figure 3.6. Une telle implémentation exprime l'intérêt d'une conjonction, et plus globalement de la hiérarchisation : dès que l'un des tests $Z_k \geq t(k)$ est faux, le traitement s'arrête. Ainsi, seules les images ambiguës demandent un calcul important.

Le choix le plus naturel pour ρ correspond à la valeur qui minimise l'erreur de faux positifs :

$$\rho^* = \arg\min_{\rho} \beta(f_\Gamma)$$

En pratique nous n'avons pas procédé à une détermination systématique de la valeur ρ et nous prenons une valeur constante : $\rho = 0.1$. De même il est impossible de construire les $\mathcal{A}(k, \rho)$ tels qu'ils ont été définis. Nous verrons comment estimer malgré tous les Z_k empiriquement dans le chapitre 4 en contrôlant les ressources (mémoire et temps de calculs) nécessaires.

3.4.1 Contre-exemple

On peut légitimement se demander dans quel cas un détecteur de la forme $Z_k \geq t$ approxime mal le maximum de vraisemblance. Nous montrons ici que l'on peut construire un grand nombre de variables booléennes Y_i, fortement dépendantes, mais indépendantes deux à deux. Si les tests élémentaires qui servent à construire les arrangements avaient un tel comportement, ils seraient fortement corrélés, mais aucun arrangement de taille 2 ne serait ρ-décomposable. Donc, aucun arrangement, quelle que soit sa taille, ne serait ρ-décomposable. Notre test aurait donc la forme $\sum_i Y_i \geq t$.

Nous calculons ici le taux de faux positifs d'un tel test, en choisissant t de manière à avoir un taux de faux négatifs nul. Nous le comparons ensuite au taux de faux positifs du maximum de vraisemblance, qui admet aussi un taux de faux négatifs nul. Et nous montrons finalement l'inefficacité de notre algorithme dans un cas comme celui-ci.

Notons $\mathcal{P}(k)$ l'ensemble des parties non vides de $\{1, \ldots, k\}$, et $\oplus$ le "ou exclusif" ($0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$, et enfin $1 \oplus 1 = 0$). Nous allons construire deux lois P_0 et P_1 pour la famille de variables aléatoires $\vec{Y} = (Y_A)_{A \in \mathcal{P}(N)}$ sur $\{0,1\}^{\mathcal{P}(N)}$.

La première est donnée en considérant que les Y_A sont i.i.d de probabilité $\frac{1}{2}$. Donc $P_0(\vec{Y}) = \frac{1}{2^{|\mathcal{P}(N)|}}$.

La seconde est donnée en considérant $\{X_i\}_{i\in\{1,...N\}}$ des variables de bernoulli i.i.d de paramètre $\frac{1}{2}$. Et pour tout $A \in \mathcal{P}(N)$

$$Y_A = \bigoplus_{i\in A} X_i$$

Nous allons montrer que les Y_A sont indépendantes deux à deux sous P_1.

Lemme : Si X, Y et Z sont trois variables booléennes, si $P(X = 0) = \frac{1}{2}$ et si X est indépendante de Y et Z, alors $X \oplus Y$ et Z sont indépendantes.

Démonstration :

$$\begin{aligned}
P(X \oplus Y = 1, Z = 1) &= \Sigma_i P(X \oplus i = 1, Z = 1|Y = i)P(Y = i) \\
&= \Sigma_i P(X = 1 - i|Y = i)P(Z = 1|Y = i)P(Y = i) \\
&= \Sigma_i P(X = 1 - i)P(Z = 1|Y = i)P(Y = i) \\
&= \frac{1}{2}\Sigma_i P(Z = 1|Y = i)P(Y = i) \\
&= \frac{1}{2}P(Z = 1)
\end{aligned}$$

$$\begin{aligned}
P(X \oplus Y = 1) &= \Sigma_i P(X = 1 - i)P(Y = i) \\
&= \Sigma_i \frac{1}{2}P(Y = i) \\
&= \frac{1}{2}
\end{aligned}$$

□

Lemme : Pour $A \neq B$, les variables Y_A et Y_B sont indépendantes.

Démonstration : si $A \neq B$, alors, éventuellement en inversant les rôles de A et de

B, $\exists i \in A \setminus B$. Donc $Y_A = X_i \oplus \left(\bigoplus_{j \in A \setminus \{i\}} X_j\right)$, donc d'après le lemme 3.4.1, Y_A et Y_B sont indépendantes.

On peut calculer le comportement des $Z_k = \sum_{A \in \mathcal{P}(k)} Y_A$ sous P_0 et sous P_1. Sous P_0 il s'agit d'une loi binomiale $\mathcal{B}(\frac{1}{2}, |\mathcal{P}(N)|)$.

Sous P_1, on peut la calculer récursivement. Pour Ψ une assertion logique, on notera $\delta(\Psi)$ l'entier 0 quand Ψ est fausse, et 1 quand elle est vraie. On a alors

$$
\begin{aligned}
P_1(Z_N = k) &= P_1\left(\sum_{A\in\mathcal{P}(N)} Y_A = k\right) \\
&= P_1\left(\sum_{A\in\mathcal{P}(N-1)} Y_A + \sum_{A\in\mathcal{P}(N-1)} Y_{\{N\}} \oplus Y_A + Y_{\{N\}} = k\right) \\
&= P_1\left(2\sum_{A\in\mathcal{P}(N-1)} Y_A = k \,\middle|\, Y_{\{N\}} = 0\right) P_1(Y_{\{N\}} = 0) \\
&\quad + P_1\left(\sum_{A\in\mathcal{P}(N-1)} Y_A + \sum_{A\in\mathcal{P}(N-1)} 1 \oplus Y_A \ + 1 = k \,\middle|\, Y_{\{N\}} = 1\right) P_1(Y_{\{N\}} = 1) \\
&= \frac{1}{2} P_1\left(2\sum_{A\in\mathcal{P}(N-1)} Y_A = k\right) \\
&\quad + \frac{1}{2} P_1\left(\sum_{A\in\mathcal{P}(N-1)} Y_A + \sum_{A\in\mathcal{P}(N-1)} 1 \oplus Y_A \ + 1 = k\right) \\
&= \frac{1}{2} P_1\left(2Z_{N-1} = k\right) \\
&\quad + \frac{1}{2} P_1\left(\sum_{A\in\mathcal{P}(N-1)} (Y_A + 1 \oplus Y_A) \ + 1 = k\right) \\
&= \frac{1}{2}\left(P_1\left(2Z_{N-1} = k\right) + \delta\left(|\mathcal{P}(N-1)| - 1 + 1 = k\right)\right) \\
&= \frac{1}{2}\left(P_1\left(2Z_{N-1} = k\right) + \delta\left(2^{N-1} - 1 + 1 = k\right)\right) \\
&= \frac{1}{2}\left(P_1\left(Z_{N-1} = \frac{k}{2}\right) + \delta\left(2^{N-1} = k\right)\right)
\end{aligned}
$$

On a de manière évidente $P_1(Z_1 = 0) = \frac{1}{2}$ et $P_1(Z_1 = 1) = \frac{1}{2}$.

Faisons l'hypothèse de récurence que $P_1(Z_{N-1} = 0) = \frac{1}{2^{N-1}}$ et $P_1(Z_{N-1} = 2^{N-2}) = 1 - \frac{1}{2^{N-1}}$. On a alors immédiatement d'après ce qui précède $P_1(Z_N = 0) = \frac{1}{2} P_1(Z_{N-1} = 0) = \frac{1}{2^N}$. De plus

$$\begin{aligned} P_1(Z_N = 2^{N-1}) &= \frac{1}{2}\left(P_1\left(Z_{N-1} = 2^{N-2}\right) + 1\right) \\ &= \frac{1}{2}\left(1 - \frac{1}{2^{N-1}} + 1\right) \\ &= 1 - \frac{1}{2^N} \end{aligned}$$

Finalement

$$\forall N,\ \begin{cases} P_1(Z_N = 0) = \frac{1}{2^N} \\ P_1(Z_N = 2^{N-2}) = 1 - \frac{1}{2^N} \end{cases}$$

Donc, en posant $t = \min\{u : P_1(Z_N \geq u) = 1\}$, on obtient $t = 2^{N-1}$, et $P_0(Z_N \geq t) = P(\mathcal{B}(\frac{1}{2}, |\mathcal{P}(N)|) \geq 2^{N-1}) = \frac{1}{2}$. Donc un classificateur de la forme $1_{\{Z_N \geq t\}}$ avec un taux de faux négatifs nul ne peut pas séparer correctement la loi P_0 de la loi P_1.

On peut comparer ce taux d'erreur à celui commis par le maximum de vraisemblance. Soit $\vec{y} = (y_A)_{A \in \mathcal{P}}$ une observation. On a

$$P_0(\vec{Y} = \vec{y}) = \frac{1}{2^{|\mathcal{P}|}} = \frac{1}{2^{2^N - 1}}$$

et

$$
\begin{aligned}
P_1(\vec{Y}=\vec{y}) &= P_1(Y_A = y_a,\ A \in \mathcal{P}(N)) \\
&= P_1(Y_A = y_a,\ A \in \mathcal{P}(N),\ |A| > 1 \mid Y_A = y_a,\ A \in \mathcal{P}(N),\ |A| = 1) \\
&\quad P_1(\forall A \in \mathcal{P}(N), |A| = 1,\ Y_A = y_a) \\
&= P_1(Y_A = y_a,\ A \in \mathcal{P}(N),\ |A| > 1 \mid X_i = y_{\{i\}},\ 1 \leq i \leq N) \\
&\quad P_1(X_i = y_{\{i\}},\ 1 \leq i \leq N) \\
&= P_1(\oplus_{i\in A} y_{\{i\}} = y_a,\ A \in \mathcal{P}(N),\ |A| > 1 \mid X_i = y_{\{i\}},\ 1 \leq i \leq N) \\
&\quad P_1(X_i = y_{\{i\}},\ 1 \leq i \leq N) \\
&= \delta(\forall A,\ y_A = \oplus_{i\in A} y_{\{i\}}) P_1(X_i = y_{\{i\}},\ 1 \leq i \leq N) \\
&= \delta(\forall A,\ y_A = \oplus_{i\in A} y_{\{i\}})\ \frac{1}{2^N}
\end{aligned}
$$

Finalement, en notant

$$
D = \left\{ \vec{y} : \quad \forall A \in \mathcal{P}, |A| > 1, y_A = \bigoplus_{i\in A} y_{\{i\}} \right\}
$$

alors si $\vec{y} \in D$, $P_1(\vec{Y}=\vec{y}) = \frac{1}{2^N}$, sinon $P_1(\vec{Y}=\vec{y}) = 0$.

La règle de maximum de vraisemblance f^{mv} choisira donc l'hypothèse P_1 si $\vec{y} \in D$, et P_0 sinon. Donc

$$
\begin{aligned}
P_0(f^{mv} = 1) &= P_0\left(Y_A = \bigoplus_{i\in A} Y_{\{i\}}, A \in \mathcal{P}, |A| > 1 \right) \\
&= \frac{|D|}{2^{|\mathcal{P}(N)|}} \\
&= \frac{1}{2^{|\mathcal{P}(N)|-N}}
\end{aligned}
$$

et

$$P_1(f^{mv} = 0) \quad = \quad 0$$

Le maximum de vraisemblance a donc un taux de faux négatifs nul. Son taux de faux positifs est égal à $\frac{1}{2^{|\mathcal{P}(N)|-N}}$, ce qui est bien meilleur que celui atteint par un test de la forme $Z_N \geq t$.

Chapitre 4

Apprentissage

4.1 Introduction

Etant donnée une partie $\Gamma \subset \Theta$, nous ne pouvons pas, en pratique, construire directement le f_ρ correspondant parce que nous ne disposons pas des ensembles $\mathcal{A}_{k,\rho}, k = 1, \ldots, K$. D'une part leur construction nécessite la connaissance de P_Γ, et d'autre part il y en a un trop grand nombre.

A la place, notre algorithme est basé sur une approximation. Comme nous ne pouvons pas construire *tous* les arrangements ρ-décomposables, nous essayons de déterminer un nombre fini d'entre eux pour chaque complexité $k \leq K$. Donc, étant donné un ensemble $\mathcal{L}$ d'images de visages dont les poses sont dans Γ, l'un des objectifs de l'apprentissage est d'estimer pour tout $k \leq K$ une sous-famille $\mathcal{A}_\mathcal{L}(k, \rho) \subset \mathcal{A}(k, \rho)$, de cardinal borné par une constante m. L'autre objectif de l'apprentissage est d'estimer les seuils $t(1), \ldots, t(K)$ utilisés dans la règle de décision.

Pour illustrer le fonctionnement du processus d'apprentissage, deux expériences sont décrites dans ce chapitre. La première est synthétique, et consiste à détecter des "V" dans un monde de lignes. Cette tâche est très difficile car ces "V" sont conçus pour

être localement similaires au bruit qui est composé de lignes diagonales. La deuxième expérience est faite sur des données réelles, et constitue une première étape vers le traitement de scènes complètes. Elle consiste à détecter des vues frontales de visages sur des images 64×64 qui chacune représente soit un visage centré, soit une image de fond quelconque.

Dans la section 4.2 nous décrivons précisément comment les arrangements sont itérativement combinés les uns avec les autres. Au 4.3 est décrit l'un des aspect les plus délicats de cet algorithme, qui réside dans le sous-échantillonnage des arrangements potentiels. La section 4.4 présente comment sont estimés les seuils $t(k)$ du détecteur, la section 4.5 présente des résultats sur un problème synthétique, et enfin la section 4.6 présente précisément la forme des tests élémentaires que nous utilisons pour les visages, ainsi que des résultats préliminaires sur des images de visages.

4.2 Algorithme de construction montant

Bien que la définition d'un arrangement ρ-décomposable soit descendante, la construction proprement dite est ascendante. Elle consiste à construire itérativement des arrangements de plus en plus complexes en combinant à chaque étape des arrangements moins complexes déjà construits (cf. fig 4.1). Les corrélations sont estimées à l'aide de $\hat{P}_\Gamma$, mesure empirique déduite de $\mathcal{L}$.

La construction est donc récursive : nous construisons d'abord une famille $\{X_i X_j\}$, puis une famille $\{X_i X_j X_k\}$, etc. Nous noterons $\mathcal{A}^*_\mathcal{L}(k, \rho)$ cette séquence de familles d'arrangements ρ-décomposables de plus en plus complexes.

Pour construire $\mathcal{A}^*_\mathcal{L}(2k, \rho)$ nous n'avons besoin que de $\mathcal{A}^*_\mathcal{L}(k, \rho)$; et pour $\mathcal{A}^*_\mathcal{L}(2k+1, \rho)$ que de $\mathcal{A}^*_\mathcal{L}(k, \rho)$ et $\mathcal{A}^*_\mathcal{L}(k+1, \rho)$.

Nous ne construisons pas directement de cette manière les $\mathcal{A}_\mathcal{L}(k, \rho)$ car nous voulons que ces derniers soient de cardinaux réduits (une centaine d'arrangements), alors qu'il

est nécessaire de conserver un grand nombre d'arrangements de chaque complexité (de l'ordre d'un millier) pour pouvoir atteindre finalement une complexité suffisante. Nous noterons M le cardinal maximum des $\mathcal{A}^*_{\mathcal{L}}(k,\rho)$.

Considérons le cas pair. Notons $\hat{\rho}$ la corrélation estimée à partir de $\mathcal{L}$. Soit $\mathcal{A}^{**}_{\mathcal{L}}(2k,\rho)$ l'ensemble de tous les arrangements $A_1 \bigcup A_2$ où

- $A_1, A_2 \in \mathcal{A}^*_{\mathcal{L}}(k,\rho)$;
- $\hat{\rho}(X_{A_1}, X_{A_2}) \geq \rho$;
- $supp\ X_{A_1} \bigcap supp\ X_{A_2} = \emptyset$.

Par définition, si $\mathcal{A}^*_{\mathcal{L}}(k,\rho)$ ne contient que des arrangements que l'on a empiriquement estimés ρ-décomposables, alors $\mathcal{A}^{**}_{\mathcal{L}}(2k,\rho)$ aussi. On construit ensuite $\mathcal{A}^*_{\mathcal{L}}(2k,\rho)$ en sous-échantillonnant M arrangements dans $\mathcal{A}^{**}_{\mathcal{L}}(2k,\rho)$, et $\mathcal{A}_{\mathcal{L}}(2k,\rho)$ en en sous-échantillonnant m dans $\mathcal{A}^*_{\mathcal{L}}(2k,\rho)$. Ces sous-échantillonnages sont décrits dans la section 4.3.

Le processus de construction est initialisé en considérant les arrangements de complexité 1 (donc les tests élémentaires), et il se termine quand on rencontre le premier k tel qu'on ne peut pas couvrir les images de visages à l'aide d'arrangements de taille k.

4.3 Sous-échantillonnage intelligent

Nous voulons sélectionner un sous-ensemble $\mathcal{A}^*_{\mathcal{L}}(2k,\rho)$, au plus de taille M. Généralement

$$M \ll |\mathcal{A}^{**}_{\mathcal{L}}(2k,\rho)| \ll M^2$$

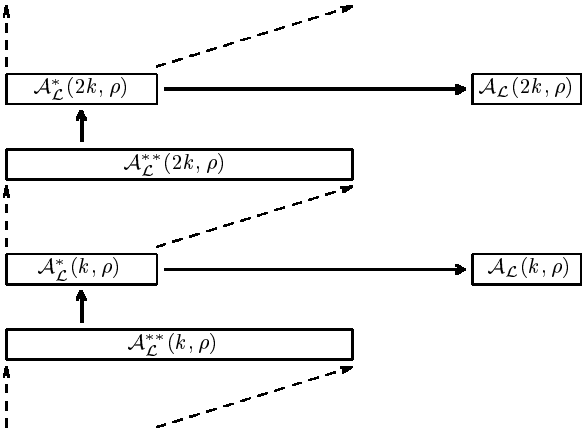


FIG. 4.1: *Les arrangements sont construits de manière itérative. On combine deux par deux les arrangements de complexité k déjà construits (ensemble $\mathcal{A}^*_{\mathcal{L}}(k,\rho)$) pour créer des arrangements de complexité $2k$ vérifiant la propriété de ρ-décomposition (ensemble $\mathcal{A}^{**}_{\mathcal{L}}(2k,\rho)$). On sous-échantillonne alors parmi ces derniers ceux que l'on conserve pour continuer le processus (ensemble $\mathcal{A}^*_{\mathcal{L}}(2k,\rho)$). Enfin, on sous-échantillonne l'ensemble des arrangements utilisés pour estimer Z_{2k} (ensemble $\mathcal{A}_{\mathcal{L}}(2k,\rho)$).*

Si $|\mathcal{A}^{**}_{\mathcal{L}}(2k,\rho)| \leq M$, alors $\mathcal{A}^*_{\mathcal{L}}(2k,\rho) = \mathcal{A}^{**}_{\mathcal{L}}(2k,\rho)$. Dans le cas contraire, il est nécessaire de sélectionner un sous-ensemble de taille M.

La sélection de ce sous-ensemble de $\mathcal{A}^{**}_{\mathcal{L}}(2k,\rho)$ est un point assez délicat, puisque l'ensemble $\mathcal{A}^*_{\mathcal{L}}(2k,\rho)$ ainsi créé est utilisé d'une part pour la constitution des $\mathcal{A}_{\mathcal{L}}(2k,\rho)$, qui servent à estimer les Z_k (c'est à dire pour la règle de décision du détecteur), mais aussi pour la construction des arrangements plus complexes. L'idée essentielle est de conserver une grande variété géométrique et statistique dans les arrangements sélectionnés. De cette manière, on conserve des arrangements qui ne sont pas très intéressants en eux même, mais qui permettent d'en construire plus tard qui le seront.

Soit un ensemble $\mathcal{A}$ d'arrangements, et soit m le nombre d'éléments que l'on veut sous-échantillonner dans $\mathcal{A}$. Le sous-échantillonnage fonctionne en choisissant successivement les éléments à mettre dans le sous-ensemble de taille m résultat. Formellement, soit une règle de sélection :

$$rs : \mathcal{P}(\mathcal{A}) \times \mathcal{P}(\mathcal{A}) \to \mathcal{A}$$

on pose

$$\begin{aligned} \mathcal{A}_0 &= \emptyset \\ \mathcal{A}_{k+1} &= \mathcal{A}_k \cup rs(\mathcal{A}_k, \mathcal{A} \setminus \mathcal{A}_k) \end{aligned}$$

Le terme $\mathcal{A}_{m+1}$ de cette suite sera le résultat du sous-échantillonnage.

Le premier critère que doit vérifier rs est de sélectionner des arrangements de manière à ce qu'il y en ait un grand nombre présents sur toutes les images de visages [1] :

$$rs^0(\mathcal{A}, \mathcal{B}) = \arg\max_{B \in \mathcal{B}} \left\{ \min_{I \in \mathcal{L}} \left\{ X_B(I) + \sum_{A \in \mathcal{A}} X_A(I) \right\} \right\}$$

L'expression $X_B(I) + \sum_{A \in \mathcal{A}} X_A(I)$ représente le nombre d'arrangements présents sur l'image I dans le cas où B est incorporé à la famille des arrangements à conserver. On cherche à sélectionner le B qui maximise la valeur minimale que prend cette expression sur l'ensemble d'apprentissage.

Si plusieurs arrangements vérifient cette propriété, celui qui a la probabilité de présence la plus élevée est sélectionné. Cette règle de sélection accumule des arrangements sur certaines zones du visage, plus riches en bords, comme par exemple la bouche. Malheureusement, ces zones ne correspondent pas à des structures étendues, comme l'est le contour de la tête par exemple. Pour prévenir cette dégénérescence, il faut forcer la fonction de sélection à conserver des arrangements variés. Nous proposons

1. La ρ-décomposibilité assure qu'individuellement, les structures sont plus probables sur les images de visages que sur d'autres images, mais ne confère aucune propriété globale à l'ensemble de structures construit.

un critère géométrique qui interdit qu'un même pixel de l'image soit utilisé par un grand nombre d'arrangements ; de cette manière, les arrangements doivent se répartir plus uniformément dans l'image. En notant σ le nombre maximal d'arrangements qui peuvent avoir un pixel donné en commun dans leur support, on modifie la règle rs^0 décrite ci-dessus :

$$rs(\mathcal{A}, \mathcal{B}) = rs^0(\mathcal{A}, \{B \in \mathcal{B} : \forall x \in supp\ X_B,\ ||\{A \in \mathcal{A}, x \in supp\ X_A\}|| < \sigma\})$$

4.4 Estimation des seuils $t(k)$

Nous avons vu au 3.4 que chacun des tests du détecteur vérifie que le nombre Z_k d'arrangements d'une certaine complexité k présents sur l'image dépasse bien un seuil donné $t(k)$.

Ces seuils doivent être fixés de manière à nous assurer un taux de faux négatifs nul. Un estimateur naturel de ce seuil $t(k)$ a la forme :

$$\hat{t(k)} = max \left\{ t : \hat{P}_\Gamma \left(\sum_{A \in \mathcal{A}_\mathcal{L}(\rho,k)} X_A \geq t \right) = 1 \right\}$$

Bien que l'on puisse craindre que cette définition surestime le seuil, nos expériences montrent que même avec une base d'apprentissage réduite, l'utilisation des $t(k)$ ainsi calculés permet d'obtenir $P_\Gamma(f_\Gamma = 0) \approx 0$, contrainte fondamentale dans notre approche du problème.

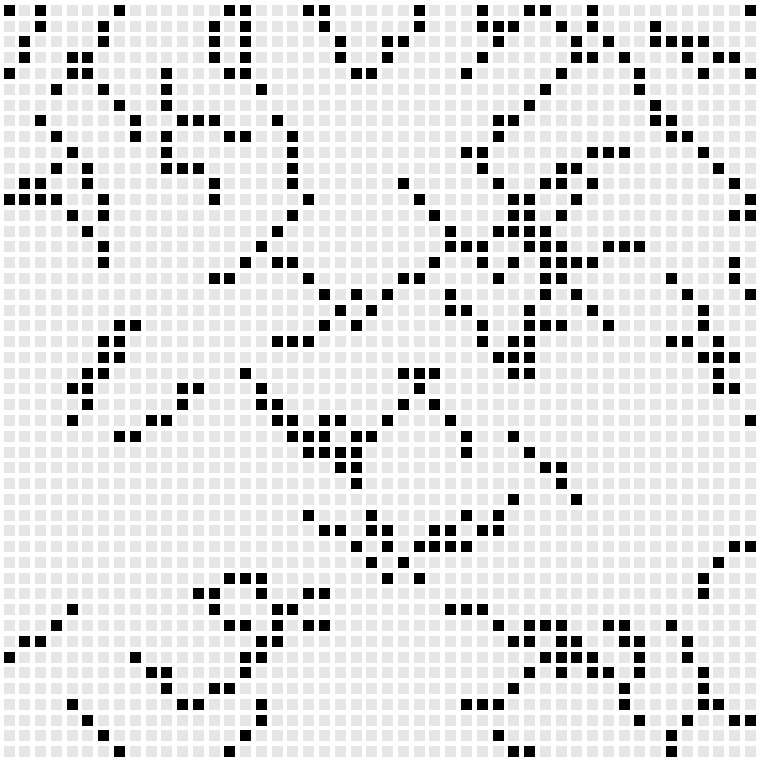


FIG. 4.2: *Une scène du "monde des V".*

4.5 Expériences sur le monde des "V"

4.5.1 Modèle

Cette expérience a été conçue pour mettre l'accent sur la confusion entre l'objet recherché et le fond, en introduisant une grande similitude locale entre les deux types d'images. Nous voulions, de plus, tester notre algorithme dans un contexte que nous contrôlions très précisément. Cela signifie que $\mathcal{L}$ est réellement un échantillon aléatoire de loi P_Γ et que les taux d'erreurs $P_0(f = 1)$ et $P_\Gamma(f = 0)$ de n'importe quel classificateur, même celui basé sur le maximum de vraisemblance, peuvent être estimés très précisément.

Les "scènes" sont binaires et composées entièrement de lignes courtes et irrégulières

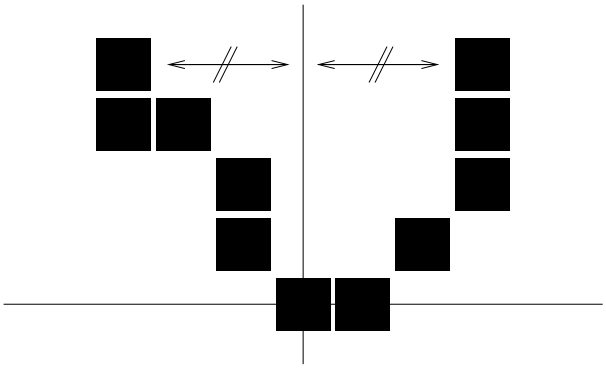


FIG. 4.3: *Un "V" consiste en deux lignes de même longueur, issues de la même origine et dont les extrémités vérifient une contrainte de symétrie.*

(cf. figure 4.2). Ces lignes sont des réalisations de marches aléatoires ; les probabilités de transitions orientent le trait soit dans la direction nord-ouest, soit dans la direction nord-est. Les scènes sont générées de la manière suivante : on commence par choisir un ensemble aléatoire de points de départ $x_1, x_2, \ldots$ dans l'ensemble de l'image, ensuite on choisit pour chaque x_i une des deux directions, NO ou NE, et on génère une marche aléatoire de longueur m partant de x_i. Dans le cas NO, le déplacement se fait dans les directions O, NO ou N avec les probabilités $\frac{1}{4}$, $\frac{1}{2}$, $\frac{1}{4}$; dans le cas NE ces directions sont E, NE, N. Les emplacements des "V" sont également choisi aléatoirement. Les "V" consistent en deux lignes diagonales, une NO, une NE, qui partent de la même origine, et dont les extrémités doivent vérifier une contrainte de symétrie (cf. figure 4.3). La seule contrainte sur la pose est donc que la pointe du "V" doit se trouver à une position précise.

La détection de ces "V" est difficile parce que les instances de l'objet sont constituées des mêmes composants que le fond, agencés de manière spécifique. Les "V" et le fond ne peuvent pas être séparés en se basant simplement sur des caractères locaux ; la corrélation lointaine doit être prise en compte. D'un autre côté il n'est pas nécessaire de considérer des représentations très denses, comme par exemple un modèle complet d'un "V" ; quelques pixels orientés NO ou NE donnent déjà une indication importante de la présence d'une ligne. Finalement, il n'est pas du tout évident de trouver comment

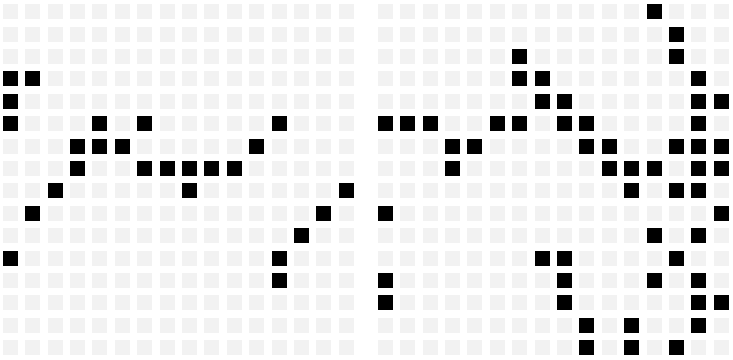


FIG. 4.4: *Gauche : une image de "V" prise dans l'ensemble d'apprentissage. Droite : une image du fond.*

un algorithme pourrait "découvrir" la contrainte de symétrie ou gérer les énormes différences entre les instances de "V".

4.5.2 Résultats

Nous générons une grande scène dont nous extrayons 1.000 sous-images de taille 16×16 centrées sur des emplacements de "V" ; donc $|\mathcal{L}| = 1.000$. Le taux d'erreur minimal semble être autour de 2%. Nous l'obtenons en connaissant le modèle géométrique et en testant la présence de deux lignes diagonales de même longueur et aux extrémités symétriques. Ceci décrit le classificateur par maximum de vraisemblance :

$$f_{ML}(I) = I_{\{P_\Gamma(I) > P_0(I)\}}$$

parce que $P_\Gamma(I) > P_0(I)$ pour n'importe quelle image de "V" et $P_\Gamma(I) = 0$ pour n'importe quelle image de fond. Comme tous les "V" sont trouvés, les erreurs sont toutes des faux positifs.

Nous utilisons des détecteurs de bords simplifiés pour ce contexte. Un bord est ici constitué par deux pixels adjacents en diagonale. Chaque test X_i possède une position

x_i dans la grille 16×16 et une orientation $\theta_i \in \{NW, NE\}$; $X_i = 1$ s'il y a un bord dans la direction θ_i quelque part dans le voisinage 2×2 de x_i. Des tests de ce type rendent le classificateur stable aux petites perturbations de la forme des "V", mais sont assez précis pour permettre d'utiliser la propriété de symétrie.

L'algorithme de construction des arrangements n'est pas exactement le même que celui que nous avons présenté jusque là, il s'agit d'une version antérieure qui sélectionne un peu moins finement les arrangements à conserver (en particulier, il n'y a pas la contrainte qui empèche un grand nombre d'arrangements d'avoir un bord en commun).

Les arrangements qui sont construits en se basant sur $\mathcal{L}$ s'accumulent sur les "V" et reconstruisent leurs formes (cf. fig 4.5). Pour des arrangements A de petites tailles, les tests sont localisés la plupart du temps d'un seul coté du "V", alors que lorsque les arrangements sont plus grands, la corrélation lointaine est exploitée et A contient des pixels des deux cotés. Les distributions de probabilités de la variable Z_k sont très différentes sous P_0 et sous P_Γ et peuvent être clairement séparées ; la forme des fonctions de répartition ressemble beaucoup à celle que l'on rencontre dans les expériences sur les visages (cf. figure 4.14). Au final, l'erreur *combinée* la plus faible que nous avons pu obtenir est $P_0(f_\Gamma = 1) + P_\Gamma(f_\Gamma = 0) = .07$; elle est un peu plus élevée si nous forçons la contrainte $P_\Gamma(f_\Gamma = 0) = 0$. Il nous semble que séparer ces deux populations est très difficile. Par exemple, l'algorithme du "plus proche voisin" donne de mauvais résultats, même avec une métrique "intelligente" (i.e. invariante).

4.6 Expérience avec des visages

4.6.1 Ensemble d'apprentissage

Pour valider l'algorithme sur des images réelles, nous présentons ici une première série de résultats obtenus sur une base d'images de visages, sans appliquer l'algorithme sur

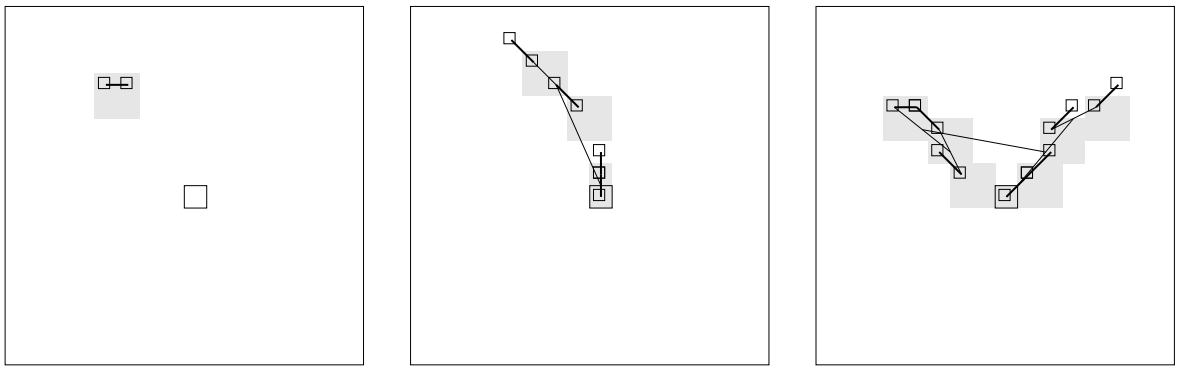


FIG. 4.5: *Exemples d'arrangements de tailles* 1, 4 *et* 7 *construits sur les images de "V".*

des scènes complètes.

L'ensemble d'apprentissage $\mathcal{L}$ est construit à partir de la base de données de visages Olivetti dont nous extrayons 300 images, correspondant à 10 vues différentes de 30 personnes représentées. Nous avons marqué pour chacune de ces images la position des yeux et de la bouche. Nous appliquons une rotation, une translation et un homothétie sur les images en niveaux de gris pour obtenir des images normalisées en taille (64×64 pixels), représentant des visages dont le point entre les yeux est à une position fixe, et tels que la distance entre les yeux s soit égale à 15 pixels et dont les yeux sont alignés horizontalement $\theta = 0$. Cette expérience correspond à une pose très contrainte, donc à un sous-ensemble $\Gamma \subset \Theta$ très réduit, plus petit que les sous-ensembles les plus fins que nous utilisons dans le traitement complet des scènes.

4.6.2 Invariance photométrique, détecteurs de bords

Une partie importante des variations de l'apparence d'un visage sont dues aux variations d'éclairage, qui provoquent des modification des niveaux de gris. Un détecteur qui utilise directement les niveaux de gris comme représentation de l'image n'est pas invariant à de tels changements d'apparence. Les techniques à base de réseaux de

neurones par exemple nécessitent donc souvent une phase de pré-traitement qui permet de palier à cette absence d'invariance (Rowley 1999). Les deux traitements les plus standards consistent d'une part à éliminer la composante linéaire de l'image, et ensuite à faire une égalisation des niveaux de gris (Sung 1996) (c'est à dire appliquer une transformation croissante des niveaux de gris pour rendre toutes les valeurs de gris équiprobables).

Un tel traitement est coûteux puisqu'il fait intervenir des fonctions de tous les pixels de la sous-image, et doit être réitéré pour chaque position de la fenêtre 64×64. Pour l'éviter, nous proposons d'utiliser non pas les niveaux de gris des pixels, mais des fonctions booléennes de ces derniers, fonctions qui ont de grandes propriétés d'invariance. Notre algorithme est semblable à celui utilisé dans (Amit & Geman 1999).

Les fonctions booléennes que nous avons choisies sont des détecteurs de bords primitifs. Elles sont définies à l'aide de comparaisons entre des différences de niveaux de gris des pixels. Nous définissons 8 types de bords différents, correspondant à quatre orientations et deux polarisations (foncé vers clair ou clair vers foncé) pour chaque orientation :

- Deux types de bords verticaux ϵ_1 et ϵ_2
- Deux types de bords horizontaux ϵ_3 et ϵ_4
- Deux types de bords diagonaux haut-gauche ϵ_5 et ϵ_6
- Deux types de bords diagonaux bas-gauche ϵ_7 et ϵ_8

Formellement, posons :

$$
\begin{aligned}
\delta(x,y) &= |I(x,y) - I(x+1,y)| \\
\delta_1(x,y) &= |I(x,y) - I(x,y-1)| \\
\delta_2(x,y) &= |I(x,y) - I(x-1,y)| \\
\delta_3(x,y) &= |I(x,y) - I(x,y+1)| \\
\delta_4(x,y) &= |I(x+1,y) - I(x+1,y-1)| \\
\delta_5(x,y) &= |I(x+1,y) - I(x+2,y)| \\
\delta_6(x,y) &= |I(x+1,y) - I(x+1,y+1)|
\end{aligned}
$$

Alors, on définit les deux types de bords verticaux sous la forme de fonctions booléennes de l'images ϵ_1 et ϵ_2 :

$$
\begin{aligned}
\epsilon_1(x,y) &= 1 \quad \text{ssi} \begin{cases} Card\left(\{i \ : \delta_i(x,y) < \delta(x,y)\}\right) \geq 5 \\ |I(x,y) > I(x+1,y)| \geq \delta_{\min} \\ I(x,y) > I(x+1,y) \end{cases} \\
&= 0 \quad \text{sinon}
\end{aligned}
$$

Le bord ϵ_2 est défini de la même manière avec la contrainte $I(x,y) < I(x+1,y)$. Le seuil $\delta_{\min}$ permet de rajouter un filtrage et d'éviter l'apparition de bords dans des zones de l'image presque constantes. La définition des autres bords est similaire. La figure 4.6 représente quelles sont les comparaisons impliquées pour la détection d'un bord vertical et d'un bord diagonale.

Mise à part la comparaison avec $\delta_{\min}$, ces définitions ne dépendent que de comparaisons de différences, et sont donc invariantes pour n'importe quelle transformation affine croissante des niveaux de gris.

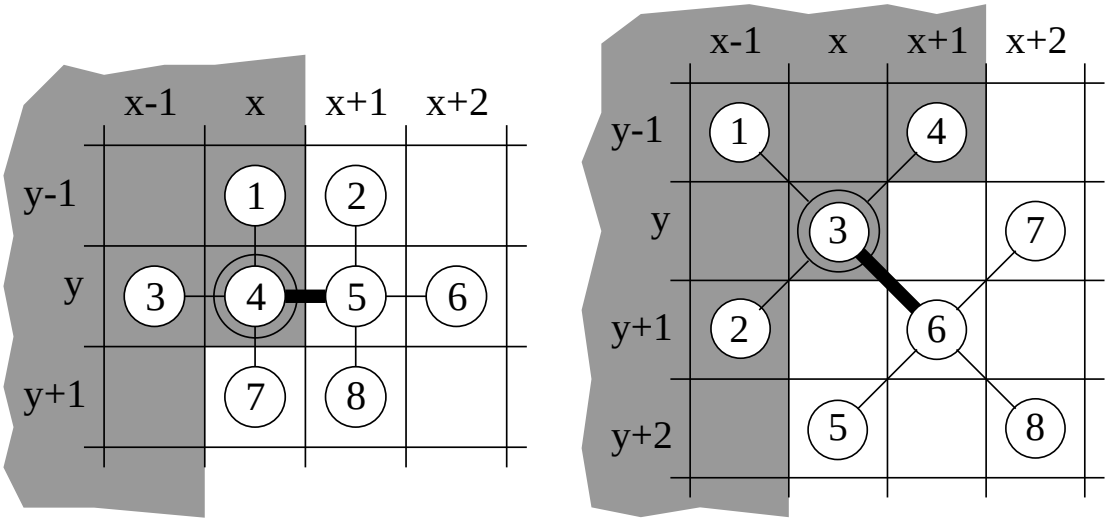


FIG. 4.6: *Les détecteurs de bords comparent les niveaux de gris des pixels entre eux. A gauche est illustrée la définition d'un bord vertical ϵ_1 ou ϵ_2 au pixel 4 : la valeur absolue de la différence entre les niveaux de gris des deux pixels reliés par l'arête large doit être supérieure à au moins cinq des valeurs absolues des différences entre les niveaux de gris des pixels reliés par une arête étroite (qui sont au nombre de six). Ce bord est polarisé en fonction de la différence entre les deux pixels reliés par l'arête épaisse : ϵ_1 correspond à une polarisation clair vers sombre, et ϵ_2 à une polarisation sombre vers clair. A droite, la figure équivalente pour un bord diagonale ϵ_5 ou ϵ_6.*

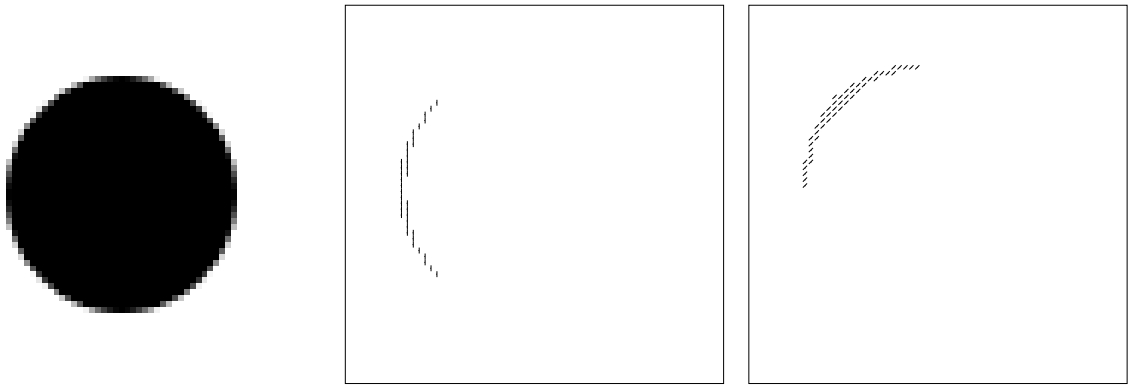


FIG. 4.7: *Exemple d'extraction des bords ϵ_1 et ϵ_7.*

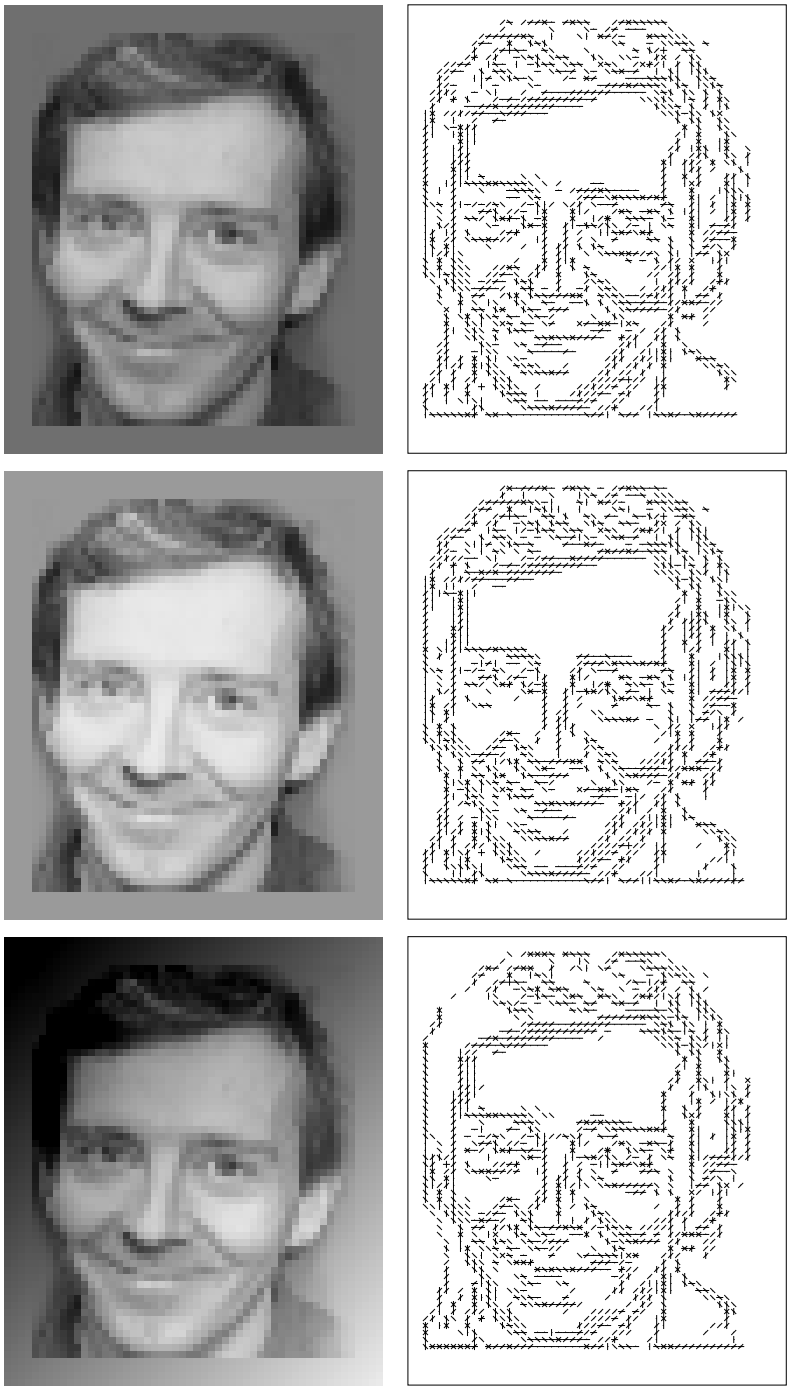


FIG. 4.8: *Malgré les variations fortes des niveaux de gris, les détections de bords varient peu.*

4.6.3 Invariance aux déformations locales

Nous avons vu au chapitre 5 que les variations globales de la pose du visage dans la sous-image étaient gérées en combinant plusieurs détecteurs, dédiés à des sous-ensembles de poses plus ou moins contraintes. L'invariance aux déformations locales de l'image, quant à elle, peut être introduite dans la définition des tests élémentaires qui composent la première couche de chaque détecteur dédié.

Un test élémentaire (c'est à dire un arrangement ne faisant intervenir qu'un seul bord) $X_{(x,y),i,t}$ est défini par ses 4 paramètres : sa position (x, y) dans l'imagette 64×64, un type de bord $i \in \{1, \ldots, 8\}$ et une tolérance $t \in N^*$. La valeur d'un tel arrangement élémentaire sera déterminée par une disjonction des ϵ_i sur t pixels, dans une direction orthogonale à la direction du bord, à proximité de (x, y) (cf. 4.9). Ainsi, un seul test élémentaire "capture" plusieurs positions d'un bord d'une direction donnée. Par exemple, si $i = 1$, et $t = 4$:

$$X_{(x,y),1,4} = \max\{\epsilon_1(x-2, y),\ \epsilon_1(x-1, y),\ \epsilon_1(x, y),\ \epsilon_1(x+1, y)\}$$

4.6.4 Sélection des tolérances des tests élémentaires

La tolérance aux déformations que nous venons de décrire est donc fixée par un paramètre qui définit la taille de la zone sur laquelle se fait la disjonction. Plus cette zone est étendue, plus la probabilité marginale du test élémentaire est élevée. Pour déterminer automatiquement ce paramètre, nous le fixons à la plus petite valeur qui permet d'atteindre une probabilité de $\frac{1}{2}$. Précisément, en définisant :

$$t(x, y, i) = \min \left\{ t : \hat{P}_\Gamma(X_{(x,y),i,t}) \geq \frac{1}{2} \right\}$$

nous prenons finalement en chaque point le test élémentaire avec cette tolérance, si

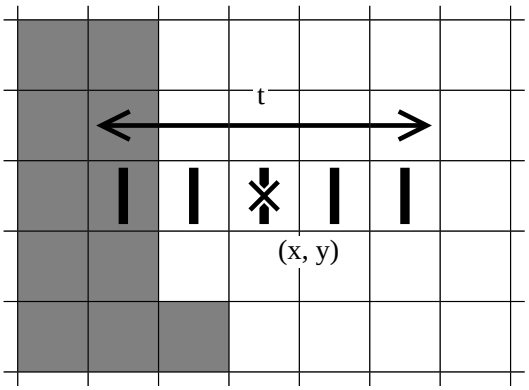


FIG. 4.9: *Les tests élémentaires sont des disjonctions des détecteurs de bords ; ils admettent donc une tolérance en localisation, dans une direction orthogonale à celle du bord. Cette figure illustre la définition de* $X_{(x,y),1,5}$*, test élémentaire dédié à un bord vertical, situé à l'emplacement* (x, y) *de l'image, avec une tolérance de* $t = 5$ *pixels.*

elle est inférieure à 7 :

$$\mathcal{A}^{**}_{\mathcal{L}}(1, \rho) = \left\{ X_{(x,y),i,t(x,y,i)} : t(x, y, i) \leq 7 \right\}$$

Cette définition va produire un ensemble de tests élémentaires qui seront peu tolérants à proximité du points de référence (entre les yeux), et plus tolérants loin de ce point. Cette tolérance plus importante est nécessaire pour "capturer" de la structure avec une probabilité supérieure à $\frac{1}{2}$. Sur la figure 4.10 sont représentées ces tolérances pour les bords de type ϵ_3 (horizontal clair vers foncé) et ϵ_7 (diagonale bas gauche clair vers foncé).

4.6.5 Arrangements appris

Les figures 4.11 et 4.12 montrent, chacune, 9 arrangements de complexité 5 et 9 respectivement. Ces arrangements ont été appris à partir de $\mathcal{L}$ et sont représentatifs des milliers que nous construisons. En fait, il apparaît que tous les arrangements sont

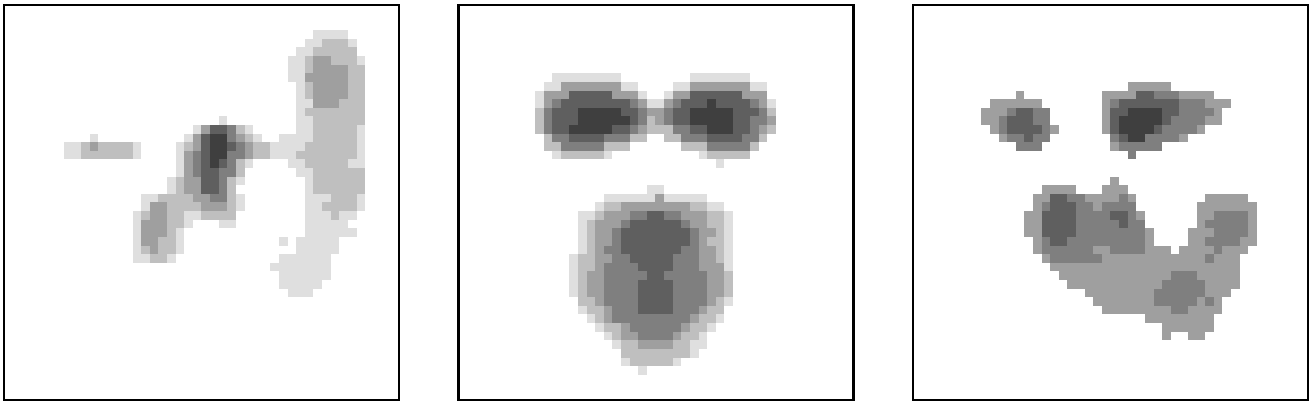


FIG. 4.10: *Représentation de la tolérance des tests élémentaires en fonction de leur position dans la grille de référence. Le gris le plus foncé représente une tolérance de* 1 *pixel, le gris le plus clair une tolérance de* 7 *pixels. Le blanc indique les zones où ne se trouve aucun test élémentaire. L'image de gauche montre la tolérance pour les bords* ϵ_1 *(verticaux, clair vers foncé), celle du milieu les bords de type* ϵ_3 *(horizontaux clair vers foncé), et enfin celle de droite pour les bords de type* ϵ_7 *(diagonale bas gauche, clair vers foncé).*

composés de bords situés autour des yeux, de la bouche, du nez et sur les contours du visage. Cette accumulation est illustrée par la figure 4.13 sur laquelle l'intensité du noir de chaque pixel est proportionnel au nombre d'arrangements qui dépendent d'un test élémentaire situé à cet endroit.

Une mesure du pouvoir discriminant des arrangements est représentée sur la figure 3.5 du chapitre précédent. L'axe vertical indique le logarithme en base 10 des probabilités et l'axe horizontal la taille des arrangements. Nous avons construit des arrangements tant que le processus le permettait, sans vérifier le critère d'arrêt énoncé à la fin de la section 4.2. Ainsi, nous avons pu atteindre une complexité de 35.

Nous échantillonnons aléatoirement dix arrangements A pour chaque $k = 1, \ldots, 35$ et nous calculons les probabilités des évènements $P_0(X_A = 1)$ et $P_\Gamma(X_A = 1)$. Les nuages sont représentés respectivement avec des $+$ et des $\diamond$. Comme on peut le remarquer, les probabilités de présence des arrangements sur les visages sont bien plus élevées que la borne théorique.

Finalement, la figure 4.14 montre les distributions de $P_0(Z_k = k)$ et $P_\Gamma(Z_k = k)$, pour $k = 5$ et $k = 8$, telles qu'elles peuvent être estimées sur les données. Comme on peut le voir à nouveau, les arrangements ρ-décomposables de tailles modestes sont beaucoup plus rares sous P_0 que sous P_Γ. Par exemple, les visages et le fond sont très bien séparés par Z_5.

4.6.6 Estimation des corrélations

Les arrangements ρ-décomposables que l'on construit sont décomposables sous $\hat{P}_\Gamma$. On peut se demander s'ils le sont sous P_Γ. Certains le sont pour un ρ plus élevé, d'autres ne le sont pas. Soit $\rho_0 = .1$; c'est cette valeur que nous utilisons dans nos expériences. Pour chaque arrangement $A \in \mathcal{A}_\mathcal{L}(k, \rho_0)$ nous disposons d'une ρ_0-décomposition. Nous pouvons utiliser des données supplémentaires (l'ensemble de test) pour ré-estimer les corrélations et vérifier que l'arrangement est toujours ρ_0-

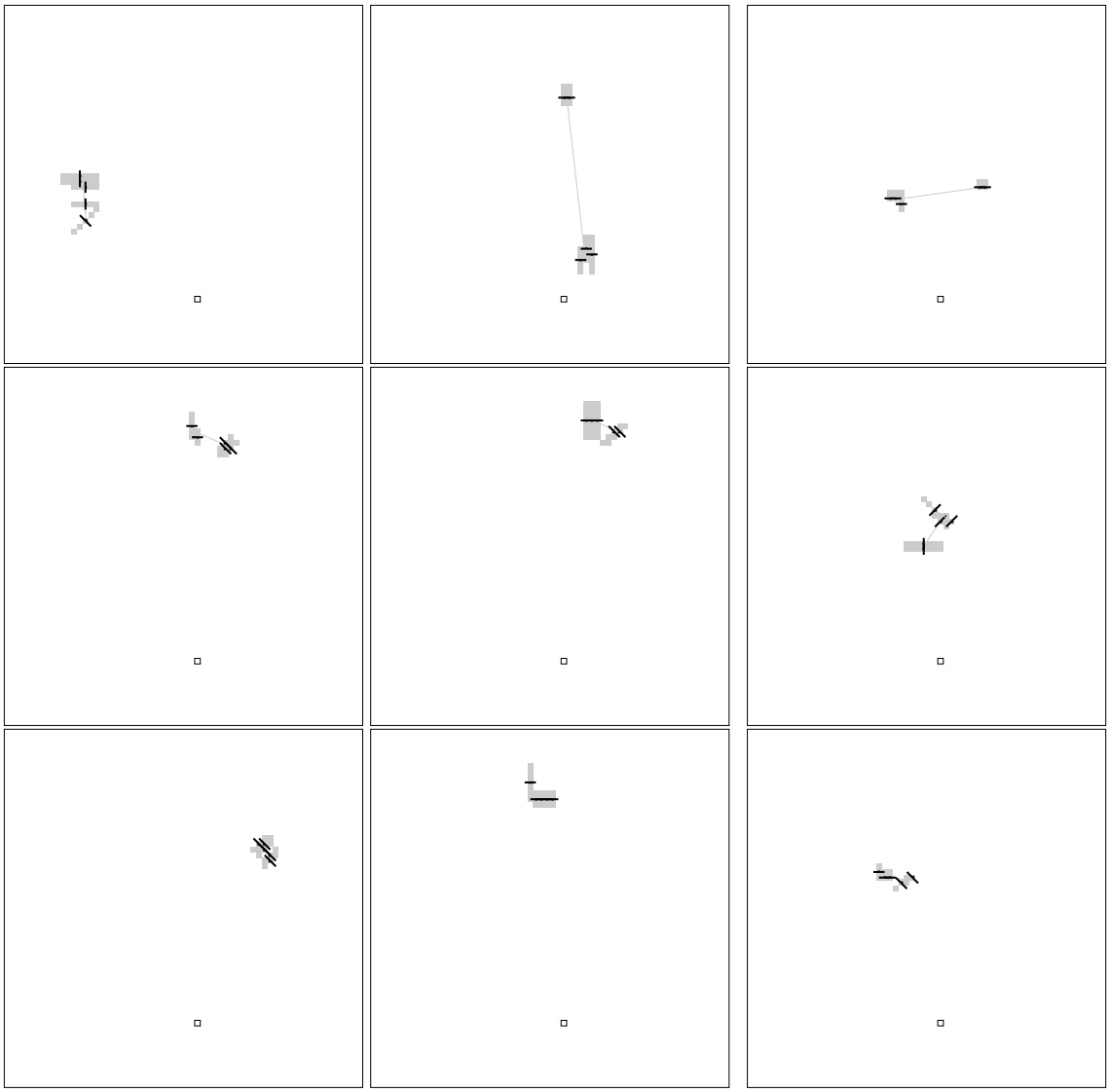


FIG. 4.11: *Exemples d'arrangements de taille* 5 *sur les visages. Les zones grisées indiquent les zones de l'images utilisées par les détecteurs de bords.*

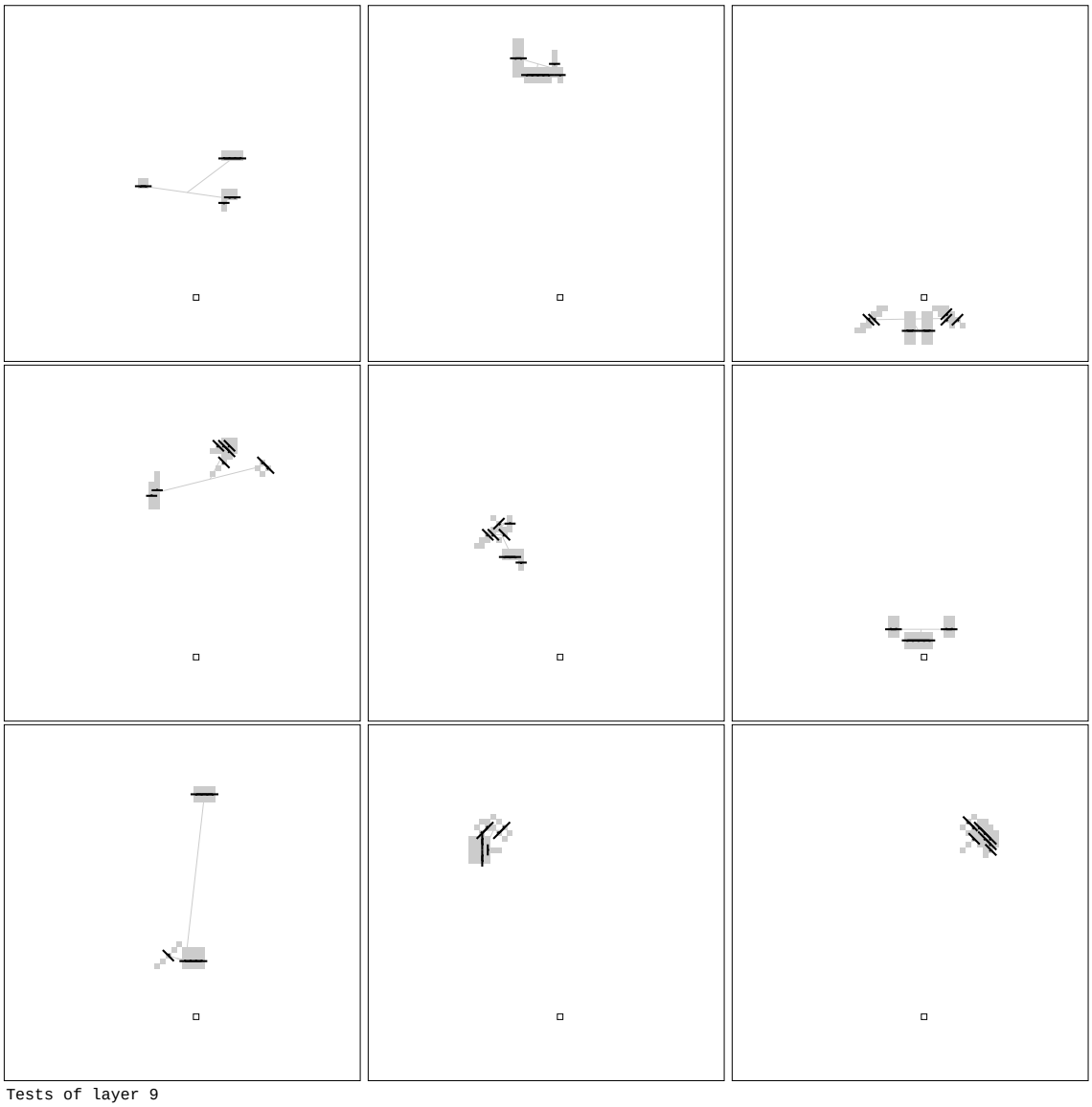


FIG. 4.12: *Exemples d'arrangements de taille* 9 *sur les visages.*

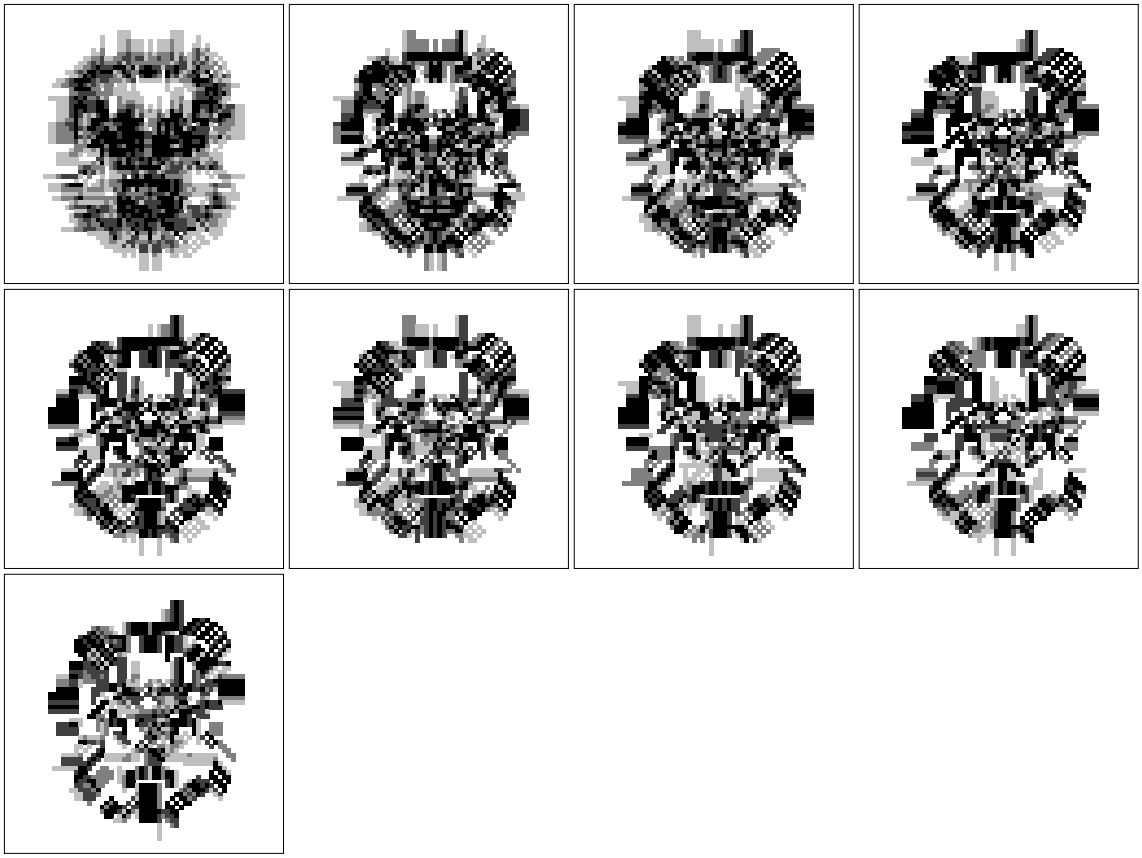


FIG. 4.13: *Accumulation des arrangements. Chaque carré représente la localisation des arrangements d'une complexité donnée. Le niveau de gris de chaque pixel dépend du nombre d'arrangements qui l'utilisent. Le noir correspond au nombre maximal d'arrangements utilisant un pixel donné.*

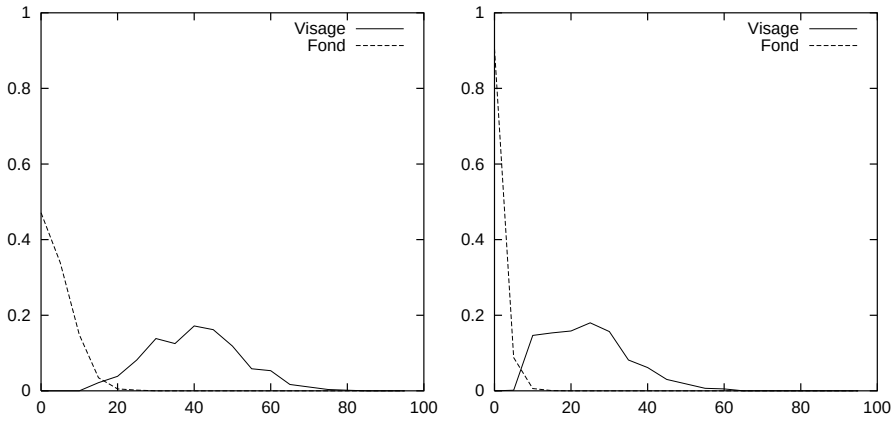


FIG. 4.14: *Distributions de Z_5 et Z_8 sur les visages*

décomposable.

Nous pouvons même estimer $\rho_{max}(A)$, la valeur maximale de ρ pour laquelle la décomposition de l'arrangement A est une ρ-décomposition. Cette valeur peut être plus petite ou plus grande que ρ_0. Le résultat représenté sur la figure 4.15 est donc pessimiste puisqu'il pourrait exister une ρ-décomposition de A sans que la décomposition dont nous disposons en soit une. Nous trouvons par exemple que les décompositions de 95% des arrangements sont valides pour $\rho > 0$, 80% pour $\rho \geq .1$ et 45% pour $\rho \geq .2$.

4.6.7 Taux d'erreur

Nous estimons ici le le taux d'erreur non pas en en testant le classificateur sur une scène complète, mais en utilisant les 100 images de visages de la base de données de test et un grand nombre d'images 64×64 diverses ne contenant pas de visage, récupérées sur le WWW. Nous présenterons dans le chapitre 6 les résultats sur des images de scènes complètes.

Grâce à la robustesse de l'estimation des seuils $t(1), \ldots, t(k)$, le nombre d'erreur

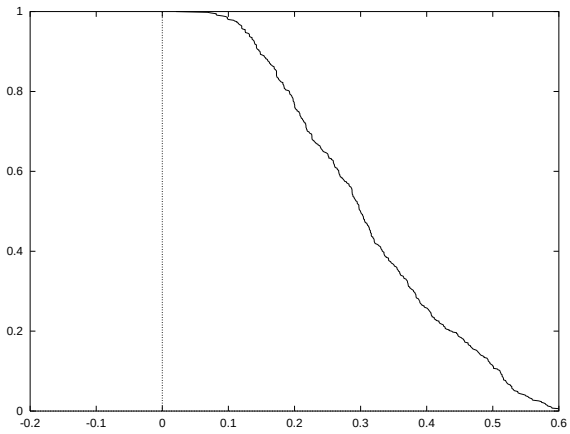


FIG. 4.15: *Proportion de décompositions apprises qui sont ρ-décomposables sur un ensemble de test.*

de faux négatifs peut être en pratique considéré comme nul. Le taux d'erreur de faux positifs, exprimé en fonction de la complexité $\hat{P}_0(Z_1 \geq t(1), \ldots, Z_k \geq t(k))$, est représenté sur la figure 4.16. Le comportement important est la forte décroissance lorsque k augmente ; la valeur finale (proche de 0.13%), est peu significative dans la mesure où elle ne correspond qu'à une seule étape du processus complet de détection hiérarchique.

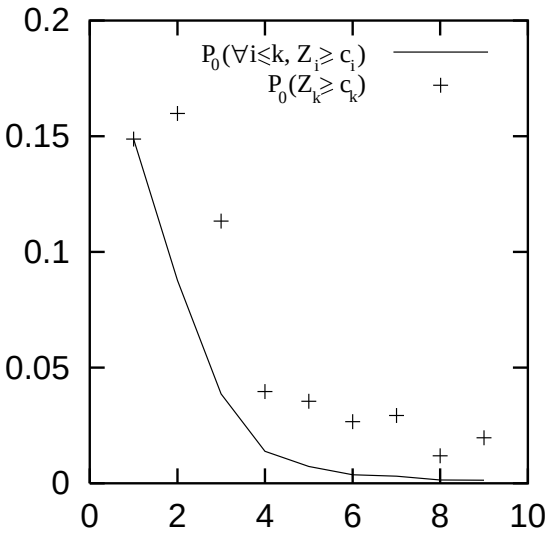


FIG. 4.16: *Taux d'erreurs de faux-positifs fonction de la complexité des arrangements. Les symboles + indiquent les probabilités $P_0(Z_k \geq t(k))$ séparément, et la ligne brisée indique la probabilité de la conjonction de ces tests jusqu'à k, $P_0(\forall i \leq k,\ Z_i \geq t(i))$. Comme on le voit, les tests ont des propriétés d'indépendance et leur conjonction a une probabilité de rejetter une image de fond plus élevée que le meilleur de ces tests pris séparément.*

Chapitre 5

Détecteur global

5.1 Introduction

Comme nous l'avons vu dans le chapitre 2, le détecteur final f combine plusieurs détecteurs dédiés à des sous-ensembles de l'espace de poses f_Γ :

$$f(I) \quad = \quad \delta_1 \left(\sum_{\theta \in \Theta} \prod_{\Gamma \downarrow \theta} f_\Gamma \right)$$

Ces détecteurs f_Γ ont été décrits dans le chapitre 3. La contrainte essentielle que la procédure d'apprentissage, décrite au chapitre 4, respecte, est de produire des détecteurs dédiés ayant un taux de faux négatifs nul ; cette contrainte étant respectée, plus un détecteur est tolérant aux variations globable de la position du visage, plus il a un taux d'erreur de faux positifs élevé.

Nous allons voir dans ce chapitre comment nous pouvons choisir les Γ afin de réaliser un détecteur final qui soit non seulement efficace algorithmiquement (et même optimal sous certaines hypothèses), mais qui de plus permet d'atteindre un taux d'erreur

minimum. La démarche consiste à partitionner dichotomiquement l'espace Θ afin de produire des cellules Γ hiérarchisées. A chacune de ces cellules sera associé un détecteur spécialisé, et le critère final de détection sera l'existence d'une séquence de détecteurs, associés à des cellules emboitées, qui détectent tous un visage.

L'intérêt de cette approche est double. Elle réside d'abord dans la quasi-indépendance de ces détecteurs sur des images qui ne représentent pas un des visages qu'ils doivent détecter. Le taux de faux positifs décroît ainsi, en première approximation, de manière exponentielle avec le nombre de détecteurs. De plus, cette combinaison de détecteurs constitue à nouveau une hiérarchisation de la représentation. Seules les images très ambiguës demandent l'utilisation d'un grand nombre de détecteurs. La plupart seront rejetées par les premiers, dédiés à des ensemble très peu contraints d'images. Enfin, comme nous le verrons, ce choix d'ensembles Γ permet d'organiser très efficacement les calculs puisque la plupart des f_Γ peuvent être mis en facteur dans les $\prod_{\Gamma \downarrow \theta} f_\Gamma$. Cette mise en facteur revient à trouver un ordre optimal d'utilisation des détecteurs dédiés.

Cette manière de combiner les détecteurs n'est pas spécialement conçue pour les détecteurs utilisant la ρ-décomposabilité que nous avons présentés dans les chapitres précédents. Elle partage avec ces détecteurs le concept d'organisation hiérarchique de la recherche, mais constitue une réponse générique au problème des la non-invariance des détecteurs aux variations globales de la pose. Comme nous le verrons dans la suite, les hypothèses que nous faisons sur les détecteurs dédiés sont très faibles. Ainsi, cette même approche pourrait parfaitement être mise en oeuvre avec des méthodes plus standards (réseaux de neurones, eigen faces, etc.).

Nous précisons dans la section 5.2 l'idée du partitionnement de l'espace des poses Γ_k^n que nous venons d'introduire. Dans la section 5.3 nous proposons un modèle pour le comportement des détecteurs dédiés X_k^n, et dans 5.5 nous justifions le choix du critère de détection utilisé pour combiner les détecteur dédiés dans un cadre statistique en montrant qu'avec notre modèle il est équivalent au maximum de vraisemblance. Dans la partie 5.6 nous définissons abstraitement la stratégie retenue pour implémenter ce

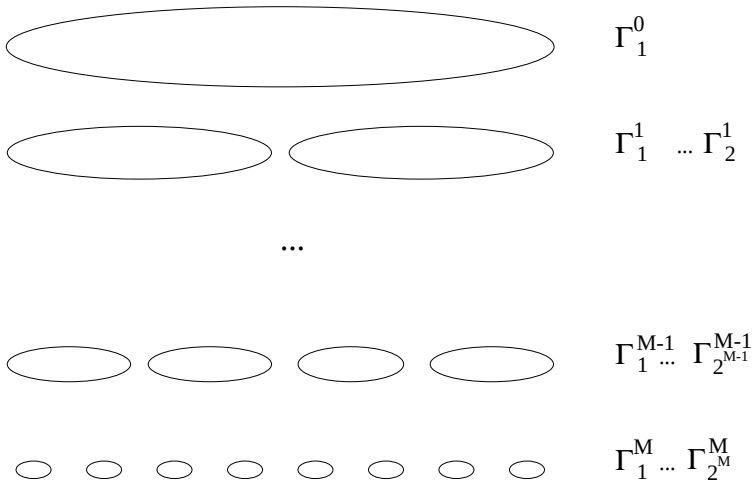


FIG. 5.1: *Les Γ_k^n constituent un partitionnement diadique de l'ensemble des poses Θ.*

critère, stratégie qui se ramène à un parcours en profondeur du partitionnement dichotomique de l'espace de poses. Enfin, dans la section 5.7, nous justifions ce parcours en profondeur en montrant son optimalité sous des hypothèses assez faibles.

5.2 Partitionnement diadique de l'espace des poses

Nous considèrons un partitionnement de Θ en cellules emboitées. Soit $\Gamma_k^n, 0 \leq n \leq M, 1 \leq k \leq 2^n$ ce partitionnement :

- $\Gamma_1^0 = \Theta$
- $\Gamma_k^n = \Gamma_{2k-1}^{n+1} \cup \Gamma_{2k}^{n+1}$
- $\Gamma_{2k-1}^{n+1} \cap \Gamma_{2k}^{n+1} = \emptyset$

Nous appellerons *cellules* les Γ_k^n. Nous considérons que pour chacune de ces cellules Γ_k^n, nous disposons d'un test booléen X_k^n, de la forme $\prod_{k=1}^K \delta_{t(k)}(Z_k)$, dédié aux images de

visages dont la pose est dans Γ_k^n. En pratique ce test sera un détecteur dédié construit sur une base d'apprentissage dont les images ont des poses contraintes. Nous noterons $\vec{X}=\{X_k^n\}_{n,k}$ le vecteur aléatoire constitué des X_k^n et $\vec{x}$ un vecteur de $\{0,1\}^{2^{M+1}-1}$.

Chacun de ces classificateurs a un coût, qui est défini comme le coût algorithmique nécessaire à sa mise en oeuvre. Ce coût représente très concrètement le nombre d'opérations nécessaires sur un ordinateur pour évaluer sa réponse. Nous faisons l'hypothèse dans la suite que le coût d'un X_k^n ne dépend que de n. Cette hypothèse est valable en première approximation.

5.3 Modèle

Les tests X_k^n sont des fonctions de l'image $X_k^n(I)$. Nous faisons les hypothèses suivantes sur la loi du vecteur (X_k^n) sous P_0 et sous les Q_i :

- Sous P_0, les X_k^n sont des variables indépendantes avec $P_0(X_k^n=0)=\beta_n$
- $\forall i$, sous Q_i, les X_k^n sont des variables indépendantes, avec $\forall n,k$:
  - Si $\{\theta_i\}\subset\Gamma_k^n$ alors $Q_i(X_k^n=1)=1$
  - Sinon $Q_i(X_k^n=1)=1-\beta_n$

Pour la loi de probabilité du fond P_0, les variables X_k^n sont des variables de bernoulli indépendantes de paramètres $1-\beta_n$. Pour les Q_i par les X_k^n sont également indépendants, et tous les X_k^n associés aux cellules dans lesquelles $\{\theta_i\}$ est inclus sont égaux à 1 avec une probabilité 1. Les autres X_k^n étant comme dans le cas de P_0, des bernoullis de probabilité $1-\beta_n$.

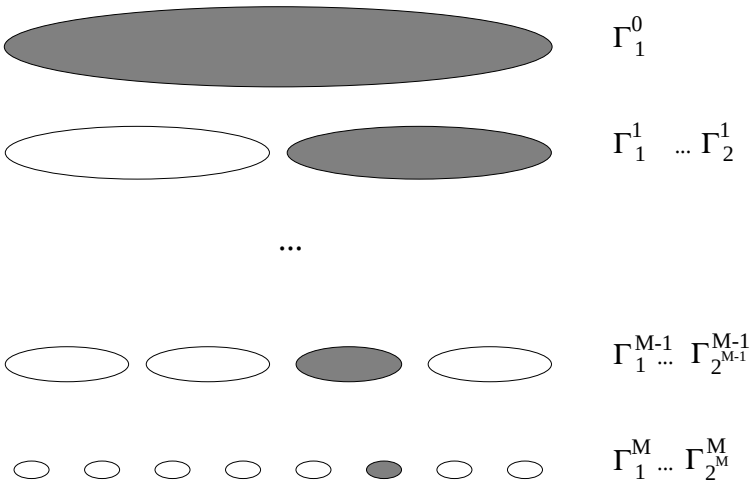


FIG. 5.2: *Le test de maximum de vraisemblance revient à tester s'il existe une suite d'ensembles diadiques emboités dont tous les tests associés sont égaux à* 1.

5.4 Optimalité du taux d'erreur

Comme nous l'avons dit dans la section 5.1, notre algorithme final détermine la classe d'une image (visage ou fond) en calculant s'il existe une séquence de cellules de poses emboitées depuis la cellule la plus grossière Γ_1^0 jusqu'à l'une des plus fines $\{\theta_i\}$ (cf. figure 5.2). Plus précisément, le critère de détection est :

$$\exists k,\ \forall (n,l),\ (\{\theta_k\} \subset \Gamma_l^n) \Rightarrow (X_l^n = 1) \qquad \text{(ce)}$$

Ce critère des *cellules emboitées* est équivalent au fait qu'il n'existe pas de partition de Θ en cellules dont les tests dédiés sont tous nuls (nous appellerons dans la suite *cellules nulles* de telles cellules). Nous noterons

$$\Lambda = \{\vec{x} \in \{0,1\}^{2^{M+1}-1} : \exists k,\ \forall (n,l),\ (\{\theta_k\} \subset \Gamma_l^n) \Rightarrow (x_l^n = 1)\}$$

Nous pouvons montrer que notre critére de détection correspond au taux de faux positifs minimum pour un taux de faux négatifs nul. Soit $\Phi : \{0,1\}^{2^{M+1}-1} \rightarrow \{0,1\}$ un détecteur. Si on fait l'hypothèse $\forall\ \vec{x} \in \Lambda,\ P_1(\vec{x}) > 0$, on a

$$\begin{aligned}
P_1(\Phi(\vec{X}) = 0) = 0 \quad &\Rightarrow \quad \sum_{\vec{x}} P_1(\Phi(\vec{X}) = 0 |\ \vec{X} = \vec{x}) P_1(\vec{X} = \vec{x}) = 0 \\
&\Rightarrow \quad \sum_{\vec{x} \in \Lambda} P_1(\Phi(\vec{X}) = 0 |\ \vec{X} = \vec{x}) P_1(\vec{X} = \vec{x}) = 0 \\
&\Rightarrow \quad \sum_{\vec{x} \in \Lambda} \delta(\Phi(\vec{x}) = 0) P_1(\vec{X} = \vec{x}) = 0 \\
&\Rightarrow \quad \forall\ \vec{x} \in \Lambda,\ \delta(\Phi(\vec{x}) = 0) = 0 \\
&\Rightarrow \quad \forall\ \vec{x} \in \Lambda,\ \Phi(\vec{x}) = 1
\end{aligned}$$

Donc si Φ a un taux de faux négatifs nul, il a un taux de faux positifs au moins égal à celui de notre détecteur.

5.5 Maximum de vraisemblance

Soit $\vec{x} = \{x_k^n\}_{n,k}$ une observation de $\vec{X} = \{X_k^n\}_{n,k}$. Définissons $\gamma(k)$ l'ensemble des indices (n,l) des cellules qui contiennent $\{\theta_k\}$:

$$\gamma(k) \quad = \quad \{(n,l), 0 \leq n \leq M, 1 \leq l \leq 2^n,\ \{\theta_k\} \subset \Gamma_l^n\}$$

On a, $\forall (n,l)$

$$\begin{aligned} P_0(X_l^n = x_l^n) &= (1 - x_l^n)\beta_n + x_l^n(1 - \beta_n) \\ P_0(\vec{X}{=}\vec{x}) &= \prod_{(n,l)} P_0(X_l^n = x_l^n) \end{aligned}$$

Donc $\forall\ \vec{x},\ P_0(\vec{X}{=}\vec{x}) > 0$.

De plus, $\forall k$

$$\begin{aligned} Q_k(X_l^n = x_l^n) &= \begin{cases} x_l^n & \text{si} \quad (n,l) \in \gamma(k) \\ (1 - x_l^n)\beta_n + x_l^n(1 - \beta_n) & \text{sinon} \end{cases} \\ Q_k(\vec{X}{=}\vec{x}) &= \prod_{(n,l)} Q_k(X_l^n = x_l^n) \end{aligned}$$

Dans le cas où une séquence de cellules emboitées dont les tests valent tous 1 n'existe pas $\forall i,\ \exists (n,l) \in \gamma(i),\ x_l^n = 0$, donc $Q_i(\vec{X}{=}\vec{x}) = 0$. Le maximum de vraisemblance choisit donc l'hypothèse P_0, soit "fond".

Sinon

$$\exists k,\ \forall (n,l) \in \gamma(k),\ x_l^n = 1$$

Donc

$$\begin{aligned} Q_k(\vec{X}{=}\vec{x}) &= \prod_{(n,l)} Q_k(X_l^n = x_l^n) \\ &= \prod_{(n,l) \notin \gamma(k)} Q_k(X_l^n = x_l^n) \\ &= \prod_{(n,l) \notin \gamma(k)} ((1 - x_l^n)\beta_n + x_l^n(1 - \beta_n)) \end{aligned}$$

et

$$\begin{aligned}
P_0(\vec{X}=\vec{x}) &= \prod_{(n,l)} P_0(X_l^n = x_l^n) \\
&= \prod_{(n,l)} ((1-x_l^n)\beta_n + x_l^n(1-\beta_n)) \\
&= \left(\prod_{(n,l)\in\gamma(k)} ((1-x_l^n)\beta_n + x_l^n(1-\beta_n)) \right) \left(\prod_{(n,l)\notin\gamma(k)} ((1-x_l^n)\beta_n + x_l^n(1-\beta_n)) \right) \\
&= \left(\prod_{(n,l)\in\gamma(k)} ((1-x_l^n)\beta_n + x_l^n(1-\beta_n)) \right) Q_k(\vec{X}=\vec{x}) \\
&< Q_k(\vec{X}=\vec{x})
\end{aligned}$$

De plus, soit $1 \leq k' \leq 2^M$, si $\exists (n,l) \in \gamma(k'),\ x_l^n = 0$, alors $Q_{k'}(\vec{X}=\vec{x}) = 0$.

Sinon, $\forall (n,l) \in \gamma(k'),\ x_l^n = 1$, on a

$$\begin{aligned}
\frac{Q_{k'}(\vec{X}=\vec{x})}{Q_k(\vec{X}=\vec{x})} &= \frac{\prod_{(n,l)} Q_{k'}(X_l^n = x_l^n)}{\prod_{(n,l)} Q_k(X_l^n = x_l^n)} \\
&= \frac{\prod_{(n,l)\notin\gamma(k')} Q_{k'}(X_l^n = x_l^n)}{\prod_{(n,l)\notin\gamma(k)} Q_k(X_l^n = x_l^n)} \\
&= \frac{\prod_{(n,l)\in\gamma(k')^c\cap\gamma(k)} Q_{k'}(X_l^n = x_l^n) \prod_{(n,l)\in\gamma(k')^c\setminus\gamma(k)} Q_{k'}(X_l^n = x_l^n)}{\prod_{(n,l)\in\gamma(k)^c\cap\gamma(k')} Q_k(X_l^n = x_l^n) \prod_{(n,l)\in\gamma(k)^c\setminus\gamma(k')} Q_k(X_l^n = x_l^n)} \\
&= \frac{\prod_{(n,l)\in\gamma(k')^c\cap\gamma(k)} 1-\beta_n}{\prod_{(n,l)\in\gamma(k)^c\cap\gamma(k')} 1-\beta_n} \\
&= 1
\end{aligned}$$

Donc dans tous les cas $Q_{k'}(\vec{X}=\vec{x}) \leq Q_k(\vec{X}=\vec{x})$.

Finalement, le maximum de vraisemblance choisira l'hypothèse P_0 si l'assertion (ce)

n'est pas vérifiée, et choisira Q_i dans le cas où les tests associés aux cellules qui contiennent θ_i sont tous égaux à 1.

Notre règle de détection revient donc à choisir "fond" dans le cas où le maximum de vraisemblance choisit l'hypothèse P_0, et à choisir "visage" dans tous les autres cas.

5.6 Exploration hiérarchique en profondeur de l'espace des poses

Pour modéliser l'ordre d'utilisation des tests X_k^n, donc notre stratégie globale de détection, nous utilisons le cadre des arbres binaires de décision (Breiman et al. 1984).

Soit $\mathcal{T}$ l'ensemble des arbres binaires T, dont chaque nœud interne $s \in T^\circ$ porte un test $X_{k_s}^{n_s}$ et chaque feuille $t \in \partial T$ porte un label $l_t \in \{0, 1\}$. Nous associons à chaque test X_k^n un coût $C(X_k^n)$, et nous faisons l'hypothèse que ce coût ne dépend que de n (soit $C(X_k^n) = c_n$). Pour chaque feuille $t \in \partial T$, soit C_t le coût des tests présents sur le chemin du sommet de l'arbre jusqu'à t.

Pour $s \in \partial T \cup T^\circ$ un nœud (interne ou feuille), nous noterons R_s l'évènement "le nœud s est atteint".

Soit $T \in \mathcal{T}$ un tel arbre binaire, nous appellerons coût de T la variable aléatoire

$$C(T) = \sum_{t \in \partial T} 1_{R_t} C_t$$

On a l'égalité suivante, que nous ne démontrerons pas

$$E_0(C(T)) = \sum_{t \in \partial T} P_0(R_t) C_t = \sum_{s \in T^\circ} P_0(R_s) C(X_{k_s}^{n_s})$$

Nous noterons $\mathcal{T}^{ce}$ l'ensemble des arbres binaires qui implémentent le test de la propriété (ce). Un arbre binaire T sera dans $\mathcal{T}^{ce}$ si, et seulement si, pour toute feuille $t \in \partial T$:

- Si $l_t = 0$, alors il existe dans le chemin jusqu'à t une famille de tests qui répondent "non" et dont les cellules associées constituent une partition de Θ
- Si $l_t = 1$, alors il existe dans le chemin jusqu'à t une suite de tests qui répondent "1", et qui sont associés à toutes les cellules qui contiennent une des cellules les plus fines

La stratégie de recherche en profondeur consiste à poser à chaque étape le test X_k^n avec n minimum, tel que tous les tests X_m^l avec $\Gamma_k^n \subset \Gamma_l^m$ aient déjà été posés et soient égaux à 1. Le critère (ce) consiste à cesser ce parcours dès que l'on sait qu'il existe ou qu'il n'existe pas une séquence de cellules emboitées dont les tests associés répondent 1. Sur les figures 5.3 et 5.4 sont représentés deux exemples de tels parcours.

L'exploration hiérarchique en profondeur peut aussi se décrire de manière récursive : pour vérifier qu'il existe un partitionnement d'une cellule Γ en cellules dont les tests associés sont nuls, on regarde le test associé à la cellule elle-même, s'il est nul, on s'arrête. Sinon on vérifie *avec la même stratégie* si la première sous-cellule de Γ admet une partition en cellules nulles. Si c'est le cas, on s'assure de la même chose pour la deuxième sous-cellule, sinon on arrête.

Nous noterons $\mathcal{T}^{rp}$ l'ensemble des arbres binaires qui implémentent la recherche en profondeur. Nous pouvons décrire ces arbres dans des cas simples. Par exemple la figure 5.5 représente l'arbre de recherche en profondeur pour $M = 1$, et la figure 5.6 pour $M = 2$.

La forme générique de ces arbres de recherche en profondeur peut être explicitée. Notons T_k^n l'arbre correspondant à la stratégie de recherche en profondeur dans la cellule Γ_k^n :

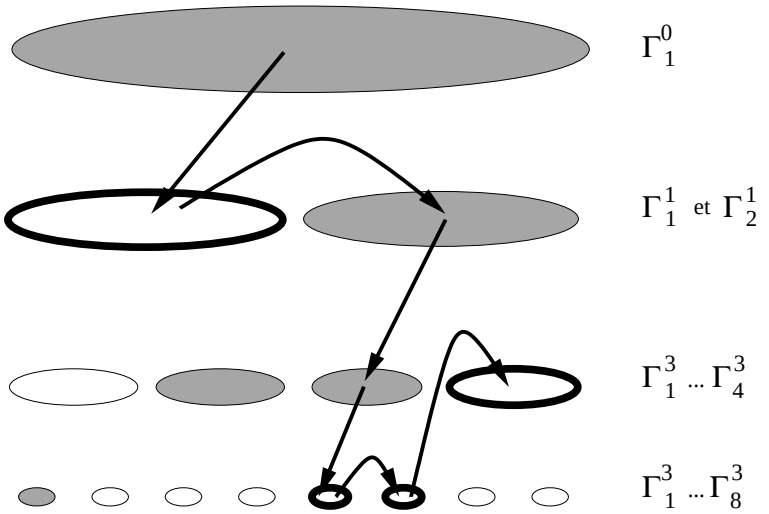


FIG. 5.3: *Les cellules dont le test dédié vaut* 1 *sont représentées en gris. La recherche met ici en évidence l'existence d'une partition de* Θ *en cellules dont les tests valent* 0 *(elles sont représentées avec un trait plus épais), et donc l'inexistence d'une séquence de cellules emboitées dont les tests dédiés valent* 1.

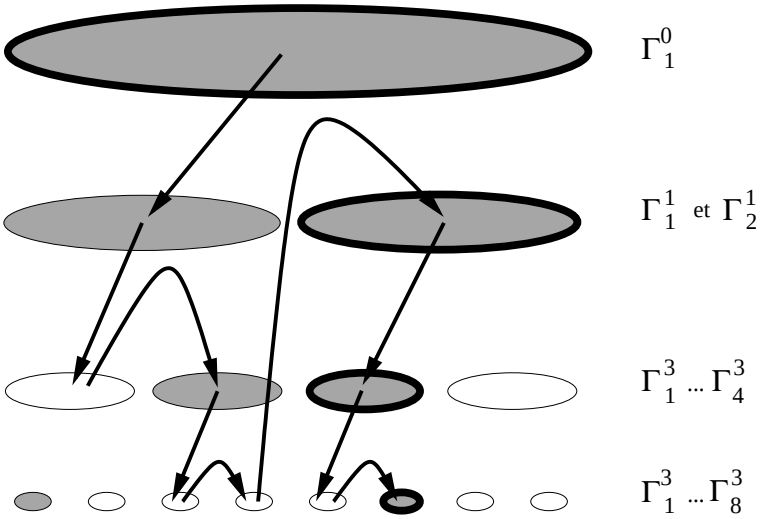


FIG. 5.4: *Les cellules dont le test dédié vaut* 1 *sont représentées en gris. Le parcours met ici en évidence une séquence emboitée de cellules dont les tests associés valent* 1 *(elles sont représentées avec un trait plus épais).*

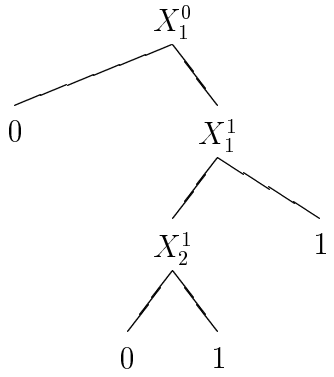


FIG. 5.5: *Arbre de recherche en profondeur pour $M = 1$.*

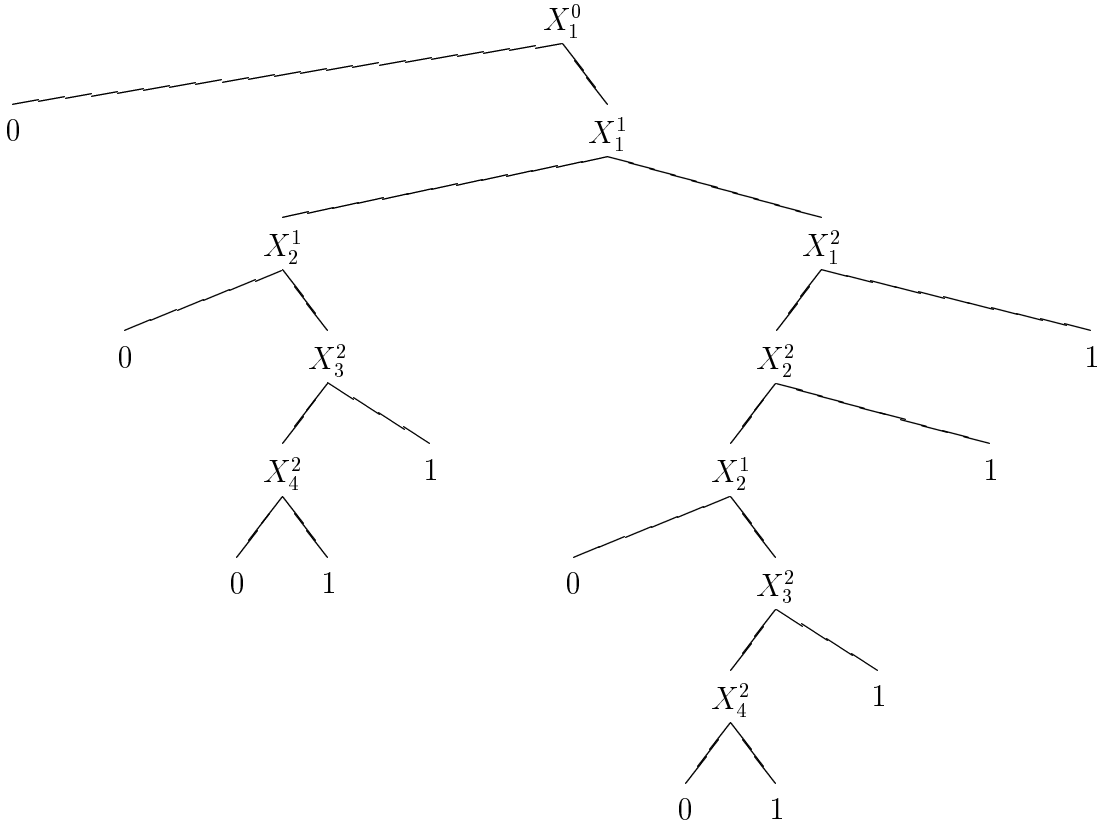
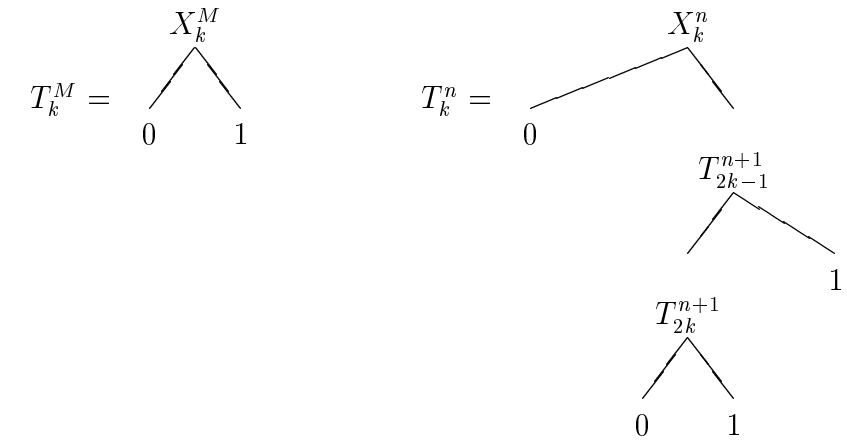


FIG. 5.6: *Arbre de recherche en profondeur pour $M = 2$.*



Dans cette notation, chaque arbre est représenté avec deux fils. Celui de gauche (respectivement celui de droite) représente toutes les feuilles portant le label "0" (respectivement toutes les feuilles portant le label "1"). Si on note $N_{(0)}^n$ le nombre de feuilles portant le label "0" dans l'arbre T_k^n, nous avons :

$$\begin{aligned} N_{(0)}^M &= 1 \\ N_{(0)}^{n-1} &= 1 + (N_{(0)}^n)^2 \end{aligned}$$

On peut remarquer que $N_{(0)}^n$ est exactement égal au nombre de partitions possibles de Γ_k^n en cellules.

De même, si nous notons $N_{(1)}^n$ le nombre de feuilles portant le label "1", on a :

$$\begin{aligned} N_{(1)}^M &= 1 \\ N_{(1)}^{n-1} &= N_{(1)}^n \cdot (1 + N_{(0)}^n) \end{aligned}$$

En notant $\beta_n^* = P_0(T_k^n = 0)$ et c_n^* le coût moyen $E_0(C(T_k^n))$, nous avons de plus :

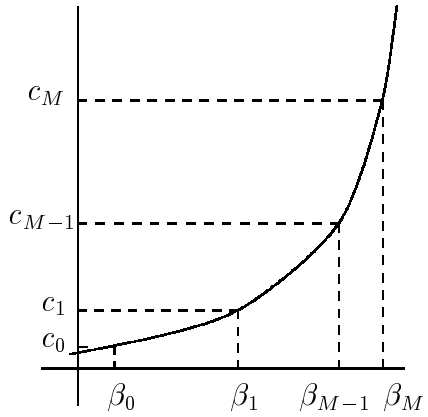
$$
\begin{aligned}
\beta^*_M &= \beta_M \\
\beta^*_n &= P_0(T^n_k = 0) \\
&= P_0(T^n_k = 0 | X^n_k = 0) \cdot P_0(X^n_k = 0) + P_0(T^n_k = 0 | X^n_k = 1) \cdot P_0(X^n_k = 1) \\
&= P_0(X^n_k = 0) + P_0(T^{n+1}_{2k-1} = 0, T^{n+1}_{2k} = 0 | X^n_k = 1)(1 - P_0(X^n_k = 0)) \\
&= P_0(X^n_k = 0) + P_0(T^{n+1}_{2k-1} = 0, T^{n+1}_{2k} = 0)(1 - P_0(X^n_k = 0)) \\
&= P_0(X^n_k = 0) + (1 - P_0(X^n_k = 0)) \cdot P_0(T^{n+1}_{2k-1} = 0) \cdot P_0(T^{n+1}_{2k} = 0) \\
&= \beta_n + (1 - \beta_n) \cdot (\beta^*_{n+1})^2
\end{aligned}
$$

Et :

$$
\begin{aligned}
c_M^* &= c_M \\
c_n^* &= E_0(C(T_k^n)) \\
&= E_0(C(T_k^n)|X_k^n = 0) \cdot P_0(X_k^n = 0) + E_0(C(T_k^n)|X_k^n = 1) \cdot P_0(X_k^n = 1) \\
&= E_0(C(T_k^n)|X_k^n = 0) \cdot P_0(X_k^n = 0) \\
&\quad + E_0(C(T_k^n)|X_k^n = 1, T_{2k-1}^{n+1} = 0) \cdot P_0(X_k^n = 1, T_{2k-1}^{n+1} = 0) \\
&\quad + E_0(C(T_k^n)|X_k^n = 1, T_{2k-1}^{n+1} = 1) \cdot P_0(X_k^n = 1, T_{2k-1}^{n+1} = 1) \\
&= c_n \beta_n \\
&\quad + (c_n + E_0(C(T_{2k-1}^{n+1})|T_{2k-1}^{n+1} = 0) + E_0(C(T_{2k}^{n+1})|T_{2k-1}^{n+1} = 0)) \cdot P_0(X_k^n = 1, T_{2k-1}^{n+1} = 0) \\
&\quad + (c_n + E_0(C(T_{2k-1}^{n+1})|T_{2k-1}^{n+1} = 1)) \cdot P_0(X_k^n = 1) \cdot P_0(T_{2k-1}^{n+1} = 1) \\
&= c_n \beta_n \\
&\quad + c_n P_0(X_k^n = 1) \cdot P_0(T_{2k-1}^{n+1} = 0) + c_n P_0(X_k^n = 1) \cdot P_0(T_{2k-1}^{n+1} = 1) \\
&\quad + E_0(C(T_{2k-1}^{n+1})|T_{2k-1}^{n+1} = 0) \cdot P_0(X_k^n = 1) \cdot P_0(T_{2k-1}^{n+1} = 0) \\
&\quad + E_0(C(T_{2k}^{n+1})|T_{2k-1}^{n+1} = 0) \cdot P_0(X_k^n = 1) \cdot P_0(T_{2k-1}^{n+1} = 0) \\
&\quad + E_0(C(T_{2k-1}^{n+1})|T_{2k-1}^{n+1} = 1) \cdot P_0(X_k^n = 1) \cdot P_0(T_{2k-1}^{n+1} = 1) \\
&= c_n \beta_n \\
&\quad + c_n (1 - \beta_n) \\
&\quad + (E_0(C(T_{2k-1}^{n+1})|T_{2k-1}^{n+1} = 0) + E_0(C(T_{2k-1}^{n+1})|T_{2k-1}^{n+1} = 1) \cdot P_0(T_{2k-1}^{n+1} = 1)) \cdot P_0(X_k^n = 1) \\
&\quad + E_0(C(T_{2k}^{n+1})|T_{2k-1}^{n+1} = 0) \cdot P_0(X_k^n = 1) \cdot P_0(T_{2k-1}^{n+1} = 0) \\
&= c_n \\
&\quad + E_0(C(T_{2k-1}^{n+1})) \cdot P_0(X_k^n = 1) \\
&\quad + E_0(C(T_{2k}^{n+1})) \cdot P_0(X_k^n = 0) \cdot P_0(T_{2k-1}^{n+1} = 0) \\
&= c_n \\
&\quad + P_0(X_k^n = 1) E_0(C(T_{2k-1}^{n+1})) \\
&\quad + E_0(C(T_{2k}^{n+1})) \cdot P_0(X_k^n = 0) \cdot P_0(T_{2k-1}^{n+1} = 0) \\
&= c_n + (1 - \beta_n) \cdot c_{n+1}^* + (1 - \beta_n) \cdot \beta_{n+1}^* \cdot c_{n+1}^* \\
&= c_n + (1 - \beta_n) \cdot (1 + \beta_{n+1}^*) \cdot c_{n+1}^*
\end{aligned}
$$

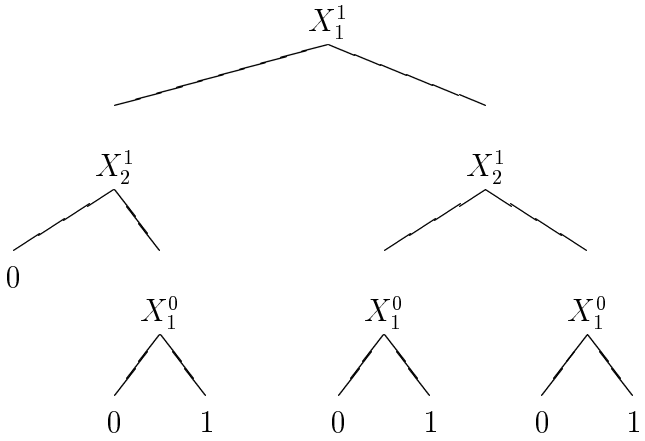
5.7 Stratégie optimale d'exploration des poses

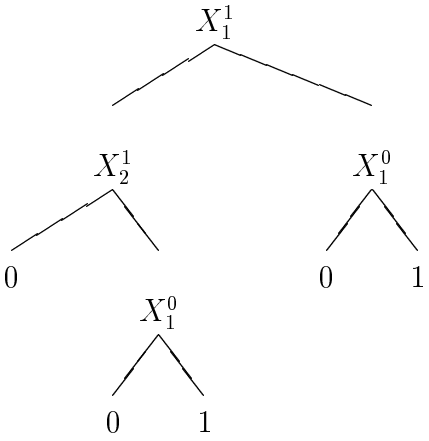
Nous voulons montrer que sous l'hypothèse qu'il existe une fonction f strictement convexe, avec $f(0) = 0$, telle que $\forall n, \quad c_n = f(\beta_n)$, et que l'on a $0 < \beta_0 < \beta_1 < \cdots < \beta_M < 1$, alors la stratégie de recherche en profondeur est optimale. Plus précisément que tous les arbres de $\mathcal{T}^{rp}$ ont le même coût, et qu'il est inférieur ou égal au coût de n'importe quel arbre de $\mathcal{T}^{ce}$.

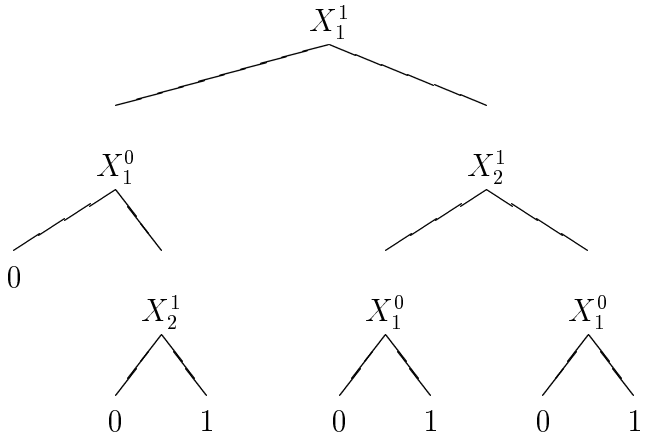


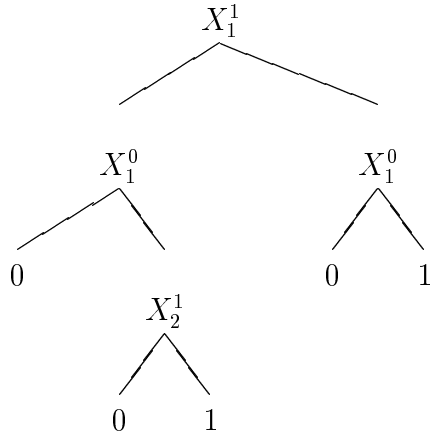
5.7.1 Cas particuliers

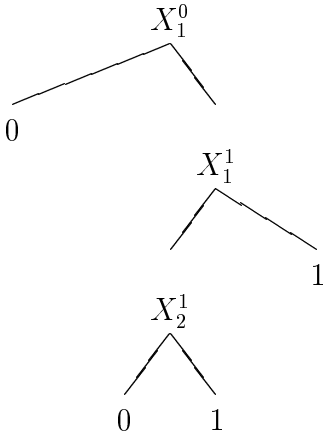
Considérons le cas $M = 1$, modulo les permutations de X_1^1 et X_2^1, et en ne considérant que des arbres qui n'ont aucun sous-arbre dont toutes les feuilles portent le même label (puisqu'ils sont de manière évidente non optimaux), il y a alors cinq "types" d'arbres différents dans $\mathcal{T}^{ce}$. Nous allons montrer que l'arbre optimal est bien celui qui implémente la recherche en profondeur. Ce résultat est un sous cas du théorème énoncé en 5.7.2.

$T_I =$ 

$T_{II} =$ 

$T_{III} =$ 

$T_{IV} =$ 

$T_V =$ 

$$\begin{aligned}
E_0(C(T_I)) &= E_0(C(T_I)|X_1^1=0,X_2^1=0)\cdot P_0(X_1^1=0,X_2^1=0)\\
&\quad +E_0(C(T_I)|(X_1^1=0,X_2^1=0)^c)\cdot P_0((X_1^1=0,X_2^1=0)^c)\\
&= 2c_1\cdot P_0(X_1^1=0,X_2^1=0)+(2c_1+c_0)\cdot(1-P_0(X_1^1=0,X_2^1=0))\\
&= 2c_1\cdot P_0(X_1^1=0)\cdot P_0(X_2^1=0)+(2c_1+c_0)\cdot(1-P_0(X_1^1=0)\cdot P_0(X_2^1=0))\\
&= 2c_1\cdot\beta_1^2+(2c_1+c_0)\cdot(1-\beta_1^2)\\
&= 2c_1+c_0\cdot(1-\beta_1^2)
\end{aligned}$$

De même :

$$\begin{aligned}
E_0(C(T_{II})) &= c_0\cdot(1-\beta_1^2)+c_1\cdot(1+\beta_1)\\
E_0(C(T_{III})) &= c_0+c_1\cdot(2-\beta_1\beta_0)\\
E_0(C(T_{IV})) &= c_0+c_1\cdot(1+\beta_1-\beta_1\beta_0)\\
E_0(C(T_V)) &= c_0+c_1\cdot(1+\beta_1-\beta_0-\beta_1\beta_0)
\end{aligned}$$

On a de manière évidente $E_0(C(T_V)) < E_0(C(T_{III}))$, $E_0(C(T_V)) < E_0(C(T_{IV}))$ et $E_0(C(T_{II})) < E_0(C(T_I))$. Finalement, par la propriété de convexité, on a $\frac{c_0}{\beta_0} \leq \frac{c_1}{\beta_1}$, donc $c_1\cdot\beta_0 - c_0\cdot\beta_1 \geq 0$, ce qui nous donne :

$$\begin{aligned}
E_0(C(T_{II}))-E_0(C(T_V)) &= c_0\cdot(1-\beta_1^2)+c_1\cdot(1+\beta_1)-(c_0+c_1\cdot(1+\beta_1-\beta_0-\beta_1\beta_0))\\
&= -c_0\cdot\beta_1^2+c_1\cdot(\beta_0+\beta_1\beta_0)\\
&= \beta_1(c_1\cdot\beta_0-c_0\cdot\beta_1)+c_1\cdot\beta_0\\
&> 0
\end{aligned}$$

L'arbre optimal est donc bien T_V, qui représente la stratégie de recherche en profondeur.

Nous avons de plus montré empiriquement ce résultat en estimant à l'aide d'un programme le coût de tous les arbres pour $\Theta = \{1, 2, 3, 4\}$, pour un millier de valeurs différentes de β_0, β_1 et β_2.

5.7.2 Cas général

Nous n'avons pas la preuve complète de l'optimalité de la recherche en profondeur, mais nous pouvons montrer le lemme suivant :

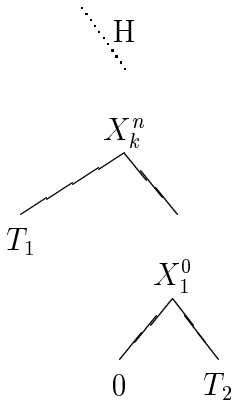
Lemme : Le test placé à la racine d'un arbre optimal de $\mathcal{T}^{ce}$ est X_1^0.

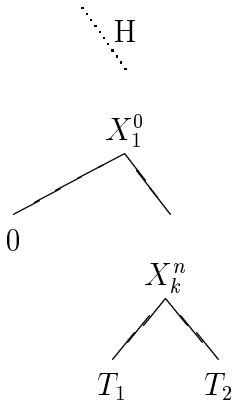
Démonstration :

Montrons que si X_1^0 n'est pas au sommet, l'arbre ne peut pas être optimal. Soit T un tel arbre. Par définition de $\mathcal{T}^{ce}$, le label de la feuille la plus à droite est forcément "1", et toujours par définition de $\mathcal{T}^{ce}$, le test X_1^0 doit être quelque part sur le parcours jusqu'à cette feuille (puisque le critère (ce) doit être respecté).

L'arbre T aura donc finalement la forme représentée ci-dessous. Considérons l'arbre T' obtenu en remontant X_1^0 le long de la branche (T et T' ne diffèrent que dans la partie représentée). Notons H l'ensemble des configurations des X_k^n qui mènent à ce

sous-arbre, et $\overline{H}$ toutes les autres. :

$T =$ 

$T' =$ 

Le sous-arbre T_2 conserve le même chemin jusqu'à lui dans T et dans T', donc toutes ses feuilles vérifient le critère des cellules emboitées. Les feuilles de T_1 quant à elles ont seulement un test de plus dans le chemin jusqu'à elles, et le critère (ce) est donc également vérifié. La feuille "0" enfin est également compatible avec le critère (ce) puisqu'elle est placée après une réponse "0" au test X_1^0. Finalement $T' \in \mathcal{T}^{ce}$.

Les arbres T et T' sont identiques conditionnellement à $\overline{H}$, d'où :

$$\frac{E_0(C(T)) - E_0(C(T'))}{P_0(H)} = E_0(C(T)|H) - E_0(C(T')|H)$$

$$
\begin{aligned}
E_0(C(T)|H) &= E_0(C(T)|H, X_k^n = 0) \cdot P_0(X_k^n = 0|H) + E_0(C(T)|H, X_k^n = 1) \cdot P_0(X_k^n = 1|H) \\
&= E_0(C(T)|H, X_k^n = 0) \cdot P_0(X_k^n = 0) + E_0(C(T)|H, X_k^n = 1) \cdot P_0(X_k^n = 1) \\
&= E_0(C(T)|H, X_k^n = 0) \cdot P_0(X_k^n = 0) \\
&\quad + E_0(C(T)|H, X_k^n = 1, X_1^0 = 0) \cdot P_0(X_1^0 = 0|H, X_k^n = 1) \cdot P_0(X_k^n = 1) \\
&\quad + E_0(C(T)|H, X_k^n = 1, X_1^0 = 1) \cdot P_0(X_1^0 = 1|H, X_k^n = 1) \cdot P_0(X_k^n = 1) \\
&= E_0(C(T)|H, X_k^n = 0) \cdot P_0(X_k^n = 0) \\
&\quad + E_0(C(T)|H, X_k^n = 1, X_1^0 = 0) \cdot P_0(X_1^0 = 0) \cdot P_0(X_k^n = 1) \\
&\quad + E_0(C(T)|H, X_k^n = 1, X_1^0 = 1) \cdot P_0(X_1^0 = 1) \cdot P_0(X_k^n = 1) \\
&= \left(c_n + E_0(C(T_1)|H, X_k^n = 0)\right) \beta_n \\
&\quad + \left(c_n + c_0\right) \beta_0 (1 - \beta_n) \\
&\quad + \left(c_n + c_0 + E_0(C(T_2)|H, X_k^n = 1, X_1^0 = 1)\right) (1 - \beta_0)(1 - \beta_n) \\
&= c_n + (1 - \beta_n) c_0 \\
&\quad + \beta_n E_0(C(T_1)|H, X_k^n = 0) \\
&\quad + (1 - \beta_n)(1 - \beta_0) E_0(C(T_2)|H, X_k^n = 1, X_1^0 = 1)
\end{aligned}
$$

$$
\begin{aligned}
E_0(C(T')|H) &= E_0(C(T')|H, X_1^0 = 0) \cdot P_0(X_1^0 = 0|H) + E_0(C(T')|H, X_1^0 = 1) \cdot P_0(X_1^0 = 1|H) \\
&= E_0(C(T')|H, X_1^0 = 0) \cdot P_0(X_1^0 = 0) + E_0(C(T')|H, X_1^0 = 1) \cdot P_0(X_1^0 = 1) \\
&= E_0(C(T')|H, X_1^0 = 0) \cdot P_0(X_1^0 = 0) \\
&\quad + E_0(C(T')|H, X_1^0 = 1, X_k^n = 0) \cdot P_0(X_k^n = 0|H, X_1^0 = 1) \cdot P_0(X_1^0 = 1) \\
&\quad + E_0(C(T')|H, X_1^0 = 1, X_k^n = 1) \cdot P_0(X_k^n = 1|H, X_1^0 = 1) \cdot P_0(X_1^0 = 1) \\
&= E_0(C(T')|H, X_1^0 = 0) \cdot P_0(X_1^0 = 0) \\
&\quad + E_0(C(T')|H, X_1^0 = 1, X_k^n = 0) \cdot P_0(X_k^n = 0) \cdot P_0(X_1^0 = 1) \\
&\quad + E_0(C(T')|H, X_1^0 = 1, X_k^n = 1) \cdot P_0(X_k^n = 1) \cdot P_0(X_1^0 = 1) \\
&= c_0 \beta_0 \\
&\quad + \left(c_0 + c_n + E_0(C(T_1)|H, X_1^0 = 1, X_k^n = 0)\right) \beta_n (1 - \beta_0) \\
&\quad + \left(c_0 + c_n + E_0(C(T_2)|H, X_1^0 = 1, X_k^n = 1)\right) (1 - \beta_n)(1 - \beta_0) \\
&= c_0 + (1 - \beta_0) c_n \\
&\quad + (1 - \beta_0) \beta_n E_0(C(T_1)|H, X_1^0 = 1, X_k^n = 0) \\
&\quad + (1 - \beta_0)(1 - \beta_n) E_0(C(T_2)|H, X_1^0 = 1, X_k^n = 1)
\end{aligned}
$$

$$
\begin{aligned}
\frac{E_0(C(T)) - E_0(C(T'))}{P_0(H)} &= c_n + (1-\beta_n)c_0 \\
&\quad + \beta_n E_0(C(T_1)|H, X_k^n = 0) \\
&\quad + (1-\beta_n)(1-\beta_0)E_0(C(T_2)|H, X_k^n = 1, X_1^0 = 1) \\
&\quad - c_0 - (1-\beta_0)c_n \\
&\quad - (1-\beta_0)\beta_n E_0(C(T_1)|H, X_1^0 = 1, X_k^n = 0) \\
&\quad - (1-\beta_0)(1-\beta_n)E_0(C(T_2)|H, X_1^0 = 1, X_k^n = 1) \\
&= \beta_0 c_n - \beta_n c_0 \\
&\quad + \beta_n \left(E_0(C(T_1)|H, X_k^n = 0) - (1-\beta_0)E_0(C(T_1)|H, X_1^0 = 1, X_k^n = 0)\right) \\
&= \beta_0 c_n - \beta_n c_0 \\
&\quad + \beta_n(1-\beta_0)E_0(C(T_1)|H, X_1^0 = 1, X_k^n = 0) \\
&\quad + \beta_n\beta_0 E_0(C(T_1)|H, X_1^0 = 0, X_k^n = 0) \\
&\quad - \beta_n(1-\beta_0)E_0(C(T_1)|H, X_1^0 = 1, X_k^n = 0) \\
&= \beta_0 c_n - \beta_n c_0 + \beta_n\beta_0 E_0(C(T_1)|H, X_1^0 = 0, X_k^n = 0)
\end{aligned}
$$

En utilisant $\beta_0 < \beta_n$ et la convexité de f, on montre que cette dernière grandeur est strictement positive, donc que l'arbre T' a un coût strictement plus faible que l'arbre T. Donc, T ne peut pas être optimal.

□

Chapitre 6

Traitement de scènes

6.1 Introduction

Nous avons présenté dans les chapitres précédents la structure d'un détecteur global capable de traiter une image 64×64 pour y détecter efficacement la présence éventuelle d'un visage dont le centre des yeux est contraint dans un carré 8×8. On peut appliquer ce détecteur à une position (x, y) d'une scène en extrayant la sous-image 64×64 dont le coin haut-gauche est à cette position. Le traitement complet d'une scène consiste donc à parcourir toutes les positions $(8.i, 8.j)$, et à appliquer le détecteur à chacune de ces positions.

Dans ce dernier chapitre, nous présentons les résultats de détections de visages sur des scènes réelles. Nous avons testé le détecteur complet, à l'aide de plusieurs scènes issues de l'ensemble "C" d'images collectées à CMU par Henry A. Rowley, Shumeet Baluja, et Takeo Kanade, et d'autres images récupérées sur le WWW.

La section 6.3 décrit comment nous générons, à partir de la base de données dont nous disposons, les ensembles $\mathcal{L}_\Gamma$ utilisés pour construire les f_Γ . Dans la section 6.4 nous présentons des résultats obtenus en appliquant le détecteur sur toute une scène,

et dans la section 6.5 nous montrons comment nous pouvons détecter des visages plus gros que la tolérance du détecteur global en sous-échantillonant la scène. La section 6.6 enfin donne des informations sur la consommation mémoire et la vitesse de l'algorithme finalement obtenu.

6.2 Partitionnement de l'espace de poses

Nous avons vu au chapitre 5 que la construction du détecteur global se fait après avoir sélectionné un partitionnement dichotomique Γ_k^n de Θ.

Pratiquement, nous proposons un partitionnement qui consiste à commencer par contraindre la position du point de référence du visage situé entre les yeux jusqu'à atteindre une précision en position suffisante (2×2), puis à contraindre l'inclinaison et enfin la distance entre les yeux. Nous avons 6 niveaux dans le partitionnement. Les cellules les plus fines, qui correspondent à la précision maximum que nous considérons pour les poses, sont donc au nombre de $2^6 = 64$.

L'intérêt d'un tel partitionnement vient du fait que les détecteurs dédiés à des contraintes différentes en translation sont exactement les mêmes à une translation des tests élémentaires près. Précisément nous ne construisons pas un détecteur pour chacune des cellules

$$\Gamma = \{(x, y, \theta, s) \in \Theta,\ i\Delta x \leq x \leq (i+1)\Delta x,\ j\Delta y \leq y \leq (j+1)\Delta y\}$$

Mais seulement celui dédié à la cellule

$$\Gamma = \{(x, y, \theta, s) \in \Theta,\ 0 \leq x \leq \Delta x,\ 0 \leq y \leq \Delta y\}$$

et nous inférons ceux dédiés aux cellules similaires pour les autres couples (i, j) en

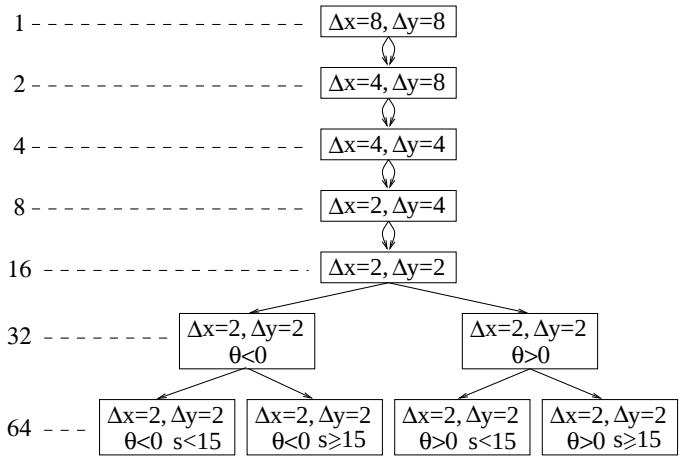


FIG. 6.1: *Liste des détecteurs dédiés effectivement construits. Le nombre de cellules* Γ *de l'espace de poses est indiqué à gauche. Alors que le partitionnement dichotomique contient finalement* 127 *cellules, nous ne construisons en pratique que* 11 *détecteurs dédiés. Nous inférons les détecteurs dédiés aux différentes contraintes en translation pour un même* $(\Delta x, \Delta y)$ *en translatant les positions des tests élémentaires d'un seul d'entre eux.*

translatant les coordonnées des tests élémentaires de celui construit. Alors que le nombre de cellules est $2^7 - 1 = 127$, nous ne construisons que 11 détecteurs dédiés, comme illustré sur la figure 6.1.

La figure 6.2 représente les premières étapes de la détection, qui correspondent donc aux contraintes sur les paramètres (x, y) de la pose.

6.3 Synthèse des ensembles d'apprentissage

Pour chacune de ces 300 images de la base d'apprentissage, nous avons indiqué à la main les positions des yeux et de la bouche, à partir desquelles nous pouvons calculer l'inclinaison et la distance entre les yeux.

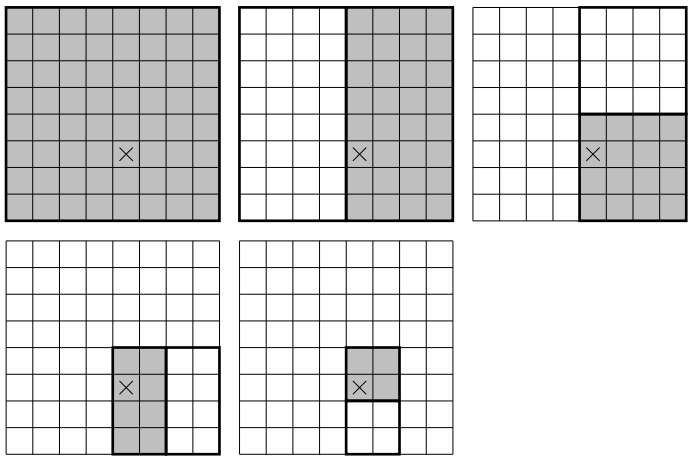


FIG. 6.2: *Premières étapes de la détection finale. Chaque carreau représente un pixel et le symbole* $\times$ *indique la position d'un visage dans la scène, c'est à dire un emplacement qui se trouve entre les deux yeux d'un visage. Les cadres épais indiquent les zones de tolérance des détecteurs, et les rectangles grisés indiquent les détecteurs qui répondent positivement.*

Si nous avions constitué les bases d'apprentissages des différents détecteurs dédiés en extrayant des images de la base complète, nous aurions eu un nombre d'exemples beaucoup trop réduit pour les détecteurs dédiés aux positions les plus contraintes. Pour éviter cette réduction, au lieu d'extraire des sous-ensembles de cet ensemble complet, nous générons une base d'exemples *synthétiques* en appliquant des rotations et des homothéties aux images originales.

Nous créons un ensemble d'apprentissage $\mathcal{L}_\Gamma$ de 1.200 image en générant aléatoirement pour chacune des 300 images originales de notre base de données quatre poses $(x, y, \theta, s) \in \Gamma$. Pour chacune de ces poses, nous générons une image en niveau de gris en appliquant une similitude sur l'image originale de la base de données (deux des images sont de plus inversées horizontalement, cf. fig 6.3).

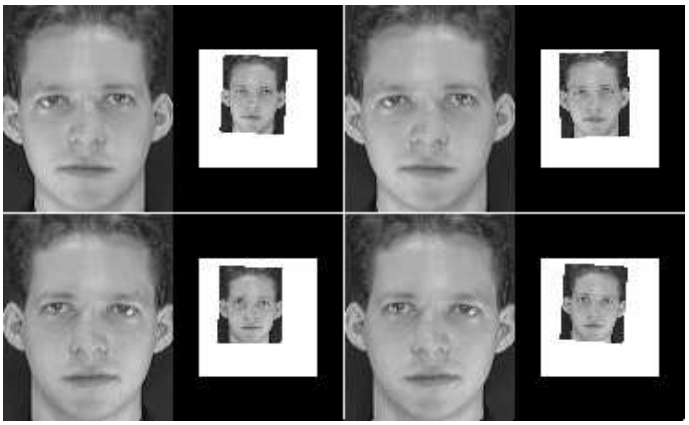


FIG. 6.3: *Les ensembles d'apprentissage sont construits à partir de la base d'apprentissage complète. Chaque exemple original de la base d'apprentissage est répété quatre fois avec quatre poses différentes. On lui applique une similitude pour forcer la distance entre les yeux et son inclinaison de manière à obtenir une base d'apprentissage synthétique contrainte de manière adéquate.*

6.4 Détection à une seule échelle

6.4.1 Exemples

La scène représentée sur la figure 6.4 est représentative du type de résultats que nous obtenons.

6.4.2 Efficacité algorithmique

Comme nous l'avons vu dans les chapitres 3 et 5, aussi bien la structure des détecteurs dédiés que celle du détecteur final sont hiérarchisées de manière à ce que le processus s'arrête dès qu'un critère de rejet est trouvé.

Comme nous l'avons dit en 3.1 et 5.1, grâce à cette architecture, le coût algorithmique à un endroit de l'image dépend fortement de la structure graphique. Finalement,

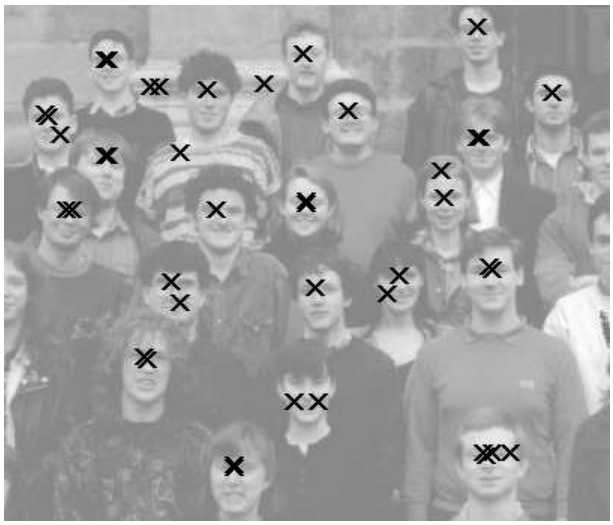


FIG. 6.4: *Exemple de détection de visages sur une scène complète.*

notre algorithme rejette un emplacement candidat en ayant fait *aussi peu de calcul que possible.*

On peut visualiser la réalité de cette remarque en comptant pour chaque pixel de l'image combien de fois le détecteur global a testé la présence d'un bord à cet endroit. En normalisant ce nombre entre 0 et 1 on obtient un niveau de gris qui permet de construire une image. Pour quantifier plus précisément l'usage qui est fait de chaque pixel on peut également représenter le nombre de requêtes faites à un endroit donné de l'image en fonction de son abscisse. Ces deux modes de visualisation sont représentés sur la figure 6.5.

On peut également justifier l'efficacité de la structure dichotomique du détecteur global en étudiant le nombre d'alarmes en fonction de la profondeur du partitionnement dichotomique de Θ. Cette courbe est représentée sur la figure 6.6. Comme on le voit, la décroissance est très rapide, même avec les détecteurs des cellules les plus grossières, dont l'utilisation est donc justifiée. On peut également représenter les alarmes elles-mêmes en fonction de la profondeur du partitionnement, figure 6.7.

Pour s'assurer que la structure hiérarchique du détecteur global réduit bien le calcul

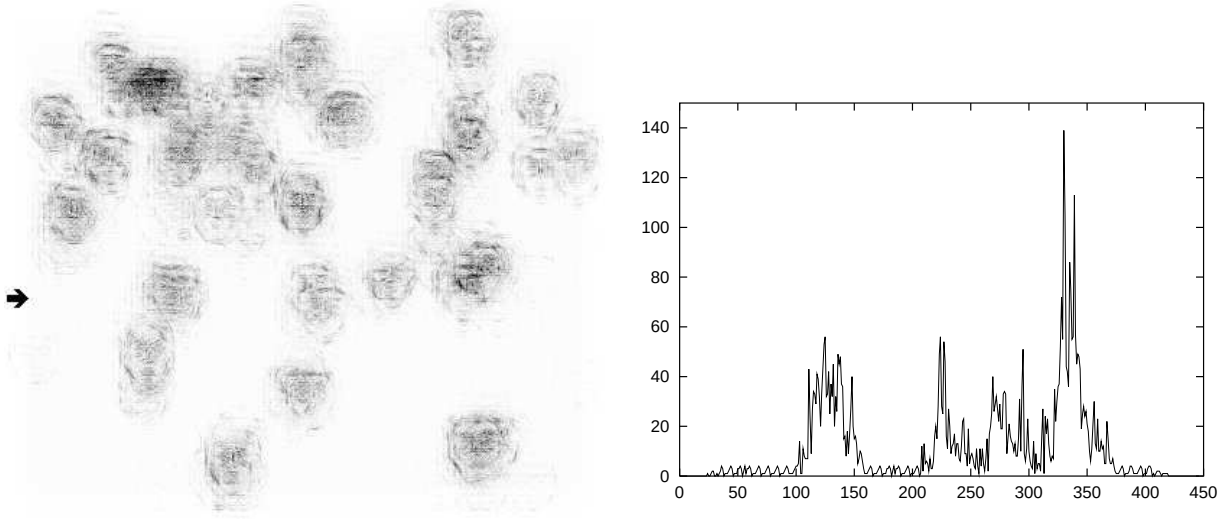
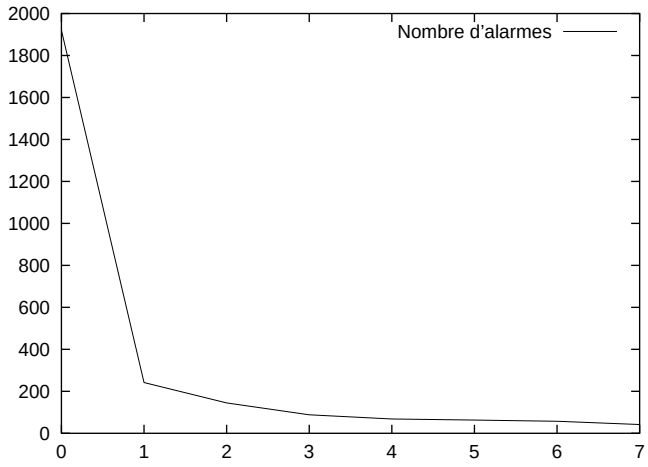


FIG. 6.5: *Intensité d'utilisation des zones de l'image par le détecteur global. Le niveau de gris d'un pixel de l'image de gauche est proportionnel au nombre de fois que le détecteur y a testé la présence d'un bord. Comme on le voit sur cette image, l'essentiel de l'utilisation est concentré sur les zones ambiguës. La flèche indique l'endroit où est faite la coupe qui a permis de tracer la courbe de droite. Cette dernière indique le nombre de fois qu'un emplacement a été utilisé en fonction de son abscisse.*



Profondeur	Nombre d'alarmes
0	1920
1	242
2	145
3	88
4	68
5	63
6	57
7	42

FIG. 6.6: *Proportion de fausses alarmes en fonction de la profondeur maximum de parcours dans les cellules de l'espace de poses* Θ.

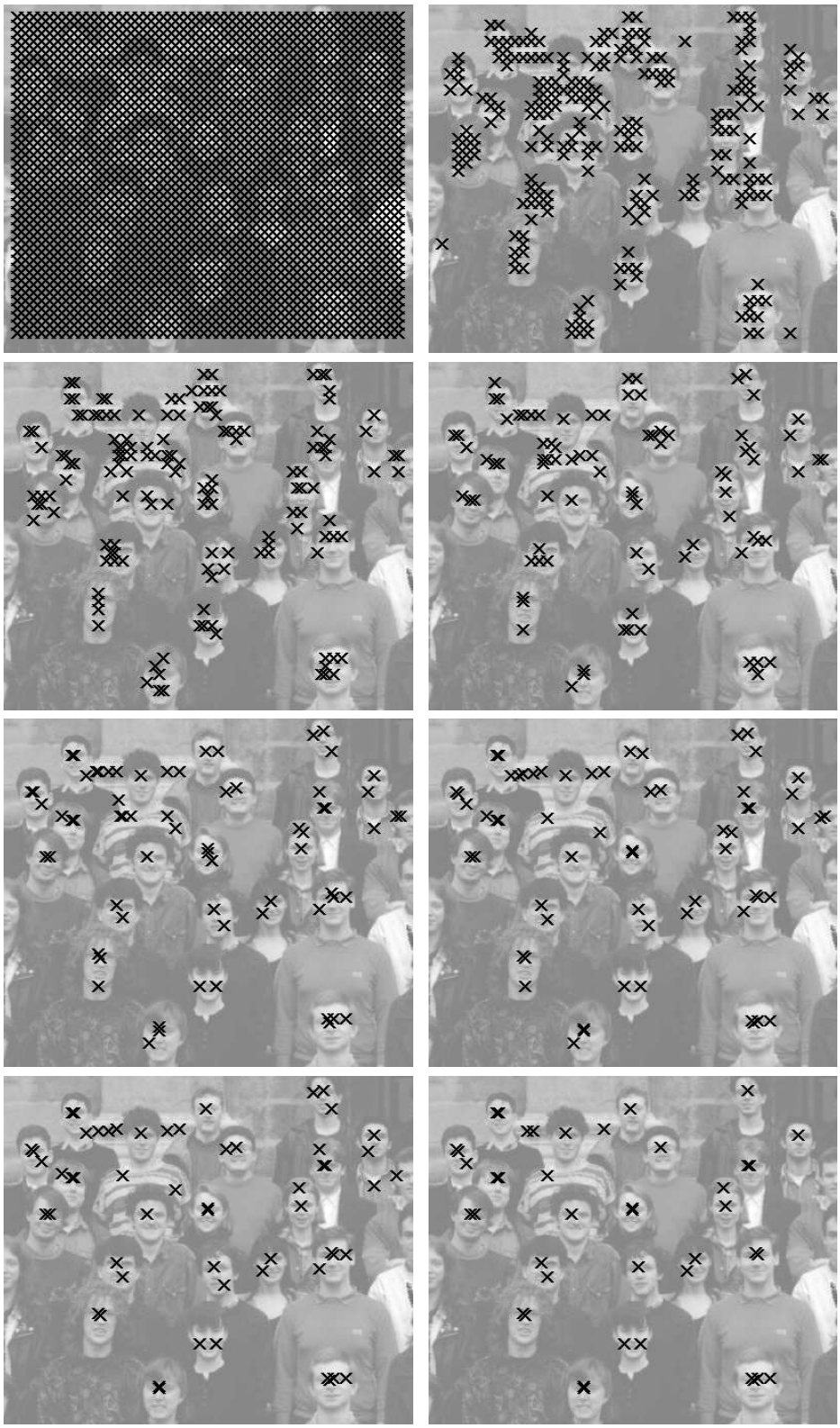


FIG. 6.7: *Illustration de l'exploration en profondeur de l'espace des poses. Chaque image correspond à une profondeur maximum dans le parcours des cellules de l'espace de poses* Θ.

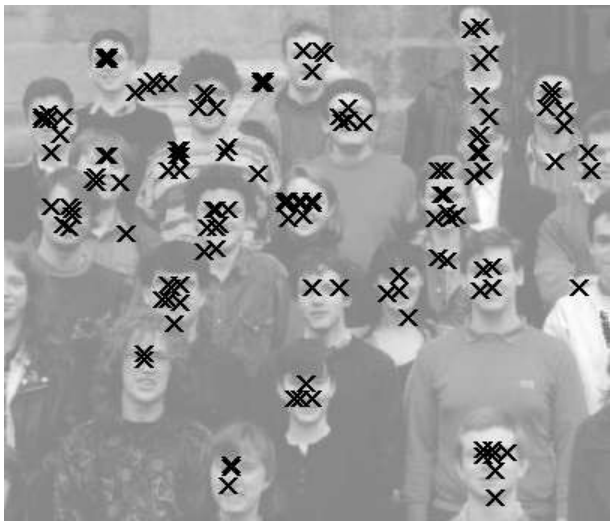


FIG. 6.8: *Alarmes obtenues avec le détecteur f' qui n'utilise que les détecteurs dédiés les plus fins.*

total, on peut tracer les mêmes graphiques pour un détecteur moins efficace, qui utilise directement les détecteurs dédiés les plus fins :

$$f'(I) \quad = \quad \delta_1 \left(\sum_{i=1}^{2^M} X_i^M(I) \right)$$

Comme ce détecteur n'utilise pas l'indépendance entre les X_k^n, le nombre de fausses alarmes est plus élevé (cf. figure 6.8). De plus, ce détecteur ne profite donc pas de la hiérarchisation du calcul. La seule optimisation consiste à interrompre la sommation dès que l'un des termes rencontrés est non nul (C'est à cause de cela que des zones claires apparaissent souvent à droite de zones foncées sur la figure 6.9).

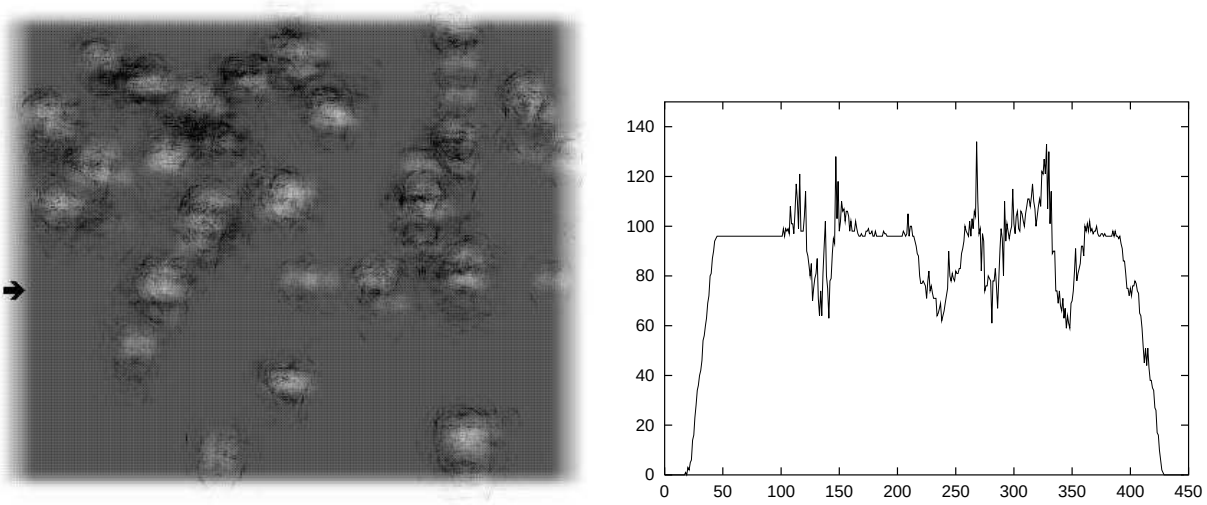


FIG. 6.9: *Cette figure est similaire à la figure 6.5, mais correspond au détecteur f' qui n'utilise que les détecteurs les plus fins au lieu de toute la hiérarchie.*

6.5 Détection multi-échelles

Le détecteur global que nous avons construit admet une tolérance importante pour le paramètre s de la pose (la distance entre les yeux). En pratique cette distance peut être comprise entre 10 et 20 pixels, donc varier d'un facteur 2. Nous ne cherchons pas à détecter des visages de tailles plus petites, considérant qu'il serait alors beaucoup trop difficile de réduire le taux de faux positifs à un chiffre raisonnable.

Pour les visages plus grands, il suffit de réduire l'image de la scène. Etant donnée la tolérance que nous venons de décrire pour le détecteur global, pour chacune des passes nous détectons les visages dont le paramètre s est le suivant :

Réduction		Distance entre les yeux
Image originale	:	$10 \rightarrow 20$
Image réduite d'un facteur 2	:	$20 \rightarrow 40$
Image réduite d'un facteur 4	:	$40 \rightarrow 80$
Image réduite d'un facteur 8	:	$80 \rightarrow 160$

Finalement, le processus de détection va consister à réduire successivement l'image d'un facteur $2, 4, 8$, à utiliser le détecteur sur ces images réduites, puis à ramener les détections à l'échelle de départ en les dilatant. La détection finale sera l'union de ces détections aux différentes échelles.

La figure 6.10 montre des exemples de détections sur des scènes complètes. Le taux de faux négatifs est sur nos exemples de 5.4% et le taux de faux positifs de l'ordre de 0.5 fausses alarmes pour chaque carré 100×100.

$$\begin{array}{ccc} \text{Image originale} & \overset{\text{réduction}}{\longrightarrow} & \text{Image à l'échelle } \frac{1}{2^k} \\ & & \downarrow \\ \text{Détections} & \overset{\text{expansion}}{\longleftarrow} & \text{Détections à l'échelle } \frac{1}{2^k} \end{array}$$

6.6 Utilisation de ressources

6.6.1 Mémoire

Il est intéressant d'estimer précisément la quantité d'information nécessaire pour représenter un détecteur global complet. Expérimentalement, les arrangements ont une complexité inférieure à 8 dans les détecteur dédiés, et il y a 11 de ces détecteurs dans le détecteur global.

Un test élémentaire est représenté complètement par un quadruplet (x, y, i, t) où x et y sont des coordonnées dans une grille 64×64, i est un type de bord parmi 8 et enfin la tolérance est comprise entre 1 et 8. Un tel test occupe donc $6 + 6 + 3 + 3 = 18$ bits de mémoire. La représentation d'un arrangement de complexité n occupe autant de mémoire que les tests élémentaires qui le composent, soit $n \times 18$ bits, et il y a 100 arrangements de chaque complexité. Un détecteur dédié occupe donc $\sum_{i=1}^{8} 100 \times i \times 18 = 1.800 \times \sum_{i=1}^{8} = 64.800$ bits, soit 8.100 octets. Ce calcul ne tient pas compte de la redondance dans les arrangements : certains arrangements sont composés

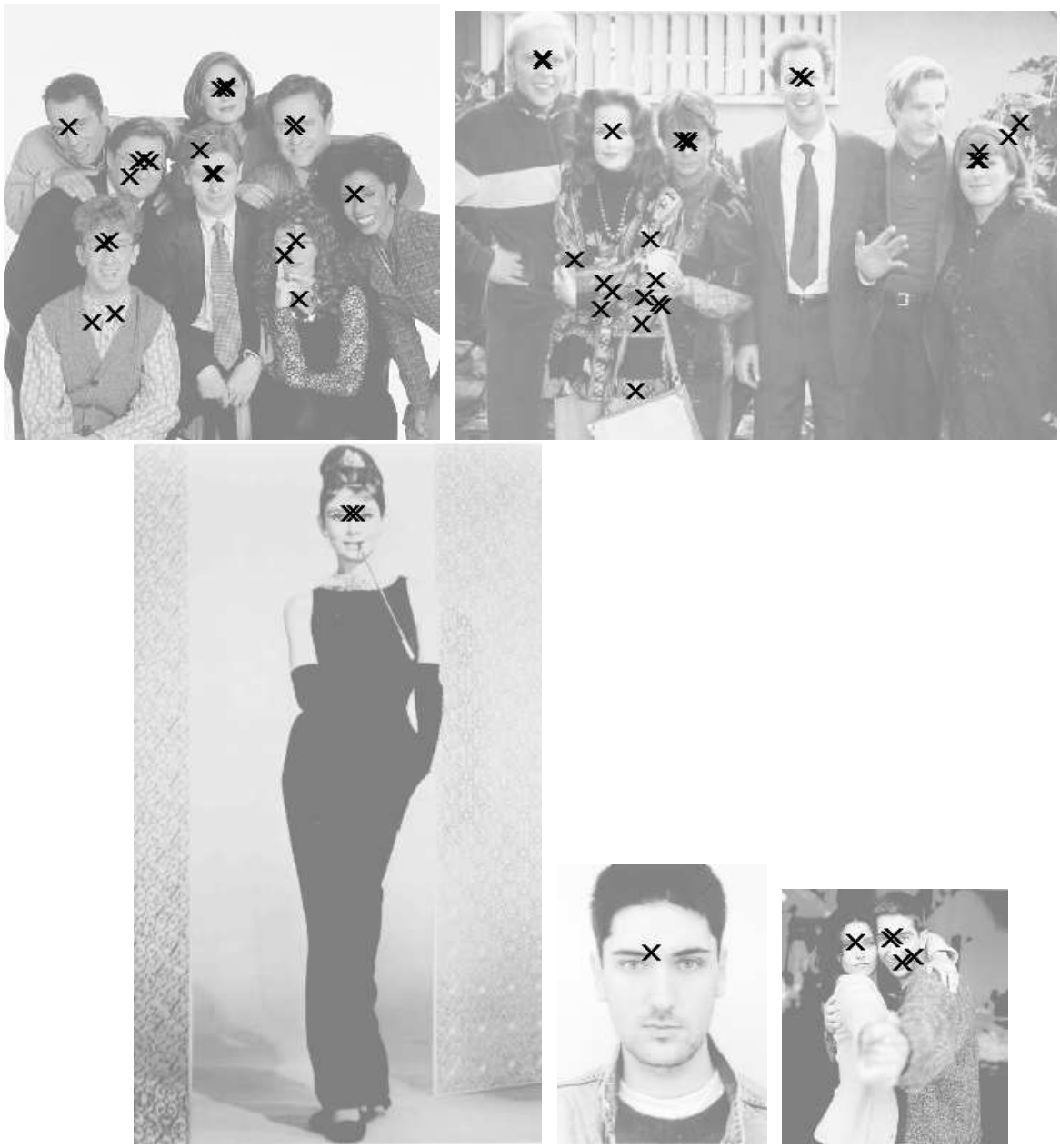


FIG. 6.10: . *Quelques exemples sur des images de scènes.*

de deux autres arrangements déja mémorisés, et utiliser cette propriété permettrait de réduire encore la quantité de données nécessaires. Nous oublions également les seuils qui demandent 8 entiers pour chaque détecteur dédié. Finalement, le détecteur global occupe $11 \times 8.100 = 89.100$ octets, soit moins de 88 kilo-octets.

Cette quantité de mémoire est équivalente à ce qu'occupent 20 eigen-faces de taille 32×32.

Pour des raisons de vitesse, la consommation mémoire lors du fonctionnement du détecteur est de l'ordre de 10Mo.

6.6.2 Vitesse

Pour estimer précisément la vitesse de l'algorithme complet, nous avons écrit un programme qui génère, à partir du détecteur appris, un source "C". De cette manière, nous profitons de l'efficacité du compilateur, et de sa capacité à exploiter au mieux les spécificités de la plate-forme sur laquelle la détection doit se faire.

La vitesse que nous obtenons, pour détecter tous les visages dans une image 450×350 comme celle de la figure 6.4, sur un PC pentium II à 450mhz est de 0.68s, qui se répartit en 0.56s pour effectuer l'extraction de bords et calculer les valeurs des tests élémentaires, et 0.12s pour la détection proprement dite. Cette vitesse a été calculée en effectuant 100 fois la détection complète et en calculant le temps moyen.

L'apprentissage, quant à lui, demande un peu plus d'une heure sur le même PC, et nécessite l'utilisation d'un peu moins de 200Mo de mémoire.

Chapitre 7

Conclusion

Cette thèse a exposé une méthode originale de détection de visages utilisant un nouveau type de classificateurs d'images à deux classes, et une approche cohérente et efficace pour combiner plusieurs de ces classificateurs en un détecteur efficace. Cette architecture complète applique à plusieurs niveaux différents l'idée de représentation hiérarchique de l'information. Nous nous sommes appuyés sur le cadre formel classique de l'apprentissage statistique, mais aussi bien la construction des détecteurs dédiés, que la manière dont ils sont combinés, sont originales.

7.1 Relation avec la biologie

Bien que notre but principal ait été uniquement d'obtenir de bonnes performances en vitesse et en invariance, l'algorithme que nous avons obtenu partage plusieurs propriétés avec les structures cérébrables dédiées à la vision, particulièrement dans le

cortex droit fusiforme. Ces propriétés peuvent être regroupées en quatre points :

- *La forme des arrangements* $\prod_i X_i$ *:* Les arrangements sont des conjonctions de tests élémentaires, chacun fonction de critères locaux. Ces conjonctions jouent ici le même rôle qu'une famille de neurones interconnectés latéralement et qui s'activent de manière cohérente. Chaque neurone possède une activation propre qui dépend de la présence de caractéristiques élémentaires dans le champ de vision, et l'activation de la famille entière correspond donc à la présence d'un arrangement complet.

- *Le critère de corrélation pour l'apprentissage :* Ce critère très primitif, qui ne fait intervenir que la corrélation entre deux arrangements, est très proche de règles d'apprentissage mises en évidence dans le cas de réseaux de neurones naturels. La loi de Hebb est précisément un renforcement de la connection entre deux neurones lorsque leurs activations respectives sont corrélées. Un renforcement des connections latérales par ce mécanisme correspond donc à notre construction itérative des arrangements.

- *L'aspect cumulatif du test élémentaire* $Z_k \geq t(k)$ Dans ce parallèle, les sommes Z_k correspondent donc à une activation globale d'une population de neurones.

- *La hiérarchisation de la représentation :* Des expériences d'imagerie fonctionnelle ont montré qu'il existe des activations de certaines parties du cerveau communes à plusieurs classes d'objets, et d'autres plus spécifiques (Chaos et al. 1999). Un tel comportement peut être comparé à notre hiérarchisation de la représentation pour les poses. Certains arrangements sont présents sur plusieurs sous-groupes de poses, d'autre sont spécifiques d'un seul sous-groupe.

7.2 Points forts

Cet algorithme a été conçu en tenant compte dès le départ de l'efficacité en terme de calcul. Alors que la plupart des techniques introduisent cette optimisation de manière artificielle en combinant deux algorithmes différents (un premier peu coûteux mais peu efficace, et un second coûteux et efficace), nous avons ici proposé un cadre pour gérer ce dilemme coût/efficacité en permanence et de manière explicite.

Nous avons justifié théoriquement et expérimentalement l'intérêt de cette approche, optimale sous certaines hypothèses d'indépendance des détecteurs et de convexité du coût d'un détecteur en fonction de sa performance sur les images à rejeter. Nous avons montré sa rapidité d'exécution sur des images de grande taille (0.65s pour une image 450×350).

Nous avons également abordé explicitement le problème de la sur-adaptation aux données. Là encore, plutôt que de chercher à appliquer des techniques pour améliorer un algorithme existant, nous avons proposé un processus d'apprentissage limitant énormément les possibilités de sur-adaptation. Tous nos résultats ont été obtenus avec une base de données contenant 300 exemples déformés synthétiquement en 1.200 exemples. Ce chiffre est faible comparativement aux tailles des bases utilisées dans d'autres expériences (4.150 exemples dans (Sung & Poggio 1998), 15.000 dans (Rowley 1999) et enfin 50.000 exemples dans (Osuna et al. 1997), sans compter des exemples générés à partir de faux positifs).

Malgré cela nous avons pu atteindre un taux de faux négatifs très faible dans les expériences que nous avons faites. Une expérimentation à grande échelle permettrait de comparer plus finement nos résultats avec ceux obtenus à l'aide d'autres méthodes.

7.3 Points faibles

Le point faible essentiel réside dans l'analyse des capacités d'invariance du détecteur. Il est difficile a priori de savoir jusqu'à quel point il s'agit, ou pas, d'apprentissage brutal de toutes les instances possibles de visages.

L'invariance réside à quatre niveaux dans cet algorithme :

- L'extraction de bords est invariante aux variations des niveaux de gris de l'image ;
- Les tests primitifs sont des disjonctions de présences de bords dans des voisinages, et sont donc invariants à de petites déformations géométriques de l'image ;
- Chaque test de la forme $Z_k \geq t(k)$ admet une tolérance quant aux arrangements présents, et peut donc supporter des occlusions ou des dégradations ;
- Chaque pose de visage est gérée par un des détecteurs dédiés à un ensemble de poses.

On ne peut pas savoir sans une étude très poussée si cette architecture, et en particulier l'accumulation d'invariance dans les niveaux élémentaires de la représentation, permet au cours de l'apprentissage de détecter et d'utiliser les véritables invariants globaux de l'objet, ou s'il s'agit finalement d'une mémorisation brutale de toutes les instances possibles de visages.

Deux arguments s'opposent malgré tout à cette critique : le premier réside dans les tests qui ont la forme d'un comptage du nombre d'arrangements présents. Par définition ces tests sont stables à des variations importantes de l'aspect du visage, et capturent un grand nombre d'instances, tout en dépendant de la présence de structures complexes. Ils cumulent donc de vraies propriétés d'invariance, tout en tenant compte de propriétés très spécifiques des visages.

Le second argument est simplement la quantité d'information nécessaire à la représentation du détecteur appris. Nous avons vu que cette quantité est faible (cf. 6.6.1) : deux ordres de grandeur plus de moins que la place occupée par la base d'apprentissage elle-même. Il n'y a donc pas une quantité d'information suffisante pour réellement stocker un nombre important d'exemples.

7.4 Développements futurs

7.4.1 Tests négatifs

Le détecteur que nous avons présenté dans cette thèse utilise comme critère de classification la présence de structures complexes dans l'image. Ainsi, si une image vérifie les contraintes et si le détecteur y détecte un visage, alors la même image continuera à vérifier ces contraintes si des structures lui sont rajoutées. Une image de texture avec de hautes fréquences, possédant un grand nombre de bords a de grandes chances de vérifier les contraintes et de produire des fausses alarmes.

Pour remédier à cela, nous proposons d'introduire des tests élémentaires qui testent l'absence de bords dans une zone de l'image, plutôt que leur présence. De tels tests élémentaires *négatifs* permettent de construire des détecteurs dédiés qui rejettent les images trop riches en bords.

7.4.2 Généralité de l'utilisation combinée de plusieurs détecteurs dédiés

Dans le chapitre 5 nous avons insisté sur le fait que l'efficacité du détecteur global s'appuyait, du moins dans nos analyses théoriques, sur un nombre réduit d'hypothèses à propos des détecteurs dédiés. Il est donc naturel de se demander ce que donnerait la

même architecture avec un type de détecteurs dédiés classiques, tels que les "support vector machines", les réseaux de neurones, ou les eigen-faces.

On pourra aussi ensuite essayer d'introduire l'idée de représentation hiérarchique dans ces modèles. Par exemple dans le cas des eigen-faces, en mettant en place une série de filtres utilisant les projections successives sur les vecteurs de la base propre.

7.4.3 Amélioration de l'algorithme

Les performances de l'algorithme peuvent être augmentées tout d'abord en jouant sur les paramètres actuels : ρ, nombre d'arrangements conservés durant l'apprentissage pour estimer $\mathcal{A}_{\mathcal{L}}(k,\rho)$, facteur de réduction des $t(k)$, etc.

Nous pensons également à des modifications plus profondes de l'algorithme. Par exemple, en réduisant la similarité des arrangements des différents $\mathcal{A}_{\mathcal{L}}(k,\rho)$, on pourrait augmenter l'indépendance statistique des Z_k. Pour cela, nous proposons de supprimer des $\mathcal{A}_{\mathcal{L}}(k,\rho)$ tous les arrangements qui sont inclus dans un arrangement plus complexe. Précisément, avec les notations introduites dans 4.2, nous introduirions :

$$\overline{\mathcal{A}^*_{\mathcal{L}}}(k,\rho) = \{A \in \mathcal{A}^*_{\mathcal{L}}(k,\rho),\ \forall k' > k,\ \forall B \in \mathcal{A}^*_{\mathcal{L}}(k',\rho),\ A \not\subset B\}$$

Et les $\mathcal{A}_{\mathcal{L}}(k,\rho)$ seraient sous-échantillonnés dans les $\overline{\mathcal{A}^*_{\mathcal{L}}}(k,\rho)$. Ainsi, les différents $Z_k \geq t(k)$ tiendraient compte de la présence de structures différentes, et gagneraient en indépendance.

7.4.4 Application de la ρ-décomposition au langage

Nous avons appliqué le principe de ρ-décomposition aux images. Pourtant, il constitue une méthode générique et peu coûteuse pour détecter des corrélations d'ordres élevés.

L'application au langage permettrait de valider cette généralité. Deux expériences seraient possibles.

Une première application au langage lui même. Partant de tests élémentaires qui dépendent de la présence d'une lettre de l'alphabet à une certaine position dans le mot, peut-on retrouver la structure du langage, ou au moins des mots ? Concrètement, considérons l'ensemble des couples de lettres de l'alphabet munis de leurs probabilités empiriques (estimées avec le texte de cette thèse par exemple). Nous pouvons calculer quelle est la proportion de couples de lettres qui sont ρ-décomposables.

Par exemple, la probabilité d'avoir dans un couple un 'g' comme première lettre est 0.0136, la probabilité d'avoir un 'a' comme seconde lettre est 0.0659, la probabilité qu'un couple soit 'ga' est 0.0070. La corrélation de l'évènement "avoir un 'g' en première position" et "avoir un 'a' en deuxième position" est donc 0.217, et le couple 'ag' est donc ρ-décomposable pour tout ρ inférieur à 0.217.

Plus généralement, on peut mesurer combien de couples sont ρ-décomposables en fonction de ρ :

ρ	Proportion ρ-décomposables
0.00	81.0%
0.05	62.5%
0.10	37.4%
0.15	23.6%
0.20	17.1%
0.25	5.3%

Cette même ρ-décomposition pourrait-elle être appliquée au langage en tant que signal sonore monodimensionnel ? Peut-on retrouver la structure des phonèmes, syllabes, et finalement mots sous la forme d'arrangements ρ-décomposables ?

7.4.5 Reconnaissance et détection multi-classes

Sur de nombreux points la détection et la reconnaissance sont similaires. Un détecteur peut être vu comme un classificateur à deux classes, et un classificateur à n classes peut être réalisé avec une famille de n détecteurs.

Un travail futur consistera à étudier comment réaliser un tel classificateur avec plusieurs détecteurs utilisant la ρ-décomposition. En particulier la question centrale sera de savoir si l'utilisation mémoire et le coût algorithmique augmentent linéairement avec le nombre de classes ou bien s'il est possible de ré-utiliser une partie des arrangements déjà construits pour une classe A afin de réaliser la détection d'une classe B.

Précisément, cette ré-utilisation des arrangements sera possible si un arrangement de bords qui est ρ-décomposable pour la statistique de l'image associée à un objet donné est encore ρ-décomposable pour la statistique des images d'un autre objet.

Cela semble probable pour les petites structures (tous les objets sont composés de bords), et plus discutable pour des structures très spécifiques.

Si la taille de la représentation est une fonction qui croît lentement avec le nombre de classes, par exemple de manière logarithmique, alors il est possible de généraliser efficacement notre approche. Dans ce cas là, il resterait à vérifier que le principe de représentation hiérarchique peut être étendu de la détermination de la pose à la détermination de la classe.

Concrétement, le détecteur pourrait possèder des filtres grossiers multi-classes qui rejettent toutes les images qui ne représentent aucun des objets d'une des classes communes. Ces filtres utiliseraient des arrangements de bords communs à plusieurs types d'objets différents. Progressivement, ces tests deviendraient de plus en plus précis pour finalement ne détecter qu'une seule classe.

Bibliographie

Amit, Y. & Geman, D. (1997), 'Shape quantization and recognition with randomized trees', *Neural Computation* **9**, 1545–1588.

Amit, Y. & Geman, D. (1999), 'A computational model for visual selection', *Neural Computation*. To appear.

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984), *Classification And Regression Trees*, Wadsworth, Statistics/probability series.

Burel, G. & Carel, D. (1994), 'Detection and localization of faces on digital images', *Pattern Recognition Letters* **15**, 963–967.

Burges, C. J. C. (1996), Simplified support vector decision rules, *in* '13th International Conference on Machine Learning', p. 71.

Chaos, L. L., V., H. J. & Alex, M. (1999), 'Attribute-based neural substrates in temporal cortex for perceiving and knowing about objects', *Nature Neuroscience* **2**, 913–919.

Cootes, T. F. & Taylor, C. J. (1996), Locating faces using statistical feature detectors, *in* 'Proceedings, Second International Conference on Automatic Face and Gesture Recognition', IEEE Computer Society Press, pp. 204–209.

Cortes, C. & Vapnik, V. (1995), 'Support vector networks', *Machine Learning* **20**, 1–25.

Geman, D. & Jedynak, B. (1996), 'An active testing model for tracking roads from satellite images', *IEEE Trans. PAMI* **18**, 1–15.

Graf, H. P., Cosatto, E., Gibbon, D. & Michael, K. (1996), Multi-modal system for locating heads and faces, Technical Report TR 96.5.1, AT&T.

Haiyuan, W., Qian, C. & Masahiko, Y. (1999), 'Face detection from color images using a fuzzy pattern matching method', *IEEE Trans. PAMI*.

Huang, J., Gutta, S. & Wechsler, H. (1996), Detection of human faces using decision trees, *in* 'Proceedings, Second International Conference on Automatic Face and Gesture Recognition', IEEE Computer Society Press, pp. 248–252.

Jacquin, A. & Eleftheriadis, A. (1995), Automatic location tracking of faces and facial features in video sequences, *in* 'Proceedings, International Workshop on Automatic Face and Gesture Recognition, Zurich'.

Jedynak, B. & Fleuret, F. (1996), Reconnaissance d'objets 3d à l'aide d'arbres de classification, *in* 'Proc. Image'Com 96', Bordeaux, France.

Jeng, S., Liao, H., Han, C., Chern, M. & Liu, Y. (1996), 'Facial feature detection using geometrical face model - an efficient approach', *Pattern Recognition*.

Lamdan, Y., Schwartz, J. T. & Wolfson, H. J. (1988), Object recognition by affine invariant matching, *in* 'Proc. IEEE Conf. on Computer Vision and Pattern Recognition', pp. 335–344.

Leung, T., Burl, M. & Perona, P. (1995), Finding faces in cluttered scenes using labeled random graph matching, *in* 'Proceedings, 5th Int. Conf. on Comp. Vision', pp. 637–644.

Maurer, T. & von der Malsburg, C. (1996), Tracking and learning graphs and pose on image sequences of faces, *in* 'Proceedings, Second International Conference on Automatic Face and Gesture Recognition', IEEE Computer Society Press, pp. 176–181.

Ming-Hsuan, Y. & Ahuja, N. (1999), Gaussian mixture model for human skin color and its applications in image and video databases, *in* 'Proceedings of the SPIE', pp. 458–466.

Ming, X. & Akatsuka, T. (1998), Multi-module method for detection of a human face from complex backgrounds, *in* 'Proceedings of the SPIE', pp. 793–802.

Osuna, E., Freund, R. & Girosi, F. (1997), Training support vector machines: an application to face detection, *in* 'Proceedings, CVPR', IEEE Computer Society Press, pp. 130–136.

Rojer, A. S. & Schwartz, E. L. (1992), A quotient space hough transform for space variant visual attention, *in* G. A. Carpenter & S. Grossberg, eds, 'Neural Networks for Vision and Image Processing', MIT Press.

Rowley, A. R. (1999), Neural Network-Based Face Detection, PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania.

Rowley, H. A., Baluja, S. & Kanade, T. (1998), 'Neural network-based face detection', *IEEE Trans. PAMI* **20**, 23–38.

Sabert, E. & Tekalp, A. M. (1998), 'Frontal-view face detection and facial feature extraction using color, shape, and symmetry-based cost functions', *IEEE Trans. PAMI* **19**, 669–680.

Sung, K.-K. (1996), Learning and Example Selection for Object and Pattern Detection, PhD thesis, MIT AI Lab.

Sung, K. K. & Poggio, T. (1994), Example-based learning for view-based human face detection, Technical Report A.I Memo 1521, Artificial Intelligence Laboratory, M.I.T.

Sung, K. K. & Poggio, T. (1998), 'Example-based learning for view-based face detection', *IEEE Trans. PAMI* **20**, 39–51.

Ullman, S. (1996), *High-Level Vision*, M.I.T. Press, Cambridge, MA.

Vaillant, R., Monrocq, C. & Le Cun, Y. (1994), Original approach for the localisation of objects in images, *in* 'IEEE Proceedings on Vision, Image, and Signal Processing', p. 141(4).

Wilder, K. (1998), Decision tree algorithms for handwritten digit recognition, PhD thesis, University of Massachusetts, Amherst, Massachusetts.

Yuille, A. L., Cohen, D. S. & Halliman, P. (1992), 'Feature extraction from faces using deformable templates', *Inter. J. Comp. Vision* **8**, 104–109.