Stratégie de perception pour la compréhension de scènes par une approche focalisante, application à la reconnaissance d'objets

Noël Trujillo Morales

N° d'ordre : 1803
EDSPIC : 394

Université Blaise Pascal - Clermont II

Ecole Doctorale

Sciences pour l'Ingénieur de Clermont-Ferrand

Thèse

présentée par

Noël TRUJILLO MORALES

pour obtenir le grade de

Docteur d'Université

Spécialité : Vision pour la robotique

Stratégie de perception pour la compréhension de scènes par une approche focalisante, application à la reconnaissance d'objets

---

Soutenue publiquement le **13 décembre 2007** devant le jury :

Mme.	C. Garbay	Directrice de Recherche au CNRS	Présidente
Mme.	A. Caplier	Maître de Conférences à l'INP Grenoble	Rapporteur
MM.	M. Devy	Directeur de Recherche au LAAS/CNRS	Rapporteur
	F. Chausse	Maître de Conférences à l'Université d'Auvergne	Examinateur
	R. Chapuis	Professeur à l'Université Blaise Pascal	Directeur de thèse
	M. Naranjo	Professeur Emérite à l'Université Blaise Pascal	Examinateur

Résumé

La problématique scientifique abordée concerne la reconnaissance visuelle d'objets s'inscrivant dans une scène observée. Nous proposons une méthodologie qui va de la définition et la construction du modèle de l'objet, jusqu'à la définition de la stratégie pour la reconnaissance ultérieure de celui-ci. Du point de vue de la représentation, cette approche est capable de modéliser aussi bien la structure de l'objet que son apparence ; à partir de caractéristiques multiples. Celles-ci servent d'indices d'attention lors de la phase de reconnaissance. Dans ce cadre, reconnaître l'objet revient à « instancier » ce modèle dans la scène courante. La tâche de reconnaissance correspond à un processus actif de génération/vérification d'hypothèses régi par le principe de focalisation. Ce dernier agissant sur quatre niveaux du « spectre attentionnel » : la sélection des opérateurs pour le traitement bas niveau, la sélection de l'intervalle d'action de ceux-ci, la sélection de la résolution et la sélection de la région d'intérêt dans l'image. Le fait d'agir sur tous ces niveaux, entraîne une diminution de la combinatoire implicite dans une problématique de recherche visuelle. Sous un regard plutôt unifié, le mécanisme de contrôle de l'attention, du type bottom-up $\leftrightarrow$ top-down, reste implicite dans la stratégie globale de reconnaissance. La « focalisation progressive » et la représentation hybride du modèle, permettent de tirer profit des deux types de représentation classiques. D'une part, la structure de l'objet permet de focaliser le processus de reconnaissance à partir d'observations locales, d'autre part, une fois détectée la région probable de l'objet, la décision finale est faite à partir de l'apparence de celui-ci. Dans le cadre proposé, en intégrant des connaissances sur la structure de la scène (paramètres 3D), d'autres tâches comme celles de la localisation et du suivi sont intégrées d'une façon naturelle. La prise en compte de ces paramètres permet d'estimer l'évolution de la zone d'intérêt dans l'image, lorsque l'objet évolue dans le monde 3D. La méthodologie proposée a été testée pour la reconnaissance, la localisation et le suivi de visages et de piétons.

Abstract

This report deals with the scientific problem of objects visual recognition within an observed scene. We propose an approache going from object's model building to the definition of a strategy for its future recognition. From the representation point of view, this methodology can represente the structure of the object as well as its appearance from multiple features. These last ones are used as attentional clues during the recognition stage. In this framework, the recognition of the object consists in instantiate it in the current scene. The recognition task is an active process of hypothesis generation/verification driven by a focusing principle. Focus acts on four levels of the "<attentional spectrum" : low level operators selection, action interval of these operators, resolution selection and image region of interest selection. The result is a reduction of the native combinatorial complexity of any visual seek process. From a rather unified point of view, the attention control mechanism both top-down and bottom-up, is part of the global recognition strategy. The "progressive focus of attention" and the hybrid representation of the objects makes itu possible to benefit from both the classical representations. On one side, the object structure makes it possible to apply the focus of attention on local parts. On the other side, the final decision is made according to the appearance. In the proposed framework, with the addition of information about the structure of the scene (3D parameters), other tasks such as localization and tracking are naturaly integrated. Thes 3D parameters are taken into account to estimate de evolution of the region of interest in the image as the object evolves in the 3D scene. The proposed methodology has been tested for face or pedestrians recognition, localization and tracking.

Remerciements

Je remercie l'ensemble des membres de mon jury, Mme Catherine Garbay qui l'a présidé, Mme Alice Caplier et M. Michel Devy qui ont accepté la tâche prenante de rapporteurs.

Je souhaite surtout remercier Roland Chapuis, mon Directeur de thèse, pour la qualité de son encadrement au quotidien, pour ces conseils et sa patience.

Je voudrais également témoigner ma gratitude à Frédéric CHAUSSE et à Michel Naranjo qui ont co-encadré ma thèse pour leur soutien constant et leurs remarques.

Bien sûr je remercie les personnes du LASMEA avec lesquelles j'ai eu l'occasion de travailler. Elles savent créer une ambiance de travail agréable.

Cherche un refuge dans la sagesse seule,
car s'attacher aux résultats est cause de malheur et de misère ...

Ludwig Van BEETHOVEN

A Estíbaliz,
a mis padres...

Table des matières

Introduction **15**

1 Vision par ordinateur : des approches à unifier ? **21**
1.1 Introduction 21
1.2 Reconnaissance d'objets en vision biologique 23
1.2.1 Aperçu des théories de la reconnaissance d'objets 24
1.2.1.1 Les modèles en primitives 24
1.2.1.2 Les modèles basés sur les exemples 26
1.2.2 Le rôle de l'attention sélective dans le processus de perception . . 27
1.2.3 Bilan 30
1.3 Reconnaissance artificielle d'objets 31
1.3.1 Introduction 31
1.3.2 Approches actuelles pour la reconnaissance d'objets 32
1.3.2.1 Appariement de modèle (template matchers) 32
1.3.2.2 Appariement relationnel (relational matchers) 35
1.3.3 Bilan 39
1.4 Processus attentionnels 41
1.4.1 Introduction 41
1.4.2 Mécanismes attentionnels 43
1.4.3 Bilan 48
1.5 Discussion et objectifs 48

2 Reconnaissance d'objets par vision focalisée **57**
2.1 Introduction 57
2.2 Représentation et structure de l'objet 62
2.2.1 Introduction 62
2.2.2 Composants d'un objet : définition de « *partie* » 63
2.2.3 Cellules 65
2.2.3.1 Définition 65
2.2.3.2 Grille de cellules 67
2.2.3.3 Opérateurs bas niveau 68

2.2.3.4 Grille de cellules multi-résolution . . . 68
2.2.4 Bilan sur la représentation . . . 68
2.3 Apprentissage . . . 69
2.3.1 Apprentissage des paramètres . . . 70
2.3.2 Apprentissage de la statistique de réponse des opérateurs . . . 71
2.3.3 Positionnement des cellules . . . 72
2.3.3.1 Premier cas : base de données annotées . . . 72
2.3.3.2 Deuxième cas : objets centrés et normalisés en taille . . . 72
Transformations d'échelle, rotation et translation . . . 73
Modèle d'apprentissage final . . . 73
2.3.4 Réduction de dimensionnalité . . . 75
2.3.4.1 Élimination des paramètres non descriptifs . . . 75
2.3.4.2 Modèle alternatif après élimination des paramètres . . . 78
2.3.4.3 Élimination des cellules non fonctionnelles : obtention des *parties* Λ . . . 78
2.3.5 Bilan sur l'apprentissage . . . 80
2.4 Stratégie de Reconnaissance . . . 81
Concernant la hiérarchie pour la focalisation . . . 82
Concernant l'adaptabilité . . . 82
2.4.1 Principe de la stratégie utilisée . . . 82
2.4.2 Préparation et initialisation : niveau zéro . . . 84
2.4.3 Génération des hypothèses (sélection de *parties*) . . . 84
2.4.4 Phase de détection des *parties* . . . 86
2.4.4.1 Zone d'intérêt . . . 86
2.4.4.2 Détection . . . 86
2.4.5 Focalisation dans l'espace des caractéristiques : mise à jour du modèle par filtrage de Kalman . . . 89
2.4.6 Phase de décision . . . 90
2.4.6.1 Caractérisation probabiliste . . . 90
$\Pr(d|O)$ (*probabilité a priori*) . . . 91
Exemple numérique d'évolution de $\mathcal{L}$ . . . 96
2.4.6.2 Critère de Reconnaissance . . . 100
2.4.6.3 Décision finale par (SVM) . . . 100
2.4.6.4 Branch and Bound . . . 101
2.4.7 Bilan sur la stratégie de reconnaissance . . . 103
2.5 Localisation et suivi d'objets . . . 104
2.5.1 Modélisation 3D . . . 104
2.5.2 Estimation de la pose 3D de l'objet . . . 106
2.5.3 Suivi 3D . . . 107
2.6 Bilan . . . 107

3 Applications : reconnaissance, localisation et suivi de visages et de piétons 109
3.1 Introduction . . . 109
3.2 Opérateurs bas niveau . . . 110
3.2.1 Orientation du contour à partir d'une carte d'orientations . . . 110
3.2.1.1 Filtre de Gabor 2D . . . 110
3.2.1.2 La carte d'orientations . . . 112
3.2.2 Filtre moyenneur . . . 113
3.3 Reconnaissance et suivi de visages . . . 114
3.3.1 Bases d'images . . . 115
3.3.2 Préparation (set-up) . . . 116
3.3.3 Apprentissage du modèle . . . 117
3.3.3.1 Positionnement des cellules . . . 117
3.3.3.2 Modélisation d'objets avec grandes variations en échelle 119
3.3.3.3 Réduction de dimensionnalité . . . 121
3.3.4 Test de l'algorithme de reconnaissance : reconnaissance pas à pas 122
3.3.4.1 Préparation et initialisation : niveau zéro . . . 123
3.3.4.2 Génération des hypothèses . . . 123
3.3.4.3 Définition de la région d'intérêt et détection . . . 124
3.3.4.4 Focalisation : mise à jour du modèle . . . 125
Focalisation spatiale . . . 125
Focalisation en résolution . . . 126
Focalisation dans l'espace des paramètres . . . 127
3.3.4.5 Décision . . . 128
3.3.4.6 Exemple d'évolution de l'algorithme . . . 130
Déplacements de l'attention lors de la tâche de recherche visuelle . . . 132
3.3.5 Résultats : exemples de reconnaissance de visages . . . 135
3.3.5.1 Modèle complet . . . 135
3.3.5.2 Modèle réduit . . . 138
3.3.6 Coût calculatoire . . . 138
3.3.7 Définition d'un nouvel espace de recherche : suivi de l'objet . . . 141
3.3.7.1 Résultats . . . 142
3.3.8 Bilan . . . 142
3.4 Reconnaissance, localisation et suivi de piétons . . . 145
3.4.1 Préparation avant l'apprentissage . . . 145
3.4.2 Apprentissage dans le repère « imagette » . . . 146
3.4.2.1 Modèle avec l'information fréquentielle exclusivement 146
3.4.2.2 Modèle avec information fréquentielle et niveau de gris 147
3.4.3 Modélisation 3D, cas des piétons . . . 148
3.4.3.1 Le problème . . . 148
3.4.3.2 Cas des piétons . . . 148

Modélisation 148
Évolution dynamique 152
3.4.4 Exemple de reconnaissance, localisation et suivi de piétons 154
3.4.4.1 Test 1 : modèle avec de l'information de contour uniquement 154
3.4.4.2 Test 2 : modèle avec l'information de contour et niveau de gris 155
3.4.4.3 Test 3 : déplacements horizontaux et en profondeur . . 155
3.4.4.4 Test 4 : déplacement en profondeur 158
3.4.5 Bilan 160
3.5 Discussion 162

4 Conclusion et perspectives **163**
4.1 Conclusion générale 163
4.2 Travaux futurs 165
4.2.1 Représentation de l'objet 165
4.2.2 La recherche séquentielle : est-elle toujours pertinente ? 168
4.2.3 Intégration pour la vision active 170
4.2.4 La perception multisensorielle 172

Introduction

HISTORIQUEMENT, c'est dès les années 50, en physique des particules, que les premières images sont traitées ; ceci pour détecter des trajectoires issues du bombardement des atomes les uns contre les autres, afin de scruter les composantes infimes de la matière. Dès les années 60 les chercheurs se sont intéressés à la lecture optique de caractères (OCR). Toutes ces applications sont issues de trois domaines forts : la restauration, l'amélioration et la compression d'images. Dans les années 70 on se concentre sur l'extraction automatique d'informations entre autres : contours, régions, et on a les premières méthodes d'interprétation d'images avec l'apparition des systèmes experts. C'est vers les années 80 que le concept de vision par ordinateur est né avec l'apparition de la première théorie formelle de la vision par ordinateur proposée par David Marr [69]. Il a été un des premiers à définir les bases formelles de la vision par ordinateur en intégrant des résultats issus de la psychologie, de l'intelligence artificielle et de la neurophysiologie. Marr propose un cadre pour le système de vision avec l'hypothèse qu'on peut étudier les principes de la perception visuelle en considérant que l'objectif (réduit) de la vision est de décrire des scènes (appelé aussi *reconstruction de scènes*[1]).

Actuellement, le développement et les applications relatifs à la vision par ordinateur sont basés sur l'extraction d'informations appartenant au monde 3D, à partir d'images 2D. Un besoin de classement conduit à quatre types de tâches principales :

- **Reconstruction** : il s'agit de construire des modèles 3D de l'environnement à partir de la localisation et la position des objets, l'estimation de la couleur des surfaces ou d'autres propriétés.

- **Asservissement visuel, manipulation et mobilité** : ces applications regroupent des tâches comme la navigation et l'évitement d'obstacles par vision. La saisie d'objet illustre parfaitement la tâche de manipulation.

- **Regroupement spatio-temporel et suivi :** le regroupement visuel consiste à associer entre eux des pixels d'une image correspondant aux régions caractéristiques des objets ou des parties de ceux-ci. Dans une séquence d'images, le suivi consiste à faire l'appariement des primitives d'une image à l'autre. Ceci correspond au re-

groupement temporel.

– **Reconnaissance d'objets et d'attitude** : il s'agit de déterminer la classe à laquelle appartient un objet e.g. « *c'est un visage* », ou de reconnaître des instances spécifiques de cette classe comme « *le visage de Jean* ». La reconnaissance des gestes, des expression des visages, sont des exemples de reconnaissance des attitudes.

Les plus grandes évolutions dans chacune des principales tâches de vision exposées ci-dessus ont été obtenues dans des applications industrielles, dont la caractéristique principale est d'être dans un environnement contrôlé (i.e. contrôle de l'éclairage, objet centré et à une distance fixe de la caméra, etc.).

En vision, un problème complexe est typiquement décomposé en sous problèmes moins complexes (avec l'espoir d'un jour remonter au problème initial). Cette approche par briques fonctionnelles a montré une bonne efficacité dans les applications en environnement contrôlé.

Cependant de nouveaux besoins sont apparus (ex. : robotique mobile autonome) et, peu à peu, les applications de la vision par ordinateur ont commencé à sortir des environnements contrôlés pour aller vers des conditions plus naturelles.

Citons les domaines d'application les plus récents de la vision artificielle :

– **Systèmes autonomes et robotique :** l'autonomie des systèmes, particulièrement des robots, est requise pour obtenir des réponses automatiques à certaines situations afin de résoudre une tâche particulière. L'autonomie demande de percevoir et comprendre l'environnement afin de pouvoir interagir avec lui. Un bon exemple est le véhicule intelligent capable de percevoir son environnement extérieur afin de prévoir des situations de danger ou de pouvoir naviguer dans son environnement d'une façon autonome.

– **Inspection industrielle et robotique industrielle :** la pratique a clairement montré qu'en particulier les tâches monotones et non ergonomiques sont susceptibles d'erreur quand elles sont effectuées par des personnes. L'automatisation de ces tâches amène à une augmentation de la qualité du produit du fait de la répétabilité des procédures.

– **Vidéo surveillance :** depuis une dizaine d'années, il y a eu une augmentation significative de la surveillance par caméra de véhicules et d'activité humaine afin de superviser les centres commerciaux, les zones dangereuses ou pour augmenter la sécurité publique. Une telle application a besoin de systèmes robustes pour la re-

connaissance et le suivi d'objets.

– **Indexation de bases de photos et analyse du contenu d'images :** grâce à la grande disponibilité des capteurs numériques, à Internet et aux prix bas des dispositifs de stockage, le flot d'images a augmenté considérablement et le besoin de systèmes pour la récupération automatique d'images est apparu.

Citons aussi d'autres domaines d'application de la vision artificielle comme les divertissements (cinéma, TV), l'analyse d'images aériennes et spatiales et l'imagerie médicale.

Toutes les applications évoquées ci-dessus demandent un système de vision doté de caractéristiques comme la flexibilité, l'adaptation et leur caractère généraliste [23].

À l'heure actuelle, même les tâches de vision bas niveau qui semblaient simples comme l'extraction des contours ou la segmentation, font appel à des mécanismes de haut niveau très complexes semblables à ceux dont la tâche de reconnaissance de formes ou d'analyse de scènes ont besoin. Ces mécanismes se caractérisent par leur capacité d'inférence, d'adaptation, généralisation, etc...

De plus, ces tâches de vision de niveau différent semblent intimement mêlées. Cette remarque peut être illustrée par deux exemples :

1. la reconnaissance d'un objet a recourt par exemple à une méthode de segmentation préalable. Or, pour effectuer « correctement » la segmentation il est préférable d'avoir des connaissances sur l'objet,

2. pour analyser une scène il faut *reconnaître* les éléments qui la composent. Par ailleurs reconnaître ces objets sans ambiguïté nécessite une certaine connaissance de la scène.

Alors, si les tâches de bas et de haut niveau sont à ce point liées, est-il souhaitable de continuer à les résoudre individuellement ?

La réponse à cette question n'est pas du tout évidente. Cependant, l'exploration d'autres voies peut donner certaines pistes sur la façon de traiter le problème. Voici notre proposition sur le sujet.

Nous nous intéressons essentiellement ici à la vision par ordinateur pour la robotique mobile, le contexte y est favorablement utilisé afin de contraindre le processus de perception. Ainsi, la quantité d'information traitée par un processus de niveau supérieur peut être

minorée. L'utilisation du contexte comprend par exemple : la connaissance de la position du robot et celle de la structure de la scène. La connaissance des éléments constituant la scène fait appel à la tâche de reconnaissance d'objets, et c'est sur cette dernière que notre travail de recherche est concentré.

Du fait de la complexité du problème global de l'analyse de scènes visuelles, nous avons choisi la tâche de reconnaissance d'objets comme niveau intermédiaire pour valider notre approche.

Dans une première partie, la problématique scientifique abordée va donc concerner la reconnaissance visuelle d'objets s'inscrivant dans une *scène observée*. Ceci nous oblige à garder, dans la mesure du possible, une vue globale sur le problème de perception. Ainsi, dans la méthodologie proposée, la tâche de reconnaissance correspond à un processus actif de génération/vérification d'hypothèses.

Ici, un objet est représenté tant pourr sa structure locale que pourr son apparence globale, par un modèle statistique à plusieurs dimensions. Ce modèle regroupe des caractéristiques de nature a priori différente. Dans ce cadre, reconnaître l'objet revient à « instancier » ce modèle dans la scène courante. La reconnaissance est supervisée par l'état courant du modèle de l'objet. Cette supervision consiste à choisir dynamiquement un sous-espace de l'espace des paramètres, pour la mise en correspondance de chaque caractéristique. Ce processus (génération d'hypothèse, détection, remise à jour) est réitéré jusqu'à ce qu'un critère de reconnaissance soit atteint.

Le processus de reconnaissance est guidé vers l'objet recherché à partir d'observations locales de celui-ci. Une fois détectée la région où l'objet est potentiellement présent, la décision finale (objet/non objet) est faite à partir de son apparence.

La mise à jour du modèle consécutive à l'observation d'une caractéristique dans la zone d'intérêt où elle est « attendue », a pour effet de réduire d'une manière drastique l'espace de recherche d'autres caractéristiques : nous parlons de la sélection des opérateurs pour le traitement bas niveau, la sélection de l'intervalle de réponse de ceux-ci, la sélection de la résolution d'analyse et de la sélection de la région d'intérêt dans l'image. Le résultat immédiat est une diminution, très importante, de la combinatoire implicite dans un problème de recherche visuelle.

Le fait de considérer l'objet comme s'inscrivant dans une scène, ouvre la possibilité de tirer profit des connaissances a priori sur la structure de celle-ci. Cela permet de donner une « cohérence » sur la position et l'évolution de l'objet dans le monde 3D (donc dans l'image), par rapport à la scène observée. De plus, il est possible de prédire l'évolution des caractéristiques dans le plan image, lorsque l'objet évolue dans le monde 3D. La tâche

de suivi s'intègre ainsi d'une façon naturelle.

Ainsi donc, le cadre proposé permet d'intégrer plusieurs éléments très importants quand il s'agit de percevoir une scène visuelle : le but, l'attention sélective, les connaissances sur la structure de la scène, ainsi que l'inter-relation globale entre ces éléments...

Dans le chapitre 1 quelques aspects généraux de la reconnaissance d'objets dans la vision biologique, ainsi que le rôle primordial du mécanisme d'attention dans le processus de perception visuelle, sont présentés. Ensuite, l'état de l'art aussi bien en méthodes structurelles qu'en méthodes d'appariement relationnel en reconnaissance d'objets est présenté. Du fait du lien existant entre attention visuelle et reconnaissance d'objets, des techniques concernant les mécanismes d'attention dans les systèmes artificiels sont décrites. Enfin, les éléments essentiels pour un système de reconnaissance sont exposés pour en arriver à un aperçu de la méthodologie proposée.

La description complète de la méthodologie proposée fait l'objet du chapitre 2. La représentation de l'objet à reconnaître, la phase d'apprentissage, ainsi que les étapes de reconnaissance et de décision sont détaillées. L'intégration de la localisation et du suivi d'objets, dans la méthodologie proposée, sont décrites à la fin du chapitre.

C'est dans le chapitre 3 que nous présentons l'application de notre méthodologie à la reconnaissance de deux classes d'objets : le « visage » et le « piéton ». La description de la base de données pour les tests et essais, ainsi que l'analyse des résultats pour la reconnaissance, la localisation et le suivi, sont détaillées.

Dans le chapitre 4 sont présentés en conclusion une discussion sur la méthodologie proposée, les résultats obtenus compte tenu des objectifs initiaux, et les propositions et perspectives d'amélioration de l'approche proposée.

Chapitre 1

Vision par ordinateur : des approches à unifier ?

1.1 Introduction

La reconnaissance visuelle d'objets est un sous-problème d'un problème plus général : celui de la perception visuelle. C'est autour de 1960 que les premiers essais de reconnaissance artificielle de formes, à partir d'images, sont faits afin de reconnaître des trajectoires issues de collisions entre particules. Dès les origines, et autour des années 80, avec l'apparition de la première théorie formelle de la vision artificielle proposée par Marr [69], et jusqu'à nos jours, un grand nombre de techniques et méthodologies ont été proposées afin de résoudre le problème. Deux grandes voies de recherche peuvent être identifiées concernant la façon d'étudier le problème de la reconnaissance visuelle d'objets :

- **Les approches structurelles :** les approches de ce type ont besoin d'une représentation explicite des parties qui forment l'objet à reconnaître.

- **Les approches basées sur les exemples :** appelée aussi *reconnaissance basée sur l'apparence*. La représentation utilisée est basée sur l'apparence globale de l'objet sans tenir compte de sa structure et de ses composantes. Cette technique correspond à l'état de l'art pour les tâches de classification du fait du bon degré de généralisation et aussi du taux de reconnaissance acceptable.

Actuellement, parmi les applications courantes on peut trouver le contrôle de qualité et la classification, et des applications plus récentes comme l'indexation des images dans de grandes bases de données (pour la récupération automatique d'images, appelée aussi « image query »), l'analyse des images de satellites, la détection et reconnaissance d'obstacles pour les véhicules intelligents, la navigation autonome pour des robots mobiles et la vidéo surveillance.

Si à l'origine le problème de la reconnaissance d'objets était concentré sur l'analyse d'images, à l'heure actuelle nous pouvons constater que des applications plus récentes comme par exemple la vidéo surveillance ou la navigation autonome des robots mobiles, demandent de traiter des objets 3D qui appartiennent à un monde hautement variable. Ceux-ci obéissent aux lois physiques comme la gravitation et avec des caractéristiques non seulement physiques comme par exemple la forme, la couleur, la texture, la taille, etc., mais aussi de nature différente comme la fonctionnalité, ou le contexte dans lequel l'objet est immergé. Les demandes actuelles reposent sur ces caractéristiques environnementales, ce qui rend le problème très complexe.

Depuis la théorie reconstructioniste de Marr, de nombreux auteurs ont étudié le problème général de vision mais sous un point de vue réductioniste en le divisant en sous problèmes de complexité inférieure donc pouvant être résolus plus facilement. Parmi les exemples nous trouvons entre autres : l'analyse de scènes, la reconnaissance et le suivi d'objets, la détection de contours, etc. Chaque sous problème est étudié comme un module séparé et indépendant. Cependant, afin de satisfaire les contraintes des applications actuelles, à notre avis le problème de la **reconnaissance d'objets** devrait être considéré comme intimement lié et dépendant du problème plus général de perception visuelle, cela afin d'améliorer la performance ou pour optimiser la tâche de reconnaissance. Cette amélioration ou optimisation peut être obtenue en utilisant des connaissances concernant le monde extérieur[1] et ainsi réduire entre autres des ambiguïtés[2] qui peuvent exister entre différents objets ou classes d'objets. La reconnaissance d'un objet, sans prendre en compte la scène dans laquelle il est situé, ne peut aboutir aux performances attendues de nos jours par de tels modules.

Est-il encore pertinent de traiter le problème de la vision, ou les sous-problèmes associés, sous un regard réductionniste ?

Dans ce chapitre, tout d'abord, nous considérons qu'il est important d'étudier quelques aspects généraux de la vision biologique même s'il est hors de nos objectifs d'imiter ou d'essayer de reproduire les systèmes biologiques. Mais cela n'empêche pas de pouvoir s'inspirer de ces systèmes pour construire des systèmes artificiels. C'est avec ce but que, dans la section §1.2, nous faisons un survol de la reconnaissance d'objets dans les systèmes biologiques en présentant ses caractéristiques principales. Deux théories pour la reconnaissance d'objets, les *modèles organisés en primitives* et les *modèles basés sur les*

[1]Ce type de connaissances peut agir, par exemple, pour se focaliser dans l'espace 3D, dans l'espace d'objets, de tâches, etc. [113]

[2]Il peut y avoir des situations dans lesquelles le système de reconnaissance peut se trouver incapable d'identifier correctement un objet dans une scène, dont le « fond » et les objets peuvent interférer avec la représentation de l'objet cible [87].

exemples, sont discutées en montrant les avantages et inconvénients de chacune. Ainsi, on étudie l'attention sélective et son rôle dans le processus de perception visuelle, notamment en ce qui concerne l'optimisation de flots d'informations.

Dans la section §1.3, on présente deux des approches les plus courantes pour la reconnaissance artificielle de formes : *l'appariement de modèle* (template matching) et *l'appariement relationnel* (relational matching) lesquelles correspondent à l'état de l'art en matière de reconnaissance et détection d'objets. Ainsi, une description des approches les plus importantes appartenant à ces catégories est faite.

De même, étant donné leur importance dans le processus de perception biologique et surtout en raison du besoin de plus en plus imminent dans les applications récentes qui requièrent de la reconnaissance d'objets, une présentation des techniques les plus importantes concernant les mécanismes d'attention visuelle est faite dans la section §1.4. Pour terminer le chapitre, en §1.5, nous présentons le bilan correspondant à l'étude effectuée sur les techniques actuelles, la définition d'objectifs ainsi qu'une brève description de l'approche proposée dans cette thèse.

1.2 Reconnaissance d'objets en vision biologique

La reconnaissance d'objets constitue une de *tâches* les plus difficiles et plus complexes de la perception visuelle. Cependant, pour l'homme, la reconnaissance d'objets à différents niveaux catégoriels, semble être une activité facile de tous les instants faite apparemment sans problème et sans aucun effort [104]. Il faut remarquer que la perception implique non seulement des stimuli physiques mais aussi physiologiques et psychologiques, ce qui rend le problème encore plus complexe.

Chez l'homme, la reconnaissance visuelle d'objets, de scènes, de visages est généralement rapide[3], automatique et fiable. Typiquement on peut la scinder en deux tâches : la **catégorisation** au niveau de base (identifier le stimulus visuel comme étant une scène de ville, un éléphant ou un visage) puis éventuellement l'**identification** des exemples (identifier le stimulus visuel comme étant la ville de Paris, l'éléphant du zoo de Vincennes ou le visage d'Einstein). Parmi les caractéristiques psychologiques et neuro-physiologiques importantes on peut citer : la rapidité, l'invariance spatiale (capacité de reconnaître un stimulus visuel alors qu'il a subi des transformations comme une translation, une dilatation et une rotation 2D et 3D), la résistance au bruit, l'inférence ou la généralisation [26]. Cela demande un mécanisme avec des caractéristiques importantes de flexibilité et d'adaptation.

[3]Des résultats montrent un temps variant de 100ms à 150ms

a) b) c)

FIG. 1.1 – a) Image originale. b) Représentation de l'image originale en basses fréquences. c) Représentation de l'image originale en hautes fréquences.

En ce qui concerne la rapidité pour faire la reconnaissance des scènes (au niveau catégoriel), des hypothèses comme celles appelée BHF (basses aux hautes fréquences) postulent que l'information issue des basses et moyennes fréquences spatiales (échelle grossière, voir figure 1.1-b) est disponible plus rapidement que l'information fine (figure 1.1-c). Même si c'est bien le cas dans de nombreuses situations, **des expérimentations montrent la possible existence d'un mécanisme de sélection de l'échelle adéquate en fonction des besoins** [77]. Le compromis entre la rapidité et la fiabilité (temps de disponibilité du stimulus visuel vs degré de fiabilité dans l'identification de la scène) reste un des critères les plus importants pour la sélection d'une telle échelle d'analyse [26].

1.2.1 Aperçu des théories de la reconnaissance d'objets

Deux théories ont fortement influencé la façon d'étudier la reconnaissance d'objets : **les modèles en primitives** et **les modèles fondés sur les exemples**.

1.2.1.1 Les modèles en primitives

L'hypothèse qui a prédominé chez nombre d'auteurs est que la variabilité de la structure spatiale pour une catégorie d'objets donnée objets est plus importante que pour les scènes. Donc, BHF ne semble pas être le traitement adéquat. En revanche, l'extraction plus précise de diverses caractéristiques et propriétés visuelles liées aux contours de l'objet et de ses parties semble être plus pertinente.

C'est à partir de l'existence de détecteurs de segments de droite, dans le cortex visuel chez le chat et le singe, que ce type de modèles a été développé dans les années 60. Mais ce n'est que dans les années 80, avec la première théorie de la vision en intelligence artificielle proposée par Marr (1982) que cette approche est formalisée. Cette théorie postule qu'un modèle de reconnaissance doit contenir des informations sur l'arrangement spatial

des traits, autrement dit, qu'il doit comprendre une description structurale.

Marr définit la reconnaissance visuelle comme un processus de *reconstruction* symbolique composé de trois étages successifs :

- **Bas niveau (extraction de caractéristiques) :** représentation en termes de contours locaux et globaux, coins, etc.

- **Niveau moyen (regroupement, théorie du Gestalt) :** organisation en caractéristiques plus complexes en termes de surfaces,

- **Haut niveau :** formation de modèles 3D des objets.

Dans ce contexte, les niveaux supérieurs ne se concevant pas sans les niveaux inférieurs, la reconnaissance est considérée comme le résultat d'une combinaison hiérarchique de processus et de représentations entre les différents niveaux de représentation. Bien que cette théorie reste très importante car elle est la première théorie formelle de la vision par ordinateur, de gros problèmes empêchent qu'elle puisse être appliquée dans des situations plus réalistes. Le plus important est celui lié à l'extraction des élément de base (contours, coins, etc.). Sans une bonne représentation de l'objet, à partir de ces éléments, il n'est pas possible de reconstruire des surfaces.

Plus tard, en partant de cette théorie, Biederman [12] postule que l'objet soit représenté en mémoire par un arrangement spatial de composantes volumétriques appelés « géons » (géométrique ions) qui sont détectés à partir de contours dans l'image uniquement. Biederman postule que, à partir de 36 géons, on est capable de représenter un grand nombre d'objets. Cette type de représentation, appelée **RpC** (**R**eprésentation **p**ar **C**omposantes), est basée sur le fait que des objets présentés en « images » ou par contours sont également bien détectés et avec exactitude. Un exemple d'un objet représenté par géons est présenté dans la figure 1.2.

Selon Biederman, les différentes phases présumées pour la reconnaissance d'objets sont : l'extraction de contours, la détection des propriétés spécifiques, l'analyse de régions de concavité, la détermination de composantes, l'appariement de composantes selon la représentation de l'objet et l'identification de l'objet. Le schéma décrivant les différentes phases est présenté dans la figure 1.3.

L'utilisation de géons pour représenter un objet a des avantages importants, comme :

- L'invariance au point de vue : les primitives sont similaires sous différents angles.

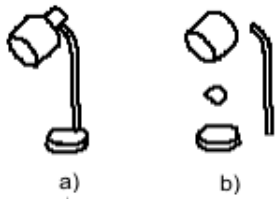


FIG. 1.2 – Exemple de la représentation par composantes structurales. a) Objet perçu. b) Objet représenté par des géons.

– Les propriétés discriminatoires : il est difficile de confondre une représentation avec une autre.

– La résistance au bruit : ils peuvent être identifiés sous des conditions réelles.

– La parcimonie : un nombre réduit de géons peut être suffisant pour créer des représentations complexes.

Même si cette représentation constitue la meilleure proposition pour la reconnaissance d'objets en psychologie cognitive [26], elle a l'important inconvénient qu'actuellement, de même que pour la théorie reconstructionniste de Marr, on ne sait toujours pas comment extraire les caractéristiques de base à partir des stimuli pour construire les géons.

1.2.1.2 Les modèles basés sur les exemples

Ce type de représentation prend comme modèle directement des images perçues des objets (2D ou 3D) et non pas un modèle construit à partir d'une conception abstraite particulière. A la différence des modèles structuraux, les modèles basés sur les exemples postulent que l'ensemble des caractéristiques visuelles, laissées par chaque exemple précédemment rencontré, contribuent à la mémoire de l'objet [26] : l'objet est vu comme un tout et non plus comme un ensemble de composantes. Dans la figure 1.4, on présente l'apparence d'un visage moyen en prenant 200 exemples.

Cette théorie a été postulée en observant des neurones qui étaient activés lors de la présentation de visages [56, 47] et la possibilité de l'existence d'un traitement spécifique pour la reconnaissance de ces derniers. Des études récentes montrent qu'il n'y a pas de preuves neurobiologiques concernant l'existence d'un tel mécanisme spécifique : ces neurones ont été activés lors de la présentation de « non visages ». Le résultat de ces études ouvre la possibilité que la reconnaissance d'objets puisse intégrer une représentation hy-

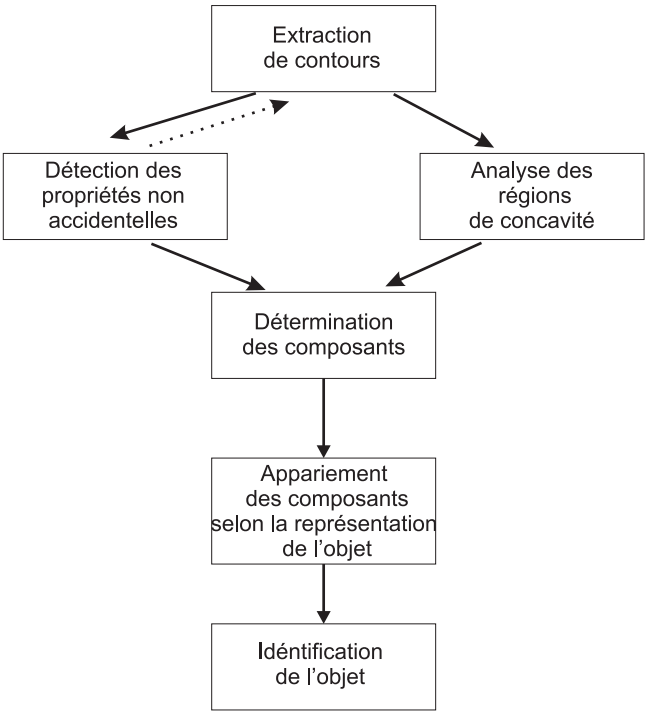


FIG. 1.3 – Schéma proposé par Biederman décrivant les différentes phases présumées pour la reconnaissance d'objets (repris de [12]).

bride, structurale et holistique, où la structure ou l'apparence de l'objet peuvent varier selon le niveau d'abstraction ou catégorisation.

1.2.2 Le rôle de l'attention sélective dans le processus de perception

L'attention sélective chez l'homme a été un sujet d'étude depuis près d'un demi-siècle et joue un rôle assez important dans le processus de perception. Il s'agit d'un mécanisme de « filtrage » pour les données perceptives, lequel permet ou non l'accès aux mécanismes plus complexes offrant une représentation détaillée de l'information [26]. Un exemple type de l'attention sélective est celui appelé du « cocktail » où se trouve un certain nombre de personnes en conversation par petits groupes. Ici, l'attention sélective permet de se concentrer sur une conversation en particulier, d'en suivre le sens et tout en se fermant aux conversations des autres groupes que le filtre attentionnel cherchera à bloquer. Récemment, dans le domaine de la psychologie cognitive pour la vision, des résultats étonnants nous montrent qu'effectivement nous sommes capables de « filtrer » certaines informations et de nous concentrer seulement sur les données qui nous intéressent, en accord avec la tâche courante. Pour plus de détails voir Simons et al. [98, 96, 97, 74]. Ce type de résultats affirme le rôle actif de l'observateur dans le processus perceptif : **notre percep-**

FIG. 1.4 – Exemple de l'apparence moyenne d'un objet.

tion n'est pas déterminée uniquement par la stimulation à laquelle nos récepteurs sensoriels sont exposés [98]. Un mécanisme comme celui-ci provoque une diminution de la quantité d'information, notamment aux données relatives aux propriétés physiques élémentaires des stimuli, qui auront accès à un traitement plus avancé. Ainsi, **l'attention visuelle peut être vue comme l'aptitude d'un système de vision, soit biologique soit artificiel, à détecter rapidement des parties potentiellement relevant d'une scène visuelle (en accord à la tâche courante), sur laquelle des tâches visuelles de haut niveau, telle que la *reconnaissance d'objets*, peuvent se focaliser**.

Tout mécanisme d'attention a besoin d'un mécanisme de contrôle qui définit la capacité du système de vision à naviguer au sein d'un univers d'informations, de modèles, d'outils et de stratégies, en vue de résoudre le problème d'interprétation posé [41]. Des expérimentations dans le domaine des sciences cognitives et en psychologie, montrent principalement l'existence de trois types de contrôle pour le processus d'attention :

- **L'attention guidée par les caractéristiques (bottom-up) :** ici, le mécanisme d'attention visuelle est guidé à partir de stimuli et d'une façon automatique. Les indices (caractéristiques) principaux sont la localisation, la couleur et l'orientation des contours. L'attention sélective guidée par les caractéristiques est *indépendante du but* et de l'objet qu'on veut reconnaître (dans le cas particulier de la reconnaissance d'objets).

- **L'attention guidée par le but (top-down) :** ici, le mécanisme d'attention visuelle va s'orienter vers des cibles qui sont définies en *fonction des besoins*. Un des exemples peut être l'utilisation du contexte pour faciliter la tâche de détection d'un objet quelconque. Ce type d'attention est volontaire et sans effort [20]. Il y a des expérimentations montrant que le système visuel peut exploiter l'information concernant la localisation de la cible, pour améliorer le traitement dans la zone d'intérêt

[87]. En plus, ce type d'attention peut améliorer la reconnaissance d'objets : les ambiguïtés provoquées par d'autres objets ou par le bruit qui perturbe la représentation d'un objet cible, peuvent être réduites [87].

– **Contrôle hybride (bottom-up ↔ top-down) :** c'est peut-être le type de contrôle le plus adéquat qui intègre l'attention guidée par le but et guidée par des indices. Dans un mécanisme de ce type, il existe une interaction entre ces deux types de contrôle. Le fait de savoir à quel moment appliquer un type de contrôle ou l'autre est un problème important.

Deux exemples sont présentés afin d'illustrer les mécanismes de contrôle du processus attentionnel. Dans la figure 1.5 a) nous présentons un exemple de l'attention top-down. Ici, la question qu'on pose à l'observateur est : où est la voiture ? La région marquée en rouge est la zone probable (focalisation dans l'espace géométrique) où la voiture est attendue. En b), exemple (bottom-up), la lettre A colorée en rouge attire l'attention de l'observateur indépendamment du but (il n'y a pas de volonté à voir la lettre colorée en rouge en particulier).

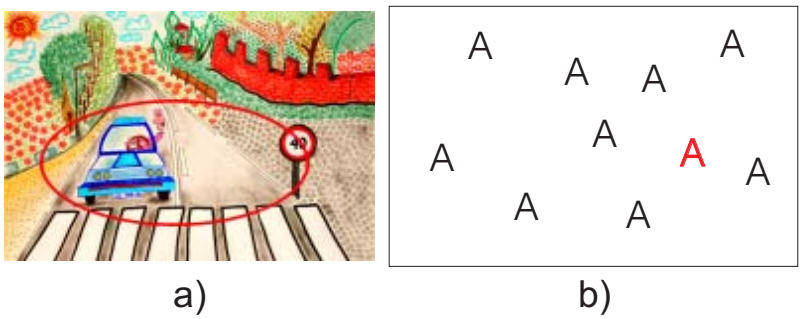


FIG. 1.5 – Exemple d'attention visuelle. a) Top-down. La zone marquée correspond à la région où l'objet « véhicule » est attendu par l'observateur. b) Dans cet exemple, parmi l'ensemble de lettres A, celle qui est colorée attire l'attention de l'observateur indépendamment du but.

Des études permettant d'évaluer l'efficacité relative de différents critères utilisés par l'attention, pour sélectionner un stimulus visuel, montrent que la sélection reposant sur la localisation, la couleur, la taille ou la brillance donne lieu à des performances excellentes ; le critère de sélection permettant la meilleure performance étant la localisation [26]. Cela implique que la sélection attentionnelle d'un stimulus, fondée sur un attribut autre que la localisation, amènera néanmoins l'observateur à traiter l'information relative à l'endroit occupé par la cible. *Il a été montré notamment que la sélection attentionnelle accélère considérablement le traitement de l'information perceptive et que l'attention est un mécanisme où les déplacements à travers l'espace perceptif ne sont pas instantanés mais*

exigent du temps (suggestion d'un processus perceptif séquentiel) [26].

D'autre part, en ce qui concerne les indices de focalisation, des expérimentations suggèrent que l'attention peut être guidée non seulement par des stimuli sur une localisation donnée mais aussi dans une résolution donnée ; ce qui permet d'avoir de l'information à différents niveaux de détail. C'est la résolution de l'attention, **ROA** (resolution of attention) des objets. Un autre élément important dans un mécanisme d'attention, pour la tâche d'une recherche en série, est celui appelé inhibition de retour, **IOR** (inhibition of return). L'**IOR** consiste à empêcher l'observateur de revenir plusieurs fois à une même zone ou cible dans l'image.

D'après la théorie de l'attention sélective, dans le processus de perception on peut trouver deux étapes :

- **L'étape pré-attentive :** elle permet de traiter, d'une façon simultanée, un grand nombre de stimuli, sans effort particulier et avec efficacité.

- **L'étape attentive :** elle permet de traiter un nombre réduit de stimuli à la fois (seulement le stimuli sélectionné par le mécanisme d'attention). C'est dans cet étape que de l'information riche et détaillée peut être obtenue.

L'attention sélective aide à surmonter le problème de limitation de ressources, mais aussi à savoir lequel des stimuli est important pour le système de perception [24].

1.2.3 Bilan

Quelques caractéristiques principales de la reconnaissance d'objets, en vision biologique, ont été présentées ; la rapidité, l'invariance spatiale, la résistance au bruit et la généralisation sont les plus remarquables. Deux théories pour la reconnaissance d'objets ont été principalement décrites : celle basée sur la représentation structurelle et celle basée sur l'apparence de l'objet. Les avantages et les inconvénients pour chacune des représentations sont résumés dans le tableau comparatif présenté dans la figure 1.6.

Les résultats issus de la focalisation dans l'espace de caractéristiques (notamment en ce qui concerne l'optimisation du flot d'information et l'amélioration du traitement local), soit du type bottom-up (guidé par les indices) ou top-down (guidé par le but), du mécanisme attentionnel, justifient l'importance de l'attention sélective dans le processus de reconnaissance d'objets ou, plus général, dans le processus de perception visuelle.

	REPRESENTATION STRUCTURELLE	**REPRESENTATION PAR APPARENCE**
Avantages	- Invariance au point de vue - Propriétés discriminatoires - Résistance au bruit - Parcimonie	- Facile à apprendre et à détecter
Inconvénients	- Composants de l'objet très difficile à définir et détecter	- Moins résistante au bruit - Pas de connaissance sur la structure de l'objet - Moins discriminante - Besoin d'un grand nombre d'exemples

FIG. 1.6 – Tableau comparatif entre les avantages et inconvénients de la représentation structurelle et de celle basée par l'apparence.

1.3 Reconnaissance artificielle d'objets

1.3.1 Introduction

Depuis quarante ans, où des progrès significatifs ont été obtenus concernant la reconnaissance d'objets 3D à partir d'une seule image 2D, il n'existe pas encore de système capable de résoudre, d'une façon satisfaisante, un problème de reconnaissance d'objets de la vie quotidienne. Percevoir des environnements avec des changements constants demande principalement d'un système des caractéristiques d'adaptabilité et de flexibilité. L'avancée a été obtenue fondamentalement avec des applications industrielles qui ont la caractéristique de se trouver dans des environnements contrôlés[4] ou sous un contexte spécifique.

Contrairement à la perception biologique jouissant des caractéristiques les plus remarquables de la reconnaissance d'objets, rapidité, invariance et robustesse au bruit, ce sont précisément celles les plus difficiles à atteindre dans un système artificiel. L'occultation partielle d'un objet, le changement d'illumination, les différents points de vue 3D (la rotation dans le plan fronto-parallèle et en profondeur), la position variable dans l'image, le changement de la taille ou la configuration, sont des situations courantes qui augmentent significativement la complexité de la tâche.

Le problème de la reconnaissance d'objets peut être posé de la façon suivante : *à partir d'une image donnée, localiser et reconnaître un objet qui appartient à une classe déterminée d'objets*. Plusieurs configurations sont à prendre en considération : la reconnaissance

[4]Par « environnements contrôlés » on veut dire par exemple : distance caméra-objet fixe, objet centré, éclairage homogène, etc.

à partir d'une seule vue, de vues multiples, d'objets rigides, non rigides, d'objets centrés et non centrés.

Typiquement, dans n'importe quel système de reconnaissance d'objets, on trouve le traitement bas niveau qui a pour rôle d'extraire l'information caractéristique et pertinente concernant l'objet en question. Parmi les traitements les plus classiques nous pouvons citer les suivants :

- **L'extraction de contours :** l'idée couramment admise est que la représentation basée sur les contours a un rôle important pour la reconnaissance d'objets. Parmi les méthodes les plus utilisées on trouve l'algorithme de Canny-Deriche, Laplace, dérivées seconde ordre, des filtres à différentes orientations (e.g. Gabor), des contours actifs (snakes), etc.

- **Le regroupement ou organisation perceptuelle :** agglomération des composantes d'une image en organisations d'un niveau plus élevé telles que les contours.

- **L'information concernant la texture ou la structure locale d'une scène naturelle :** analyse de la texture locale (e.g. analyse de Gabor, Haralick, textons, etc. [13, 60, 43]).

- **La représentation en vecteurs propres :** e.g. eigen-faces [115].

- **L'analyse multi-résolution :** e.g. transformée d'ondelettes [68].

Plusieurs techniques ont été proposées pour la reconnaissance visuelle d'objets dans les systèmes artificiels, ce domaine de recherche présentant une extension assez large. Sans vouloir faire une revue exhaustive de tous ces travaux, seuls ceux qui nous ont semblé les plus significatifs, tant pour les approches **par appariement de modèle** (appelés aussi « template matchers ») comme pour les approches **par appariement relationnel** (appelées aussi « relational matchers »), sont présentés.

1.3.2 Approches actuelles pour la reconnaissance d'objets

1.3.2.1 Appariement de modèle (template matchers)

Ce type de technique, appelée aussi *basée par l'apparence*, consiste à tester (classifier) dans l'image en question, toutes les fenêtres contenant le modèle des objets à rechercher (un template), pour indiquer si l'objet est présent ou non. Dans une technique comme celle-ci, typiquement l'objet (2D ou 3D) est modélisé par son apparence ; cela veut dire par son image perçue et non pas par un modèle construit à partir d'une conception abs-

traite particulière. Afin d'apprendre son apparence, très souvent on a besoin d'une base d'images avec un nombre assez large d'exemples.

Le test d'une fenêtre dans l'image peut être vu comme une tâche de classification : *objet* ou *non objet*. D'une forme générale, le classifieur est entraîné en utilisant la base de données d'entraînement $(\mathbf{x}_i, y_i)$, où $\mathbf{x}_i$ (souvent appelé vecteur de caractéristiques) correspond aux mesures des propriétés[5] de l'objet, et y_i indique si les mesures correspondent à *objet* ou *non objet*. Du fait que les paramètres calculés pour l'apparence sont obtenus en prenant l'image comme un tout, la plupart des systèmes de reconnaissance basés par l'apparence requièrent des exemples avec des petites variations en éclairage et des objets non occultés [2]. Sur la figure 1.7, on montre les étapes typiques pour la phase d'apprentissage et pour celle de classification.

Les points forts de cette technique sont que les caractéristiques tendent à être détectées avec facilité : il n'est pas nécessaire de définir une représentation ou un modèle pour une classe particulière d'objets, car la classe est implicitement définie par la sélection des images pour l'entraînement. En outre, l'approche statistique fournit un cadre commun pour la reconnaissance et la catégorisation.

Les inconvénients les plus importants sont que la structure de l'objet est représentée implicitement et non explicitement[6], et l'espace de décision tend à être de dimension assez élevée, avec des difficultés de calcul associées [34]. Un autre inconvénient aussi important est le besoin d'un grand nombre d'exemples pour l'apprentissage.

Avec l'objectif de traiter le problème des espaces à hautes dimensions, des techniques comme par exemple l'analyse en composantes principales (PCA[7]), l'analyse discriminante ou « multidimensional scaling », sont utilisées.

Des techniques de reconnaissance d'objets basées sur les classifieurs par réseaux de neurones (NN), les machines à vecteurs supports (SVM) et les histogrammes de classes (class histograms), correspondent à la catégorie d'appariement du modèle.

Après l'apparition des SVM introduits par Vapnik [116, 22], Papageorgiou et al. (1998) [83] présentent un cadre général pour la détection d'objets. L'approche proposée a été testée pour la détection de visages et de piétons, où l'apparence de l'objet est modélisée en utilisant un dictionnaire sur-complet de fonctions de base fondées sur des

[5] De telles propriétés peuvent être : géométriques, photométriques, basées sur descripteurs de fréquence spatiale tels que : la transformée cosinus, les descripteurs de Fourier, les ondelettes ou les « eigenimages ».

[6] Quand l'objet à reconnaître correspond à un objet déformable, ou dont l'apparence est fortement variable, il est souhaitable d'avoir les parties explicites qui composent la structure de l'objet.

[7] Principal Component Analysis en anglais.

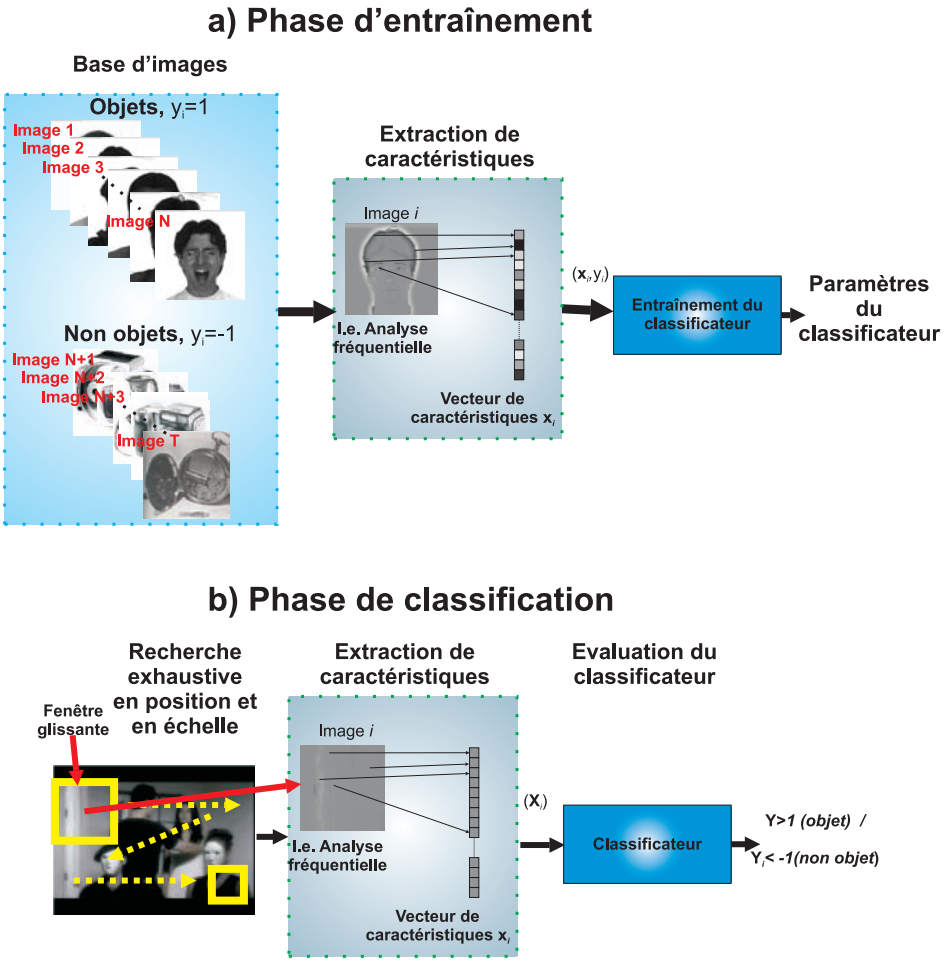


FIG. 1.7 – Schéma blocs pour l'appariement de modèle. a) Phase d'entraînement. b) Phase de classification

ondelettes de Haar. Leur approche permet de détecter des objets de taille différente (grâce au balayage du classifieur dans différentes versions sous échantillonnées de l'image originale) et dans différentes positions spatiales (grâce au balayage exhaustif du classifieur dans l'image). Afin d'accélérer la tâche de classification, une sélection de coefficients d'ondelettes est faite.

Pour leur part, (2001) Viola et Jones [118] proposent un algorithme robuste pour la détection d'objets en temps réel. Les caractéristiques principales de cette approche sont la rapidité des détecteurs et le bon taux de reconnaissance. Ici, même si les primitives utilisées sont calculées très rapidement grâce à la notion d'image intégrale, les auteurs ont intégré un mécanisme de focalisation à base d'un ensemble de classifieurs[8] en cas-

[8]Le classifieur utilisé dans cette approche (l'Ada-Boost) est basé sur un classifieur fort $h(x)$, et un en-

cade ; des classifieurs de basse résolution sont balayés pour identifier des régions qui ressemblent à l'objet et, dans cette région, des classifieurs plus spécifiques sont appliqués afin de reconnaître l'objet.

Shneiderman et Kanade [93] proposent en 2004 une méthodologie probabiliste pour la détection de voitures et de visages à différents points de vue et à différentes échelles. Dans cette approche chaque objet est représenté par un ensemble de parties, où chaque partie est définie comme un ensemble de coefficients d'ondelettes (position, fréquence et orientation) et elles présentent une dépendance statistique. Afin de détecter l'objet dans l'image, cette dernière est balayée à différentes échelles, orientations et points de vue, par une fenêtre de taille fixe. Plus tard, en partant de l'idée de Viola et Jones [117], (2004) Schneiderman [93] présente une approche alternative utilisant une cascade de sous-classifieurs pour accélérer la détection d'objets. A la différence de [117], Shneiderman réutilise les évaluations des caractéristiques entre les fenêtres superposées. Après le balayage du classifieur à la première étape, les auteurs rapportent que presque 99% du nombre initial de candidats sont éliminés. Si un classifieur échoue, il n'est pas nécessaire d'évaluer les classifieurs restants. Donc, comme résultat de cette élimination des régions candidates non potentielles, une grande accéleration de la tâche de reconnaissance est obtenue.

Plus récemment, inspirés par les systèmes biologiques, (2004) Serre et al. [94] proposent un cadre pour la reconnaissance visuelle d'objets basée sur un classifieur linéaire SVM. Des caractéristiques appelées « C2 features » sont utilisées pour représenter les objets à reconnaître. De telles caractéristiques sont obtenues en appliquant un ensemble de filtres de Gabor à différents orientations et échelles. La valeur maximale pour chaque bande, en échelle et en position, est conservée.

Par ailleurs, Torralba et al. [109] (en 2004) introduisent une nouvelle approche pour la détection d'objets de classes multiples. Les caractéristiques principales de cette approche sont de deux sortes : l'utilisation *partagée* des caractéristiques des objets, appartenant aux différentes classes, pour la tâche de classification, et l'entraînement des classifieurs effectué d'une façon conjointe et non pas indépendante. Les *caractéristiques partagées* sont obtenues en utilisant le GentleBoost (une extension de l'algorithme de *renforcement* (boosting) développé par Schapire [91]). La détection de l'objet est faite en balayant l'image, en position et en échelle, par une fenêtre glissante.

1.3.2.2 Appariement relationnel (relational matchers)

Comme son nom l'indique, cette approche décrit les objets en termes de relations entre « templates ». Typiquement on cherche des « patches » (normalement basés sur leur

semble de classifieurs faibles. A chaque caractéristique du vecteur de caractéristiques correspond un classifieur faible.

apparence caractéristique, appelés aussi points d'intérêt ou caractéristiques invariantes) puis on raisonne sur les relations existantes entre eux. Un « patch » peut être par exemple un détecteur de coin ou un détecteur d'objet simple (e.g. oeil figure 1.8).

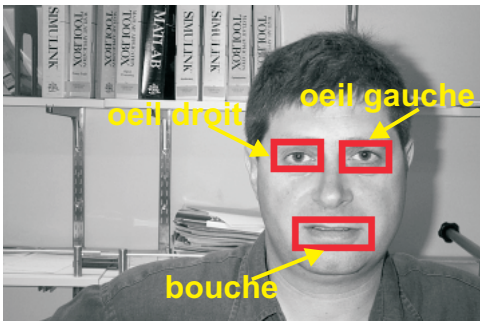


FIG. 1.8 – Exemple de « patches » pour la modélisation de visage. Chaque « patch » correspond à une région de l'image d'apparence caractéristique (dans cet exemple les patches correspondent aux yeux et à la bouche).

L'intérêt est d'avoir une représentation des *parties* de l'objet (comme pour les approches structurelles) qui présente les avantages suivants :

- **Propriétés d'invariance dues aux changements de point de vue** (à condition que ses parties puissent être identifiées) : cela permet de reconnaître des objets 3D à différentes angles de rotation (dans le plan image et en profondeur) avec un nombre réduit de prise de vue.

- **Bon support pour catégorisation :** la possibilité de représenter des nouveaux objets à partir des mêmes primitives utilisées.

- **Parcimonie conceptuelle :** un nombre relativement réduit de primitives peut permettre de représenter un grand nombre de classes d'une façon concise.

D'autres avantages tels que la réduction de candidats possibles pour l'appariement et la robustesse aux occultations, font de ce type d'approches une approche intéressante quand il s'agit de reconnaître des objets variables en apparence ou des objets déformables. Pour la première, même si les détecteurs des parties ne sont pas trop discriminants, on peut compenser cette faiblesse en ajoutant des contraintes concernant la localisation spatiale des parties. Ainsi, les détecteurs doivent répondre, dans des régions spécifiques dans l'image, selon la configuration spatiale de l'arrangement des parties de l'objet. Cette contrainte aide à réduire le nombre de candidats possibles pour faire l'appariement avec le modèle. En ce qui concerne la deuxième, même s'il y a des parties que ne sont pas dé-

tectées, l'algorithme d'appariement peut prendre en considération le manque de détection des parties[9].

Un des principaux inconvénients est le problème lié au choix de l'interprétation structurale correcte parmi plusieurs possibilités. Dans ce type d'approches, typiquement l'utilisateur définit lui même les « parties » qui composent l'objet. Donc, la représentation choisie par l'utilisateur peut ne pas être la plus pertinente pour discerner une classe d'objets déterminée parmi d'autres classes.

Le travail de Perona et al. (1995) [61] correspond à cette catégorie. Dans ce travail, les auteurs proposent une solution pour le problème de détection et localisation de visages (en vues quasi frontales), en utilisant l'*appariement aléatoire de graphes*[10] (random graph matching). La méthode proposée présente deux caractéristiques importantes : 1) elle permet de prendre en considération des parties manquantes d'une façon explicite, et 2) le nombre des éléments, pour l'appariement du modèle, est réduit significativement grâce à l'introduction de contraintes[11] sur la position relative des parties. Ici, un ensemble de détecteurs[12] est appliqué à l'image de test afin d'identifier des régions candidates aux caractéristiques faciales telles que les yeux, le nez, etc., où chaque partie est détectée par l'appariement du vecteur des réponses des filtres directionnels avec un modèle ou prototype. Une fois identifiées toutes les « parties », des groupes de parties, appelés « constellations », sont formés. La constellation qui présente le meilleur score d'appariement est gardée comme le meilleur candidat à être l'objet. Afin de délimiter le nombre de constellations, une *recherche contrôlée* est réalisée : deux parties candidates sont sélectionnées à partir de l'image et, ensuite, la localisation d'autres parties et la nouvelle matrice de covariance sont estimées. Les constellations seront donc formées à partir des candidats situés dans les ellipses définies par la matrice de covariance estimée. Ce processus est répété avec les deux paires de caractéristiques candidates. Toutes les constellations sont testées[13] afin de voir si elles correspondent ou non à l'objet. Des invariances aux translations, aux rotations dans le plan image et aux changements d'échelle, sont obtenues avec la méthode proposée.

[9] On peut définir une heuristique comme par exemple : l'objet est reconnu si 80% des parties recherchées sont détectées.

[10] Le problème consiste à trouver, parmi l'ensemble de constellations formées, la meilleure constellation (celle avec une configuration la plus similaire au modèle).

[11] L'algorithme exploite le fait que les parties sont reliées entre elles (les parties ne peuvent pas apparaître en configurations arbitraires). Connaissant les positions de plusieurs caractéristiques, il peut estimer les positions des autres caractéristiques et leur matrice de covariance associée.

[12] Les détecteurs sont basés sur des réponses de filtres de dérivées Gaussiennes à multiples orientations et échelles.

[13] Le test consiste à faire une compétition entre la constellation gagnante et le reste de constellations, par rapport au test de similitude.

A son tour, (1998) Burl et al. présentent une approche probabiliste pour la reconnaissance d'objets, où chaque objet est modélisé par un ensemble de parties qui sont arrangées en une configuration déformable [15]. Chaque partie est définie à partir d'une variété de caractéristiques visuelles (photométrie locale) comme la luminance, l'orientation, la texture, la couleur, le mouvement ou la symétrie. Le modèle de l'objet est constitué d'un ensemble de N parties reliées entre elles, dont chacune apparaît à une localisation spatiale particulière, ce qui permet de modéliser la géométrie globale de l'objet.

En 2001, Mohan et al. [73] présentent une approche basée-exemple pour la reconnaissance d'objets. Dans cette approche l'objet est modélisé en utilisant l'apparence locale (basée sur des ondelettes d'Haar) et la structure globale (relation spatiale entre parties). L'algorithme est testé pour la détection de piétons où chaque piéton est décomposé en 4 parties (tête, jambes, bras gauche/droite). Chaque partie est détectée en utilisant un classifieur SVM, et la classification finale piéton/non piéton est obtenue en utilisant une combinaison des classifieurs des parties avec un classifieur du type SVM. Afin de réduire la zone d'intérêt pour la recherche des parties, un algorithme de corrélation entre parties, et des contraintes (la moyenne et la variance) pour la localisation et l'échelle sont utilisées.

Dans [10], Belongie et al. présentent une nouvelle approche pour l'appariement de modèle et de reconnaissance d'objets en utilisant le « contexte de forme ». Le « contexte de forme » est utilisé pour décrire la distribution grossière du reste de la forme en relation avec un point donné dans la forme de l'objet (pour un point p_i appartenant à la forme de l'objet, il faut calculer un histogramme grossier h_i des coordonnées relatives de coordonnées de n-1 points restantes). Un algorithme d'appariement de graphes « bipartite » est utilisé. Des invariances pour la translation et la rotation sont modélisées (transformations affines) ce qui permet une certaine robustesse aux petites distortions géométriques, occultations, etc. La méthode a été testée pour la reconnaissance de nombres, la reconnaissance d'objets 3D (COIL-20 database) et la reconnaissance de silhouettes (MPEG-7 shape silhouette database).

Dans le cadre de l'apprentissage non supervisé, inspiré par les travaux de Burl et al. [15], Fergus et al. [38] présentent une méthodologie pour l'apprentissage non supervisé et la reconnaissance d'objets invariante aux changements d'échelle. Cette approche est basée sur un modèle probabiliste et l'objet est représenté par des constellations aléatoires de parties[14]. Chaque partie a une apparence, une position et une échelle relative, et elle peut être partiellement occultée ou non. Ici, à la différence des travaux de Burl et al. [15], la variabilité en apparence est aussi modélisée et pas seulement l'aspect géométrique. La forme de l'objet est modélisée à partir de l'information mutuelle qui existe parmi les

[14]Pour cette approche, chaque partie est détectée avec le détecteur de Kadir et Brady (détection de régions de saillance en position et en échelle). Seulement les P régions plus saillantes sont considérées comme des parties.

positions relatives des parties (c'est grâce à cela qu'une focalisation dans l'espace géométrique peut être atteinte afin d'éviter une recherche exhaustive dans l'image). Pendant la reconnaissance, il faut d'abord détecter les régions les plus saillantes et ses échelles correspondantes, et ensuite évaluer ces régions avec une approche Bayésienne en utilisant les paramètres estimés du modèle dans l'étape d'apprentissage[15]. Pour la détection de visages, cet algorithme présente un taux de reconnaissance de 96.4%.

Pour sa part, (2003) Felzenszwalb [37] présente une extension de [36] en donnant un cadre pour la modélisation et la reconnaissance d'objets, avec l'application de reconnaissance de visages et des objets articulés (i.e. un corps humain). Ce travail est motivé par la représentation de « structure pictorial[16] » introduite par Fischler et Elschlager. A la différence de [36], où chaque partie était représentée en utilisant un modèle d'apparence simple[17], ici l'apparence de chaque partie est modélisée à partir d'une représentation iconique qui est basée sur des réponses de filtres de dérivées Gaussiennes à ordres, échelles et orientations différents. Les paramètres pour modéliser l'apparence, les paramètres de connexion et l'information concernant les parties qui sont connectées, sont estimés en utilisant l'algorithme du maximum de vraisemblance.

Plus récemment et inspirés aussi par le travail de Burl et al., (2004) Fei-Fei et al. [35] présentent une approche pour le « incremental learning » afin de réduire le nombre d'exemples qui doivent être présentés pour apprendre une classe d'objets donnée. L'apprentissage est complété par la contribution des nouvelles images qui s'ajoutent aux images présentées précédemment. Même s'il y a une réduction significative du nombre d'exemples (apprentissage plus rapide), le nombre réduit d'exemples est reflété dans le bas taux de reconnaissance.

1.3.3 Bilan

Deux approches courantes pour la reconnaissance d'objets ont été présentées : les approches par appariement de modèle et les approches par appariement relationnel. Même si ces approches ont donné des résultats très favorables en matière de reconnaissance d'objets, notamment en ce qui concerne la détection de visages, piétons et voitures, il existe plusieurs points sur lesquels elles peuvent être critiquées.

Les approches par appariement de modèle ont fait l'objet d'une attention particulière

[15] Les paramètres du modèle (apparence, échelle et position) sont estimés en utilisant l'algorithme EM (expectation maximisation).

[16] L'idée principale est de représenter un objet par une collection de parties arrangées en une configuration déformable, dont certaines paires sont connectées.

[17] Pour les exemples donnés dans [36], les parties des objets sont des rectangles avec rapport d'aspect fixe, couleur moyenne et variance correspondante.

pour plusieurs raisons :

- le développement et la mise au point de techniques de classification comme les réseaux de neurones et les SVM ayant un bon degré de généralisation,

- le développement d'outils de traitement d'images pour l'extraction de caractéristiques et représentation de l'image : théorie d'ondelettes [68], analyse de structure locale [13], « eigenimages » [115], filtres orientés [59], etc.,

- et naturellement, pour les applications en temps réel, l'évolution significative des ordinateurs (notamment par rapport à la vitesse du traitement des données).

Quand on analyse des techniques correspondant à cette catégorie, on trouve certains points en commun comme par exemple le balayage exhaustif de l'image, en position et en échelle, pour détecter l'objet [83, 93, 94, 109], le besoin de grandes bases d'exemples d'*objets* et *non-objets*, des objets centrés dans la base d'apprentissage et avec de petites variations en point de vue et le fait que typiquement le classifieur répond plusieurs fois autour de l'objet. Pour réduire le temps de calcul dû au balayage exhaustif de l'image, on utilise généralement les techniques suivantes :

- L'utilisation de classifieurs en cascade [117, 93],

- L'introduction des indices de focalisation comme la couleur [100], les ombres pour la détection de véhicules [21], ou la délimitation d'une zone d'intérêt dans l'image définie par l'utilisateur [83],

- La réduction du nombre de paramètres dans le vecteur de caractéristiques [83] (ce qui provoque une augmentation du nombre de fausses détections due au classifieur résultant peu discriminant[18]),

- La réduction du nombre d'échelles ou le pas de résolution spatiale.[19]

D'autre part, en ce qui concerne les approches d'appariement relationnel, nous pouvons remarquer les points suivants :

- C'est une technique adéquate pour reconnaître des objets déformables ou d'apparence variable,

[18] Afin d'améliorer le performance de la tâche de détection, l'intégration du contexte ou la diminution de l'espace de recherche peut être utile.

[19] Quand on diminue le nombre d'échelles, il y a des difficultés pour détecter des objets variables en taille.

– il n'est pas difficile d'apprendre l'apparence des « patches », en revanche, ils doivent être définis par l'utilisateur, comme c'est le cas dans [61, 15, 73, 37, 36],

– il existe toujours le problème associé à la combinatoire.[20]

1.4 Processus attentionnels

1.4.1 Introduction

Quand on cherche à optimiser le processus de perception on trouve des notions comme la focalisation (FOA, focus of attention), l'attention visuelle ou attention sélective, dont l'objectif principal est de délimiter la quantité d'information et le champ de recherche. La focalisation peut être atteinte sur plusieurs niveaux comme par exemple la focalisation dans l'espace 3D, la focalisation dans le champ visuel (exploration de l'image), mais aussi on peut trouver la focalisation dans l'espace des modèles, des tâches, etc. On verra plus tard que cette notion de focalisation est généralisée sous le cadre de la *recherche visuelle* proposé par Tsotsos [113].

Après le travail de D. Marr (1982), on commence à trouver des idées de focalisation avec le travail proposé par Aloimonos et al. (1987) en introduisant le concept de *vision active* [1]. Cette approche diffère de celle proposée par Marr en postulant que l'interaction de l'observateur dans son milieu est une riche source d'information supplémentaire dans le processus de la vision ; fait que Marr ne prend pas en compte en partant des images statiques pour faire la reconstruction des scènes et en négligeant le capteur dans son modèle. Donc, **pour Aloimonos, l'activité perceptuelle est une activité exploratrice et par conséquent active**. D'après cette hypothèse, Aloimonos propose de traiter le problème de la vision en utilisant les capteurs passifs employés en forme active : plutôt que d'avoir un maximum d'informations à partir d'une image, la caméra est un capteur actif qui fournit de l'information limitée sur la scène. Le déplacement du capteur image revient à faire la focalisation dans l'espace 3D pour obtenir de l'information dont l'observateur a besoin.

Plus tard, (1988) R.K. Bajcsy introduit le concept de *Perception active* [6]. La perception active consiste à élaborer des stratégies de perception et d'action dans le but d'améliorer les performances des algorithmes de vision ou de détection par le contrôle des paramètres du capteur, ou de réaliser des tâches robotiques (positionnement, saisie, suivi,...). A la différence d'Aloimonos, Bajcsy vise le contrôle entre les différents modules d'un système de vision : les modules locaux représentent des procédures et paramètres tels que des distortions focales des objectifs, la résolution spatiale, des filtres passe-bande, etc.

[20]Pour diminuer la combinatoire pour la correspondance entre parties, on peut limiter l'espace de recherche d'autres parties en introduisant des contraintes selon la configuration spatiale [61].

A son tour, Tsotsos dans [113] analyse le problème de la perception visuelle dans une perspective de *recherche visuelle* : « à partir d'une cible et d'une image de test, est-ce qu'il existe une instance de la cible dans l'image de test ? » **Cette hypothèse est basée sur le fait que n'importe quelle tâche de vision où il y a un but peut être traitée comme une tâche de recherche visuelle**.

« *The basic visual search task is precisely what any model-based computer vision system has as its goal : given a target or set of targets (models), is there an instance of a target in the test display ?* » [113]

Des tâches comme l'appariement de forme, le problème d'alignement [102] et les procédures de reconnaissance connexionistes sont des versions spécialisées de la recherche visuelle. Quand il s'agit d'étudier la perception sous cette perspective, Tsotsos montre que le système de perception artificielle a besoin d'un mécanisme d'attention afin de faire, d'un point de vue de la complexité de la tâche, un problème pouvant être traité. Donc, sous cet angle, la recherche visuelle a une vue plus large que la vision active car elle intègre différents niveaux du spectre attentionnel qui vont de la sélection de la tâche, des événements, des objets à la sélection du modèle du monde, d'un champ visuel pour une analyse plus détaillée (mouvement de l'oeil), du champ visuel (mouvement de corps/tête), de la dimension d'intérêt dans l'espace géométrique et des caractéristiques dans le champ visuel (inhibitory beam), et finalement jusqu'à celle de paramètres (adaptation). Le spectre du mécanisme attentionnel proposé par Tsotsos est montré figure 1.9.

Le mécanisme attentionnel d'un système de vision peut être étudié sous la forme d'un problème de contrôle [41] et, principalement, tout mécanisme d'attention cherche à répondre aux trois interrogations suivantes :

- Quels sont les éléments de la scène à explorer (le problème du « où ») ?
- Quels sont les objets ou propriétés à rechercher (le problème du « quoi ») ?
- Quels sont les algorithmes, méthodes et stratégies à utiliser (le problème du « comment ») ?

Chacune de ces interrogations donne lieu à l'apparition de nouveaux problèmes :

- L'algorithme de contrôle doit être guidé par les indices, par le contexte ou les deux ?
- La modélisation de la scène doit elle être constituée de primitives ou d'objets ?

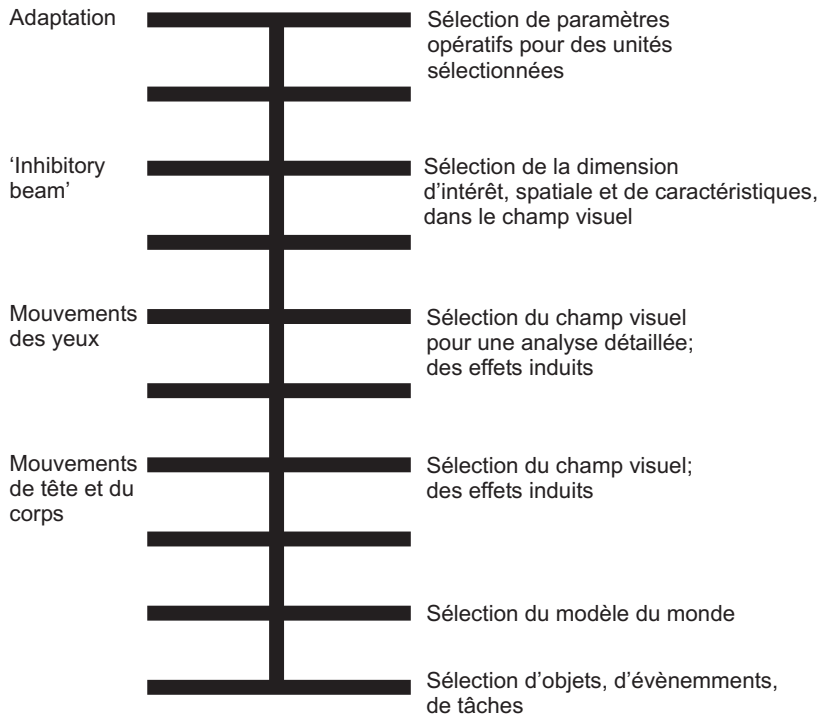


FIG. 1.9 – Le spectre des mécanismes attentionnels proposé par Tsotsos [113].

– Dans le cas où l'on connaît les réponses aux deux questions précédentes, dans quelle situation utiliser chacune des stratégies de contrôle ou type de modélisation ?

Plusieurs méthodes décrites dans la suite de ce chapitre répondent au moins à une de ces interrogations.

1.4.2 Mécanismes attentionnels

Des nombreux travaux ont été proposés comme mécanismes d'attention sélective, soit pour simuler et essayer de comprendre le fonctionnement de la vision biologique [114, 49, 55] ou soit pour optimiser des tâches de la vision artificielle [70, 117, 123, 108, 75, 92]. Dans cette section nous discuterons d'un des mécanismes les plus importants (à notre connaissance) concernant l'attention sélective.

D'après Tsotsos, l'attention sélective est un mécanisme des plus importants en vue de la réduction de la combinatoire dans la tâche de recherche en vision. « *L'attention sélective agit pour optimiser la procédure de recherche inhérente à une solution de vision, quelle soit naturellement mise en œuvre apar le cerveau, ou dans l'ordinateur* [114] ».

Comme proposé par Tsotsos, le mécanisme d'attention sélective semble impliquer les composantes basiques suivantes :

- sélection d'une région d'intérêt dans le champ visuel,
- sélection de la dimension de la caractéristique et les valeurs d'intérêt,
- contrôle du flot d'information à travers le réseau de neurones qui constituent le système visuel,
- déplacement d'une région sélectionnée à une autre dans le temps (l'itération suivante).

Parmi les éléments les plus importants dans le processus d'attention visuelle on trouve :

- **Le calcul des caractéristiques visuelles dans l'étape pré-attentive :** il s'agit de calculer, sous forme massive et parallèle, des caractéristiques bas-niveau qui peuvent fournir de l'information concernant les attributs visuels comme l'intensité, le contraste, la couleur, l'orientation, etc.
- **La définition des régions de saillance pour le contrôle bottom-up :** il s'agit d'une carte scalaire à deux dimensions qui représente la saillance visuelle. Il faut remarquer que, même si la saillance est indépendante de la nature d'une tâche particulière, elle peut être influencée par le contexte et les effets figure-fond [46, 49, 107].
- **La sélection d'attention et inhibition de retour (IOR) :** afin d'éviter de « revisiter » des positions, une mémoire à court terme est mise en place pour stocker des zones déjà traitées.
- **L'interprétation des scènes et la reconnaissance d'objets :** ils contraignent la sélection des positions espérées.

Afin d'étudier le rapport des réseaux simples, en considérant des éléments du « type neurone », avec une variété de phénomènes associés au déplacement de l'attention visuelle, en 1985 Koch et Ullman [57] ont été les premiers à proposer une représentation *topographique* pré-attentive (an early representation) de l'information visuelle. Cette représentation, appelée « saliency maps », est basée sur des caractéristiques élémentaires comme la couleur, l'orientation, la direction du mouvement, la disparité, etc. La sélection de la région la plus saillante est réalisée en utilisant un réseau du type « Winner-Take-All (WTA) ». Les points ou régions de saillance obtenus après la sélection, vont servir à leur tour comme indices de focalisation pour guider le processus de perception.

En 1993, Marsic [70] introduit la notion de SSC[21] qui peut représenter des objets et des parties des objets. Un mécanisme d'attention, du type bottom-up, est mis en place. Ainsi, les SSC sont utilisés, par un modèle de niveau plus élevé, comme indices de focalisation pour attirer l'attention : le mécanisme attentionnel sélectionne la caractéristique la plus saillante (de la carte de saillance, Koch and Ullman [57]) pour se focaliser dans une région spécifique de l'image et en échelle. Cela permet de naviguer à travers l'espace d'échelle.

Tsotsos et al (1995) [114] proposent un modèle d'attention par réglage sélectif du réseau du traitement visuel. Le processus de sélection est basé sur un réseau du type WTA afin de déterminer l'unité la plus saillante.

Dickinson et al (1997) dans [29] présentent une stratégie pour la reconnaissance active d'objets laquelle combine l'utilisation d'un mécanisme d'attention pour se focaliser dans l'image, avec une stratégie de commande de « point de vue » pour lever l'ambiguïté entre les caractéristiques obtenues de l'objet (vision active). Les objets 3D sont modélisés par des graphes d'aspect où chaque aspect correspond à un ensemble de parties volumiques. Ils utilisent un cadre Bayésien pour formaliser le mécanisme de focalisation.

Takacs et Wechsler (1998) dans [103], dans le cadre de la détection des indices faciaux (facial landmark detection), se sont concentrés sur la mise en œuvre d'un mécanisme d'attention guidé par les indices pour localiser des régions d'intérêt dans l'image. Le mécanisme proposé est dynamique et multirésolution pour le calcul de saillance locale et la génération de trajectoires de déplacement visuel (shift-of-attention).

Itti et al. (1998) [50] utilisent une carte de saillance, laquelle représente la saillance dans chaque localisation dans le champ visuel pour une quantité scalaire, pour sélectionner des régions dans une image où il est probable de trouver l'objet. Dans cette approche, le mécanisme d'attention est guidé par des indices (bottom-up) et il est atteint en utilisant un réseau du type WTA pour sélectionner les régions d'intérêt. Le FOA (focus of attention) est déplacé vers la localisation où se trouve le neurone gagnant. Cette approche utilise une méthode parallèle pour la sélection rapide d'un nombre réduit de positions d'intérêt dans l'image pour après utiliser des processus de reconnaissance d'objets.

Afin d'optimiser le processus de reconnaissance, Draper et al. [30] présentent le système ADORE dont la caractéristique principale est d'avoir la capacité d'apprendre l'ordre dans lequel les stratégies spécifiques pour la reconnaissance d'objets doivent être appliquées. De la même façon, Autio et al [4] font l'intégration des différents types de mé-

[21] Par les sigles en anglais Scale Space Cell. Une SSC est une représentation compacte d'un objet dans une zone spécifique dans l'espace d'échelle (scale-space).

thodes en proposant un système pour la détection d'objets. Il combine un mécanisme rapide d'attention visuelle (une variation des machines de supports vectoriels, SVM) et des algorithmes d'appariement moins rapides (basés sur l'algorithme de corrélation croisée normalisée). Le module de détection est utilisé pour classer les images « les plus difficiles » : celles qui exigent le plus d'attention.

Dans le cadre de la reconnaissance d'objets, Deco et Schürmann (2000) [25] introduisent un modèle hiérarchique d'attention sélective pour la *reconnaissance active d'objets*. Cette approche seulement considère l'aspect appelé *résolution d'hypothèse* (contrôle actif de la résolution spatiale par attention visuelle) dont le but principal est de prédire la résolution la plus adéquate afin d'obtenir des détails plus fins dans la région attendue. Deux modules appelés « where » et « what » sont utilisés : le premier analyse le champ visuel dans la résolution la plus basse afin d'extraire la position de l'objet (pas de détails), le deuxième est chargé d'extraire des attributs (comme la forme ou la couleur) et génère un groupe d'« objets candidats » en accord aux caractéristiques observées dans ce niveau d'analyse.

Walther et al. (2002), dans [119] présentent un modèle qui combine un mécanisme d'attention spatiale avec un module de reconnaissance d'objets. L'objectif principal du mécanisme attentionnel est de fournir une approximation au premier ordre concernant la localisation et l'extension des objets dans la scène. Ici, le mécanisme attentionnel fonctionne d'une façon bottom-up $\leftrightarrow$ top-down dont les indices correspondent aux régions de saillance obtenues à partir d'une carte de saillance (différentes orientations, intensités et couleurs). Le module pour la reconnaissance d'objets est basé sur un modèle hiérarchique pour la reconnaissance d'objets appelé HMAX.

Dans [76], Oliva et al. (2003) proposent un modèle d'attention guidé par le contexte (basé sur la configuration globale de la scène). L'hypothèse est que les caractéristiques bas-niveau sont hautement corrélées avec la localisation de l'objet dans la scène. La « modulation contextuelle » de la saillance est obtenue à partir de la probabilité de présence d'un objet sachant la position, un ensemble de mesures locales et caractéristiques contextuelles. Ce système intègre la saillance locale et des a priori concernant la position de l'objet. De même, Murphy et al. (2003) [75] suggèrent d'utiliser la scène (l'image comme un tout) comme une caractéristique globale, afin de résoudre des ambiguïtés locales pour la tâche de la reconnaissance d'objets. Leur approche est basée sur un modèle graphique probabiliste afin de faire l'inférence de la position possible de l'objet dans le plan image et d'initialiser la scène à partir de la détection d'un objet. Afin d'améliorer la vitesse et l'exactitude des détecteurs, une réduction de l'espace de recherche est faite ; ainsi le détecteur est balayé seulement sur des positions et des échelles où l'objet est attendu. Le

classifieur pour la reconnaissance d'objets utilisé est le GentleBoost.[22]

En 2004, Sun et Fisher [101] ont développé un modèle d'attention visuelle orienté-objet [23] lequel intègre l'attention visuelle guidée par le contexte et par des indices (l'extraction de caractéristiques est obtenue à partir la carte de saillance et le processus de sélection est mis en œuvre en utilisant un réseau du type WTA). La nouveauté de cette approche est le mécanisme pour le calcul de saillance basé sur le regroupement et, selon les auteurs, il représente la première réalisation d'un mécanisme d'attention visuelle hiérarchique et orienté-objet.

Ramström et Christensen (2004), dans [86] présentent un modèle pour la détection d'objets en utilisant le fond de l'image comme contexte. Une carte de saillance est utilisée pour obtenir de l'information pertinente et attirer l'attention dans une région spécifique de l'image. Le contexte est extrait en tant que régions cohérentes par un procédé de segmentation distribué. Ensuite, il est utilisé pour trouver des caractéristiques visuelles remarquables et des régions saillantes. Les déplacements de l'attention sont effectués en allant d'une échelle grossière à une échelle fine (coarse-to-fine).

Frintrop et al. (2004) [40], pour la détection et la classification d'objets, proposent un module d'attention visuelle (une variante du système proposé par Itti et al., 1998) qui est utilisé pour se focaliser dans une région de l'image ; le module d'attention proposé est guidé par les indices ou par une combinaison bottom-up↔top-down. Ensuite, la région choisie est l'entrée d'un classifieur de type Viola et Jones, 2001. Des classifieurs en cascade sont utilisés afin d'optimiser la tâche de classification. Seulement 30% de l'image est balayée en utilisant le module de focalisation.

Afin d'améliorer l'apprentissage non-supervisé et la reconnaissance d'objets dans des scènes hautement « bruitées », Walther et al (2004) dans [120] ont utilisé l'algorithme de Lowe [24] [62] pour démontrer l'utilité de l'attention visuelle pour la reconnaissance d'objets indépendante de la tâche et pour l'apprentissage de multiples objets à partir des images seules.

Draper et al (2005), dans [32] proposent un mécanisme d'attention sélective appelé SAFE[25]. Ce mécanisme est basé sur des principes similaires au NVT[26]. La caractéristique

[22] Une variante de l'AdaBoost qui utilise moins d'itérations pour être entraîné.

[23] L'attention visuelle peut être « object-based » (attention is allocated to a region, like a spotlight) ou « space-based » (attention is actually directed to an object or a group of objects to process any properties of selected objects rather than regions in the space)

[24] Lowe présente une nouvelle classe de caractéristiques locales appelées *SIFT features* (Scale-invariant feature transform) lesquelles sont localement stables dans l'espace d'échelle.

[25] Selective Attention as a Front-End (L'attention sélective comme entrée).

[26] Neuromorphic Vision toolkit développé par Itti et al, au CalTech et l'université de southern California.

principale du mécanisme proposé réside dans le fait que les points détectés (fixations) sont hautement invariantes aux transformations de similitude (2D transformations).

Machrouh et Tarroux (2005), dans [67] décrivent une architecture qui intègre des approches bottom-up↔top-down pour identifier des régions d'intérêt en fonction d'un but donné. D'abord, un contrôle bottom-up est utilisé pour identifier des régions d'intérêt, ensuite de l'information concernant la cible guide le mécanisme attentionnel pour faire la reconnaissance. La cible est considérée reconnue si le score de similitude (basé sur une fonction de base radiale), entre le point de salliance et la représentation de la cible, est supérieur à un certain seuil.

1.4.3 Bilan

Plusieurs approches concernant l'attention visuelle ont été décrites. Parmi elles, nous pouvons remarquer deux axes : le premier étudie l'attention visuelle principalement pour extraire des régions ou points d'intérêt, et le second qui se sert de l'attention visuelle afin de améliorer la reconnaissance d'objets [29, 50, 30, 4, 25, 119, 75, 76, 86, 40, 67]. Pour le premier, l'attention visuelle a été étudiée soit sous un point de vue bottom-up [57, 70, 114, 103, 50], où la caractéristique principale est l'utilisation d'une carte de saillance pour extraire des régions d'intérêt, soit sous un point de vue top-down [76, 75, 32, 86] qui se sert du contexte pour contraindre l'espace de recherche dans l'image, ou soit des approches hybrides où le cycle d'aller-retour bottom-up↔top-down est mis en place [119, 101, 40, 67]. Pour le deuxième axe, la présence de deux modules, celui d'attention visuelle et celui de reconnaissance d'objets, est une caractéristique commune indépendamment de la stratégie de contrôle choisie. Dans tous les cas, il est important de remarquer le **découplage** existant entre le but et l'information pré-attentive : bien que l'information extraite de la carte de saillance permette de guider le module de reconnaissance, les points d'intérêt obtenus sont indépendants des caractéristiques de l'objet en question, donc du but, en contradiction avec le principe fondamental de la vision active. Quel que soit le modèle (de l'objet, de la scène, du monde), il n'y a pas de lien existant entre l'information dite bas-niveau et les besoins.

1.5 Discussion et objectifs

Dans la section 1.2, nous avons présenté quelques caractéristiques les plus remarquables de la reconnaissance d'objets en vision biologique, ainsi que les deux théories les plus acceptées en psychologie cognitive : les modèles en primitives et les modèles basés sur l'apparence. L'invariance spatiale, la résistance au bruit et la capacité de généralisation sont parmi les caractéristiques les plus importantes. Pour représenter un objet, c'est le modèle en primitives le plus accepté et le mieux adapté d'après ces caractéristiques.

Par ailleurs, il a été discuté l'extrême importance du **mécanisme attentionnel** quand il s'agit d'optimiser le processus de perception. Cela peut expliquer la *rapidité* de la tâche de reconnaissance : une autre qualité remarquable de la vision biologique.

Dans la section §1.3.2, nous avons présenté les techniques correspondant à l'état de l'art en matière de reconnaissance d'objets pour la vision artificielle. Bien que des techniques issues de chaque catégorie aient donné des résultats assez performants, il y a plusieurs points sur lesquels elles peuvent être critiquées.

Tout d'abord, concernant l'approche d'**appariement du modèle** ou « template matching », même si cette technique a donné des résultats très satisfaisants pour la tâche de classification [83, 118, 93] nous pouvons apercevoir certains problèmes liés à celle-ci. Parmi les problèmes les plus notables on trouve :

- **Le besoin d'une recherche exhaustive en position et en échelle :** pour ce type de technique, très souvent on a besoin de faire le balayage de l'image par un « template », en position et en échelle (pour la détection d'objets variables en taille), d'une forme exhaustive et sans a priori. Ceci augmente significativement le temps de calcul. Afin d'éviter de faire une recherche de ce type, des indices de focalisation sont utilisés telles que : la couleur de la peau pour la détection de visages [100] ou des ombres pour la détection de véhicules [21]. D'autres solutions pour réduire l'espace de recherche sont l'utilisation des classifieurs en cascade, pour générer des hypothèses sur des régions potentielles où l'objet peut être trouvé [117, 92] ; ou directement la définition des contraintes données par l'utilisateur.

- **Pas de connaissance sur la structure de l'objet :** comme décrit dans §1.2.1.1, la structure est entre autre nécessaire pour connaître les parties occultées ou manquantes de l'objet, pour la localisation 3D et pour simplifier la reconnaissance des objets à différents points de vue[27]. Une représentation structurale basée sur l'appariement relationnel peut être cependant considérée.

- **Une représentation compacte :** les modèles basés en apparence n'utilisent typiquement que de l'information fréquentielle comme descripteur de l'objet. Du point de vue de l'attention visuelle, il peut être pertinent d'avoir plus de caractéristiques discriminatoires décrivant l'objet. Cela pourrait permettre de guider le processus de reconnaissance par des indices autres que fréquentielles : e.g. la couleur, le contraste, texture, etc. Cependant, le fait de considérer d'autres attributs de l'image, dans une approche comme celle-ci, peut augmenter considérablement la taille du

[27] La reconnaissance d'objets 3D à vues multiples est aussi possible avec un technique basée sur l'apparence en utilisant de multiples classifieurs (un pour chaque vue de l'objet).

vecteur de caractéristiques et par conséquence entraîner un temps de calcul prohibitif.

– **Besoin d'un grand nombre d'exemples :** un autre inconvénient des approches par apparence est lié à la base d'images qui doit servir pour l'entraînement du classifieur : les objets doivent être plus ou moins normalisés en taille, il faut réunir un grand nombre d'images d'objets, et avec une grande diversité, pour chaque catégorie à apprendre. On peut trouver dans la littérature des travaux qui ont identifié et qui commencent à traiter ce type de problèmes comme celui de Perona et al. [35]. Sinon, de gros problèmes peuvent survenir. Il y a des difficultés de réutilisation des éléments pour la description d'objets des différentes classes [109].

A notre avis, même si les outils et méthodes mathématiques sont très bien adaptés pour une tâche de classification, le problème de la perception visuelle (voire la reconnaissance d'objets) est loin d'être résolu en utilisant seulement ces techniques.

Concernant les méthodes d'**appariement relationnel** ou « relational matchers », on voit qu'elles représentent une option intéressante pour détecter des objets qui ont une apparence variable (comme par exemple observés par différents points de vue, ou des objets déformables) et que la tâche de reconnaissance peut être simplifiée avec l'utilisation d'informations a priori de parties précédemment détectées (voir les travaux réalisés dans [61, 73]). Cette contrainte pour délimiter la zone de recherche pour des futures parties, revient à ne faire qu'une *focalisation* seulement dans l'espace géométrique. On commence à percevoir le besoin de mécanismes de focalisation au niveau *reconnaissance d'objets*.

Par ailleurs, afin d'optimiser le processus de perception, on a les mécanismes attentionnels décrits en §1.4.2. Des approches comme celles de [119, 50, 49, 114, 29] montrent une bonne performance pour la sélection de régions où la présence de l'objet est probable.

En conclusion, quand il s'agit de reconnaître un objet, l'approche **classique** des méthodes décrites auparavant, et en général d'un système de perception, est la suivante : avoir un module de reconnaissance d'objets et un mécanisme d'attention visuelle chargé d'orienter le premier dans des régions d'intérêt. C'est donc une approche modulaire où il est nécessaire d'entrer dans le détail pour examiner quels sont les inconvénients qu'ils abordent.

La figure 1.10 énumère les différentes fonctions pour chacun des deux problèmes, reconnaissance et attention visuelle. A l'évidence il y a des fonctions semblables dans les deux cas (fléchées dans cette figure). Par exemple :

– **L'algorithme de contrôle** dans la reconnaissance d'objets, pour une approche du

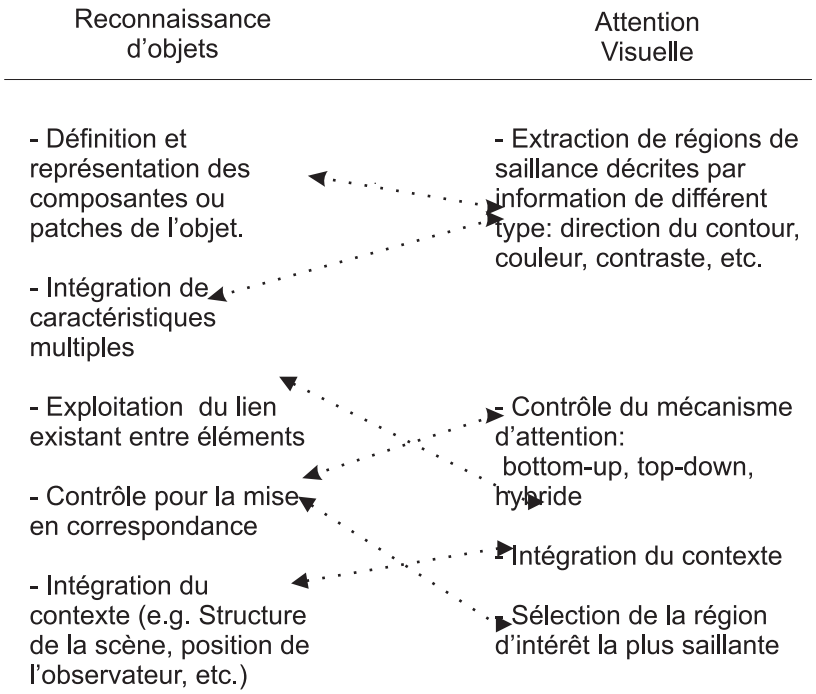


FIG. 1.10 – Approche modulaire d'un système de perception.

type appariement relationnel, a entre autre pour fonction de guider la détection des « patches » pour la mise en correspondance avec le modèle ; dans les mécanismes attentionnels l'algorithme de contrôle gère les déplacements de l'attention (orientation du module de reconnaissance dans des régions de saillance), ainsi comme l'inhibition de retour. Orientation des détecteurs d'objets, de détecteurs de « patchs », gestion d'opérateurs bas niveau... bien qu'à des niveaux différents, l'algorithme de contrôle est très identique ;

– **La représentation de l'information**, dans la reconnaissance d'objets, permet d'obtenir un modèle de l'objet à reconnaître (e.g. définitions de « patches ») ; dans l'attention visuelle, elle se présente sous forme d'une carte de saillance afin de définir des régions ou points d'intérêt. Une région issue de la modulation des données d'entrée par les caractéristiques du « patch » (e.g. couleur, texture, fréquence...), pourrait bien jouer le rôle d'indice de focalisation ou point d'intérêt afin de guider le processus de reconnaissance.

Ceci suggère que les **problèmes** de la reconnaissance d'objets et celui de l'attention visuelle, pourraient être liés de manière très intime et être étudiés sous un même formalisme. Est-ce qu'il est encore pertinent de poursuivre l'approche modulaire concernant les tâches d'un système de vision sachant que, pour résoudre une tâche locale sans ambi-

guïté, des informations sur les autre tâches sont nécessaires, qui pourraient elles-mêmes nécessiter des informations a priori sur la tâche en question – et/ou réciproquement ?

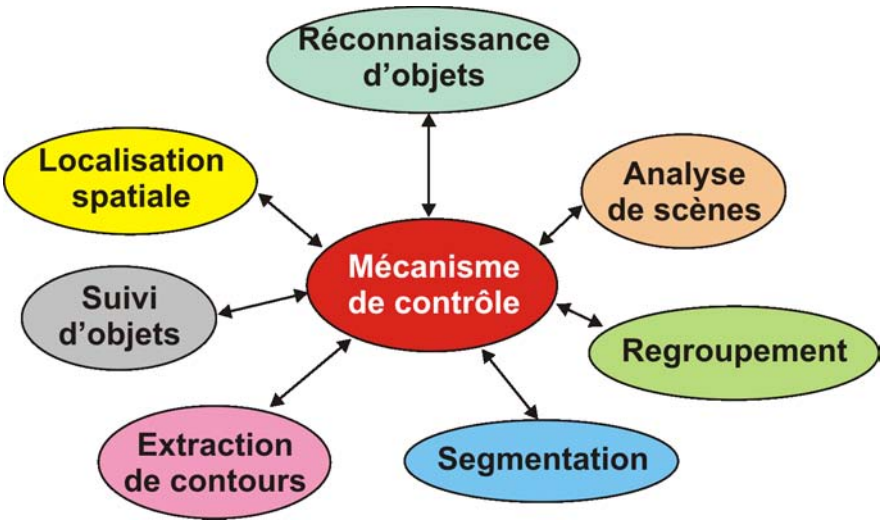


FIG. 1.11 – Approche modulaire d'un système de perception.

A la différence d'une approche classique modulaire telle que celle représentée figure 1.11 où un mécanisme de contrôle gère un ensemble de modules séparés, notre idée du processus de perception visuelle consiste en seul processus dans lequel les tâches comme par exemple la segmentation, la reconnaissance d'objets et l'analyse de scènes pourraient être des résultats intermédiaires dépendant du niveau d'abstraction. Un tel processus est structuré par étapes (voir figure 1.12) lesquelles sont décrites par la suite :

– L'étape d'apprentissage infère un modèle allant du plus complexe au plus élémentaire : monde, scènes, tâches, objets, opérateurs...

– L'étape de génération/vérification d'hypothèses s'appuie sur différents critères : pertinence, facilité, aptitude des opérateurs, vraisemblance, complétude, caractérisant la qualité d'adaptation. Pour plus de détails sur ces critères voir [41].

– L'étape de validation donne des résultats intermédiaires selon le niveaux d'abstraction qui va des primitives jusqu'à la scène. L'évaluation du critère de reconnaissance et de pertinence en sont des exemples.

– L'étape de focalisation réduit l'espace de recherche.

Ainsi, en fonction des besoins par rapport au modèle, le processus de perception à

boucle fermée émet les meilleures hypothèses selon certains critères, les vérifie et valide si l'objectif est atteint. Les observations précédemment réalisées sont utilisées afin de contraindre (focaliser) le processus pour la génération de futures hypothèses.

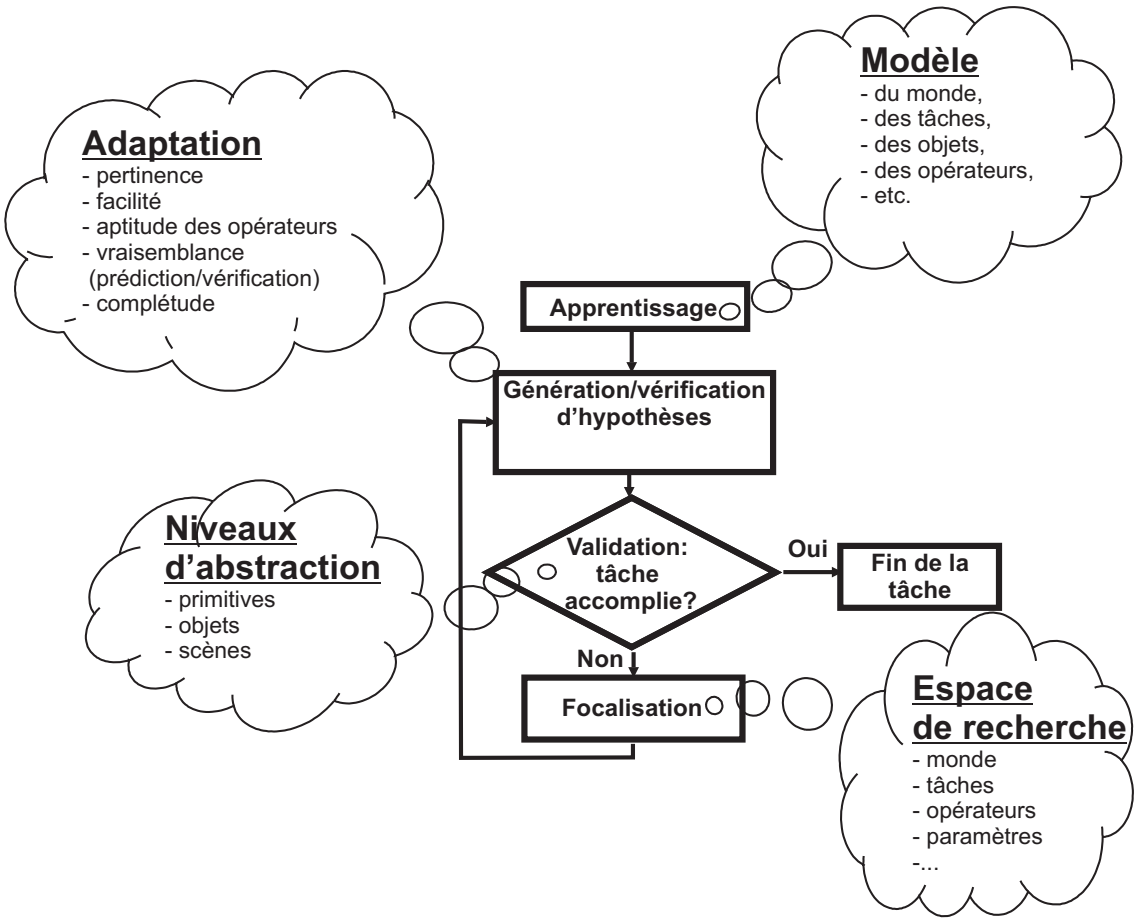


FIG. 1.12 – Schéma simplifié du processus de perception proposé.

Donc, la structure qu'on préconise prend en compte les points suivants :

– Le rôle actif de l'observateur : prendre en considération les mouvements du capteur et sa position par rapport à la scène observée, ainsi que le but à atteindre.

– L'exploitation de l'information concernant l'inter-relation existante parmi les objets ou éléments de l'objet. Ceci afin de contraindre l'espace de paramètres et ainsi éviter de faire une recherche exhaustive dans l'image, dans l'espace-échelle et dans l'espace des paramètres. Cela peut être atteint de deux façons :

1. en utilisant le contexte pour guider le processus de perception en fonction des buts (top-down), ou

2. en utilisant des indices visuels pour guider le processus de perception en fonction des propriétés ou éléments de l'objet (bottom-up).

– L'intégration de plusieurs caractéristiques de l'objet (à plusieurs niveaux de détail) pour avoir une représentation plus complète, et pour pouvoir guider le processus de perception par des indices de différents types et non seulement géométriques ou fréquentiels.

– L'adaptation de la structure à différents niveaux d'abstraction (e.g. reconnaissance d'objets ou analyse de scènes).

Notre travail de recherche a été initié par le travail développé par Aufrère et al. dans [3] pour le problème de la reconnaissance de la chaussée pour les véhicules intelligents. Devant la complexité de la tâche, les auteurs ont été forcés de trouver une méthodologie capable de prendre en compte des imprécisions des détections des caractéristiques, la variabilité en forme (des lignes blanches peuvent être vues comme un objet flexible), et le manque d'information. La stratégie de reconnaissance est basée sur un algorithme récursif de génération et vérification d'hypothèses où un mécanisme d'attention, guidé par le contexte, a été mis en place. Ceci a donné l'idée de formaliser et tester notre approche pour un niveau d'abstraction donné, i.e. la reconnaissance d'objets. Bien qu'il serait intéressant de donner une méthodologie sur la conception des systèmes de perception, il nous a paru plus judicieux de traiter ce *problème local* de la vision par ordinateur, en donnant une architecture de départ et en faisant attention qu'elle puisse être adaptée à d'autres problèmes de niveaux d'abstraction différents.

Afin de valider et tester notre approche, nous avons choisi une représentation structurelle de l'objet, au moyen d'une grille de cellules, où chaque élément est décrit par sa structure locale et l'ensemble décrit l'apparence globale de l'objet. Cela permet de rechercher un objet à partir de ses éléments, traiter les problèmes de la détection d'objets variables en échelle et faire l'intégration de multiples caractéristiques. Le fait de modéliser l'objet par sa structure locale, nous permet d'éliminer le besoin des entités abstraites comme celles de contours, surfaces, etc. ou « parties » des objets telles que : oeil, nez, bouche, etc., et traiter directement les données provenant du bas niveau. Ici, on ne parle plus des représentations *abstraites* des éléments de l'objet mais plutôt de représentations acquises et apprises par le système lui-même à partir des stimuli. On pense qu'un système de vision doit être capable d'acquérir par lui-même la représentation de l'objet en ques-

tion et non à partir des primitives que l'utilisateur doit définir.

Notre idée ici est de voir la reconnaissance d'objets comme un processus actif dont les entrées sont les réponses des opérateurs appliqués directement sur l'image : stimuli sous forme d'impulsions, et la sortie est l'objet reconnu. *L'attention visuelle* reste un mécanisme implicite du processus de perception lequel est régi par le cycle d'allers-retours *bottom-up* $\leftrightarrow$ *top-down*. Nous verrons que d'autres tâches comme la localisation et le suivi s'intègrent naturellement dans le cadre proposé.

La méthodologie proposée a été testée pour la reconnaissance, localisation et suivi de visages et piétons.

Chapitre 2

Reconnaissance d'objets par vision focalisée

2.1 Introduction

Dans le chapitre I, nous avons réalisé un survol des techniques existantes pour la reconnaissance d'objets. Nous avons vu le rôle assez important du mécanisme d'attention visuelle pour optimiser le processus de perception chez l'homme. Ensuite, on a pu apercevoir la tendance et le besoin d'intégrer des mécanismes d'attention visuelle dans les systèmes artificiels, comme le montrent également les techniques décrites jusqu'aujourd'hui [29, 103, 50, 120, 121, 4, 25, 119, 75, 86, 40, 67]. L'intégration de ces mécanismes est nécessaire à *l'amélioration de la compréhension et l'analyse d'une scène visuelle*, ce qui correspond à un des problèmes les plus importants de la vision par ordinateur. C'est autour de ce problème que nous devons nous concentrer et ainsi adapter au mieux les méthodologies et techniques développées.

Dans une scène, mis à part le fait de connaître les éléments constituant la scène, il est nécessaire de connaître l'interaction entre ces différents éléments. Par exemple, une situation courante dans une scène 3D est la superposition d'objets. Ce type d'interaction peut donner comme résultat qu'un objet se trouve derrière un autre, ce qui entraîne des occultations partielles ou totales. Ainsi, la détection ou la reconnaissance correcte d'un objet peut *dépendre* de la connaissance d'autres éléments composant la scène. Imaginons une scène urbaine composée d'un véhicule et d'un piéton, où ce dernier traverse la scène de gauche à droite en passant derrière le véhicule. Quand le piéton n'est pas caché derrière le véhicule, un algorithme de reconnaissance d'objets, tel qu'un algorithme du type classifieur d'appariement de modèle, peut réaliser une détection correcte de celui-ci. Cependant, quand le piéton est occulté à 50% par le véhicule, un algorithme de ce type peut ne pas détecter le piéton correctement, du fait qu'il n'a pas le contrôle des parties locales mais de l'objet comme un tout. Pour l'homme, il semble facile de reconnaître cor-

rectement l'objet dans une situation comme celle-ci : il nous suffit d'observer quelques éléments composant l'objet (comme par exemple la tête du piéton) pour émettre un jugement correct.

Dans ce cas, si le système d'analyse de scènes connaît la présence du véhicule, il doit savoir que la superposition des deux objets implique l'occultation et le fait de ne pas pouvoir observer certaines parties de l'objet. Ainsi, le module chargé de détecter le piéton pourrait éventuellement entraîner une modification sur son critère de reconnaissance en fonction des *parties observables*. Une telle solution est impossible à proposer avec des approches du type « template matching », car la modification du seuil de reconnaissance peut impliquer potentiellement de fausses détections.

Typiquement le problème de la compréhension d'une scène se fait d'une manière « découplée » : la reconnaissance d'un objet est complètement indépendante de la structure de la scène, et les caractéristiques de l'objet ne sont pas exploitées pour faciliter la détection d'autres objets. *Nous orientons notre recherche sur le fait que le problème de l'analyse de scènes doit être traité sous une approche unificatrice. Des problèmes comme ceux de la compréhension des scènes, de la reconnaissance d'objets et de l'attention visuelle doivent être traités dans un même cadre formel.* Dans le cadre de la reconnaissance de scènes, cette technique fournira des résultats en termes de reconnaissance d'objets s'inscrivant dans une scène. On peut espérer que ces résultats seront plus pertinents que dans le cadre d'une approche uniquement « orientée objet ».

Dans un souci de rendre ce travail plus pertinent et concret, nous avons décidé d'étudier cette approche dans le cadre particulier de **la reconnaissance visuelle d'objets**. Il est évident que notre position ne s'inscrit pas dans le cadre de « reconnaissance de formes » (pattern recognition) avec des approches spécifiques, mais plutôt dans un cadre général où l'objet s'inscrit dans une scène. L'objectif à terme, sera de l'étendre vers la compréhension des scènes visuelles.

Dans ce travail, nous proposons donc une méthodologie pour la reconnaissance visuelle d'objets. Cette méthodologie intègre un mécanisme d'attention afin de guider le *processus de reconnaissance* à partir d'indices visuels (*bottom-up*), ou à partir du modèle de l'objet ou du contexte (*top-down*). On parle de *processus* car, ici la reconnaissance est faite à partir d'une recherche séquentielle des parties qui composent l'objet, et la détection de chaque partie contraint l'état du processus. Il y a d'autres raisons pour lesquelles on préfère voir la reconnaissance comme un *processus* et non comme une tâche de classification où l'objet est vu comme un tout. En voici quelques-unes :

– avoir une connaissance des parties cachées ou absentes de l'objet afin de conditionner le critère de reconnaissance,

– avoir une augmentation du rapport signal bruit (SNR) pour l'observation des *parties*, i.e. à partir de la détection d'une *partie* on peut conditionner le processus afin de mieux en détecter d'autres,

– faire une recherche progressive (du coût de détection le plus faible au plus élevé) des *parties* de l'objet. Commencer le processus de reconnaissance à partir de caractéristiques simples, pertinentes, faciles à détecter et se concentrer ensuite sur une zone spécifique de l'image pour chercher des caractéristiques plus complexes ou plus coûteuses.

Le schéma présenté sur la figure 2.1 montre le système proposé pour la reconnaissance d'objets. Dans ce système on trouve trois parties principales :

– l'image,

– le traitement primaire (bas niveau),

– le haut niveau incluant le mécanisme de contrôle.

La première partie correspond au capteur d'image (rétine) qui permet d'avoir une représentation du monde extérieur à partir des stimuli. Dans ce niveau, l'observateur n'a pas un rôle actif sur le processus de reconnaissance sauf si le capteur image peut être orienté. Même si l'approche pourrait permettre de gérer cette situation, nous supposons dans notre cas utiliser un capteur fixe. La deuxième partie correspond au traitement primaire. Cette partie est composée de N modules indépendants chargés d'extraire des propriétés de l'image comme information fréquentielle, couleur, texture, etc. L'information provenant du capteur image est traitée (à différents niveaux de résolution) par ces modules. Il faut remarquer qu'ils sont contrôlés par le haut niveau en fonction des besoins. Comme évoqué dans le chapitre I, l'attention visuelle peut être vue comme un processus de « filtrage » d'information, et c'est dans cette partie que le filtrage est fait.

Le mécanisme d'attention sélective opère sur quatre aspects différents :

– choix du ou des modules qui doivent opérer sur l'information provenant de la rétine (sélection des opérateurs),

– intervalle d'action des opérateurs choisis (**ROI** dans l'espace de caractéristiques). Par exemple : l'intervalle de niveau de gris moyen, l'orientation du contour attendue, etc.,

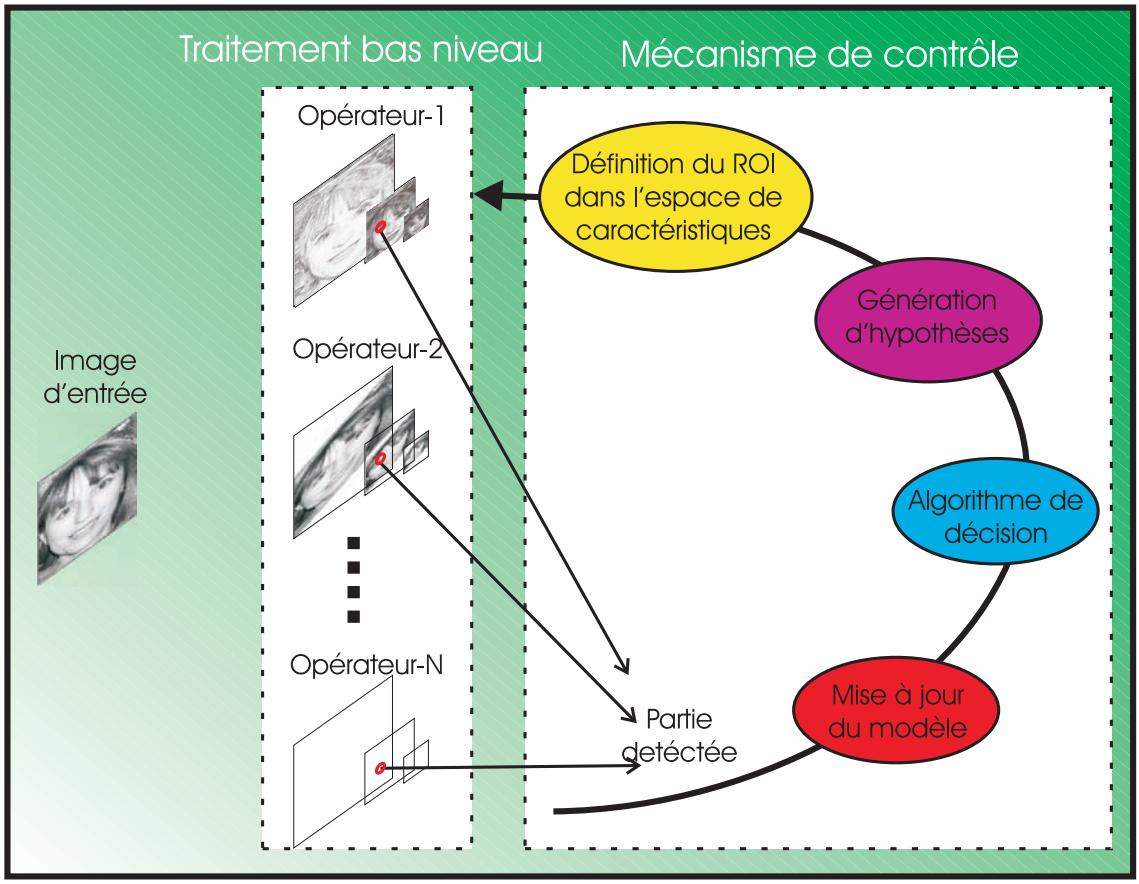


FIG. 2.1 – Schéma montrant le système que l'on propose pour la reconnaissance d'objets.

– niveau(x) de résolution sur le(s)quel(s) chaque opérateur doit agir, et

– région dans l'image où les opérateurs doivent agir (fenêtrage).

La sortie de chacun des modules est concentrée sur une grille de cellules décrite dans la section suivante ; chaque cellule contient l'information provenant des modules primaires dans une région spécifique de l'image (capture de la structure locale). C'est à partir de cette grille de cellules que le haut niveau peut accéder aux réponses des opérateurs.

La partie *haut niveau*, qui correspond au « cœur » du système, a pour rôle de « gérer » le processus de reconnaissance de l'objet. Dans cette partie se trouve le modèle de l'objet à chercher. Ce modèle comprend non seulement l'information concernant la structure ou

l'apparence de l'objet mais aussi d'autres informations comme : sa position probable dans la scène, les relations qui peuvent exister entre d'autres objets et les éléments nécessaires pour faire la hiérarchisation des parties composant l'objet. C'est à partir de ce modèle que le haut niveau guide le processus de reconnaissance. La génération d'hypothèses, le contrôle du processus de reconnaissance, la mise à jour du modèle et la prise de décisions sont des tâches correspondant à ce niveau.

Donc, la méthodologie que l'on propose est appliquée à la reconnaissance d'objets mais elle a été développée en gardant, autant que possible, un caractère générique afin de pouvoir s'adapter à différents niveaux d'abstraction, i.e. analyse des scènes, reconnaissance d'objets. Parmi les caractéristiques les plus importantes de cette approche, on peut citer les suivantes :

- elle permet de faire la détection, la reconnaissance, la localisation et le suivi simultanément,
- elle est adaptable en fonction des besoins en accord avec la tâche courante,
- elle prend explicitement en compte le contexte pour faciliter la tâche de reconnaissance,
- elle intègre un contrôle du processus guidé par le but (top-down) et guidé par des indices (bottom-up),
- elle construit une représentation de l'objet capable d'intégrer de multiples caractéristiques sans augmenter significativement la complexité de sa détection,
- elle est robuste aux changements d'échelle, aux changements de luminosité et aux occultations.

Afin de mieux décrire notre approche, la méthodologie sera présentée de la façon suivante : d'abord, nous expliquerons la représentation et modélisation de l'objet à reconnaître ; ensuite, la construction du modèle et la définition de tous ses éléments seront présentées dans la phase d'apprentissage. La troisième partie correspond à la description de la stratégie de reconnaissance choisie, en présentant chacun des modules qui composent le mécanisme de contrôle du processus. Un exemple d'évolution de l'algorithme proposé sera donné. Enfin, une quatrième partie est destinée à la localisation et au suivi d'objets.

2.2 Représentation et structure de l'objet

2.2.1 Introduction

Quand on cherche à développer une méthodologie qui puisse utiliser un mécanisme pour guider le processus de reconnaissance, on rencontre certains problèmes très importants (par exemple celui lié à la modélisation de l'objet à reconnaître) qui bien évidement, ne sont pas propres à cette technique.

Un objet est une entité indépendante caractérisée par différentes propriétés physiques telles que la couleur, la taille, la forme, la texture, etc. Dès lors se pose la question de savoir quelle est la représentation la plus pertinente de celui-ci (soit une représentation structurelle, basée sur l'apparence ou une approche hybride), pour non seulement permettre sa reconnaissance mais aussi pour que le mécanisme d'attention visuelle puisse en tirer profit. Le type de représentation doit être le plus conforme à la tâche qu'on souhaite réaliser, par exemple, pour une tâche de recherche visuelle donnée, n'importe laquelle des propriétés de l'objet peut être utilisée comme indice de focalisation pour guider le processus de perception visuelle[1]. Si on veut un système qui reste assez général pour la modélisation d'un objet quelconque, on doit répondre aux questions suivantes :

– Le contenu fréquentiel, la couleur, la texture ou la direction du contour dépendent de l'endroit où l'on observe l'objet. Quelle est la meilleure manière de représenter l'objet pour prendre en compte ces paramètres ?

– Dans quelle échelle d'analyse doit-on observer telle ou telle propriété ? Globale, détaillée ?[2]

– Comment faire pour éviter d'indiquer *explicitement* les régions que l'on pense pertinentes pour la bonne caractérisation de l'objet ?

Concernant le premier point, cela suggère[3] une représentation par composantes struc-

[1] Les entités « abstraites » comme par exemple des *parties* déterminées de l'objet, i.e. un segment de ligne, nez, oeil, etc., peuvent être aussi utilisées comme des indices pour focaliser le processus de reconnaissance.

[2] D'après l'analyse fréquentielle, un objet est composé de différentes composantes typiquement classées en hautes, moyennes et basses fréquences. Afin d'avoir une représentation plus complète de l'objet, les différentes bandes fréquentielles doivent être prises en compte.

[3] Cependant une représentation RpC n'est pas obligatoire, comme dans le cas de l'attention *top-down* dans l'espace géométrique. On peut délimiter la zone d'intérêt sur la position probable de l'objet et appliquer un classificateur dont l'objet est vu comme un tout, i.e. classificateur SVM ou NN. Pour la focalisation indépendante du modèle (*bottom-up*), une représentation basée sur l'apparence peut être suffisante ; comme c'est le cas de l'utilisation des cartes de saillance [53, 50, 49, 119] pour identifier des régions d'intérêt afin de focaliser la reconnaissance.

turales (RpC). Comme décrit au chapitre I, l'utilisation de ce type de représentation amène au problème de la définition et de la détection des composantes de l'objet ; ce qui a favorisé l'expansion des approches basées sur une représentation par apparence. Ainsi, afin de surmonter le problème de la modélisation et d'avoir un point de départ, une idée cohérente est de choisir une représentation hybride de l'objet, en d'autres termes, une représentation basée sur l'apparence mais aussi qui intègre un ensemble de *parties*[4] composant la structure de l'objet. Cette idée n'est pas nouvelle. Schneiderman [93] propose une représentation du même type, dans laquelle l'objet est représenté par un ensemble de *parties* ; chacune correspondant à une région dans l'image (groupe de pixels) qui capture la structure locale, tout en gardant les dépendances statistiques en apparence de l'objet. Nous partons de cette notion de *partie* pour l'étendre afin d'avoir une représentation à caractéristiques multiples. Dans cette optique, on ne parle plus des représentations avec des segments de lignes, surfaces, icônes, etc., ni d'aucune représentation « abstraite »[5] de l'objet, mais plutôt de représentations acquises et apprises par le système lui-même à partir des stimuli (sorties des opérateurs bas niveau). Dans notre esprit, un système de vision doit être capable d'acquérir, par lui-même, les éléments qui composent l'objet en question, non pas à partir des primitives que l'utilisateur doit définir mais des réponses des opérateurs agissant directement sur l'image.

Un avantage d'avoir les *parties* caractérisées par la statistique sur la structure locale, est qu'elles peuvent être détectées avec facilité. En plus, le fait de disposer d'une structure de l'objet permet d'avoir une connaissance explicite sur les composantes de ce dernier ; ainsi il devient possible de traiter des objets déformables ou d'apparence globale variable.

2.2.2 Composants d'un objet : définition de « *partie* »

L'objet est considéré comme composé d'un ensemble de *parties* dépendantes entre elles. Comme indiqué précédemment, **chaque *partie* correspond à une région de l'image (groupe de pixels) qui décrit la structure locale de l'objet**. Elle est notée Λ_m ($m = 1,\ldots,M^p$). M^p est le nombre total de parties composant l'objet.

Pour une *partie* Λ_m quelconque, on définit :

– **Un vecteur de paramètres, λ_m.** Le vecteur $\lambda_m = [\zeta_{m,1}, \zeta_{m,2}, \ldots, \zeta_{m,N_m}, \mathbf{a}_m^t]^t$ est composé d'un ensemble d'éléments ζ_{ij}, où chaque élément peut être une propriété physique comme la couleur ou un niveau de gris moyen, la réponse fréquentielle

[4] Désormais on utilisera le mot italique *partie* pour désigner un composant de l'objet.

[5] La problématique liée à l'utilisation de surfaces, contours, etc., comme primitives de base pour atteindre le but final est bien connue dans la communauté scientifique. En effet, pour faire la segmentation ou extraire des contours sans trop d'ambiguïté, on a besoin de mécanismes beaucoup plus complexes que de simples détecteurs bas niveau.

adaptée dans une zone donnée de l'image, etc., et d'un vecteur $\mathbf{a}_m = [u, v]^t$ qui désigne les coordonnées de Λ_m dans l'image. L'idée ici est d'avoir non seulement une représentation plus complète de l'objet, mais aussi de permettre le guidage du processus de perception à partir d'indices de différents types, intégrant des **caractéristiques multiples**.

– **Un poids de pertinence, w_m.** Ce poids indique la pertinence dans le processus de reconnaissance, de la *partie* à laquelle il est associé. Il est de nature probabiliste et il indique la probabilité d'avoir une bonne détection de Λ_m dans une image donnée. Ce poids est important car il permet d'optimiser le processus de perception en sélectionnant, par exemple, la *partie* qui a le moindre coût de détection et une probabilité élevée de présence.

– **Un indice de résolution, r_m.** L'indice r_m sert à indiquer le niveau de résolution auquel appartient la *partie* Λ_m.

– **Une fonction de détection.** La fonction de détection $\hat{\lambda}_m = f_m(\bar{\lambda}_m, \Sigma_{\lambda_m})$, associée à une *partie* Λ_m, extrait une observation $\hat{\lambda}_m$ dans une région d'intérêt centrée sur $\bar{\lambda}_m$, où $\bar{\lambda}_m$ et Σ_{λ_m} correspondent au vecteur moyen de paramètres et la matrice de covariance associée, respectivement. Il faut remarquer que cette région d'intérêt est définie dans l'espace des caractéristiques de dimension $(N_m + 2)$ et donc pas seulement dans l'espace géométrique. Lorsque cette fonction est lancée, plusieurs candidats possibles peuvent être trouvés pour la *partie* cherchée. Dans cette approche, on n'a pas cherché à concevoir des détecteurs très spécialisés pour un nombre réduit de *parties*, mais on exploite plutôt l'idée d'un détecteur plus généraliste. Cette faiblesse sera compensée par l'intégration d'un grand nombre de *parties*, de multiples caractéristiques et la focalisation des détecteurs (voir §2.4.5).

La figure 2.2 montre la structure hiérarchique de l'objet et ses éléments associés, ainsi que la représentation du modèle de l'objet.

Afin de savoir à quelle région de l'image correspond une *partie* Λ_m, on utilise une grille de M^c cellules, définie ci-après. Cette grille a pour objectif de « disséminer » les N paramètres dans plusieurs régions de l'image qui pourraient, lors de l'apprentissage, être définies comme des *parties* représentatives de l'objet.

Un des buts de l'apprentissage (voir §2.3) sera d'extraire de la grille de M^c cellules, les M^p cellules les plus caractéristiques de l'objet et ne garder que les N_m paramètres les plus discriminants de celles-ci (N_m potentiellement différent pour chaque cellule, et $N_m \leq N$). Les M^p cellules sélectionnées **seront les *parties* composant l'objet**. Chaque *partie* sera caractérisée par un vecteur $\lambda_m = [\zeta_{m,1}, \zeta_{m,2}, \ldots, \zeta_{m,N_m}, \mathbf{a}_m^t]^t$ dont les N_m paramètres sont

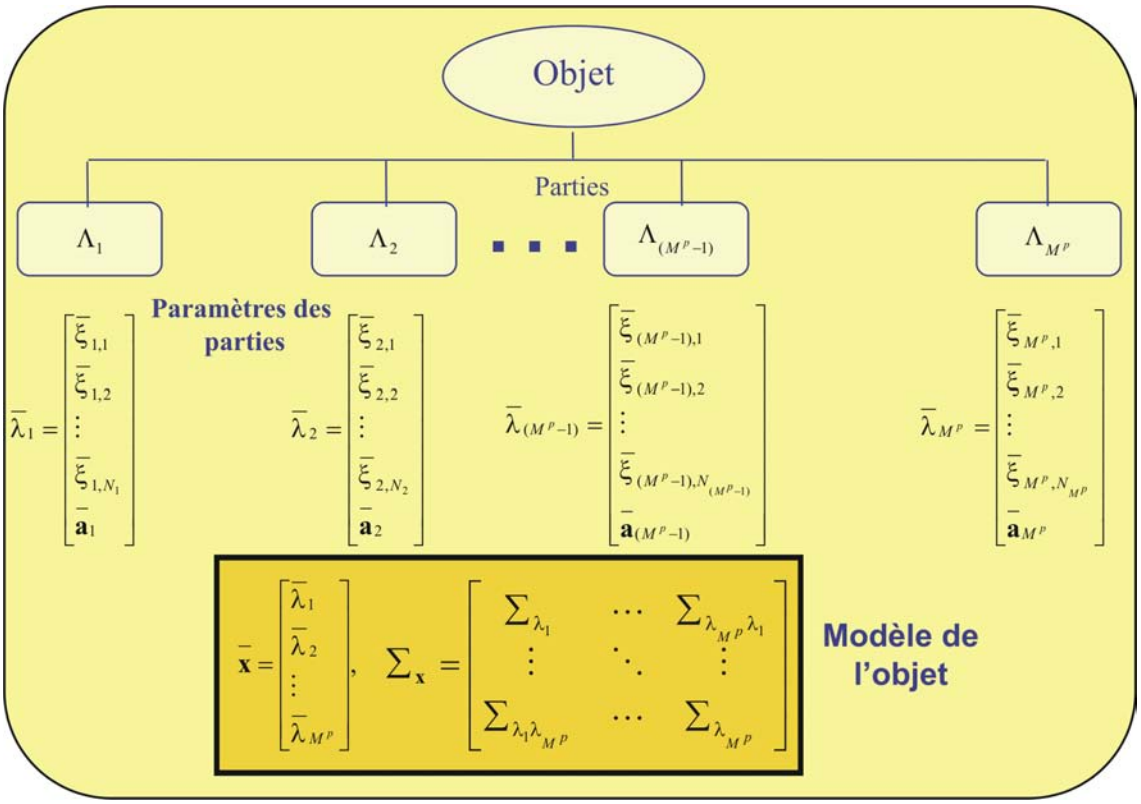


FIG. 2.2 – Structure hiérarchique de l'objet.

un sous-ensemble des N paramètres initiaux à l'intérieur de chaque cellule.

2.2.3 Cellules

Le but est donc d'extraire la ***structure locale*** de l'objet. Cette structure est caractérisée par un ensemble de propriétés de l'objet comme la couleur, le contenu fréquentiel, etc. Pour cela on utilise des « cellules » $\mathcal{C}$ définies ci-après.

2.2.3.1 Définition

Une cellule, $\mathcal{C}$, est définie comme une entité qui capture la structure locale d'une image à partir d'opérateurs bas niveau. Cette entité présente la particularité de pouvoir faire l'observation de multiples propriétés de l'objet dans une région spécifique de l'image (intégration de caractéristiques multiples). A l'intérieur de chaque cellule (voir figure 2.3) sont accessibles les données provenant des N opérateurs bas niveau sélectionnés avant l'apprentissage. **L'objectif de l'apprentissage est d'extraire, à partir de l'ensemble des**

M^c **cellules, les** M^p **plus représentatives qui seront les parties de l'objet** (voir §2.3). Avec ce type de représentation, on est capable de faire la focalisation à partir des indices comme la couleur, la direction du gradient dans une partie de l'image, par exemple, ou d'un ensemble d'indices e.g. un contour localisé dans une région avec une couleur déterminée. Ensuite, l'étape d'apprentissage ne doit pas seulement se charger de déterminer quels attributs sont les meilleurs descripteurs ou les plus discriminants dans une région donnée de l'objet, mais également lequel de ces attributs est le plus pertinent pour guider le processus de reconnaissance vers les candidats possibles.

Une cellule, $\mathcal{C}$, sert à extraire l'information locale de l'image à partir des opérateurs. Une *partie*, Λ, est définie comme une **cellule déjà caractérisée** par sa statistique locale lors de l'apprentissage, et qui devient un descripteur local. C'est là ce qui les différencie.

Donc, pour une cellule d'indice m, on a

$$\varsigma_m = \begin{bmatrix} \zeta_{m,1} \\ \zeta_{m,2} \\ \vdots \\ \zeta_{m,N} \\ \mathbf{a}_m \end{bmatrix}, \quad \Sigma_{\varsigma_m} = \begin{pmatrix} \sigma^2_{\zeta_{m,1}} & & & & \\ & \sigma^2_{\zeta_{m,2}} & & \mathbf{0} & \\ & & \ddots & & \\ & \mathbf{0} & & \sigma^2_{\zeta_{m,N}} & \\ & & & & \Sigma_{\mathbf{a}_m} \end{pmatrix} \tag{2.1}$$

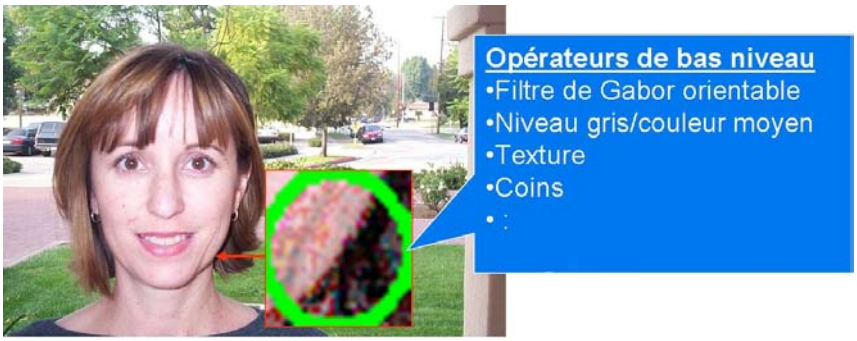


FIG. 2.3 – Exemple d'une cellule

où ς_m correspond au vecteur de paramètres de la cellule $\mathcal{C}_m$, Σ_{ς_m} est sa matrice de covariance, N est le nombre total des opérateurs bas niveau, $\mathbf{a}_m$ et $\Sigma_{\mathbf{a}_m}$ correspondent respectivement aux coordonnées de $\mathcal{C}_m$ et à sa matrice de covariance associée. Les zéros dans la matrice de covariance sont dûs à la supposition d'indépendance statistique entre

les N opérateurs **pour une cellule donnée**. On gardera par contre la dépendance dans un même opérateur entre plusieurs cellules (on supposera par exemple que le niveau de gris moyen d'une cellule peut être statistiquement lié à celui d'une autre cellule mais qu'il n'est pas lié à l'orientation du contour dans cette même cellule).

2.2.3.2 Grille de cellules

Comme on ne connaît pas a priori à quoi correspond chaque partie de l'objet, nous utilisons une grille de cellules distribuées d'une façon uniforme en formant un carré ou un rectangle, selon la boîte englobante de l'objet à apprendre ou selon la base de données. Cette grille est placée sur l'objet à reconnaître qui doit être centré. C'est pourquoi on a besoin soit d'une base de données annotée afin de placer la grille sur l'objet, soit des images avec des objets plus ou moins centrés et normalisés en taille.

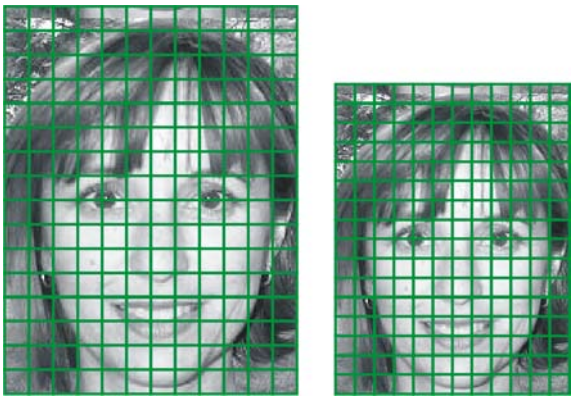


FIG. 2.4 – Exemple d'une grille de cellules. Pour deux objets de taille différentes on a un nombre fixe de cellules.

Dans la mesure où les objets que l'on peut trouver dans une image sont variables en taille et en position, à chaque cellule doivent être associées des coordonnées $\mathbf{a}_m = [u, v]^t$ afin d'apprendre leur position relative. Sur la figure 2.4 on montre une grille de cellules placée sur deux images de taille différente. Le nombre de cellules pour les deux images reste le même mais la taille de chaque cellule varie selon la taille de l'image[6]. Cette notion de « grille élastique » nous permet d'avoir un nombre fixe d'éléments représentant l'objet, à la différence des pixels de l'image classiquement utilisés.

[6]Cela n'est vrai que pour l'apprentissage. Pendant la phase de reconnaissance, car on ne connaît pas a priori la taille de l'objet, la taille de la cellule reste fixe.

2.2.3.3 Opérateurs bas niveau

Un opérateur peut être n'importe quel type de traitement d'image bas niveau (ou éventuellement on pourrait par exemple envisager d'autres sources d'information : radar, télémètre laser, information stéréo, ...). Tous les opérateurs sont considérés ici comme statistiquement indépendants et on peut les classifier en deux types : les opérateurs qui répondent toujours (e.g. couleur) et ceux qui ne répondent pas toujours (e.g. contour). Cette différentiation est pertinente lors de la caractérisation probabiliste des opérateurs dans la phase de décision (voir §2.4.6). Comme nous l'avons décrit auparavant, cette notion d'opérateurs désigne un traitement primaire après la rétine. La façon d'utiliser l'information provenant des opérateurs est déterminée par la stratégie de contrôle (haut niveau).

2.2.3.4 Grille de cellules multi-résolution

On a discuté dans le premier chapitre du besoin de disposer d'une représentation multi-échelle d'une image. Afin de prendre en considération ce type d'information pour avoir une représentation plus complète de l'objet, une grille multi-résolution est nécessaire. Une intégration de ce type présente l'avantage de permettre au processus de reconnaissance d'avoir des indices multi-échelle pour la focalisation. De nombreux travaux ont utilisé une approche « coarse to fine » (de l'échelle grossière à l'échelle plus fine) afin d'utiliser des descripteurs de basse résolution (échelle grossière) pour détecter plus rapidement des descripteurs à haute résolution [70].

Afin de diminuer le temps de calcul, une décomposition pyramidale est utilisée au lieu d'une analyse multi-échelle. Pour la grille de cellules multi-résolution on utilise l'analyse multi-résolution typique 2^n, ce qui signifie que l'image est décomposée en n résolutions dont le sous-échantillonnage est une puissance de 2. La grille multi-résolution utilisée est présentée sur la figure 2.5.

2.2.4 Bilan sur la représentation

Dans cette section nous avons procédé au choix de la forme de représentation de l'objet qui nous semble la plus pertinente. En effet, nous avons fait le choix d'une forme de type hybride grâce à laquelle l'objet peut être représenté à partir d'un ensemble de *parties* décrivant la structure locale de l'objet, tout en conservant son apparence globale. Chaque *partie* Λ_m est composée de :

- Un vecteur de paramètres, $\lambda_m = [\zeta_{m,1}, \zeta_{m,2}, \ldots, \zeta_{m,N_m}, \mathbf{a}_m^t]^t$, dont $\zeta_{m,n}$ correspond au paramètre issu de l'opérateur n.

- Un poids de pertinence, w_m.

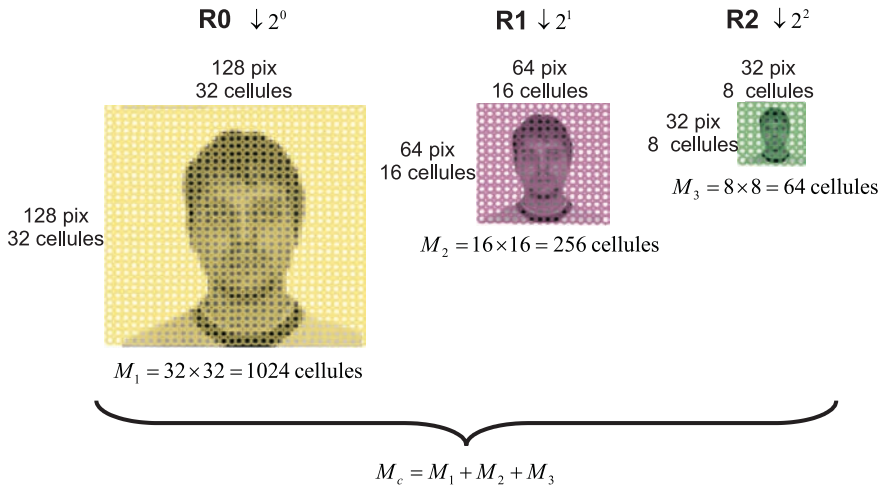


FIG. 2.5 – Exemple d'une grille multi-résolution.

– Un indice de résolution, r_m.

– Une fonction de détection. $\hat{\lambda}_m = f_m(\bar{\lambda}_m, \Sigma_{\lambda_m})$

En outre, nous avons défini l'élément de base, la « cellule », permettant d'obtenir les *parties* lors de l'apprentissage (voir §2.3). L'avantage de cet élément est qu'il permet d'observer plusieurs caractéristiques de l'image. Dans le but d'échantillonner tout l'espace dans lequel l'objet est présent, nous avons construit un maillage à partir de ces cellules. Ce maillage peut être à différents niveaux de résolution. Ayant ainsi défini tous ces éléments, nous pouvons procéder à l'obtention des *parties*, c'est-à-dire à la phase d'apprentissage.

2.3 Apprentissage

L'objectif principal dans cette phase est de fournir un modèle initial de l'objet en question, c'est-à-dire de définir chaque *partie* de l'objet, ce qui correspond à donner une valeur moyenne et une dispersion à chaque paramètre de chaque cellule.

L'apprentissage nécessite la définition de chaque *partie*, de l'importance de chacune, des détecteurs qui leur sont associés et de l'inter-relation entre elles.

Cette étape est divisée principalement en trois phases :

– la première correspond à **l'apprentissage des paramètres** dont l'objectif est de

donner des valeurs moyennes aux paramètres de chaque cellule avec leur dispersion,

– la deuxième étape correspond à **l'apprentissage sur le positionnement des cellules** (donner des valeurs moyennes des positions des cellules dans l'image avec leur dispersion), et

– la troisième étape correspond à la **réduction du modèle**, dont le rôle principal est de ne garder que ce qui est utile et négliger les paramètres peu pertinents, ainsi que les cellules non fonctionnelles.

2.3.1 Apprentissage des paramètres

On cherchera ici à déterminer, pour chaque cellule $\mathcal{C}_m$, et à partir de T exemples, la valeur moyenne et la dispersion du vecteur de paramètres ς_m : $\bar{\varsigma}_m$ et Σ_{ς_m}.

Dans un premier temps, dans cette phase d'apprentissage la collecte des statistiques est brute, c'est-à-dire que l'on va observer ce qui se passe à l'intérieur de chaque cellule lorsque l'objet d'intérêt est présent. Comme évoqué plus haut, dans chaque cellule sont observés les N opérateurs Op_n, pour $n \in [1,N]$, disposés au départ pour l'apprentissage. Chaque opérateur Op_n fournit une valeur $\zeta_{m,n}$.

Pour un exemple donné, les N opérateurs sont appliqués et on observe leur réponse à l'intérieur de chaque cellule. Comme l'on suppose l'indépendance statistique parmi les opérateurs pour une cellule donnée (mais la dépendance entre plusieurs cellules est considérée non nulle) l'observation peut être faite séparément pour chaque opérateur Op_n. Ce processus est répété pour chacun des T exemples ; on obtient N matrices, $\mathbf{OP}_n = [\mathbf{op}_{n,1}\mathbf{op}_{n,2}\dots\mathbf{op}_{n,T}]$, $n \in [1,N]$, de taille $M^c \times T$ où M^c est le nombre total de cellules et T est le nombre total d'exemples (voir figure §2.6).

On suppose que la fonction de distribution de chaque opérateur répond à une loi normale, il suffit de calculer la moyenne $\bar{\mathbf{op}}_n$ et la covariance $\Sigma_{\mathbf{op}_n}$ pour caractériser sa fonction de distribution. L'inter-relation d'un seul paramètre parmi toutes les cellules, à n'importe quel niveau de résolution, est donnée par la matrice de covariance $\Sigma_{\mathbf{op}_n}$.

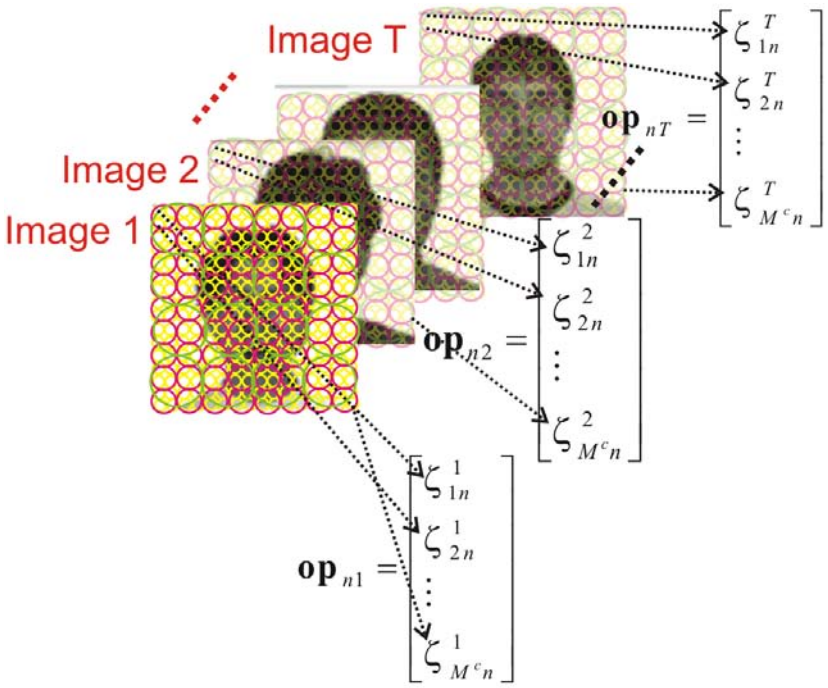


FIG. 2.6 – Collecte des statistiques pour la caractérisation des N paramètres, $\zeta_1, \zeta_2, \ldots, \zeta_N$, des M_c cellules de la grille, à partir des T exemples. $\zeta^t_{m,n}$: t-ième image, m-ième cellule et n-ième opérateur.

Donc, pour un opérateur Op_n donné, on a

$$\bar{\mathbf{op}}_n = \begin{bmatrix} \bar{\zeta}_{1,n} \\ \bar{\zeta}_{2,n} \\ \vdots \\ \bar{\zeta}_{M^c,n} \end{bmatrix} \quad \text{et} \quad \Sigma_{\mathbf{op}_n} = \begin{pmatrix} \sigma^2_{\zeta_{1,n}} & \sigma_{\zeta_{2,n}\zeta_{1,n}} & \cdots & \sigma_{\zeta_{M^c,n}\zeta_{1,n}} \\ \sigma_{\zeta_{1,n}\zeta_{2,n}} & \sigma^2_{\zeta_{2,n}} & \cdots & \\ \vdots & & \ddots & \\ \sigma_{\zeta_{1,n}\zeta_{M^c,n}} & & & \sigma^2_{\zeta_{M^c,n}} \end{pmatrix}$$

où $\bar{\mathbf{op}}_n$ correspond au vecteur moyen de paramètres de l'opérateur $n \in [1,N]$, $\bar{\zeta}_{m,n}$ correspond au paramètre moyen observé dans la cellule $m \in [1,M^c]$ issu de l'opérateur n, et $\Sigma_{\mathbf{op}_n}$ est la matrice de covariance de $\mathbf{op}_n$.

2.3.2 Apprentissage de la statistique de réponse des opérateurs

Le fait de caractériser la statistique des paramètres de chaque cellule, quand l'objet est présent, est extrêmement important, mais ce n'est pas tout. Il est aussi de grand intérêt pour nous de caractériser la statistique sur la *réponse des opérateurs*. Cette statistique

nous donne l'information concernant la « fiabilité » que peut avoir un paramètre d'une cellule donnée, en prenant en compte le fait qu'il y a des opérateurs qui ne fournissent pas toujours de l'information. Un exemple d'un tel opérateur est l'extraction de contours. Cela est pertinent pour connaître la probabilité de présence de l'objet, sachant que nous avons détecté une *partie* quelconque (voir §2.4.6) . Ainsi donc, afin de caractériser la statistique de réponse, il nous faut aussi garder le nombre de fois $R_{m,n}$ qu'un opérateur Op_n a répondu dans une cellule $\mathcal{C}_m$.

2.3.3 Positionnement des cellules

Puisqu'un objet quelconque peut être trouvé dans n'importe quelle position de l'image, à différentes échelles et avec des petites variations de point de vue (rotation 2D et 3D), il faut que le modèle de l'objet intègre ces variations pour avoir une reconnaissance invariante en position, échelle et point de vue. Donc, pour apprendre la position probable et la dispersion des cellules, deux cas se présentent selon la provenance des exemples.

2.3.3.1 Premier cas : base de données annotées

Dans une base de ce type, on a l'image où se trouve l'objet à modéliser et aussi un fichier qui indique les coordonnées de la boîte englobante de l'objet, par rapport au référentiel de l'image. Du fait que l'on connaît les coordonnées de la boîte englobante, on doit placer la grille de cellules sur cette boîte englobante. Ainsi, on peut apprendre, par simple statistique, la position moyenne des cellules, dans le référentiel image, $\bar{\mathbf{a}} = [\bar{\mathbf{a}}_1, \bar{\mathbf{a}}_2, \ldots, \bar{\mathbf{a}}_m]^t$, avec $\bar{\mathbf{a}}_m = [\bar{u}_m, \bar{v}_m]$, pour $m \in [1, M^c]$. La relation existante parmi les positions des cellules, et sa dispersion, peut être obtenue en calculant la matrice de covariance $\Sigma_{\mathbf{a}}$ de $\mathbf{a}$. L'objet peut être de taille différente dans chaque exemple et, du fait qu'on a un nombre fixe de cellules, la grille doit être adaptée en fonction de la taille de l'objet.

2.3.3.2 Deuxième cas : objets centrés et normalisés en taille

Dans ce type de base de données, on a typiquement des images de taille fixe où l'objet est centré et normalisé en taille. Dans ce cas, le référentiel est celui de l'imagette, la position absolue des cellules reste *toujours la même* : la variance sur la position de celles-ci est donc nulle.

Il est nécessaire d'apprendre les variations possibles en position, taille et échelle tenant compte du fait que l'objet peut-être situé à des positions et distancesv différentes de celles utilisées lors de l'apprentissage.

On cherche donc à caractériser la position moyenne $\bar{\mathbf{a}}$ de l'ensemble des cellules, et sa covariance dans l'image.

Transformations d'échelle, rotation et translation On considère ici qu'une transformation de similitude est définie par une rotation, une translation, et un changement d'échelle. On considérera donc un changement d'échelle s_u, s_v des axes suivi d'une rotation du repère d'angle ϕ dans le plan image source, suivie d'une translation t_u, t_v.

Ainsi, le vecteur de coordonnées $\mathbf{a}$ définissant la position des cellules dans l'image sera donné par :

$$\mathbf{a} = g(\mathbf{a}_0, s_u, s_v, \phi, t_u, t_v) = g(\mathbf{a}_0, \mathbf{t})$$

– $\mathbf{a}_0$ représente le vecteur constant de position des cellules dans l'imagette d'apprentissage qui a donc une matrice de covariance nulle,

– $\mathbf{t}$ représente le vecteur des paramètres définissant la transformation : $\mathbf{t} = [s_u, s_v, \phi, t_u, t_v]^t$.

Modèle d'apprentissage final Tenant compte de la position incertaine de l'objet, on considérera ainsi que le vecteur $\mathbf{a}$ suit une loi normale :

$$\mathbf{a} \sim \mathcal{N}(\bar{\mathbf{a}}, \Sigma_{\mathbf{a}})$$

Le vecteur moyen $\bar{\mathbf{a}}$ sera ainsi donné par :

$$\bar{\mathbf{a}} = g(\bar{\mathbf{a}}_0, \bar{\mathbf{t}})$$

La matrice de covariance $\Sigma_{\mathbf{a}}$ du vecteur $\mathbf{a}$ sera, quant à elle donnée par :

$$\Sigma_{\mathbf{a}} = \mathbf{J}_g \Sigma_{\mathbf{t}} \mathbf{J}_g^t$$

Avec :
– $\bar{\mathbf{t}}$ valeur moyenne du vecteur $\mathbf{t}$,

– $\Sigma_{\mathbf{t}}$: matrice de covariance du vecteur $\mathbf{t}$ définissant la connaissance que l'on peut accorder à $\mathbf{t}$. Si l'on considère l'indépendance totale entre les paramètres de $\mathbf{t}$, on pourra choisir $\Sigma_{\mathbf{t}}$ comme suit :

$$\Sigma_{\mathbf{t}} = \begin{pmatrix} \sigma^2_{s_u} & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{s_v} & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{\phi} & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{t_u} & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{t_v} \end{pmatrix}$$

Dans le cas où l'on considère que les deux facteurs d'échelle s_u et s_v sont identiques : $s_u = s_v$, la matrice $\Sigma_{\mathbf{t}}$ prendra la forme suivante :

$$\Sigma_{\mathbf{t}} = \begin{pmatrix} \sigma_{s_u}^2 & \sigma_{s_u}^2 & 0 & 0 & 0 \\ \sigma_{s_u}^2 & \sigma_{s_u}^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\phi}^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{t_u}^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{t_v}^2 \end{pmatrix}$$

– $\mathbf{J}_g$: matrice jacobienne de la fonction g définie par :

$$\mathbf{J}_g = \begin{pmatrix} \frac{\partial u_1}{\partial s_u} & \frac{\partial u_1}{\partial s_v} & \frac{\partial u_1}{\partial \phi} & \frac{\partial u_1}{\partial t_u} & \frac{\partial u_1}{\partial t_v} \\ \frac{\partial v_1}{\partial s_u} & \frac{\partial v_1}{\partial s_v} & \frac{\partial v_1}{\partial \phi} & \frac{\partial v_1}{\partial t_u} & \frac{\partial v_1}{\partial t_v} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial u_m}{\partial s_u} & \frac{\partial u_m}{\partial s_v} & \frac{\partial u_m}{\partial \phi} & \frac{\partial u_m}{\partial t_u} & \frac{\partial u_m}{\partial t_v} \\ \frac{\partial v_m}{\partial s_u} & \frac{\partial v_m}{\partial s_v} & \frac{\partial v_m}{\partial \phi} & \frac{\partial v_m}{\partial t_u} & \frac{\partial v_m}{\partial t_v} \end{pmatrix}$$

Cette matrice sera calculée autour de la valeur centrale $\bar{\mathbf{t}}$ du vecteur $\mathbf{t}$.

Donc, dans les deux cas présentés précédemment concernant l'apprentissage sur l'intervalle de confiance et la position des cellules, on obtient

$$\bar{\mathbf{a}} = \begin{bmatrix} \bar{\mathbf{a}}_1 \\ \bar{\mathbf{a}}_2 \\ \vdots \\ \bar{\mathbf{a}}_{M^c} \end{bmatrix} \quad \text{et} \quad \Sigma_{\mathbf{a}} = \begin{pmatrix} \Sigma_{\mathbf{a}_1} & & \cdots & \\ & \Sigma_{\mathbf{a}_2} & \cdots & \\ \vdots & & \ddots & \\ & & & \Sigma_{\mathbf{a}_{M^c}} \end{pmatrix}$$

où $\bar{\mathbf{a}}$ correspond au vecteur moyen contenant la position la plus probable de toutes les cellules, et $\Sigma_{\mathbf{a}}$ correspond à la matrice de covariance associée. Il faut remarquer que, dans la matrice de covariance $\Sigma_{\mathbf{a}}$, on a les variations possibles sur la position de chaque cellule mais aussi l'inter-relation existante parmi les positions de toutes les cellules.

Après la collecte de statistiques, le modèle général initial est

$$\bar{\mathbf{x}} = \begin{bmatrix} \bar{\varsigma}_1 \\ \bar{\varsigma}_2 \\ \vdots \\ \bar{\varsigma}_{M^c} \end{bmatrix}, \quad \Sigma_x = \begin{pmatrix} \Sigma_{\varsigma_1} & & & \cdots \\ & \Sigma_{\varsigma_2} & & \\ & & \ddots & \\ & \vdots & & \Sigma_{\varsigma_{M^c}} \end{pmatrix} \tag{2.2}$$

Pour la représentation multi-résolution d'un objet on fait typiquement la décomposition en 3 niveaux. Si par exemple, pour le niveau plus grossier on a une grille de taille

8×8, une autre de taille 16×16 pour le niveau moyen et 32×32 pour un niveau de résolution plus élevé, on aurait une grille avec 1344 cellules. Cela veut dire qu'au départ, après la collecte de statistiques, on aurait N vecteurs $\bar{\mathbf{op}}_n$ de taille 1344 et N matrices de covariances $\Sigma_{\mathbf{op}_n}$ de taille 1344×1344, plus le vecteur des coordonnées $\bar{\mathbf{a}}$ de taille 2688 et sa matrice de covariance $\Sigma_{\mathbf{a}}$ de taille 2688×2688. Il est indispensable de faire une réduction de dimensionalité afin d'obtenir un modèle plus compact.

2.3.4 Réduction de dimensionnalité

Le but ici est de simplifier le modèle initial de l'objet afin d'éliminer l'information non pertinente. Comme résultat on aura un modèle moins lourd et donc plus facile à gérer.

Lors de la caractérisation des cellules à partir d'exemples, nous pouvons constater qu'il y a des cellules qui tombent hors de l'objet et qui correspondent au fond, ou simplement des cellules qui sont placées sur une région de l'image qui n'est pas du tout descriptive pour une classe d'objets déterminée. Toutes ces cellules ne sont pas utiles et peuvent être éliminées sans problème. Cette élimination des cellules n'est cependant pas la seule possibilité pour simplifier le modèle ; on verra qu'il est aussi possible de réduire la dimensionnalité en éliminant des paramètres qui ne sont pas descriptifs d'une région échantillonnée par une cellule quelconque, ainsi que des paramètres issus des opérateurs qui ne répondent pas souvent. Deux techniques seront donc décrites : l'élimination des paramètres non descriptifs et l'élimination des cellules non fonctionnelles.

2.3.4.1 Élimination des paramètres non descriptifs

Lors de la collecte de statistiques nous pouvons constater que, pour une cellule donnée, il y a des paramètres qui ne décrivent pas l'objet. Ce manque de description peut être dû à deux raisons : soit parce que l'information qu'il donne appartient au fond ou à une région qui n'est pas descriptive pour une classe d'objets, soit parce que l'opérateur ne fournit pas d'information (exemple la direction du contour sur une région où il n'y a pas de contour). La *connaissance* qu'a le processus pour un objet déterminé, est contenue dans la matrice de covariance ; c'est là que l'on cherchera l'information nécessaire pour guider le processus de reconnaissance.

Un critère important, pour l'élimination des paramètres, peut être basé sur le coefficient de corrélation de ce paramètre dans une cellule avec le même paramètre pour d'autres cellules de la grille. Si le paramètre dans cette cellule n'est corrélé avec aucune autre cellule dans la grille ou si le coefficient de corrélation est faible, il peut être éliminé de cette cellule et de toutes les autres qui en dépendent.

Un autre critère est celui basé sur la variance d'un paramètre pour une cellule déterminée. Si la variance est grande, et s'il a un faible coefficient de corrélation, on le considère comme du bruit et on l'élimine de la cellule. Un bon choix peut être de garder le paramètre dans une cellule s'il est corrélé avec au moins une autre cellule avec un coefficient de corrélation supérieur à un seuil, 95% par exemple.

Il faut remarquer que, en plus de faire la réduction de dimensionalité, on construit des détecteurs « spécialisés » pour chaque partie.

Pour faciliter l'élimination des paramètres non discriminants pour une cellule déterminée, il convient de représenter le modèle sous la forme suivante (les opérateurs étant supposés décorrélés) :

$$\bar{\mathbf{x}}^p = \begin{bmatrix} \bar{\mathbf{op}}_1 \\ \bar{\mathbf{op}}_2 \\ \vdots \\ \bar{\mathbf{op}}_N \\ \bar{\mathbf{a}} \end{bmatrix}, \quad \Sigma_{\mathbf{x}^p} = \begin{pmatrix} \Sigma_{\mathbf{op}_1} & & & & \\ & \Sigma_{\mathbf{op}_2} & & \mathbf{0} & \\ & & \ddots & & \\ & \mathbf{0} & & \Sigma_{\mathbf{op}_N} & \\ & & & & \Sigma_{\mathbf{a}} \end{pmatrix} \tag{2.3}$$

avec

$$\bar{\mathbf{a}} = \begin{bmatrix} \bar{\mathbf{a}_1} \\ \bar{\mathbf{a}_2} \\ \vdots \\ \bar{\mathbf{a}_{M^c}} \end{bmatrix} \text{ et } \quad \Sigma_{\mathbf{a}} = \begin{pmatrix} \Sigma_{\mathbf{a}_1} & & \cdots & \\ & \Sigma_{\mathbf{a}_2} & & \\ \vdots & & \ddots & \\ & & & \Sigma_{\mathbf{a}_{M^c}} \end{pmatrix}$$

Pour chaque opérateur Op_n, on peut calculer la matrice de coefficients de corrélation donnée par

$$R_{\mathbf{op}_n} = \begin{pmatrix} r^2_{\zeta_{1,n}} & r_{\zeta_{1,n}\zeta_{2,n}} & \cdots & r_{\zeta_{1,n}\zeta_{M^c,n}} \\ r_{\zeta_{2,n}\zeta_{1,n}} & r^2_{\zeta_{2,n}} & \cdots & r_{\zeta_{2,n}\zeta_{M^c,n}} \\ \vdots & \vdots & \ddots & \vdots \\ r_{\zeta_{M^c,n}\zeta_{1,n}} & r_{\zeta_{M^c,n}\zeta_{2,n}} & \cdots & r^2_{\zeta_{M^c,n}} \end{pmatrix}$$

avec $-1 \leq r_{i,j} \leq 1$. Ce coefficient de corrélation peut être calculé à partir de la matrice de covariance par la relation suivante :

$$r_{\zeta_i,\zeta_j} = \frac{\sigma_{\zeta_i,\zeta_j}}{\sqrt{\sigma_{\zeta_i,\zeta_i}\sigma_{\zeta_j,\zeta_j}}} \tag{2.4}$$

Les paramètres qui ont un intérêt pour nous sont ceux qui obéissent à $|r_{i,j}| \geq 0,95$, les autres peuvent être éliminés. Ainsi, on supprimera l'influence de ces opérateurs au sein de certaines cellules.

Ainsi, si

$$\mathbf{o\bar{p}}_n = \begin{pmatrix} \bar{\zeta}_{1,n} \\ \bar{\zeta}_{2,n} \\ \bar{\zeta}_{3,n} \\ \vdots \\ \bar{\zeta}_{M^c,n} \end{pmatrix}$$

on ne gardera que

$$\mathbf{o\bar{p}}_n^p = \left.\begin{pmatrix} \bar{\zeta}_{1,n} \\ \bar{\zeta}_{2,n} \\ \bar{\zeta}_{3,n} \\ \vdots \\ \bar{\zeta}_{M^c,n} \end{pmatrix}\right\} M_n \text{ paramètres restants}$$

Pour cet exemple, le barré indique les paramètres éliminés.

Donc, après élimination on obtient le modèle réduit suivant,

$$\mathbf{X}_r^p \sim \begin{cases} \mathcal{N}(\mathbf{o\bar{p}}_1^p, \Sigma_{\mathbf{op}_1^p}), \\ \mathcal{N}(\mathbf{o\bar{p}}_2^p, \Sigma_{\mathbf{op}_2^p}), \\ \vdots \\ \mathcal{N}(\mathbf{o\bar{p}}_N^p, \Sigma_{\mathbf{op}_N^p}), \\ \mathcal{N}(\bar{\mathbf{a}}, \Sigma_{\mathbf{a}}). \end{cases}$$

où $\mathbf{o\bar{p}}_n^p$ est le vecteur réduit avec une taille $M_n \leq M^c$, et la matrice de covariance $\Sigma_{\mathbf{op}_n^p}$ a une taille $M_n \times M_n$. L'élimination d'un paramètre du vecteur $\mathbf{o\bar{p}}_n^p$ correspond à ne pas faire l'observation de ce paramètre dans la cellule correspondante. Par exemple, si le paramètre i du vecteur $\mathbf{o\bar{p}}_n^p$, $\bar{\zeta}_{i,n}$, est éliminé, l'opérateur Op_n ne sera pas pris en compte au moment de faire la détection de la *partie* Λ_i (dans le cas hypothétique où la cellule $\mathcal{C}_i$ puisse devenir une *partie* lors de l'élimination de cellules non fonctionnelles

(voir §2.3.4.3)). Cela revient à inhiber le module correspondant à l'opérateur Op_n dans le traitement primaire (la région de l'image n'est pas traitée par cet opérateur). Ainsi, lors de la phase de reconnaissance, on sera capable de faire la **focalisation dans l'espace des opérateurs** en fonction des besoins.

2.3.4.2 Modèle alternatif après élimination des paramètres

Pour des raisons pratiques, on représente le modèle réduit avec l'organisation originale. Donc, pour chaque cellule on a

$$\bar{\varsigma}_m^p = \begin{bmatrix} \bar{\zeta}'_{m,1} \\ \bar{\zeta}'_{m,2} \\ \vdots \\ \bar{\zeta}'_{m,N_m} \\ \mathbf{a}'_m \end{bmatrix}, \quad \Sigma_{\varsigma_m^p} = \begin{pmatrix} \sigma^2_{\zeta'_{m,1}} & & & \\ & \sigma^2_{\zeta'_{m,2}} & & \mathbf{0} \\ & & \ddots & \\ & \mathbf{0} & & \sigma^2_{\zeta'_{m,N_m}} \\ & & & \Sigma_{\mathbf{a}'_m} \end{pmatrix}$$

donc, le modèle devient

$$\bar{\mathbf{x}}_r = \begin{bmatrix} \bar{\varsigma}_1^p \\ \bar{\varsigma}_2^p \\ \vdots \\ \bar{\varsigma}_M^p \end{bmatrix}, \quad \Sigma_{\mathbf{x}_r} = \begin{pmatrix} \Sigma_{\varsigma_1^p} & & & \cdots \\ & \Sigma_{\varsigma_2^p} & & \\ & & \ddots & \\ & \vdots & & \Sigma_{\varsigma_M^p} \end{pmatrix}$$

Il faut remarquer que, après l'élimination des paramètres, chaque cellule a un nombre de paramètres $N_m \leq N$ et que $\Sigma_{\mathbf{x}_r}$ n'est pas une matrice diagonale.

2.3.4.3 Élimination des cellules non fonctionnelles : obtention des *parties* Λ

Dès l'élimination des paramètres non descripteurs de l'objet à l'intérieur de certaines cellules, on s'aperçoit rapidement qu'il peut y avoir des cellules sans aucun paramètre pertinent. Dans ce cas particulier, toutes les cellules où il n'y a aucun paramètre $\zeta'_{i,j}$, peuvent être éliminées de la grille.

Après l'élimination des cellules non fonctionnelles, le modèle final réduit est :

$$\bar{\mathbf{x}}_f = \begin{bmatrix} \bar{\lambda}_1 \\ \bar{\lambda}_2 \\ \vdots \\ \bar{\lambda}_{M^p} \end{bmatrix}, \quad \Sigma_{\mathbf{x}_f} = \begin{pmatrix} \Sigma_{\lambda_1} & & & \cdots \\ & \Sigma_{\lambda_2} & & \\ & & \ddots & \\ & \vdots & & \Sigma_{\lambda_{M^p}} \end{pmatrix}$$

avec :

$$\bar{\lambda}_m = \begin{bmatrix} \bar{\zeta'}_{m,1} \\ \bar{\zeta'}_{m,2} \\ \vdots \\ \bar{\zeta'}_{m,N_m} \\ \mathbf{a}'_m \end{bmatrix}, \quad \Sigma_{\lambda_m} = \begin{pmatrix} \sigma^2_{\zeta'_{m,1}} & & & \\ & \sigma^2_{\zeta'_{m,2}} & & \mathbf{0} \\ & & \ddots & \\ & \mathbf{0} & & \sigma^2_{\zeta'_{m,N_m}} \\ & & & & \Sigma_{\mathbf{a}'_m} \end{pmatrix}$$

avec $m \in [1, M^p \leq M^c]$. Rappelons que le vecteur λ_m, correspond au vecteur de paramètres de chaque *partie* Λ_m. Comme décrit dans le paragraphe §2.2.2, chaque *partie* Λ_m décrit une **région locale** de l'objet à partir d'un sous ensemble N_m de paramètres du nombre total des N paramètres.

Pour donner une idée de l'effet de la réduction de dimensionalité, des expérimentations montrent que, pour un exemple donné où il y a 1344 cellules au départ, il en reste environ 150 après la réduction.

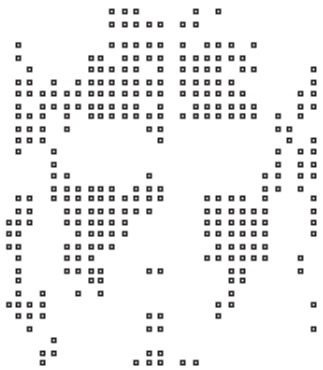


FIG. 2.8 – Exemple du modèle du visage après la réduction de dimensionnalité. Dans cet exemple, seulement l'aspect géométrique est affiché (le positionnement des cellules résultantes).

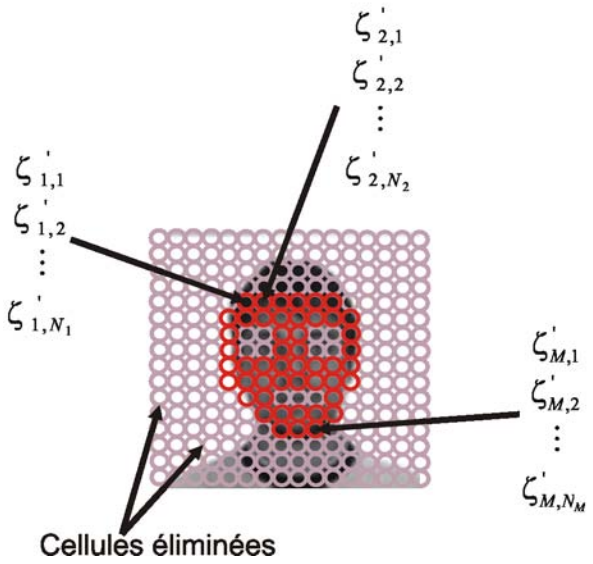


FIG. 2.7 – A titre d'exemple, on montre les cellules restantes après l'élimination de cellules non fonctionnelles. En rouge sont affichés les M^p *parties* qui composent le modèle de l'objet.

2.3.5 Bilan sur l'apprentissage

Trois étapes concernant l'apprentissage de l'objet ont été décrites :

1. La caractérisation de paramètres : obtention des valeurs moyennes et de la dispersion des paramètres,

2. le positionnement des cellules : obtention des valeurs moyennes et de la dispersion concernant la position des cellules,

3. et la réduction de dimensionnalité : obtention des M^p parties en éliminant les paramètres non descriptifs dans chaque cellule, et des cellules non fonctionnelles de la grille de M^c cellules.

En résumé, après l'apprentissage on dispose de :

– Un modèle $X \sim \mathcal{N}(\bar{\mathbf{x}}, \Sigma_{\mathbf{x}})_{obj}$ de l'objet contenant M^p *parties* Λ_m, avec $m \in [1, M^p \leq M^c]$, étant des cellules caractérisées par

  – un vecteur de paramètres $\lambda_m = [\zeta'_{m,1}, \zeta'_{m,2}, \ldots, \zeta'_{m,N_m}, \mathbf{a}'_m]$, avec $N_m \leq N$,

– une matrice de covariance diagonale avec des éléments dans la diagonale principale $\Sigma_{\lambda_m} = (\sigma^2_{\zeta'_{m,1}}, \sigma^2_{\zeta'_{m,2}}, \ldots, \sigma^2_{\zeta'_{m,N_m}}, \Sigma_{\mathbf{a}'_m})$ indiquant leur dispersion, où $\Sigma_{\mathbf{a}'_m}$ correspond à la matrice de covariance des coordonnées de la cellule m.

– M^p fonctions de détection $\hat{\lambda}_m = f_m(\bar{\lambda}_m, \Sigma_{\lambda_m})$ associées à chacune des *parties* Λ_m. La fonction de détection f_m, associée à la *partie* Λ_m, considérera la *partie* Λ_m reconnue si

$$d_m = \sqrt{(\bar{\lambda}_m - \hat{\lambda}_m)\Sigma^{-1}_{\lambda_m}(\bar{\lambda}_m - \hat{\lambda}_m)^t} \leq s$$

voir §2.4.4.2 pour plus de détails.

Dans la section suivante on expose la stratégie d'analyse de chaque *partie* en vue de la reconnaissance de l'objet, et comment la reconnaissance d'une *partie* permet d'affiner l'espace de recherche des autres *parties* (processus de focalisation).

2.4 Stratégie de Reconnaissance

Dans un premier temps, nous avons traité la notion de « cellule » définie comme une entité dont le but principal est de permettre l'observation de plusieurs caractéristiques décrivant l'objet. Ces informations sont le résultat de l'analyse d'opérateurs simples comme : la détection des contours, l'analyse de la couleur, l'entropie, etc. Afin de représenter l'apparence globale de l'objet, nous avons utilisé une grille multi-résolution de cellules. Après la phase d'apprentissage, nous disposons d'un modèle de l'objet qui est composé d'un ensemble de *parties* Λ_m, $m \in [1, M^p]$. Chaque *partie* correspond à une zone dans l'image (groupe de pixels) qui capturent la structure locale de l'objet. Les dépendances statistiques entre chaque *partie* de l'objet, forment l'apparence globale. Nous avons donc maintenant la valeur moyenne et la dispersion de l'ensemble des paramètres qui décrivent chacune des ces *parties*, ainsi que la valeur moyenne et la dispersion concernant la position de ces *parties*.

Comme indiqué auparavant, l'utilisation d'un mécanisme de focalisation pour la reconnaissance d'objets est fortement liée à la nécessité d'optimiser le processus de reconnaissance. Cela peut être réalisé dans la mesure où l'on dispose d'indices donnant la position possible de l'objet, à partir des observations locales. Ainsi, une recherche exhaustive et non pertinente, partout dans l'image en cherchant toutes les parties à la fois sans a priori, peut être évitée.

La manière dont l'algorithme doit effectuer la recherche de ces parties, afin de « mieux guider » vers l'objet en question, est complètement définie par la stratégie de reconnaissance choisie. On évitera ainsi que cette stratégie consiste en un simple balayage dans

l'image. C'est le but principal de cette phase. Dans notre cas, « mieux guider » implique la facilité avec laquelle l'algorithme pourra converger vers l'objet recherché, avec le moins d'itérations possibles.
Au moment de définir une stratégie, les points suivants sont pris en compte :

- tirer profit de la dépendance statistique entre les *parties* de l'objet,
- la hiérarchie pour la focalisation : par quelle partie faut-il commencer ? Et pour quelle raison ? (rapport avec la convergence de l'algorithme),
- besoin de réduire les candidats à partir des observations effectuées : notion d'adaptabilité.

Par ailleurs, d'autres problèmes apparaissent : que se passe-t-il quand une *partie* n'est pas là ?, et que se passe-t-il quand la *partie* choisie n'a pas été une bonne hypothèse ? (recherche séquentielle ou récursive ?)

Concernant la hiérarchie pour la focalisation Vis-à-vis de la focalisation, deux principales questions se posent : quelle *partie* peut amener au plus vite à la reconnaissance de l'objet recherché ? Et, quels sont les critères pour lesquels cette *partie* a été choisie ? On verra que les critères de recherche peuvent être basés non seulement sur la nature de la primitive (paramètres qui la composent) mais aussi sur d'autres facteurs comme la capacité de discrimination de cette *partie* ou le coût pour la calculer. Tous ces aspects seront discutés plus loin au moment de définir cette hiérarchie.

Concernant l'adaptabilité Le besoin d'avoir un système adaptatif, quand il s'agit d'un système de perception, est bien connu. Cette adaptabilité est traduite par le fait qu'une observation réalisée, sur un contexte donné, peut affecter et contraindre la façon dont l'observation suivante sera faite. On verra que cette notion d'adaptabilité peut aussi jouer un rôle assez important sur le processus de focalisation.

2.4.1 Principe de la stratégie utilisée

Supposons que l'on ait choisi la *partie* Λ_m qui semble être très discriminante de l'objet à reconnaître. Cela pourrait être, par exemple une *partie* correspondant à l'œil gauche d'un visage. Le point suivant est de réaliser la détection de cette partie. Pour ce faire, les N_m opérateurs de la *partie* Λ_m sélectionnée concernée sont appliqués dans une région de l'image. Cette dernière est centrée sur les coordonnées $\bar{\mathbf{a}}_m$ de Λ_m, dont la taille est définie par la matrice de covariance $\Sigma_{\mathbf{a}_m}$ (région d'intérêt de la *partie*). Si le vecteur $\hat{\lambda}_m$, issu de la

fonction de détection, en sortie des N_m opérateurs est jugé correct, la *partie* est considérée reconnue.

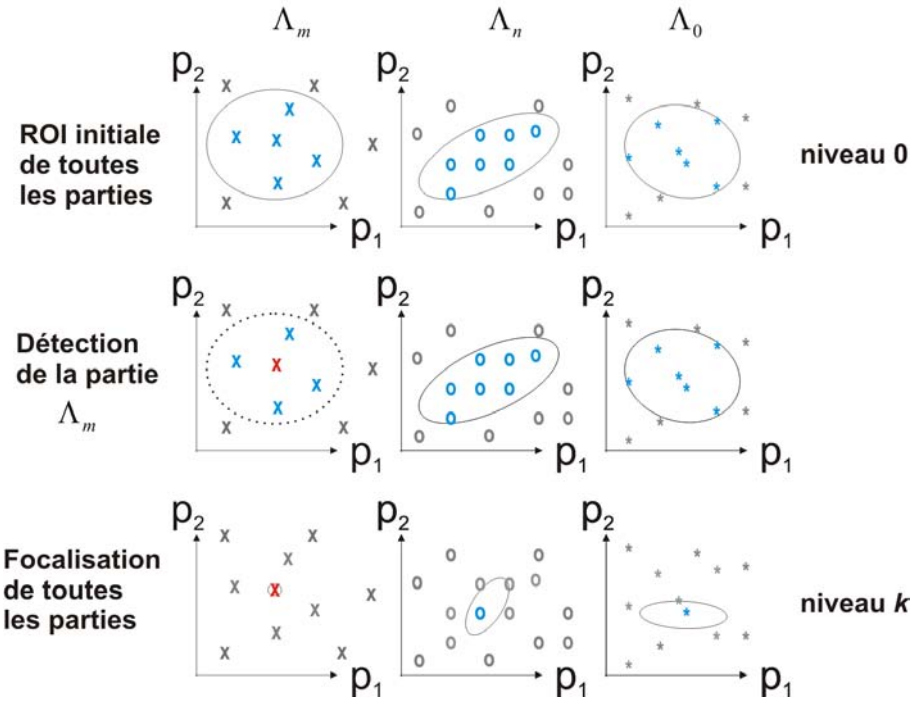


FIG. 2.9 – Principe de focalisation

Dans le cas où un nombre suffisant de *parties* est détecté, l'objet est supposé reconnu. Dans le cas contraire, l'observation $\hat{\lambda}_m$, du fait du lien statistique fort entre tous les autres paramètres des autres *parties* (résumé dans la matrice de covariance $\Sigma_{\mathbf{x}}$), permettra de préciser la position et les paramètres des autres *parties* à détecter (voir figure 2.9). Ce processus (génération d'hypothèses, détection, remise à jour) est réitéré jusqu'à ce qu'un critère de reconnaissance soit atteint.

Sur la figure 2.10 on montre le schéma qui décrit, d'une façon générale, la stratégie développée pour le contrôle du processus de reconnaissance. Cette stratégie englobe plusieurs étapes qui vont, de la sélection de la *partie* à reconnaître, jusqu'à la mise à jour du modèle (focalisation) ; en passant par d'autres étapes intermédiaires comme la détection et la prise de décision. Dans la suite, chaque étape est expliquée en détail.

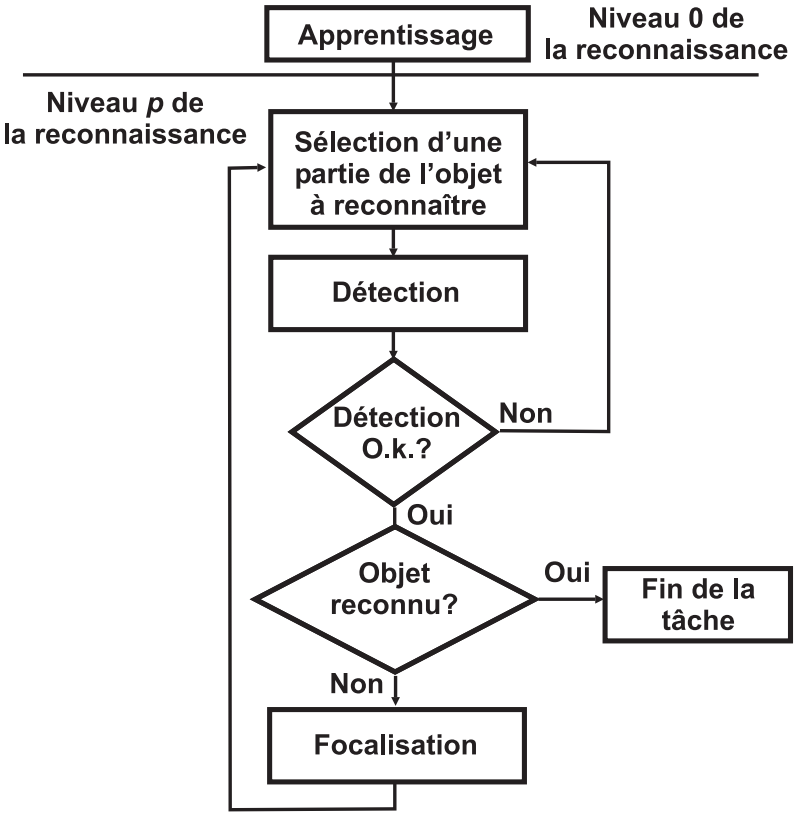


FIG. 2.10 – Stratégie du système de reconnaissance.

2.4.2 Préparation et initialisation : niveau zéro

Une fois l'apprentissage fait, le résultat obtenu correspond au modèle moyen initial de l'objet $\mathcal{N}(\bar{\mathbf{x}}, \Sigma_{\mathbf{x}})_0$, dont l'indice 0 correspond au niveau zéro du processus. Le modèle initial définit pour chaque *partie*, la valeur moyenne de ses paramètres (avec notamment sa position moyenne dans l'image) et leur dispersion (ce qui permet de définir la zone d'intérêt de chaque *partie* dans l'image mais aussi la dispersion des autres paramètres comme le niveau de gris, l'orientation du contour, ...). C'est à partir de là que commence la première recherche de l'algorithme, soit le niveau de profondeur $k = 0$ du processus de reconnaissance. Ensuite l'algorithme récursif commence.

2.4.3 Génération des hypothèses (sélection de *parties*)

Effectuer la génération d'hypothèses revient à faire la hiérarchisation pour la sélection des *parties* composant l'objet. On connaît a priori les facteurs qu'il faut prendre en considération au moment de choisir une *partie,* soit :

– les indices d'attention visuelle donnés par la sortie des N_m opérateurs Op_n,

– la pertinence d'une partie (variance, corrélation, résolution),

– la capacité de générer le moins de candidats possibles,

– le temps de calcul.

Donc, on voit que les critères pour la sélection de la *meilleure partie* sont basés sur l'expérience obtenue à partir de l'analyse statistique effectuée dans la phase d'apprentissage, et d'autres critères imposés comme ceux liés au temps de calcul prévisible (en fonction des opérateurs et de la taille de la zone d'intérêt).

Il faut remarquer que le critère de sélection ne dépend pas que d'un seul facteur mais d'un ensemble. Ainsi, on est obligé de regrouper tous les facteurs ensemble et de trouver une règle unique qui correspond au meilleur compromis parmi tous les facteurs. Cependant, dans le futur on pourrait être capable de faire la collecte statistique sur le parcours type du processus et trouver, grâce à l'expérience, le meilleur parcours probable parmi les plus pertinents. Ce paramètre pourra être pris en compte pour la future génération d'hypothèses. Ce type d'apprentissage est effectué pendant le processus de reconnaissance donc, il correspond à un apprentissage en ligne. Dans son état actuel, l'algorithme ne fait que l'apprentissage hors ligne.

Dans notre cas, la partie la plus pertinente sera celle qui a le plus d'indices d'attention visuelle, la variance moindre, l'indice de corrélation maximum et le temps de calcul probable le plus faible (ce qui conduira l'algorithme à privilégier les *parties* de résolution la plus basse).

Comme décrit au premier chapitre dans la section §1.2, la théorie BHF postule que le processus de perception peut être guidé initialement à partir de composantes de basse fréquence, pour après analyser le détail correspondant aux composantes de haute fréquence. A la différence de ce que postule cette théorie, cette méthode pour la sélection des *parties* permet de guider le processus de reconnaissance d'une manière non séquentielle. L'échelle d'analyse sélectionnée est donc la plus pertinente selon l'état courant du processus. Par exemple, en fonction de la disponibilité du temps, il peut être plus pertinent de commencer à rechercher une *partie très discriminante* appartenant aux hautes fréquences et non un composant de basse fréquence moins discriminant.

En résumé, la procédure est la suivante : à l'état k du processus on cherche, parmi toutes les parties qui ne sont pas encore sélectionnées, la meilleure selon ce critère. En fait, elle correspond à la partie la plus pertinente dans l'état *courant* k du processus. Si l'on

insiste sur le caractère *courant*, c'est qu'il existe une dépendance évidente entre le critère de sélection et l'état courant k du modèle $\mathcal{N}(\bar{\mathbf{x}}, \Sigma_{\mathbf{x}})_k$, notamment en ce qui concerne la recherche de la partie avec la moindre variance (on verra plus loin, dans §2.4.5, que la mise à jour du modèle entraîne un ajustement des paramètres de chaque *partie*).

2.4.4 Phase de détection des *parties*

2.4.4.1 Zone d'intérêt

Avant de réaliser la détection d'une *partie* Λ_m quelconque, il faut d'abord définir la région d'intérêt dans l'image où cette *partie* peut être située potentiellement. En adéquation avec ce qui a été cité précédemment, le vecteur des paramètres de chaque *partie* est composé de plusieurs caractéristiques $\zeta_{m,n}$, où $n \in [1, N_m]$ décrivant une *partie* de l'objet (voir §2.2.2), ainsi que les coordonnées $\mathbf{a}_m$ concernant sa position. Ce vecteur est caractérisé par une fonction de densité de probabilité modélisée selon une loi normale. Donc, la région d'intérêt dans l'espace image peut être obtenue à partir de la valeur moyenne sur la position de chaque *partie* Λ_m, $\bar{\mathbf{a}}_m$, et de sa matrice de covariance associée $\Sigma_{\mathbf{a}_m}$. Jusqu'à présent, seule a été faite la délimitation de la région d'intérêt pour l'espace géométrique. Cependant cette région d'intérêt doit être définie dans l'espace des caractéristiques.

2.4.4.2 Détection

Quand on veut réaliser la détection d'une *partie* Λ_m, on doit premièrement observer quels types de paramètres composent cette *partie*. Puis, la région d'intérêt définie précédemment est traitée par le sous ensemble des N_m opérateurs. On peut noter qu'à ce stade il y a déjà une optimisation du processus de reconnaissance, principalement en ce qui concerne le traitement primaire. Ainsi, non seulement on délimite la région dans l'espace image où les opérateurs doivent agir, mais aussi le nombre d'opérateurs (potentiellement $N_m \leq N$).

Prendre une décision sur la zone d'intérêt dans laquelle une partie a été reconnue ou non, revient à définir des bornes sur la fonction de distribution. Typiquement, pour une *partie* Λ_m, de vecteur de paramètres moyen $\bar{\lambda}_m$ et de covariance Σ_{λ_m}, on a L_m vecteurs candidats $\hat{\lambda}_l$; $l \in [1, L_m]$. Un candidat $\hat{\lambda}_l$ sera retenu si

$$d_l = \sqrt{(\bar{\lambda}_m - \hat{\lambda}_l)\Sigma_{\lambda_m}^{-1}(\bar{\lambda}_m - \hat{\lambda}_l)^t} \leq s$$

Cette relation (distance de Mahalanobis) détermine si le vecteur $\hat{\lambda}_l$ est à l'intérieur de l'ellipsoïde ou non. L'indice $l \in [1, L_m]$ indique le nombre de candidats possibles pouvant correspondre à λ_m, où L_m est le nombre total de candidats. Dans notre cas, le paramètre s a été fixé à deux (compromis entre risque de non détection et ambiguïté de détection).

Dans le cas où il y a au moins une détection parmi tous les candidats $\hat{\lambda}_l$, le détecteur retourne celle qui est la plus proche de la valeur moyenne $\bar{\lambda}_m$ (celle pour laquelle d_l est minimale).

Dans cette approche, à la différence des travaux de Itti et al. [50], où le mécanisme d'attention est guidé seulement par les régions de saillance (bottom-up), ici le mécanisme d'attention est guidé aussi par le modèle (top-down). C'est le modèle qui, d'après la configuration probable de ses *parties*, non seulement indique la région sur sa position probable mais aussi quel type d'information (orientation, couleur, texture, etc.) est accessible par le haut niveau pour un traitement futur. Pour leur part, les données filtrées serviront d'indices pour attirer l'attention et focaliser la recherche d'une *partie* (bottom-up). Ainsi, on observe que le guidage par les indices est aussi possible.

A la différence des cartes de saillance, où l'information est extraite d'une façon indépendante des besoins, ici on construit une espèce de « carte de saillance dynamique » où les régions extraites dépendent de l'état courant du processus. Cependant, l'information saillante n'a ici d'intérêt que pour l'étape actuelle du processus de reconnaissance. La manière dont les points de saillance seront évalués comme étant la *partie* Λ_m recherchée, sera sous la responsabilité de la stratégie de reconnaissance. Si la *partie* détectée ne correspond pas à la *partie* de l'objet en question, un autre candidat (un autre point d'intérêt de notre carte de saillance) sera évalué, et ainsi successivement jusqu'au balayage total de cette carte (si besoin). Il faut remarquer que la « carte de saillance dynamique » construite avec notre approche n'est pas une carte correspondant à toute l'image analysée, mais seulement à l'imagette correspondant à la région d'intérêt.

La figure 2.11 montre l'ensemble des hypothèses (Λ_m, $m \in [1, M^p]$) représenté par une structure d'arbre de recherche. Λ_m, $m \in [1, M^p]$ correspondent aux *parties* de l'objet à rechercher. $D_m(\cdot)$ est la fonction de détection correspondant à Λ_m. $d_{m,l}$, où $l \in [1, L_m]$ est la détection lors de la recherche de Λ_m. k correspond au niveau de profondeur dans l'arbre de recherche.

A la profondeur de recherche k, une hypothèse est émise quant au choix d'une *partie* à rechercher. La détection de cette *partie* est tentée par le détecteur associé ($D_m(\cdot)$) qui retourne un ensemble de détections candidates $d_{m,l}$, où $l \in [1, L_m]$. La détection candidate la plus proche du vecteur moyen $\bar{\lambda}_m$ est retenue. Elle constitue l'observation. Ceci conduit à émettre une nouvelle hypothèse de choix d'une autre *partie* Λ_i. Cette seconde hypothèse est la « fille » de la première. Le processus se trouve alors à la profondeur de recherche $k+1$. Ce processus (génération d'hypothèse $\rightarrow$ détection $\rightarrow$ validation) est réitéré jusqu'à ce qu'un critère d'arrêt soit atteint. Ce point est détaillé dans la suite du mémoire.

La figure 2.11 doit être lue de bas en haut puis de droite à gauche. La première chaîne

d'hypothèses est constituée des *parties* Λ_1, Λ_2, et Λ_3. Dans cet exemple Λ_3 n'a aucune détection candidate ce qui entraîne l'invalidation de la partie Λ_3. Une nouvelle hypothèse Λ_4 est générée à partir de Λ_2. Comme pour Λ_3, Λ_4 n'a pas de candidats. En conséquence, la branche correspondant à l'hypothèse mère Λ_1 est considérée comme morte (tracé estompé sur la figure 2.11). Le processus génère alors une nouvelle hypothèse mère Λ_2 et il est réitéré. La branche finalement valide est tracée en traits plus vifs.

Ce schéma simplifié du parcours de l'arbre ne correspond pas tout à fait au déroulement réel du processus. En fait chaque hypothèse est générée en fonction du besoin courant de la reconnaissance qui n'est pas déterministe.

L'apparition de deux hypothèses distinctes correspondant à la même *partie* (par exemple Λ_2) peut surprendre. Ceci n'a en effet de sens que si le principe de focalisation exposé dans la section suivante (§2.4.5) est mis en œuvre.

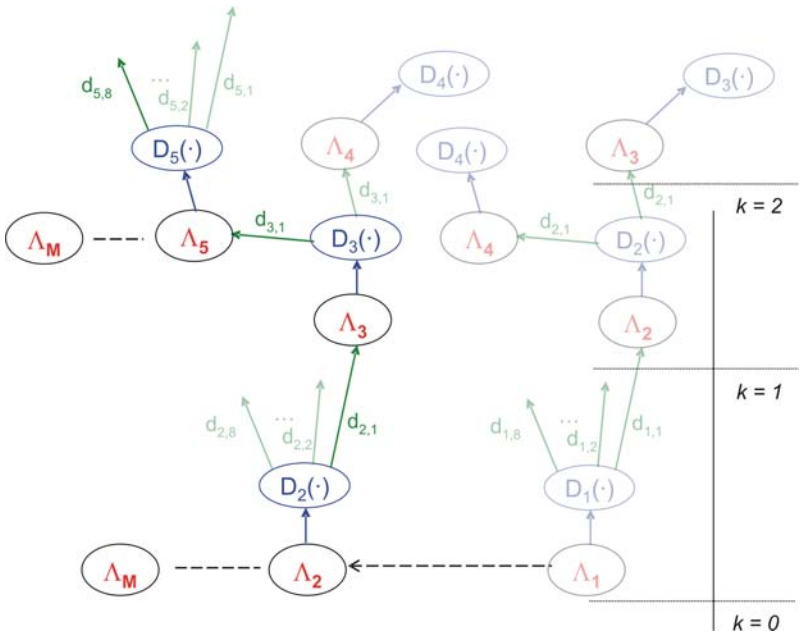


FIG. 2.11 – Représentation de l'ensemble des hypothèses par une structure d'arbre de décision.

2.4.5 Focalisation dans l'espace des caractéristiques : mise à jour du modèle par filtrage de Kalman

Dans le cas général, la région d'intérêt de Λ_m sera déduite de la matrice de covariance $\Sigma_{\mathbf{x}}$ comme expliqué en §2.4.4. Du fait du lien statistique entre ses différents paramètres pris en compte dans cette matrice de covariance, il est possible, à l'aide d'une simple détection d'une des parties Λ_m, de réajuster non seulement la position des autres parties mais aussi de préciser l'intervalle de variation des paramètres de celles-ci en fonction de cette détection. En effet, si l'on considère l'observation $\hat{\lambda}_m = [\hat{\zeta}_{m,1}, \ldots, \hat{\zeta}_{m,n}, \hat{\mathbf{a}}_m]^t$ de $\lambda_m = [\zeta_{m,1}, \ldots, \zeta_{m,n}, \mathbf{a}_m]$, pour $n \in [1, N_m]$, on pourra écrire les équations d'état suivantes :

$$\begin{aligned} \bar{\mathbf{x}}(k+1) &= \mathbf{A}\bar{\mathbf{x}}(k) + \mathbf{B}U(k) + v(k) && (2.5) \\ \bar{\mathbf{Y}}(k) = \hat{\lambda}_m &= \mathbf{H}\bar{\mathbf{x}}(k) + w(k) && (2.6) \end{aligned}$$

Dans notre cas la première équation (2.5) (équation d'évolution) n'a pas d'intérêt : le vecteur $\bar{\mathbf{x}}$ n'évolue pas d'une détection à l'autre, donc $\mathbf{A} = \mathbf{I}$ (matrice identité). On pourrait néanmoins, dans le cadre d'un processus dynamique prendre en compte l'évolution des paramètres et de la position des *parties* durant le processus de reconnaissance. Ce ne sera pas le cas ici. De même la matrice de covariance $\mathbf{Q}$ du vecteur de bruit d'évolution v sera nulle.

Seulement dans le cas où il y a eu une détection, il est nécessaire de mettre à jour le modèle $\mathcal{N}(\bar{\mathbf{x}}, \Sigma_{\mathbf{x}})_k$ (où k indique l'état courant du processus de reconnaissance) afin de donner des a priori pour mieux cerner la recherche des *parties* futures. Cette mise à jour est effectuée en utilisant un filtre de Kalman dégénéré (sans évolution dynamique) donné par :

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{x}(k) + \mathbf{K}[\hat{\lambda}_m - \hat{\lambda}_m(k)] && (2.7) \\ \Sigma_{\mathbf{x}}(k+1) &= \Sigma_{\mathbf{x}}(k) - \mathbf{K}\mathbf{H}\Sigma_{\mathbf{x}}(k) \\ &\text{avec} \\ \mathbf{K} &= \Sigma_{\mathbf{x}}(k)\mathbf{H}^t[\mathbf{H}\Sigma_{\mathbf{x}}(k)\mathbf{H}^t + \Sigma_{\hat{\lambda}_m}]^{-1} \end{aligned}$$

- $\mathbf{H}$ telle que $\hat{\lambda}_m = \mathbf{H}\mathbf{x}$: lien entre $\hat{\lambda}_m$ et la totalité du vecteur $\mathbf{x}$.
- $\hat{\lambda}_m$: estimée du vecteur de paramètres de la *partie* Λ_m,
- $\Sigma_{\hat{\lambda}_m}$: matrice de covariance des paramètres de Λ_m.

Dans la pratique, étant donnée l'indépendance entre les N_m paramètres (plus la position), on appliquera successivement $N_m + 1$ filtres (ce qui conduit à une complexité bien moindre).

2.4.6 Phase de décision

2.4.6.1 Caractérisation probabiliste

Dans la phase de reconnaissance, il est nécessaire de savoir à partir de combien de détections des *parties* on peut dire que l'objet est reconnu ; ou d'avoir un indice sur la pertinence de continuer à chercher d'autres *parties*, selon l'état du processus. Comme décrit dans §2.4.3, chaque *partie* est caractérisée par un poids indiquant l'importance de cette *partie* par rapport à l'objet. Il est donc bien évident que le critère de décision doit dépendre de cet indice d'importance mais pas seulement : il y a aussi le fait d'avoir détecté ou non une *partie* quelconque, tout au long du processus. En effet, le fait de ne pas avoir détecté une *partie* donnée peut dévalider complètement l'hypothèse courante. La dépendance qui existe entre la décision de savoir si un objet est présent dans l'image et la détection de ses composants, peut être modélisée en termes de probabilité conditionnelle.

Donc, le problème peut être posé de la façon suivante :

On considère une zone d'analyse dans laquelle on recherche un objet, et en particulier une *partie* de cet objet.

Soient :
– d : l'évènement : « *une détection compatible avec la* partie *recherchée est réalisée* »
– $\neg d$: l'évènement contraire de d.
– O : l'évènement : « *l'objet est présent dans la zone d'analyse* »
Il est important de considérer non seulement $\Pr(O|d)$ mais aussi $\Pr(O|\neg d)$.

On aura, $\Pr(O|d)$ (*probabilité a posteriori*) que l'on peut comprendre comme *la probabilité que l'objet soit présent dans la zone d'analyse sachant qu'on a une détection compatible avec la* partie *recherchée* :

$$\Pr(O|d) = \frac{\Pr(d|O)\Pr(O)}{\Pr(d)} \tag{2.8}$$

Dans le cadre de la reconnaissance d'objets, une mesure typique pour savoir si l'objet est présent ou non consiste à évaluer le logarithme du rapport de vraisemblance donné par :

$$\mathcal{L} = \log\left(\frac{\Pr(O|d)}{\Pr(\neg O|d)}\right) = \log\left(\frac{\Pr(d|O)}{\Pr(d|\neg O)}\right) + \log\left(\frac{\Pr(O)}{\Pr(\neg O)}\right) \tag{2.9}$$

Avec :

– $\Pr(O)$: il s'agit de la probabilité que l'objet se trouve en effet dans la zone d'analyse. Pour la première itération, on peut convenir d'une valeur fixe (par exemple 0,5). Rappelons que la zone d'analyse sera réduite au fur et à mesure de la détection des *parties* (voir §2.4.5).

– $\Pr(\neg O)$: elle sera simplement donnée par $1 - \Pr(O)$.

Le calcul de la probabilité d'avoir une détection de la *partie* recherchée dans les deux cas possibles, $\Pr(d|O)$ et $\Pr(d|\neg O)$, sera décrite par la suite.

$\Pr(d|O)$ **(*probabilité a priori*)** Il s'agit de la probabilité qu'a le détecteur de trouver une *partie* compatible avec Λ_m, dans la zone sachant que l'objet s'y trouve. Pour cette valeur il faut prendre en compte de manière plus précise les choses : dans le cas où la *partie* est là, on peut estimer la probabilité que le détecteur réponde en prenant en compte les données d'apprentissage. Il ne faut pas oublier que dans l'apprentissage on a appris seulement ce qui se passe dans une région de l'image, de taille de l'aire d'une cellule, quand l'objet s'y trouve. Mais cette information statistique n'est pas suffisante pour caractériser la probabilité d'avoir une détection dans une région d'analyse quelconque sachant que l'objet s'y trouve. Pour cela, il faut d'abord définir $\Pr(d|O)$ pour le cas ponctuel (c'est-à-dire le cas d'une cellule de taille de celle de l'apprentissage) et après pour le cas d'une zone d'analyse quelconque. Dans la figure 2.12 on montre les régions d'analyse pendant le processus de reconnaissance. A : correspond à la zone d'analyse totale, A_c : correspond à la zone d'analyse de la taille d'une cellule.

– **Cas ponctuel A_c : région d'analyse de taille de la cellule.** Cela correspond au cas similaire à l'apprentissage : on a placé la grille de cellules et on a observé ce qui s'est passé dans un endroit précis (de taille de la cellule) sachant que l'objet s'y trouvait.

  Afin de calculer la probabilité a priori $\Pr(d|O)_{A_c}$, il faut d'abord analyser dans quel cas on peut avoir une détection pour une *partie* Λ quelconque.

  La *partie* Λ_m sera présente si :

  – les opérateurs concernant la détection de Λ_m ont répondu,

  – les paramètres issus des opérateurs correspondent bien à l'intervalle $a_{m,n} \leq \zeta_{m,n} \leq b_{m,n}$, (dans le cas où les opérateurs ont répondu).

  En termes de probabilités, pour le cas spécifique où le vecteur de paramètres λ_m est

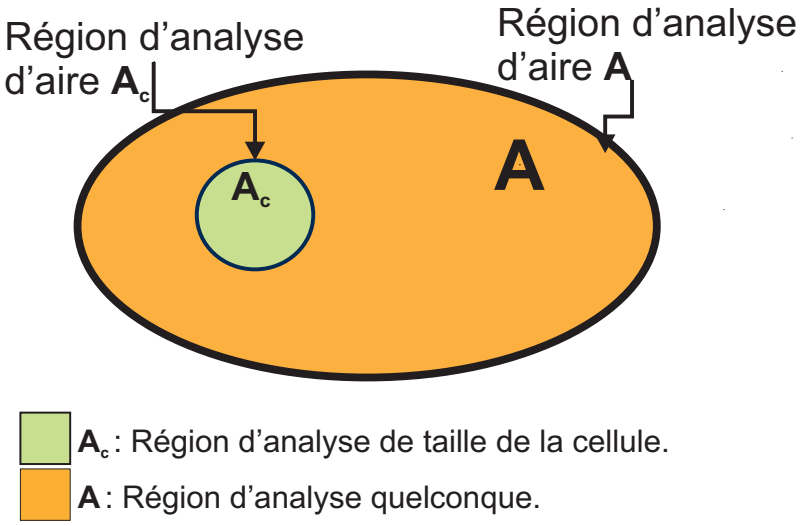


FIG. 2.12 – Régions d'analyse pendant le processus de reconnaissance.

composé d'un seul paramètre $\zeta_{m,n}$, la probabilité d'avoir une détection d_m sachant que l'objet O est présent, $\Pr(d_m|O)_{A_c}$ est calculée par :

$$\begin{aligned} \Pr(d_m|O)_{A_c} &= \Pr(\text{opérateur } n \text{ répond})_{A_c} \Pr(a_{m,n} \leq \zeta_{m,n} \leq b_{m,n})_{A_c} \\ &= \left(\frac{R_{m,n}}{T}\right)\left(\int_{a_{m,n}}^{b_{m,n}} p(\zeta_{m,n})\mathrm{d}\zeta_{m,n}\right) \end{aligned} \tag{2.10}$$

avec « $R_{m,n}$:nombre de réponses de Op_n pour Λ_m » et « T : nombre total des exemples » issus de l'apprentissage (voir §2.3.2).

Pour le cas où λ_m est composée de multiples paramètres $\lambda_m = [\zeta_{m,1}, \zeta_{m,2}, \ldots, \zeta_{m,N_m}]^t$, l'équation (2.10) est généralisée par

$$\Pr(d_m|O)_{A_c} = (0,95)^{N_m} \prod_{n=1}^{N_m} \left(\frac{R_{m,n}}{T}\right) \tag{2.11}$$

du fait que l'on suppose l'indépendance statistique parmi les opérateurs. Le $0,95$ est dû à l'hypothèse que les paramètres $\zeta_{i,j}$ sont distribués selon une loi normale $\zeta_{i,j} \sim \mathcal{N}(\bar{\zeta}_{i,j}, \sigma_{\zeta_{i,j}})$. Typiquement 95% de la surface est utilisée pour un intervalle $[a,b] = \bar{\zeta} \pm 2\sigma_\zeta$.

– **Cas général : région d'analyse quelconque.** Dans ce cas, il faut prendre en compte le fait que, plus la zone d'analyse est grande, plus il y a de chances de détecter

quelque chose d'autre à cause de l'information « parasite ». Cette information parasite est modélisée comme du bruit uniforme ; tant pour sa position dans l'image que pour la valeur des paramètres. On considère ainsi qu'on peut avoir des détections qui correspondent bien à la *partie* qu'on cherche mais qu'elles n'appartiennent pas à l'objet.

Pour ce cas particulier, la probabilité d'avoir une détection dans la zone d'analyse totale est donnée par

$$\Pr(d_m|O)_A = 1 - \Pr(\neg d_m|O)_A, \tag{2.12}$$

$$\text{où } \Pr(\neg d_m|O)_A = (\Pr(\neg d_m|O)_{A_c})^{\frac{A}{A_c}} \tag{2.13}$$

avec

$$\Pr(\neg d_m|O)_{A_c} = 1 - \Pr(d_m|O)_{A_c} \tag{2.14}$$

$$\tag{2.15}$$

La probabilité $\Pr(d_m|\neg O)_A$ est la probabilité d'avoir une détection dans la zone alors que l'objet n'est pas là. On peut se servir pour cela de la probabilité que l'on ait une détection dans une zone de la taille de la cellule considérée (sachant que l'on a pas l'objet) puis d'étendre cette probabilité au cas de la zone totale.

- **Cas ponctuel A_c : probabilité de réponse dans une aire de la taille de la cellule.** Dans le cas où l'objet n'est pas présent, on aura une détection de Λ_m si
  - les opérateurs concernant la détection de Λ_m ont répondu,
  - les paramètres issus des opérateurs correspondent bien à l'intervalle $a_{m,n} \leq \zeta_{m,n} \leq b_{m,n}$ dans le cas où ils ont répondu.

  En termes de probabilités on a :

$$\Pr(d_m|\neg O)_{A_c} = \Pr(\text{l'opérateur } n \text{ répond}|\neg O)_{A_c} \Pr(a_{m,n} \leq \zeta_{m,n} \leq b_{m,n})|\neg O)_{A_c}$$

  - La probabilité pour que la valeur de l'opérateur soit dans $\pm 2\sigma$ autour de sa moyenne pourra être donnée avec l'hypothèse que le signal est uniforme et que l'opérateur n a une dynamique D_n. On aura donc dans ce cas une probabilité $\frac{4\sigma_{\zeta_{m,n}}}{D_n}$ (voir figure 2.13).

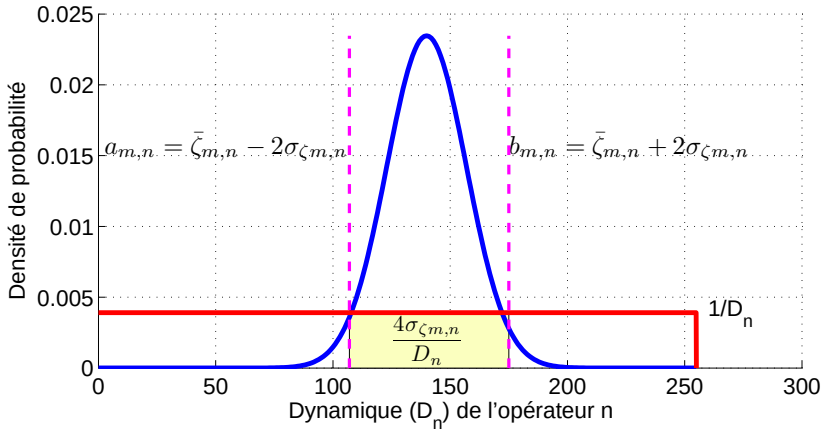


FIG. 2.13 – Lors de l'apprentissage dans une région de la taille d'une cellule, le paramètre $\zeta_{m,n}$ issu de l'opérateur n et appartenant à Λ_m, est supposé suivre une loi normale telle que $\zeta_{m,n} \sim \mathcal{N}(\bar{\zeta}_{m,n}, \sigma_{\zeta_{m,n}})$. Lors de la reconnaissance, dans une région de la taille d'une cellule, ce paramètre est supposé suivre une loi uniforme avec une valeur maximale de $\frac{1}{D_n}$, supposant que l'opérateur a une dynamique D_n. Ainsi, la probabilité qu'a le paramètre $\zeta_{m,n}$, issu de l'opérateur n, d'appartenir à l'intervalle $(a_{m,n} \leq \zeta_{m,n} \leq b_{m,n})$ est donnée par $\frac{4\sigma_{\zeta_{m,n}}}{D_n}$.

Donc, la probabilité $\Pr(d_m|\neg O)_{A_c}$ est calculée par :

$$\Pr(d_m|\neg O)_{A_c} = \frac{4\sigma_{m,n}}{D_n}\left(\frac{R_{m,n}}{T}\right) \tag{2.16}$$

Le cas où λ_m est composé de multiples paramètres $\lambda_m = [\zeta_{m,1}, \zeta_{m,2}, ..., \zeta_{m,N_m}]^t$ est généralisé par

$$\Pr(d_m|\neg O)_{A_c} = \prod_{n=1}^{N_m} \frac{4\sigma_{m,n}}{D_n}\left(\frac{R_{m,n}}{T}\right) \tag{2.17}$$

– **Probabilité de réponse dans l'aire totale** $\Pr(d_m|\neg O)_A$**.** Dans le cas où on doit balayer la zone totale, la probabilité d'avoir une détection sera supérieure (en ce sens on doit prendre en compte le fait qu'analyser une zone grande conduit à des fausses détections potentielles). La probabilité sera donnée par :

$$\Pr(d_m|\neg O)_A = 1 - \Pr(\neg d_m|\neg O)_A \tag{2.18}$$

et,

$$\Pr(\neg d_m|\neg O)_A = (\Pr(\neg d_m|\neg O)_{A_c})^{\frac{A}{A_c}} \quad (2.19)$$

$$= (1-\Pr(d_m|\neg O)_{A_c})^{\frac{A}{A_c}} \quad (2.20)$$

La probabilité $\Pr(\neg d_m|\neg O)_A$ correspond à n'avoir aucune détection dans la zone d'analyse totale.

Jusqu'à présent, on a discuté comment obtenir une mesure de la présence de l'objet sachant qu'on a eu une détection d. La question qui nous intéresse réellement est d'avoir cette mesure de présence de l'objet en ne connaissant pas seulement la détection d, mais un ensemble de détections $d_1, d_2, ..., d_k$ qui correspondent aux *parties* $\Lambda_1, \Lambda_2, ..., \Lambda_k$, respectivement.

Étant donnée une détection d_k, correspondant à la *partie* Λ_k, où k est l'état courant du processus de reconnaissance, on peut formuler le problème de la façon suivante :

$$\Pr(O_k|d_k) = \frac{\Pr(d_k|O_k)\Pr(O_k)}{\Pr(d_k)} \quad (2.21)$$

où

$$\Pr(O_k) = \Pr(O_{k-1}|d_{k-1}) \quad (2.22)$$

Ici, $\Pr(O_{k-1}|d_{k-1})$ correspond à la « nouvelle » probabilité que l'objet soit présent après avoir réalisé la détection d'une des *parties* de l'objet dans l'état précédent. Pour le cas initial, on peut convenir d'une probabilité $\Pr(O_0) = 0,5$ qui correspond à avoir la même probabilité de présence et absence de l'objet, dans la zone d'analyse initiale.

De même, la probabilité que ce ne soit pas l'objet sachant d_k, est obtenue par :

$$\Pr(\neg O_k|d_k) = \frac{\Pr(d_k|\neg O_k)\Pr(\neg O_k)}{\Pr(d_k)} \quad (2.23)$$

où

$$\Pr(\neg O_k) = 1-\Pr(O_k)$$

$$= 1-\Pr(O_{k-1}|d_{k-1}) \quad (2.24)$$

Ce qui nous intéresse est le rapport de vraisemblance entre la probabilité de présence et d'absence de l'objet après une détection donnée :

$$\begin{aligned}\mathcal{L}_k &= \log\left(\frac{\Pr(O_k|d_k)}{\Pr(\neg O_k|d_k)}\right) \\ &= \log\left(\frac{\Pr(d_k|O_k)}{\Pr(d_k|\neg O_k)}\right) + \log\left(\frac{\Pr(O_k)}{\Pr(\neg O_k)}\right) \qquad (2.25)\end{aligned}$$

Et, en intégrant (2.22) dans (2.24) en (2.25) on a

$$\mathcal{L}_k = \mathcal{L}_{k-1} + B_k \qquad (2.26)$$

où le terme $B_k = \log\left(\frac{\Pr(d_k|O_k)}{\Pr(d_k|\neg O_k)}\right)$ et les termes $\Pr(d_k|O_k)$ et $\Pr(d_k|\neg O_k)$ sont calculés en utilisant les équations (2.12) et (2.18) respectivement.

Pour le cas de non détection, et en suivant le même procédé que pour le cas de détection, on obtient

$$\mathcal{L}_k = \mathcal{L}_{k-1} + \neg B_k \qquad (2.27)$$

avec $\neg B_k = \log\left(\frac{\Pr(\neg d_k|O_k)}{\Pr(\neg d_k|\neg O_k)}\right)$, où

$$\begin{aligned}\Pr(\neg d_k|O_k) &= 1 - \Pr(d_k|O_k) \qquad (2.28)\\ \Pr(O_k) &= \Pr(O_{k-1}|\neg d_{k-1}) \qquad (2.29)\\ \Pr(\neg O_k) &= 1 - \Pr(O_k) \\ &= 1 - \Pr(O_{k-1}|\neg d_{k-1}) \qquad (2.30)\\ \Pr(\neg d_k|\neg O_k) &= 1 - \Pr(d_k|\neg O_k) \qquad (2.31)\\ & \qquad (2.32)\end{aligned}$$

Les trois états possibles sont résumés dans le tableau 2.1.

Exemple numérique d'évolution de $\mathcal{L}$ Ici, le but est d'illustrer, par un exemple très simple, l'évolution du rapport de vraisemblance en fonction des détections. Prenons comme exemple l'objet « carré », dont le modèle est composé de quatre coins. Des variations en position, échelle et rotation dans le plan 2D, sont prises en compte. Sur la figure 2.14 on illustre la valeur moyenne sur la position des quatre coins (points rouges), ainsi que l'intervalle d'analyse (ellipses).

Les détecteurs de coins sont simulés comme ayant une fiabilité de 95%. Au départ, nous supposons une probabilité de $0,5$ sur la présence du carré dans l'image ($\Pr(O_0) = 0,5$).

État initial : $\mathcal{L}_0 = \log\left(\frac{\Pr(O_0)}{\Pr(\neg O_0)}\right)$
$\mathcal{L}_k = \begin{cases} \mathcal{L}_{k-1} + B_k, \text{ si détection } d_k \\ \mathcal{L}_{k-1} + \neg B_k, \text{ si non détection } \neg d_k \end{cases}$
$\mathcal{L}_{k-1} = \begin{cases} \log\left(\frac{\Pr(O_{k-1}\|d_{k-1})}{\Pr(\neg O_{k-1}\|d_{k-1})}\right), \text{ si } d_{k-1} \\ \log\left(\frac{\Pr(O_{k-1}\|\neg d_{k-1})}{\Pr(\neg O_{k-1}\|d_{k-1})}\right), \text{ si } \neg d_{k-1} \end{cases}$

TAB. 2.1 – Calcul de $\mathcal{L}$ selon les trois états possibles. Avec $B_k = \log\left(\frac{\Pr(d_k|O_k)}{\Pr(d_k|\neg O_k)}\right)$ et $\neg B_k = \log\left(\frac{\Pr(\neg d_k|O_k)}{\Pr(\neg d_k|\neg O_k)}\right)$.

Sur la figure 2.15 nous montrons les résultats des simulations en prenant trois cas différents.

Dans le premier cas (figure 2.15-A), nous avons simulé les détections réussies successivement (de 1 à 4) des quatre coins. La colonne gauche montre la réduction de l'espace de recherche au fur et à mesure des détections. Dans la colonne droite, on peut observer l'évolution du rapport de vraisemblance. Pour cet exemple, $\mathcal{L}$ croît d'une manière exponentielle du fait de la réduction assez élevée de la région d'analyse au fur et à mesure des détections. Il faut remarquer que $\mathcal{L}$ a été tracé en fonction de l'inverse de la taille de la zone d'analyse.

Dans le deuxième exemple (figure 2.15-B), trois des quatre *parties* ont été supposées détectées ; seul le dernier coin n'a pas pu être détecté (voir Fig. 2.15-B.4). Ici, $\mathcal{L}$ décroît lors de la non détection du quatrième coin. Du fait que le détecteur a une fiabilité de 95% et que la région d'analyse est assez restreinte, la non détection du coin remet en cause la présence du carré en faisant chuter $\mathcal{L}$ à presque la valeur initiale.

Dans le troisième exemple (voir figure 2.15-C), deux des quatre coins ont été supposés détectés. A la différence du premier et deuxième exemple, ici il devient très peu probable que l'objet soit dans la région d'analyse du fait que deux des coins n'ont pas été détectés : les détections réussies peuvent avoir été engendrées par quelque chose d'autre que le carré. La *région d'analyse assez restreinte* (si le carré était là, il aurait fallu détecter quelque chose dans cette région d'analyse) et la *non détection des coins en ayant un détecteur assez fiable*, fait que la présence de l'objet devient très faible. Avec cette modélisation, les occulations sont prises en compte par la caractérisation probabiliste du

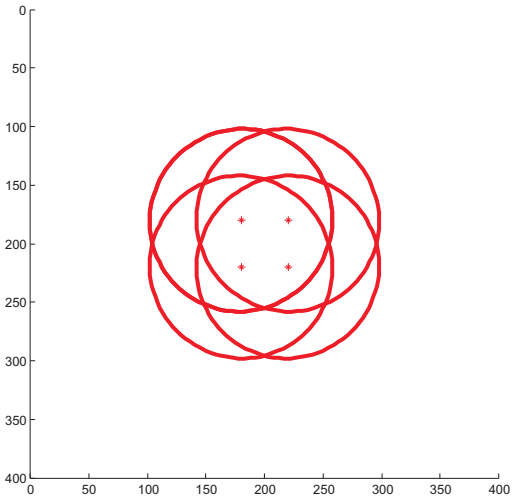


FIG. 2.14 – Modèle du « carré ». Les points rouges correspondent à la position moyenne des quatre coins composant l'objet. Les ellipses montrent l'intervalle permis pour la position des quatre coins. Ce modèle prend en compte des translations, des changements en échelle ainsi que des rotations dans le plan 2D.

détecteur.

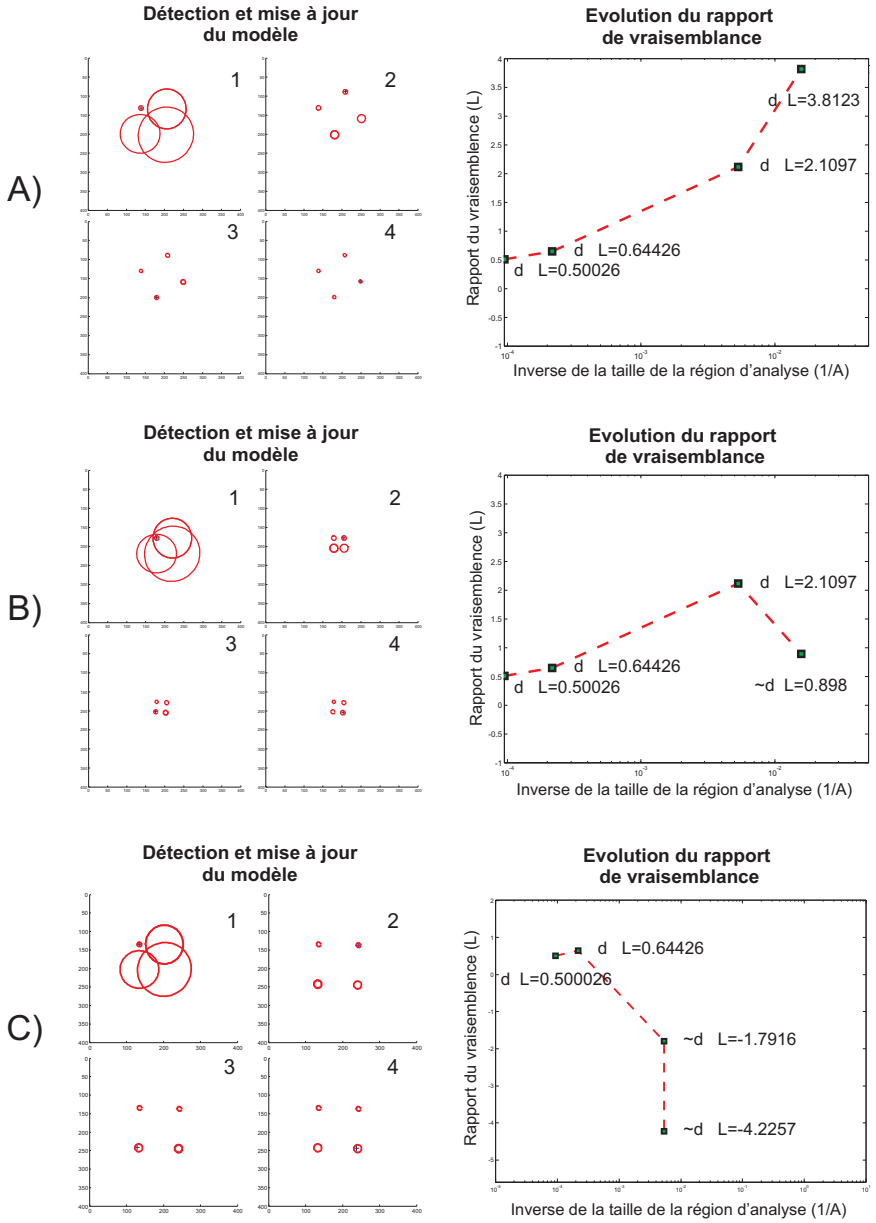


FIG. 2.15 – Résultats des simulations des trois exemples de détection du « carré ». Dans la colonne à gauche sont affichés les modèles mis à jour lors de détections/non détection des coins. Le points rouges représentent la position probable des quatre coins du carré, alors que les ellipses donnent l'intervalle permis sur ces positions. Dans le premier cas (A), les quatre coins ont été supposés détectés. Pour le deuxième exemple (B), les trois premières détections ont réussi alors que la quatrième ne l'est pas. Pour le dernier (C), uniquement les deux premières détections ont été supposées réussies.

2.4.6.2 Critère de Reconnaissance

Une fois établi comment on peut avoir une mesure de la probabilité de présence d'un objet connaissant le résultat des détections, il reste à déterminer le seuil de reconnaissance τ qui définit si, d'après les observations réalisées, l'objet est présent ou non.

Le critère de décision doit prendre la forme suivante :

$$\mathcal{L}_k = \log\left(\frac{\Pr(O_k|d_k)}{\Pr(\bar{O}_k|d_k)}\right) \geq \tau \tag{2.33}$$

Ainsi, si ce critère est atteint, on considère que l'objet est reconnu.

Strictement, la valeur de τ doit être déterminée en faisant la minimisation de l'erreur de classification. Cependant, du fait de la complexité que cela implique de par la nature récursive de l'algorithme de recherche, on est obligé de trouver une autre façon pour définir le critère de décision.

2.4.6.3 Décision finale par (SVM)

Après certaines détections, nous pouvons observer que les régions d'intérêt dans l'espace géométrique sont très petites pour les *parties* restantes. Dans ce cas là, l'algorithme doit arrêter la récursivité et faire soit une recherche séquentielle soit une observation « globale » des *parties* restantes.

Du fait des bonnes performances des classificateurs basés sur les SVM, nous avons choisi d'en utiliser un pour donner la classification finale pendant le processus de reconnaissance : objet/non objet. Ainsi, le but est de savoir si l'objet est présent ou non, une fois détectées un certain nombre de *parties*. Pour cela, nous ajoutons, au modèle de l'objet, des *parties* qu'on appelle « composées ». Ces *parties* composées correspondent à un ensemble de *parties primitives ou de base* qui décrivent l'apparence globale de l'objet.

Chaque *partie* composée est caractérisée aussi par un poids w, indiquant la pertinence de cette dernière dans le processus de reconnaissance. Normalement la recherche de ce type de primitives est très coûteux, du fait du balayage exhaustif en position et en échelle, et de la dimension du vecteur de caractéristiques. En revanche, une seule évaluation du classificateur ne demande pas trop de ressources.

Donc, la stratégie proposée est la suivante :

1. Surveiller les variances de l'ensemble des *parties* qui ne sont pas encore détectées, et obtenir celle qui a la valeur maximale σ^2_{max}, pour un niveau de résolution donné.

2. Si la variance maximale σ^2_{max} est inférieure à un certain seuil τ_σ, cela veut dire que toutes les *parties* restantes sont quasiment fixées en position et peu variables en apparence. Donc, dans ce cas, une observation « globale » de toutes les *parties* est pertinente[7]. La décision de savoir si l'ensemble de *parties* observées correspond ou non à l'objet qu'on cherche, est faite en utilisant un classificateur SVM.

Même si le critère défini dans l'équation (2.33) n'est pas utile pour la décision finale, objet/non objet, il est de grande utilité pour donner une idée sur la pertinence de continuer le processus de recherche pour d'autres *parties*, ou d'arrêter le parcours d'une branche de l'arbre de recherche et changer d'hypothèse.

2.4.6.4 Branch and Bound

On doit indiquer à l'algorithme le plus tôt possible quand il doit abandonner le parcours d'une branche de l'arbre dans le cas où il est peu probable de trouver l'objet. En d'autres termes, il faut évaluer la pertinence de détecter probablement l'objet ou non. Si la probabilité de détecter l'objet devient très faible, peut-être ne vaut-il plus la peine de continuer à chercher le long de cette branche ? Car même si les futures détections sont réussies, le critère de reconnaissance ne sera jamais atteint asymptotiquement. Dans ce cas, l'algorithme doit éliminer cette hypothèse et revenir en arrière dans l'arbre de recherche.

Donc, pour chaque itération du processus de reconnaissance, on doit évaluer la **pertinence** de détecter probablement l'objet. Cette pertinence est donnée par :

$$\hat{\mathcal{L}}_k \quad < \quad \tau_s \tag{2.34}$$

où $\hat{\mathcal{L}}_k$ correspond à l'estimation du rapport de vraisemblance en supposant que les K' *parties* restantes seront détectées, et $k \in [1, K']$. τ_s est un seuil défini par l'utilisateur.

Si la relation (2.34) est vraie, le processus élimine l'hypothèse courante et revient en arrière dans l'arbre de recherche pour tester une nouvelle hypothèse.

En résumé, l'algorithme proposé est constitué des étapes suivantes :

[7] A partir de ce moment, le coût de recherche par l'algorithme récursif, dépasserait celui du classifieur.

1. Sélection d'une *partie* de l'objet à reconnaître (parmi toutes les *parties* qui n'ont pas été encore détectées) selon le critère de maximum du coefficient de corrélation et de variance minimale (dans l'espace géométrique)
2. Délimitation de la région d'intérêt
3. Détection de la *partie* sélectionnée
4. Si la *partie* n'a pas été détectée :
   (a) marquer la *partie* sélectionnée comme « non détectée »
   (b) $\mathcal{L}_k = \mathcal{L}_{k-1} + \neg B_k$
   (c) revenir à l'étape numéro 1
5. Si la *partie* a été détectée :
   (a) marquer la *partie* comme « détectée »
   (b) $\mathcal{L}_k = \mathcal{L}_{k-1} + B_k$
6. Si l'objet est reconnu :
   – fin du processus de reconnaissance
7. Si $\hat{\mathcal{L}}_k \geq \tau_s$ (il est pertinent de continuer à explorer la branche courante)
   – Mise à jour du modèle
   – Revenir à l'étape numéro 1
8. Si $\hat{\mathcal{L}}_k < \tau_s$
   – $\mathcal{L}_k \leftarrow \mathcal{L}_{k-1}$
   – $k \leftarrow k-1$ (retour en arrière en niveau de profondeur)
   – Revenir à l'étape numéro 1

Sur la figure 2.16 on montre un exemple de l'évolution de l'algorithme, avec des éventuels retours en arrière.

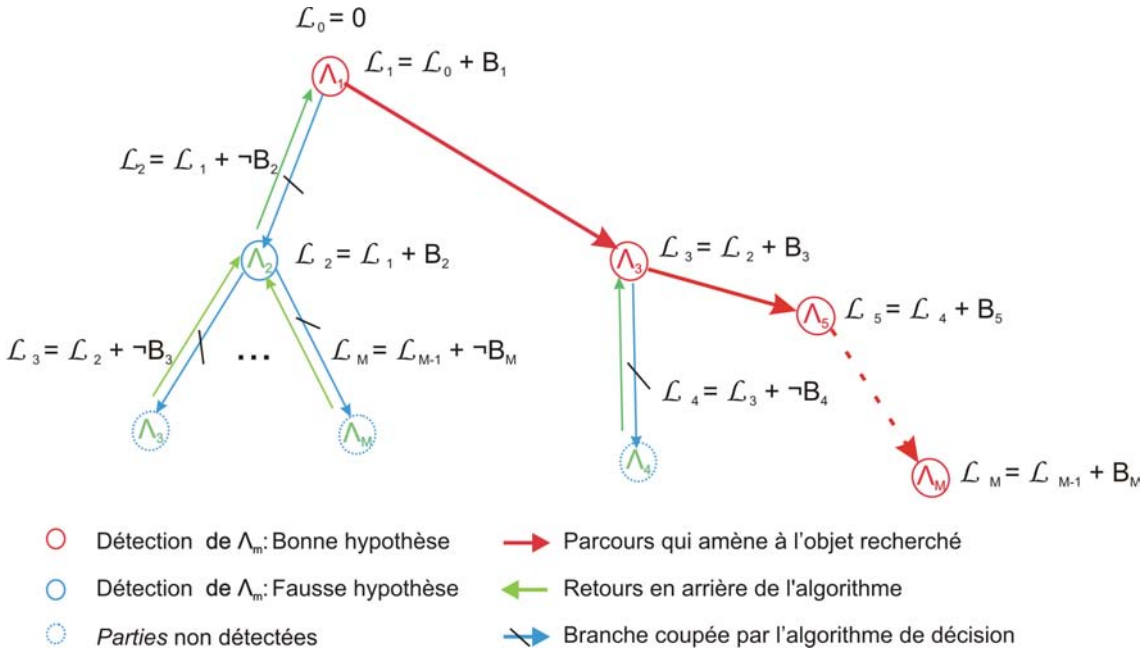


FIG. 2.16 – Exemple d'évolution de l'algorithme et du rapport de vraisemblance. Sur la figure montrée ci-dessus, on présente un exemple du parcours de l'arbre de recherche à partir de la détection des bonnes hypothèses (cercles rouges) et des fausses hypothèses (cercles bleus). Les cercles pointillés indiquent les non détections. Pour faciliter la compréhension, le parcours de l'arbre s'est fait de haut en bas et de gauche à droite. Les flèches bleues montrent le parcours que fait l'algorithme en partant sur de fausses hypothèses. A côté de chaque cercle, on présente la mise à jour du rapport de vraisemblance une fois que la *partie* en question a été détectée. Les légendes à coté des flèches (lecture uniquement au retour) correspondent à la mise à jour du rapport de vraisemblance qui prend en considération la non détection de la *partie* recherchée (à la pointe de la flèche bleue). Le retour en arrière (flèches vertes) se fait lorsque que l'on considère, au moyen du rapport de vraisemblance, qu'il n'est plus pertinent d'explorer cette branche de l'arbre (branch and bound). A son tour, la flèche rouge pointillé nous indique le parcours de l'arbre avec toutes les hypothèses possibles jusqu'à la détection de l'objet.

2.4.7 Bilan sur la stratégie de reconnaissance

Nous avons choisi d'aborder le problème de la stratégie de reconnaissance en prenant comme principe fondamental le concept de focalisation. Ceci, à son tour nous a conduit à la proposition d'un mécanisme de contrôle du processus de reconnaissance. Ce dernier est défini par un ensemble d'étapes qui va de la génération d'hypothèses jusqu'à la prise de décision, en passant par la détection des *parties* et la mise à jour du modèle. Le processus de reconnaissance est un algorithme récursif qui est guidé de façon optimale en fonction

des parties de l'objet déjà détectées.

L'ensemble de cette démarche nous permet d'aboutir à quelques résultats qu'il nous semble intéressant d'énoncer ici. D'abord, nous pouvons remarquer que le mécanisme d'attention visuelle est guidé conjointement par le modèle et par les indices. Ensuite, étant donné qu'il existe une dépendance entre le critère de sélection des *parties* et l'état du processus de reconnaissance, la hiérarchisation des hypothèses est alors un résultat implicite et aussi dépendant de l'état du processus. Ainsi, la *partie* sélectionnée sera celle qui génère le moindre coût de détection et qui contribue à la convergence la plus efficace du processus.

2.5 Localisation et suivi d'objets

Nous avons vu que l'apprentissage prenait en compte la position des cellules dans l'image et que le processus de reconnaissance mettait à jour (entre autre) ces paramètres de positionnement pour rechercher les cellules suivantes.

Dans le cas où l'on dispose d'informations dimensionnelles sur l'objet, il devrait donc être possible de mettre à jour aussi sa position dans le monde 3D au fur et à mesure de l'évolution du processus de reconnaissance.

2.5.1 Modélisation 3D

On suppose ici que l'objet à reconnaître est plan et que l'on connaît une métrique le caractérisant comme la distance Y_0 à laquelle ont été réalisées les prises de vue pour l'apprentissage (on pourrait aussi bien choisir une de ses dimensions largeur ou hauteur).

Considérons que l'on a réalisé l'apprentissage d'une cellule $\mathcal{C}_i$, liée à cet objet, de coordonnées $\mathbf{a}_{0i} = (u_{0i}, v_{0i})^t$ dans l'image. Notons $\mathbf{a}_{i3D} = (X_i, Y_0, X_i)^t$ la position de cette cellule dans le monde 3D lors de l'apprentissage. En considérant que l'objet était parfaitement centré dans l'image, et qu'aucune rotation ou translation autre que la distance de l'objet au centre optique n'apparaisse lors de l'apprentissage, nous aurons :

$$\mathbf{a}_{0i} = \begin{pmatrix} u_{0i} \\ v_{0i} \end{pmatrix} = \begin{pmatrix} e_u \dfrac{X_i}{Y_0} \\ e_v \dfrac{Z_i}{Y_0} \end{pmatrix}$$

Avec e_u et e_v les paramètres de projection perspective de la caméra respectivement en u et v selon l'axe Y (on suppose un repère image centré et on ne considère pas de distorsion géométrique).

Les paramètres de positionnement 3D $\mathbf{a}_{i3D} = (X_i, Y_0, Z_i)$ peuvent être déduits de $\mathbf{a}_{0i}$ par :

$$\mathbf{a}_{i3D} = \begin{pmatrix} X_i \\ Y_0 \\ Z_i \end{pmatrix} = \begin{pmatrix} u_{0i}\frac{Y_0}{eu} \\ Y_0 \\ v_{0i}\frac{Z_0}{e_v} \end{pmatrix}$$

On voit que l'on pourra ainsi trouver un lien entre la position des cellules dans l'image lors de l'apprentissage et celle après déplacement de l'objet dans le monde.

Ainsi, considérons à présent le cas où la caméra s'est déplacée par rapport à l'objet (voir figure 2.17) et que l'on prenne en compte des paramètres de translation autres que Y ainsi que des rotations. On regroupera dans le vecteur $\mathbf{t} = (\alpha, \beta, \gamma, X, Y, Z)$ l'ensemble de ces paramètres.

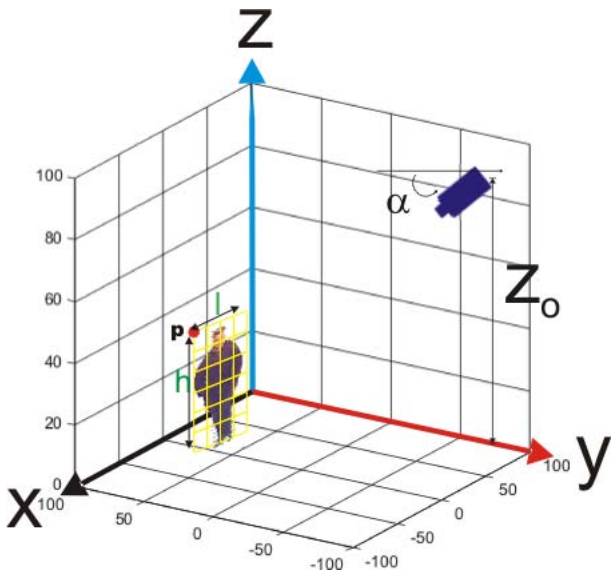


FIG. 2.17 – L'objet considéré comme appartenant à une scène 3D observée. A la différence de la phase d'apprentissage, ici la caméra s'est déplacée par rapport à l'objet par des paramètres de translation (X, Y, Z) et des rotations (α, β, γ).

On considérera que ce vecteur de paramètres est inconnu mais suit une loi normale telle que :

$$\mathbf{t} \sim \mathcal{N}(\bar{\mathbf{t}}, \Sigma_{\mathbf{t}})$$

La valeur moyenne $\bar{\mathbf{t}}$ ainsi que sa covariance $\Sigma_{\mathbf{t}}$ seront dépendantes de l'application.

La position $\mathbf{a}_i$ de la cellule C_i dans l'image sera, à présent :

$$\mathbf{a}_i = g_i(\mathbf{t}, \mathbf{a}_{0i})$$

La fonction g_i représente ici la projection de la cellule 3D i dans l'image en fonction des paramètres $\mathbf{t}$. En effet, nous avons vu que la position 3D des cellules lors de l'apprentissage pouvait être liée à celle 2D dans l'image d'apprentissage.

2.5.2 Estimation de la pose 3D de l'objet

A présent, le problème posé est d'estimer le positionnement 3D de l'objet ayant localisé la position $\mathbf{a}_i$ d'une cellule dans l'image.

Pour cela, on peut réaliser l'approximation de Taylor-Young suivante :

$$\mathbf{a}_i = g_i(\mathbf{t}) \approx g_i(\mathbf{t}_0) + \mathbf{J}_{g_i}(\mathbf{t} - \mathbf{t}_0)$$

$\mathbf{t}_0$ est le vecteur de positionnement initial qui vaut $\mathbf{t}_0 = (0,0,0,0,Y_0,0)^t$ lors de la première itération de l'algorithme mais qui s'affine selon la profondeur de recherche.

$\mathbf{J}_{g_i}$ est la matrice jacobienne de la fonction g_i définie par :

$$\mathbf{J}_{g_i} = \begin{pmatrix} \frac{\partial u_i}{\partial \alpha} & \frac{\partial u_i}{\partial \beta} & \frac{\partial u_i}{\partial \gamma} & \frac{\partial u_i}{\partial X} & \frac{\partial u_i}{\partial Y} & \frac{\partial u_i}{\partial Z} \\ \frac{\partial v_i}{\partial \alpha} & \frac{\partial v_i}{\partial \beta} & \frac{\partial v_i}{\partial \gamma} & \frac{\partial v_i}{\partial X} & \frac{\partial v_i}{\partial Y} & \frac{\partial v_i}{\partial Z} \end{pmatrix}$$

Soit donc :

$$\mathbf{a}_i - g_i(\mathbf{t}_0) + \mathbf{J}_{g_i}\mathbf{t}_0 = \mathbf{J}_{g_i}\mathbf{t}$$

On peut élaborer, lors de chaque mise à jour de la position d'une cellule, l'équation d'observation $\mathbf{y}$ suivante :

$$\mathbf{y} = \mathbf{a}_i - g_i(\mathbf{t}_0) + \mathbf{J}_{g_i}\mathbf{t}_0 = \mathbf{J}_{g_i}\mathbf{t}$$

à laquelle est associée une erreur de covariance Σ_{a_i} du fait du caractère constant de $g_i(\mathbf{t}_0) + \mathbf{J}_{g_i}\mathbf{t}_0$.

Le vecteur $\mathbf{t}$ et sa matrice de covariance sont donnés par la mise à jour de $\mathbf{t}_0$ et $\Sigma_{\mathbf{t}_0}$ selon les équations d'un filtre de Kalman dégénéré (pas d'évolution dynamique des paramètres du fait que l'on considère l'objet immobile lors de sa phase de reconnaissance) :

$$\begin{cases} \mathbf{K} &= \Sigma_{\mathbf{t}_0}\mathbf{J}_{g_i}^t(\mathbf{J}_{g_i}\Sigma_{\mathbf{t}_0}\mathbf{J}_{g_i}^t + \Sigma_{\mathbf{a}_i})^{-1} \\ \mathbf{t} &= \mathbf{t}_0 + \mathbf{K}_p(\mathbf{y} - \mathbf{J}_{g_i}\mathbf{t}_0) = \mathbf{t}_0 + \mathbf{K}_p(\mathbf{a}_i - g_i(\mathbf{t}_0)) \\ \Sigma_{\mathbf{t}} &= \Sigma_{\mathbf{t}_0} - \mathbf{K}_p\mathbf{J}_{g_i}\Sigma_{\mathbf{t}_0} \end{cases}$$

2.5.3 Suivi 3D

Dès lors que l'on a réussi à reconnaître l'objet, le suivi consiste à initialiser la position des parties autour des positions prévisibles $\mathbf{a}_i(t|t-1) = g_i(\mathbf{t}(t|t-1))$.

Le vecteur $\mathbf{t}(t|t-1)$ est issu d'une étape de prédiction conforme au modèle d'évolution du procédé, c'est-à-dire prenant en compte un modèle de mouvement à la fois de la caméra et de l'objet.

Ainsi, le suivi s'intègre d'une façon naturelle dans notre approche. En fait, nous faisons la reconnaissance à chaque instant t de la séquence d'images. Cependant, il existe une grande différence par rapport au cas initial : la région d'intérêt initiale à l'instant $t+1$, pour chacune des *parties* du modèle, est énormément réduite par rapport à la région d'intérêt du modèle initial tel qu'il est issu de la phase d'apprentissage. Cette diminution de la région d'intérêt est due au fait de garder le dernier modèle une fois l'objet reconnu. Ainsi, pour l'image d'après, la position moyenne, de toutes les *parties* du modèle, sera autour des *parties* de l'objet en question. Si l'on ajoute un modèle d'évolution sur la position probable des *parties*, à l'instant $t+1$ on aura la région d'intérêt où il est probable de trouver une *partie* de l'objet une fois qu'elle s'est déplacée. N'oublions pas que, même si nous avons traité le cas de la position des *parties*, il est aussi possible d'étendre cette idée au cas plus général : nous pouvons avoir un modèle d'évolution non seulement sur la position des *parties*, mais aussi sur les paramètres. Ainsi, il est possible non seulement de suivre l'objet en position et en échelle dans l'image, mais aussi en apparence[8]. Il faut remarquer qu'il n'y a pas deux modules différents, un pour la reconnaissance et un autre pour le suivi : c'est toujours le même processus de reconnaissance mais avec une dynamique temporelle où un modèle d'évolution entre en jeu.

2.6 Bilan

Dans l'objectif d'étudier la reconnaissance d'objets, sous un regard de compréhension de scènes, nous avons procédé à l'analyse de quatre étapes décisives. Nous avons étudié d'abord la représentation et la structure de l'objet, ce qui correspond à notre positionnement en termes de modélisation. Cette étape nous a permis de définir d'une part la nature structurelle de l'objet et d'autre part les composants de celui-ci. Ces composants ou *parties* de l'objet sont à leur tour définis comme des régions qui décrivent la structure locale de l'objet à partir de plusieurs caractéristiques. De manière analogue, nous avons défini l'apparence globale comme résultant des inter-dépendances statistiques entre les

[8]L'apparence est un cas particulier où les paramètres, qui composent une *partie*, décrivent effectivement l'apparence de l'objet. Dans le cas le plus général nous parlons du suivi dans l'espace des paramètres.

différentes *parties*. Ensuite nous avons décrit les étapes successives nécessaires qui permettent d'obtenir les *parties* qui caractérisent l'objet. Après, une troisième partie a permis d'identifier un ensemble d'étapes qui composent la stratégie de reconnaissance. L'étude de l'algorithme de contrôle est fondamental dans le guidage du processus de reconnaissance. En particulier le guidage conjoint bottom-up $\leftrightarrow$ top-down ainsi que les effets de la focalisation sur quatre niveaux différents : l'espace 2D (image), l'espace du niveau de détail (résolution), l'espace des paramètres (propriétés) et l'espace des opérateurs bas niveau. Enfin, nous avons décrit comment estimer les paramètres de localisation (quand il s'agit d'une scène 3D), ainsi que les éléments nécessaires pour faire le suivi d'objets.

Finalement, nous avons tous les éléments nécessaires pour mettre en place une telle méthode. C'est dans le chapitre qui suit, que l'on présentera une application pour deux classes différentes d'objets : la reconnaissance, la localisation et le suivi de visages et de piétons.

Chapitre 3

Applications : reconnaissance, localisation et suivi de visages et de piétons

3.1 Introduction

Tout au long de ce mémoire, nous avons concentré tout notre intérêt au développement de la méthodologie de reconnaissance d'objets. Elle doit évidemment être validée.

Pour ce faire, et en raison de leur importance, nous avons choisi deux exemples de classe d'objets : d'une part la classe d'objets « visages », d'autre part la classe d'objets « piétons ». Ces deux classes constituent un défi important du fait des caractéristiques qu'elles présentent : une haute variabilité dans leurs apparences, des objets articulés (dans le cas de l'objet piéton), ils sont intégrés dans des scènes visuelles complexes ce qui entraîne des changements de luminosité et la possibilité d'avoir un grand nombre d'éléments perturbants.

Des fonctionnalités importantes comme la reconnaissance, la localisation et le suivi d'objets simultanés, sont comprises dans la méthodologie proposée. En fait, la mise en place du mécanisme attentionnel est complètement justifiée : on constate une diminution importante de la complexité de la tâche de recherche visuelle.

Ainsi donc, le but de ce chapitre est de montrer la viabilité d'une approche par vision focalisée pour la reconnaissance visuelle d'objets.

L'organisation du chapitre est la suivante : d'abord, nous allons définir des opérateurs bas niveau qui seront utilisés pour le test avec les deux classes d'objets. Ensuite, la mise en place de la méthodologie est montrée en prenant comme exemple l'objet « visage ».

Après, nous présentons les résultats pour l'objet « piéton ». Enfin, un bilan des résultats obtenus est présenté.

3.2 Opérateurs bas niveau

Inspirés par les systèmes biologiques, nous avons décidé de caractériser la structure locale de l'objet à partir de trois propriétés qui nous semblent les plus discriminantes pour l'attention visuelle : l'orientation du contour, le niveau de gris moyen et la position spatiale de chaque descripteur local.

3.2.1 Orientation du contour à partir d'une carte d'orientations

Concernant les descripteurs bas niveau indiqués au §2.2.3.3 (traitement primaire), le filtre de Gabor présente plusieurs propriétés qui le rendent pertinent pour analyser la structure locale de l'image. A la différence de la transformée de Fourier, dont la valeur des coefficients dépend de l'image entière, le filtre de Gabor est localisé, d'une façon optimale, dans le domaine de la fréquence et de l'espace simultanément, et à une orientation spécifique. Pour la vision biologique, vers les années 60, un nombre extrêmement important d'études a été réalisé sur l'aire visuelle V1 (première étape de traitement cortical chez le primate). Dès les premiers travaux, Hubel et Wiesel [48] montrent que la majorité des neurones de V1 répond aux différentes orientations des contours dans l'image (des cellules sélectives à l'orientation). Certains auteurs ont proposé que les cellules de l'aire corticale soient modélisées par des fonctions de Gabor [59]. En principe, si on applique un nombre assez grand de filtres de Gabor à différentes échelles, orientations et fréquences spatiales, on peut analyser une image avec une description locale assez détaillée [39].

3.2.1.1 Filtre de Gabor 2D

Le filtre de Gabor est une fonction Gaussienne modulée par une sinusoïde qui prend la forme mathématique suivante :

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp^{[-\frac{1}{2}(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2})+\frac{i2\pi x}{\lambda}]} \tag{3.1}$$

où :

- σ_x et σ_y donnent l'échelle du filtre,
- λ est la longueur d'onde du filtre (en pixels).

Un changement de coordonnées (x', y') par une rotation θ correspond à l'angle d'orientation du filtre :

$$\begin{aligned} x' &= x\cos(\theta) + y\sin(\theta) \\ y' &= -x\sin(\theta) + y\cos(\theta) \\ g_\theta(x,y) &= g(x',y') \end{aligned}$$

Ainsi, l'analyse d'une image I par le filtre de Gabor sera simplement

$$\begin{aligned} I_r &= I * \mathrm{Re}(g_\theta) \\ I_i &= I * \mathrm{Im}(g_\theta) \end{aligned} \tag{3.2}$$

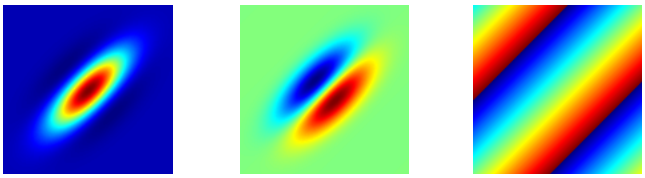
où $*$ est l'opérateur de convolution, $\mathrm{Re}(g_\theta)$ et $\mathrm{Im}(g_\theta)$ correspondent respectivement aux parties réelle et imaginaire de g_θ.

Du fait que la composante de phase porte la partie principale de l'information de l'image [13], une analyse de phase avec le filtre de Gabor est souhaitable. En fait, la *phase locale* donne une description invariante aux changements de luminosité de la structure locale. Cette dernière a été très utilisée pour l'analyse de textures [13, 58].

La phase locale ϕ est calculée par

$$\phi = \arctan \frac{I_i}{I_r} \tag{3.3}$$

Sur la figure 3.1 on montre les masques (réponse impulsionnelle) des parties réelle et imaginaire du filtre de Gabor (de taille 128 pixels) ainsi que l'angle de phase ϕ, avec $\sigma_x = 12,5$, $\sigma_y = 25$, $\lambda = 80$ pixels et $\theta = -45°$.



a) Partie réelle b) Partie imaginaire c) Phase locale

FIG. 3.1 – Masques du filtre de Gabor. a) Partie réelle. b) Partie imaginaire. c) Phase locale.

Sur la figure 3.2 on montre un exemple d'analyse de phase locale sur une image de test avec luminosité variable (figure 3.2-a). On peut observer comment, à la différence des parties réelle, imaginaire et du module, même dans les régions où il y a peu de contraste

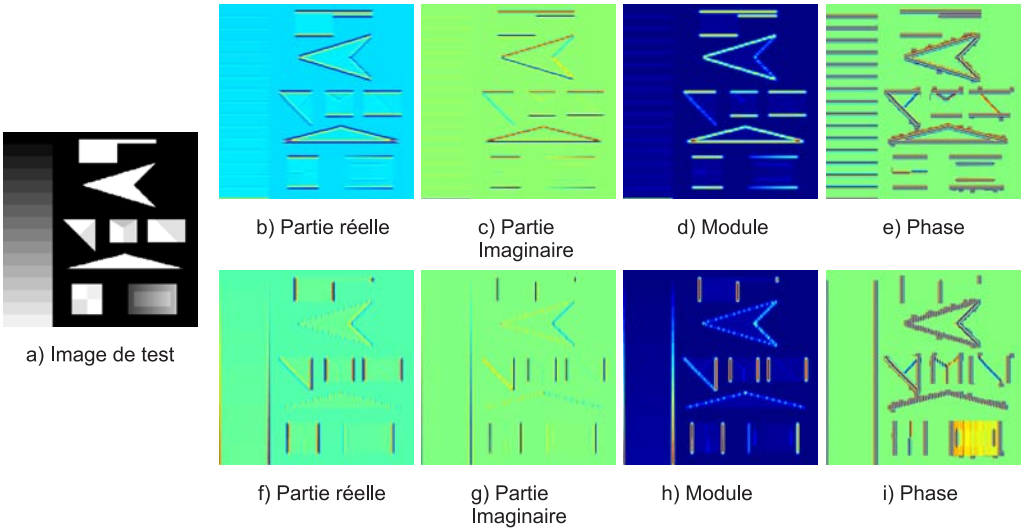


FIG. 3.2 – Analyse de phase avec le filtre de Gabor. a) Image de test. b,f) Partie réelle. c,g) Partie imaginaire. d,h) Module. e,i) Phase locale, pour le filtre de Gabor avec paramètres $\sigma_x = \sigma_y = 5$, $\lambda = 20$, $\theta = 90°$ et $\theta = 0°$ respéctivement.

les contours horizontaux et verticaux sont bien détectés (voir figure 3.2-e,i).

Pour avoir un bon recouvrement du spectre fréquentiel on doit faire varier la taille du filtre, l'orientation et la fréquence spatiale. Des études montrent qu'en changeant la taille du filtre par un facteur puissance 2, à différentes orientations (pas de $\pi/6$ rad), on arrive à couvrir tout le spectre fréquentiel et spatial. Certaines approches, plutôt qu'utiliser un filtre variable en échelle, réalisent une décomposition pyramidale afin de détecter différentes fréquences spatiales [16, 51]. Le filtre de Gabor est utilisé pour construire une carte d'orientations décrite par la suite.

3.2.1.2 La carte d'orientations

Afin d'obtenir l'orientation du filtre qui répond le mieux dans une zone spécifique de l'image, on se sert d'une carte d'orientations. Pour un niveau de résolution donné, cette carte est obtenue à partir du filtrage de l'image par une « banque » de filtres à différentes orientations : $0, 30°, 60°, 90°, 120°$ et $150°$, et de largeur d'onde fixe définie par $\lambda = L$, où L est la taille du masque. La composante de phase $\phi(x,y)$ est calculée par la relation (3.3). Donc, pour un niveau de résolution donné, l'orientation du filtre qui répond le mieux dans une région est obtenue en prenant la direction du filtre où sa phase est maximale. Sur la figure 3.3-d on montre la carte d'orientations (6 orientations au total) pour trois images de test avec éclairage différent. Ici, chaque couleur représente l'orientation du filtre le mieux

adapté.

De façon analogue, sur la figure 3.4-b on montre la carte d'orientations construite à partir de l'image de test présentée en 3.4-a. Même si l'éclairage est très défavorable à l'intérieur du visage, l'analyse de phase extrait la structure locale de cette région. De la carte d'orientations obtenue, nous pouvons choisir d'extraire les zones dans l'image avec une structure verticale (voir Fig. 3.4-c) ou horizontale (voir Fig. 3.4-d).

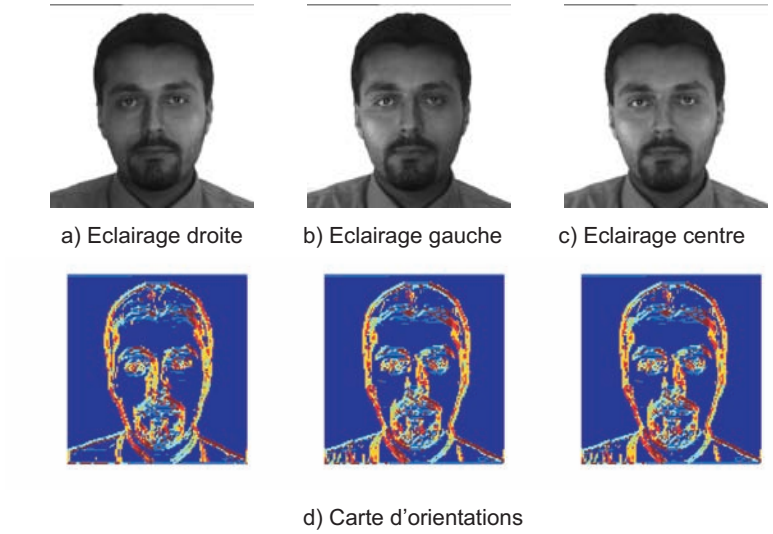


FIG. 3.3 – Exemple de carte d'orientations pour trois images avec différents éclairages. a) Éclairage droit. b) Éclairage gauche. c) Éclairage au centre. d) Carte d'orientations pour a), b), et c), respectivement. La couleur du rouge foncé au bleu clair, représente l'angle d'orientation pour $0°, 30°, 60°, 90°, 120°, 150°$ respectivement.

3.2.2 Filtre moyenneur

Un autre paramètre important est le niveau de gris moyen. Pour un pixel de l'image I avec un voisinage de a pixels en x et y direction, le niveau de gris moyen est calculé par :

$$\bar{I}(x,y) = \frac{1}{(2a+1)^2}\left(\sum_{v=y-a}^{y+a}\sum_{u=x-a}^{x+a} I(u,v)\right) \tag{3.4}$$

où $2a+1$ est la largeur de la région (supposant une région carrée). Par une approche par filtrage, l'image moyenne $\bar{I}$ peut être calculée par :

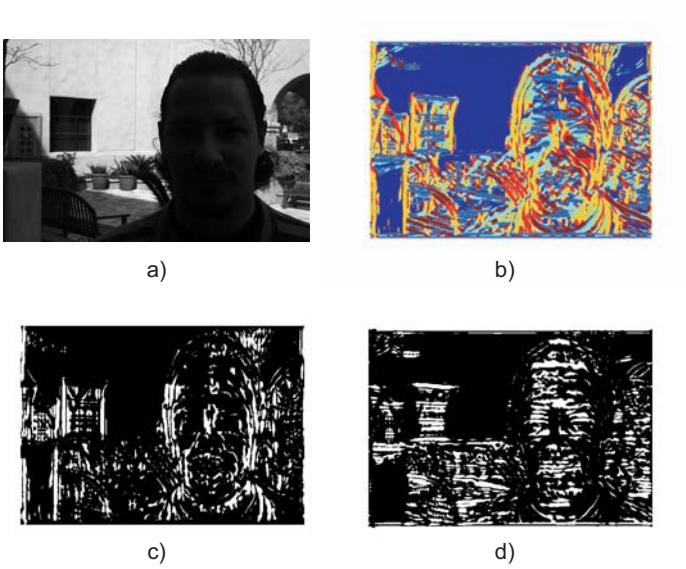


FIG. 3.4 – a) Image de test peu contrastée. b) Carte d'orientations. c) Réponse du filtre à 0°. d) Réponse du filtre à 90°. Pour cet exemple, même si l'éclairage est très défavorable, avec l'analyse de phase locale on est capable d'extraire la structure à l'intérieur du visage.

$$\bar{I} = \frac{I * h}{(2a+1)^2}$$

où $*$ est l'opérateur de convolution, et h correspond au masque du filtre du taille $(2a+1) \times (2a+1)$. Tous les éléments de h sont égaux à 1.

3.3 Reconnaissance et suivi de visages

Les visages ont été largement utilisés ces dernières années pour tester les algorithmes de détection et de reconnaissance d'objets [124, 115, 72, 61, 71, 100]. Mis à part le besoin généré par de nombreuses applications, ils représentent un test significatif pour les algorithmes de reconnaissance. Ceci est dû principalement au fait de leurs caractéristiques : elles présentent une variabilité assez importante dans leur apparence. On peut trouver par exemple des visages avec des expressions différentes, partiellement occultés (des sujets portent des lunettes, des moustache, etc), avec des teintes de peau différentes, ainsi que des variations importantes en pose et en échelle.

Dans un premier temps, l'objectif est de montrer les résultats de reconnaissance obtenus dans des images statiques. Nous verrons plus loin que la reconnaissance est toujours

présente dans le cas du suivi qui est cependant traité à part (voir §3.3.7).

D'abord, nous faisons une description de la base d'images utilisée, tant pour l'apprentissage du modèle que pour le test. Ensuite, tous les éléments de configuration, tels que les opérateurs, le nombre de cellules, les niveaux de résolution, etc., sont définis. Après, une description détaillée est donnée pour l'apprentissage du positionnement des cellules ainsi que la réduction du modèle. Plus tard, nous passons au test de l'algorithme de reconnaissance en donnant, dans la mesure du possible, des exemples montrant des résultats intermédiaires lors de la phase de reconnaissance. Une attention particulière est donnée aux effets issus de la focalisation. Enfin, une analyse des résultats est donnée.

3.3.1 Bases d'images

Pour ce test, nous avons utilisé deux bases d'images. D'une part, l'ARdatabase[1] [71] est utilisée pour l'apprentissage du modèle de visage. D'autre part, la Caltech database [38] est utilisée pour la phase de reconnaissance du fait que les visages apparaissent dans différentes positions dans l'image, elles présentent des variations en taille et avec un fond complexe. Voici une description des deux bases :

- **Base d'images pour l'apprentissage : l'ARdatabase.** La base d'images ARdatabase, a été créée par Aleix Martinez et Robert Benavente au Centre de Vision par Ordinateur (CVC) à l'université Autonome de Barcelone (UAB). La base de données est librement disponible pour les instances académiques et de recherche. Cette base contient environ 4000 images en couleur, de taille 128×128 pixels, correspondant aux visages de 126 personnes (70 hommes et 56 femmes). Les images présentent des vues frontales avec différentes expressions faciales, conditions d'éclairage, et occultations partielles. Le visage est centré et plus ou moins normalisé en taille. SUr la figure 3.5 nous montrons quelques exemples de l'ARdatabase utilisés pour l'apprentissage.
- **Base d'images pour le test : Human face Caltech database[2].** Cette base a été construite par Markus Weber à l'Institut Technologique de Californie (CalTech). La base de données correspond à une base annotée et elle est composée de 450 images de visages de 27 sujets différents (hommes et femmes), dont la taille de chaque image est de 896×592 en format Jpeg. Les images sont prises sous différentes conditions d'éclairage, avec différentes expressions faciales et fond complexe. La figure 3.6 présente quelques exemples des images composant la base de données et utilisées pour tester l'algorithme de reconnaissance.

[1] La base d'images peut être obtenue à partir du cite web http ://cobweb.ecn.purdue.edu/RVL/ARdatabase/ar.html

[2] La base d'images peut être téléchargé gratuitement du site web http ://www.robots.ox.ac.uk/vgg/data3.html.

FIG. 3.5 – Exemples de visages appartenant à l'ARdatabase utilisés lors de l'apprentissage du modèle.

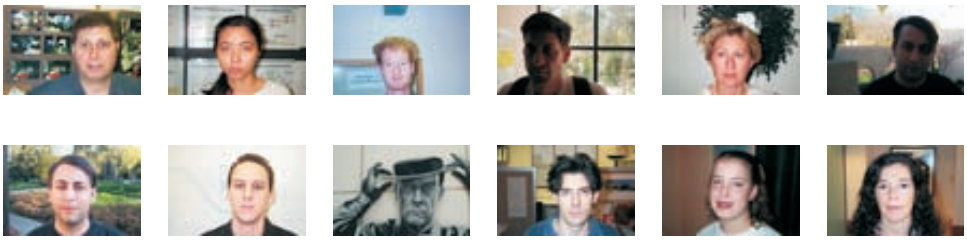


FIG. 3.6 – Exemples des images, appartenant à la Caltech database, utilisées pour le test de notre algorithme.

3.3.2 Préparation (set-up)

Afin de laisser notre système apprendre le modèle de visage, il est nécessaire de donner au préalable quelques éléments indispensables comme : les opérateurs qui seront utilisés pour le traitement primaire (bas niveau), le nombre de résolutions auxquelles l'image est décomposée, la définition de la grille de cellules, etc. Ces éléments sont décrits par la suite :

– **Opérateurs bas niveau.** Pour représenter le visage, nous avons choisi d'utiliser deux opérateurs bas niveau : l'analyse fréquentielle à partir de la carte d'orientations par filtrage de Gabor (voir §3.2.1) et le niveau de gris moyen (voir §3.2.2). En effet, les visages présentent des caractéristiques particulières en orientation du contour (des contour latéraux verticaux, des traits horizontaux pour les yeux et la bouche, etc.) ainsi qu'en niveaux de gris[3].

– **Grille de cellules.** La grille de cellules utilisée pour cet exemple prend de l'information à trois niveaux de résolution différente en ayant 32×32, 16×16 et 8×8

[3]Même si le niveau de gris est hautement variable parmi les visages, nous pouvons constater que celui-ci, dans différentes régions locales, est hautement corrélé pour un même visage.

cellules respectivement. Cela veut dire qu'on échantillonne de l'information dans 1344 régions différentes dans l'image. Ainsi donc, du fait de l'utilisation de deux opérateurs pour extraire des caractéristiques de l'objet, le vecteur de paramètres ς, pour chaque cellule $\mathcal{C}$, est composé de 4 éléments : $\varsigma = [\theta, e, \mathbf{a}^t]^t$. Où θ correspond à l'orientation du filtre de Gabor, e décrit le niveau de gris moyen à l'intérieur de la cellule, et $\mathbf{a} = [u, v]^t$ correspond aux coordonnées de chaque cellule.

Ainsi donc, nous disposons de :

- deux opérateurs bas niveau ($N = 2$) : Op_θ, Op_e, correspondant à l'extracteur de l'orientation du contour et le niveau de gris moyen, respectivement,
- trois grilles de cellules correspondantes aux trois niveaux de résolution, où $R = 3$,
- $M^c = 1344$ cellules $\mathcal{C}_m$, $m \in [1, M^c]$, où $M_1 = 32 \times 32$, $M_2 = 16 \times 16$, $M_3 = 8 \times 8$ et $M^c = M_1 + M_2 + M_3$,
- un vecteur de paramètres $\varsigma = [\theta_m, e_m, \mathbf{a}_m^t]^t$, pour chaque cellule $\mathcal{C}_m$.

Passons alors à l'apprentissage du modèle.

3.3.3 Apprentissage du modèle

Le but de cette étape est de fournir un modèle statistique de l'objet que l'on veut reconnaître. Pour ce faire, nous devons suivre les étapes décrites en §2.3, avec :

- $N = 2$: le nombre des opérateurs (carte d'orientations et niveau moyen de gris),
- $M^c = 1344$: nombre des cellules au total (pour tous les niveaux de résolution),
- $R = 3$: nombre d'échelles d'analyse.

La couleur n'étant pas utilisée, l'image est préalablement convertie sur 256 niveaux de gris.

3.3.3.1 Positionnement des cellules

La base de données que l'on utilise pour l'apprentissage correspond à une base normalisée en taille (voir §3.3.1). Nous devons suivre l'étape décrite en §2.3.3.2 afin d'introduire des invariants au changement d'échelle et de position.

Ainsi, si l'on applique une transformation de similitude au vecteur de coordonnées $\mathbf{a}0$, définissant la position des cellules dans l'imagette, le vecteur $\mathbf{a}_m = [u_m, v_m]^t$ de la position de la cellule m dans l'image sera donné par :

$$\begin{bmatrix} u_m \\ v_m \\ 1 \end{bmatrix} = \begin{bmatrix} s_u\, cos(\phi) & -s_v\, sin(\phi) & \frac{t_u}{2^r} \\ s_u\, sin(\phi) & s_v\, cos(\phi) & \frac{t_v}{2^r} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u0_m \\ v0_m \\ 1 \end{bmatrix}$$

s_u et s_v correspondent au facteur d'échelle, ϕ est l'angle de rotation, t_u, t_v sont les translations en u et v, respectivement. Le paramètre $r \in [0,2]$ correspond au niveau de résolution auquel l'image sera analysée (voir §3.3.3.2 pour la reconnaissance d'objets avec grandes variations en échelle).

Si l'on considère une valeur petite pour ϕ, nous avons :

$$\begin{bmatrix} u_m \\ v_m \\ 1 \end{bmatrix} \approx \begin{bmatrix} s_u & 0 & \frac{t_u}{2^r} \\ 0 & s_v & \frac{t_v}{2^r} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u0_m \\ v0_m \\ 1 \end{bmatrix} = \begin{bmatrix} s_u\, u0_m + \frac{t_u}{2^r} \\ s_v\, v0_m + \frac{t_v}{2^r} \\ 1 \end{bmatrix}$$

Ainsi, $\mathbf{a} = g(\mathbf{a}0, s_u, s_v, t_u, t_v) = g(\mathbf{a}0, \mathbf{t})$.

Du fait que l'on considère que le vecteur $\mathbf{a} = [\mathbf{a}_1^t\ \mathbf{a}_2^t\ \ldots\ \mathbf{a}_m^t]^t$ suit une loi normale, le vecteur moyen $\bar{\mathbf{a}}$ est donné par $\bar{\mathbf{a}} = g(\bar{\mathbf{a}}0, \bar{\mathbf{t}})$, avec $\bar{\mathbf{t}} = [1, 1, \frac{t_{u_0}}{2^r}, \frac{t_{v_0}}{2^r}]^t$, où (t_{u_0}, t_{v_0}) est le centre de l'image. La matrice de covariance $\Sigma_{\mathbf{a}}$ du vecteur $\mathbf{a}$ est, quant à elle donnée par :

$$\Sigma_{\mathbf{a}_m} = \mathbf{J}_g \Sigma_{\mathbf{t}} \mathbf{J}_g^t$$

Avec,

– $\Sigma_{\mathbf{t}}$: matrice de covariance du vecteur $\mathbf{t}$ définissant la connaissance que l'on peut accorder à $\mathbf{t}$. Du fait que l'on a considéré l'indépendance totale entre les paramètres $\mathbf{t}_u$ et $\mathbf{t}_v$, et $s_u = s_v$, on pourra choisir $\Sigma_{\mathbf{t}}$ comme suit :

$$\Sigma_{\mathbf{t}} = \begin{pmatrix} \sigma_{s_u}^2 & \sigma_{s_u}^2 & 0 & 0 \\ \sigma_{s_u}^2 & \sigma_{s_u}^2 & 0 & 0 \\ 0 & 0 & (\frac{\sigma_{t_u}}{2^r})^2 & 0 \\ 0 & 0 & 0 & (\frac{\sigma_{t_v}}{2^r})^2 \end{pmatrix}$$

Les deux facteurs d'échelle s_u et s_v sont considérés comme égaux et la dispersion en échelle $\sigma_{s_u} = \sigma_{s_v} = 0,12$ est telle que, pour une image donnée, l'objet le plus grand et le plus petit que l'on pourra détecter sont compris dans l'intervalle d'échelle 2 :1. $\sigma_{\mathbf{t}_u}$ et $\sigma_{\mathbf{t}_v}$ sont choisis de telle façon que toute l'image soit couverte. Dans notre cas nous les avons fixés à $\sigma_{\mathbf{t}_u} = 100$ pixels et $\sigma_{\mathbf{t}_v} = 150$ pixels.

– $\mathbf{J}_g$: matrice jacobienne de la fonction g définie par :

$$\mathbf{J}_g = \begin{pmatrix} u0_1 & 0 & \frac{1}{2^r} & 0 \\ 0 & v0_1 & 0 & \frac{1}{2^r} \\ \vdots & \vdots & \vdots & \vdots \\ u0_m & 0 & \frac{1}{2^r} & 0 \\ 0 & v0_m & 0 & \frac{1}{2^r} \end{pmatrix}$$

3.3.3.2 Modélisation d'objets avec grandes variations en échelle

Afin de reconnaître des objets avec des grandes variations en échelle (par exemple des objets dont l'échelle varie entre 128 et 1024 pixels), nous faisons une décomposition pyramidale de l'image d'entrée.

Dans le paragraphe §3.3.3.1, nous avons décrit l'apprentissage sur le positionnement des cellules où les images d'entraînement étaient de taille 128×128 pixels. Avec la formulation proposée, l'intervalle d'échelle qui est pris en compte par le modèle (à trois fois l'écart type d'échelle $\sigma_{s_u} = \sigma_{s_v}$) est compris entre 82 et 174 pixels (la taille moyenne de l'objet étant égale à 128 pixels). De cette manière, nous sommes capables de reconnaître un objet appartenant à cet intervalle d'une façon continue. Il faut remarquer que dans cette formulation, nous supposons que le traitement bas niveau est quasiment invariant d'une échelle à l'autre. Donc, si nous faisons une décomposition pyramidale de l'image d'entrée, nous serons capables de reconnaître des objets de taille supérieure. Ainsi, avec seulement trois échelles d'analyse différentes (en faisant le sous échantillonnage d'un facteur puissance 2), nous serons capables de reconnaître des objets d'une façon continue dont la taille varie entre 128 et 1024 pixels.

La stratégie que l'on propose est la suivante : lancer d'abord l'algorithme de reconnaissance dans l'échelle la plus grossière (sous-échantillonnée par 2^2), et continuer sur les images de résolution plus élevée. Éventuellement, on pourrait inhiber la zone dans l'image où un objet a été détecté et lancer la recherche sur les zones restantes ; de cette façon, on pourrait éviter la recherche coûteuse dans des échelles de résolution plus élevée. S'il y a un objet détecté, cette zone est inhibée et pour chaque niveau d'analyse, on lance l'algorithme de reconnaissance avec le modèle initial issu de la phase d'apprentissage. A titre d'exemple et pour une image donnée, sur la figure 3.7 on montre les versions sous-échantillonnées avec le modèle initial superposé. Le modèle affiché correspond seulement ici aux quatre coins de la boite englobante du modèle de taille moyenne égale à 128 pixels. Les carrés jaune et rouge correspondent respectivement à la taille de l'objet le plus petit et le plus grand qui peut être détecté. Les ellipses décrivent l'intervalle de confiance où les quatre coins peuvent être localisés. Pour cet exemple et afin d'avoir une meilleure compréhension, nous avons quasiment supprimé la variation en position de l'objet ($\sigma_{t_u} = \sigma_{t_v} \approx 5$

pixels).

Sur la figure 3.8, sont affichés les trois modèles superposés (sur-échantillonnés à l'échelle correspondante) afin de montrer comment l'espace-échelle est couvert par les trois niveaux d'analyse. Avec ces trois échelles, les objets qui peuvent être détectés sont compris dans l'intervalle $(128, 1024)$ pixels.

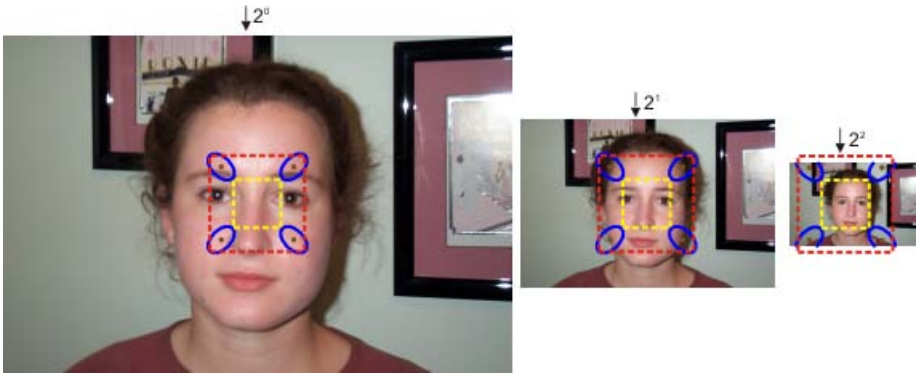


FIG. 3.7 – Exemple de la décomposition pyramidale pour une image donnée. Le modèle (de quatre coins ici) est superposé sur l'image montrant la taille de l'objet le plus petit (carré jaune) et le plus grand (carré rouge) qui peut être détecté. Les ellipses montrent l'intervalle de confiance où les quatre coins peuvent être localisés.

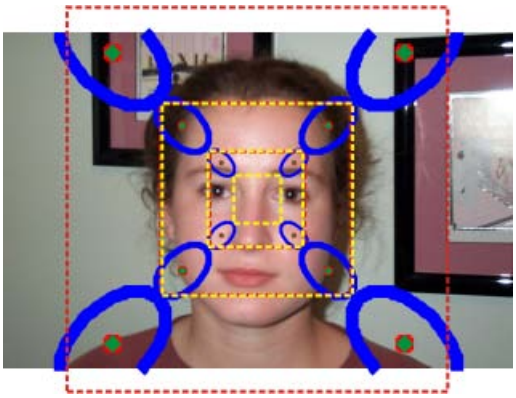


FIG. 3.8 – Le modèle de coins sur-échantillonné et superposé sur l'image originale. On observe comment, avec seulement trois échelles d'analyse, l'espace-échelle est couvert complètement. Dans cet exemple, l'objet le plus petit et l'objet le plus grand correspondent à une taille de 128 et 1024 pixels respectivement.

3.3.3.3 Réduction de dimensionnalité

Afin de réduire le nombre de paramètres du modèle initial, on fait la réduction de dimensionnalité (voir §2.3.4) en appliquant le critère de moindre variance de l'orientation θ du filtre (peu de variabilité de l'angle d'orientation du filtre qui a répondu dans une cellule spécifique). Les variances $\sigma^2_{\theta_i}$, pour $i \in [1, M^p]$, sont obtenues à partir de la diagonale de la matrice de covariance Σ_θ. Un critère adéquat peut être de laisser les orientations θ_m qui varient de moins de 15°. Donc, les paramètres qui seront éliminés du modèle $\mathcal{N}(\bar{\theta}, \Sigma_\theta)$ seront ceux pour lesquels $\sigma_\theta > 15°$.

Un autre critère est celui de prendre en considération le coefficient de corrélation r_{θ_i,θ_j}. Plus les paramètres sont corrélés, plus cela sert au modèle pour guider et contraindre la recherche d'autres paramètres. Comme décrit au paragraphe §2.3.4.1, il est intéressant de ne garder que les paramètres ayant une valeur de corrélation d'au moins $0,95$.

Une fois éliminés les paramètres non descripteurs dans chaque sous-modèle, $\mathcal{N}(\bar{\theta}, \Sigma_\theta)$ et $\mathcal{N}(\bar{\mathbf{e}}, \Sigma_\mathbf{e})$, on procède à l'élimination des cellules non fonctionnelles (dont aucun paramètre n'est un bon descripteur). Pour cela on élimine, du sous-modèle $\mathcal{N}(\bar{\mathbf{a}}, \Sigma_\mathbf{a})$, toutes les coordonnées des cellules dont tous les paramètres correspondants ont été éliminés. Pour donner un exemple de l'effet de la réduction de dimensionnalité par rapport au modèle initial, sur la figure 3.9 on présente le modèle réduit après élimination des cellules non fonctionnelles. Il faut remarquer qu'il correspond au modèle géométrique, c'est-à-dire les positions des *parties* de l'objet sont uniquement affichées et non les paramètres caractérisant chaque *partie*.

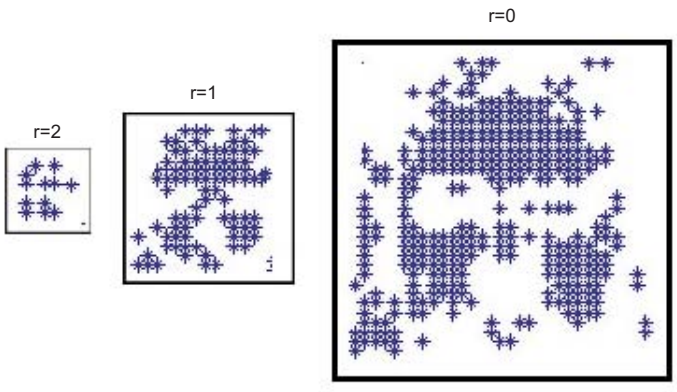


FIG. 3.9 – Modèle réduit du visage après la réduction de dimensionnalité. Le nombre de cellules qui restent, après l'élimination des cellules non discriminantes, est d'environ 300 sur 1344 au total. Il faut préciser que les cellules résultantes sont celles qui ont au moins un paramètre pertinent.

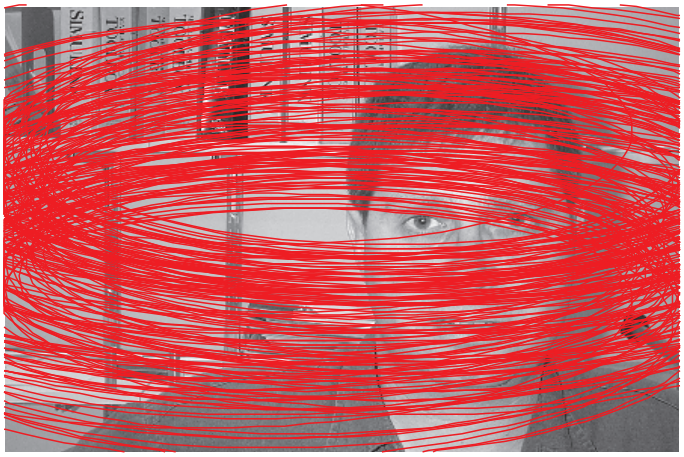


FIG. 3.10 – Intervalle de confiance (pour $r = 0$) sur la position de l'ensemble des *parties* après l'apprentissage.

A l'état actuel on dispose du vecteur d'apprentissage $\mathbf{x}$ qui suit une loi normale telle que $\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, \Sigma_{\mathbf{x}})$, avec $\bar{\mathbf{x}} = [\bar{\theta}, \bar{\mathbf{e}}, \bar{\mathbf{a}}]$ et

$$\Sigma_{\mathbf{x}} = \begin{pmatrix} \Sigma_{\theta} & 0 & 0 \\ 0 & \Sigma_{\mathbf{e}} & 0 \\ 0 & 0 & \Sigma_{\mathbf{a}} \end{pmatrix}$$

Ce modèle comprend les éléments suivants : l'apparence moyenne d'un visage décrit par les paramètres de direction de contour (θ) et du niveau moyen de gris (e), chacune des *parties* composant le modèle et la relation statistique existant entre elles (donnée par les matrices de covariance), le niveau de résolution de chaque *partie*, et l'indice de pertinence (de chaque *partie*) caractérisé par la statistique de réponse des opérateurs (voir §2.3.2).

Sur la figure 3.10, on montre l'intervalle permis pour la position des cellules après l'apprentissage.

3.3.4 Test de l'algorithme de reconnaissance : reconnaissance pas à pas

Le but de cette partie est de donner un exemple d'évolution et d'illustrer les résultats obtenus à chaque étape de l'algorithme proposé. Même si ce n'est pas une tâche facile, en raison de la nature récursive de l'algorithme de contrôle, on essaie de présenter quelques situations typiques pendant l'exécution de l'algorithme. Dans la mesure du possible, elles sont accompagnées d'illustrations qui montrent les effets de la focalisation. Ces effets sont : la réduction de l'espace de recherche dans le plan image, la sélection du niveau de résolution pour rechercher la *partie* en question, la réduction du nombre de candidats

possibles qui puissent correspondre à la *partie* qu'on cherche, ainsi que la gestion des opérateurs bas niveau.

3.3.4.1 Préparation et initialisation : niveau zéro

Une fois bien défini le modèle de l'objet $\mathcal{N}(\bar{\mathbf{x}}, \Sigma_{\mathbf{x}})$, on initialise l'état de reconnaissance au niveau zéro, $k = 0$. L'état k du processus correspond à la profondeur dans laquelle on se situe dans l'arbre de recherche. Dans ce niveau, le modèle à l'état $k = 0$, $\mathcal{N}(\bar{\mathbf{x}}, \Sigma_{\mathbf{x}})_0 = \mathcal{N}(\bar{\mathbf{x}}, \Sigma_{\mathbf{x}})$ correspond au modèle tel qu'il a été issu de la phase d'apprentissage, lequel nous donne les valeurs des caractéristiques les plus probables dans une situation courante. Cela implique la position, la taille et l'apparence *probable* du visage dans l'image.

3.3.4.2 Génération des hypothèses

Dans le paragraphe §2.4.3 nous avons souligné que la génération d'hypothèses revient à faire la hiérarchisation de primitives (ou *parties*) dans le processus de reconnaissance. Cette hiérarchisation nous permet de rechercher, en fonction des critères comme le caractère discriminant et le coût de détection, la *partie* la plus pertinente selon l'état du processus. A titre d'exemple, sur la figure 3.11 on montre la *partie* qui correspond le mieux aux critères en tant que coût de détection (typiquement correspondant à une primitive appartenant à la plus basse résolution), le caractère discriminant (moindre variance) et la capacité de faire converger le modèle plus rapidement (haut coefficient de corrélation).

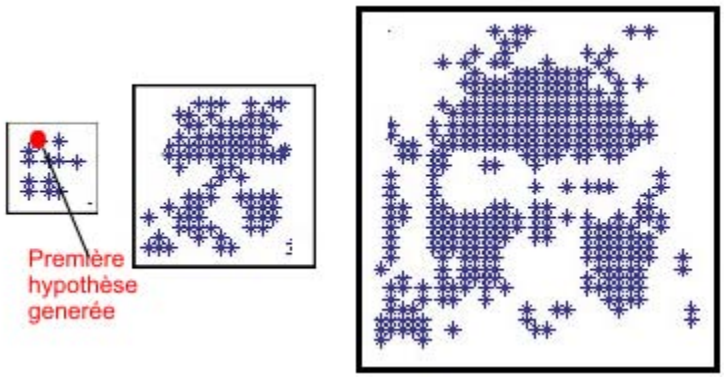


FIG. 3.11 – Résultat de la hiérarchisation des primitives : première hypothèse générée.

Dans cet exemple, l'hypothèse générée (correspondant plus au moins à la frontière entre les cheveux et le front du sujet) est décrite par les deux paramètres donnés au départ.

3.3.4.3 Définition de la région d'intérêt et détection

Une fois que l'on sait quelle *partie* nous allons rechercher, nous procédons à la délimitation de l'espace de recherche. La fenêtre d'intérêt est obtenue à partir du vecteur moyen $\bar{\mathbf{a}}_m = (\bar{u}_m, \bar{v}_m)^t$ (vecteur moyen des coordonnées de Λ_m) et de la matrice de covariance $\Sigma_{\mathbf{a}_m}$, obtenues lors de l'apprentissage. Ainsi, nous avons :

$$\begin{aligned}
\mathbf{p}_1 &= (\bar{u}_m + 2\sigma_{u_m}, \bar{v}_m - 2\sigma_{v_m}) \\
\mathbf{p}_2 &= (\bar{u}_m + 2\sigma_{u_m}, \bar{v}_m + 2\sigma_{v_m}) \\
\mathbf{p}_3 &= (\bar{u}_m - 2\sigma_{u_m}, \bar{v}_m - 2\sigma_{v_m}) \\
\mathbf{p}_4 &= (\bar{u}_m - 2\sigma_{u_m}, \bar{v}_m + 2\sigma_{v_m})
\end{aligned}$$

Où $\mathbf{p}_i,\ i \in [1,4]$, sont les quatre coins définissant l'imagette qui sera traitée par les N_m opérateurs. La ROI pour Λ_m dans l'espace géométrique est affichée suyr la figure 3.12. La définition de la ROI peut être interprétée comme une « modulation contextuelle » de l'image. La région plus claire correspond à la région la plus probable où la partie peut être trouvée, alors que la partie plus sombre correspond à la région la moins probable.

FIG. 3.12 – Région d'intérêt dans le seul espace géométrique, pour la position attendue de la première hypothèse émise.

Puisqu'il est probable qu'il existe plusieurs candidats pour la partie recherchée, une liste de tous ces candidats est construite, en mémoire à court terme, en mettant les détections en ordre du plus proche au plus loin du vecteur moyen $\bar{\lambda}_m$. L'observation $\hat{\lambda}_m$ issue de la détection de λ_m, est donc la première de la liste. Dans le cas où le candidat choisi n'est pas celui correspondant à la *partie* cherchée, il est nécessaire de revenir en arrière dans le processus de reconnaissance et d'essayer un autre candidat pour la même partie Λ_m. C'est pour cela qu'on a besoin de garder en mémoire l'indice (la position dans la liste) de chaque candidat testé afin d'éviter de revenir à la même détection plusieurs fois dans

le processus de reconnaissance. Cette idée d'une liste temporaire et de l'incrémentation de l'indice de détection équivaut à faire un mécanisme de IOR (inhibition de retour). Le fait de garder en mémoire tous les candidats pour une *partie* Λ_m, revient à éviter de traiter l'image, par les opérateurs bas niveau, à chaque fois qu'il est nécessaire de tester un autre candidat (parmi les L) pouvant être Λ_m.

3.3.4.4 Focalisation : mise à jour du modèle

Dans cette étape, l'objectif principal est d'utiliser et de profiter des détections réalisées précédemment afin de réajuster le modèle, notamment la position des *parties* et ses paramètres correspondants. L'espace de recherche pour des futures *parties* sera ainsi réduit. Du fait que nous considérons le processus de reconnaissance comme une tâche de recherche visuelle, la focalisation joue un rôle primordial pour l'optimisation d'un tel processus (voir §1.4 et §2.4.5).

Comme évoqué au chapitre II, la focalisation est réalisée à quatre niveaux différents : la focalisation dans le plan image (sélection de la région d'intérêt : dans quelle position de l'image doit être réalisée la recherche ?), la focalisation en résolution (niveau de détail : échelle fine ou échelle grossière ?), la focalisation dans l'espace de paramètres et dans l'espace des opérateurs (e.g. niveau de gris moyen et/ou direction du contour ?).

Par la suite nous illustrons les effets de la focalisation dans chacun des niveaux ; principalement en ce qui concerne la diminution du nombre de candidats potentiels dans la tâche de recherche.

Focalisation spatiale Dans ce niveau on répond à la question, quelle est la région dans l'image où chaque *partie* peut être potentiellement trouvée ?

L'information concernant la position probable et l'intervalle permis de chacune des parties Λ_m pour $m \in [1, M^p]$, est contenue dans le vecteur moyen $\bar{\mathbf{a}}_m$ et dans la matrice de covariance $\Sigma_{\mathbf{a}_m}$ du modèle appris (voir §3.3.4.3).

Ainsi donc, une fois réalisée la détection de la *partie* Λ_i de l'objet, l'estimation sur la position et l'intervalle de confiance, des autres *parties* restantes, sera obtenue par filtrage de Kalman (§2.4.5 Eq. 2.5). En effet, l'intervalle de confiance sur la position des *parties* est réduit par rapport à l'état initial, ce qui entraîne une focalisation dans le plan image.

Sur la figure 3.13 sont présentés les effets dérivés de la focalisation, après avoir pris en compte la détection courante. On observe comment la région d'intérêt initiale (Fig. 3.13-a), de chacune des *parties*, est réduite significativement après la mise à jour du modèle (Fig. 3.13-b).

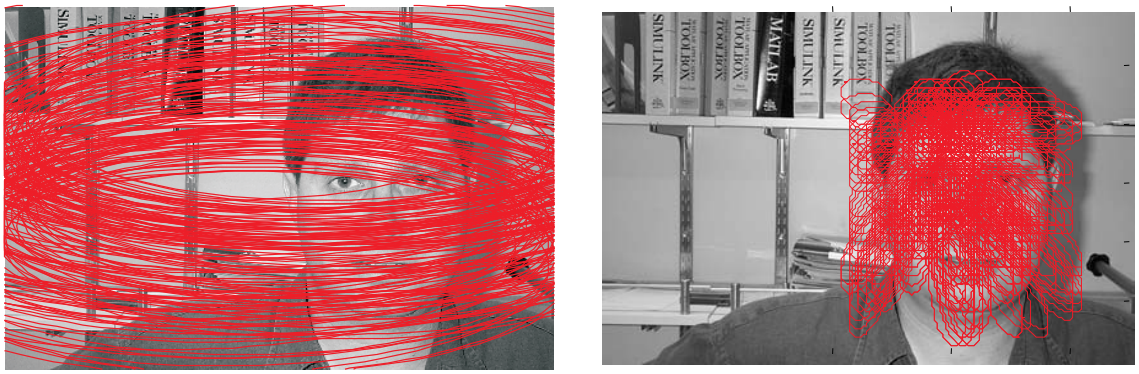


FIG. 3.13 – Chaque ellipse montre : a) les régions d'intérêt initiales pour toutes les *parties* de l'objet, b) les régions d'intérêt réduites après la détection de la première *partie* sélectionnée et la mise à jour du modèle.

L'intervalle permis après la mise à jour du modèle prend en compte, non seulement l'incertitude dérivée par les erreurs de détection, mais aussi celle dérivée par les variations en taille du visage. Ceci explique la direction et la largeur des axes des ellipses.

Focalisation en résolution Précédemment nous avons décrit les effets de la focalisation dans le plan image. En fait, la mise à jour du modèle géométrique n'implique pas seulement la diminution de l'espace de recherche dans le plan image, mais aussi celles du nombre de niveaux de résolution à traiter.

Afin de bien comprendre comment est réalisée cette focalisation, on doit faire référence à la phase de génération d'hypothèses (voir §3.3.4.2). La sélection de la *partie* la plus pertinente à rechercher, pour l'état courant du processus de reconnaissance, est fonction du coefficient de corrélation (vis-à-vis de la dernière *partie* qui a été détectée) et de la valeur de la variance (vis-à-vis des autres *parties* restantes). Il ne faut pas oublier que les *parties* qui composent l'objet, ne décrivent pas seulement la structure locale de l'image dans une seule résolution mais à différentes résolutions. Nous devons souligner que le fait de mettre à jour le modèle géométrique, après une détection donnée, implique une diminution de la région d'intérêt pour les *parties* restantes. La nouvelle *partie* sélectionnée (celle qui a la moindre variance) peut appartenir à n'importe lequel des trois niveaux de résolution, voilà pourquoi après la mise à jour du modèle implicitement on fait une focalisation en résolution. En conséquence, la recherche d'une *partie* donnée est effectuée, non seulement en sélectionnant une région d'intérêt dans le plan image, mais aussi en sélectionnant le niveau de résolution où la *partie* doit être recherchée.

Sur la figure 3.14 on montre trois exemples de la focalisation en résolution. Ici, la *partie* qui a été sélectionnée tout au début correspond à une *partie* qui appartient au niveau de résolution le plus bas. Le fait de réaliser la détection d'une *partie* qui appartient au niveau de résolution le plus bas, contraint la position probable des *parties* appartenant aux niveaux de résolution supérieurs. Cela permet de réaliser la reconnaissance d'objets invariante aux changements d'échelle d'une façon optimale.

A la différence d'autres approches où la recherche des primitives est faite d'une façon « coarse to fine » (de basses aux hautes fréquences, comme postule la théorie BHF [27]) dans notre cas le parcours en résolution n'a pas un ordre particulier : en dépendant de l'état courant du processus de reconnaissance, l'algorithme peut se focaliser sur une basse, moyenne ou haute résolution. L'importance d'un tel résultat réside dans le fait qu'il n'y a pas une contrainte sur la résolution dans laquelle le processus doit commencer à réaliser la tâche de recherche. On pourrait éventuellement ajouter des critères de disponibilité du temps et qualité de détection pour favoriser la préférence en sélection de résolution.

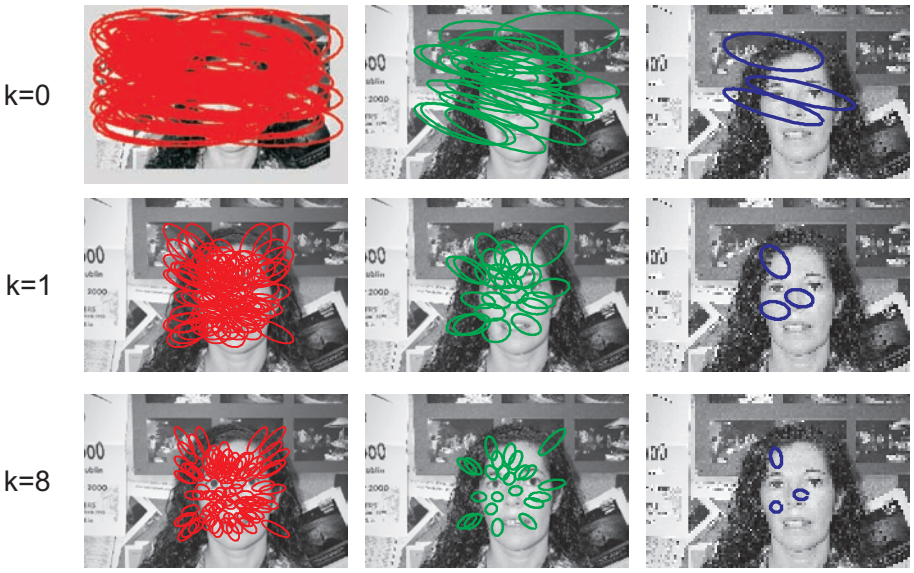


FIG. 3.14 – Sélection de la résolution. Chaque ligne représente les différents états du modèle lors de la reconnaissance. Les première, deuxième et troisième colonnes correspondent aux trois niveaux de résolution respectivement (de plus élevé au plus bas).

Focalisation dans l'espace des paramètres Concernant la focalisation dans l'espace de paramètres, on a deux niveaux de focalisation : celui correspondant à la **sélection d'un sous ensemble des N_m opérateurs** qui interviennent pour la détection d'une *partie*, et celui qui correspond à la focalisation sur l'**intervalle de réponse** de chacun des opérateurs.

– **Sélection des opérateurs.** Cette focalisation est issue de la *partie* qui est recherchée dans l'état courant du processus de reconnaissance. Étant donnée une *partie* sélectionnée et en fonction du nombre de paramètres qui composent cette *partie*, ce sera le sous ensemble d'opérateurs qui interviendront dans la détection de la *partie*. Grâce à cette focalisation il n'est pas nécessaire de traiter une région de l'image avec tous les opérateurs (qui ont été définis dans la phase d'initialisation), mais seulement le sous ensemble des N_m opérateurs qui interviennent pour la détection de cette *partie*.

– **Sélection de l'intervalle de réponse des opérateurs.** Elle est issue de la mise à jour de chacun des sous modèles (un sous modèle pour chaque paramètre qui compose la *partie* (voir §2.3.4.3)), après la détection d'une *partie*.

  De façon analogue à la focalisation dans le plan image, la focalisation dans l'espace des caractéristiques est réalisée avec la mise à jour de chacun des sous modèles après la détection d'une *partie*.

  Les modèles à mettre à jour dépendent des paramètres qui composent la *partie* qui a été détectée. Par exemple, si la *partie* qui a été détectée est composée seulement du paramètre de direction du contour, uniquement le sous modèle d'orientation du contour sera mis à jour (voir §2.4.5). Par conséquent, étant donnée la détection d'une *partie*, nous pouvons estimer les valeurs moyennes de chacun des paramètres des *parties* restantes, ainsi que les **nouveaux intervalles** de confiance.

Sur la figure 3.15-a on montre un exemple du résultat de la focalisation dans l'espace de paramètres après la détection d'une *partie*. Nous devons remarquer l'intérêt de la focalisation. En effet, grâce à la focalisation dans l'espace de paramètres, le nombre des candidats potentiels qui correspondent à la *partie* Λ_i, est réduit significativement par rapport au nombre de points d'intérêt extraits sans aucun a priori (voir 3.15-b). Dans cet exemple, on peut observer que la valeur de niveau de gris des *parties* est conditionnée par la détection précédente.

3.3.4.5 Décision

Jusqu'à présent, on a montré quelques étapes du processus de reconnaissance : l'initialisation, la génération des hypothèses, la définition de la région d'intérêt, la détection et la focalisation. Mise à part la phase d'initialisation, le reste des étapes est réitéré. L'arrêt de l'algorithme est une tâche confiée à la phase de décision. Dans §2.4.6, nous avons vu que cette décision est réalisée à partir de la caractérisation probabiliste de la présence de l'objet en sachant qu'un certain nombre de parties ont pu être détectées. L'évaluation de la présence d'un objet dans une région d'intérêt, est donnée par le rapport de vraisemblance

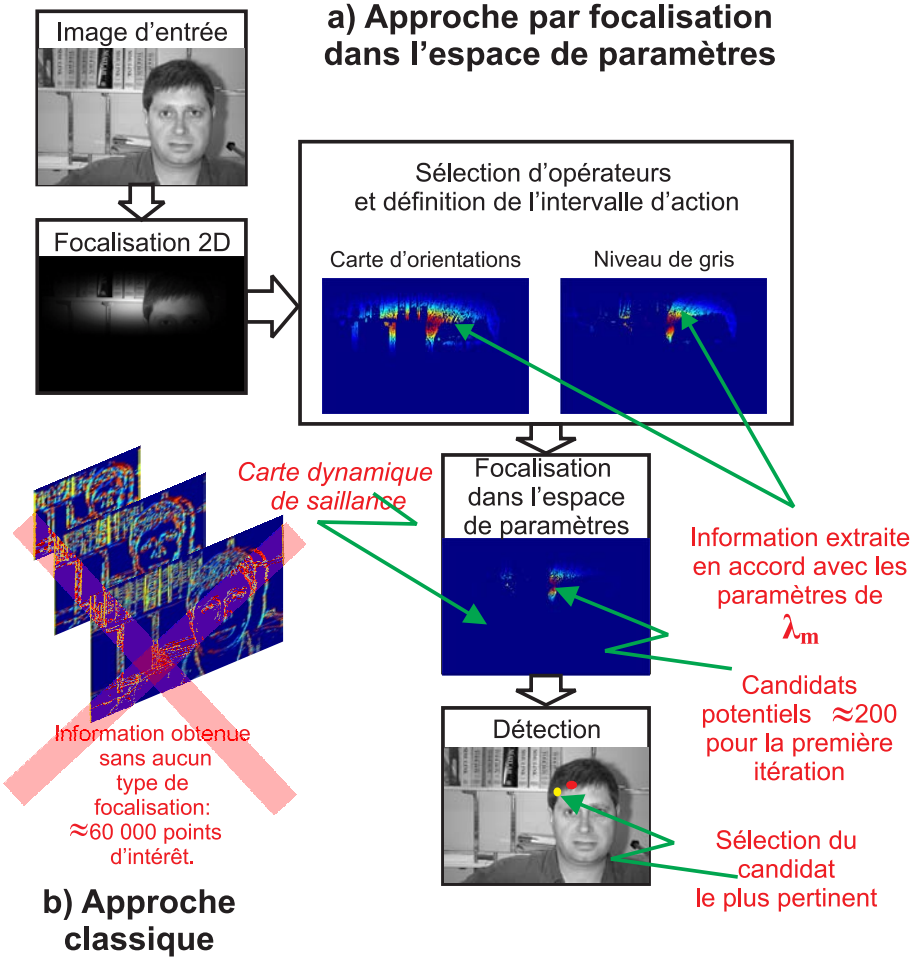


FIG. 3.15 – a) Exemple montrant les effets de la focalisation dans trois des quatre niveaux du spectre attentionnel : la sélection de la région d'intérêt dans le plan image, la sélection des opérateurs et la délimitation de l'intervalle de réponse des opérateurs impliqués. Les points d'intérêt, montrés dans la « carte dynamique de saillance », correspondent aux régions locales où les opérateurs de direction du contour et niveau de gris, ont répondu dans l'intervalle attendu pour la *partie* recherchée. Dans l'étape *détection*, le point jaune correspond au candidat le plus pertinent (couleur plus foncée dans la carte de saillance), alors que le point rouge correspond à la position réelle de la *partie* qui est cherchée. b) Points d'intérêt obtenus par une approche classique.

de la probabilité que l'objet soit présent sur la probabilité qu'il ne le soit pas en ayant un certain nombre des parties détectées (voir §2.4.6). Donc, dans cette partie on essaie de montrer le rôle et l'évolution du rapport de vraisemblance, $\mathcal{L}$ dans le processus de reconnaissance.

On se sert de $\mathcal{L}$ pour traiter le problème suivant : il s'agit de savoir s'il est pertinent ou non de continuer à rechercher d'autre parties dans une branche particulière de l'arbre de recherche. Pour cela, on doit définir un seuil inférieur τ_s pour lequel si $\mathcal{L} < \tau_s$ l'algorithme doit revenir en arrière dans le niveau de profondeur précédent et tester un autre candidat ou rechercher une autre partie. Pour cette application on a fixé une valeur de $\tau_s = -1,5$ (32 fois plus probable que ce soit un non objet plutôt qu'un objet). Si $\mathcal{L}$ atteint cette valeur, cela veut dire qu'il est plus probable que les parties détectées appartiennent à un non objet plutôt qu'à l'objet que l'on cherche.

Dans le cas où plusieurs *parties* ont été détectées il faut décider si, en ayant toutes ces détections, l'objet est présent dans la zone d'analyse. La décision de savoir si l'objet est présent ou non est donnée par un classificateur de type SVM. Ainsi, une fois que le modèle géométrique de l'objet est pratiquement bien positionné sur l'objet probable, le vecteur de caractéristiques, obtenu dans cette région de l'image, est évalué par le classifieur SVM. En fait, nous considérons que le modèle est bien positionné quand l'intervalle le plus petit, sur la position des toutes les *parties* restantes, est inférieur à une région de la taille de 5×5 pixels (pour la résolution la plus basse). Pour cela on doit faire la normalisation de l'imagette où le modèle est positionné, à la taille de 128x128 pixels ; de cette imagette on extrait le vecteur des caractéristiques de taille 1024 qui correspondent aux caractéristiques appartenant au troisième niveau de résolution (32x32 pixels). Si la sortie du classifieur est supérieure ou égale à un, alors l'objet est considéré reconnu.

3.3.4.6 Exemple d'évolution de l'algorithme

Un exemple d'évolution du processus de reconnaissance est montré sur la figure 3.16. Pour cet exemple, on a choisi de ne montrer que 9 itérations du processus. Ces itérations correspondent au cas où l'algorithme a trouvé la bonne hypothèse qui amène à l'objet.

Dans la première colonne (voir figure 3.16-b), l'ellipse correspond à la région d'intérêt sur la position probable de la *partie* en question, pour chaque état *k* du processus de reconnaissance. Dans la deuxième colonne est affichée la position de la détection la plus proche de la valeur moyenne à l'intérieur de l'ellipse (voir figure 3.16-c). Dans la troisième colonne (voir figure 3.16-d) est affiché le modèle mis à jour après la détection des *parties*. Nous pouvons observer comment, peu à peu, la région d'intérêt des *parties* manquantes est réduite après chaque mise à jour du modèle (voir figure 3.16-b), ainsi que l'adaptation du modèle à l'objet (voir figure 3.16-d).

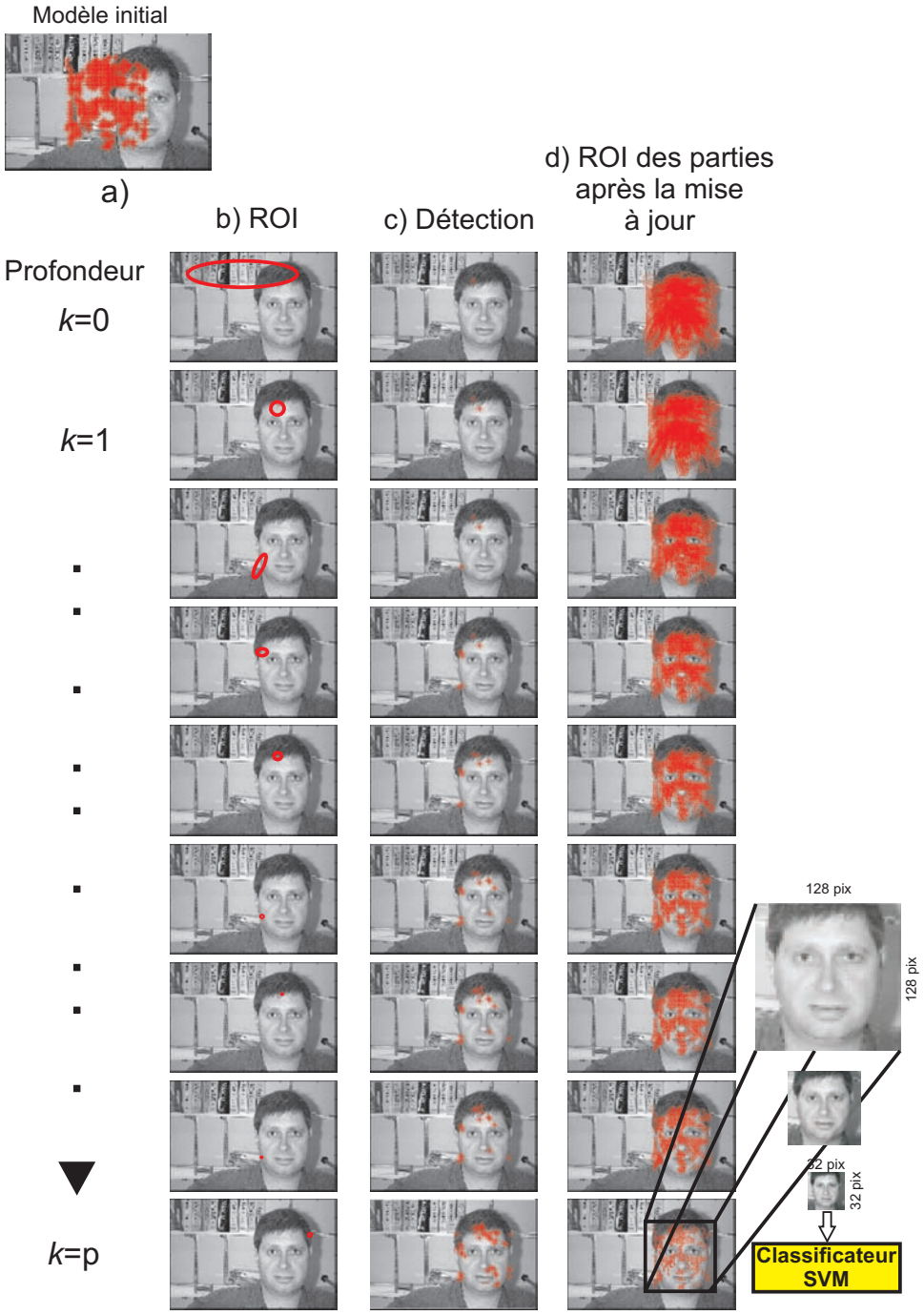


FIG. 3.16 – Cette figure montre un exemple d'évolution de l'algorithme proposé. a) Position moyenne des *parties* du visage. b) Région d'intérêt de chaque hypothèse générée, k étant l'état du processus de reconnaissance. c) Les détections réalisées dans la région d'intérêt préalablement définie. d) Les nouvelles régions d'intérêt estimées après la mise à jour du modèle. Une fois que le modèle est positionné sur le visage, la classification finale est faite avec un classificateur de type SVM.

Le fait de « cerner » de plus en plus le modèle, implique une augmentation du rapport signal/bruit (SNR) pour les détections ultérieures. Ceci du fait que le nombre de candidats possibles, pour une *partie* déterminée, est réduit significativement.

Afin de mieux illustrer la réduction de l'espace de recherche lors de détections, l'exemple précédent est obtenu en prenant la « bonne détection » qui amène à l'objet cherché. A la différence de cet exemple, sur la figure 3.17 on montre un autre exemple avec plusieurs hypothèses (mauvaises au départ) lors de la recherche du visage. On peut observer que l'algorithme tente d'adapter le modèle à la bonne taille du visage. Au départ (première image de la séquence) les détections effectuées amènent l'algorithme à tenter de reconnaître un visage plus petit que le visage présent dans l'image. Après plusieurs échecs lors de plusieurs tentatives, l'algorithme arrive à converger vers le visage en s'adaptant à la taille correcte de celui-ci (voir dernière image de la séquence).

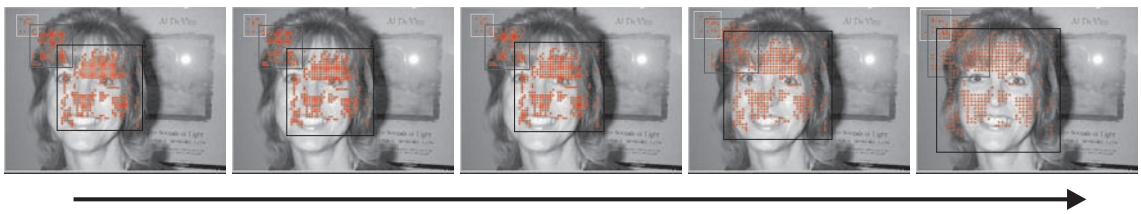


FIG. 3.17 – Exemple d'adaptation en échelle du modèle lors du test de plusieurs fausses hypothèses. La dernière image correspond à la reconnaissance réussie du visage. Les trois niveaux de résolution du modèle du visage sont affichés, ce qui explique les points rouges dans la partie supérieure gauche de l'image.

Déplacements de l'attention lors de la tâche de recherche visuelle Lors de la phase de reconnaissance, plusieurs hypothèses sont générées entraînant des déplacements de l'attention sur le plan image. L'ensemble de ces déplacements au cours du temps est ce qu'on appelle *le parcours du déplacement de l'attention*.

Un exemple d'un tel parcours est affiché sur la figure 3.18-b. Ce parcours correspond à la trajectoire suivie, après la recherche des 20 premières *parties*. Ces dernières ont été sélectionnées selon la hiérarchisation des *parties* à l'état courant du processus de reconnaissance. On peut observer que les parties recherchées correspondent d'abord au contour du visage mais qu'elles contiennent aussi l'information concernant le niveau de gris moyen d'un visage.

Dans cet exemple, toutes les *parties* n'ont pas été détectées. Cependant, l'algorithme est capable de continuer la recherche d'autres *parties* afin de mieux « cerner » le modèle et décider si l'objet est présent ou non. Concernant la vision biologique, on pourrait associer

ce type de déplacements avec la saccade oculaire chez l'homme (voir figure 3.18-a). Les observations locales pourraient être associées aux *fixations*.

Ici, les déplacements de l'attention sont guidés de deux façons (voir paragraphe §2.4.4) :

1. Top-down : d'après l'état courant k du processus de reconnaissance, une région d'intérêt où la présence de la *partie* Λ_i est probable, est définie dans l'espace des caractéristiques en fonction du vecteur moyen $\bar{\lambda}_i$ et sa dispersion Σ_{λ_i}. L'algorithme se sert du contexte et des connaissances a priori pour se focaliser sur cette région afin de rechercher la *partie* en question.

2. Bottom-up : une fois que la région d'intérêt est définie par le mécanisme de focalisation, il peut y avoir plusieurs candidats $\hat{\lambda}_h$, pour $h \in [1, H]$ pouvant être la *partie* cherchée. Ces points d'intérêt attirent l'attention du processus de reconnaissance. L'algorithme de détection choisit le candidat $\hat{\lambda}_h$ le plus proche de la valeur moyenne $\bar{\lambda}_i$ de λ_i.

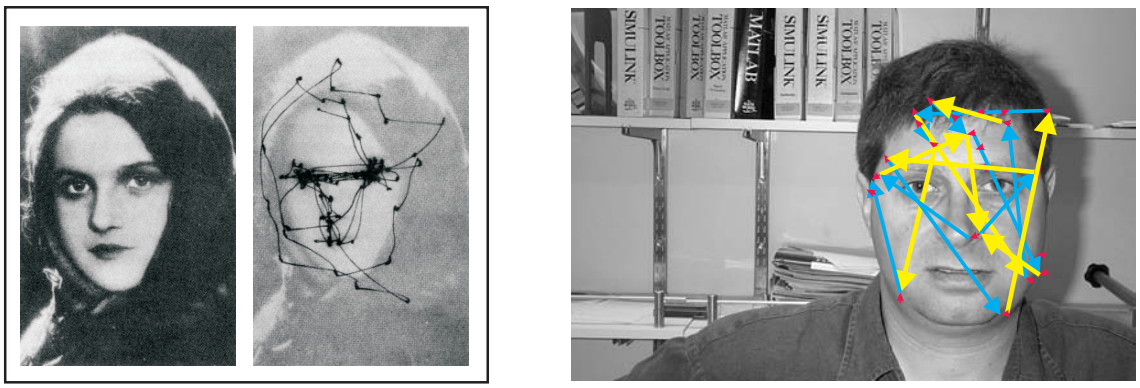


FIG. 3.18 – a) L'image à droite montre les traces du mouvement oculaire lors qu'un sujet explore le portrait à gauche (extrait du travail de A. Yarbus (1967) [125]. b) Exemple du parcours de déplacement de l'attention lors du processus de reconnaissance.

Pour un exemple donné, sur la figure 3.19, on montre plusieurs hypothèses testées pendant le processus de reconnaissance. Nous devons remarquer que seulement la *partie* cherchée est censée d'être détectée et non pas toutes les *parties* du modèle. On peut observer comment le modèle « saute » d'une position à une autre, en fonction de l'hypothèse générée, pour que le processus puisse trouver la meilleure configuration en accord avec celui-ci.

FIG. 3.19 – Différentes hypothèses testées. Pour chaque niveau de résolution, les points rouges correspondent à la position attendue des *parties* du visage. Lors de la recherche du visage, les « fixations » sautent d'une région à l'autre afin de tester plusieurs candidats et trouver la meilleure configuration possible en accord avec le modèle. La dernière image correspond à la détection correcte du visage.

3.3.5 Résultats : exemples de reconnaissance de visages

3.3.5.1 Modèle complet

Voici quelques exemples de reconnaissance de visages de douze personnes différentes (voir figures 3.20 et 3.21).

Les résultats présentés ci-après montrent une reconnaissance réussie du visage contenu dans l'image. Elles représentent un test significatif pour les algorithmes de mise en correspondance du fait de la présence d'un grand nombre d'éléments perturbants, surtout si l'on considère la simplicité des détecteurs de chaque *partie*. En fait, comme nous l'avons évoqué précédemment, nous n'avons pas cherché à développer des détecteurs très sophistiqués. Ceci entraîne un nombre important de candidats pour la première itération. Cependant cette faiblesse est fortement compensée par la focalisation dans l'espace de caractéristiques.

Bien que dans la majorité des images le modèle se place correctement sur le visage détecté, il y a des cas moins favorables comme le montrent les exemples Fig. 3.20-f-2,3. La dispersion de la position des *parties* reflètent l'adaptation du modèle par rapport à la taille du visage. Les visages présents dans ces images sont à différentes positions, avec des petites variations en échelle et avec luminosité variable (voir figure 3.21-d-1,2).

Pour ce test, le nombre maximal des « template matching » (classification SVM) lors du processus de reconnaissance est de 24. Le nombre d'itérations du processus est typiquement d'environ 50, mais il y a des cas où l'algorithme peut arriver à 250 ou 300 itérations pour une image donnée.

Même si l'algorithme montre des bons résultats par rapport à la reconnaissance et à la qualité[4] de la détection, il présente l'inconvénient du temps de calcul assez élevé. Ce temps de calcul élevé est dû principalement à la taille du modèle obtenu dans la phase d'apprentissage. Concernant le temps de calcul global, c'est dans la mise à jour du modèle que l'algorithme consomme le plus de ressources. Lors de l'apprentissage l'algorithme n'a gardé qu'environ 300 cellules les plus représentatives. Donc, pour le modèle géométrique (le modèle qui contient les coordonnées de l'ensemble de *parties*) nous avons un vecteur de taille 600 et la matrice de covariance associée de taille 600×600. La mise à jour d'une matrice d'une telle taille dans un processus récursif fait que l'algorithme devient trop lent (environ 1min par image, avec un programme Matlab© en utilisant un processeur pentium4-1GHz). Afin de diminuer le temps de calcul, une solution proposée est de réduire le modèle en ne gardant que les premières 50 *parties* les plus corrélées.

[4] La « qualité » concerne ici la précision de la position de l'objet dans l'image.

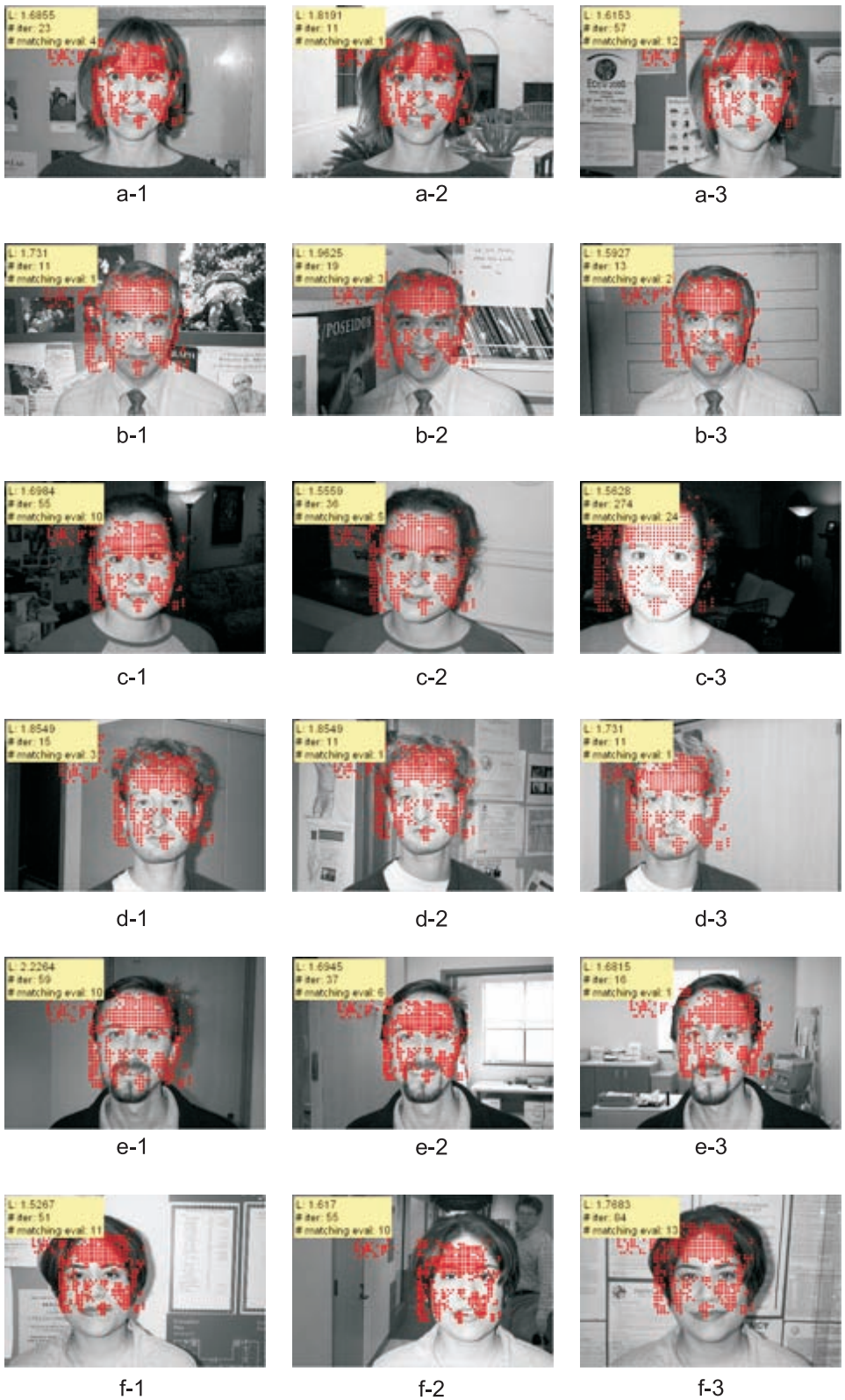


FIG. 3.20 – Exemples de reconnaissance de visages en utilisant le modèle tel qu'issu de la phase d'apprentissage (modèle complet).

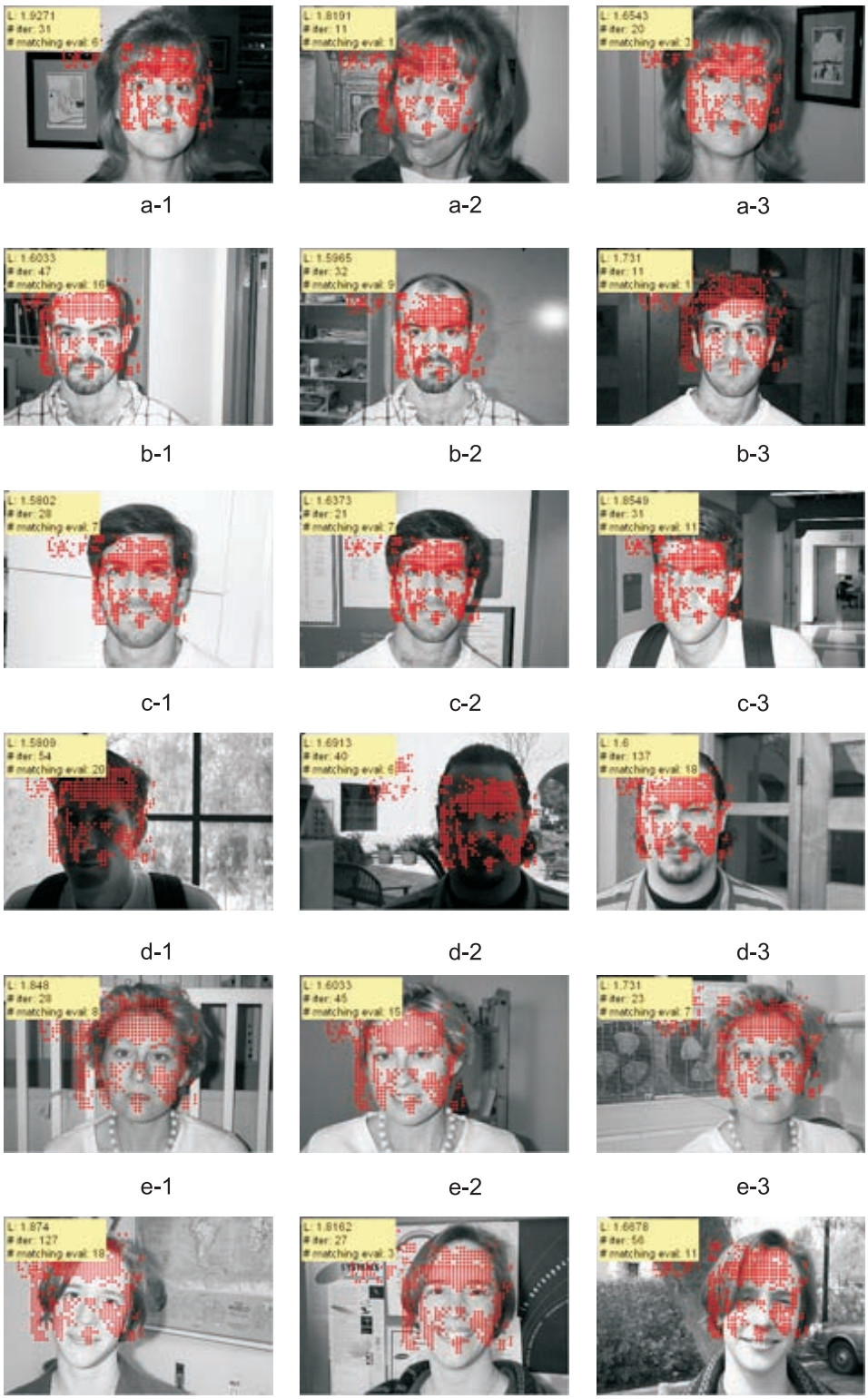


FIG. 3.21 – Exemples de reconnaissance de visages en utilisant le modèle tel qu'issu de la phase d'apprentissage (modèle complet).

3.3.5.2 Modèle réduit

Si on limite à 50 le nombre des *parties* composant l'objet, le temps de calcul chute significativement à environ 1s par image[5].

Sur la figure 3.22 on présente quelques exemples des images testées pour la reconnaissance de visages avec le modèle réduit. Pour cet exemple, la majorité des *parties* appartient au 2-ème niveau de résolution ($\downarrow 2^{r-1}, r = 2$). Donc, afin de mieux afficher les résultats, les images dans la résolution originale et sa version sous-échantillonnée sont présentées superposées. Les points rouges, affichés sur les images, représentent le modèle géométrique du visage dans les niveaux de résolution correspondants.

Dans les résultats obtenus, on peut observer que les visages ont été aussi bien détectés même si le modèle a été réduit significativement ; ceci suggère qu'on n'a pas besoin d'un modèle plus complexe pour avoir une reconnaissance correcte d'un objet. Les *parties* éliminées du modèle correspondent plutôt à celles appartenant au niveau de résolution plus élevé, d'ailleurs assez redondantes : e.g. des *parties* décrivant le niveau de gris moyen à l'intérieur du visage. Le critère de reconnaissance pour la décision finale est donné par le classificateur SVM. Les modèles pour la reconnaissance par appariement n'ont pas été touchés par rapport à l'expérimentation précédente.

3.3.6 Coût calculatoire

A la différence des approches comme le « template matching », ici le temps de détection de l'objet dans une image donnée n'est pas constant. Ceci du fait de la nature récursive de l'algorithme et des retours en arrière pendant le parcours de l'arbre de recherche. Dans le cas où l'algorithme a choisi la bonne hypothèse au départ, ce dernier peut converger rapidement vers l'objet en question (environ 15 itérations). En revanche, s'il faut parcourir plusieurs branches de l'arbre de recherche et revenir en arrière plusieurs fois avant d'arriver à la bonne solution (c'est d'ailleurs un phénomène typique lors de l'adaptation en échelle), l'algorithme peut prendre un temps élevé pour converger. Il est donc nécessaire de limiter le temps de détection afin d'arrêter la recherche de l'objet.

Pour le cas du deuxième test (modèle réduit), nous avons limité le temps de détection à 1 seconde. En fait, nous avons observé que dans la majorité des cas quand l'objet était présent, le temps de détection était inférieur à 1s. Quand l'algorithme ne détectait pas l'objet en moins d'une seconde, il ne le détectait non plus en 30 secondes.

[5]Implémentation en Matlab©avec un processeur pentium4-1GHz.

FIG. 3.22 – Exemples de reconnaissance de visages avec le modèle réduit. Uniquement les 50 *parties* les plus corrélées sont gardées. Avec ce modèle le temps de détection est d'environ 1s par image.

Parmi les objectifs de notre travail, un des plus importants est celui de l'optimisation du processus de reconnaissance : éviter le balayage du modèle partout dans l'image et à différentes échelles, et plutôt guider le processus de reconnaissance vers des zones probables où l'objet peut se trouver. Les résultats nous montrent qu'un visage peut être détecté en moyenne avec 46,25 mises à jour du modèle et 8,86 évaluations SVM pour la première expérmientation (modèle complet). Dans le tableau 3.1 sont regroupées les valeurs moyennes et la dispersion pour trois modèles avec un nombre différent de *parties* composant l'objet. On peut observer que les nombres de mises à jour et d'évaluations SVM sont extrêmement réduits par rapport au balayage exhaustif. Par exemple, pour une image de 640×480 pixels on devrait faire environ 6800 évaluations du vecteur de caractéristiques pour balayer l'image à quatre échelles différentes (pas de 0,25 comme facteur d'échelle) et avec une fenêtre de taille 128×128 pixels.

	Modèle 1	**Modèle 2**	**Modèle 3**
nombre de *parties*	300	150	50
taux de reconnaissance	79%	90.1%	92.3%
nombre de MAJ	46.25	38.28	18
nombre d'évaluations SVM	8.86	5.91	4.01

TAB. 3.1 – Moyenne du nombre d'itérations et d'évaluations SVM pour trois modèles de taille différente.

On doit remarquer que les résultats montrés dans le tableau 3.1 sont pour la détection d'un seul visage dans l'image. Dans le cas où il y a plusieurs visages, l'algorithme va détecter celui qui est le plus proche du modèle moyen. Si l'on veut détecter tous les visages dans une image donnée, il faut inhiber la zone où l'objet a été trouvé et relancer l'algorithme dans la zone restante.

Même si les nombres de mises à jour (MAJ) et d'évaluations SVM restent assez réduits, il faut remarquer que le temps total pour la détection reste grand[6] par rapport aux techniques typiques. Ce fait est dû principalement à deux raisons : la première est issue de la nature exploratoire de l'algorithme (des allers et retours sur plusieurs branches de l'arbre de recherche). Même si notre algorithme coupe assez tôt les mauvaises hypothèses et qu'il reste un nombre réduit de candidats potentiels (grâce à la focalisation dans l'espace de caractéristiques), la combinatoire reste assez élevée car le nombre des *parties* peut être d'environ 50 et le nombre de candidats par *partie* peut arriver jusqu'à 1000 pour le cas d'une hypothèse « mère » (zone d'analyse assez grande). La deuxième raison concerne la mise à jour du modèle. Comme on a expliqué dans le chapitre II, la mise à jour est effectuée par filtrage de Kalman. Un tel algorithme requiert l'inversion de la

[6]Ceci en considérant que notre approche détecte un seul visage à la fois pour une image donnée.

matrice de covariance du modèle. Pour cet exemple de reconnaissance de visage l'objet est caractérisé avec deux paramètres : la direction du contour et le niveau de gris. Si on analyse le pire des cas, c'est lorsque la *partie* qu'on cherche décrit l'objet avec ces deux paramètres. Dans ce cas particulier, on doit mettre à jour les sous modèles $\theta \sim \mathcal{N}(\bar{\theta}, \Sigma_\theta)$, $\mathbf{e} \sim \mathcal{N}(\bar{\mathbf{e}}, \Sigma_\mathbf{e})$ et $\mathbf{a} \sim \mathcal{N}(\bar{\mathbf{a}}, \Sigma_\mathbf{a})$...

Un autre facteur qu'il faut prendre en considération dans l'évaluation du temps de calcul est celui lié à l'intégration de multiples caractéristiques. Puisque toutes les *parties* ne décrivent pas toutes les caractéristiques d'un objet, le fait de représenter l'objet avec des caractéristiques multiples n'entraîne pas une augmentation significative de la taille du modèle. Donc, le temps de calcul n'augmente pas significativement. En revanche, si l'on intègre d'autres caractéristiques dans une approche du type « template matching », le vecteur peut grandir d'une façon significative et le temps calculatoire peut devenir prohibitif. L'impossibilité d'intégrer d'autres caractéristiques implique une représentation compacte. Le fait d'avoir une telle représentation de l'objet implique l'augmentation des fausses détections ou des non détections.

3.3.7 Définition d'un nouvel espace de recherche : suivi de l'objet

Quand il s'agit de faire la reconnaissance d'objets dans des séquences vidéo, nous pouvons profiter de la détection à l'image t pour contraindre la zone de recherche pour l'image $t+1$. En plus de gagner du temps de calcul, cela entraîne une amélioration du rapport signal sur bruit et par conséquente une diminution du nombre des fausses détections. Dans ce sens, dans la phase de suivi notre but est de fournir, à la procédure de reconnaissance pour l'image $t+1$, un plus petit intervalle de recherche que celui défini par la phase d'apprentissage initiale.

Ici, nous n'utilisons pas un modèle d'évolution et on applique une règle simple : après la reconnaissance de l'objet dans l'image à l'instant t, pour l'instant $t+1$ on initialise une nouvelle zone de recherche à partir du modèle mis à jour à l'instant t. La position attendue de l'objet sera la même que pour l'instant précédent mais on rajoute des erreurs sur la variation possible des *parties* en position et en échelle.

Ainsi, pour l'image suivante $t+1$, le nouvel espace de recherche géométrique est défini par $\bar{\mathbf{a}}(t+1) = \bar{\mathbf{a}}(t)$ et $\Sigma_\mathbf{a}(t+1) = \mathbf{J}_g \Sigma_\mathbf{t} \mathbf{J}_g^t + \Sigma_\mathbf{a}(t)$ (voir §2.3.3.2), où :

– $\Sigma_\mathbf{t}$: matrice de covariance du vecteur $\mathbf{t}$, donnée par :

$$\Sigma_{\mathbf{t}} = \begin{pmatrix} \sigma_{s_u}^2 & \sigma_{s_u}^2 & 0 & 0 \\ \sigma_{s_u}^2 & \sigma_{s_u}^2 & 0 & 0 \\ 0 & 0 & \sigma_{t_u}^2 & 0 \\ 0 & 0 & 0 & \sigma_{t_v}^2 \end{pmatrix}$$

– $\mathbf{J}_g$: matrice jacobienne de la fonction g (voir §2.3.3.2). Dans notre cas, $\sigma_{s_u} = \sigma_{s_v} = 0,024$, $\sigma_{t_u} = \sigma_{t_v} = 5$ pixels.

3.3.7.1 Résultats

Sur la figure 3.23 on présente des images appartenant à une séquence vidéo où l'objet est reconnu et suivi avec la méthode proposée. Dans cet exemple l'objet est présenté dans différentes positions, et avec des petites variations en échelle. Il faut remarquer que l'algorithme présente une robustesse aux occultations partielles de l'objet. Sur la figure 3.24 on montre un exemple où l'objet est observé avec une caméra avec zoom variable. Avec cet exemple on montre la capacité qu'a l'algorithme à s'adapter à la taille de l'objet lors de la phase de suivi.

3.3.8 Bilan

Dans cette section, la mise en place de la méthodologie proposée a été réalisée. Elle a été testée dans le cadre de la reconnaissance et suivi de visages.

Avec les résultats obtenus, nous avons montré la pertinence de l'approche qui semble bien adaptée aux tâches de reconnaissance et suivi d'objets. Des résultats très importants méritent d'être soulignés. D'abord, le nombre de fois où le classificateur SVM intervient, est réduit d'une façon significative avec une approche comme celle-ci. A la différence des approches faisant un balayage exhaustif en position et en échelle (environ 6800 évaluations du vecteur de caractéristiques pour une image 640×480 et à trois niveaux de résolution), ici le nombre d'évaluations est la plupart du temps inférieur à 20. Le classificateur SVM entre en jeu uniquement quand le modèle a « cerné » suffisamment l'objet en question. Ensuite, grâce à la focalisation à quatre niveaux différents : la sélection des opérateurs, l'intervalle d'action, la résolution et la région de l'image, il y a une grande diminution de la combinatoire implicite pour une approche de mise en correspondance. Ceci favorise aussi la diminution des fausses détections pendant la tâche de recherche. En outre, avec ce cadre, on n'a pas besoin d'avoir un module de suivi spécifique : ici, le suivi s'intègre d'une façon naturelle au processus de reconnaissance. Grâce au modèle « élastique » de l'objet et à la stratégie de reconnaissance, on est capable de détecter des objets variables en échelle d'une façon continue : trois échelles d'analyse suffisent pour reconnaître un objet de taille variable entre 128 et 1024 pixels.

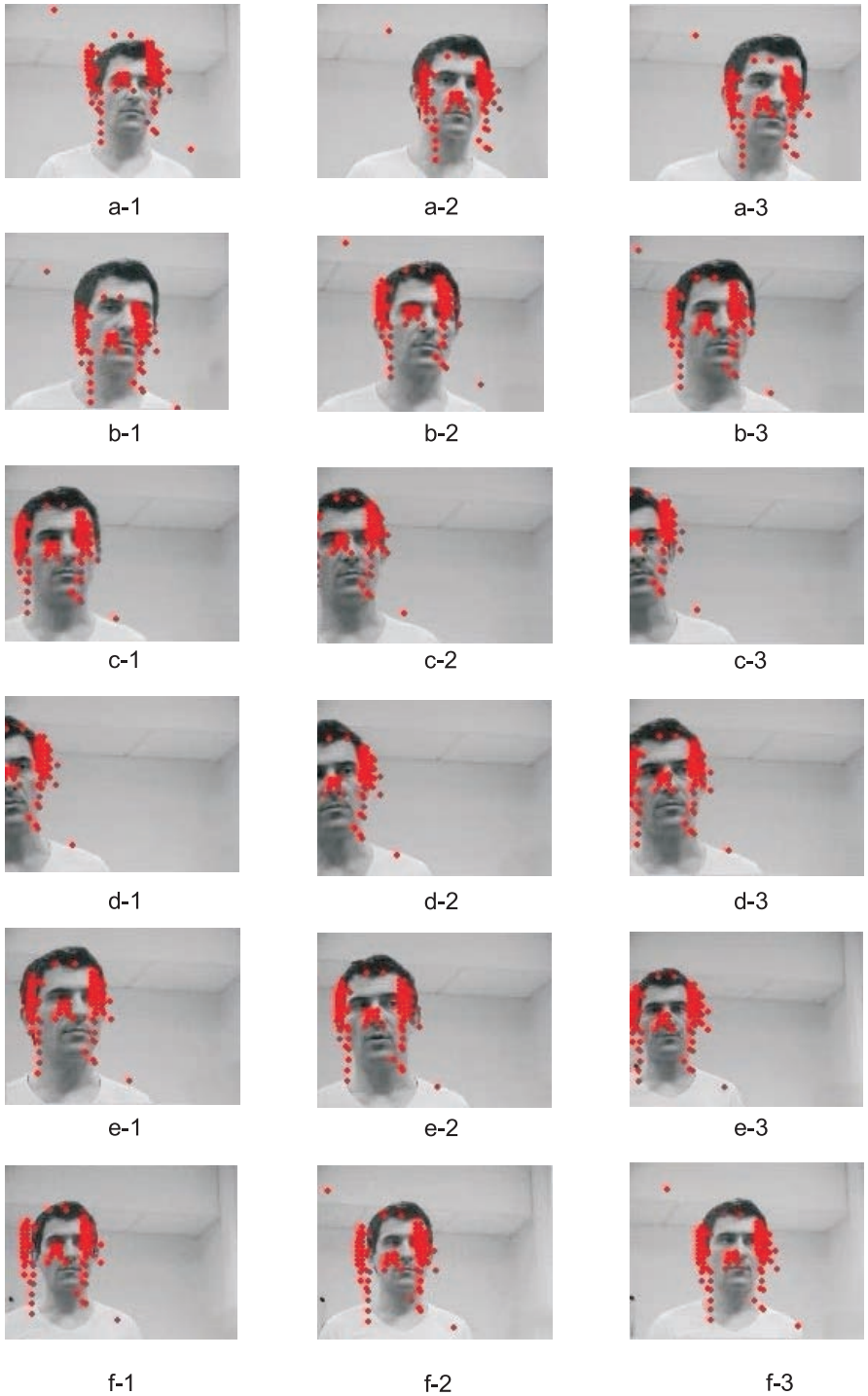


FIG. 3.23 – Test 1 : reconnaissance et suivi du visage se déplaçant dans le plan horizontal et vertical. La détection présente une certaine robustesse lors des occultations partielles (voir les lignes c et d de la séquence).

FIG. 3.24 – Test 2 : reconnaissance et suivi de visage se déplaçant en profondeur (adaptation en échelle).

Les problèmes observés pendant la reconnaissance (des non détections ou fausses détections) sont attribués principalement à la mauvaise représentation de l'objet (voir §4.2.1 pour une discussion plus détaillée sur ce sujet).

3.4 Reconnaissance, localisation et suivi de piétons

Afin de valider l'approche avec une autre classe d'objet, nous avons choisi la classe d'objet « piéton » pour tester notre algorithme. On s'intéresse ici à la détection des piétons dans des scènes réelles. Cette classe d'objet a la particularité d'être assez variable dans son apparence et avec des conditions d'éclairage hautement variables. De plus il existe une difficulté dans le fait que les piétons, dans une scène naturelle, peuvent apparaître à différentes tailles et positions dans l'image. Plusieurs approches comme [79, 82, 81] ont utilisé les piétons pour tester leurs algorithmes et montrent la complexité de la tâche.

Dans cette section nous allons procéder de manière différente que dans la section précédente. Nous ne reproduirons pas tous les aspects intermédiaires de notre méthodologie. En effet les résultats issus des étapes intermédiaires sont valables pour toutes les classes d'objets. Il serait donc inutile de reproduire ici ces mêmes résultats. En revanche notre objectif est de montrer les résultats de la reconnaissance, de la localisation et du suivi en prenant l'objet piéton comme deuxième exemple.

Dans cette partie on va se concentrer sur la préparation et l'initialisation de l'algorithme pour l'apprentissage du modèle de piéton. Pour la partie de reconnaissance on se limitera à ne donner que les résultats issus des tests.

3.4.1 Préparation avant l'apprentissage

– **Bases d'images.** Deux bases d'images ont été utilisées pour cette application. D'une part la « *MIT **cbcl** pedestrian database* » pour l'apprentissage du modèle du piéton, d'autre part une base constituée de séquences vidéo construites au sein de notre laboratoire pour le test de reconnaissance.

  – **Bases d'image pour l'apprentissage : *MIT cbcl pedestrian database***. Cette base a été créée au « Center for biological and computational learning (cbcl) » au MIT. Celle-ci a été l'objet de test pour différents méthodes pour la reconnaissance d'objets, notamment les travaux de Papageorgiou et al. dans un cadre général pour la détection d'objets [79, 82, 81]. La base est constituée de 924 images en couleur de taille 64x128 pixels, où le piéton est aligné au centre de l'image. Les piétons sont présentés en vue frontale et en vue de dos.

– **Bases d'image de test : séquences vidéo**. Pour la reconnaissance, la localisation et le suivi du piéton nous avons utilisé trois séquences vidéo enregistrées dans notre laboratoire. Les séquences présentent un ou deux piétons qui se déplacent sur le plan horizontal et/ou en profondeur (éloignement de la caméra). Également, une quatrième séquence (*ShopAssistant2cor*) appartenant au **CAVIAR video database** a été utilisée.

– **Préparation (set up) :** comme opérateur bas niveau nous avons utilisé les mêmes opérateurs que pour la modélisation de visages : orientation du filtre de Gabor par analyse de phase et niveau de gris moyen. Avec l'objectif de montrer l'apport de l'intégration de caractéristiques multiples (autres que fréquentielles), principalement pour le suivi, nous avons fait deux expérimentations : pour la première nous avons utilisé un modèle où l'objet est caractérisé par un seul opérateur (l'orientation des contours) ; pour la deuxième le modèle est caractérisé par deux opérateurs qui sont : la direction des contours et le niveau de gris. Chacune de ces expérimentations est décrite ultérieurement.

3.4.2 Apprentissage dans le repère « imagette »

Ici, nous nous sommes intéressés à l'analyse de deux cas différents pour la reconnaissance des piétons. Cela dans le but de mettre en valeur l'intégration de caractéristiques multiples dans le modèle. Pour ce faire, de manière analogue à l'application de l'objet visage, nous avons suivi toutes les étapes pour l'apprentissage des paramètres (voir §2.3) et l'obtention des *parties*. Cependant, nous distinguons des valeurs différentes concernant le nombre d'opérateurs bas niveau utilisés. Ainsi, deux modèles différents ont été appris par le système : le premier qui prend en compte seulement l'information fréquentielle, le deuxième caractérisé par deux paramètres (niveau de gris et fréquence). Pour cette application, nous avons utilisé une grille avec $M^c = 672$ cellules et trois niveaux de résolution, donc $R = 3$.

Comme résultat de l'apprentissage on obtient les modèles moyens de piétons montrés sur la figure 3.25.

3.4.2.1 Modèle avec l'information fréquentielle exclusivement

Dans ce modèle obtenu, on peut voir que les cellules qui demeurent, lors de l'apprentissage, correspondent bien au contour du piéton moyen. De l'ensemble de cellules données au départ (672 cellules), il ne reste qu'autour de 80 cellules caractérisées (*parties* composant l'objet). Il est important de garder à l'esprit, que les points représentés dans ce modèle initial résultant, n'indiquent que la position relative des *parties* de l'objet. Inversement, aucune information sur la direction de contour (correspondant à chaque *partie*)

FIG. 3.25 – Modèles du piéton après réduction de dimensionnalité (uniquement la position des *parties*). a) Modèle basé sur l'orientation du contour. b) Modèle basé sur l'orientation du contour et le niveau de gris.

n'est donnée.

3.4.2.2 Modèle avec information fréquentielle et niveau de gris

A la différence du résultat donné par le modèle précédent, on voit apparaître des informations supplémentaires, à la fois sur le niveau de gris à l'intérieur de la silhouette du piéton et sur l'extérieur de celle-ci. En effet, on voit apparaître dans ce modèle des *parties* représentées qui ne correspondant pas directement à l'objet piéton. Celles-ci, se situent aux alentours de l'objet et correspondent au fond de l'image. Cette information provient d'une identification du niveau de gris faisant référence aux éléments particuliers qu'on retrouve dans le fond d'une scène urbaine quelconque, i.e. de la chaussée, du mobilier urbain, etc. Il s'agit d'un résultat qui pose problème dans la mesure où l'objet au sens strict, n'est pas directement identifié. Cela revient à apprendre l'objet dans son contexte et pas seulement de manière isolée. L'apprentissage effectue les tâches qu'on lui attribue et ce biais vient de la composition de la base d'exemples. Ce problème de redondance du contexte ne se retrouve pas dans le cas de l'objet visage dont le fond de la base d'exemples est hautement variable (en structure et couleur/niveau de gris). Le nombre de *parties* comprises dans le modèle est d'environ 170.

A ce stade, nous avons appris les paramètres des *parties* composant l'objet piéton. Du fait que ces dernières ont été obtenues à partir des imagettes 64×128 pixels, la variation en position de chaque *partie* est nulle en sortie d'apprentissage. Donc, il est nécessaire d'apprendre les variations des *parties* en position et en échelle lorsque le piéton est dans la scène. La modélisation 3D de la scène est nécessaire pour la tâche de localisation. Cette modélisation est décrite dans la section qui suit.

3.4.3 Modélisation 3D, cas des piétons

3.4.3.1 Le problème

On suppose que l'on dispose d'un ensemble d'imagettes de taille fixe dans l'image à partir desquelles on a réalisé l'apprentissage de l'objet (voir §2.3.3.2).

Ce modèle d'objet dispose donc d'un ensemble de cellules $\mathcal{C}_i$ de positions définies par $\mathbf{a}_{0i} = (u_{0i}, v_{0i})^t$ fixes et connues. Le vecteur $\mathbf{a}_0$ représentant l'ensemble des positions de toutes les cellules sera donc constant et de matrice de covariance $\Sigma_{\mathbf{a}_0}$ nulle.

La question qui se pose à présent est la suivante : que deviennent les positions $\mathbf{a}_i$ des cellules lorsque l'objet est vu dans une position différente de celle de l'apprentissage ? Pour répondre à cette question, il faut donc connaître le nouveau vecteur $\mathbf{a}$ caractérisant la position des cellules dans l'image, ainsi que sa matrice de covariance Σ_a (voir §2.5).

3.4.3.2 Cas des piétons

Modélisation On considère ici le cas où l'on souhaite reconnaître des piétons à partir d'une caméra embarquée dans un véhicule. On supposera la caméra placée à une hauteur h du sol, et inclinée d'un angle α vers le bas (configuration classique pour des applications d'aide à la conduite).

Soit une cellule $\mathcal{C}_i$ de coordonnées dans l'image (u_{0i}, v_{0i}). Cette cellule est issue de la projection d'une cellule 3D supposée être issue d'un objet plan placé à une distance y_0 de la caméra.

On considérera que la position (u_{0i}, v_{0i}) est donnée dans un repère tel que montré en figure 3.26.

Ces cellules seront supposées provenir d'une banque d'objets plans situés tous à une distance Y_0 de la caméra.

On aura ainsi les relations suivantes :

$$u_{0i} = e_u \frac{X_{0i}}{Y_0} \qquad \text{et} \qquad v_{0i} = e_v \frac{Z_{0i}}{Y_0} \tag{3.5}$$

X_{0i}, Y_0 et Z_{0i} sont les positions de la cellule 3D projetée.

On cherche à déterminer la position (u, v) de la cellule $\mathcal{C}_i$ lorsque l'objet sera translaté de T_x, T_y et lorsqu'il sera vu par une caméra (de mêmes caractéristiques intrinsèques e_u et e_v pour simplifier) inclinée d'un angle α et montée à une hauteur h (voir figure 3.27).

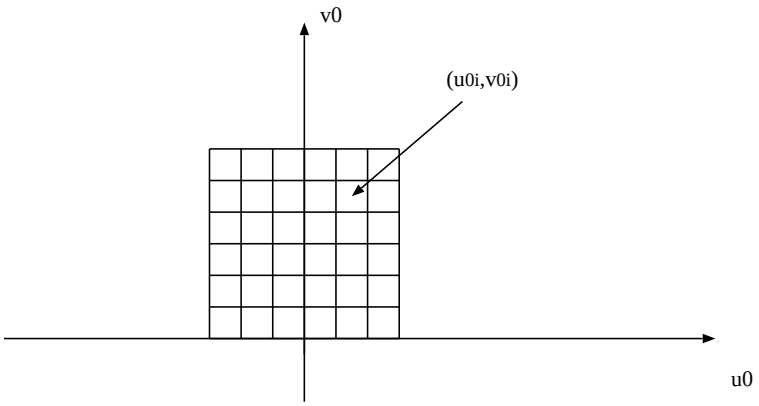


FIG. 3.26 – Repère cellules utilisé.

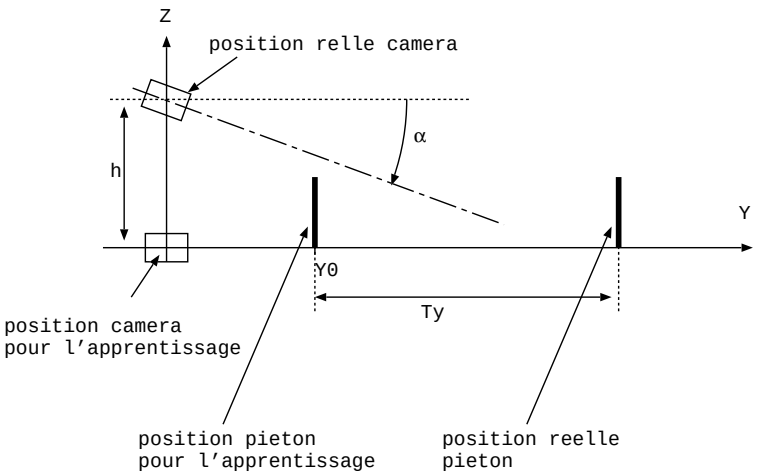


FIG. 3.27 – Paramètres utilisés pour la reconnaissance de piétons.

Dans ce cas, la projection d'un point 3D de position (X_{0i}, Y_0, Z_{0i}) se fera dans le repère caméra 3D (U, V, W) comme suit :

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} X_{0i} + T_x \\ Y_0 + T_y \\ Z_{0i} - h \end{pmatrix}$$

Soit, en considérant que α est faible :

$$\begin{pmatrix} U_i \\ V_i \\ W_i \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -\alpha \\ 0 & \alpha & 1 \end{pmatrix} \begin{pmatrix} X_{0i} + T_x \\ Y_0 + T_y \\ Z_{0i} - h \end{pmatrix}$$

les coordonnées de ce point dans l'image seront à présent :

$$u_i = e_u \frac{U_i}{V_i} \qquad \text{et} \qquad v_i = e_v \frac{W_i}{V_i}$$

soit :

$$u_i = e_u \frac{X_{0i} + T_x}{Y_0 + T_y + \alpha Z_{0i} - \alpha h} \qquad \text{et} \qquad v = e_v \frac{\alpha Y_0 + \alpha T_y + Z_{0i} - h}{Y_0 + T_y + \alpha Z_{0i} - \alpha h} \tag{3.6}$$

Les positions 3D des cellules lors de l'apprentissage peuvent être déterminées par la relation (3.5) :

$$X_{0i} = \frac{Y_0 u_{0i}}{e_u} \qquad \text{et} \qquad Z_{0i} = \frac{Y_0 v_{0i}}{e_v}$$

Soit, en substituant dans (3.6) :

$$u_i = \frac{Y_0 u_{0i} + T_x e_u}{Y_0 + T_y + \alpha \frac{v_{0i} Y_0}{e_v} - \alpha h} \qquad \text{et} \qquad v_i = e_v \frac{\alpha(Y_0 + T_y) + \frac{Y_0 v_{0i}}{e_v} - h}{Y_0 + T_y + \alpha \frac{v_{0i} Y_0}{e_v} - \alpha h} \tag{3.7}$$

On obtient ainsi une relation donnant la position $a_i = (u_i, v_i)$ de la cellule en fonction des translations T_x, T_y, h et de la rotation α de la caméra.

Ainsi on dispose de la relation f telle que :

$$\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_M^c)^t = f(\mathbf{a}_0, \mathbf{v}) \qquad \text{avec} \qquad \mathbf{v} = (T_x, T_y, \alpha, h)^t$$

La figure 3.28 montre divers cas obtenus.

La matrice de covariance $\Sigma_{\mathbf{a}}$ pourra être déduite de la matrice de covariance $\Sigma_{\mathbf{a}_0}$ et de celle définissant les erreurs sur les paramètres T_x, T_y, α et h (que l'on appellera Σ_V) par la relation :

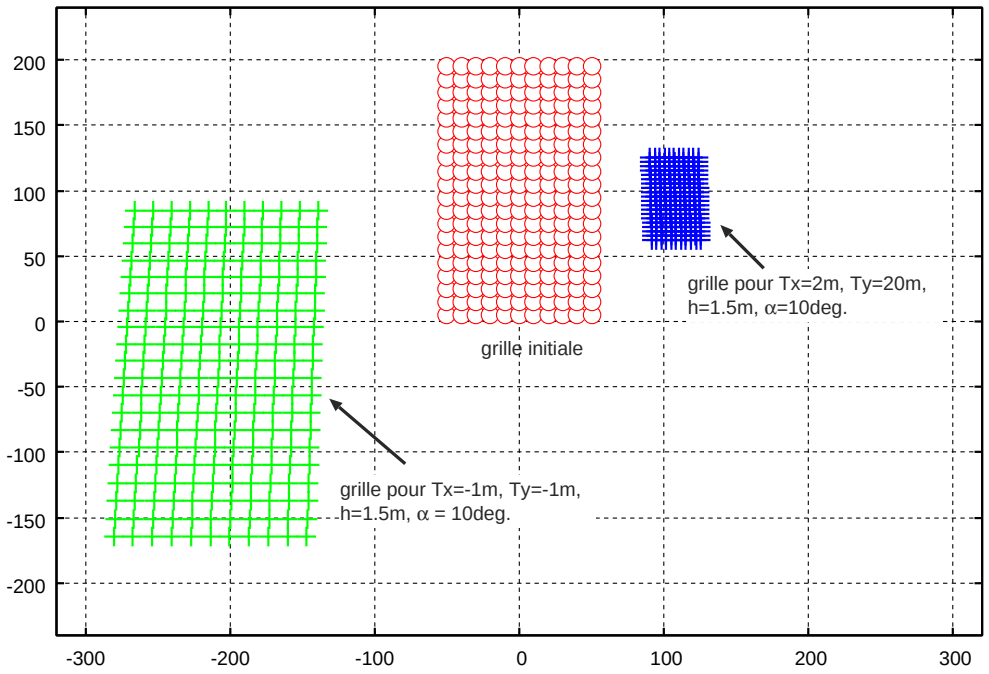


FIG. 3.28 – Exemple de grilles de cellules obtenues

$$\Sigma_{\mathbf{a}} = \mathbf{J}_{f_{\mathbf{a}}} \Sigma_{\mathbf{a}_0} \mathbf{J}_{f_{\mathbf{a}}}^t + \mathbf{J}_{\mathbf{v}} \Sigma_{\mathbf{v}} \mathbf{J}_{\mathbf{v}}^t$$

La matrice $\mathbf{J}_{f_{\mathbf{a}}}$ est la matrice jacobienne de la fonction f par rapport à chaque composante u_{0i}, v_{0i}. Comme ces grandeurs sont constantes, la matrice de covariance $\Sigma_{\mathbf{a}_0}$ sera nulle et l'expression deviendra :

$$\Sigma_{\mathbf{a}} = \mathbf{J}_{\mathbf{v}} \Sigma_{\mathbf{v}} \mathbf{J}_{\mathbf{v}}^t \tag{3.8}$$

avec $\mathbf{J}_V$ matrice jacobienne de f par rapport aux paramètres T_x, T_y, h et α. Elle sera donnée par :

$$\mathbf{J}_{\mathbf{v}} = \begin{pmatrix} \frac{\partial u_0}{\partial T_x} & \frac{\partial u_0}{\partial T_y} & \frac{\partial u_0}{\partial h} & \frac{\partial u_0}{\partial \alpha} \\ \frac{\partial v_0}{\partial T_x} & \frac{\partial v_0}{\partial T_y} & \frac{\partial v_0}{\partial h} & \frac{\partial v_0}{\partial \alpha} \\ \frac{\partial u_1}{\partial T_x} & \frac{\partial u_1}{\partial T_y} & \frac{\partial u_1}{\partial h} & \frac{\partial u_1}{\partial \alpha} \\ \frac{\partial v_1}{\partial T_x} & \frac{\partial v_1}{\partial T_y} & \frac{\partial v_1}{\partial h} & \frac{\partial v_1}{\partial \alpha} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial u_{M^c}}{\partial T_x} & \frac{\partial u_{M^c}}{\partial T_y} & \frac{\partial u_{M^c}}{\partial h} & \frac{\partial u_{M^c}}{\partial \alpha} \\ \frac{\partial v_{M^c}}{\partial T_x} & \frac{\partial v_{M^c}}{\partial T_y} & \frac{\partial v_{M^c}}{\partial h} & \frac{\partial v_{M^c}}{\partial \alpha} \end{pmatrix}$$

Soit :

$$\mathbf{J_v} = \begin{pmatrix} \frac{eu}{\frac{\alpha v_{01} Y_0}{ev}+Y_0+T_y-\alpha h} & -\frac{u_{01} Y_0+eutx}{\left(\frac{\alpha v_{01} Y_0}{ev}+Y_0+T_y-\alpha h\right)^2} & \frac{\alpha(u_{01} Y_0+eutx)}{\left(\frac{\alpha v_{01} Y_0}{ev}+Y_0+T_y-\alpha h\right)^2} & -\frac{(u_{01} Y_0+eutx)\left(\frac{v_{01} Y_0}{ev}-h\right)}{\left(\frac{\alpha v_{01} Y_0}{ev}+Y_0+T_y-\alpha h\right)^2} \\ \frac{eu}{\frac{\alpha v_{01} Y_0}{ev}+Y_0+T_y-\alpha h} & -\frac{u_{01} Y_0+eutx}{\left(\frac{\alpha v_{01} Y_0}{ev}+Y_0+T_y-\alpha h\right)^2} & \frac{\alpha(u_{01} Y_0+eutx)}{\left(\frac{\alpha v_{01} Y_0}{ev}+Y_0+T_y-\alpha h\right)^2} & -\frac{(u_{01} Y_0+eutx)\left(\frac{v_{01} Y_0}{ev}-h\right)}{\left(\frac{\alpha v_{01} Y_0}{ev}+Y_0+T_y-\alpha h\right)^2} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{eu}{\frac{\alpha v_{0i} Y_0}{ev}+Y_0+T_y-\alpha h} & -\frac{u_{0i} Y_0+euT_x}{\left(\frac{\alpha v_{0i} Y_0}{ev}+Y_0+T_y-\alpha h\right)^2} & \frac{\alpha(u_{0i} Y_0+euT_x)}{\left(\frac{\alpha v_{0i} Y_0}{ev}+Y_0+T_y-\alpha h\right)^2} & -\frac{(u_{0i} Y_0+euT_x)\left(\frac{v_{0i} Y_0}{ev}-h\right)}{\left(\frac{\alpha v_{0i} Y_0}{ev}+Y_0+T_y-\alpha h\right)^2} \\ \frac{eu}{\frac{\alpha v_{0i} Y_0}{ev}+Y_0+T_y-\alpha h} & -\frac{u_{0i} Y_0+euT_x}{\left(\frac{\alpha v_{0i} Y_0}{ev}+Y_0+T_y-\alpha h\right)^2} & \frac{\alpha(u_{0i} Y_0+euT_x)}{\left(\frac{\alpha v_{0i} Y_0}{ev}+Y_0+T_y-\alpha h\right)^2} & -\frac{(u_{0i} Y_0+euT_x)\left(\frac{v_{0i} Y_0}{ev}-h\right)}{\left(\frac{\alpha v_{0i} Y_0}{ev}+Y_0+T_y-\alpha h\right)^2} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{eu}{\frac{\alpha v_{0M^c} Y_0}{ev}+Y_0+T_y-\alpha h} & -\frac{u_{0M^c} Y_0+euT_x}{\left(\frac{\alpha v_{0M^c} Y_0}{ev}+Y_0+T_y-\alpha h\right)^2} & \frac{\alpha(u_{0M^c} Y_0+euT_x)}{\left(\frac{\alpha v_{0M^c} Y_0}{ev}+Y_0+T_y-\alpha h\right)^2} & -\frac{(u_{0M^c} Y_0+euT_x)\left(\frac{v_{0M^c} Y_0}{ev}-h\right)}{\left(\frac{\alpha v_{0M^c} Y_0}{ev}+Y_0+T_y-\alpha h\right)^2} \\ \frac{eu}{\frac{\alpha v_{0M^c} Y_0}{ev}+Y_0+T_y-\alpha h} & -\frac{u_{0M^c} Y_0+euT_x}{\left(\frac{\alpha v_{0M^c} Y_0}{ev}+Y_0+T_y-\alpha h\right)^2} & \frac{\alpha(u_{0M^c} Y_0+eutx)}{\left(\frac{\alpha v_{0M^c} Y_0}{ev}+Y_0+T_y-\alpha h\right)^2} & -\frac{(u_{0M^c} Y_0+euT_x)\left(\frac{v_{0M^c} Y_0}{ev}-h\right)}{\left(\frac{\alpha v_{0M^c} Y_0}{ev}+Y_0+T_y-\alpha h\right)^2} \end{pmatrix}$$

Évolution dynamique On s'intéresse ici au cas de piétons vus par une caméra embarquée dans un véhicule. On considèrera ici le piéton évoluant dans le temps alors que le véhicule est immobile, bien que l'on puisse sans difficulté étendre l'approche à un véhicule mobile. Le piéton, sera supposé évoluer sur un sol plan.

L'objectif recherché ici sera de réaliser la reconnaissance / localisation et suivi du piéton grâce à l'approche proposée.

On considérera la position (T_x, T_y) du piéton par rapport à sa position initiale dans le repère d'apprentissage.

Dans ce repère, nous pourrons écrire les relations suivantes :

$$\begin{cases} T_x(t+1) &= T_x(t)+\varepsilon_x \\ T_y(t+1) &= T_y(t)+\varepsilon_y \\ \alpha(t+1) &= \alpha(t)+\varepsilon_\alpha \\ h(t+1) &= h(t)+\varepsilon_h \end{cases} \tag{3.9}$$

On se propose ici d'une part de prédire quelles seront les nouvelles positions des cellules $\mathbf{a}(t+1|t)$, pour l'instant $t+1$ sachant leur position à l'instant k, et d'autre part, de déterminer quels seront les paramètres $\mathbf{v} = (T_x, T_y, \alpha, h)^t$ lorsque l'objet aura été détecté.

– **Prédiction de la position des cellules**
On cherchera donc ici à prédire la position $\mathbf{a}(t+1|t)$ des cellules dans l'image vue par la caméra sachant que le piéton s'est déplacé selon le modèle donné en (3.9).

Si à l'instant discret t on dispose de la connaissance $\mathbf{v}(t)$ et de sa covariance $\Sigma_{V(t)}$ alors, selon la relation (3.8) nous pourrons connaître la position $\mathbf{a}$ des cellules ainsi que leur taille déduite de la matrice de covariance $\Sigma_{\mathbf{a}}$ telles que :

$$\mathbf{a}(t) = f(\mathbf{a}_0, \mathbf{v}(t)) \qquad \text{et} \quad \Sigma_{\mathbf{a}}(t) = \mathbf{J_t}\Sigma_{\mathbf{v}}(t)\mathbf{J_t^t}$$

Pour l'instant $t+1$, le vecteur $\mathbf{v}(t)$ peut être prédit en utilisant l'équation d'évolution (3.9). Sa matrice de covariance $\Sigma_{\mathbf{v}}(t)$ va, elle aussi évoluer en $\Sigma_{\mathbf{v}}(t+1|t)$:

$$\Sigma_{\mathbf{v}}(t+1|t) = \Sigma_{\mathbf{v}}(t) + \mathbf{Q}_{\varepsilon}$$

Ici, $\Sigma_{\mathbf{v}}(t)$ est la matrice de covariance de $\mathbf{v}(t)$ et $\mathbf{Q}_{\varepsilon}$ est la matrice de covariance du vecteur d'erreur d'évolution $\varepsilon = [\varepsilon_x, \varepsilon_y, \varepsilon_\alpha, \varepsilon_h]^t$ qui sera choisie diagonale et avec des valeurs réalistes sur l'évolution potentielle du piéton entre les instants t et $t+1$, et des valeurs faibles pour la variance de ε_α et ε_h.

On aura donc, pour l'instant $t+1$ les valeurs suivantes :

$$\begin{cases} \mathbf{a}(t+1|t) &= f(\mathbf{a}_0, \mathbf{v}(t+1|t)) \\ \Sigma_{\mathbf{a}}(t+1|t) &= \mathbf{J_v}\Sigma_{\mathbf{v}}(t+1|t)\mathbf{J_v^t} \end{cases}$$

– **Localisation du piéton**
Une fois que l'on a prédit la position des cellules, ainsi que leur matrice de covariance, le processus de reconnaissance peut être lancé.

Dans le cas où l'objet (le piéton) est reconnu, on dispose de la position précise de chaque cellule mise à jour par la reconnaissance.

On dispose donc d'une observation $\mathbf{a}(t+1)$ et de sa matrice de covariance $\Sigma_{\mathbf{a}}(t+1)$.

L'objectif recherché ici est de déterminer quelle sera la position piéton qui correspond à cette vue dans l'image. On cherche donc ici à mettre à jour le vecteur $\mathbf{v}(t+1|t+1)$ pour l'instant $t+1$ connaissant la mesure $\mathbf{a}(t+1)$ et sa matrice de covariance $\Sigma_{\mathbf{a}}(t+1)$.

Cette opération peut être réalisée par les équations de Kalman suivantes :

$$\left\{\begin{array}{lcl} \mathbf{K} & = & \Sigma_{\mathbf{v}}(t+1|t)\mathbf{J}_{\mathbf{v}}^{t}\left[\mathbf{J}_{\mathbf{v}}\Sigma_{\mathbf{v}}(t+1|t)\mathbf{J}_{\mathbf{v}}^{t}+\Sigma_{\mathbf{a}}(t+1)\right]^{-1} \\ \mathbf{v}(t+1|t+1) & = & \mathbf{v}(t+1|t)+\mathbf{K}\left[\mathbf{a}(t+1)-f(\mathbf{a}_0,\mathbf{v}(t+1|t))\right] \\ \Sigma_{\mathbf{v}}(t+1|t+1) & = & (\mathbf{I}-\mathbf{K}\mathbf{J}_{\mathbf{v}})\Sigma_{\mathbf{v}}(t+1|t) \end{array}\right. \tag{3.10}$$

Cette opération permet donc de mettre à jour, en fonction d'une détection d'un piéton dans l'image, les paramètres T_x, T_y de la position du piéton. Les paramètres α et h quant à eux seront aussi mis à jour, néanmoins, si les variances sur ε_α et ε_h sont choisies faibles, cela signifiera qu'il s'agira, lors des première itérations d'un ajustement des paramètres α et h qui vont se stabiliser de manière naturelle au fur et à mesure des détections de piétons (cela peut être vu comme une calibration extrinséque de la caméra).

– **Suivi de piéton**
Lorsqu'un piéton a été reconnu puis localisé, il est très simple de lui appliquer le modèle d'évolution de la relation (3.9). Ainsi, on peut prédire sa nouvelle position dans la scène ainsi que la position probable dans l'image des cellules qui le constituent. Le processus peut ainsi être itéré afin de *suivre* le piéton de manière similaire aux visages.

3.4.4 Exemple de reconnaissance, localisation et suivi de piétons

Dans le but d'analyser le comportement de notre algorithme lors du suivi et de la localisation simultanées, nous avons procédé à la réalisation de quatre tests différents. D'abord, le premier test est orienté vers l'analyse du suivi, dans le plan horizontal, en prenant le modèle avec uniquement de l'information fréquentielle. Ensuite, la même séquence est testée mais cette fois-ci en utilisant le modèle qui intègre de l'information fréquentielle et le niveau de gris. Ici nous cherchons à analyser l'apport de l'intégration des caractéristiques multiples. Un troisième test est destiné à l'analyse du suivi pour un objet se déplaçant dans le plan horizontal et en profondeur. Enfin, nous présentons un test pour évaluer l'adaptation de l'algorithme pour des objets variables en échelle.

3.4.4.1 Test 1 : modèle avec de l'information de contour uniquement

Sur la figure 3.29 nous montrons quelques extraits d'une séquence vidéo enregistrée dans notre laboratoire. Cette séquence consiste en deux piétons habillés avec des vêtements de couleur différente (noir et blanc). Le piéton habillé en blanc reste fixe et placé au centre de l'image, le piéton habillé en noir se déplace de gauche à droite dans le plan horizontal. A un moment donné, les deux piétons se superposent dans l'image.

DAns l'objectif de reconnaître le piéton situé à gauche, la région d'intérêt a été initialisée avec un positionnement du piéton sur la gauche (x=-2m) (tel que montré sur la première image de la séquence). Après le lancement du processus de reconnaissance, le piéton à gauche est reconnu après 16 itérations et après trois évaluations du classificateur SVM. Dès lors, pour l'image suivante $t+1$, la phase de suivi commence : c'est toujours l'algorithme de reconnaissance qui est lancé mais cette fois ci avec un intervalle nettement réduit par rapport à l'intervalle initial (voir Fig. 3.29). Pour le reste des images (où l'objet est reconnu), la reconnaissance est faite avec une moyenne de 4 itérations et 3 évaluations SVM. Au moment de la superposition des deux piétons, on observe que l'algorithme « perd de vue » l'objet initial. L'algorithme reste donc bloqué sur le piéton immobile. Ceci est dû principalement au fait que la reconnaissance n'est faite qu'à partir de l'information fréquentielle.

3.4.4.2 Test 2 : modèle avec l'information de contour et niveau de gris

Contrairement à l'exemple précédent, sur la figure 3.30 nous présentons la même séquence, sauf qu'ici nous avons utilisé le modèle qui intègre deux caractéristiques différentes : d'une part, des caractéristiques basées sur le contenu fréquentiel, d'autre part des caractéristiques basées sur le niveau de gris. Dans cet exemple nous pouvons observer que l'algorithme « ne perd pas de vue » l'objet initial (le piéton habillé en noir). Ceci est dû au fait qu'après la reconnaissance du piéton, dans la première image, le modèle prend les valeurs des caractéristiques de celui-ci. Ainsi, au moment de la superposition des deux piétons, l'algorithme recherche des éléments *uniquement* correspondant à l'intervalle (tant en position que dans l'espace des paramètres) autour du modèle « instancié ». On voit donc, tout l'intérêt d'utiliser une représentation plus riche du modèle. Pour cet exemple, nous avons une moyenne de 32 itérations par reconnaissance avec 9 évaluations SVM au maximum. Le nombre d'itérations est doublé car ce modèle porte de l'information « parasite » qui trompe l'algorithme. Cette information correspond au niveau de gris du fond de la base d'entraînement. Pour synthétiser, nous pouvons dire que le niveau de gris apporte une valeur ajoutée par rapport au modèle précédent du fait que le piéton n'est pas perdu. Cependant, l'intégration du niveau de gris génère des fausses hypothèses qui font ralentir l'algorithme. La cause principale est à chercher du côté de l'élimination des caractéristiques non pertinentes lors de la construction du modèle.

3.4.4.3 Test 3 : déplacements horizontaux et en profondeur

Un troisième exemple est présenté sur la figure 3.31. Dans cette séquence le piéton se déplace d'un côté à l'autre en s'éloignant de la caméra. Le but ici est d'observer l'adaptation du modèle en fonction de la taille de l'objet.

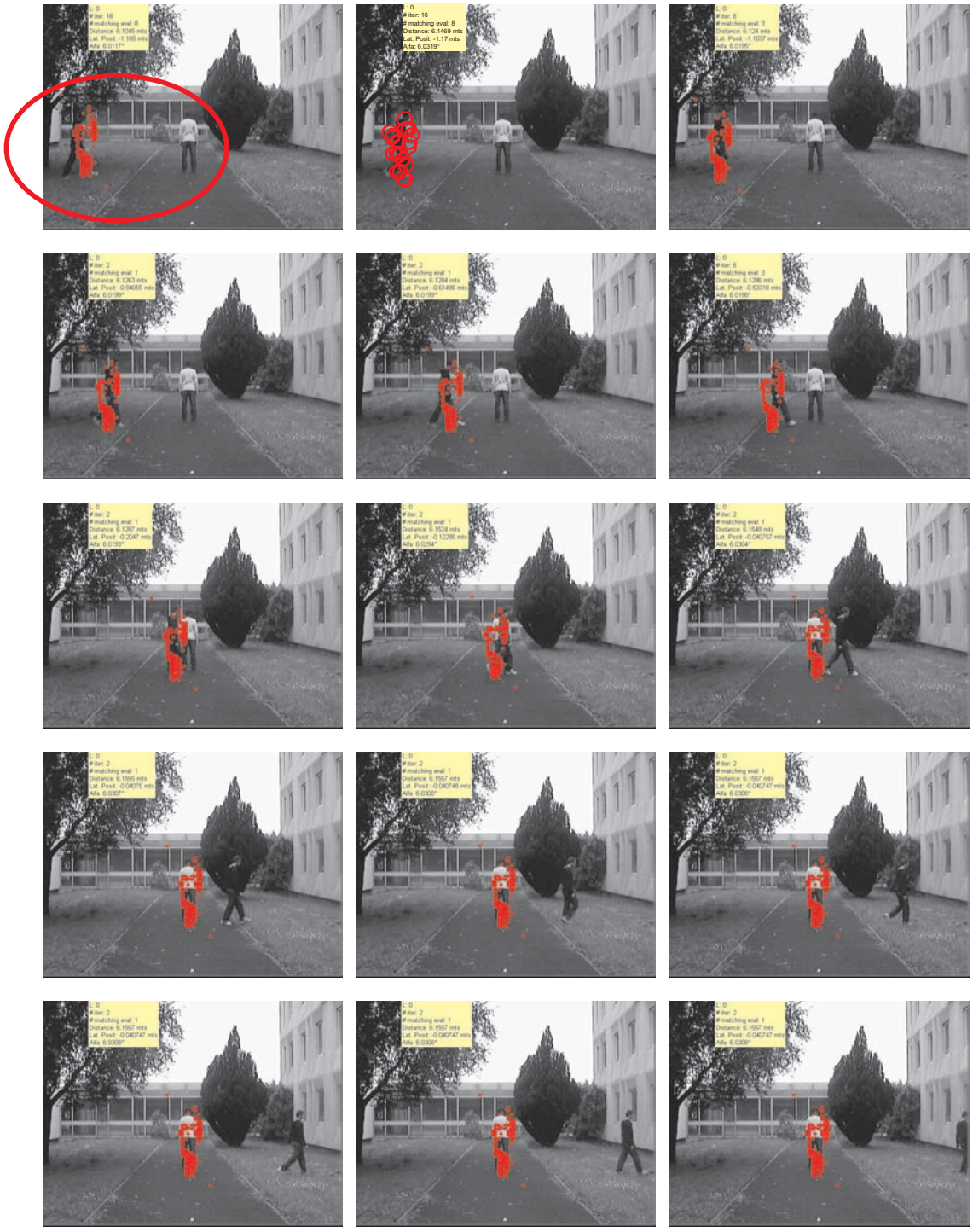


FIG. 3.29 – Suivi d'un piéton se déplaçant de gauche à droite dans le plan horizontal. Dans cette séquence, les deux piétons se superposent à un moment donné. Ici, l'algorithme « perd de vue » le piéton initialement reconnu du fait du manque de description autre que fréquentielle. L'ellipse dans la première image de la séquence montre la région d'intérêt initiale pour toutes les *parties*. Dans la deuxième image de la séquence, les ellipses correspondent aux régions d'intérêt de chaque *partie* réduites après la mise à jour du modèle.

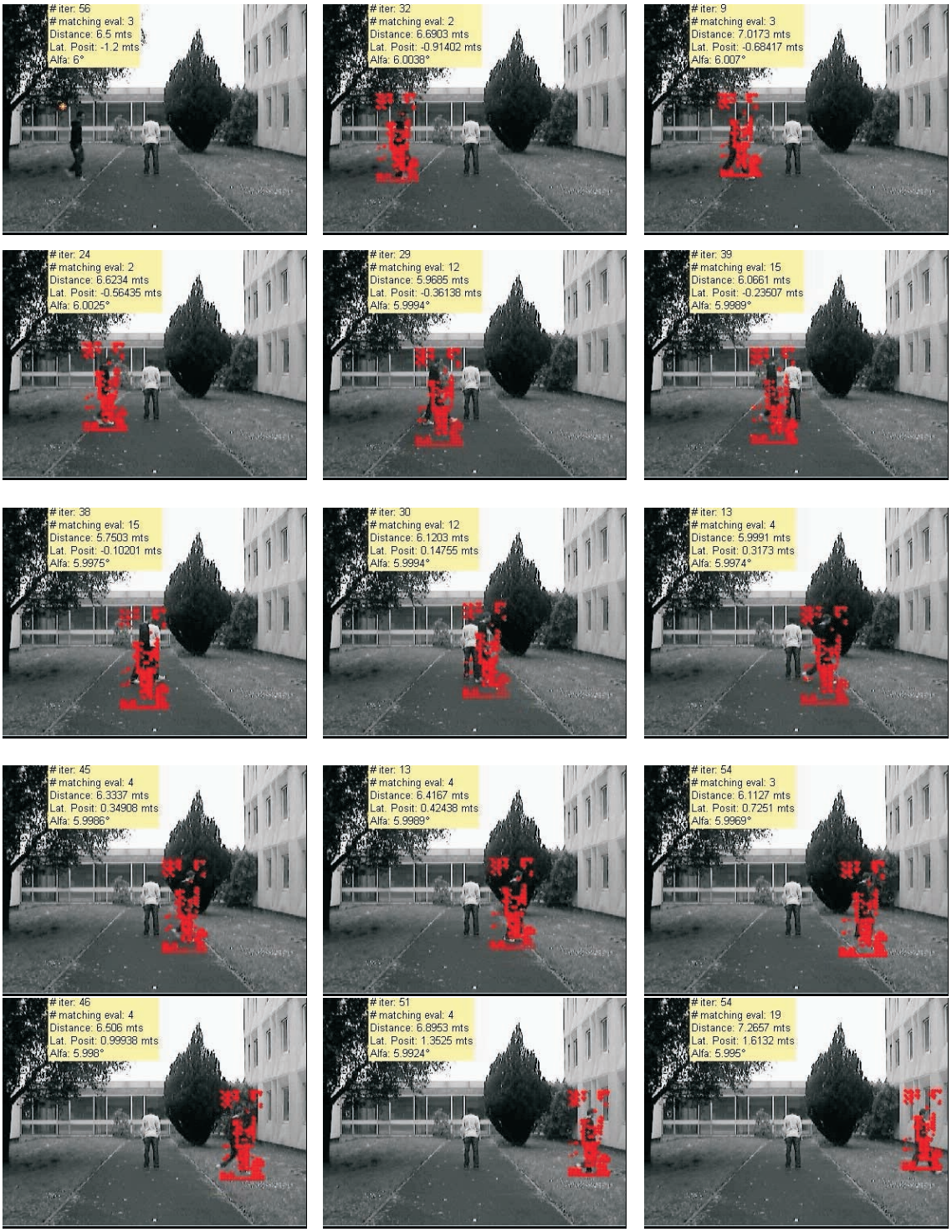


FIG. 3.30 – Suivi d'un piéton se déplaçant de gauche à droite. Ici, le modèle utilisé pour l'algorithme de reconnaissance est composé de deux types des caractéristiques : celles basées sur l'information fréquentielle, et celles basées sur le niveau de gris. A la différence de l'exemple présenté en §3.4.4.1, ici l'algorithme ne « perd pas de vue » le piéton reconnu initialement. Ceci démontre l'intérêt d'intégrer des caractéristiques multiples pour la représentation des objets.

Dans la première image, l'algorithme a détecté le piéton après 12 itérations et 2 évaluations du classificateur SVM. Pour les images suivantes, la moyenne est de 4 itérations par détection et 2 évaluations SVM. Grâce à l'intégration de l'information 3D sur la scène (position et angle de la caméra) et à la réduction de l'espace de recherche, l'algorithme est capable de suivre l'objet et de s'adapter selon les variations en échelle. En réduisant l'espace de recherche, on restreint la recherche dans des zones qui sont plus pertinentes ce qui diminue potentiellement les fausses détections (des ambiguïtés qui peuvent exister avec le modèle).

Le fait d'intégrer l'information 3D oblige à trouver des « objets » qui correspondent bien au modèle. L'objet doit, à son tour, être cohérent avec la configuration de la scène : à une certaine hauteur de l'image, il est impossible d'avoir un piéton de grande taille ; ou inversement, très bas dans l'image il n'est pas possible de trouver un piéton de petite taille. Du fait de la nature de cette expérimentation, il n'est pas évident de donner une évaluation concernant la « qualité » de l'adaptation du modèle. Le jugement est nettement subjectif car on ne peut pas compter avec l'information nécessaire pour mesurer la différence entre l'objet et le modèle. En revanche, nous pouvons dire que l'algorithme arrive à suivre les variations en taille de l'objet, d'une façon « quasi » continue.

Une chose importante qu'il faut souligner est qu'avec un nombre réduit de *parties* détectées (de l'ordre de 10 d'après nos expérimentations), le modèle est suffisamment réduit pour réaliser la tâche de classification. Typiquement, l'algorithme détecte d'abord les *parties* appartenant au contour du piéton. Cela permet d'adapter le modèle, de la manière la plus rapide à la taille de l'objet. Une fois le modèle placé, nous extrayons l'imagette et on la normalise à la taille de l'imagette de référence pour la classification SVM (64×128 pix). Ce résultat nous permet d'éviter de faire une recherche exhaustive en échelle, de l'objet qui nous intéresse.

3.4.4.4 Test 4 : déplacement en profondeur

De même, une autre séquence, enregistrée dans un environnement différent, est présentée sur la figure 3.32. Pour cette séquence nous avons initialisé l'angle et la hauteur de la caméra conformément à la réalité ($\alpha = 5^\circ$, $\sigma_\alpha = 0,2^\circ$, $h = 2,0m$, $\sigma_h = 20cm$). Ici, on s'intéresse à la détection des piétons qui sortent tout en bas de l'image. Pour ce faire, on initialise la position probable du piéton entre 6m et 9m.

Dans les deux premières images, l'algorithme est incapable de détecter le piéton. Même si celui-ci arrive à détecter certaines *parties*, la classification SVM ne le prend pas comme objet détecté. On observe sur l'image malgré tout la dernière détection réalisée. On peut expliquer cela du fait que l'objet n'est pas entièrement dans l'image : on ne fait pas l'évaluation SVM si les limites de la fenêtre d'intérêt sont hors de l'image.

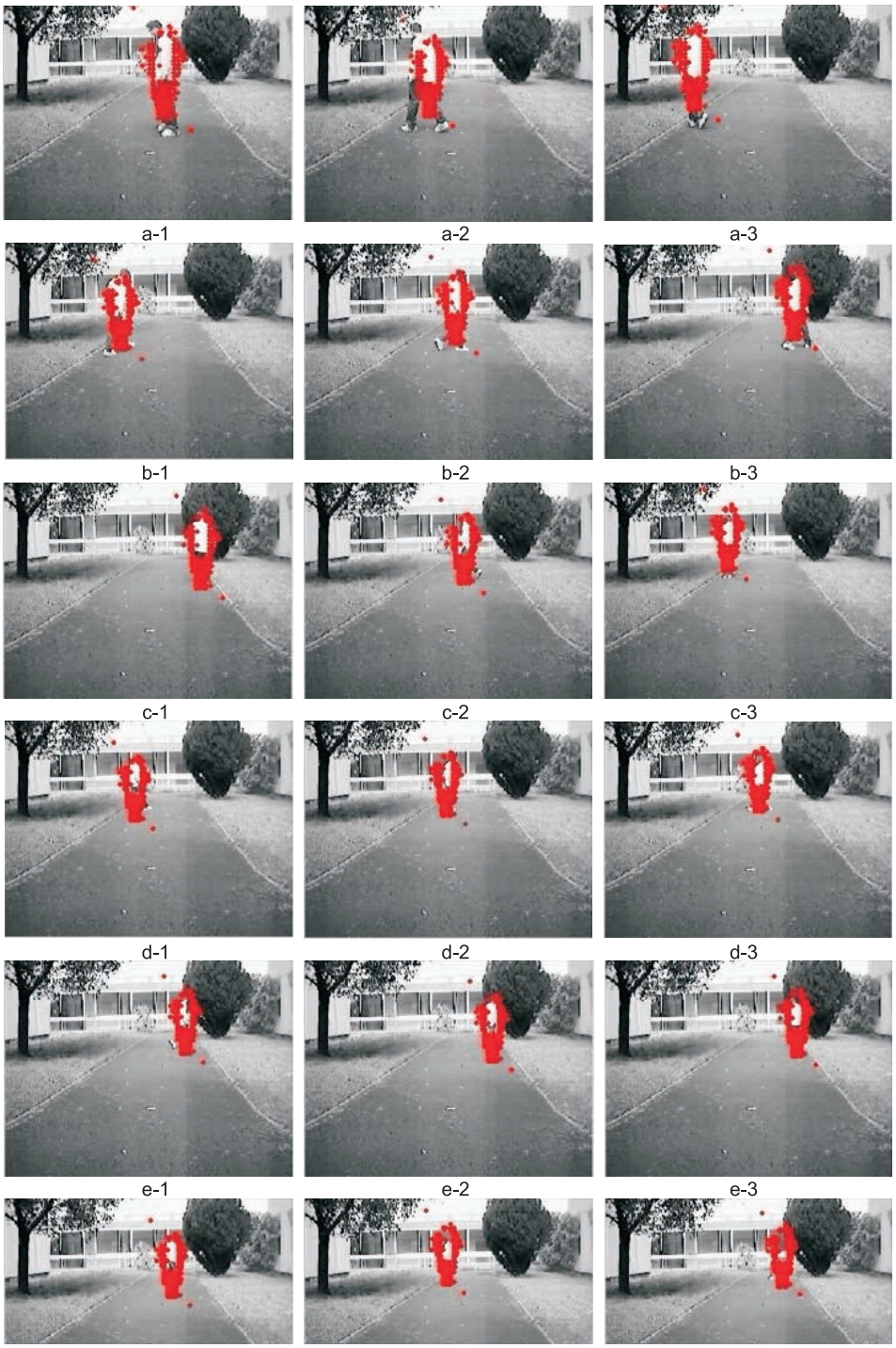


FIG. 3.31 – Exemple d'un piéton se déplaçant de gauche à droite et en s'éloignant de la caméra. Nous pouvons observer la bonne adaptation du modèle à l'objet dans la phase de suivi lorsque l'objet varie en échelle.

C'est après une quinzaine d'images que l'algorithme arrive à détecter le piéton. On peut observer que, au départ, l'ajustement du modèle à l'objet, ne correspond pas exactement à la taille de ce dernier. Ces erreurs sont liées à la façon de représenter l'objet : nous avons appris le modèle avec de l'information à moyenne et basse résolution ; ce qui entraîne des erreurs de détection sur la position des *parties*. Ici, l'algorithme arrive à suivre le piéton jusqu'au moment où, dans la zone d'intérêt, apparaissent d'autres piétons. Dès lors, le mécanisme d'attention fait que l'algorithme commence à « sauter » d'un piéton à l'autre jusqu'à la perte définitive du piéton initial. En fait, l'algorithme détecte celui qui correspond au plus proche en position par rapport au modèle moyen. En outre, nous trouvons aussi des fausses détections. L'algorithme confond un piéton avec la colonne au fond du couloir. Enfin, l'algorithme perd la détection et reste bloqué dans la zone d'intérêt finale, en attendant la réapparition de l'objet.

3.4.5 Bilan

En faisant différents tests pour la reconnaissance des piétons, nous avons montré la pertinence d'intégrer le contexte lors de la phase de reconnaissance. L'intégration notamment de l'information concernant la structure de la scène, position et angle de la caméra ainsi que les connaissances sur le possible comportement de l'objet, aident largement à réduire des ambiguïtés lors de la reconnaissance. Le nombre de fausses détections est réduit significativement par rapport au balayage exhaustif. En complément aux quatre niveaux de sélection de l'attention présentés auparavant, la sélection dans l'espace 3D est complètement justifiée quand il s'agit d'analyser une scène dynamique.

En outre, les résultats obtenus dans la phase de suivi, notamment ceux issus de l'intégration des caractéristiques multiples, encouragent à avoir une représentation plus riche de l'objet pour améliorer sa détection.

FIG. 3.32 – Exemple du piéton s'éloignant de la caméra. Ici, les paramètres de la caméra (angle et hauteur) sont initialisés pour que le système soit « attentif » aux piétons qui puissent apparaître tout au début du couloir. Le fait d'intégrer de l'information 3D sur la scène, aide à réduire significativement les fausses détections. Les objets qui sont détectés sont ceux qui ont une cohérence en accord avec la structure de la scène : des piétons apparaissant de grande taille en bas, et de petite taille vers l'horizon. La prise en compte de cette information entraîne une amélioration de la détection du piéton lors du suivi de celui-ci.

3.5 Discussion

Afin de valider notre approche, nous avons montré trois tâches de la vision par ordinateur sur deux classes d'objets différentes : la reconnaissance, la localisation et le suivi de visages et de piétons. Nous avons insisté sur le fait qu'avec la méthodologie que l'on propose, en prenant en compte certains éléments additionnels, nous sommes capables de faire non seulement la reconnaissance mais aussi la localisation et le suivi simultanément.

Quand il s'agit de reconnaître un objet, qui appartient à une scène 3D, l'intégration au modèle d'informations concernant la structure de la scène permet de faciliter la tâche de suivi et de reconnaissance. Ainsi, outre le fait de réaliser la focalisation dans le plan image, l'espace de paramètres et l'échelle d'analyse, nous avons montré qu'il est aussi possible de faire de la focalisation dans l'espace 3D. Cela est d'une grande importance car l'observateur (dans notre cas la caméra) peut prendre un rôle actif dans le processus de reconnaissance.

Les résultats nous montrent que la méthodologie que l'on propose est adaptée pour faire de la reconnaissance visuelle d'objets. Même si les *parties* (isolées) qui composent l'objet sont des descripteurs très simples et peu discriminants sur une classe d'objets donnée (ce qui peut entraîner une grande combinatoire au moment d'apparier le modèle), cette faiblesse est compensée par le fait que l'on intègre plusieurs facteurs :

- la représentation de l'objet à partir de caractéristiques multiples,
- la dépendance statistique existante entre les différentes *parties* composant l'objet,
- la focalisation sur les quatre niveaux (plan image, espace de paramètres, échelle d'analyse et 3D) et
- l'adaptation du modèle en fonction de la tâche courante.

Avec l'exemple d'application à la reconnaissance de visages, nous avons montré qu'il nous faut un nombre réduit de *parties* pour guider le processus de reconnaissance de l'objet. Ceci suggère l'utilisation des *parties* de l'objet plutôt comme des indices de focalisation et non comme éléments pour la décision *objet/non objet*.

Chapitre 4

Conclusion et perspectives

4.1 Conclusion générale

Notre travail de recherche a été orienté vers l'analyse globale de scènes, ce qui nous a conduit à l'exploration d'une voie alternative pour la tâche de reconnaissance visuelle d'objets. Cela nous a permis d'être témoins de l'évolution, tant pour les applications que pour les techniques, de cette tâche de la vision artificielle.

Premièrement, nous avons présenté l'état de l'art des techniques de reconnaissance d'objets, apportant une attention particulière aux techniques basées sur l'apparence et à celles basées sur l'appariement relationnel. Ensuite, étant donnés les besoins des applications actuelles, nous avons cru pertinent de décrire l'attention visuelle en privilégiant son rôle dans le processus de perception, principalement concernant la réduction du flot d'information. Une discussion a été établie concernant la façon de voir et d'étudier le problème de la vison artificielle. Nous avons vu le besoin de penser globalement le système de vision par ordinateur sous un point de vue plutôt *unifié* des différentes problématiques de ce domaine. Ainsi donc, en tenant compte des éléments nécessaires, nous avons lancé notre proposition sur le sujet avec une démarche qui peut garder un caractère générique et intégral.

La voie explorée nous a amené à développer une méthodologie où la tâche de reconnaissance correspond à un processus *actif* de perception ; le processus de reconnaissance et le mécanisme d'attention étant gouvernés par le même algorithme de contrôle. L'ensemble de cette méthodologie va de la définition et la construction du modèle de l'objet, jusqu'à la définition de la stratégie pour la future reconnaissance de celui-ci. Ainsi, du point de vue de la représentation, cette approche est capable de modéliser aussi bien la structure de l'objet que son apparence, à partir de caractéristiques multiples, qui à leur tour servent comme indices d'attention. Dans ce cadre, des concepts importants comme celui de l'attention visuelle, nous ont permis d'intégrer, d'une façon très satisfaisante,

plusieurs niveaux du spectre attentionnel. Ces niveaux du spectre attentionnel sont : la sélection des opérateurs bas niveau, la sélection de l'intervalle de réponse des opérateurs, la sélection du niveau de détail et la sélection de régions d'intérêt dans l'image perçue, les sélections étant guidées par la connaissance de l'objet et la structure de la scène. Le fait d'intégrer tous ces niveaux, entraîne une optimisation globale du processus de reconnaissance. D'ailleurs, grâce à la « focalisation progressive » et à la représentation hybride du modèle, nous pouvons profiter des avantages des deux types de représentations classiques. D'une part, la classification finale objet/non objet est faite par une décision sur la vue globale de l'objet. D'autre part, la structure de ce dernier permet de guider le processus de reconnaissance à partir des observations locales, permettant de réduire de manière drastique la zone d'application du classificateur. La pertinence de l'utilisation des éléments issus d'une représentation ou de l'autre est une tâche confiée à l'algorithme de décision. Par ailleurs, d'autres tâches de la vision par ordinateur, comme celles de la localisation et du suivi, ont été décrites en s'inscrivant dans ce cadre formel d'une façon naturelle.

Des résultats très importants, notamment concernant la diminution des fausses détections et l'évitement de la recherche exhaustive en position et en échelle, nous permettent de justifier pleinement une approche par vision focalisée. En plus, d'autres résultats méritent d'être cités. D'une part, le fait de prendre en compte des informations a priori sur la structure de la scène ou le modèle de l'objet, permet de réduire d'une manière significative la combinatoire implicite dans une tâche de recherche visuelle. D'ailleurs la reconnaissance et le suivi sont notamment améliorés en intégrant de l'information 3D. D'autre part, les résultats nous montrent qu'il est possible de faire la reconnaissance, la localisation et le suivi d'une façon simultanée.

A notre avis, l'idée mérite d'être retenue surtout du fait du caractère général qu'elle présente, ainsi que la vaste gamme de perspectives et d'améliorations envisageables.

Les problèmes rencontrés qui ont empêché une validation rigoureuse pour l'application présentée, sont engendrés principalement par la mise en oeuvre et les techniques utilisées. La stratégie globale reste cependant toujours intacte. A titre d'exemple nous avons l'apprentissage du modèle : nous avons proposé une méthode qui a répondu à nos besoins, mais il ne faut pas oublier que celle-ci représente une problématique assez importante par elle même. Elle mérite d'être profondément étudiée mais sans jamais perdre de vue l'objectif final, ainsi que la place occupée dans le processus de perception.

Le fait de ne pas avoir un modèle correct de l'objet entraîne des erreurs sur le résultat final. D'ailleurs, pour l'apprentissage, tout laisse penser que celui-ci devrait être aussi intégré dans la boucle de perception et non traité à part comme dans la mise en œuvre actuelle. Ceci pourrait nous amener directement vers *l'apprentissage non-supervisé*.

Dans la section suivante nous présentons certains sujets qui méritent d'être analysés en détail. Nous restons convaincus que des améliorations sur ces sujets pourraient rendre un résultat global beaucoup plus performant.

4.2 Travaux futurs

4.2.1 Représentation de l'objet

L'idée originale qui nous a amenés à la représentation proposée, est attachée au fait d'avoir plusieurs composants de l'objet pouvant servir comme indices d'attention visuelle, cette dernière étant un des principes fondamentaux de notre approche dans un processus actif de reconnaissance d'objets.

L'extraction des *parties* au moyen d'une grille de cellules, distribuées d'une façon uniforme sur l'objet et à différentes résolutions, nous a permis, d'une part d'avoir une représentation pour la mise en place, d'une façon satisfaisante, d'un processus de reconnaissance guidé par le modèle, d'autre part d'éviter que ce soit l'utilisateur qui définisse les composants de l'objet.

Analysons d'abord les résultats importants qui sont à conserver pour des études ultérieures compte-tenu des éléments suivants :

- pour une même classe d'objets on a un nombre fixe de *parties* qui décrivent l'objet,
- les indices de focalisation sont basés sur des caractéristiques multiples (potentiellement différentes) : ce qui permet d'avoir des *parties* assez discriminantes,
- les *parties* caractérisent l'objet à plusieurs résolutions : ce qui permet un parcours guidé du processus à travers l'espace-échelle.

En revanche le mode opératoire n'est pas optimal et ceci dû à plusieurs facteurs :

1. **le recouvrement spatial de la grille de cellules.** Avec cette représentation, les cellules sont placées et distribuées d'une façon uniforme, l'une à coté de l'autre. Il n'est donc pas possible de capter des informations plus globales dans l'image.

2. **Redondance de l'information.** Du fait que l'on ne prend pas en compte la possibilité qu'une *partie*, de niveau de résolution plus élevé, peut contenir la même information qu'une autre correspondant à un niveau inférieur (plus grossier), nous nous retrouvons avec le problème de la redondance de l'information dans le modèle de l'objet. Cela représente un double inconvénient dans la mesure où d'une part le modèle de l'objet prend une taille assez importante, d'autre part cette information redondante peut fonctionner plutôt comme une élément perturbant lors de la tâche

de recherche : plus complexe est le modèle, plus il faut de temps pour le mettre à jour ; plus il y a de *parties* avec de l'information redondante, plus le processus peut passer de temps pour chercher celles qui n'aident pas à réduire rapidement l'espace de recherche.

3. **« Aspect » de chaque cellule.** Bien que la représentation choisie ait répondu à nos besoins, la forme carrée de la seule cellule n'est pas toujours la mieux adaptée à la structure locale ou globale de l'image.

Ainsi donc, nous préconisons les solutions suivantes :

1. **Augmentation de la résolution spatiale.** Le placement des cellules superposées, dans toutes les échelles d'analyse, permettrait d'avoir un meilleur recouvrement de l'espace 2D du fait que l'on augmente la résolution spatiale, par exemple des cellules séparées en pas d'un quart.

2. **Élimination de la redondance.** Afin d'éliminer cette redondance, on pourrait envisager d'utiliser une technique d'analyse similaire à celle de la décomposition de l'image en « blocs » ou « quad-tree decomposition ». Ainsi, en reprenant notre approche, une seule *partie* décrivant la structure globale de l'objet, peut être suffisante en lieu de plusieurs *parties* (plus petites) décrivant la structure locale aux niveaux supérieurs.

3. **Cellules d'« aspects » différents.** Lors de l'apprentissage, on devrait disposer d'un jeux de cellules d'« aspects » différents de façon qu'elles puissent être adaptées selon la structure locale ou globale de l'image. Cette cellule peut être elle-même une seule *partie* et non pas trois ou quatre comme actuellement. On peut proposer une technique par filtrage où une cellule pourrait bien correspondre à un *masque du filtre de Gabor*. Ce filtre ne sera pas seulement décrit par la position et l'orientation mais aussi par son échelle.

   Sur la figure 4.1 on montre comment la structure globale d'un piéton peut être décrite par une seule *partie*. Dans ce cas, la *partie* correspond à un masque du filtre de Gabor (§3.2.1.1) orienté à 90° (vertical) et avec $\sigma_y = 2\sigma_x$ (voir Fig. 4.1-c). Les étoiles dans l'image sont les maximaux locaux de la réponse de l'ondelette de Gabor. Nous pouvons remarquer la constance de réponse du filtre le piéton étant à deux échelles différentes. En filtrant l'image par une ondelette avec les caractéristiques citées ci-dessus, on préserve uniquement de l'information en accord avec la structure choisie. Pour cet exemple, seulement 14 régions dans l'image répondent à ce type de structure. A ce stade, nous avons déjà une diminution énorme du nombre

de candidats potentiels par rapport au nombre de pixels dans l'image.

Si l'on poursuit cette idée, on pourrait imaginer avoir une représentation de l'objet à partir d'un ensemble de masques du filtre à différentes orientations, échelles et aspects, comme présenté sur la figure 4.2-a. Pour cet exemple on essaie d'illustrer d'une façon très simple comment un jeu d'ondelettes pourraient servir comme *parties* de l'objet. Les ondelettes montrées ont été choisies « à la main ». Dans la figure 4.2-b, nous présentons une image quelconque filtrée par l'ensemble de ces ondelettes de Gabor. Comme résultat on obtient uniquement deux réponses correspondant à la *partie* A_1 (carrés), 19 réponses correspondant à la *partie* A_2 et A_3 (cercles), et plusieurs réponses pour les *partie* A_4 et A_5 (étoiles et points respectivement). Si l'on intègre une représentation de ce type dans notre approche, on pourrait éventuellement, non seulement avoir une convergence beaucoup plus rapide de l'algorithme de reconnaissance, du fait du lien statistique sur la position spatiale de chacune, mais aussi une augmentation du pouvoir discriminant lors de la détection de chaque *partie*.

La construction automatique du modèle à partir de ces éléments, devient une problématique importante pour la phase d'apprentissage.

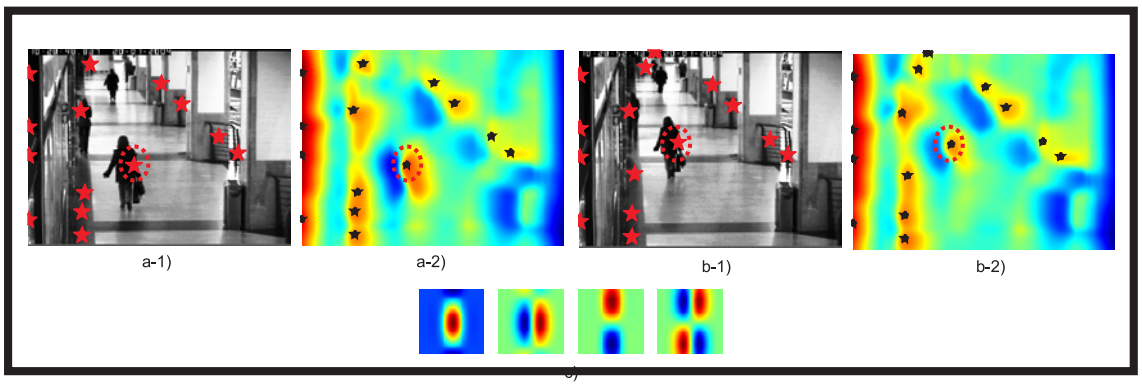


FIG. 4.1 – Exemple de l'utilisation du filtre quaternion de Gabor comme descripteur de l'objet. Dans (a-1), les étoiles représentent les maximaux locaux (voisinage de 8 pixels) de l'image filtrée (a-2). (b-1) et (b-2) correspondent à l'image extraite de la même séquence 3s plus tard. Les quatre masques du quaternion de Gabor sont affichés dans (c) (partie réelle, et trois imaginaires i, j, k [13, 9]). Dans cet exemple σ_x et σ_y sont adaptées de façon que la structure globale de l'objet soit captée. Il nous faut donc un seul descripteur pour quasiment décrire la structure globale.

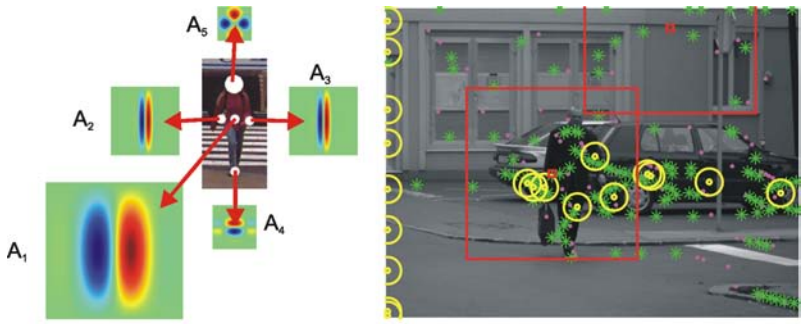


FIG. 4.2 – Exemple du modèle de piéton basé sur des ondelettes de Gabor. A1 : carré. A2, A3 : cercle. A4 : étoile. A5 : point.

4.2.2 La recherche séquentielle : est-elle toujours pertinente ?

Les résultats obtenus lors des expérimentations pour la reconnaissance de visages et de piétons nous confirment la pertinence de cette approche. En fait, la « focalisation progressive », lors de chaque hypothèse validée, nous permet d'avoir une diminution assez importante de l'espace de recherche. Ceci entraîne une diminution aussi importante de la combinatoire pour la mise en correspondance. Du fait de la mise en oeuvre de la stratégie choisie, il y a quelques problèmes qui se posent malgré ces résultats.

La mise en oeuvre séquentielle actuelle ne permet de gérer qu'une hypothèse à la fois (une seule branche de l'arbre est suivie). Même si le critère d'arrêt décrit dans §2.4.6.4 permet de couper très tôt les branches engendrées par des *mauvaises hypothèses*, l'algorithme peut passer beaucoup de temps pour explorer tous les candidats possibles engendrés pour une *hypothèse mère*. Deux solutions, d'ailleurs pas incompatibles, pourraient être envisageables pour diminuer l'impact de ce problème. La première est d'éliminer presque complètement les *mauvaises hypothèses mère*, en d'autres termes avoir des *parties* très discriminantes ; problème analysé dans la section précédente. La deuxième correspond à la gestion d'hypothèses multiples. Ceci a le grand inconvénient de devenir vite très coûteux en temps de calcul. Dans ce cas, une modification de la mise en œuvre de l'algorithmique est nécessaire.

L'idée pourrait être de construire une mise en oeuvre différente, laquelle pourrait être basée sur une parallélisation à trois niveaux algorithmiques :

1. Un parallélisme à bas niveau peut être mis en place pour chaque détecteur ce qui permet de réduire son temps de calcul (voir §4.2.3).

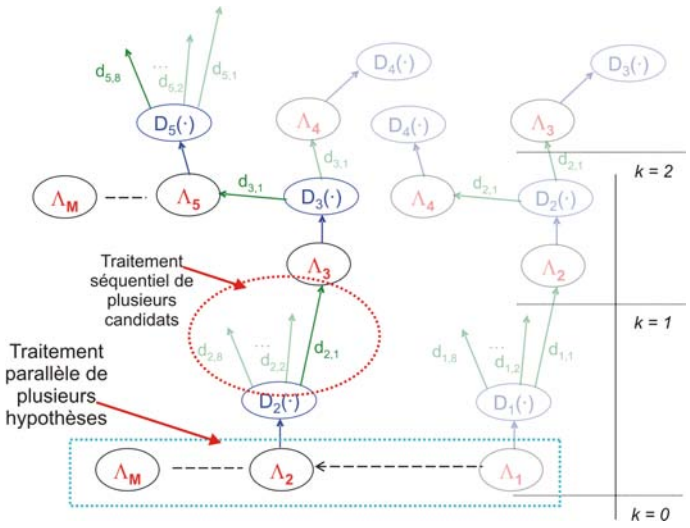


FIG. 4.3 – Ensemble d'hypothèses représentées par une structure d'arbre de recherche. Au niveau de base, c'est-à-dire au niveau zéro du processus, plusieurs hypothèses pourront être testées en parallèle. A son tour, chaque candidat engendré par la détection d'une *partie* hypothèse Λ_m, sera traité sous forme séquentielle. Un processus à part sera créé lors du test de chaque candidat. La « mort » d'un processus sera donnée par la règle de décision donnée par le rapport de vraisemblance (voir §2.4.6).

2. Un parallélisme à moyen niveau qui consiste à appliquer en même temps l'ensemble des détecteurs d'une caractéristique dans la zone d'intérêt courante.

3. Un parallélisme à haut niveau permettant la gestion multi hypothèses, c'est-à-dire le parcours simultané de plusieurs branches de l'arbre de recherche. Dans la conception actuelle de l'algorithme, cela signifie que plusieurs caractéristiques peuvent être observées en même temps (voir Fig. 4.3).

Ainsi, le processus de reconnaissance peut être vu comme une « compétition » entre plusieurs hypothèses « mère », où l'hypothèse gagnante sera celle qui aurait eu une détection réussie de l'objet. Les « mauvaises » hypothèses seront à leur tour éliminées d'une façon naturelle par le critère de *branch&bound* développé (voir Fig. 2.4.6).

En outre, cette voie d'exploration devrait permettre d'étudier l'influence de l'approche parallèle sur la structure de l'algorithme de reconnaissance. Par exemple, les hypothèses doivent pouvoir « communiquer » entre elles, c'est-à-dire que l'infirmation ou la confirmation de l'une peut avoir des conséquences sur les autres. De même, l'ensemble des détections déjà effectuées doit être accessible à toutes les hypothèses à venir pour éviter

qu'un même détecteur ne soit appliqué plusieurs fois sur les mêmes données.

On voit donc que ceci représente un problème de complexité assez élevée mais tout à fait envisageable comme voie d'exploration.

4.2.3 Intégration pour la vision active

Le besoin de réaliser des tâches visuelles au plus proche du capteur est une des bases du paradigme de la vision active. Elle peut être définie comme un *contrôle actif de tous les paramètres du capteur afin de réaliser une tâche donnée* [1]. Cette définition montre la place importante qu'occupe le capteur dans un système de vision active. Ainsi, par analogie aux systèmes de vision biologiques, ce dernier établit qu'un système de perception ne doit pas se résumer à envoyer la totalité des informations perçues au mécanisme de décision/interprétation. L'information envoyée doit être filtrée/traitée au préalable, afin de ne conserver que l'information la plus pertinente selon le contexte, réduisant ainsi fortement la quantité d'information à transmettre (voir §1.4 concernant les mécanismes d'attention visuelle).

K. Pahlavan et al., dans [80], divisent les composants d'un système de vision active en quatre catégories :

- **les paramètres optiques** qui définissent la projection d'une scène 3D en une image 2D (zoom, éclairement...),
- **les paramètres sensoriels** qui conditionnent le signal électrique porteur de l'information image 2D (sensibilité, échantillonnage...),
- **les paramètres mécaniques** qui déterminent le positionnement et les mouvements de la caméra,
- **les paramètres algorithmiques** qui caractérisent le traitement et les transformation de l'information.

Concernant notre approche, elle permet dans l'état actuel surtout une gestion des paramètres sensoriels et algorithmiques. Cependant, elle n'est pas limitée à ce type de paramètres. On pourrait introduire dans la boucle de perception les paramètres sur le positionnement et le mouvement de la caméra, ainsi que des paramètres optiques (e.g. contrôle du zoom).

Ainsi donc, afin de reproduire ce type de comportement perceptif dans un système

électronique, le capteur de vision (caméra) peut être doté d'un certain nombre de ressources matérielles lui permettant d'exécuter des traitements embarqués. Ces ressources peuvent être implantées sous la forme de processeurs embarqués, processeurs de signal de type DSP, logique configurable (FPGA), circuits dédiés (ASIC)... [28].

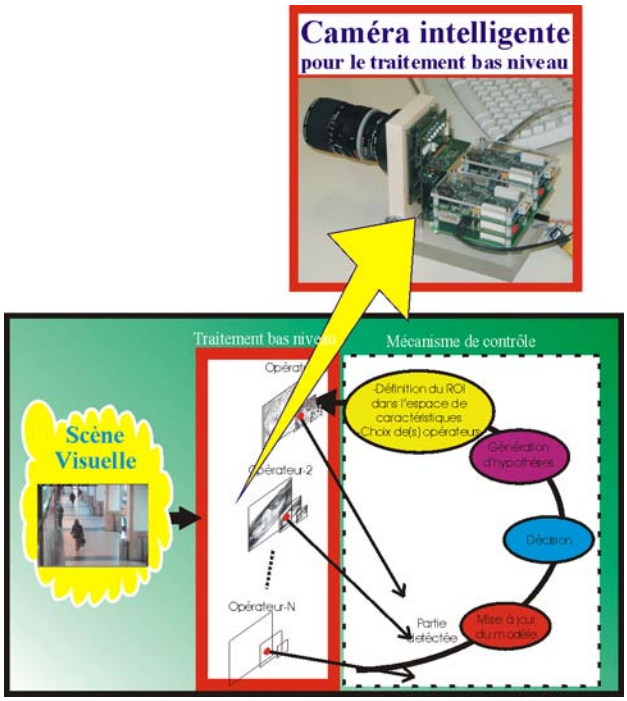


FIG. 4.4 – Acquisition de données avec une « caméra intelligente ». Ce module s'intégre d'une façon naturelle dans le cadre que l'on propose.

A l'heure actuelle tout le traitement d'image bas niveau est fait dans l'ordinateur. Il pourrait être tout à fait envisageable d'utiliser une plate-forme de *caméra intelligente*. Ainsi, l'information provenant directement de l'environnement pourrait être traitée juste à la sortie du capteur et avoir un gain en temps de calcul. Ce sera le processus de reconnaissance qui fera les requêtes d'information nécessaires pour une future interprétation : région(s) dans l'image, niveau(x) de détail, opérateur(s) bas niveau et intervalle(s) de réponse. Donc, en reprenant le schéma décrit tout au début de deuxième chapitre, sur la figure 4.4 notre processus de reconnaissance pourrait parfaitement intégrer une caméra de ce type.

Au sein de notre laboratoire Berry et al. [28, 17, 11] ont développé une plateforme de recherche composée d'un imageur 4 millions de pixels, d'un dispositif FPGA ALTERA Stratix, d'une interface IEEE1394, de plusieurs modules mémoires, de capteurs inertiels

et d'une connexion permettant l'ajout d'une carte contenant un dispositif DSP Texas Instruments.

L'intégration de ce capteur dans notre système pourrait faire l'objet de développements futurs orientés vers les applications rapides temps réel.

4.2.4 La perception multisensorielle

La méthodologie proposée repose sur une focalisation dans l'espace des paramètres. Il est possible de l'utiliser dans le cadre d'applications de localisation multi-sensorielles. Un tel travail a été mené par Cédric Tessier dans le cadre de sa thèse [106]. L'objectif recherché est de localiser un robot mobile grâce à des capteurs embarqués (caméra, télémètre laser, odomètre, gyromètre), et par l'observation d'indices géoréférencés au préalable dans la scène (bords de route, arbres, murs, ... Voir Fig. 4.5).

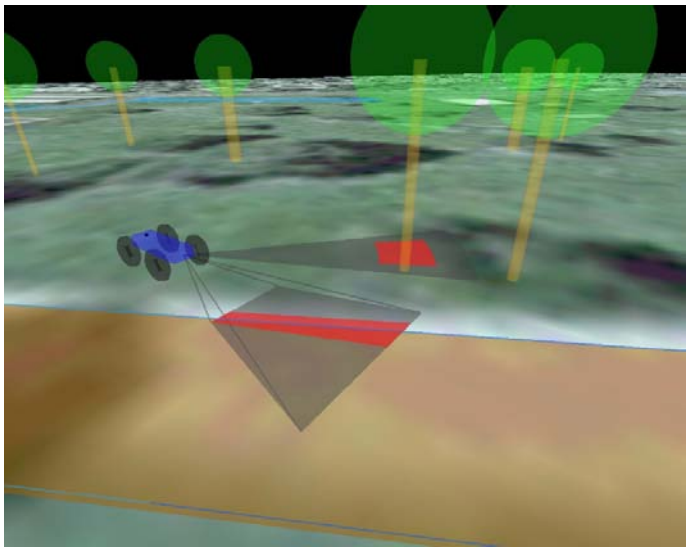


FIG. 4.5 – Exemple de perception multi-capteurs de l'environnement par un robot mobile.

Initialement, l'incertitude de positionnement du robot dans le monde est grande. Le système va sélectionner, au vu de critères similaires à notre approche, quel indice observer dans la scène et avec quel capteur pour préciser sa position. La zone de recherche des indices est calculée dans l'espace capteur et en fonction de l'erreur de positionnement du robot. Ainsi, le système doit choisir, au vu de la pertinence des indices, de leur coût de recherche et de la taille de la zone d'analyse, quel capteur utiliser pour détecter quel indice.

Une fois la détection réalisée, la position du robot est mise à jour. Les zones d'analyse sont resserrées (par focalisation : on précise non seulement la position des autres indices dans les repères capteurs, mais aussi les autres paramètres comme par exemple l'orientation du contour du bord de la route) et le processus est itéré en considérant un modèle d'évolution dynamique du robot.

Bibliographie

[1] J.Y. Aloimonos, I. Weiss, and A. Bandopadhay. Active vision. *International Journal on Computer Vision*, pages 333–356, 1987.

[2] J. Andrade-Cetto and A.C. Kak. *Wiley encyclopedia of electrical engineering*, chapter Object recognition, pages 449–470. John Wiley & Sons, Sup. 1, 2000.

[3] R. Aufrere. *Reconnaissance et suivi de route par vision artificielle, application à l'aide à la conduite*. PhD thesis, Université Blaise Pascal-Clermont II. Ecole Doctorale Science pour L'Ingenieur de Clermont-Ferrand, Juin 2001.

[4] I. Autio and K.T. Lindgren. Attention-driven parts-based object detection. In *Proc. 16th European Conference on Artificial Intelligence, ECAI*, pages 917–921, 2004.

[5] R. Bajcsy. An active observer. In *Image Understanding : 21st workshop : papers 1992 Jan. San Diego, Ca.*, pages 137–147, 1992.

[6] R. Bajcsy. Active perception. *IEEE Proceedings*, 76(8) :996–1006, August 1988.

[7] R.K. Bajcsy. Active perception vs passive perception. In *Computer vision : representation and control : 3rd workshop : papers 1985 oct : Bellaire, MI*, pages 55–59, 1985.

[8] E. Bayro-Corrochano, N. Trujillo, and M. Naranjo. The role of the quaternion fourier descriptors for preprocessing in neuralcomputing. In *Proceedings of the International Joint Conference on Neural Networks*, volume 4.

[9] E. Bayro-Corrochano, Noel Trujillo, and M. Naranjo. Quaternion fourier descriptors for the preprocessing and recognition of spoken words using images of spatiotemporal representations. *Journal of Mathematical Imaging and Vision*, 28(2) :179–190, 2007.

[10] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE PAMI*, 24(24) :509–522, April 2002.

[11] F. Berry and P. Chalimbaud. Smart camera and active vision : the active detector formalism. *SPIE Newsletters on Electronic Imaging*, 14(1), January 2004.

[12] Irving Biederman. Recognition-by-components : A theory of human image understanding. *Psychological Review*, 94(2) :115–147, 1987.

[13] Thomas Bülow and Gerald Sommer. Quaternion gabor filters for local structure classification. In *ICPR '98 : Proceedings of the 14th International Conference on Pattern Recognition-Volume 1*, page 808, Washington, DC, USA, 1998. IEEE Computer Society.

[14] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2) :121–167, June 1998.

[15] M.C. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *Proc. 5th European Conf. Comp. Vision*, pages 628–641, 1998.

[16] P.J. Burt and E.H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31,4 :532–540, 1983.

[17] P. Chalimbaud and F. Berry. Embedded active vision system based on an fpga architecture. In *EURASIP Journal on Embedded Systems*, 2007.

[18] R. Chapuis. *Localisation et Suivi de Route pour l'Aide à la Conduite.* L'Habilitation à Diriger des Recherches. Université Blaise Pascal Clermont-Ferrand. N° d'ordre :110, 2000.

[19] F. Chausse, N. Trujillo, R. Chapuis, and M. Naranjo. Object recognition by model based focus vision. In *AISTA 2004 (International Conference on Advances in Intelligent Systems - Theory and Applications) organized in cooperation with the IEEE Computer Society, Luxembourg, November 15-18*, 2004.

[20] M.M. Chun and J.M. Wolfe. *Blackwell Handbook of Perception*, chapter chapter 9 : Visual Attention. Version of July 7, 2000.

[21] X. Clady, F. Collange, F. Jurie, and P. Martinet. Object tracking with a pan tilt zoom camera application to car driving assistance. In *ICRA*, pages 1653–1658, 2001.

[22] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press ; 1st edition, (March 28, 2000).

[23] James L. Crowley. Cognitive vision research roadmap. Technical report, ECVision European Research Network for Cognitive AI-enabled Computer Vision Systems. Information society technologies (IST) programme, 2003.

[24] P. Dayan, S. Kakade, and P. Read. Learning and selective attention. *Nature Neuroscience*, 3((supp)) :1218–1223, Nov. 2000.

[25] G. Deco and B. Schürmann. A hierarchical neural system with attentional top-down enhancement of the spatial resolution for object recognition. *Vision Research*, 40 :2845–2859, 2000.

[26] Flückiger Delorm. *Perception et réalité. Une introduction à la psychologie des perceptions Neurosciences & Cognition*. De boeck, 2003.

[27] Flückiger Delorm. *Perception et réalité. Une introduction à la psychologie des perceptions Neurosciences & Cognition*, chapter Psychological and neurophysiological characteristics of visual recognition, pages 247–273. De boeck, 2003.

[28] F. Dias-Real, F. Berry, F. Marmoiton, and J. Serot. A configurable window-based processing element for image processing on smart cameras. In *IAPR Machnie, Vision and Application (MVA'07) Tokyo, Japan,*, Mai 2007.

[29] S.J. Dickinson, H.I. Christensen, J.K. Tsotsos, and G. Olofsson. Active object recognition integrating attention and viewpoint control. *Computer vision and image understanding*, 67(3) :239–260, September, 1997.

[30] B.A. Draper, J. Bins, and K. Baek. Adore : Adaptive object recognition. In *Computer Vision Systems : first international conference, ICVS'99, Las Palmas, Gran Canaria, Spain. Proceedings.*, pages 522–537. H.E. Christensen, ed., January 13-15, 1999.

[31] B.A. Draper, A.R. Hanson, and E. M. Riseman. Knowledge directed vision control learning integration. *Proceedings of the IEEE*, 84(11) :1625–1639, 1996.

[32] B.A. Draper and A. Lionelle. Evaluation of selective attention under similarity transformations. *Computer vision and image understanding.*, 100 :152–171, 10 August 2005.

[33] S. Edelman. *Encyclopedia of Artificial Intelligence*, chapter Visual perception, pages 1655–1664. Wiley-Interscience, New York, 1992.

[34] S. Edelman. Computational theories of object recognition. *Trends in Cognitive Sciences*, 1 :296–304, 1997.

[35] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples : an incremental bayesian approach tested on 101 object categories. In *CVPR, Workshop on Generative-Model Based Vision*, 2004.

[36] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages II :66–73, 2000.

[37] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *Int. J. Comput. Vision*, 61(1) :55–79, Jan. 2005.

[38] R. Fergus, P. Perona, and A. Zisserman. A object class recognition by unsupervised scale-invariant learning. In *In Proc. IEEE Conf. computer vision and pattern recognition.*, volume 2, pages 264–271, 2003.

[39] D.A. Forsyth and J. Ponce. *Computer Vision : a Modern Approach*. Prentice Hall, 2003.

[40] S. Frintrop and E. Rome. Simulating visual attention for object recognition. In *Proceedings of the Workshop on Early Cognitive Vision*, Isle of Skye, Scotland. May 2004.

[41] C. Garbay. *Les systèmes de vision*, chapter 7. Architectures logicielles et contrôle dans les systèmes de vision., pages 197–251. Hermes Science Publications, 2001.

[42] W.E.L Grimson and T. Lozano Pérez. Model based recognition and localization from sparse range or tactile data. *The international journal of robotics research*, 3. No. 3 :3–35, 1984.

[43] R.M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. 3(6) :610–621, November 1973.

[44] Gunther. Heidemann. Focus-of-attention from local color symmetries. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(7) :817–830, 2004.

[45] D. Heinke and G.W. Humphreys. Computational models of visual selective attention : a review. *Connectionist Models in Psychology*, 2005.

[46] J.M. Henderson and A. Hollingworth. High level scene perception. *Annual Review of Psychology*, 50 :243–271, 1999.

[47] K. Henke, S. R. Schweinberger, Klos T. Grigo, A., and W. Sommer. Specificity of face recognition : recognition of exemplars of non-face objects in prosopagnosia. *Cortex*, 34 :289–296, 1998.

[48] T. N. Hubel, D. H. & Wiesel. Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology*, 195 :215–243, 1968.

[49] L. Itti and C. Koch. Computational modelling of visual attention. *Nature Neuroscience*, 2 :194–203, March 2001.

[50] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE transactions on pattern analysis and machine intelligence*, 20(11) :1254–1259, November 1998.

[51] Anil K. Jain and Farshid Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recogn.*, 24(12) :1167–1186, 1991.

[52] Anil K. Jain, Robert P. W., and Jianchang Mao. Statistical pattern recognition : A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1) :4–33, 2000.

[53] M. Jägersand. Saliency maps and attention selection in scale and spatial coordinates : an information theoretic approach. In *Proc. Of the 5th Internacional Conference on Computer Vision, ICCV-95*, pages 195–202. Computer Society Press, 1995.

[54] J.M. Jolion (sous la direction de). *Les systèmes de vision*. Hermes Science Publications, 2001.

[55] T. Jost, N. Ouerhani, R. Wartburg, R. Müri, and H. Hügli. Assessing the contribution of color in visual attention. *Computer Vision and Image Understanding*, 100 :107–123, May 2005.

[56] N. G. Kanwisher, J. McDermott, and M. M. Chun. The fusiform face area : A module in human extrastriate cortex specialized for face perception. *Journal of Neuroscience*, 17 :4302–4311, 1997.

[57] C. Koch and S. Ullman. Shifts in selective visual attention : towards the underlying neural circuitry. *Human Neurobiology*, 4 :219–227, 1985.

[58] Anne M. Landraud and Suk Oh. Yum. Texture segmentation using local phase differences in gabor filtered images. In *ICIAP '95 : Proceedings of the 8th International Conference on Image Analysis and Processing*, pages 447–452, London, UK, 1995. Springer-Verlag.

[59] T.S. Lee. Image representation using 2d gabor wavelets. *IEEE Transactions on pattern analysis and machine intellitgence*, 18(10) :1–13, 1996.

[60] T. Leung and J. Malik. Recognizing surface using three-dimensional textons. In *7th Int'l Conf. on Computer Vision, Corfu, Greece*, September 1999.

[61] T.K. Leung, M.C. Burl, and P. Perona. Finding faces in cluttered scenes using random labelled graph matching. In *In fifth Intl. Conf. on Computer Vision.*, pages 637–644, 1995.

[62] D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision Corfu, Greece*, pages 1150–1157, 1999.

[63] D. Lowe. Towards a computational model for object recognition in it cortex. In *First IEEE International Workshop on Biologically Motivated Computer Vision, Seoul, Korea*, pages 20–31, 2000.

[64] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(issue 2), November 2004.

[65] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3) :355–395, 1987.

[66] Gareth Loy. Fast computation of the wavelet transform. In *DICTA2002. Digital Image Computing Teechniques and Applications*, 2002.

[67] J. Machrouh and P. Tarroux. Attentional mechanisms for interactive image exploration. *EURASIP Journal on Applied Signal Processing*, 2005(14) :2391–2396, 2005. doi :10.1155/ASP.2005.2391.

[68] S. Mallat. Wavelets for a vision. *Proceedings of the IEEE.*, 84(4) :604–614, 1996.

[69] D. Marr. *Vision : a computational investigation into the human representation and processing of visual information*. San Francisco : W. H. Freeman., 1982.

[70] I. Marsic. Data-driven shifts of attention in wavelet scale space. Technical report, CAIP Center, Rutgers University, September 1993.

[71] A.M. Martinez and R. Benavente. The AR face database. Technical Report no. 24, Computer Vision Center (CVC) at the U.A.B., 1998.

[72] B. Moghaddam, T. Jebara, and A. Pentland. Bayesian face recognition. Technical report, Mitsubishi Electric Research Laboratories, 2002.

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

[74] S. B. Most, B. J. Scholl, E. R. Clifford, and D. J. Simons. What you see is what you set : Sustained inattentional blindness and the capture of awareness. *Psychological Review*, 112(1) :217–242, 2005.

[75] M. Murphy, A. Torralba, and W. Freeman. Using the forest to see the trees : a graphical model relating features, objects, and scenes. volume 16. MIT Press, 2003.

[76] A. Oliva, A. Torralba, M.S. Castelhano, and J.M. Henderson. Top-down control of visual attention in object detection. *IEEE Proceedings of the International Conference on Image Processing*, I :253–256, 2003.

[77] P. Oliva, A. Schyns. Coarse blobs or fine edges ? evidence that information diagnosticity changes the perception of complex visual stimuli. *Cognitive Psychology*, 34 :72–107, 1997.

[78] Søren I. Olsen. Exemplar based recognition of visual shapes. In *SCIA*, pages 852–861, 2005.

[79] M. Oren, C.P. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. pages 193–99, 1997.

[80] K. Pahlavan, T. Uhlin, and Eklund J.O. Active vision as a methodology. *Robotics and Automated Systems*, pages 19–46, 1993.

[81] C. Papageorgiou and T. Poggio. A trainable system for object detection. *Int. J. Comput. Vision*, 38(1) :15–33, 2000.

[82] C.P. Papageorgiou, Oren M., and T. Poggio. A general framework for object detection. In *Proceedings of 6th International Conference on Computer Vision*, 1998.

[83] C.P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *ICCV '98 : Proceedings of the Sixth International Conference on Computer Vision*, page 555, Washington, DC, USA, 1998. IEEE Computer Society.

[84] T.V. Pham and W.M. Smeulders. Object recognition with uncertain geometry and uncertain part detection. *Computer vision and image understanding*, 99 :241–258, January 2005.

[85] H. Prendinger, M. Ishizuka, and T. Yamamoto. The hyper system : Knowledge reformation for efficient first-order hypothetical reasoning. In *Pacific Rim International Conference on Artificial Intelligence*, pages 93–103, 2000.

[86] O. Ramström and H.I. Christensen. Object detection using background context. In *Procedings of the 17th International Conference on Pattern Recognition (ICPR'04). IEEE*. Computer Society IEEE, 2004.

[87] M. Riesenhuber. Object recognition in cortex : Neural mechanisms, and possible roles for attention. *Neurobiology of Attention, (Eds. L. Itti, G. Rees, and J. Tsotsos), Elsevier*, pages 279–287, 2005.

[88] M. Riesenhuber and T. Poggio. Models of object recognition. *Nature Neuroscience*, 3 :1199–1204, November 2000.

[89] E.T. Rolls. *MIT Encyclopedia of the Cognitive Sciences*, chapter Object recognition, animal studies, pages 613–615. MIT Press : Cambridge, Mass, 1999.

[90] U. Rutishauser, D. Walther, C. Koch, and P. Perona. Is bottom-up attention useful for object recognition ? In *International Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, volume 2, pages 37–44, 2004.

[91] R.E. Schapire. A brief introduction to boosting. In *IJCAI*, pages 1401–1406, 1999.

[92] H. Schneiderman. Feature-centric evaluation for efficient cascaded object detection. In *CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on Publication*, pages II 29–36, 2004.

[93] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, 56(3) :151–177, 2004.

[94] T. Serre, L. Wolf, and T. Poggio. A new biologically motivated framework for robust object recognition. Technical report, 2004.

[95] B. Shiele. *Reconnaissance d'Objets utilisant of Histogrammes Multi-. Dimensionels of Champs Receptifs*. PhD thesis, INPG, July 1997.

[96] D. J. Simons. To see but not to see : Review of inattentional blindness by a. mack and i. rock (1998). *Journal of Mathematical Psychology*, 43 :165–171, 1999.

[97] D. J. Simons. Attentional capture and inattentional blindness. *Trends in Cognitive Sciences*, 4 :147–155, 2000.

[98] D. J. Simons and C. F. Chabris. Gorillas in our midst : Sustained inattentional blindness for dynamic events. *Perception*, 28 :1059–1074, 1999.

[99] D. J. Simons, S. R. Mitroff, and S. L. (2003) Franconeri. *Perception of Faces, Objects, and Scenes : Analytic and Holistic Processes*, chapter Scene perception : what we can learn from visual integration and change detection, pages 335–355. Oxford : Oxford University Press, 2003.

[100] K. Sobottka and I. Pitas. Looking for faces and facial features in color images. *Pattern Recognition and Image Analysis : Advances in Mathematical Theory and Applications*, 7(1), 1997.

[101] Y. Sun and R. Fisher. Object-based visual attention for computer vision. *Informatics research report EDI-INF-RR-0213*, June, 2004.

[102] R. Szeliski. Image alignment and stitching : A tutorial. *Foundations and Trends in Computer Graphics and Computer Vision*, 2(1) :1–104, December 2006.

[103] B. Takacs and H. Wechsler. A dynamical and multiresolution model of visual attention and its application to facial landmark detection. *Computer Vision and Image Understanding*, 70(1) :63–73, April, 1998.

[104] M.J. Tarr. *Encyclopedia of Psychology*, chapter Visual pattern recognition.

[105] M.J. Tarr. *Perception of Faces, Objects, and Scenes : Analytic and Holistic Processes*, chapter Visual Object Recognition : Can a Single Mechanism Suffice ?, pages 177–211. Oxford, UK : Oxford University Press, 2003.

[106] C. Tessier, C. Debain, R. Chapuis, and F. Chausse. Characterization of feature detection algorithms for a reliable vehicle localization. In *6th IFAC Symposium on Intelligent Autonomous Vehicles, Toulouse, France*, September 3-5 2007.

[107] A. Torralba. Contextual influences on saliency. Technical report, AI Memo 2004-2009. Massachusetts Institute of Technology - Computer Science and Artificial Intelligence Laboratory, April 2004.

[108] A. Torralba, K. Murphy, W. Feeman, and M. Rubin. Context-based vision system for place and object recognition. In *In Intl. Conf. Computer Vision. Ninth IEEE International Conference on Publication*, volume 1, pages 273– 280, Oct. 2003.

[109] A. Torralba, K.P. Murphy, and W.T. Freeman. Sharing visual features for multiclass and multiview object detection. In *CVPR*, 2004.

[110] A.M. Treisman and N.G. Kanwisher. Perceiving visually presented objects : recognition, awareness, and modularity. *Current opinion in neurobiology.*, 8 :218–226, 1998.

[111] N. Trujillo, R. Chapuis, F. Chausse, and Blanc C. On road simultaneous vehicle recognition and localization by model based focused vision. In *IAPR Conference on Machine Vision Applications*, 2005.

[112] N. Trujillo, R. Chapuis, F. Chausse, and M. Naranjo. Object recognition : A focused vision based approach. In *(to appear) Lecture Notes in Computer Science (LNCS)*, pages 631–642, 2007.

[113] J.K. Tsotsos. On the relative complexity of active vs. passive visual search. *International Journal of Computer Vision*, 7(2) :127–141, 1992.

[114] J.K. Tsotsos, S.M. Culhane, W. Yan Key Wai, Y. Lai, N. Davis, and F. Nuflo. Modeling visual attention via selective tuning. *Artificial intelligence*, 78 :507–545, 1995.

[115] M. Turk and A. Pentland. Eigenfaces for recognition. *J. of Cognitive Neuroscience*, 3(1), 1991.

[116] V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, New York, 1998.

[117] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Conference on computer vision and pattern recognition*, volume 1, pages 511–518.

[118] P. Viola and M. Jones. Robust real-time object detection. In *ICCV*, 2001.

[119] D. Walther, L. Itti, M. Riesenhuber, T. Poggio, and C. Koch. Attentional selection for object recognition- a gentle way. In *Second IEEE International Workshop, BMCV*, pages 472–479, 2002.

[120] D. Walther, U. Rutishauser, C. Koch, and P. Perona. On the usefulness of attention for object recognition. In *2nd Workshop on Attention and Performance in Computational Vision, at the European Computer Vision Confererence (ECCV04)*, pages 96–103, 2004.

[121] D. Walther, U. Rutishauser, C. Koch, and P. Perona. Selective visual attention enables learning and recognition of multiple objects in cluttered scenes. *Computer Vision and Image Understanding.*, 100 :41–63, 2005.

[122] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *ECCV (1)*, pages 18–32, 2000.

[123] K. Yanai and K. Deguchi. A multi-resolution image understanding system based on multi-agent architecture for high-resolution images. *PAPER Special issue on Machine Vision Applications*, E84-D(12) :1642–1650, 2001.

[124] M. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images : A survey. *IEEE Transactions on Pattern Analysys and Machine Intelligence*, 24(1) :34–58, 2002.

[125] A. Yarbus. *Eye Movements and Vision [translated from Russian by Haigh B]*. New York : Plenum Press, 1967.

[126] I.T. Young. Recursive gabor filtering. *IEEE Transactions on Signal Processing*, 50(11) :2798–2805, 2002.

[127] Y. Zhengrong and D. Castañon. Partially occluded object recognition using statistical models. *International Journal of Computer Vision.*, 49(1) :57–78, August, 2002.

[128] S.W. Zucker. Computer vision and human perception : an essay on the discovery of constraints. In *Artificial intelligence : 7th international joint conference : Papers 1981 aug. Vancouver, Canada*, pages 1102–1116, 1981.

Table des figures

1.1 a) Image originale. b) Représentation de l'image originale en basses fréquences. c) Représentation de l'image originale en hautes fréquences. . . 24
1.2 Exemple de la représentation par composantes structurales. a) Objet perçu. b) Objet représenté par des géons. 26
1.3 Schéma proposé par Biederman décrivant les différentes phases présumées pour la reconnaissance d'objets (repris de [12]). 27
1.4 Exemple de l'apparence moyenne d'un objet. 28
1.5 Exemple d'attention visuelle. a) Top-down. La zone marquée correspond à la région où l'objet « véhicule » est attendu par l'observateur. b) Dans cet exemple, parmi l'ensemble de lettres A, celle qui est colorée attire l'attention de l'observateur indépendamment du but. 29
1.6 Tableau comparatif entre les avantages et inconvénients de la représentation structurelle et de celle basée par l'apparence. 31
1.7 Schéma blocs pour l'appariement de modèle. a) Phase d'entraînement. b) Phase de classification 34
1.8 Exemple de « patches » pour la modélisation de visage. Chaque « patch » correspond à une région de l'image d'apparence caractéristique (dans cet exemple les patches correspondent aux yeux et à la bouche). 36
1.9 Le spectre des mécanismes attentionnels proposé par Tsotsos [113]. . . . 43
1.10 Approche modulaire d'un système de perception. 51
1.11 Approche modulaire d'un système de perception. 52
1.12 Schéma simplifié du processus de perception proposé. 53

2.1 Schéma montrant le système que l'on propose pour la reconnaissance d'objets. 60
2.2 Structure hiérarchique de l'objet. 65
2.3 Exemple d'une cellule 66
2.4 Exemple d'une grille de cellules. Pour deux objets de taille différentes on a un nombre fixe de cellules. 67
2.5 Exemple d'une grille multi-résolution. 69

2.6 Collecte des statistiques pour la caractérisation des N paramètres, $\zeta_1, \zeta_2, \ldots, \zeta_N$, des M_c cellules de la grille, à partir des T exemples. $\zeta^t_{m,n}$: t-ième image, m-ième cellule et n-ième opérateur. 71

2.8 Exemple du modèle du visage après la réduction de dimensionnalité. Dans cet exemple, seulement l'aspect géométrique est affiché (le positionnement des cellules résultantes). 79

2.7 A titre d'exemple, on montre les cellules restantes après l'élimination de cellules non fonctionnelles. En rouge sont affichés les M^p *parties* qui composent le modèle de l'objet. 80

2.9 Principe de focalisation 83

2.10 Stratégie du système de reconnaissance. 84

2.11 Représentation de l'ensemble des hypothèses par une structure d'arbre de décision. 88

2.12 Régions d'analyse pendant le processus de reconnaissance. 92

2.13 Lors de l'apprentissage dans une région de la taille d'une cellule, le paramètre $\zeta_{m,n}$ issu de l'opérateur n et appartenant à Λ_m, est supposé suivre une loi normale telle que $\zeta_{m,n} \sim \mathcal{N}(\bar{\zeta}_{m,n}, \sigma_{\zeta_{m,n}})$. Lors de la reconnaissance, dans une région de la taille d'une cellule, ce paramètre est supposé suivre une loi uniforme avec une valeur maximale de $\frac{1}{D_n}$, supposant que l'opérateur a une dynamique D_n. Ainsi, la probabilité qu'a le paramètre $\zeta_{m,n}$, issu de l'opérateur n, d'appartenir à l'intervalle $(a_{m,n} \leq \zeta_{m,n} \leq b_{m,n})$ est donnée par $\frac{4\sigma_{\zeta_{m,n}}}{D_n}$. 94

2.14 Modèle du « carré ». Les points rouges correspondent à la position moyenne des quatre coins composant l'objet. Les ellipses montrent l'intervalle permis pour la position des quatre coins. Ce modèle prend en compte des translations, des changements en échelle ainsi que des rotations dans le plan 2D. 98

2.15 Résultats des simulations des trois exemples de détection du « carré ». Dans la colonne à gauche sont affichés les modèles mis à jour lors de détections/non détection des coins. Le points rouges représentent la position probable des quatre coins du carré, alors que les ellipses donnent l'intervalle permis sur ces positions. Dans le premier cas (A), les quatre coins ont été supposés détectés. Pour le deuxième exemple (B), les trois premières détections ont réussi alors que la quatrième ne l'est pas. Pour le dernier (C), uniquement les deux premières détections ont été supposées réussies. 99

2.16 Exemple d'évolution de l'algorithme et du rapport de vraisemblance. Sur la figure montrée ci-dessus, on présente un exemple du parcours de l'arbre de recherche à partir de la détection des bonnes hypothèses (cercles rouges) et des fausses hypothèses (cercles bleus). Les cercles pointillés indiquent les non détections. Pour faciliter la compréhension, le parcours de l'arbre s'est fait de haut en bas et de gauche à droite. Les flèches bleues montrent le parcours que fait l'algorithme en partant sur de fausses hypothèses. A côté de chaque cercle, on présente la mise à jour du rapport de vraisemblance une fois que la *partie* en question a été détectée. Les légendes à coté des flèches (lecture uniquement au retour) correspondent à la mise à jour du rapport de vraisemblance qui prend en considération la non détection de la *partie* recherchée (à la pointe de la flèche bleue). Le retour en arrière (flèches vertes) se fait lorsque que l'on considère, au moyen du rapport de vraisemblance, qu'il n'est plus pertinent d'explorer cette branche de l'arbre (branch and bound). A son tour, la flèche rouge pointillé nous indique le parcours de l'arbre avec toutes les hypothèses possibles jusqu'à la détection de l'objet. 103

2.17 L'objet considéré comme appartenant à une scène 3D observée. A la différence de la phase d'apprentissage, ici la caméra s'est déplacée par rapport à l'objet par des paramètres de translation (X,Y,Z) et des rotations (α,β,γ). 105

3.1 Masques du filtre de Gabor. a) Partie réelle. b) Partie imaginaire. c) Phase locale. 111

3.2 Analyse de phase avec le filtre de Gabor. a) Image de test. b,f) Partie réelle. c,g) Partie imaginaire. d,h) Module. e,i) Phase locale, pour le filtre de Gabor avec paramètres $\sigma_x = \sigma_y = 5$, $\lambda = 20$, $\theta = 90^\circ$ et $\theta = 0^\circ$ respéctivement. 112

3.3 Exemple de carte d'orientations pour trois images avec différents éclairages. a) Éclairage droit. b) Éclairage gauche. c) Éclairage au centre. d) Carte d'orientations pour a), b), et c), respectivement. La couleur du rouge foncé au bleu clair, représente l'angle d'orientation pour $0^\circ, 30^\circ, 60^\circ, 90^\circ, 120^\circ, 150^\circ$ respectivement. 113

3.4 a) Image de test peu contrastée. b) Carte d'orientations. c) Réponse du filtre à 0°. d) Réponse du filtre à 90°. Pour cet exemple, même si l'éclairage est très défavorable, avec l'analyse de phase locale on est capable d'extraire la structure à l'intérieur du visage. 114

3.5 Exemples de visages appartenant à l'ARdatabase utilisés lors de l'apprentissage du modèle. 116

3.6 Exemples des images, appartenant à la Caltech database, utilisées pour le test de notre algorithme. 116

3.7 Exemple de la décomposition pyramidale pour une image donnée. Le modèle (de quatre coins ici) est superposé sur l'image montrant la taille de l'objet le plus petit (carré jaune) et le plus grand (carré rouge) qui peut être détecté. Les ellipses montrent l'intervalle de confiance où les quatre coins peuvent être localisés. 120

3.8 Le modèle de coins sur-échantillonné et superposé sur l'image originale. On observe comment, avec seulement trois échelles d'analyse, l'espace-échelle est couvert complètement. Dans cet exemple, l'objet le plus petit et l'objet le plus grand correspondent à une taille de 128 et 1024 pixels respectivement. 120

3.9 Modèle réduit du visage après la réduction de dimensionnalité. Le nombre de cellules qui restent, après l'élimination des cellules non discriminantes, est d'environ 300 sur 1344 au total. Il faut préciser que les cellules résultantes sont celles qui ont au moins un paramètre pertinent. 121

3.10 Intervalle de confiance (pour $r = 0$) sur la position de l'ensemble des *parties* après l'apprentissage. 122

3.11 Résultat de la hiérarchisation des primitives : première hypothèse générée. 123

3.12 Région d'intérêt dans le seul espace géométrique, pour la position attendue de la première hypothèse émise. 124

3.13 Chaque ellipse montre : a) les régions d'intérêt initiales pour toutes les *parties* de l'objet, b) les régions d'intérêt réduites après la détection de la première *partie* sélectionnée et la mise à jour du modèle. 126

3.14 Sélection de la résolution. Chaque ligne représente les différents états du modèle lors de la reconnaissance. Les première, deuxième et troisième colonnes correspondent aux trois niveaux de résolution respectivement (de plus élevé au plus bas). 127

3.15 a) Exemple montrant les effets de la focalisation dans trois des quatre niveaux du spectre attentionnel : la sélection de la région d'intérêt dans le plan image, la sélection des opérateurs et la délimitation de l'intervalle de réponse des opérateurs impliqués. Les points d'intérêt, montrés dans la « carte dynamique de saillance », correspondent aux régions locales où les opérateurs de direction du contour et niveau de gris, ont répondu dans l'intervalle attendu pour la *partie* recherchée. Dans l'étape *détection*, le point jaune correspond au candidat le plus pertinent (couleur plus foncée dans la carte de saillance), alors que le point rouge correspond à la position réelle de la *partie* qui est cherchée. b) Points d'intérêt obtenus par une approche classique. 129

3.16 Cette figure montre un exemple d'évolution de l'algorithme proposé. a) Position moyenne des *parties* du visage. b) Région d'intérêt de chaque hypothèse générée, k étant l'état du processus de reconnaissance. c) Les détections réalisées dans la région d'intérêt préalablement définie. d) Les nouvelles régions d'intérêt estimées après la mise à jour du modèle. Une fois que le modèle est positionné sur le visage, la classification finale est faite avec un classificateur de type SVM. 131

3.17 Exemple d'adaptation en échelle du modèle lors du test de plusieurs fausses hypothèses. La dernière image correspond à la reconnaissance réussie du visage. Les trois niveaux de résolution du modèle du visage sont affichés, ce qui explique les points rouges dans la partie supérieure gauche de l'image. 132

3.18 a) L'image à droite montre les traces du mouvement oculaire lors qu'un sujet explore le portrait à gauche (extrait du travail de A. Yarbus (1967) [125]. b) Exemple du parcours de déplacement de l'attention lors du processus de reconnaissance. 133

3.19 Différentes hypothèses testées. Pour chaque niveau de résolution, les points rouges correspondent à la position attendue des *parties* du visage. Lors de la recherche du visage, les « fixations » sautent d'une région à l'autre afin de tester plusieurs candidats et trouver la meilleure configuration possible en accord avec le modèle. La dernière image correspond à la détection correcte du visage. 134

3.20 Exemples de reconnaissance de visages en utilisant le modèle tel qu'issu de la phase d'apprentissage (modèle complet). 136

3.21 Exemples de reconnaissance de visages en utilisant le modèle tel qu'issu de la phase d'apprentissage (modèle complet). 137

3.22 Exemples de reconnaissance de visages avec le modèle réduit. Uniquement les 50 *parties* les plus corrélées sont gardées. Avec ce modèle le temps de détection est d'environ 1s par image. 139

3.23 Test 1 : reconnaissance et suivi du visage se déplaçant dans le plan horizontal et vertical. La détection présente une certaine robustesse lors des occultations partielles (voir les lignes c et d de la séquence). 143

3.24 Test 2 : reconnaissance et suivi de visage se déplaçant en profondeur (adaptation en échelle). 144

3.25 Modèles du piéton après réduction de dimensionnalité (uniquement la position des *parties*). a) Modèle basé sur l'orientation du contour. b) Modèle basé sur l'orientation du contour et le niveau de gris. 147

3.26 Repère cellules utilisé. 149

3.27 Paramètres utilisés pour la reconnaissance de piétons. 149

3.28 Exemple de grilles de cellules obtenues 151

3.29 Suivi d'un piéton se déplaçant de gauche à droite dans le plan horizontal. Dans cette séquence, les deux piétons se superposent à un moment donné. Ici, l'algorithme « perd de vue » le piéton initialement reconnu du fait du manque de description autre que fréquentielle. L'ellipse dans la première image de la séquence montre la région d'intérêt initiale pour toutes les *parties*. Dans la deuxième image de la séquence, les ellipses correspondent aux régions d'intérêt de chaque *partie* réduites après la mise à jour du modèle. 156

3.30 Suivi d'un piéton se déplaçant de gauche à droite. Ici, le modèle utilisé pour l'algorithme de reconnaissance est composé de deux types des caractéristiques : celles basées sur l'information fréquentielle, et celles basées sur le niveau de gris. A la différence de l'exemple présenté en §3.4.4.1, ici l'algorithme ne « perd pas de vue » le piéton reconnu initialement. Ceci démontre l'intérêt d'intégrer des caractéristiques multiples pour la représentation des objets. 157

3.31 Exemple d'un piéton se déplaçant de gauche à droite et en s'éloignant de la caméra. Nous pouvons observer la bonne adaptation du modèle à l'objet dans la phase de suivi lorsque l'objet varie en échelle. 159

3.32 Exemple du piéton s'éloignant de la caméra. Ici, les paramètres de la caméra (angle et hauteur) sont initialisés pour que le système soit « attentif » aux piétons qui puissent apparaître tout au début du couloir. Le fait d'intégrer de l'information 3D sur la scène, aide à réduire significativement les fausses détections. Les objets qui sont détectés sont ceux qui ont une cohérence en accord avec la structure de la scène : des piétons apparaissant de grande taille en bas, et de petite taille vers l'horizon. La prise en compte de cette information entraîne une amélioration de la détection du piéton lors du suivi de celui-ci. 161

4.1 Exemple de l'utilisation du filtre quaternion de Gabor comme descripteur de l'objet. Dans (a-1), les étoiles représentent les maximaux locaux (voisinage de 8 pixels) de l'image filtrée (a-2). (b-1) et (b-2) correspondent à l'image extraite de la même séquence 3s plus tard. Les quatre masques du quaternion de Gabor sont affichés dans (c) (partie réelle, et trois imaginaires i, j, k [13, 9]). Dans cet exemple σ_x et σ_y sont adaptées de façon que la structure globale de l'objet soit captée. Il nous faut donc un seul descripteur pour quasiment décrire la structure globale. 167

4.2 Exemple du modèle de piéton basé sur des ondelettes de Gabor. A1 : carré. A2, A3 : cercle. A4 : étoile. A5 : point. 168

4.3 Ensemble d'hypothèses représentées par une structure d'arbre de recherche. Au niveau de base, c'est-à-dire au niveau zéro du processus, plusieurs hypothèses pourront être testées en parallèle. A son tour, chaque candidat engendré par la détection d'une *partie* hypothèse Λ_m, sera traité sous forme séquentielle. Un processus à part sera créé lors du test de chaque candidat. La « mort » d'un processus sera donnée par la règle de décision donnée par le rapport de vraisemblance (voir §2.4.6). 169
4.4 Acquisition de données avec une « caméra intelligente ». Ce module s'intégre d'une façon naturelle dans le cadre que l'on propose. 171
4.5 Exemple de perception multi-capteurs de l'environnement par un robot mobile. 172

Résumé

La problématique scientifique abordée concerne la reconnaissance visuelle d'objets s'inscrivant dans une scène observée. Nous proposons une méthodologie qui va de la définition et la construction du modèle de l'objet, jusqu'à la définition de la stratégie pour la reconnaissance ultérieure de celui-ci. Du point de vue de la représentation, cette approche est capable de modéliser aussi bien la structure de l'objet que son apparence ; à partir de caractéristiques multiples. Celles-ci servent d'indices d'attention lors de la phase de reconnaissance. Dans ce cadre, reconnaître l'objet revient à « instancier » ce modèle dans la scène courante. La tâche de reconnaissance correspond à un processus actif de génération/vérification d'hypothèses régi par le principe de focalisation. Ce dernier agissant sur quatre niveaux du « spectre attentionnel » : la sélection des opérateurs pour le traitement bas niveau, la sélection de l'intervalle d'action de ceux-ci, la sélection de la résolution et la sélection de la région d'intérêt dans l'image. Le fait d'agir sur tous ces niveaux, entraîne une diminution de la combinatoire implicite dans une problématique de recherche visuelle. Sous un regard plutôt unifié, le mécanisme de contrôle de l'attention, du type bottom-up $\leftrightarrow$ top-down, reste implicite dans la stratégie globale de reconnaissance. La « focalisation progressive » et la représentation hybride du modèle, permettent de tirer profit des deux types de représentation classiques. D'une part, la structure de l'objet permet de focaliser le processus de reconnaissance à partir d'observations locales, d'autre part, une fois détectée la région probable de l'objet, la décision finale est faite à partir de l'apparence de celui-ci. Dans le cadre proposé, en intégrant des connaissances sur la structure de la scène (paramètres 3D), d'autres tâches comme celles de la localisation et du suivi sont intégrées d'une façon naturelle. La prise en compte de ces paramètres permet d'estimer l'évolution de la zone d'intérêt dans l'image, lorsque l'objet évolue dans le monde 3D. La méthodologie proposée a été testée pour la reconnaissance, la localisation et le suivi de visages et de piétons.

Abstract

This report deals with the scientific problem of objects visual recognition within an observed scene. We propose an approache going from object's model building to the definition of a strategy for its future recognition. From the representation point of view, this methodology can represente the structure of the object as well as its appearance from multiple features. These last ones are used as attentional clues during the recognition stage. In this framework, the recognition of the object consists in instantiate it in the current scene. The recognition task is an active process of hypothesis generation/verification driven by a focusing principle. Focus acts on four levels of the "<attentional spectrum" : low level operators selection, action interval of these operators, resolution selection and image region of interest selection. The result is a reduction of the native combinatorial complexity of any visual seek process. From a rather unified point of view, the attention control mechanism both top-down and bottom-up, is part of the global recognition strategy. The "progressive focus of attention" and the hybrid representation of the objects makes itu possible to benefit from both the classical representations. On one side, the object structure makes it possible to apply the focus of attention on local parts. On the other side, the final decision is made according to the appearance. In the proposed framework, with the addition of information about the structure of the scene (3D parameters), other tasks such as localization and tracking are naturaly integrated. Thes 3D parameters are taken into account to estimate de evolution of the region of interest in the image as the object evolves in the 3D scene. The proposed methodology has been tested for face or pedestrians recognition, localization and tracking.