HAL
open science

Application du Modèle à Distribution de Points au corps humain pour la ré-identification de personnes

Olivier Huynh

To cite this version:

Olivier Huynh. Application du Modèle à Distribution de Points au corps humain pour la ré-identification de personnes. Vision par ordinateur et reconnaissance de formes [cs.CV]. Université Paris sciences et lettres, 2016. Français. NNT : 2016PSLEM032 . tel-01632375

HAL Id: tel-01632375

https://pastel.hal.science/tel-01632375

Submitted on 10 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE DE DOCTORAT

de l'Université de recherche Paris Sciences et Lettres
PSL Research University

Préparée à MINES ParisTech

Application du Modèle à Distribution de Points au corps humain pour la ré-identification de personnes

Ecole doctorale n°432

Sciences des métiers de l'ingénieur

Spécialité « Informatique temps-réel, robotique et mathématique »

COMPOSITION DU JURY :

M. Jean-Philippe THIRAN
Professeur, EPFL, Président

M. Antoine MANZANERA
Maître de Conférence, ENSTA ParisTech, Rapporteur

M. Mounim EL YACOUBI
Professeur, Telecom SudParis, Rapporteur

M. Bogdan STANCIULESCU
Maître de Conférence, Mines ParisTech, Examinateur

M. Philippe FUCHS
Professeur, Mines ParisTech, Examinateur

Soutenue par **Olivier HUYNH**
le 31/05/2016

Dirigée par **Philippe FUCHS**
Encadrée par **Bogdan STANCIULESCU**

Remerciements

Les années consacrées à cette thèse marqueront durablement ma vie tant sur le plan professionnel qu'au niveau humain.

Mes remerciements s'adressent tout d'abord aux membres du jury pour avoir accepté d'évaluer ce travail. En particulier, je souhaite vivement remercier M. Antoine MANZANERA et M. Mounim EL YACOUBI d'avoir rapporté ce manuscrit. Leurs rapports détaillés et leurs précieuses remarques m'ont permis de porter un regard nouveau sur l'ensemble de mes travaux. Je remercie M. Jean-Philippe THIRAN d'avoir présidé ce jury et animé une discussion des plus intéressantes ouvrant sur de nouvelles perspectives. Je remercie également M. Bogdan STANCIULESCU pour m'avoir accompagné tout au long de cette thèse et pour l'autonomie qu'il m'a accordée.

Je tiens grandement à remercier mes proches pour tout le soutien qu'ils m'ont apporté au cours de ces années. Notamment, ma conjointe pour avoir supporté mes horaires de travailleur nocturne, mes parents et mon oncle VTT pour leurs conseils, leurs relectures et leur zèle orthographique.

Ce doctorat a été pour moi une expérience humaine des plus chaleureuses grâce à la présence de mes collègues du Centre de Robotique. Je ne vais pas les énumérer exhaustivement mais je pense principalement à Houssem, Fernando, Martyna, Papo-Bruno, Jun, Tao-Jin, Zhuowei, Ravi, Jorge, Florent et Martin. Certains d'entre vous êtes devenus des amis proches et j'espère que nous entretiendrons ces liens pour très longtemps.

Table des matières

1 Introduction **1**
1.1 Contexte . . . 1
1.2 Objectifs de la thèse . . . 2
1.3 Contributions . . . 2
1.4 Organisation du manuscrit . . . 3

2 Détection de Personnes **5**
2.1 Introduction . . . 5
2.2 Techniques générales . . . 6
2.3 Etat de l'art - Fenêtre Glissante . . . 7
2.3.1 Principe général . . . 7
2.3.2 Approches existantes . . . 9
2.4 Approche retenue . . . 14
2.4.1 Caractéristiques . . . 15
2.4.2 Classification . . . 20
2.5 Résultats et validations . . . 24
2.5.1 Dataset utilisée . . . 24
2.5.2 Évaluations . . . 25
2.5.3 Portage sur smartphone . . . 30

3 Caractérisation de la structure corporelle par Modèle à Distribution de Points **33**
3.1 Introduction . . . 33
3.2 Représentation de la forme d'une personne . . . 34
3.3 État de l'art - Alignement de Modèle à Distribution de Points . . . 38
3.4 Pré-requis à l'alignement d'un PDM . . . 44
3.4.1 Définition du modèle . . . 44
3.4.2 Annotation d'une nouvelle Dataset . . . 46
3.5 Modèle d'apparence par boosting sur une forme paramétrique . . . 48
3.5.1 Modélisation paramétrique . . . 48
3.5.2 Modèle d'apparence par GentleBoost . . . 50
3.5.3 Évaluations et validations . . . 57
3.5.4 Limitations et discussions . . . 61
3.6 Régression de forme par classification des déformations . . . 63
3.6.1 Cascade de régressions de forme . . . 63
3.6.2 *Clustering* et classification des déformations . . . 64
3.6.3 Discussions . . . 74

4 Ré-identification de personnes **77**
4.1 Introduction . . . 77
4.2 État de l'art . . . 78
4.3 Renforcement de signatures basées sur la couleur . . . 81
4.3.1 Histogramme de couleurs HSV . . . 81
4.3.2 Modèle d'ensemble de couleurs . . . 81
4.3.3 Renforcement structurel . . . 82
4.4 Signature de forme sur les amers d'un PDM . . . 85
4.4.1 Shape Context . . . 85
4.4.2 Évaluations et discussions . . . 86

5 Conclusion et Perspectives **89**
5.1 Résumé 89
5.2 Futurs travaux 90

Publications **91**
.1 Conférences internationales avec comité de relecture 91

Bibliographie **93**

Chapitre 1

Introduction

Sommaire

1.1 Contexte 1
1.2 Objectifs de la thèse 2
1.3 Contributions 2
1.4 Organisation du manuscrit 3

1.1 Contexte

Depuis la fin des années 1990, les avancées dans le domaine de vision par ordinateur et de la puissance de calcul ont rendu possible la réalisation de systèmes performants temps réel pour l'analyse et la compréhension des images.

Les besoins grandissants en sécurité et l'augmentation des flux de déplacement de personnes font que les systèmes de vidéo-surveillance prennent de plus en plus d'ampleur (+37% de caméras déclarées entre 2010 et 2011 [1]). Un système de vidéo-surveillance, ou vidéo-protection, se compose d'un réseau de caméras fixes disposées dans un espace public ou privé afin de le surveiller à distance. Les opérateurs, engagés sur les systèmes classiques, ont pour mission de contrôler les flux des vidéos et de reconnaître les objets apparaissant et leurs actions. Afin d'alléger la charge de travail des opérateurs de contrôle, les algorithmes de vision cherchent à analyser et comprendre de façon automatique le contenu des images transmises par le réseau de caméras. Une de ces tâches est la ré-identification. Son principe s'établit sur l'enregistrement avec un identifiant unique de la signature de la personne, calculée depuis une image ou une séquence vidéo, dans une base ou galerie de personnes. La ré-identification intervient ultérieurement (ou simultanément, dans le cas de caméras partageant un recouvrement des champs de vue) en retrouvant l'identifiant correct par mise en correspondance avec la signature en train d'être acquise parmi celles enregistrées dans la galerie de personnes.

En parallèle, ces 10 dernières années ont été marquées par la démocratisation des plate-formes mobiles et notamment des smartphones et des drones. D'autres appareils commencent également à émerger, telles que les lunettes de réalité augmentée ou les robots domestiques. Par rapport à des caméras fixes, ces plate-formes mobiles possèdent des contraintes supplémentaires, comme une puissance de calcul ou une résolution limitées. Cependant, la transposition de la ré-identification à ces dispositifs permet d'ouvrir de nouveaux horizons en terme d'applications. On constate notamment l'émergence de drones de vidéo-surveillance, qui tirent parti, dans ce contexte applicatif, des avantages procurés par leur mobilité. En plus d'être capables de couvrir une plus grande zone de surveillance, ces plate-formes peuvent adapter leur point de vue et leur direction sans être contraintes par un emplacement statique.

1. www.cnil.fr/sites/default/files/typo/document/CNIL-DP_Video.pdf

1.2 Objectifs de la thèse

Cette thèse propose un *framework* de ré-identification basée sur la monovision et destiné à des applications embarquées. Actuellement, la majorité des systèmes de ré-identification tirent parti de la position statique des caméras pour modéliser l'environnement et segmenter les objets dans l'image grâce à des techniques de soustraction de fond. Puisque nous cherchons à obtenir un fonctionnement sur plate-forme mobile, les algorithmes utilisés doivent respecter une certaine indépendance vis-à-vis du décor.

Par ailleurs, les limitations matérielles des plate-formes embarquées nécessitent de concevoir des méthodes peu coûteuses en temps de calcul, ne permettant de traiter que des images de faibles résolutions. [Layne et al., 2014] ont identifié d'autres problématiques qui découlent d'un fonctionnement sur des systèmes comme les drones. Il peut s'agir notamment d'une dégradation de la qualité de l'image avec le flou induit par le mouvement de la caméra ou encore une plus grande diversité de points de vue d'acquisition. En nous plaçant dans un contexte applicatif sur smartphone, nous délaissons les deux dernières problématiques évoquées en admettant que l'utilisateur manie l'appareil d'un point de vue raisonnable avec une vitesse de déplacement faible.

Cette thèse s'inscrit dans le cadre du projet EMMA (*Embedded software for Mass Market connected Applications*) de type BGLE (Briques Génériques pour les Logiciels Embarqués). Ce projet a pour objectif de développer et commercialiser une offre logicielle sous forme de briques R&D embarquées afin d'être intégrées par des développeurs d'applications. De par la nature du projet, une attention particulière est portée sur les problématiques d'intégration et de performances de temps de calcul. L'ensemble du *framework* de ré-identification a été porté et fonctionne sur smartphone.

1.3 Contributions

Pour répondre à ces problématiques, nous proposons d'introduire dans la chaîne de traitement de ré-identification, habituellement composée par la segmentation de l'objet et le calcul de sa signature visuel, une étape intermédiaire d'alignement du Modèle à Distribution de Points au corps humain. Ce type de représentation structurelle est principalement utilisé dans le cadre du visage et a très peu été expérimenté sur le corps entier. Il s'agit d'un modèle de forme constitué par des points clés et alignés sur l'image par des méthodes statistiques. Les systèmes de reconnaissance faciale [Taigman et al., 2014] qui s'appuient sur ce support donnent de très bons résultats et nous ont encouragés à explorer sa transposition au corps humain entier. Il s'agit d'un défi important car, par rapport au visage, le corps humain est un objet articulé et présente beaucoup plus de complexité d'apparence dûe, en partie, aux vêtements. Une fois aligné, ce modèle améliore la robustesse du processus de ré-identification vis-à-vis de la pose adoptée par la personne.

Nous proposons un *framework* construit en trois modules successifs (illustré Figure 1.1). La première de ces étapes est la détection de personnes, qui a pour but de localiser spatialement dans l'image la personne. La seconde étape, qui opère au niveau de la région détectée, est l'alignement du Modèle à Distribution de Points sur la personne. Enfin, la dernière étape est la ré-identification et s'appuie sur les coordonnées des amers fournis par l'alignement.

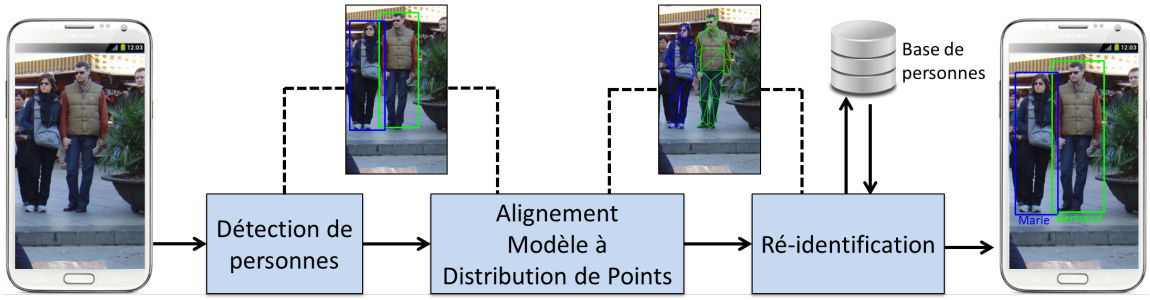


FIGURE 1.1 – *Framework* proposé dans cette thèse.

1.4 Organisation du manuscrit

Ce manuscrit reprend la structure du *framework* et des différents modules que nous avons développés.

Le chapitre 2 décrit la détection de personnes. Il se concentre sur les techniques de fenêtre glissante en donnant une vue d'ensemble des méthodes existantes. Puis, il décrit une approche basée sur les *Channel Features*, pour laquelle nous proposons des améliorations, entre autres, au niveau des approximations des caractéristiques visuelles. Une évaluation des améliorations que nous proposons est ensuite présentée.

Le chapitre 3 s'articule sur la majeure partie du travail accompli dans cette thèse, l'alignement d'un Modèle à Distribution de Points sur le corps humain. Il est découpé en trois parties. Dans un premier temps, nous justifions notre choix sur ce type de représentation et présentons un état de l'art des différentes techniques d'alignement. Puis, nous proposons une première approche basée sur une représentation de forme de type paramétrique, couplée avec un modèle de *boosting* pour gérer l'apparence. Nous évaluons ensuite ce système grâce à une nouvelle base d'apprentissage et de test que nous introduisons. Enfin, nous présentons un second système d'alignement basé sur une cascade de régressions de forme.

Le chapitre 4 concerne le module de ré-identification. Nous illustrons l'apport de l'utilisation de cette représentation dans le cadre d'une ré-identification basée sur l'apparence intégrant les informations liées aux couleurs. Puis nous donnons un aperçu d'une expérimentation avec un descripteur de forme.

Nous consacrons le chapitre 5 aux conclusions et aux perspectives de travail de cette thèse.

Chapitre 2

Détection de Personnes

Sommaire

2.1 Introduction **5**
2.2 Techniques générales **6**
2.3 Etat de l'art - Fenêtre Glissante **7**
2.3.1 Principe général 7
2.3.2 Approches existantes 9
2.4 Approche retenue **14**
2.4.1 Caractéristiques 15
2.4.1.1 Décomposition de l'image en channels 15
2.4.1.2 Images intégrales 17
2.4.1.3 Agrégation de caractéristiques 18
2.4.1.4 Approximation multi-échelle 18
2.4.2 Classification 20
2.4.2.1 AdaBoost 20
2.4.2.2 SoftCascade 23
2.5 Résultats et validations **24**
2.5.1 Dataset utilisée 24
2.5.2 Évaluations 25
2.5.3 Portage sur smartphone 30

2.1 Introduction

La détection de personnes ou détection de piétons est une tâche préliminaire à accomplir dans le cadre d'une ré-identification. Son objectif est de localiser le ou les piétons sur une image. Elle représente un vaste sujet de recherche englobant de nombreux domaines d'applications. Elle peut servir à comptabiliser l'affluence et la concentration des flux piétons. Dans le secteur de l'automobile, la détection de piétons est utilisée dans des systèmes automatisés embarqués sur les voitures tels que les ADAS (*Advanced Driver Assistance System*). Ces systèmes ont comme rôle de prévenir d'une collision avec un piéton en déclenchant une alarme pour le conducteur ou en actionnant un freinage d'urgence. La détection de personnes peut aussi être utilisée dans l'indexation d'images ou de vidéos sur internet grâce à la recherche de ressources par contenu à la place des mots-clés. Dans le contexte de la vidéo-surveillance, elle peut assurer la protection de zones à accès restreint. Combinée avec des systèmes de reconnaissance, elle intervient dans les applications de ré-identification de personnes dont le but est de retrouver l'emplacement d'une personne sur un réseau de caméras. Sur cette problématique, elle permet de focaliser la construction d'une signature visuelle sur l'objet d'intérêt qu'est la personne. Le type de positionnement spatial renvoyé dépend directement de l'algorithme de détection. Le plus souvent, il s'agit d'une boîte englobant la personne. En se reposant sur ce positionnement, des pré-traitements peuvent être effectués pour générer la signature visuelle de la personne en vue d'une ré-identification.

Ce chapitre est structuré de la façon suivante. Dans un premier temps, nous évoquons les méthodes et technologies générales utilisées pour la détection de piétons.

La seconde partie porte sur l'état de l'art des techniques à fenêtre glissante.

En dernière partie, nous détaillons l'approche retenue et les améliorations que nous apportons en nous appuyant sur une série d'évaluations.

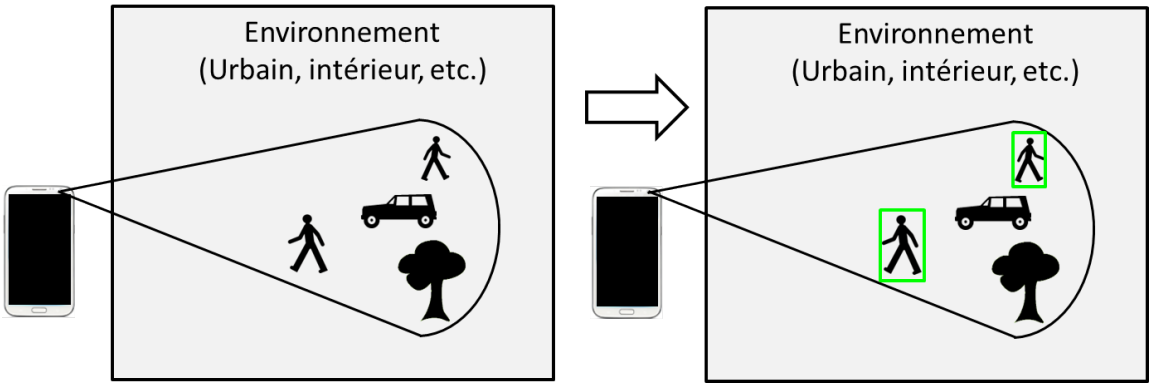


FIGURE 2.1 – Illustration schématique du processus de détection de personnes.

2.2 Techniques générales

La détection de personnes est un domaine classique de la vision par ordinateur qui constitue toujours un véritable challenge. Elle présente de nombreuses difficultés inhérentes à la cible qu'est la personne, mais aussi au décor qui l'entoure. On peut identifier comme principales difficultés :

- Les nombreuses poses qu'une personne peut adopter
- La grande variabilité d'apparence dû au port des vêtements
- Les occultations provoquées par la foule ou par des parties du décor
- Les changements d'illumination et d'éclairage
- Une grande palette d'environnements et de décors, en intérieur ou en extérieur

Pour répondre à cela, la détection de personnes s'appuie sur des technologies diverses et des algorithmes adaptés au cas d'utilisation.

Infrarouge et stéréo-vision

Ces deux premiers systèmes sont basés sur des caméras spécifiques utilisant les images infrarouges et l'information 3D de la scène grâce à la stéréo-vision. A l'heure actuelle, ces types de caméras sont encore peu embarquées dans des appareils tels que les smartphones. C'est pourquoi nous ne tournons pas vers ce type de technologie.

Modélisation d'environnement

Les algorithmes de cette catégorie cherchent à détecter des cibles en mouvement en s'appuyant sur le principe de soustraction de fond. Dans les algorithmes traditionnels, l'environnement, qui ne doit contenir aucun objet à détecter, est dans un premier temps modélisé. Cette modélisation se fait grâce à un apprentissage et doit être robuste aux changements dynamiques de l'image tels que ceux d'illumination ou encore des mouvements répétitifs comme le feuillage des arbres ou les ombres. La seconde étape consiste à détecter ce qui se trouve au premier plan. En soustrayant le modèle d'environnement à l'image courante, les objets mobiles tels que les personnes peuvent être détectés.

Des travaux comme ceux de [Sheikh et al., 2009] ont exploré une soustraction de fond avec des caméras mobiles en se basant sur les déplacements des pixels. Néanmoins, les résultats se dégradent

à partir du moment où la caméra se déplace rapidement [Sobral and Vacavant, 2014]. Ainsi, les algorithmes par soustraction de fond sont en général réservés à des scénarios de vidéo-surveillance sur caméras fixes et sont difficilement adaptables pour le cas d'utilisation présenté dans ce manuscrit.

Fenêtre glissante

La fenêtre glissante (en anglais *sliding window*) est une technique classique utilisée dans les tâches de détection. Elle ne nécessite pour fonctionner que des images statiques. Elle peut aussi bien être déployée dans des applications de vidéo-surveillance que dans des applications embarquées. De plus, elle démontre de meilleurs résultats sur des images de moyenne à faible résolution [Dollár et al., 2012] que d'autres méthodes telles que les points d'intérêts. Cela garantit une performance accrue en terme de temps de calcul sur smartphone, étant donné que travailler sur des images plus petites nécessite moins d'opérations. Tous ces avantages en font une technique privilégiée dans le cadre de cette thèse. Par conséquent, nous en présentons un état de l'art détaillé dans la prochaine section.

2.3 Etat de l'art - Fenêtre Glissante

2.3.1 Principe général

Le principe général de la fenêtre glissante repose sur le coulissement d'une fenêtre de taille fixe sur toute l'image. Pour chaque position du coulissement, on évalue la présence de l'objet à détecter grâce à un classifieur entraîné au préalable sur une base de données et s'appuyant sur des descripteurs visuels. La Figure 2.2a illustre ce fonctionnement.

Le nombre d'évaluations à opérer est de : $N = \lfloor \frac{L-l}{\Delta x} + 1 \rfloor * \lfloor \frac{H-h}{\Delta y} + 1 \rfloor$. La taille de la fenêtre étant fixée à l'apprentissage, N dépend principalement des paramètres de saut ainsi que de la taille de l'image. Ainsi, il conviendra d'ajuster Δx et Δy pour trouver le meilleur compromis entre temps de calcul et performance de détections. De plus, travailler sur des images de grandes résolutions permettra de détecter des piétons de plus petites échelles mais augmentera aussi directement la charge de calcul.

A ce stade, la fenêtre glissante ne permet de détecter que des piétons possédant une taille fixe dans l'image. Pour arriver à une détection multi-échelle, la solution est de procéder à un redimensionnement de l'image (*resampling*) et de ré-effectuer une fenêtre glissante dessus. Ce processus est illustré sur la Figure 2.2b. Les instances positives (c'est-à-dire contenant une personne selon le classifieur) à cette nouvelle échelle sont ensuite transposées sur l'échelle d'origine. On nomme cet ensemble d'images redimensionnées, pyramide multi-échelle. Elle se construit en général sur une réduction de la résolution (*downsampling*) mais potentiellement aussi sur une augmentation (*upsampling*). Cette dernière opération de *upsampling* extrapole les pixels sans créer de nouvelle structure dans l'image [Dollar et al., 2014]. Si l'on ne considère que l'opération de *downsampling*, la plus petite taille de piétons prise en compte dans l'image originale est de $\Delta x \times \Delta y$ et la plus grande de $S(\Delta x \times \Delta y)$, S étant le facteur d'échelle permettant d'obtenir la plus petite image redimensionnée.

Les instances positives sont ensuite regroupées par un algorithme de suppression des non-maxima (*Non-Maxima Suppression* ou NMS) (Figure 2.2c). Généralement, cet algorithme NMS sélectionne les plus fortes réponses de détection en terme de confiance sur un voisinage défini. Cela permet d'obtenir les boîtes englobantes finales de détection dans l'image.

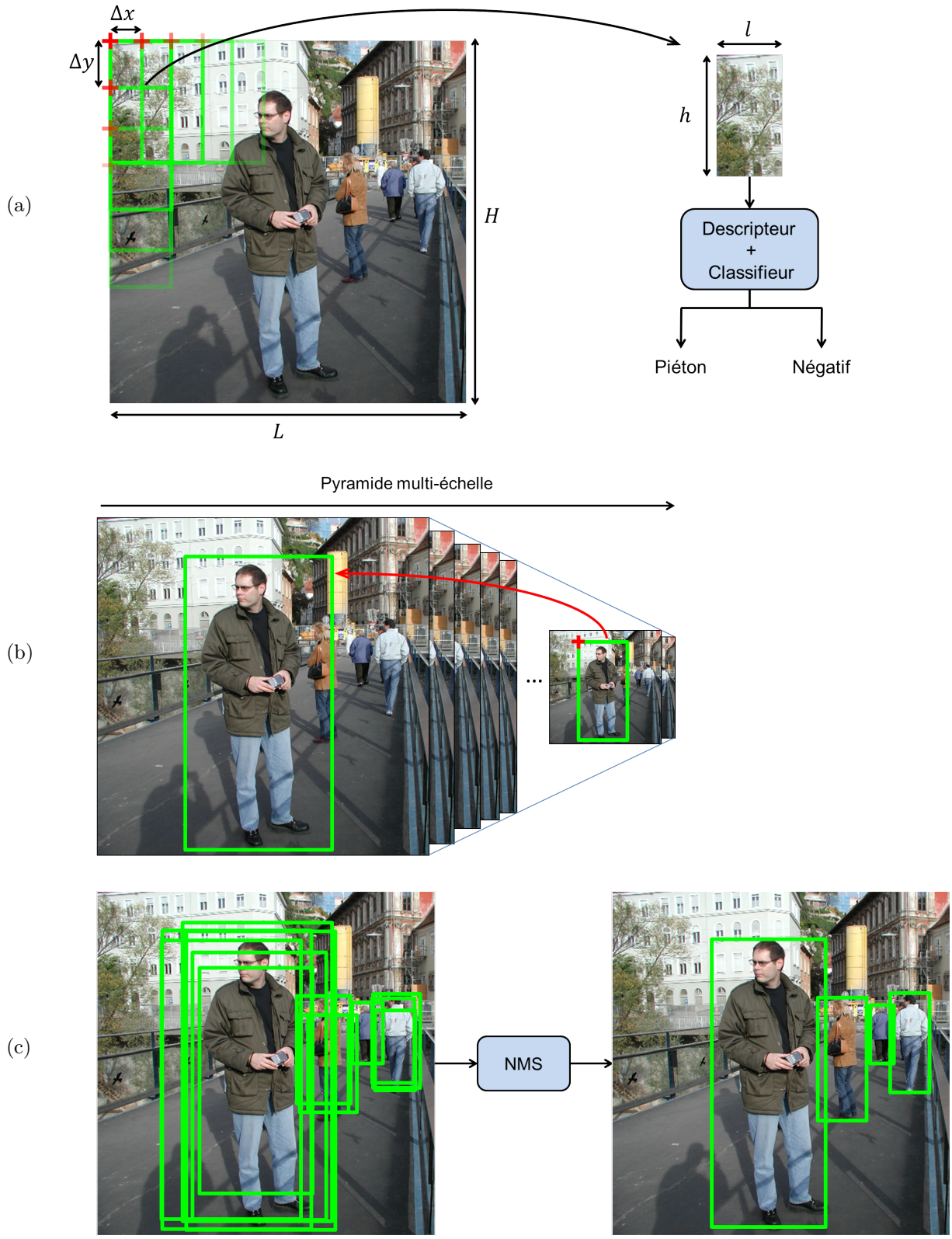


FIGURE 2.2 – (a) Fonctionnement classique de la fenêtre glissante. Les croix rouges indiquent le bord gauche haut de la fenêtre. Chaque vignette est évaluée par l'algorithme de détection pour déterminer s'il s'agit ou non d'un piéton. (b) Le processus de fenêtre glissante est étendu à une pyramide d'images et l'instance de détection est retransformée sur l'échelle originale. (c) L'algorithme NMS permet de regrouper plusieurs instances de détection se chevauchant en une seule.

2.3.2 Approches existantes

[Papageorgiou and Poggio, 2000] ont introduit un des premiers détecteurs de piétons basé sur une technique de fenêtre glissante. Au lieu d'utiliser directement les pixels, une représentation de la personne est calculée grâce aux ondelettes de Haar. Cet outil mathématique permet d'encoder la structure visuelle d'un objet en un dictionnaire multi-échelle de caractéristiques. L'ensemble de ces caractéristiques est utilisé ensuite avec un classifieur de type SVM (*Support Vector Machine* ou Séparateurs à Vaste Marge).

Par la suite, [Viola and Jones, 2001] reprennent ce type de représentation pour l'appliquer à la détection de visages et l'étendent à celle de piétons [Viola et al., 2003]. Ces travaux introduisent un moyen de calcul rapide de ces caractéristiques (nommées caractéristiques pseudo-Haar ou *Haar-Like features*, Figure 2.3) grâce aux images intégrales. Nous décrivons le fonctionnement de ces dernières dans la Section 2.4.1.2. Une autre innovation se trouve dans le processus d'apprentissage par l'utilisation d'une machine de type boosting : le Adaptative Boosting ou *Adaboost* [Freund and Schapire, 1997a]. Ce dernier permet de sélectionner les caractéristiques locales les plus pertinentes. En outre, plusieurs *Adaboost* de plus petites tailles ont été mis en cascade pour améliorer le temps de calcul. Cela revient à enchaîner une succession de filtres et permet d'éviter une évaluation complète de tous les échantillons par le système. Ces idées servent toujours aux algorithmes de détection modernes. Récemment, [Zhang et al., 2014] montrent que l'utilisation de caractéristiques pseudo-Haar spécialement conçues pour les personnes fournit de très bons résultats de détections.

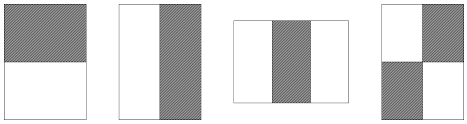


FIGURE 2.3 – Exemples de caractéristiques pseudo-Haar [Viola and Jones, 2001]. La somme de l'intensité des pixels contenus dans les zones blanches est soustraite de celle des zones sombres. L'utilisation des images intégrales améliore grandement les temps de calcul de ces sommes.

En 2005, la détection de personnes a connu une grande avancée grâce aux travaux de [Dalal and Triggs, 2005]. En effet, ils ont proposé un descripteur performant pour pouvoir détecter des piétons : les histogrammes de gradients orientés (en anglais *Histogram of Oriented Gradients* ou *HOG*), illustrés Figure 2.4. Ce descripteur permet de caractériser efficacement les contours locaux d'une personne. L'idée de gradient orienté avait déjà été abordée dans les caractéristiques locales SIFT (*Scale-invariant feature transform*) [Lowe, 2004]. La nouveauté de ces travaux est de considérer la répartition de ces gradients sur toute l'imagette grâce à une grille dense (division de l'image en blocs et cellules). Pour chacune des cellules de cette grille, un histogramme de gradients orientés est construit et concaténé au descripteur global. Cela permet de mesurer la distribution de direction des contours dans l'image et de caractériser notamment les zones de saillance. Depuis, le descripteur HOG est devenu une caractéristique classique pour la détection de piétons et a été décliné dans de nombreux travaux.

Pour accélérer le processus de calcul du descripteur HOG, [Zhu et al., 2006] utilisent les propriétés des images intégrales en pré-générant des *maps* pour chaque orientation des gradients. De plus, la mise en cascade de classifieurs leur permet d'améliorer aussi bien les performances de détections que les temps de calcul.

Certains détecteurs ont cherché à combiner d'autres informations en complémentarité aux gradients orientés. [Wojek and Schiele, 2008] proposent notamment d'utiliser une combinaison de caractéristiques pseudo-Haar ainsi que des descripteurs de formes comme le *Shape Context* [Belongie et al., 2002] ou les *Shapelets* [Sabzmeydani and Mori, 2007] (Figure 2.5a). Le *Shape Context* est un descripteur sur lequel nous reviendrons plus en détail en Section 4.4.1. Dans le cas de leurs travaux,

FIGURE 2.4 – Visualisation du descripteur HOG appliqué à des personnes [Dalal and Triggs, 2005]. Les histogrammes de gradients orientés de chaque cellule sont représentées par des petites étoiles vertes. On peut notamment remarquer l'orientation verticale dominante au niveau des jambes et des bras des personnes.

il s'appuie sur la distribution intrinsèque de points obtenus par un détecteur de Canny (ou détecteur de contours). Les *Shapelets* sont des caractéristiques *mid-level* s'appuyant sur les gradients orientés et qui sont sélectionnées et pondérées par l'algorithme *AdaBoost* dans des sous-fenêtres. La combinaison de toutes ces caractéristiques permet de dépasser en performance leur utilisation isolée. [Mu et al., 2008] proposent d'employer les caractéristiques *Local Binary Pattern* (LBP) [Ojala et al., 1996] (Figure 2.5b) en les adaptant pour une tâche de détection de personnes. Ces caractéristiques encodent de façon binaire la structure d'une zone locale dans l'image par utilisation de seuillages des pixels par rapport à un pixel central. Elles permettent notamment de décrire efficacement les textures des objets et sont robustes aux changements monotones en niveau de gris (causés par exemple par des changements d'illumination). [Wang et al., 2009] combinent les LBP avec le HOG et proposent une façon de gérer les occultations partielles. En se basant sur le descripteur HOG, ils construisent une carte de probabilités de zones potentiellement occultées et appliquent des détecteurs par parties sur les zones non occultées. [Wu and Nevatia, 2008] combinent comme caractéristiques le descripteur HOG, un descripteur de covariance et les *Edgelets*. Ces derniers, introduits par [Wu and Nevatia, 2005] et illustrés Figure 2.5c, se basent sur la silhouette d'une personne et représentent de petites «bordures» comme des lignes ou des courbes. Cette combinaison est ordonnée en faisant un compromis entre charge de calcul requise et performances de classification.

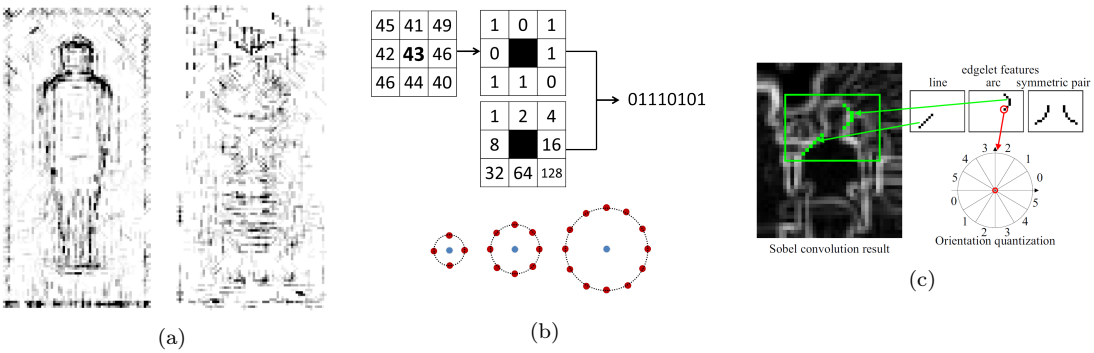


FIGURE 2.5 – Illustrations de caractéristiques utilisées pour la détection de personnes. (a) *Shapelets* [Sabzmeydani and Mori, 2007] (b) *Local Binary Patterns* [Ojala et al., 1996] [Mu et al., 2008] (c) *Edgelets* [Wu and Nevatia, 2005]

En plus de ces caractéristiques basées sur la forme des personnes, des travaux récents ont commencé à utiliser la couleur dans une tâche de détection de personnes. Son utilisation directe en tant que telle s'avère limitée car le spectre de couleurs associées à une personne est vaste (notamment dû aux vêtements) et peut se recouper avec celui du décor. [Ott and Everingham, 2009] proposent d'effectuer une segmentation «légère» entre le décor et la personne dans le but de faciliter la tâche du détecteur avec le descripteur HOG. [Walk et al., 2010] considèrent que la structure des personnes peut être retrouvée grâce aux couleurs, notamment pour la tête et les bras (couleur de peau), mais aussi pour la plupart des vêtements (couleur uniforme). Pour ce faire, ils encodent les similarités de couleurs dans différentes sous-régions de l'image grâce aux histogrammes *HSV*.

[Dollar et al., 2009] s'inspirent des caractéristiques pseudo-Haar pour créer des caractéristiques nommées *Integral Channel Features*, qui sont des sommes de pixels d'une zone rectangulaire sur des *channels*. Ces *channels* sont des images calculées par une transformation donnée depuis l'image originale. Leurs expérimentations ont montré qu'ils obtiennent les meilleurs résultats pour les *channels* suivants : 6 consacrés aux gradients orientés, 1 à la magnitude des gradients et 3 canaux de couleurs *CIELUV*. De nombreuses extensions de ces travaux ont été proposées pour améliorer les temps de calcul des caractéristiques par rapport au multi-échelle. [Dollar et al., 2010] proposent une fonction d'approximation en échelle des caractéristiques grâce à une pyramide d'échelle réduite. [Benenson et al., 2012] étendent cette idée en inversant le problème et en le reportant en phase d'apprentissage par l'entraînement de modèles sur des caractéristiques multi-échelle. Une sélection efficace de caractéristiques suffisamment discriminantes est un facteur important pour les performances de détection. [Benenson et al., 2013] explorent exhaustivement l'espace des caractéristiques et montrent que la sélection des meilleures caractéristiques améliore sensiblement les performances de détection. [Dollar et al., 2014] reviennent en détail sur la fonction d'approximation de [Dollár et al., 2012] par une analyse statistique et proposent d'agréger ces caractéristiques directement en pixels.

De plus, dès lors que l'on travaille sur une succession d'images ou une séquence vidéo, une autre caractéristique pouvant être exploitée est le mouvement produit par le déplacement des personnes. Ainsi, [Viola et al., 2003] appliquent des caractéristiques pseudo-Haar sur des images successives dans une configuration où la caméra est supposée fixe. Cependant, cette tâche se complique lorsque la caméra est mobile car son déplacement doit être alors pris en compte dans le calcul du mouvement à l'image. Pour parvenir à cela, [Dalal et al., 2006] proposent de modéliser le mouvement du corps humain en se basant sur les différences internes du flot optique (qui est un champ de déplacement visuel entre les images). Ainsi, ils parviennent à compenser le mouvement uniforme du champ et à extraire les mouvements locaux dans l'image. Les caractéristiques qu'ils calculent sont des histogrammes de flot optique orienté. Ces derniers, combinés avec d'autres caractéristiques apportent une légère amélioration sur les performances de détection [Walk et al., 2010]. [Park et al., 2013] s'appuient sur un type de stabilisation du mouvement dans l'image pour extraire des caractéristiques efficaces pour la détection, notamment les mouvements liés aux membres du corps humain (*part-centric motion*). Ainsi, ils stabilisent la séquence en supprimant le mouvement lié à la caméra et au déplacement global de la personne. Pour cela, ils estiment le flot optique par l'approche de Lucas-Kanade [Lucas and Kanade, 1981] de manière « grossière » en élargissant la fenêtre locale (autour d'un pixel) dans laquelle le déplacement est considéré constant d'une image à l'autre. Ces images stabilisées servent de base pour calculer des caractéristiques basées sur le gradient temporel, qui, combinées à un descripteur statique HOG, permettent à ces travaux d'obtenir les meilleurs résultats parmi les méthodes existantes exploitant le mouvement.

Avec une augmentation des moyens matériels et des bases de données, les algorithmes de *Deep Learning* ont resurgi ces dernières années, notamment pour des tâches de classification [Krizhevsky et al., 2012]. [Sermanet et al., 2013] appliquent les *Convolutional Neural Networks* (*ConvNets*) par une étape de pré-apprentissage non-supervisé (données non labellisées) suivie d'un apprentissage supervisé (données labellisées) basé sur les données brutes des pixels. [Ouyang and Wang, 2013a] proposent d'entraîner conjointement les composants d'occultation, de modélisation par parties, d'ex-

(a) (b)

FIGURE 2.6 – Les modèles par parties détectent séparément des sous-parties du corps humain. (a) Modèle par parties rigides utilisé dans [Wu and Nevatia, 2005] avec séparation de la tête, du torse et des jambes. (b) Modèle à parties déformables, [Felzenszwalb et al., 2008], où chaque partie possède une position non prédéfinie dans l'image.

traction de caractéristiques et de classification, en s'appuyant sur le *framework* de *Deep Learning*.

Par ailleurs, pour répondre au problème de l'articulation du corps humain, la détection de personnes peut être abordée par une approche basée par parties (*Part-Based Model*). Ces types de détecteurs s'appuient sur une considération séparée des parties du corps humain grâce à un apprentissage spécifique de chaque zone. Cette détection par parties permet de gérer, dans une certaine mesure, les occultations en évaluant une détection sur les parties seulement visibles sur l'image. Certains systèmes considèrent un modèle avec des parties rigides prédéfinies. C'est le cas pour [Wu and Nevatia, 2005] qui découpent verticalement l'image pour la tête, le torse et les jambes (Figure 2.6a) ou [Wang et al., 2009] sur la partie haute et basse du corps humain. D'autres considèrent des parties déformables, c'est-à-dire ne possédant pas une position fixe dans l'image (*Deformable Part Model* ou *DPM*, illustré Figure 2.6b). [Felzenszwalb et al., 2008, 2010] utilisent cette notion grâce à une version modifiée de *SVM* : *LatSVM* modélisant les positions des parties comme des variables latentes. Un descripteur HOG pyramidale leur permet d'établir, dans un premier temps, grossièrement une position couvrant l'objet et ensuite d'évaluer à plus haute résolution les parties sur cette zone couverte.

Un dernier moyen existant pour améliorer les performances de détection consiste à prendre en compte l'information contextuelle de l'image. Ainsi, dans l'optique de détecter des piétons de très petites tailles en pixels dans l'image ($<$ 30 pixels en hauteur), [Park et al., 2010] font l'hypothèse que les personnes se trouvent sur un plan horizontal et pénalisent les détections potentielles se trouvant loin de ce modèle plan. Dans des scènes de trafic urbain, [Yan et al., 2013] infèrent une relation spatiale entre les personnes et des voitures (qui sont une catégorie d'objets plus facilement détectables) pour valider ou non une détection. Enfin, pour gérer les difficultés engendrées par les foules (occultation ou proximité des modèles), [Ouyang and Wang, 2013b] proposent de combiner la détection d'une seule personne avec un détecteur groupant 2 personnes, entraîné sur une base annotée spécialement.

Du fait de leur apprentissage statistique, un important facteur lié aux détecteurs de personnes repose sur les bases d'apprentissage et de tests. Au fur et à mesure des années, les bases publiées sont devenues de plus en plus complètes en terme de nombre d'échantillons, mais aussi en diversité de scénarios. Une qualité primordiale pour une base d'apprentissage en *machine learning* est sa capacité à être représentative face au maximum de situations possibles. Parmi les bases de piétons les plus notables, nous pouvons citer ETHz, Inria Pedestrian Dataset [Dalal and Triggs, 2005], Caltech-USA. Nous invitons le lecteur à consulter [Dollár et al., 2012] qui décrivent en détail les caractéristiques de chacune de ces bases. Ces derniers travaux cités proposent d'harmoniser les procédures d'évaluation des détecteurs de piétons grâce à un *framework* commun. Cela permet de pouvoir clairement comparer les méthodes existantes en terme de performances. Nous récapitulons dans le Tableau 2.1 les principaux algorithmes de cette section ayant été évalués sur ce *framework*.

Algorithme	Caractéristiques	Classifieurs	Miss-Rate	Base	Notes
[Viola et al., 2003]	Haar	AdaBoost	94.73%	I	
[Sabzmeydani and Mori, 2007]	Shapelet	AdaBoost	91.37%	I	Apprentissage des caractéristiques
[Felzenszwalb et al., 2008]	HOG	LatentSVM	79.78%	P	DPM
[Sermanet et al., 2013]	Pixels bruts	Deep Learning	77.20%	I	ConvNets
[Dalal and Triggs, 2005]	HOG	SVM	68.46%	I	
[Wojek and Schiele, 2008]	HOG+Haar+Shape Context+Shapelet (MultiFtr)	AdaBoost	68.26%	I	
[Wang et al., 2009]	HOG+LBP	SVM	67.77%	I	
[Felzenszwalb et al., 2010]	HOG	LatentSVM	63.26%	I	DPM
[Walk et al., 2010]	MultiFtr+Couleur	SVM	60.89%	T	
[Dollar et al., 2010]	Channel	AdaBoost	57.40%	I	Approximation Multi-échelle
[Dollar et al., 2009]	Channel	AdaBoost	56.34%	I	
[Walk et al., 2010]	MultiFtr+Couleur+ Mouvement	SVM	50.88%	T	
[Benenson et al., 2013]	Channel	AdaBoost	50.17%	I	Recherche exhaustive
[Park et al., 2010]	HOG	SVM	48.45%	C	DPM+Contexte plan
[Dollar et al., 2014]	Channel	AdaBoost	44.22%	C	Approximation Multi-échelle+Agrégation
[Ouyang and Wang, 2013b]	HOG	SVM	43.42%	C	DPM+Contexte 2-personnes
[Ouyang and Wang, 2013a]	Gradient+Couleur	Deep Learning	39.32%	C	Apprentissage joint
[Yan et al., 2013]	HOG	SVM	37.64%	C	DPM+Contexte piéton-voiture
[Park et al., 2013]	HOG+mouvement (gradient temporel)	SVM	37.34%	C	Stabilisation du mouvement
[Benenson et al., 2013]	Channel	AdaBoost	34.81%	C	Recherche exhaustive
[Zhang et al., 2014]	Haar optimisé	AdaBoost	34.60%	C	

TABLE 2.1 – Tableau récapitulatif des principaux détecteurs de piétons, triés par performance. Les résultats d'évaluation sont tirés du *framework* [Dollár et al., 2012] sur la dataset *Testing Caltech-USA*. Ils sont indiqués par le *log-average miss rate*, qui est une valeur obtenue par la moyenne du taux de détection raté sur 9 valeurs de faux positifs par image (voir Section 2.5.2). Plus ce taux est bas, meilleur est le détecteur. La colonne 'Base' représente la base utilisée pour l'apprentissage : I = Inria Pedestrian Dataset, C = Caltech-USA, P = Pascal, T = TUD-Motion.

Nous constatons sur cette liste que les détecteurs présentant les meilleurs résultats s'appuient sur des caractéristiques de type HOG, *channel features*, ou Haar optimisé. De plus, le choix d'une base complète d'apprentissage telle que Caltech-USA possède un rôle important sur les performances des détecteurs. Concernant le modèle déformable par parties, on peut voir qu'il présente une meilleure utilisation dans le cadre du *Deep Learning*. Cependant, à part pour la gestion des occultations, l'apport de ces modèles par rapport à des systèmes basés sur le corps entier n'est pas évident pour une tâche de détection de personnes [Benenson et al., 2014]. L'inférence avec le contexte apporte un léger gain de performances. [Ouyang and Wang, 2013b] obtiennent une amélioration relative de 5% par rapport à leur modèle sans information contextuelle, tandis que [Yan et al., 2013] n'obtiennent que 3%.

Un facteur non abordé en détail jusqu'ici est la vitesse de temps de calcul. Il s'agit d'un point important dans le cadre de nos travaux du fait de la puissance limitée à disposition pour des applications embarquées. Pour cette raison, nous n'envisageons pas d'utiliser des techniques basées sur du *Deep Learning* car leurs implémentations sont, pour la plupart, portées sur GPU.

Actuellement, les détecteurs les plus rapides algorithmiquement sont ceux se basant sur les caractéristiques de type *channel* [Dollar et al., 2014]. Cela vient principalement du fait que la majorité du calcul requis à la création des caractéristiques provient de la génération de l'image en *channels*, opération qui ne se fait qu'en préalable, une seule fois pour une échelle donnée (ou moins pour les travaux faisant de l'approximation multi-échelle [Dollar et al., 2010; Benenson et al., 2012; Ferreira Da Costa and Dai, 2016]). La lecture des caractéristiques est par la suite très rapide grâce, notamment, aux images intégrales. Le Tableau 2.2 reprend ces détecteurs classés en fonction de leurs temps de calcul.

Algorithme	Miss-Rate	FPS	Méthode
[Dalal and Triggs, 2005]	68.46%	0.2	HOG
[Wojek and Schiele, 2008]	68.26%	0.7	MultiFtr
[Felzenszwalb et al., 2010]	63.26%	0.6	DPM+HOG
[Dollar et al., 2009]	56.34%	1.2	Channel
[Dollar et al., 2010]	57.40%	6.5	Channel+Approximation
[Dollar et al., 2014]	51.00%	16.4	Channel+Approximation
[Dollar et al., 2014]	44.22%	31.9	Channel+Approximation+Agrégation

TABLE 2.2 – Temps de calcul en FPS (*frames per second* ou images par seconde) des détecteurs de piétons évalués par [Dollar et al., 2014]. Ces performances ont été obtenues pour des images 640x480 sur 1 CPU.

2.4 Approche retenue

Parmi les moyens de détections précédemment décrits, nous avons évalué et implémenté un des détecteurs utilisant les *Channel Features*. Son intérêt est double : il fournit des résultats corrects de détection et de temps de calcul, et il est possible de réutiliser la génération des *channels* dans le calcul des caractéristiques pour le module d'alignement. De plus, cet algorithme ne se base que sur des images statiques et ne nécessite, par conséquent, pas de séquence vidéo. Ceci laisse la possibilité d'appliquer le *framework* de ré-identification sur des photos ou des images extraites d'Internet. Nous décrivons ce détecteur en détail dans cette section.

2.4.1 Caractéristiques

L'état de l'art des détecteurs de piétons montre que l'utilisation d'une combinaison de caractéristiques permet de décrire efficacement les personnes (gradients, couleurs, mouvements, texture, etc.). Les caractéristiques de type *channel features* dérivent de ce principe puisqu'elles s'appuient sur un panel de transformés de l'image regroupant des informations spécifiques, nommées canaux (*channels*). Ces derniers, associés à une image numérique, désignent une représentation de même taille, en hauteur et largeur, que l'image originale. Cette représentation, aussi appelée image en niveau de gris, est constituée de pixels à une seule valeur. Habituellement, les canaux sont définis pour un contexte lié aux couleurs de l'image et sont associés à des modèles numériques de couleurs tels que le modèle RGB (rouge vert bleu), HSV, CMYK, etc. [Dollar et al., 2009] en proposent une extension, un *channel* est une image en niveau de gris, dont chaque pixel a été obtenu par une opération de transformation à partir d'un patch centré sur le pixel original. Cette opération préserve la taille et la structure globale de l'image originale. La sélection de caractéristiques se fait ensuite localement sur ces *channels*.

2.4.1.1 Décomposition de l'image en channels

Cette décomposition de l'image en *channels* peut s'écrire sous la forme suivante : $C = F(I)$ où C représente un *channel*, F la fonction de transformation et I l'image originale. Une étude a été menée par [Dollar et al., 2009] pour déterminer quelles F étaient utiles à une tâche de détection de personnes. Leur évaluation montre que l'utilisation des gradients orientés, de la magnitude des gradients et des CIELUV fournissent des résultats satisfaisants. Cette combinaison prouve l'efficacité du descripteur HOG dans le cadre d'une détection de personnes mais aussi l'apport utile des couleurs par les *channels* CIELUV.

Lissage de l'image (*smoothing*)

Le lissage de l'image permet principalement d'atténuer le bruit pouvant fausser l'information dans une image. Cependant, il doit être appliqué raisonnablement car il diminue les informations dans les hautes fréquences, pouvant, par exemple, causer la perte de gradients. [Dollar et al., 2009] montrent que les *channels* générés après un pré-lissage apportent des performances de détection légèrement meilleures. Comme pour les travaux originaux, nous choisissons d'appliquer un filtre gaussien avec les facteurs suivants : $\begin{pmatrix}1 & 2 & 1\end{pmatrix}/4$.

Gradients orientés et magnitude de gradients

Notons une image $I = (x, y)$ en niveau de gris qui peut être considérée comme un signal discret à deux dimensions. Le gradient d'une image est une grandeur vectorielle qui décrit la variation d'intensité d'un pixel. Dans le contexte de piétons, les gradients mettent en lumière les contours distinctifs d'une personne, tels que les verticaux associés aux jambes. Ils sont calculés à partir des dérivés discrètes de I en x et en y : $\partial I/\partial x$, $\partial I/\partial y$. Ces deux derniers termes résultent de la convolution d'un filtre linéaire spécifique avec l'image : $\partial I/\partial x = I * h_x$ et $\partial I/\partial y = I * h_y$. On obtient h_y de h_x par une rotation de $\pi/2$. Les noyaux classiques utilisés pour calculer les gradients d'une image sont :

- des masques 2x2 diagonaux $h_x = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$ $h_y = \begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix}$
- des masques 3x3 $h_x = \begin{pmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{pmatrix}$ $h_y = \begin{pmatrix}-1 & -2 & -1\\ 0 & 0 & 0\\ 1 & 2 & 1\end{pmatrix}$ (filtre de Sobel)
- des filtres 1-D centrés $h_x = \begin{pmatrix}-1 & 0 & 1\end{pmatrix}$ $h_y = \begin{pmatrix}-1\\ 0\\ 1\end{pmatrix}$ ou non centrés $h_x = \begin{pmatrix}-1 & 1\end{pmatrix}$ $h_y = \begin{pmatrix}-1\\ 1\end{pmatrix}$

L'efficacité de ces noyaux est variable suivant les objets visuels considérés. Pour des personnes,

[Dalal and Triggs, 2005] montrent, par leur évaluation, que l'utilisation du filtre-1D centré fournit les résultats les plus satisfaisants dans le calcul de leur descripteur HOG. L'étape suivante consiste à calculer la magnitude du gradient à partir des dérivés partielles $C_M(x,y) = \sqrt{\frac{\partial I(x,y)}{\partial x}^2 + \frac{\partial I(x,y)}{\partial y}^2}$. Les valeurs de magnitude de gradients peuvent varier fortement, notamment à cause des variations locales d'illumination. Pour assurer la robustesse à ces changements, on peut procéder à une opération de normalisation du gradient. Dans le cas présent, une normalisation L1 est privilégiée du fait de sa simplicité de calcul. La magnitude de chaque pixel est divisée par la moyenne des magnitudes d'un bloc centré ($\overline{M}(x,y)$) sur le-dit pixel : $C'_M(x,y) = M(x,y)/(\overline{M}(x,y)+\varepsilon)$. Il est nécessaire, dans ce cas, de considérer la moyenne et non la somme des magnitudes pour ne pas dépendre directement de la taille de la boîte. En effet, pour normaliser les pixels proches des bords de l'image, le bloc de normalisation doit être tronqué.

Une fois le *channel* de magnitude normalisé C'_M obtenu, on peut calculer les n *channels* liés aux orientations des gradients C_{θ_i} avec θ_i désignant un intervalle d'orientation. Cette opération passe par la quantification de l'orientation signée des gradients :

$$\frac{(i-1)2\pi}{n} \leq \varphi(x,y) = \text{atan2}\left(\frac{\partial I(x,y)}{\partial x}, \frac{\partial I(x,y)}{\partial y}\right) \leq \frac{i2\pi}{n} \qquad i = 1, ..., n \tag{2.1}$$

ou non signée :

$$\frac{(i-1)\pi}{n} \leq \varphi(x,y) = \text{atan2}\left(\text{signe}\left(\frac{\partial I(x,y)}{\partial y}\right)\frac{\partial I(x,y)}{\partial x}, \left|\frac{\partial I(x,y)}{\partial y}\right|\right) \leq \frac{i\pi}{n} \qquad i = 1, ..., n \tag{2.2}$$

Le signe du gradient permet de déterminer dans quel sens varie l'intensité du pixel (à savoir la notion du clair vers le sombre ou inversement). Or, la grande variabilité d'apparence des piétons ne garantit pas que les pixels d'une personne soient plus clairs ou plus sombres que le décor. C'est pourquoi, les gradients orientés non signés sont en général utilisés pour la détection de piétons [Dalal and Triggs, 2005]. Les *channels* C_{θ_i} se construisent de la façon suivante :

$$\begin{aligned} &C_{\theta_i}(x,y) = C'_M(x,y) \ \text{ si } \varphi(x,y) \text{ répond à la condition de l'équation 2.2} \\ &C_{\theta_i}(x,y) = 0 \ \text{ sinon} \end{aligned} \tag{2.3}$$

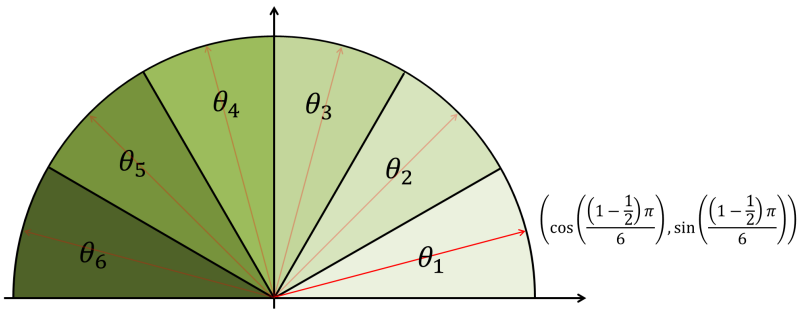


FIGURE 2.7 – Quantification en 6 classes de l'orientation non-signée des gradients. Les vecteurs rouges sont pré-calculés suivant n et servent de supports à une projection orthogonale pour déterminer à quelle classe d'orientation de gradient appartient un pixel.

Sachant que l'opération atan2 est coûteuse en temps de calcul et qu'il n'est pas nécessaire de calculer l'angle exact d'un gradient, une optimisation possible pour retrouver la classe d'orientation est de considérer i tel que : $i = \underset{i}{\text{argmax}} \sum_{j=1}^{n} \left(\text{signe}\left(\frac{\partial I(x,y)}{\partial y}\right)\frac{\partial I(x,y)}{\partial x}, \left|\frac{\partial I(x,y)}{\partial y}\right|\right) . \left(\cos\left(\frac{(j-\frac{1}{2})\pi}{n}\right), \sin\left(\frac{(j-\frac{1}{2})\pi}{n}\right)\right)$. La Figure 2.7 montre un exemple des orientations considérées pour 6 classes dans le cas non-signé.

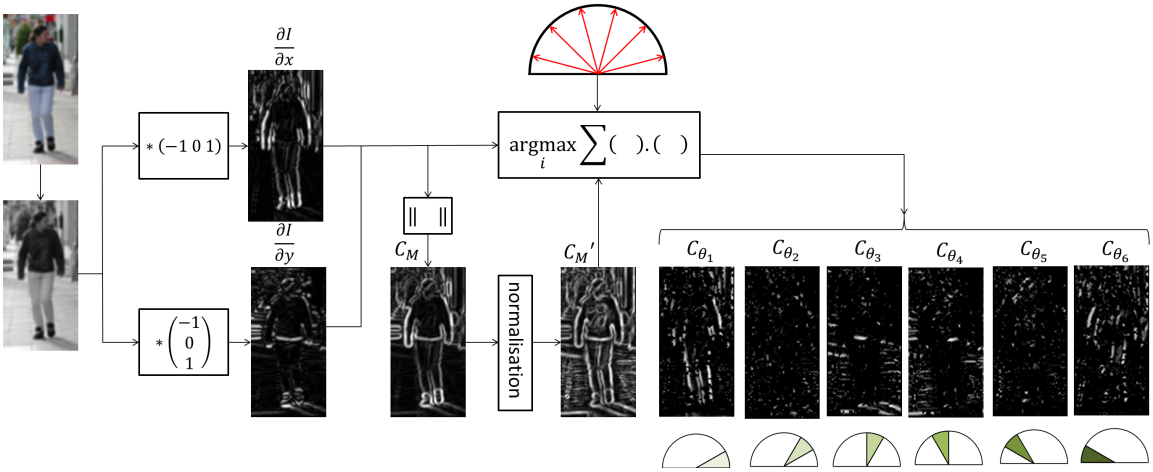


FIGURE 2.8 – Schéma récapitulatif du calcul des *channels* de gradients orientés et de la magnitude de gradient.

CIELUV

Les *channels* de couleur **CIELUV** ou **CIE L*u*v*** sont un espace colorimétrique dans lequel la notion de couleurs, ou chrominance, est portée par les *channels* **u*** et **v*** et la clarté par le *channel* **L***. Nous renvoyons le lecteur vers [Sève, 2009] pour consulter les fonctions de transformation à partir du système de codage RGB. Pour optimiser le calcul de la racine cubique intervenant dans les fonctions de transformation (dont la variable est bornée entre 0 et 1), nous proposons de l'approximer par une table de correspondance sur une série de valeurs pré-enregistrées.

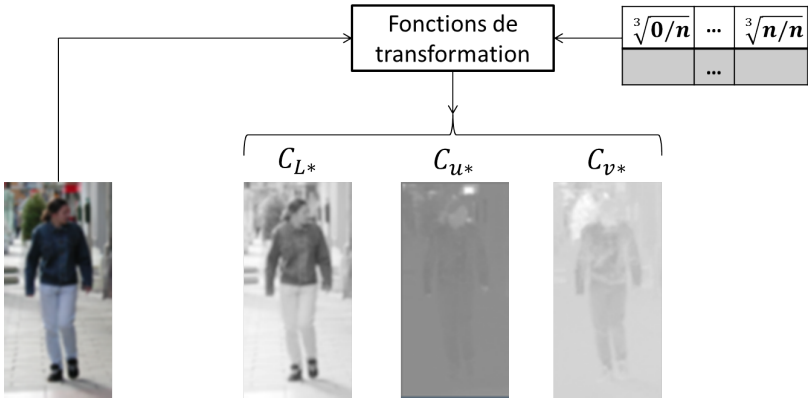


FIGURE 2.9 – Exemple de calcul des *channels* CIELUV, en se servant d'une table de correspondances pour approximer la racine cubique intervenant dans les fonctions de transformation.

2.4.1.2 Images intégrales

Les *channels* exposés précédemment servent de support au calcul des caractéristiques utilisées par le détecteur. Ces dernières sont des sommes de pixels de zones rectangulaires sur un *channel* donné. Pour éviter d'avoir à additionner un à un chaque pixel, l'astuce consiste à recourir aux images intégrales. Cette idée fut avancée dans les travaux de [Viola and Jones, 2001] pour le calcul de caractéristiques pseudo-haar, qui sont en fait, une combinaison des caractéristiques locales blocs.

Une image intégrale est une représentation pour laquelle la valeur à un pixel donné est la somme de tous les pixels situés «au dessus et à gauche», soit :

$$\begin{cases} I_{\sum}(x+1, y+1) = \sum\limits_{\substack{i \leq x \\ j \leq y}} I(i,j) \\ I_{\sum}(0,y) = I_{\sum}(x,0) = 0 \end{cases} \tag{2.4}$$

En se référant à la Figure 2.10, on peut déduire par soustraction de zones la somme de pixels situés dans la zone hachurée :

$$\begin{aligned} &\sum(D) = \sum(A) + \sum(?) + (\sum(B) - \sum(A)) + (\sum(C) - \sum(A)) \\ \Leftrightarrow &\sum(?) = \sum(A) + \sum(D) - \sum(B) - \sum(C) \end{aligned} \tag{2.5}$$

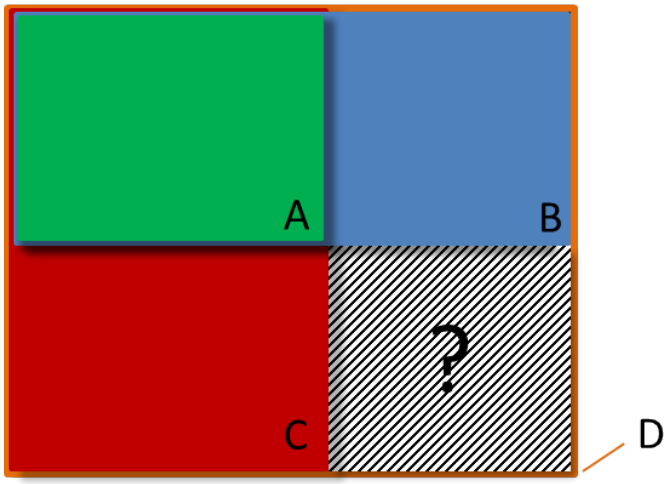


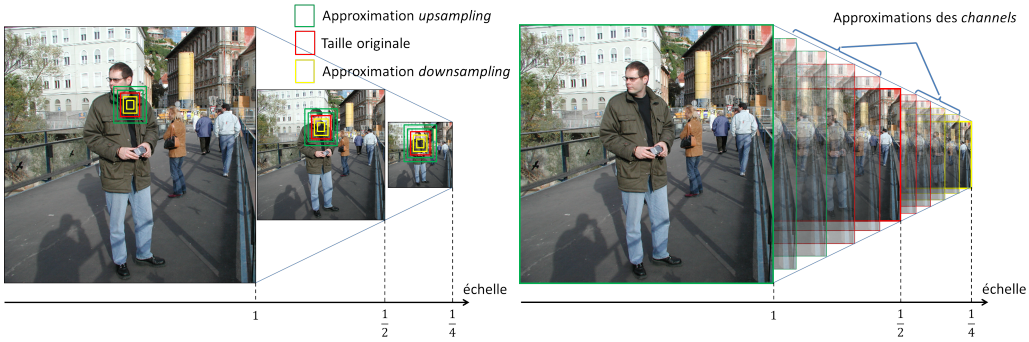
FIGURE 2.10 – Les images intégrales ont comme propriété de pouvoir calculer des sommes de pixels très rapidement. La somme de pixels de la zone hachurée $= A + D - B - C$.

2.4.1.3 Agrégation de caractéristiques

Pour réduire la taille de l'espace des caractéristiques et le temps de calcul lié à leur lecture sur l'image, [Dollar et al., 2014] proposent de générer, pour chaque *channel*, une image formée de l'agrégation contiguë de blocs. L'opération revient à sommer tous les pixels de chaque bloc, usuellement de taille 4x4, constituant l'image. Cela s'apparente à la grille pour établir le descripteur dense HOG, néanmoins sans chevauchement. Ces sommes sont ensuite concaténées et lire la valeur d'une caractéristique se fait de façon directe sans passer par les 4 accès mémoires requis pour les images intégrales. Du fait de cette grille, l'agrégation de caractéristiques impose à l'objet étudié d'avoir une structure relativement fixe dans l'image (comme un piéton, une voiture, etc.).

2.4.1.4 Approximation multi-échelle

Nous avons vu que la détection par fenêtre glissante nécessitait de créer une pyramide d'images pour pouvoir gérer les piétons à différentes échelles sur l'image. Or, la construction de cette pyramide complète nécessite une grande charge de calcul vis-à-vis de la génération des *channels*. C'est



(a) (b)

FIGURE 2.11 – Principe d'approximations multi-échelles avec une pyramide d'images allégée. (a) Approximation multi-échelle sur des caractéristiques locales de type sommes de blocs. (b) Génération de *channels* entre chaque octave d'échelle.

pourquoi, les auteurs de [Dollar et al., 2010, 2014] se sont posés la question suivante : est-il possible d'estimer, à une échelle donnée, la valeur d'une caractéristique sur un *channel* à une autre échelle ?

Pour répondre à cette question, les auteurs ont mené une analyse statistique sur les images. Il ressort de leurs études deux cas distincts. Lorsque l'on souhaite retrouver une caractéristique sur une image agrandie (*upsampling*), la valeur estimée d'une caractéristique est directement proportionnelle au facteur d'agrandissement. En effet, lorsque l'on agrandit une image avec une fonction d'extrapolation, on ne crée pas plus d'information ou de structure par rapport à l'image originale. Cela peut s'écrire :

$$f_s(I) = \frac{s}{s_0} f_{s_0}(I) \tag{2.6}$$

où s_0 représente l'échelle originale, s l'échelle de redimensionnement et f une caractéristique sous la forme d'une somme de pixels sur l'image. En revanche, dans le cas où l'on rétrécit une image (*downsampling*), on perd de l'information structurelle, notamment dans les hautes fréquences, aboutissant à une détérioration des gradients. [Dollár et al., 2012] montrent empiriquement que les caractéristiques peuvent être approximées en multi-échelle par une loi de puissance :

$$f_s \approx \left(\frac{s}{s_0}\right)^{-\lambda_F} f_{s_0} \tag{2.7}$$

où λ correspond à un paramètre d'approximation propre à un type de transformation F. Néanmoins, l'erreur de cette approximation augmente avec l'éloignement des échelles. C'est pourquoi, il est proposé de constituer une pyramide d'images allégées en générant des *channels* «repères» $C_{\hat{s}}$ à chaque octave d'échelle (échelle à un facteur de 2 par rapport à l'échelle originale : $\hat{s} = \{1, \frac{1}{2}, \frac{1}{4}, ...\}$), puis, d'approximer les caractéristiques des échelles intermédiaires s entre 2 octaves par rapport au *channel* «repère» le plus proche en échelle (Figure 2.11a).

Au lieu de considérer f à une échelle donnée et d'y appliquer le facteur d'approximation, une autre façon de faire, proposée dans [Dollar et al., 2014], est de générer une pyramide de caractéristiques ou *channels* en étendant l'approximation 2.7 aux *channels* :

$$C_s = R(C_{\hat{s}}, s) \left(\frac{s}{\hat{s}}\right)^{-\lambda_F} \tag{2.8}$$

avec R comme fonction de redimensionnement et $\hat{s}$ sélectionnée le plus proche de s (Figure 2.11b). En se basant sur cette équation et la valeur moyenne des pixels dans une image, il est possible

d'évaluer λ_F statistiquement entre deux octaves :

$$\lambda_F = \frac{-\log\left(\dfrac{\frac{1}{(\hat{s}/2)LH}\sum\limits_{x,y}^{(\hat{s}/2)(L,H)} F(R(I_{\hat{s}}, \hat{s}/2))(x,y)}{\frac{1}{\hat{s}LH}\sum\limits_{x,y}^{\hat{s}(L,H)} F(I_{\hat{s}})(x,y)}\right)}{\log\left(\frac{1}{2}\right)} \tag{2.9}$$

Le système original applique le même facteur λ_F pour les *channels* de gradients orientés et celui de la magnitude de gradient. Or, nos expérimentations Figure 2.17 montrent que les comportements des *channels* de gradients orientés diffèrent entre eux lors du redimensionnement. Il en est de même avec celui du *channel* de magnitude de gradients. De plus, comme signalé dans [Dollar et al., 2014], nous constatons que le comportement de *channels* clairsemés (*sparse*), à savoir avec de nombreuses valeurs à 0, tels que les gradients orientés (du fait de la quantification d'orientation) sont plus instables au redimensionnement que des images en *niveau de gris* comme les *channels* **L*u*v***. Par ailleurs, ce comportement peut aussi varier entre différentes octaves, jusqu'à devenir trop imprévisible sur les petites échelles (du fait d'une trop grande perte d'information). C'est pourquoi, nous proposons de définir un facteur $\lambda_{F,\hat{s}}$ pour chaque intervalle d'octave. De plus, nous considérons que le facteur $\lambda_{F,\hat{s},I}$ dépend de l'image et, par conséquent, nous le calculons au fil de l'eau depuis les *channels* références. Même si cela peut s'avérer bruité, cette solution permet, dans une certaine mesure, de s'accommoder aux images atypiques en terme de distribution d'orientation des gradients.

2.4.2 Classification

Les caractéristiques constituent des éléments représentatifs et quantitatifs d'un objet sur lesquels le système va baser ses critères d'évaluations. Dans ce cadre, il s'agit d'une classification binaire, du fait de la présence de deux classes : Piétons et Négatifs. Le classifieur binaire est entraîné en phase d'apprentissage et délivre une prédiction en phase de test ou d'évaluation. L'algorithme d'apprentissage construit un modèle à partir d'une base d'entraînement en s'appuyant sur les caractéristiques visuelles et cherche à minimiser l'erreur de classification. Il s'agit d'un apprentissage dit supervisé car tous les échantillons de la base sont connus et étiquetés.

2.4.2.1 AdaBoost

AdaBoost (*Adaptive Boosting*) est un algorithme de classification binaire introduit par [Freund and Schapire, 1997b], qui repose sur le principe de *boosting*, c'est à dire la combinaison par un vote conjoint et pondéré d'un ensemble,nommé classifieur fort H, de classifieurs faibles h_t. Notons la base d'entraînement D composée de N échantillons : $D = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$. Chaque échantillon possède un vecteur $x_n \in X$, X étant l'espace des caractéristiques, et un label $y_n = \{-1, +1\}$ respectivement négatif (décor) et positif (piéton). Considérons la notation suivante D^-, D^+ pour définir les 2 sous-ensembles contenant respectivement les échantillons négatifs et positifs.

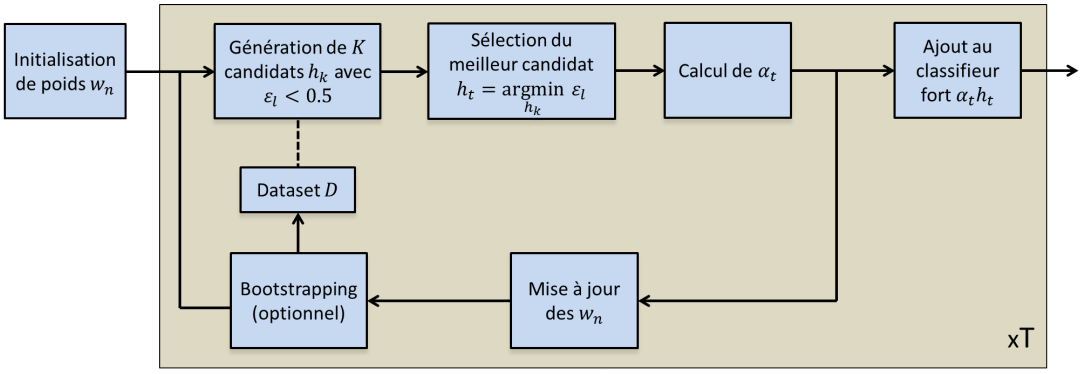


FIGURE 2.12 – Schéma d'apprentissage de l'algorithme *AdaBoost*.

AdaBoost tire son nom de sa faculté à concentrer son apprentissage itératif sur les échantillons considérés comme «difficiles». Pour cela, il attribue à chaque échantillon (x_n, y_n) un poids w_n, qui évolue, à l'itération t, en fonction d'une bonne ou d'une mauvaise classification par h_t. Nous représentons schématiquement l'apprentissage d'*AdaBoost* Figure 2.12 ainsi que le pseudo-code Algorithme 1.

Une des forces d'*AdaBoost* est de sélectionner le meilleur classifieur faible sur chaque itération, et, par cet intermédiaire, les caractéristiques locales les plus pertinentes sur l'image. Nous proposons d'introduire une étape de *bootstrapping* dans l'apprentissage d'*AdaBoost* qui consiste en l'introduction progressive de I échantillons négatifs. Nous proposons, dans notre cas, de n'introduire que des échantillons mal classifiés par les classifieurs faibles retenus jusqu'à l'itération courante. De plus, I doit être choisi judicieusement. Il ne doit être ni trop grand, sinon *AdaBoost* aura des difficultés à «accrocher» cet ajout d'échantillons, ni trop petit, car, dans ce cas, les effets du *bootstrapping* sont minimes. Nous optons pour un ajustement de I à chaque cycle de sorte que le nombre global d'échantillons mal classifiés issus du *bootstrapping* soit fixe. Cela permet à *AdaBoost* de se concentrer sur ces exemples sans être dépassé par l'ajout de nouveaux échantillons.

En phase de test, le classifieur fort est prédit de la façon suivante : $H(x_n) = \sum_{t=1}^{T} \alpha_t h_t(x_n)$ qui représente un score de confiance sur la détection compris entre -1 et 1. On peut établir un seuil de détection β de telle sorte que l'échantillon n est classé positif si :

$$\beta < H(x_n) \tag{2.10}$$

Comme pour [Dollar et al., 2009], nous utilisons des classifieurs faibles sous la forme d'arbres binaires à 2 niveaux possédant à chaque nœud une opération de seuillage sur une caractéristique donnée. L'apprentissage de ces classifieurs lors de l'étape de génération de candidats (fonction EntraînementClassifieurFaible dans l'Algorithme 1) consiste à retrouver le seuil optimal pour une caractéristique donnée par rapport au set d'exemples D pondéré par W. Nous utilisons l'Algorithme 2, pour la détermination du seuil qui est une méthode similaire à celle proposée par [Viola and Jones, 2004]. Au lieu de calculer l'erreur à chaque cas de seuil ($\mathcal{O}(n^2)$, cet algorithme permet de parcourir en une fois la liste préalablement triée des exemples $\mathcal{O}(n \log(n))$. De plus, nous proposons de prendre en compte le meilleur cas possible entre un seuillage inférieur et supérieur.

Algorithme 1: Apprentissage d'AdaBoost

Entrée: $D = \{(x_1, y_1), ..., (x_N, y_N)\}$

Sortie : Classifieur fort $H(x_n) = \sum_{t=1}^{T} \alpha_t h_t(x_n)$

Initialisation : $W = \{w_n, n \in \{1, ..., N\}\} \begin{cases} w_n(y_n = 1) = \frac{1}{2\mathbf{card}(D^+)} \\ w_n(y_n = -1) = \frac{1}{2\mathbf{card}(D^-)} \end{cases}$

pour $t = 1$ **à** T **faire**

- `// Génération des candidats`
- **pour** $k = 1$ **à** K **faire**
  - **répéter**
    - $c_k \leftarrow$ `EntraînementClassifieurFaible(`D, W`)`
    - $\varepsilon_k = \sum_{n=1}^{N} w_n(\frac{1}{2}\,|h_k(x_n) - y_n|)$
  - **jusqu'à** $\varepsilon_k < 0.5$ `// Meilleurs que le hasard`
- `// Sélection du meilleur candidat`
- $h_t = \underset{h_k}{\operatorname{argmin}}\, \varepsilon_k$
- `// Calcul du poids du classifieur`
- $\alpha_t = \frac{1}{2} \log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$
- `// Ajout au classifieur fort`
- $H \leftarrow \alpha_t h_t + H$
- `// Mise à jour des poids des échantillons`
- **pour** $n = 1$ **à** N **faire** $w_n \leftarrow w_n e^{-\alpha y_n h_t(x_n)}$
- **pour** $n = 1$ **à** N **faire**
  - $w_n(y_n = 1) \leftarrow \dfrac{w_n}{2 \sum_{w_n \in W} \{w_n \mid y_n = 1\}}$
  - $w_n(y_n = -1) \leftarrow \dfrac{w_n}{2 \sum_{w_n \in W} \{w_n \mid y_n = -1\}}$
- `// *Bootstrapping* d'exemples négatifs`
- **pour** $i = 1$ **à** I **faire**
  - $(x_i, y_i) \leftarrow$ `CréerNouvelEchantillonNégatif` ()
  - $D \leftarrow D + (x_i, y_i)$
  - $w_i = \dfrac{1}{2(\mathbf{card}(D^-) + I)}$
  - $W \leftarrow W + w_i$
- **pour** $n = 1$ **à** N **faire**
  - $w_n(y_n = -1) \leftarrow \dfrac{w_n \mathbf{card}(D^-)}{(\mathbf{card}(D^-) + I)}$ `// Met à niveau les poids des anciens négatifs`
- $N \leftarrow N + I$

pour $t = 1$ **à** T **faire** $\alpha_t \leftarrow \dfrac{\alpha_t}{\sum \alpha}$

Algorithme 2: Calcul d'un seuil optimal sur une liste pondérée

Entrée: $L = \{(x_1, y_1, w_1), ..., (x_N, y_N, w_N)\}$
Sortie : Seuil optimal et sens du seuillage
Initialisation : $L \rightarrow TrierListeParOrdreCroissant(X)$

`// Calcul de l'erreur associée à chaque seuil`
pour $n = 1$ **à** $N - 1$ **faire**

$$\varepsilon_{n,\leqslant} = \sum_{i=1}^{N}\{w_i \mid y_i = 1\} - \sum_{i=1}^{n}\{w_i \mid y_i = 1\} + \sum_{i=1}^{n}\{w_i \mid y_i = -1\}$$

$$\text{Seuil}_n = (x_n + x_{n+1})/2$$

pour $n = N$ **à** 2 **faire**

$$\varepsilon_{n-1,\geqslant} = \sum_{i=1}^{N}\{w_i \mid y_i = 1\} - \sum_{i=N}^{n}\{w_i \mid y_i = 1\} + \sum_{i=N}^{n}\{w_i \mid y_i = -1\}$$

`// Sélection du meilleur seuil et du sens de l'inégalité`

$$(\text{Seuil}_{\text{opt}}, \text{Sens}) = \underset{\text{Seuil}_n, \text{Sens}}{\text{argmin}} \ (\varepsilon_{n,\leqslant}, \varepsilon_{n,\geqslant})$$

2.4.2.2 SoftCascade

En phase de test, l'utilisation d'*AdaBoost* en tant que telle requiert l'évaluation des T classifieurs faibles sur chaque position de la fenêtre glissante dans l'image. Sachant que ceci peut s'avérer très coûteux en temps de calcul, une solution classique est de recourir aux cascades pour rejeter les échantillons s'apparentant rapidement à la classe des négatifs. Classiquement, plusieurs *AdaBoost* de petites tailles sont mis en cascade et entraînés les uns à la suite des autres comme pour [Viola and Jones, 2001]. Dans notre cas, nous utilisons l'algorithme SoftCascade [Bourdev and Brandt, 2005] dont le principe consiste à réarranger les classifieurs faibles pour les disposer en cascade. Cette étape de réarrangement s'appelle phase de calibration et nécessite une base d'entraînement distincte de celle utilisée pour l'apprentissage *AdaBoost*. Nous donnons une représentation schématique de la calibration du SoftCascade Figure 2.13.

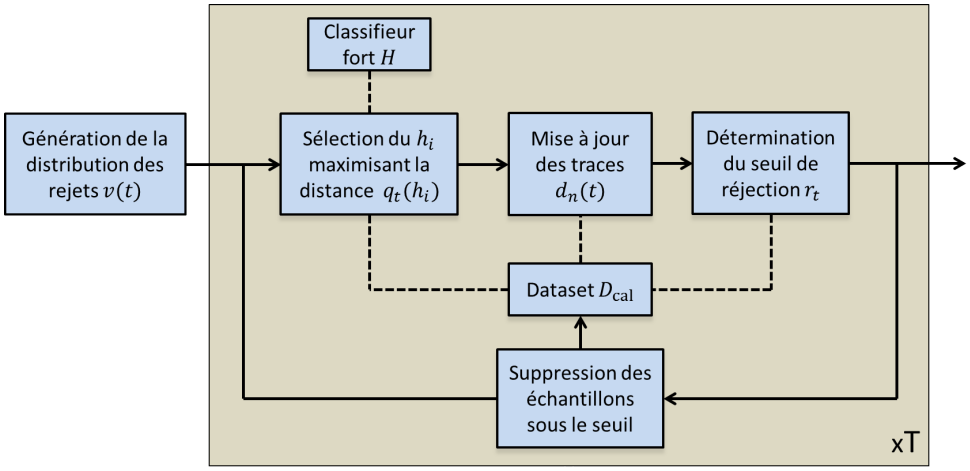


FIGURE 2.13 – Schéma de calibration de SoftCascade [Bourdev and Brandt, 2005].

L'idée principale derrière le SoftCascade est de placer sur les premiers étages les classifieurs faibles les plus discriminants sur la base de calibration. Notons la trace $d_{x_n}(t)$ d'un échantillon

représentant le score cumulé des t-ièmes classifieurs faibles triés : $d_{x_n}(t) = \sum_{i=0}^{t} \alpha_i h_i(x_n)$. Sur chaque itération t de la calibration, SoftCascade sélectionne un classifieur faible en se basant sur un critère de distance entre les positifs et les négatifs de la base. Cette distance est définie de la façon suivante :

$$q_t(h_i) = \sum_{(x_n,y_n)\in D^+_{\text{cal}}} \frac{1}{|D^+_{\text{cal}}|} \left(d_{x_n}(t-1) + \alpha_i \frac{h_i(x_n)+1}{2} \right) - \sum_{(x_n,y_n)\in D^-_{\text{cal}}} \frac{1}{|D^-_{\text{cal}}|} \left(d_{x_n}(t-1) + \alpha_i \frac{h_i(x_n)-1}{2} \right) \quad (2.11)$$

Une fois h_t sélectionné, SoftCascade apprend un seuil de réjection r_t par rapport à une tolérance de rejet des positifs, définie par une distribution de rejets $v(t)$ préalablement établie. Ainsi, r_t est défini de sorte qu'au maximum $\sum_{i=0}^{t} v(i)$ positifs aient été rejetés à l'itération t. v possède la forme suivante :

$$v(t) = \begin{cases} ke^{-\mu\,(1-t/T)} & \text{pour } \mu < 0 \\ ke^{\mu\, t/T} & \text{pour } \mu \geq 0 \end{cases} \quad (2.12)$$

k est choisi en fonction du nombre de positifs que l'on souhaite rejeter sur la base de calibration ou, autrement dit, par le taux de détections manquées que l'on accepte d'avoir sur la base de calibration. μ est un paramètre permettant d'agir sur la forme de la distribution. Avoir un μ négatif, donc une exponentielle décroissante, va entraîner le rejet d'un nombre plus important d'échantillons sur les premiers niveaux et par conséquent rendre la cascade plus rapide, mais ceci se fait au détriment du taux de détection final. C'est pourquoi, le choix de μ dépend du critère que l'on souhaite favoriser pour la cascade, vitesse ou taux de détection.

Une fois r_t établi, tous les échantillons dont la trace est inférieure à ce seuil sont retirés de la base de calibration et l'algorithme boucle sur une nouvelle itération. En phase de test, l'échantillon est rejeté de la cascade si la condition de sortie $d_x(t) < r_t$ est atteinte.

2.5 Résultats et validations

2.5.1 Dataset utilisée

Comme présenté dans l'état de l'art, des bases de piétons sont mises à disposition par la communauté scientifique pour la détection de personnes. Nous avons utilisé pour nos expérimentations la base piétonne de l'INRIA [Dalal and Triggs, 2005] proposée dans le cadre du descripteur HOG. Cette base est composée de photos prises en environnement naturel, principalement urbain, sous des conditions d'illumination variées, notamment entre les photos en extérieur et intérieur. Un inconvénient de cette base est la faible proportion d'exemples présentant des occultations.

Avant de pouvoir utiliser cette base de données, il est nécessaire de la formater. Il s'agit en fait de normaliser les échantillons à une taille fixe, qui représente la taille de la fenêtre glissante. Cette opération requiert un ensemble d'annotations, nommées vérité terrain (ou *ground truth*) indiquant où se trouvent les piétons sur l'image par des boîtes englobantes. Ainsi pour générer les positifs, le contenu de chaque boîte englobante annotée est redimensionné de façon à créer une imagette de 64x128 (largeur x hauteur). Pour éviter un phénomène de bruit causé par un redimensionnement trop important, ce dernier se fait octave par octave. On constate une meilleure qualité de l'imagette du redimensionnement progressif par rapport à un redimensionnement brutal Figure 2.14b. Il est aussi bénéfique d'incorporer une petite marge autour de la personne [Dalal and Triggs, 2005]. Cela permet d'exploiter efficacement les gradients liés aux contours de la personne. Nous choisissons d'incorporer une marge de 10 pixels. Cette opération permet d'obtenir 1237 images de piétons normalisées. Une

manière simple d'augmenter le nombre d'échantillons est de retourner horizontalement chaque image (effet miroir), permettant de porter le nombre de positifs à 2474.

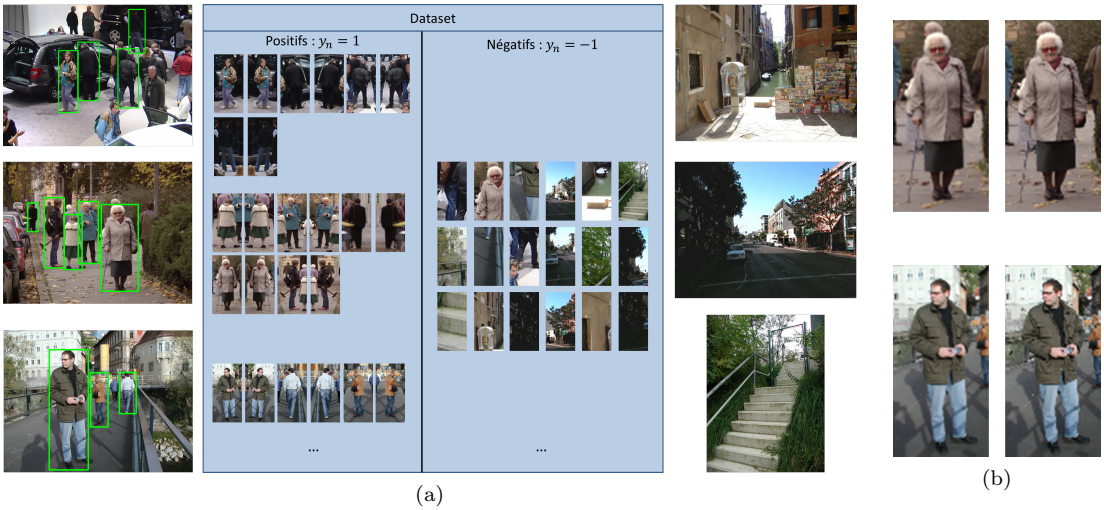


FIGURE 2.14 – (a) Mise en forme de la base piétonne INRIA [Dalal and Triggs, 2005]. Tous les piétons sont normalisés sur une taille fixe et les négatifs sont générés à partir d'images sans piétons. Une part des échantillons négatifs est également prise sur des parties du corps humain. (b) Effet d'un redimensionnement progressif (gauche) sur la qualité de l'image, par rapport à un redimensionnement direct (droite).

Pour générer les négatifs, nous nous basons sur les images vierges de piétons fournie dans la base de l'INRIA. À partir de ces images, nous obtenons les imagettes par une sélection aléatoire de échelle et de la position. De plus, nos premières évaluations ont montré que beaucoup de fausses détections étaient placées sur des parties du piéton lui-même, telles que les jambes ou les bras. C'est pourquoi, nous enrichissons notre base de négatifs avec des images de parties de personnes, en s'assurant bien que le critère de chevauchement défini 2.13 ne soit pas atteint. Cette procédure de génération des négatifs est celle utilisée pour l'étape de *bootstrapping* d'*AdaBoost*. Au total, nous générons 10 000 échantillons négatifs initiaux. Nous découpons cette base initiale de façon à avoir 4/5ème des échantillons pour la base d'apprentissage d'*AdaBoost* et 1/5ème pour la calibration du *SoftCascade*.

2.5.2 Évaluations

Prédiction \ Vérité Terrain	Positif	Négatif
Positif	Vrai Positif (VP)	Faux Positif (FP)
Négatif	Faux Négatif (FN)	Vrai Négatif (VN)

TABLE 2.3 – Dénomination de la prédiction d'un échantillon en fonction de la vérité terrain

Dans cette section, nous évaluons les performances du détecteur sur la base de validation de l'INRIA, distincte de celle d'apprentissage, ainsi que les impacts des éléments apportés par rapport aux détecteurs d'origine. Cette évaluation se base sur diverses mesures statistiques qui s'appuient principalement sur le type de résultat fourni par le classifieur défini dans la Table 2.3 :

- Vrai Positif : échantillon de la classe positif correctement prédit
- Faux Positif : échantillon de la classe négatif, incorrectement prédit en tant que positif
- Faux Négatif : échantillon de la classe positif, manqué par le classifieur et prédit en négatif
- Vrai Négatif : échantillon de la classe négatif, correctement prédit

Dans ce contexte de détection, pour évaluer qu'une boîte englobante est déclarée comme Vrai Positif, il faut qu'elle respecte le critère de recouvrement suivant :

$$\frac{\text{NombrePixels}(D_{\text{Eval}} \cap D_{\text{Vérité}})}{\text{NombrePixels}(D_{\text{Eval}} \cup D_{\text{Vérité}})} \geq 0.5 \tag{2.13}$$

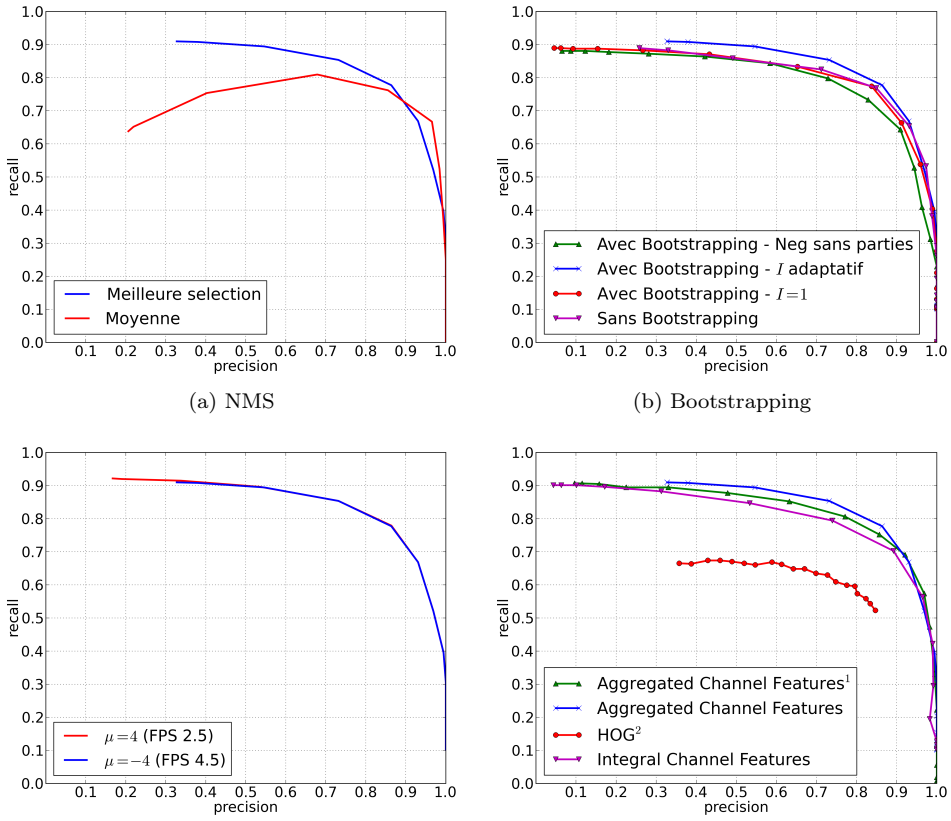
avec $D_{\text{Vérité}}$ la zone de l'instance vérité terrain et D_{Eval} celle de l'instance détectée. Ce critère est l'indice de Jaccard, et est utilisé pour le challenge PASCAL [Everingham et al., 2010]. Plusieurs représentations graphiques existent afin de visualiser la qualité d'un classifieur binaire à partir de ces 4 mesures statistiques. Elles s'appuient notamment sur les performances délivrées par la variation d'un seuil de confiance (ou score de la détection) β, abordé Équation 2.10. Les représentations les plus notables sont :

- **La courbe ROC** (en anglais *Receiver Operating Characteristic*) : *rappel* (ou *sensibilité*) d'un test en fonction du taux de faux positif. Le *rappel* est un indicateur mesurant la capacité du détecteur à retrouver les échantillons aux labels positifs : $\textit{rappel} = \dfrac{VP}{VP+FN}$. Le taux de faux positif représente la proportion d'échantillons considérés comme positifs sur la population totale de négatifs : $\textit{Taux}_{FP} = \dfrac{FP}{VN+FP}$. Cette valeur est reliée à la *spécificité* par la relation suivante : $\textit{Taux}_{FP} = 1 - \textit{spécificité}$.
- **La courbe Rappel/Précision** : *rappel* en fonction de la *précision*. La *précision* est une mesure représentative des résultats corrects sur les échantillons renvoyés positifs par le détecteur : $\textit{précision} = \dfrac{VP}{VP+FP}$.
- **La courbe Taux de Détection Manquée/Faux Positifs par Fenêtre** (en anglais *miss rate*/*False positives per window* (FPPW)). Le taux de détection manquée est complémentaire à la *sensibilité* : $\textit{miss rate} = \dfrac{FN}{FN+VP} = 1 - \textit{sensibilité}$ et indique la proportion d'échantillons positifs que le détecteur a raté. La variable en abscisse est le nombre moyen de faux positif par position de la fenêtre glissante. Cette courbe n'est adaptée qu'à des détecteurs fonctionnant sur une technique de fenêtre glissante. Un inconvénient est qu'elle ne peut pas intégrer les traitements postérieurs à l'évaluation de la fenêtre (comme le *Non-Maxima Suppression*).
- **La courbe Taux de Détection Manquée/Faux Positifs par Image** (*miss rate*/*FPPI*). Cette courbe est équivalente à la précédente, à ceci près que le nombre de faux positifs moyens est considéré par image afin de prendre en compte l'ensemble des traitements d'un détecteur. Cette courbe est devenue un standard d'évaluation de détecteurs de piétons pour des bases sous forme de vidéos telles que Caltech-USA [Dollár et al., 2012].

Nous utiliserons principalement, pour nos évaluations, la courbe Rappel/Précision. La métrique ROC est moins adaptée car elle rejoint dans une certaine mesure la courbe du *miss rate*/FPPW sur l'évaluation d'imagettes et non sur l'image complète. La courbe *miss rate*/FPPI est privilégiée pour les séquences vidéos, notamment dans les systèmes automobiles où l'on fixe un seuil de tolérance pour le taux de faux positifs [Dollár et al., 2012]. Dans le cadre d'une base statique comme celle de l'INRIA, les deux courbes : Rappel/Précision et *miss rate*/FPPI délivrent des mesures équivalentes.

Paramétrages

Nous entraînons un *AdaBoost* avec $T = 2000$ classifieurs faibles. Pour chaque itération de l'apprentissage, nous générons un ensemble de $K = 600$ candidats. Concernant le *bootstrapping*, nous ajoutons autant d'échantillons négatifs de sorte que le nombre d'exemples issus du *bootstrapping*



(a) NMS (b) Bootstrapping

(c) Calibration SoftCascade (d) Travaux existants

FIGURE 2.15 – Évaluations du détecteur sur des courbes rappel/précision. Plus la courbe se rapproche du point (1,1) et meilleures sont les performances du détecteur. (a) Comparaison de l'approche de *Non-Maxima Suppression* par la sélection de la meilleure instance de détection avec le moyennage de toutes les instances. (b) Apport du *bootstrapping* en fonction de la manière d'ajouter des échantillons négatifs. (c) Effet de la calibration SoftCascade sur les performances de détection (d) Comparaison de nos implémentations avec la version officielle [1]et la version HOG fournie par la librairie OpenCV [2].

reste constant. Pour la calibration *SoftCascade*, nous fixons un taux de rejet de positifs de 5% et $\mu = -4$. Le multi-échelle à l'étape de détection est traité sur 4 octaves comprenant entre chacun 8 approximations d'échelles.

Non-Maxima Suppression (NMS)

Comme avancé dans la Section 2.3.1, le rôle de l'algorithme NMS est de regrouper les instances renvoyées positives après le passage de la fenêtre glissante. Pour des détecteurs non basés par parties, le regroupement se considère selon le critère de chevauchement suivant :

$$\frac{\text{NombrePixels}(D_{\text{Eval}} \cap D_{\text{Vérité}})}{\min(\text{NombrePixels}(D_{\text{Eval}}), \text{NombrePixels}(D_{\text{Vérité}}))} \geq 0.65 \tag{2.14}$$

Ce groupe s'élargit avec toutes les instances chevauchant un de ses éléments et respectant cette condition. De façon similaire aux travaux de [Dollar et al., 2009], le critère de recouvrement utilisé permet de limiter le nombre de faux positifs du type sous-parties du corps humain en pénalisant

1. http://vision.ucsd.edu/~pdollar/toolbox/doc/
2. http://openCV.org/

les résultats de petites taille. Nous comparons deux façons de regrouper les détections. La première est de ne retenir que la détection du groupe ayant le meilleur score de confiance. La deuxième est de définir la zone comme étant la moyenne, pondérée par les scores du détecteur, des instances du groupe. La Figure 2.15a indique que le NMS basé sur la sélection de la détection avec le plus de confiance au sein d'un groupe fournit les meilleurs résultats.

Bootstrapping

Nous évaluons l'impact que peut avoir le *bootstrapping* sur l'apprentissage d'*AdaBoost* 1. On constate avec la Figure 2.15b qu'il permet d'améliorer substantiellement les résultats par rapport à un apprentissage sans bootstrapping. Néanmoins, il est intéressant de noter que lorsque le nombre I d'échantillons à ajouter est fixé, les performances fournies par le classifieur décroissent. Cela vient du fait qu'*AdaBoost* ne parvient pas à suivre le rythme imposé par un ajout régulier d'échantillons difficiles. Dans le cas d'un I adaptatif, environ 500 échantillons ont été ajoutés au cours du processus d'entraînement.

Classifieurs faibles

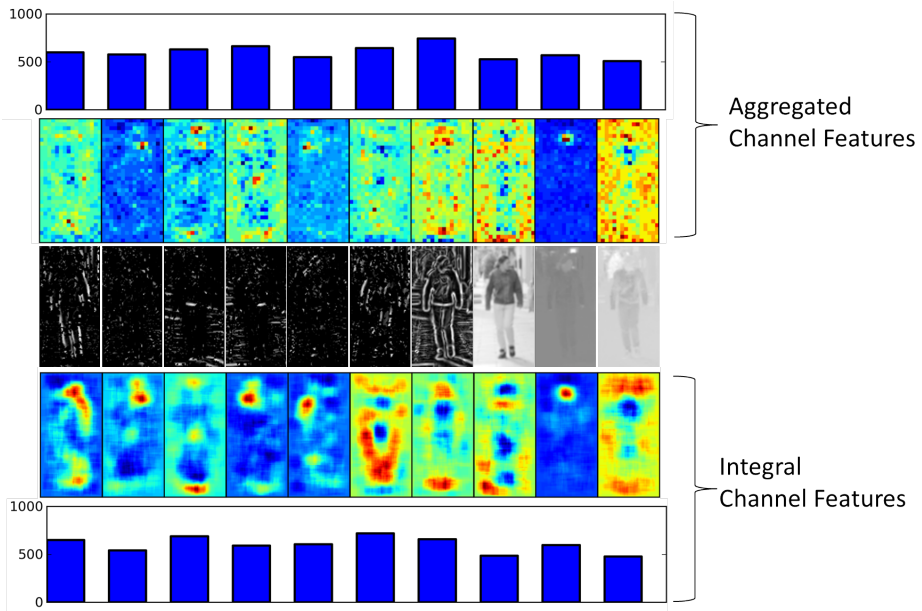
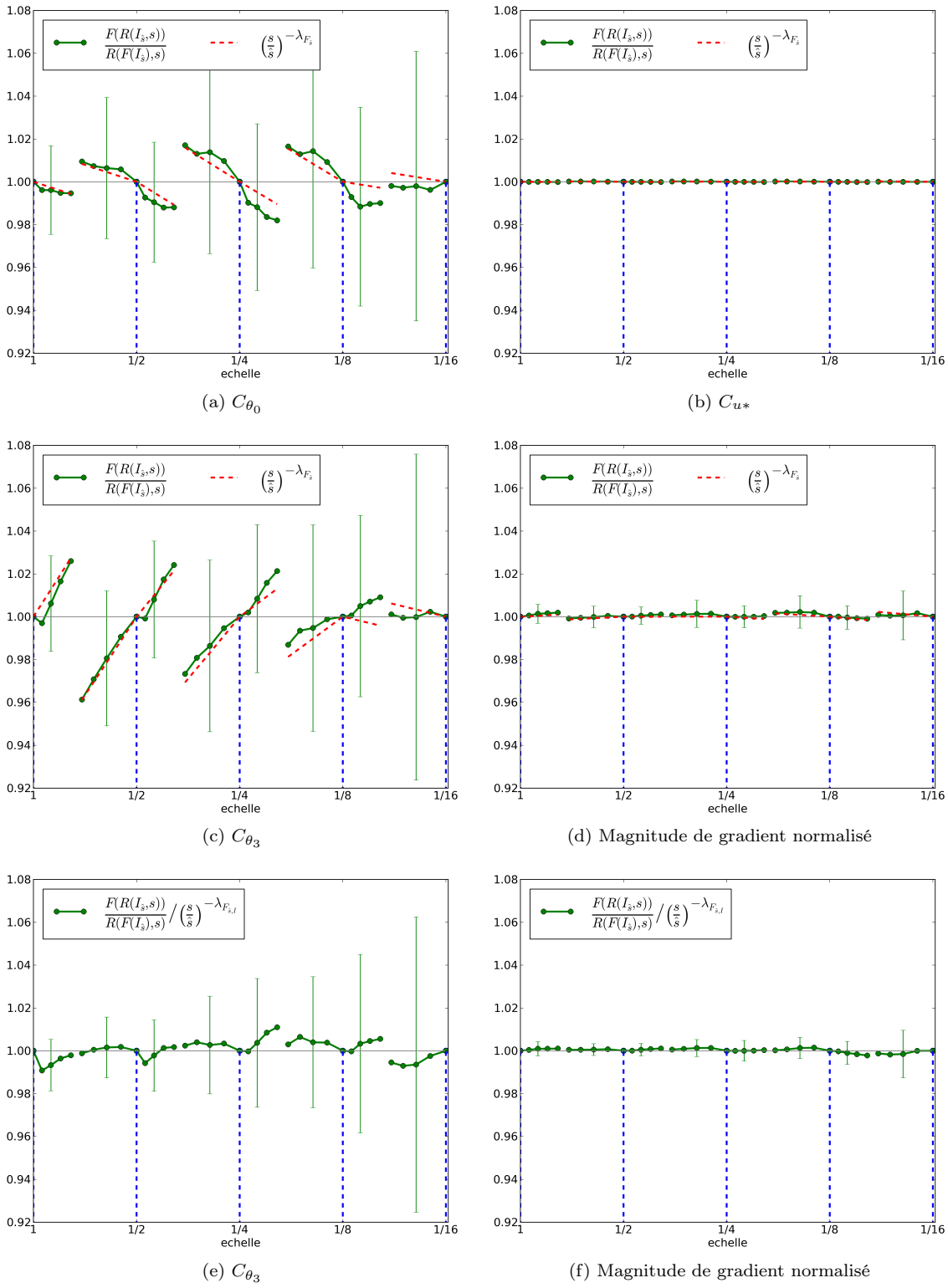


FIGURE 2.16 – Visualisation des 2000×3 (arbres binaires à 2 niveaux) caractéristiques locales sélectionnées par *AdaBoost* pour l'agrégation (en haut) et les sommes de blocs (en bas). Les couleurs chaudes indiquent des zones où les classifieurs faibles attendent des valeurs élevées (seuillage supérieur) tandis que les couleurs froides indiquent des zones pour des valeurs faibles (seuillage inférieur). La ligne du milieu représente un exemple des *channels* calculés.

En expérimentant la possibilité d'un seuillage supérieur unique, nous avons constaté une baisse drastique des performances (environ 50% en terme de rappel). En effet, dans ce cas, les classifieurs sont dans l'incapacité de considérer des zones locales à faible valeur, comme le centre du corps qui possède peu de gradients. L'erreur d'apprentissage obtenue sur un seuillage unique est d'environ 13%, ce qui est élevé pour un apprentissage d'*AdaBoost* dont l'erreur d'apprentissage est habituellement proche de 0%. Il est aussi intéressant de connaître la répartition des caractéristiques sur les *channels* et quelles sont les valeurs attendues suivant les zones de l'image. On visualise, Figure 2.16, qu'*AdaBoost* a sélectionné des caractéristiques représentatives du piéton. On devine sur le *channel* de magnitude de gradient (7ème) le contour global des personnes, ainsi que la présence du visage sur le *channel* $\mathbf{u^*}$ (9ème) marqué par une forte concentration de hautes valeurs. En outre, les caractéristiques sont plus ou moins distribuées équitablement entre tous les *channels*.



(a) C_{θ_0} (b) C_{u*}

(c) C_{θ_3} (d) Magnitude de gradient normalisé

(e) C_{θ_3} (f) Magnitude de gradient normalisé

FIGURE 2.17 – Évaluation de l'approximation multi-échelle de génération des *channels* sur une loi de puissance sur 2572 images de la base piétonne INRIA [Dalal and Triggs, 2005]. (a)(b)(c)(d) représentent la fidélité, en terme de moyenne globale, de l'approximation par rapport aux échantillons, tandis que le facteur d'approximation pour (e) et (f) est calculé pour chaque image. Les points de courbe indiqués par les pointillés bleus à l'octave $\hat{s}$ représentent les *channels* repères. Le comportement est plus instable pour des *channels* éparpillés, illustrés par des écarts-types plus importants pour (a) et (c) par rapport à (b) et (d). L'adaptation du facteur d'approximation image par image (e) et (f) permet de stabiliser l'écart-type par rapport à un facteur global d'approximation (c) et (d).

Approximations adaptatives

Cette amélioration intervient lors de la génération multi-échelle des caractéristiques. Nous proposons d'adapter le facteur d'approximation en fonction de chaque intervalle d'octaves en échelle. Nous vérifions l'approximation Équation 2.8 pour interpoler les *channels* sur l'ensemble des images de la base de l'INRIA. Pour cela, nous étudions Figure 2.17 la différence de comportement de l'énergie (somme totale des pixels) des *channels* générés après redimensionnement de l'image $F(R(I, s))$ et ceux obtenus à partir du redimensionnement avec les *channels* à l'échelle originale $R(F(I), s)$, en fonction du facteur d'approximation proposé. Nous voyons que les écarts-types dans le cas où le facteur λ_F est adapté à l'image sont moins importants, traduisant une meilleure fidélité globale du *channel* approximé par rapport à celui calculé de l'image redimensionnée.

Avec les travaux existants

Nous avons évalué les performances du détecteur par rapport à l'implémentation originale (Figure 2.15d). Le détecteur de cette dernière est ré-entraîné sur la base de l'INRIA avec un seul *AdaBoost* de 2000 classifieurs faibles. Nous obtenons des résultats proches de cette dernière version et voyons que l'utilisation de caractéristiques agrégées sur une grille surpasse effectivement en performance les caractéristiques somme de blocs. Il est à noter que, de façon similaire à [Viola and Jones, 2001], la version officielle utilise une cascade d'*AdaBoost* (4 *AdaBoost* mis en cascade, respectivement de 32, 128, 512 et 2048 classifieurs faibles) permettant d'augmenter les performances globales d'environ 6%.

Temps de calcul

Les expérimentations ont été réalisées sur un processeur i5 de 2.5GHz en *monothread* en C++. Nous calculons le temps de calcul moyen d'évaluation sur l'ensemble des images de 640x480 de la base de test de l'INRIA. La génération des *channels* (sur 3 octaves dans le cas présent) fonctionne en moyenne à 16.9 FPS. En ajoutant l'approximation entre chaque octave, notre implémentation tourne autour de 4.7 FPS. Enfin, avec la fenêtre glissante et le NMS, nous atteignons au final une vitesse d'exécution de 4.5 FPS. Cette vitesse est en deçà de celle obtenue pour [Dollar et al., 2014], principalement ralentie par la fonction de redimensionnement d'images qui provient de la librairie OpenCV (chute de 12 FPS). Une manière d'accélérer le système serait d'optimiser cette opération de redimensionnement grâce aux instructions optimisées telles que les SSE2. Concernant le *SoftCascade*, en paramétrant $\mu = 4$ de sorte que l'on rejette moins de positifs de calibration sur les premiers étages de la cascade, le FPS tombe à 2.5. Le gain en performance réside dans l'obtention de détections à faible valeur de confiance (Figure 2.15c) et se révèle peu intéressant par rapport à une vitesse de calcul presque 2 fois plus élevée.

2.5.3 Portage sur smartphone

Pour nos expérimentations sur smartphone, nous utilisons le Samsung Galaxy Note 2 qui est un smartphone fonctionnant avec le système d'exploitation Android. L'implémentation précédente est embarquée dans le smartphone par l'utilisation d'une interface [3] permettant la communication entre du code natif et le code Java supporté par Android. Nous obtenons sur ce smartphone un fonctionnement aux alentours de 3 FPS. Par ailleurs, la plupart des smartphones actuels disposent de capteurs qui peuvent être utilisés afin de déterminer l'orientation de l'appareil. Nous faisons l'hypothèse que la prise d'images à partir d'un smartphone se fait de manière statique. Une façon simple pour compenser la rotation de l'image dû au roulis de l'appareil est de s'appuyer sur la centrale inertielle embarquée qui fournit un vecteur accélération où la composante de la gravité intervient :

$$\overrightarrow{A_d} = -\overrightarrow{g} - \sum \frac{\overrightarrow{F}}{masse} \tag{2.15}$$

3. L'interface utilisée est JNI (Java Native Interface)

L'application d'un filtre passe-bas nous permet d'obtenir la direction de la gravité dans le repère du smartphone et par conséquent, son orientation. Nous appliquons la matrice de rotation avec l'opposé du roulis de l'appareil sur l'image obtenue de la caméra pour l'orienter de la bonne façon par rapport au smartphone. Pour les systèmes non équipés de ce genre de capteurs, il est nécessaire de gérer soi-même l'invariance à la rotation directement sur l'image avec par exemple un apprentissage sur plusieurs rotations [Smedt and Goedemé, 2015].

Chapitre 3

Caractérisation de la structure corporelle par Modèle à Distribution de Points

Sommaire

3.1 Introduction **33**
3.2 Représentation de la forme d'une personne **34**
3.3 État de l'art - Alignement de Modèle à Distribution de Points **38**
3.4 Pré-requis à l'alignement d'un PDM **44**
3.4.1 Définition du modèle 44
3.4.2 Annotation d'une nouvelle Dataset 46
3.5 Modèle d'apparence par boosting sur une forme paramétrique **48**
3.5.1 Modélisation paramétrique 48
3.5.2 Modèle d'apparence par GentleBoost 50
3.5.2.1 Procédure d'apprentissage 51
3.5.2.2 Caractéristiques 52
3.5.2.3 Régresseurs faibles 54
3.5.2.4 Alignement par maximisation du score 55
3.5.3 Évaluations et validations 57
3.5.3.1 Métrique d'évaluation 57
3.5.3.2 Validation de l'algorithme 57
3.5.4 Limitations et discussions 61
3.6 Régression de forme par classification des déformations **63**
3.6.1 Cascade de régressions de forme 63
3.6.2 *Clustering* et classification des déformations 64
3.6.2.1 Simulation et généralisation des formes 65
3.6.2.2 Stratégie pour la classification 67
3.6.2.3 Caractéristiques et indexations 68
3.6.2.4 Évaluations et validations 69
3.6.3 Discussions 74

3.1 Introduction

Ce chapitre est consacré à la seconde étape du système global présenté dans cette thèse. Il s'agit de la phase intermédiaire de traitement précédant celle de ré-identification. Elle s'articule sur la caractérisation structurelle de la personne sur l'image, opérée dans la boîte englobante fournie par la phase de détection. L'intérêt d'introduire une phase de pré-caractérisation à la génération de la signature de la personne est triple :

- La segmentation. Il s'agit de séparer la personne du fond et de conserver uniquement l'information utile à la génération d'une signature.
- Le renforcement spatial de la signature visuelle. Dès lors que la configuration structurelle de la personne est connue, l'association des descripteurs visuels à des zones locales identifiées devient possible. Ici, l'intérêt est de rendre invariant la signature à la pose que peut adopter la personne.
- Le support à la génération d'une signature. On peut envisager d'utiliser directement la structure extraite de la personne pour sa ré-identification, notamment dans le cadre d'une reconnaissance de forme (par exemple, la forme globale de la silhouette). Une autre exploitation possible relève de l'étude comportementale de la personne. L'évolution de la structure est évaluée sur une fenêtre temporelle, ce qui se fait, par exemple, pour de la reconnaissance de démarche.

La caractérisation structurelle se décline sous diverses formes qui vont répondre aux besoins du domaine d'application. En complément de la détection qui fournit une boîte englobante approximative de la personne, l'extraction de la structure précise la localisation de la personne dans l'image pour améliorer la génération de la signature.

Ce chapitre est présenté de la façon suivante. Dans un premier temps, nous donnons un bref aperçu des différents modèles de représentation des personnes et justifions notre orientation vers le Modèle à Distribution de Points.

La seconde partie porte sur l'état de l'art autour de la problématique d'alignement de ce modèle, jusqu'alors principalement appliqué aux visages.

En troisième partie, nous décrivons une nouvelle base annotée que nous proposons d'introduire ainsi que les spécificités du modèle choisi.

La dernière partie est consacrée aux approches proposées et adaptées au corps humain ainsi qu'à leurs évaluations sur la nouvelle base.

3.2 Représentation de la forme d'une personne

Il existe de nombreuses façons de modéliser une personne sur l'image. Le type de modèle adopté va notamment dépendre de l'application considérée. En effet, suivant les besoins, l'information de pose de la personne va primer sur la qualité de segmentation. Dans d'autres cas d'utilisation, connaître seulement l'emplacement qu'occupe la personne sur l'image peut suffire. Nous listons ci-après les principaux types de modélisation existants :

Silhouette (Figure 3.1a)

La silhouette est la représentation d'une zone pleine délimitée par un ou plusieurs contours de l'objet considéré. En d'autres termes, la silhouette est un masque binaire de l'image indiquant quels pixels appartiennent à l'objet. Elle peut posséder des zones non contigües qui peuvent résulter soit d'une erreur dans le processus d'extraction, soit de l'occultation d'une partie de l'objet. En revanche, la silhouette telle quelle ne permet pas de distinguer l'occultation provoquée par l'objet lui-même. Dans le cas d'un corps humain, il peut s'agir des bras occultant une partie du corps ou une jambe placée devant l'autre.

Son extraction peut se faire par des approches de séparation entre le fond et la personne. Il s'agit d'un vaste domaine d'études affilié à la segmentation d'objets qui s'appuie sur le fait qu'un objet à segmenter va présenter des régions homogènes en terme d'apparence. [Jojic et al., 2009] exploitent cette idée en modélisant statistiquement la structure d'un objet entre ces régions. D'autres méthodes présentées par [Boykov and Jolly, 2001] ou [Rother et al., 2004] s'appuient sur les coupes de graphes, qui permettent, en partant d'une zone initiale dans l'image, de segmenter de façon optimale le fond et la personne en regroupant les régions homogènes connexes.

Une autre méthode d'extraction de silhouette, déjà abordée dans la Section 2.2 est la soustraction d'environnement. Cette méthode opère le plus souvent dans une configuration où la caméra est fixe. Cependant, la gestion d'un point de vue dynamique est aussi possible en faisant un apprentissage préalable de l'environnement. C'est le cas de [Zhuang and Chen, 2007] qui utilisent le flot optique induit par le mouvement de la caméra pour pouvoir caractériser le fond.

La silhouette est fréquemment employée pour des tâches de ré-identification. Pour améliorer la robustesse d'une signature basée sur l'apparence, certains travaux proposent, en post-traitement, d'extraire une structure approximative de la personne. C'est le cas de [Farenzena et al., 2010] qui la découpent verticalement et génèrent une signature par rapport à un axe de symétrie vertical. De plus, la silhouette est un support classique pour faire de la reconnaissance de démarche [Wang et al., 2003].

Contours (Figure 3.1b)

Les modèles de contours sont similaires à la silhouette en terme d'information sur l'image. Ils s'en distinguent par leur technique d'extraction. La représentation par contours a comme avantage, par rapport à la silhouette, de pouvoir distinguer les membres occultant une partie du corps. Dans ce cas, l'établissement d'une structure initiale est requise comme pour les travaux de [Freifeld et al., 2010]. D'autres approches utilisent les modèles de contours actifs qui consistent en l'évolution d'une courbe soumise à certaines contraintes. On peut notamment citer les méthodes de contours de type *Snake* [Kass et al., 1988] qui vont chercher à approcher l'objet suivant ses bordures. [Chan and Vese, 2001] proposent d'exploiter la formulation *level-set* pour résoudre le cas où l'objet présente des zones séparées. [Bresson et al., 2007] établissent une minimisation globale de la fonction d'énergie permettant de s'affranchir des minima locaux et de garantir une indépendance vis-à-vis de la position initiale du contour. [Horbert et al., 2011] améliorent la robustesse de ce genre d'approche en prenant en compte l'apparence spécifique des personnes.

Modèle volumique 3D (Figure 3.1c)

Une autre façon de représenter une personne est un modèle volumique tridimensionnel. Suivant le degré de précision souhaité pour modéliser la personne, des capteurs plus ou moins performants vont être utilisés. Il peut s'agir de marqueurs (le plus souvent infrarouge) que doit revêtir la personne [Anguelov et al., 2005]. D'autres systèmes proposent d'utiliser les images de profondeur pour modéliser en 3D la forme de la personne [Shotton et al., 2011]. Les modèles 3D ont comme cadre d'application l'animation numérique de personnes, la télé-présence, la reconnaissance de pose ou la ré-identification. Ce dernier cas d'application ne s'est développé que récemment dû notamment à l'émergence des caméras de profondeur. La signature de la personne est construite à partir de la forme globale modélisée de la personne, par exemple par un jeu de distances au niveau des membres du corps [Munaro et al., 2014; Barbosa et al., 2012]. Bien que ce type de modèle soit très précis pour décrire la forme d'une personne et montre des résultats prometteurs dans le domaine de la ré-identification, les capteurs à mettre en œuvre dépassent le cadre d'application de cette thèse.

Pictorial Structures et Squelettes (Figure 3.1d et Figure 3.1e)

Originellement introduits par [Fischler and Elschlager, 1973] et popularisés par [Felzenszwalb and Huttenlocher, 2005] pour la reconnaissance d'objets, ce modèle repose sur la recomposition spatiale de parties détectées séparément selon des contraintes structurelles apprises statistiquement. Sa formulation s'adapte bien au corps humain car les articulations peuvent être modélisées par des liaisons cinématiques. Son fonctionnement s'établit sur la construction d'une carte de probabilité de l'emplacement des parties du corps, par les scores de confiance fournis par les détecteurs de parties. Ces derniers peuvent fonctionner sur un modèle de fenêtre glissante comme dans les travaux de [Andriluka et al., 2009]. La configuration spatiale des parties du corps est déterminée avec les contraintes liées au modèle d'articulations.

Les domaines d'application des *Pictorial Structures* sont principalement la reconnaissance de pose et la détection de personnes. Ils rejoignent en ce sens les modèles déformables par parties.

FIGURE 3.1 – Illustration des principaux modèles existant pour représenter une personne. (a) Silhouettes. Gauche : Soustraction de fond [Bak et al., 2010a]. Droite : Grabcut [Rother et al., 2004] (b) Contours. Gauche : *Level-set* [Horbert et al., 2011]. Droite : Basé sur un modèle articulé [Freifeld et al., 2010] (c) Modèle Volumique 3D SCAPE [Anguelov et al., 2005] (d) *Pictorial Structures* [Andriluka et al., 2009] (e) Squelette obtenu par modèle de mélange [Yang and Ramanan, 2011] (f) Modèle à Distribution de Points [Liu et al., 2008]

Quelques travaux récents ont cherché à exploiter ce modèle pour de la ré-identification [Cheng and Cristani, 2014], renforçant la signature par l'information de localisation des différentes parties. Par ailleurs, des travaux comme ceux de [Zuffi et al., 2012] étendent les *Pictorial Structures* pour obtenir des modèles dits déformables où les différentes parties s'adaptent aux courbes du corps améliorant la précision à l'image des formes de la personne. [Pishchulin et al., 2013] conditionnent l'arbre cinématique des articulations avec les *Poselets* [Bourdev and Malik, 2009] (partie d'une personne sous une certaine pose) afin de pouvoir gérer les dépendances entre les parties non adjacentes dans l'arbre.

Les modèles de squelettes se composent d'un ensemble de points de jointures placés au niveau des articulations. Nous les plaçons dans la même rubrique que les *Pictorial Structures* car leur obtention présente des similitudes. En effet, l'approche s'appuie aussi sur la combinaison entre une modélisation de la configuration spatiale du corps et d'apparence pour les parties. Pour rendre plus flexible l'estimation de pose, [Yang and Ramanan, 2011] proposent de délaisser les opérations de transformation (telles que la rotation) afin de détecter les parties au profit de *templates* groupés par un modèle de mélanges (en anglais *Mixture Model*). D'autres travaux tels que ceux de [Toshev and Szegedy, 2013] utilisent le *Deep Learning* présentant l'avantage de considérer de façon globale l'apparence de la personne pour déterminer sa pose. La représentation par un squelette s'avère efficace sur des caractéristiques de type métriques [Munaro et al., 2014] mais est mal adaptée pour exploiter l'apparence de la personne. En effet, l'information visuelle de la personne n'est pas englobée dans le modèle parce que le squelette est constitué de segments centrés dans la forme d'origine.

Modèles à Distribution de Points (Figure 3.1f)

Les Modèles à Distribution de Points (en anglais *Point Distribution Model*, que nous abrégerons en *PDM*) ont été initialement introduits dans les travaux de [Cootes et al., 1995]. Il s'agit d'un modèle de forme constitué par plusieurs points placés à des localisations clés et pré-définies nommées amers (ou *landmarks* en anglais). Leur utilisation s'apparente à celle utilisée pour les *Pictorial structures* dans la mesure où ils sont usuellement couplés à un modèle d'apparence. Par contre, les contraintes spatiales de forme sont principalement déterminées statistiquement par rapport à une base d'apprentissage, là où les *Pictorial Structures* s'appuient sur un modèle de graphe défini. De ce fait, les PDM s'adaptent bien aux objets déformables et vont se révéler précis au niveau du placement des amers. C'est pourquoi, ils sont utilisés majoritairement sur les visages Cootes et al. [2004] et ce, dans le but de faire une reconnaissance faciale, d'expressions ou encore pour de l'animation. En contrepartie, ce type de modèle va globalement moins bien gérer un objet articulé tel que le corps humain du fait de la grande variété de formes qu'il peut adopter. À notre connaissance, seuls les travaux de [Liu et al., 2008] ont abordé l'alignement d'un Modèle à Distribution de Points sur le corps humain. Cette étude est arrivée à des résultats satisfaisants à condition que l'initialisation soit proche dans l'espace des déformations de la forme escomptée.

	Segmentation	Configuration structurelle	Description comporte-mentale	Métrique du corps
Silhouette	++++	+	++	+
Contours	++++	++	++	+
Pictorial Structures	++	++++	++++	++
Modèles volumiques 3D	++++	++++	++++	++++
Squelettes	+	++++	++++	+++
Modèles à Distribution de Points	+++	+++	+++	+++

Table 3.1 – Comparaison qualitative des modèles de représentation pour le corps humain en vue d'une application de ré-identification.

Nous considérons que le modèle volumique 3D est le support le plus performant pour une tâche de ré-identification (Table 3.1) car il est le plus expressif en terme de description de la forme de la personne. Néanmoins, des moyens spécifiques à mettre en œuvre pour une extraction précise sont encore nécessaires, tels qu'une caméra de profondeur. Ce type de technologie montre ses limites dans le cadre d'un réseau de vidéo-surveillance où les sujets sont loin des caméras et peut s'avérer encore coûteux à embarquer sur des appareils tels que les smartphones. Nous proposons dans cette thèse d'exploiter les Modèles à Distribution de Points pour décrire la structure et la forme d'une personne à l'image. Outre le caractère innovant de ce type de représentation (exploité une seule fois sur le corps humain [Liu et al., 2008] et non essayé sur une tâche de ré-identification), nous sommes motivés par le fait que ce type de modèle permet d'obtenir les résultats très performants pour la reconnaissance faciale [Taigman et al., 2014]. Ainsi, nous souhaitons transposer le potentiel de ce modèle à la ré-identification sur le corps entier. Par ailleurs, ce modèle présente en théorie la possibilité de segmenter efficacement la personne sur l'image avec le placement précis des amers. Il peut aussi servir de support à la génération de caractéristiques basées sur la forme, mais aussi sur le comportement en étudiant l'évolution temporelle de points précis du corps.

3.3 État de l'art - Alignement de Modèle à Distribution de Points

L'alignement d'un Modèle à Distribution de Points a été proposé par [Cootes et al., 1995] avec les *Active Shape Models* (ASM). L'objectif était de palier au manque de robustesse des modèles jusqu'alors utilisés, comme les *Snakes*, par rapport au contenu de l'image. En effet, ces modèles sacrifient une part des contraintes de forme dans le but d'être plus flexibles pour s'aligner sur n'importe quel objet. Les auteurs avancent que, pour gagner en robustesse, le modèle ne se déforme que sur l'appui de caractéristiques propres à chaque type d'objet, apprises grâce à une base d'apprentissage annotée. Les modèles à base de PDM sont dits statistiques et dépendent fortement de la base d'entraînement. L'enjeu est de concevoir un modèle capable de généraliser, à partir d'une base d'apprentissage, la variabilité d'un objet en terme d'apparence et de déformations tout en respectant les contraintes de forme liées à cet objet.

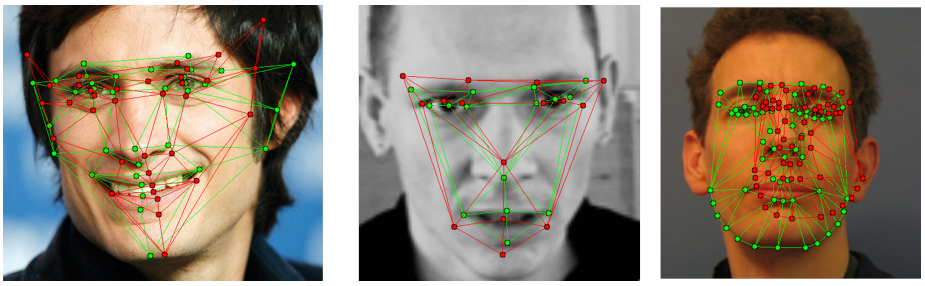


FIGURE 3.2 – Exemples de l'alignement de 3 Modèles à Distribution de Points différents appliqués au visage. L'objectif est d'aligner chaque point rouge sur les points verts. Le maillage représenté est facultatif et peut servir de support d'extraction de la texture.

Le modèle de forme utilisé pour retranscrire la variabilité des déformations est construit avec une Analyse en Composante Principale (ACP) sur l'ensemble des formes alignées de la base d'entraînement. Cette opération a pour but d'extraire d'un phénomène multivarié, dans le cas présent les coordonnées de chaque amer, les axes principaux ou modes. Cela permet de :

- réduire la dimensionnalité du problème
- d'extraire la tendance générale en terme de formes au sein de la base d'apprentissage par les modes
- de décrire le modèle de forme de façon paramétrique, avec une combinaison linéaire des modes et de la forme moyenne. De plus, il est possible de contraindre la forme en bornant ces paramètres

Pour pouvoir aligner la forme sur l'image, les ASM modélisent une force d'attraction de chaque point du PDM avec les bords les plus proches extraits dans l'image, pondérée par l'intensité de ces derniers. Ce processus se fait itérativement, avec une mise à jour de la forme, permettant l'ajustement progressif de cette dernière sur l'image.

Une évolution de ce modèle est apportée par les *Active Appearance Models* (AAM) [Cootes et al., 2001]. Jusque là, les ASM ne proposent pas d'apprentissage de l'apparence par rapport à la base d'entraînement. De façon similaire au modèle de forme, les AAM calculent un modèle d'apparence avec une ACP en alignant toutes les images de la base sur une forme commune de référence par une fonction de transformation (*warping*). Cela permet de disposer de deux modèles paramétriques, l'un décrivant la forme et l'autre l'apparence globale de l'objet (Figure 3.3). Les AAM sont des modèles génératifs. En effet, ils sont capables de générer eux-mêmes des échantillons de l'objet considéré en faisant varier les paramètres des modèles construits avec les ACP. La phase d'alignement est guidée par l'image résiduelle, calculée entre l'image contenue dans la forme et l'instance du modèle

d'apparence avec les paramètres courants. Il s'agit d'un problème d'optimisation non-linéaire dont l'objectif est de minimiser la norme de l'image résiduelle vectorisée en faisant varier les paramètres du modèle de forme. Les auteurs proposent l'approximation d'une relation linéaire entre l'image résiduelle et la forme résiduelle (différence entre la forme courante et la forme objective). Ils effectuent ensuite une régression linéaire multivariée sur les échantillons de la base en perturbant les paramètres de forme de ces derniers. Avec ces travaux, les AAM sont devenus une référence pour l'alignement de modèles déformables appliqués aux visages et montrent des résultats prometteurs pour la reconnaissance faciale [Edwards et al., 1998].

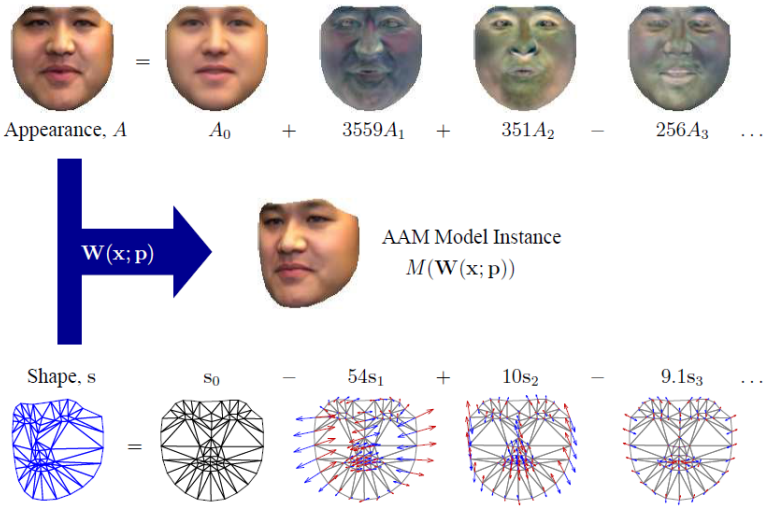
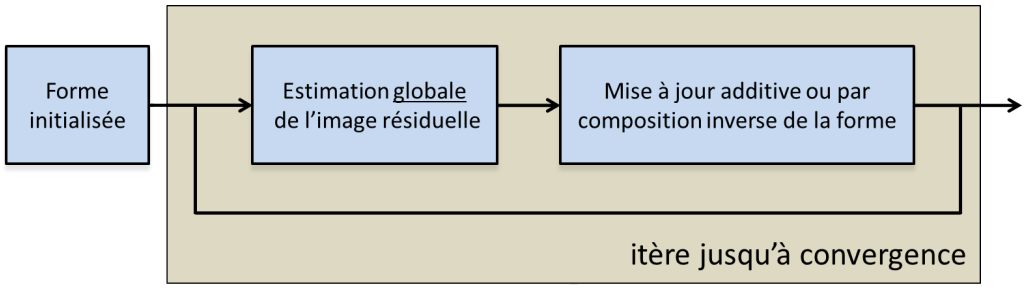


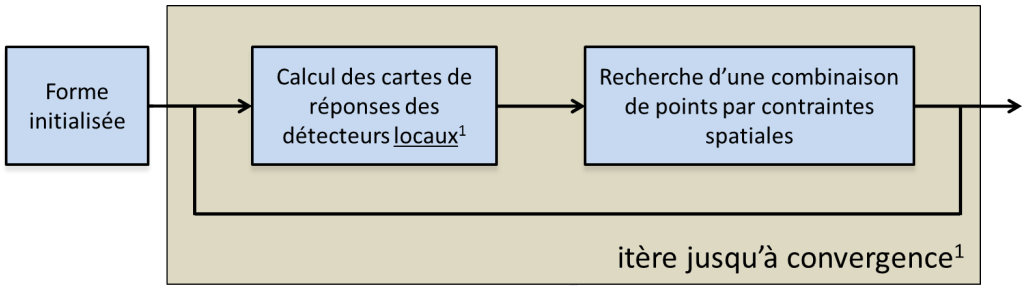
FIGURE 3.3 – Les *Active Appearance Models* sont des modèles paramétriques (d'apparence en haut et de forme en bas), décrits par une combinaison linéaire entre la moyenne et les modes obtenus par une analyse en composante principale. Ce modèle est capable de générer de nouveaux exemples réalistes en faisant varier les différents paramètres. Image tirée de [Matthews and Baker, 2004].

[Hou et al., 2001] proposent de considérer qu'à une texture donnée correspond un jeu de déformations à appliquer. Pour cela, ils entraînent une fonction de régression entre les paramètres liés à l'apparence et ceux liés à la forme. Bien que cette méthode montre des améliorations dans le cas des visages dans un environnement sous contrainte (visage pris de face à la caméra), cette considération a des limites dans le cas où l'objet considéré possède exactement la même apparence mais une forme différente [Cootes et al., 2004].

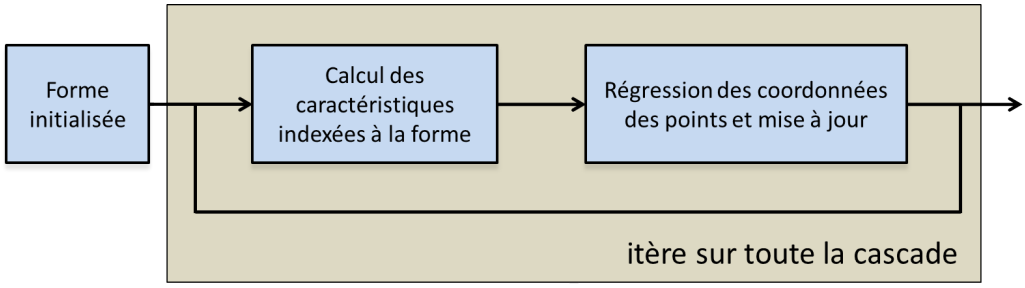
Une autre extension aux AAM a été proposée par [Matthews and Baker, 2004]. Soit $W(x, y, p)$ la correspondance entre le pixel (x, y) d'une forme de base et la forme obtenue par les paramètres de forme p. Au lieu de mettre à jour les paramètres de forme de façon additive $p \leftarrow p + \Delta p$, les auteurs proposent de mettre à jour la correspondance des pixels par une composition avec la correspondance d'incrément estimée $W(x, y, p) \leftarrow W(x, y, p) \circ W(x, y, \Delta p)$. De plus, ils démontrent qu'inverser le problème pour calculer cette correspondance d'incrément vers l'image de référence plutôt que vers l'image d'exemple permet à l'algorithme de faire des mises à jour seulement linéaires $W(x, y, p) \leftarrow W(x, y, p) \circ W^{-1}(x, y, \Delta p)$. Ce problème d'optimisation non-linéaire est résolu grâce à un algorithme modifié de Gauss-Newton [Baker and Matthews, 2004]. Le principal avantage de cette *composition inverse* est de transposer la majorité des calculs à l'initialisation, rendant l'algorithme plus rapide et temps réel. De plus, cette formulation du problème par rapport à celle de base des AAM permet d'obtenir un alignement plus précis.



(a) Active Appearance Models



(b) Constrained Local Models



(c) Régression de forme

FIGURE 3.4 – Diagrammes généraux des approches d'alignement d'un Modèle à Distribution de Points. Ces techniques diffèrent sur les critères suivants : les contraintes spatiales de forme, la prise en compte locale ou globale de l'apparence, une approche par optimisation ou reposant sur des apprentissages statistiques.
[1] [Zhu and Ramanan, 2012; Belhumeur et al., 2013] calculent la carte de réponse à l'initialisation et leur approche n'est pas itérative.

Des efforts ont été accomplis pour rendre cette méthode plus robuste en terme de généralisation de l'apparence. Ainsi, [Papandreou and Maragos, 2008] proposent d'adapter le modèle d'apparence de base (vers lequel les incréments sont calculés dans le cadre d'une *composition inverse*) afin de s'accommoder d'échantillons dont l'apparence est éloignée de celle présente dans la base. [Tzimiropoulos et al., 2013] proposent de renforcer le modèle d'apparence en utilisant les gradients orientés et montrent comment incorporer ce type de caractéristiques dans le *framework* des AAM. Récemment, [Antonakos et al., 2015] combinent la structure en arbre des *Pictorial Structures* avec une formulation d'optimisation de type *composition inverse*. La formulation basée sur un graphe renforce la qualité de l'alignement sur des modèles articulés tels que le corps humain. Malheureusement, les auteurs n'ont pas encore expérimenté leur méthode sur ce cas d'application. D'un autre côté, [Liu, 2007] entraîne une machine de boosting de type GentleBoost [Friedman et al., 1998] sur des caractéristiques de type pseudo-Haar pour discriminer les exemples dont l'alignement est correct et

ceux où il est incorrect. En phase d'alignement, les auteurs font évoluer les paramètres de forme en augmentant le score fourni par GentleBoost par une descente de gradient. Ces travaux sont ensuite déclinés pour s'appliquer au corps humain total [Liu et al., 2008] en utilisant comme caractéristiques des HOG locaux projetés sur des hyperplans obtenus par une analyse linéaire discriminante. Nous reviendrons sur cette technique Section 3.5 en y proposant des améliorations.

D'autres considérations ont été prises en compte pour pouvoir aligner un PDM sur une image. Les algorithmes de type *Constrained Local Models* ont été établis par [Cristinacce and Cootes, 2008]. Ils proposent de combiner des détecteurs de points entraînés de façon discriminante (à l'instar des *Pictorial Structures* pour les parties) et s'appuyant sur des caractéristiques locales et des contraintes d'un modèle global de forme. De cette manière, il est plus aisé de modéliser les changements complexes dans les images liés notamment à l'illumination par rapport à une considération globale de l'apparence.

Dans la même lignée, [Saragih et al., 2010] proposent de s'appuyer sur un modèle non paramétrique. Pour cela, ils construisent la carte de réponses de détecteurs des amers puis optimisent et régularisent le positionnement de ces derniers avec des fonctions de mises à jour déterminées par une méthode de type *Mean shift* [Carreira-Perpindn, 2007]. [Zhu and Ramanan, 2012] emploient des modèles de mélange et une structure en arbre pour décrire les points de la forme du visage afin de gérer l'élasticité des déformations. Pour améliorer la robustesse en terme d'occultation, de pose ou d'illumination, [Belhumeur et al., 2013] proposent de s'appuyer sur une inférence bayésienne combinant la sortie de détecteurs locaux avec un consensus de modèles non paramétriques globaux de forme appris sur les données d'entraînement. Les auteurs démontrent le fort potentiel de ce consensus à gérer un grand nombre de situations en introduisant une nouvelle base de visages en situation naturelle.

En parallèle de ces méthodes se sont développées des techniques basées sur la régression de forme qui consistent à estimer directement les coordonnées des points à partir des caractéristiques d'apparence. [Cristinacce and Cootes, 2007] entraînent pour chaque point deux fonctions de régression du déplacement local (en x et en y) vers le point cible grâce à GentleBoost. Les coordonnées des points sont mis à jour en même temps que les paramètres du modèle obtenu par l'ACP afin de respecter les contraintes de forme. [Dollár et al., 2010] incorporent à cette technique des caractéristiques indexées par rapport à la pose. Ainsi, à chaque itération, de nouveaux régresseurs faibles sont appris par rapport à des caractéristiques calculées sur la dernière estimation de la localisation des amers. Les caractéristiques utilisées sont les *control points* [Moutarde et al., 2008], qui sont équivalentes à une différence de pixels. Un avantage de cette approche est la répartition de l'information à apprendre sur l'ensemble des étapes du processus et, ainsi, la spécialisation de chaque régresseur sur différents ordres de grandeur d'échelle. Cela se traduit par un raffinement de l'alignement de type *grossier à fin* (ou en anglais *coarse-to-fine*).

Dû en partie aux expressions, certaines parties du visage peuvent présenter plus de déformations que d'autres. C'est, par exemple, le cas de la bouche par rapport aux yeux ou au nez. [Valstar et al., 2010] organisent l'alignement en identifiant, dans un premier temps, les points stables. Cela leur permet de définir des contraintes d'espace de recherche pour les points plus instables, grâce à l'utilisation de modèles graphiques probabilistes. Cette approche est similaire aux *Pictorial Structures* dans le sens où la partie stable peut être identifiée par rapport au torse humain. Néanmoins, l'inconvénient de cette approche réside lorsque la localisation de points stables est incorrecte, ce qui peut arriver dans le cas du visage lorsque les yeux sont occultés par des lunettes, par exemple. Pour pouvoir gérer cette situation, [Burgos-Artizzu et al., 2013] choisissent d'expliciter les zones dans l'image potentiellement occluses en ajoutant un état visible ou non visible aux amers.

[Cao et al., 2013] introduisent la régression de forme sur une cascade à deux niveaux et obtiennent de très bons résultats couplés à un algorithme rapide. Le premier niveau de la cascade met

à jour la forme tandis que le second niveau se base sur la même forme en entrée pour calculer les caractéristiques indexées sur la forme. Cela permet notamment de stabiliser le processus de mise à jour et de pallier au manque de pouvoir d'alignement des régresseurs faibles jusqu'alors utilisés. Nous reviendrons plus en détail sur cet algorithme dans la seconde implémentation d'un système d'alignement que nous proposons Section 3.6. D'autres travaux ont porté sur une façon d'obtenir des régresseurs performants. [Cootes et al., 2012] expérimentent les régresseurs de type Forêts Aléatoires [Breiman, 2001] et obtiennent de très bons résultats d'alignement en les couplant dans le *framework* des *Constrained Local Models*. [Xiong and De la Torre, 2013] proposent une méthode de descente supervisée qui est une cascade de régressions linéaires entre les coordonnées de forme et des caractéristiques performantes de type SIFT. Leur contribution principale est de mettre en relation la fonction d'optimisation d'alignement utilisée classiquement pour les modèles paramétriques avec une méthode basée sur la régression. [Qu et al., 2015] renforcent cette approche en améliorant la robustesse au bruit lors de l'apprentissage des régresseurs, en choisissant une distance plus adaptée pour les descripteurs SIFT et en prenant en compte l'angle de la forme pour le calcul des caractéristiques. [Sun et al., 2013] s'appuient quant à eux sur le *Deep Learning* pour construire une cascade de *ConvNets* en charge de la régression de localisation des points. Des efforts ont également été faits pour rendre les algorithmes de cascade de régressions plus performants en terme de temps de calcul. Des vitesses très élevées sont atteintes dans les travaux [Ren et al., 2014] (de l'ordre de 300 FPS sur smartphone) qui s'appuient sur des caractéristiques binaires prises localement autour des points et extrêmement rapides à calculer. [Kazemi and Sullivan, 2014] obtiennent également de très bonnes performances de temps de calcul en utilisant un ensemble d'arbres régresseurs s'appuyant sur des différences de pixels sélectionnés préférentiellement autour du point d'intérêt.

Dans cette thèse, nous cherchons à porter ces méthodes dans le cadre d'application du corps humain. Certaines caractéristiques utilisées pour le visage sont inadaptées dans le cadre du corps humain, notamment celles s'appuyant directement sur les valeurs des pixels. En effet, la variabilité d'apparence du corps humain est beaucoup plus importante que celle du visage, principalement à cause des vêtements et accessoires portés. Il est nécessaire aussi d'adapter nos approches afin d'être capable d'appréhender les nombreuses poses que peuvent adopter le corps humain. Nous proposons deux nouveaux systèmes. Le premier est basé sur un modèle paramétrique de forme et est une extension des travaux proposés par [Liu et al., 2008], en renforçant le pouvoir des régresseurs faibles par une conception de type classement (*ranking*) en apprentissage. Le second système que nous proposons s'appuie sur une approche à base de cascade de régressions et se focalise sur une gestion efficace des nombreuses déformations possibles que peut adopter le modèle pour le corps humain.

Algorithme	Caractéristiques & Apparence	Techniques pour l'alignement
[Cootes et al., 1995]	Bords proches des points	Fonction d'alignement local + contraintes de forme ACP
[Cootes et al., 2001]	Pixels	Optimisation non-linéaire. Régression entre l'image résiduelle et les paramètres de formes.
[Matthews and Baker, 2004]	Pixels	Composition inverse (optimisation non-linéaire avec Gauss-Newton).
[Liu, 2007]	Pseudo-Haar	Maximisation du score d'un *GentleBoost* par descente de gradient
[Liu et al., 2008]	HOG locaux	Maximisation du score d'un *GentleBoost* par descente de gradient
[Papandreou and Maragos, 2008]	Pixels	Composition inverse + Adaptation du modèle d'apparence de base
[Tzimiropoulos et al., 2013]	Gradients orientés	Composition inverse
[Antonakos et al., 2015]	Pixels	Composition inverse + *Pictorial Structures*
[Cristinacce and Cootes, 2008]	Gabarits locaux	Détecteurs locaux + contraintes de forme ACP
[Saragih et al., 2010]	Pixels (patchs locaux)	Détecteurs locaux + contraintes de forme par un modèle non paramétrique
[Zhu and Ramanan, 2012]	HOG pour chaque partie	Modèles de mélange + structures en arbres
[Belhumeur et al., 2013]	SIFT	Détecteurs locaux + Consensus de modèles non paramétriques
[Cristinacce and Cootes, 2007]	Pseudo-Haar	Régression de coordonnées pour chaque point par GentleBoost + contraintes de forme ACP
[Dollár et al., 2010]	Différences de pixels (indexées à la forme)	Cascade de régressions
[Valstar et al., 2010]	Pseudo-Haar	Contraintes d'alignement par rapport à des points considérés stables du PDM
[Cootes et al., 2012]	Pseudo-Haar	*Random Forest* + Constrained Local Models
[Burgos-Artizzu et al., 2013]	Différences de pixels	Occlusions gérées par un statut de visibilité pour chaque point
[Cao et al., 2013]	Différences de pixels	Double cascade de régressions
[Xiong and De la Torre, 2013]	SIFT	Cascade de régressions
[Sun et al., 2013]	Pixels	*Convolutional Neural Network*
[Ren et al., 2014]	Caractéristiques binaires induites par des *Random Forests* basées sur différences de pixels	Apprentissage local et indépendant des caractéristiques + régression globale
[Kazemi and Sullivan, 2014]	Différences de pixels	Arbres régresseurs appris par *gradient boosting*
[Qu et al., 2015]	SIFT	Cascade de régressions

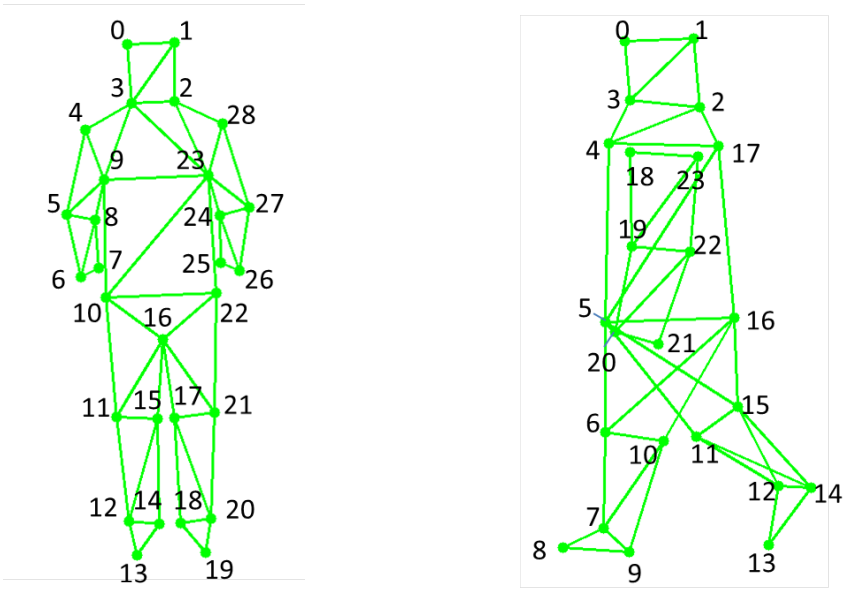
TABLE 3.2 – AAM/ASM *Constrained Local Models* Régression de forme
Tableau récapitulatif des méthodes d'alignement de Modèles à Distribution de Points. La tendance va à l'abandon d'un modèle paramétrique résultant d'une Analyse en Composante Principale au profit de modèles non paramétriques.

3.4 Pré-requis à l'alignement d'un PDM

3.4.1 Définition du modèle

Avant de déterminer le modèle à distribution que l'on souhaite appliquer au corps humain, nous rappelons une définition de ce type de modèle donnée par [Cootes et al., 1995] qui listent les spécificités auxquelles doivent répondre les amers :

- Ils marquent une partie de l'objet ayant une signification identifiée. Pour le corps humain, il peut s'agir d'une partie anatomique définie, telle que les épaules.
- Ils marquent une partie de l'objet n'ayant pas de signification particulière mais possédant un emplacement particulier selon l'orientation par rapport à la caméra. Il peut s'agir de points placés aux extrémités, ou à des intersections de l'objet.
- Ils peuvent résulter d'une interpolation de points répondant aux deux conditions précédentes (point pour compléter un contour par exemple).



(a) Face/Dos

(b) Profil gauche/droite

FIGURE 3.5 – Modèles à Distribution de Points que nous adoptons pour décrire le corps humain. Le maillage est représenté ici afin de faciliter la visualisation des points.

Nous proposons de nous baser sur la représentation proposée dans les travaux de [Liu et al., 2008] et représentée Figure 3.5a. Ce PDM se compose de 29 points que nous indexerons de la façon suivante :

0. Tête - coin haut gauche
1. Tête - coin haut droite
2. Tête - coin bas droite
3. Tête - coin bas gauche
4. Épaule gauche
5. Coude gauche - extérieur
6. Poignet gauche - extérieur
7. Poignet gauche - intérieur
8. Coude gauche - intérieur
9. Aisselle gauche
10. Bassin gauche
11. Genou gauche - extérieur
12. Cheville gauche - extérieure
13. Pied gauche
14. Cheville gauche - intérieure
15. Genou gauche - intérieur

16. Bassin - centre
17. Genou droit - intérieur
18. Cheville droite - intérieure
19. Pied droit
20. Cheville droite - extérieure
21. Genou droit - extérieur
22. Bassin droit - extérieur
23. Aisselle droite
24. Coude droit - intérieur
25. Poignet droit - intérieur
26. Poignet droit - extérieur
27. Coude droit - extérieur
28. Épaule droite

Les points définis peuvent être rattachés aux points d'articulation du squelette (épaule-aisselle/coudes/bassin/genoux) ou représentent une extrémité du corps en vue d'obtenir une segmentation du corps entier (poignets/coins de la tête et pieds). Nous avons la possibilité d'utiliser le même modèle pour des vues de face et de dos de la personne. La dénomination gauche/droite n'est pas relative à la personne mais à la disposition dans l'image. On remarquera que les mains n'ont pas été prises en compte. Ce choix a été décidé par rapport au cadre d'application de cette thèse. En effet, dans un contexte urbain, les mains peuvent fréquemment être occultées, notamment par le port d'effets personnels, ou lorsque la personne les place dans les poches de ses habits. En cela, elles diffèrent des pieds qui restent pour la plupart du temps visibles. De plus, la plupart des travaux de pose sur les modèles squelettes ou *Pictorial Structures* regroupent les mains avec les avant-bras [Toshev and Szegedy, 2013; Pishchulin et al., 2013]. Ce regroupement obtient les moins bons résultats en terme de localisation. En conséquence, les mains représentent les parties du corps les plus difficiles à aligner à cause de la propagation de l'erreur d'alignement des bras.

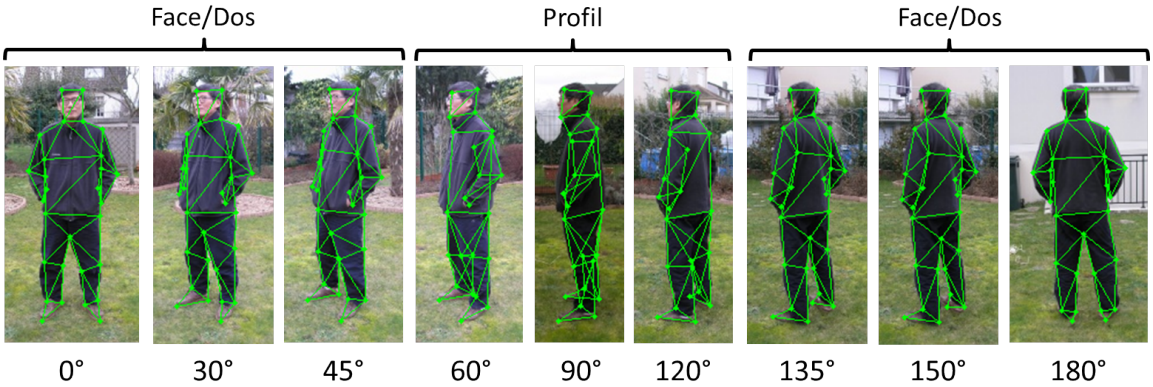


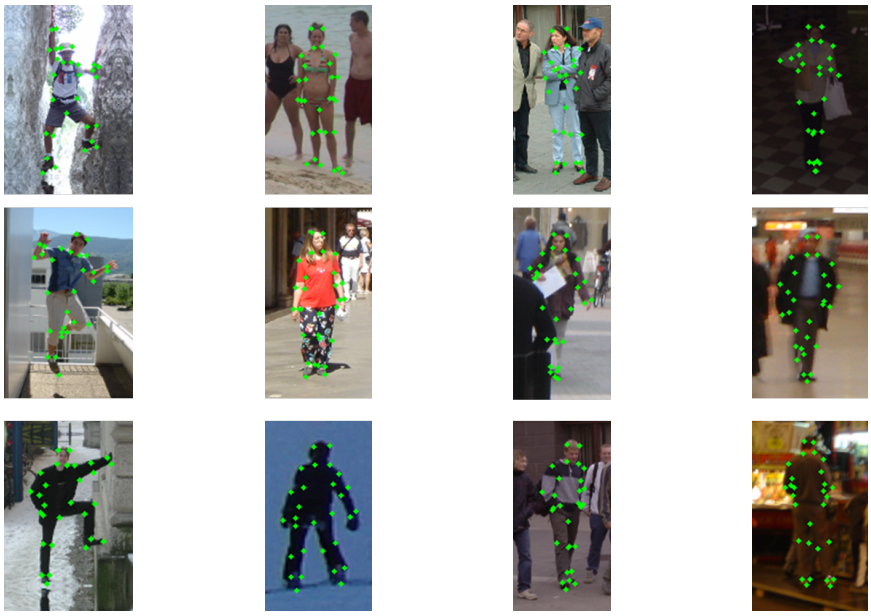
FIGURE 3.6 – Visualisation des deux Modèles à Distribution de Points proposés par rapport à l'orientation de la personne.

Ce modèle présente des incohérences pour une configuration de profil. Tout d'abord, la notion extérieure/intérieure n'est plus respectée et est remplacée par une notion avant/arrière. De plus, en conservant la même dénomination, le déplacement de certains points peut être très important dans l'image (épaules et jambes notamment). Cela complexifie fortement la tâche d'alignement du fait de l'éloignement des formes profil/face dans l'espace des déformations. Enfin, la plupart du temps, le profil occultera une grande partie du bras situé à l'opposé de la caméra. C'est pourquoi nous proposons d'utiliser un deuxième modèle spécifique au profil sur une orientation appartenant à l'intervalle estimé entre 60° et 120° plus ou moins 180°. Cette approche est similaire au modèle *CardBoard* proposé dans les travaux de [Ju et al., 1996]. Ce modèle est représenté Figure 3.5b et comporte les 24 points suivants :

0. Tête - coin haut gauche
1. Tête - coin haut droite
2. Tête - coin bas droite
3. Tête - coin bas gauche
4. Haut du corps - avant
5. Bassin - avant
6. Genou jambe premier plan - avant
7. Cheville jambe premier plan - avant
8. Pied jambe premier plan
9. Cheville jambe premier plan - arrière
10. Genou jambe premier plan - arrière
11. Genou jambe second plan - avant
12. Cheville jambe second plan - avant
13. Pied jambe second plan
14. Cheville jambe second plan - arrière
15. Genou jambe second plan - arrière
16. Bassin - arrière
17. Haut du corps
18. Épaule - avant
19. Coude - avant
20. Poignet - avant
21. Poignet - arrière
22. Coude - arrière
23. Épaule - arrière

Les points avant et arrière peuvent être intervertis suivant le profil adopté à l'image. Il est préférable d'accorder aux points les deux significations possibles afin de leur éviter un déplacement horizontal important dans l'image. Cependant, la distinction peut être établie dès l'identification du profil gauche ou droite de la personne.

3.4.2 Annotation d'une nouvelle Dataset



(a) Pose (b) Variabilité d'apparence (c) Occlusions partielles (d) Illumination et flou

FIGURE 3.7 – Illustrations d'échantillons présentant des défis pour l'alignement d'un PDM sur les personnes.

Comme spécifié auparavant, l'utilisation d'un Modèle à Distribution de Points nécessite une base d'apprentissage statistique pour définir des contraintes de forme et un modèle d'apparence.

Puisque la base utilisée par [Liu et al., 2008] n'est pas mise à disposition, nous proposons d'introduire une nouvelle base pour répondre à nos besoins. Nous avons décidé d'utiliser les images de la base piétonne INRIA [Dalal and Triggs, 2005] exploitée pour le module de détection. En effet, les piétons apparaissent dans des situations naturelles urbaines, répondant aux conditions de ré-identification établies pour cette thèse. Puisque deux modèles de points sont avancés, il est nécessaire d'annoter deux ensembles de personnes distincts pour la vue face/dos et profil. Néanmoins, par manque de temps, nous concentrerons nos travaux sur une vue de face et de dos. Une extension des travaux de cette thèse portera sur l'annotation pour la condition de profil.

Nous avons annoté au total 408 personnes pour la base d'apprentissage et 129 pour la base de test avec les images dédiées respectivement à l'apprentissage et à la validation du système de détection. Leur sélection a été faite de sorte que le corps de la personne soit en principe entièrement contenu dans l'image (pas de sortie d'images) avec une taille acceptable (hauteur minimale annotée : $\sim$ 60 pixels). L'outil d'annotation utilisé est celui fourni par [Matthews and Baker, 2004] qui a la fonctionnalité de placer les points à une précision subpixellique (grâce à un zoom). Afin de disposer de points corrects, les annotations ont été réalisées une première fois puis reprises et corrigées dans un second temps. Pour se placer dans des conditions de moyennes/faibles résolutions, les images sont normalisées de façon à ce que la taille du PDM atteigne au maximum soit 64 pixels de largeur, soit 128 de hauteur. Pour enrichir cette base, nous proposons de générer les images en les retournant horizontalement, portant le nombre d'échantillons d'entraînement à 816 et de validation à 258.

Nous nommons cette base InriaLHB pour Inria Labeled Human Body et la rendons disponible à la communauté scientifique à l'adresse web suivante : `http://caor-mines-paristech.fr/en/2015/09/inrialhb/`. Des exemples d'échantillons présentant des défis d'alignement sont donnés Figure 3.7.

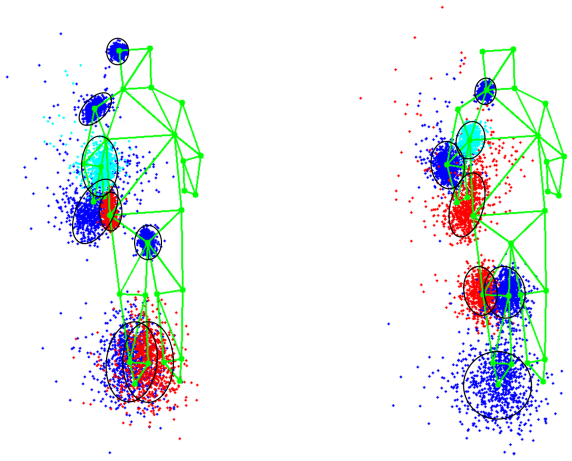


FIGURE 3.8 – Visualisation de la distribution statistique des points de la dataset proposée. La représentation est donnée en deux fois pour plus de clarté. Les points à la droite du modèle ne sont pas représentés mais sont distribués de la même manière que ceux de gauche puisque la dataset est étendue avec une symétrie horizontale.

Il est possible de faire une étude statistique de cette base afin d'analyser la différente répartition des points en fonction des parties du corps. Pour parvenir à cela, nous alignons dans un premier temps toutes les formes avec une analyse procustéenne que nous décrivons ultérieurement Algorithme 3. Les points sont ensuite placés par rapport à la forme moyenne obtenue de cette analyse. Les amers instables dans l'image sont ceux où leur distribution est la plus dispersée. Ils correspondent aux membres capables d'adopter des postures complexes. Ce fait se confirme par la

distribution des points illustrée Figure 3.8 qui montre que les points relatifs aux bras et aux jambes sont ceux présentant le plus de déformations possibles.

3.5 Modèle d'apparence par boosting sur une forme paramétrique

Soit un Modèle à Distribution de Points (que nous désignerons aussi par forme) défini par $S = [x_1, y_1, x_2, y_2, ..., x_K, y_K]^T$ un ensemble de K amers décrits par un système de coordonnées à deux dimensions $\{x_k, y_k\}$. L'objectif de l'alignement d'un PDM est d'estimer pour un échantillon n le PDM S_n au plus proche de celui de vérité terrain $\hat{S_n}$, ce qui consiste à minimiser la norme :

$$||\hat{S_n} - S_n||_2 \tag{3.1}$$

Nous exprimerons la base annotée D de la façon suivante : $D = \{\{I_1, \hat{S_1}\}, \{I_2, \hat{S_2}\}, ..., \{I_N, \hat{S_N}\}\}$, I_n représentant l'image et $\hat{S_N}$ le PDM vérité terrain associé.

3.5.1 Modélisation paramétrique

Nous proposons dans un premier temps d'évaluer un système basé sur un modèle de forme décrit paramétriquement. L'ACP est l'algorithme le plus couramment employé dans la littérature pour exprimer la forme de façon paramétrique :

$$S(p) = \bar{S_0} + \sum_{i=1}^{T} p_i \bar{S_i} \tag{3.2}$$

où $\bar{S_0}$ représente la forme moyenne, $\bar{S_i}$ le i-ème mode ou axe principal, et p_i le paramètre du mode associé. Nous écrirons dorénavant p pour désigner le vecteur de paramètres formé par p_i. Le nombre de mode T est déterminé par la proportion de variance que l'on souhaite conserver de la base d'apprentissage. Cette proportion est calculée par rapport à la somme cumulée des plus grandes valeurs propres (de la matrice de covariance de la base d'apprentissage) sur la somme totale des valeurs propres. Elle est définie élevée (aux alentours de 98%) de sorte à pouvoir conserver la possibilité de modéliser une grande partie des poses adoptées dans la base. Sous cette forme Équation 3.2, le problème consiste à approcher les paramètres p des composants de la vérité terrain projetée sur la base formée par $\bar{S_0}$ et les $\bar{S_i}$.

Pour pouvoir opérer l'ACP uniquement sur les déformations locales, il est nécessaire de superposer les formes de la base. La méthode classique est de recourir à une analyse procustéenne généralisée [Gower, 1975] afin de supprimer les transformations globales de forme telles que les différences de translation, rotation et échelle entre chaque PDM. Nous proposons une légère modification de cette analyse (Algorithme 3). Pour prendre en compte la grande amplitude du champ de déformation des bras et des jambes, nous calculons les paramètres de transformation uniquement sur les parties stables dans l'image : tête, torse et épaules (soit les indices : $\{0, 1, 2, 3, 4, 9, 10, 16, 22, 23, 28\}$). L'intérêt de procéder ainsi est illustré Figure 3.9. Pour 816 échantillons et 98% de variance conservée, nous obtenons 29 modes au total. Nous donnons une représentation des premiers modes de forme calculés sur la base InriaLHB Figure 3.10a.

Algorithme 3: Analyse procustéenne généralisée [Gower, 1975]

Entrée: $\{S_1, S_2, ..., S_N\}$
Sortie : $\{S_1, S_2, ..., S_N\}$ alignés et moyenne $\bar{S}_0$
Initialisation : Choix arbitraire d'un PDM de référence $S_{0,0}$
$t = 0$

répéter
- $t \leftarrow t + 1$
- **pour** $n = 1$ **à** N **faire**
  - Calcul des paramètres de transformation par similitude [Cootes et al., 2004] de S_n vers $S_{0,t-1}$ sur un sous-ensemble de points stables
  - Mise à jour de S_n avec ces paramètres sur l'ensemble des points
- $S_{0,t} \leftarrow \frac{\sum_{n=1}^{N} S_n}{N}$

jusqu'à $||S_{0,t} - S_{0,t-1}|| < \varepsilon$ `// Convergence de la moyenne`
$\bar{S}_0 \leftarrow S_{0,t}$

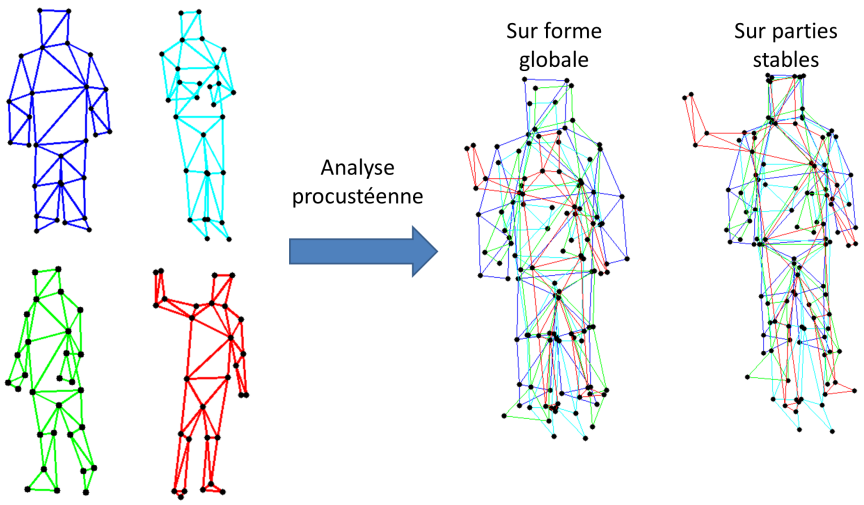
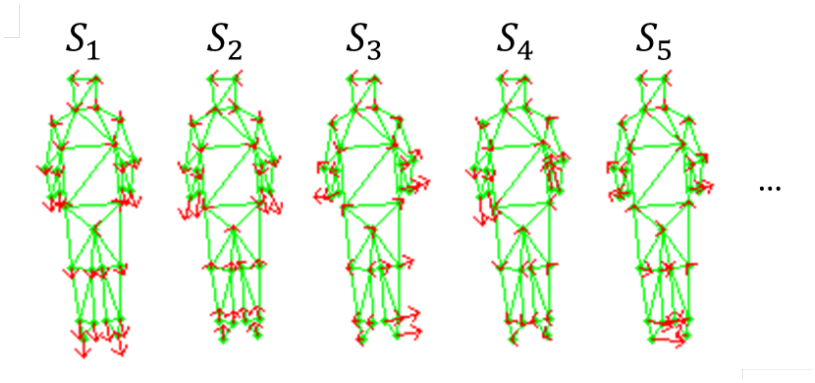


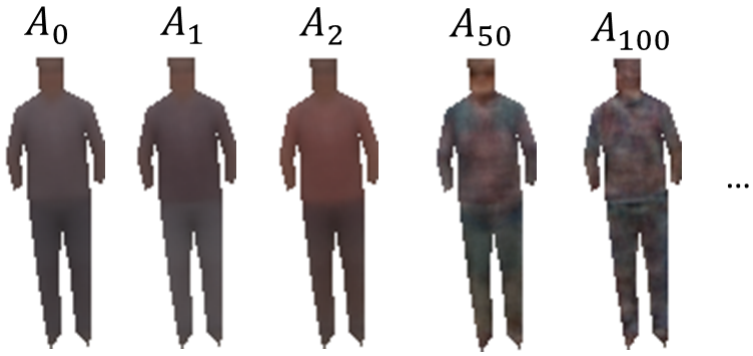
FIGURE 3.9 – Illustration de l'analyse procustéenne sur 4 PDM. L'alignement par rapport à des points stables permet une mise à l'échelle correcte, notamment de la forme rouge.

Il convient à présent de construire un modèle d'apparence pour pouvoir guider ce modèle de forme à trouver les paramètres corrects. Avant de considérer d'autres algorithmes plus adaptés, nous avons jugé intéressant d'expérimenter le potentiel de l'algorithme AAM dans ce cas d'étude. Comme pour le modèle paramétrique de forme, il convient de calculer un modèle paramétrique d'apparence.

Les apparences sont projetées sur une forme commune (en l'occurrence S_0) grâce à une transformation affine par parties (illustrée Figure 3.12a) en s'appuyant sur le maillage représenté Figure 3.5a. La transformation affine par parties utilise les triangles en tant que base et peut être calculée efficacement grâce à l'approche présentée par [Matthews and Baker, 2004]. Cette méthode, que nous avons implémentée, pré-calcule un ensemble de paramètres pour chaque paire de triangles entre deux formes. Ainsi, en faisant intervenir ces paramètres, le calcul des coordonnées transformées d'un pixel consiste en une fonction affine et est très rapide. Nous invitons le lecteur à consulter les



(a) Modes de forme



(b) Modes d'apparence

FIGURE 3.10 – Illustration des modes de forme et d'apparence provenant d'une analyse en composante principale sur InriaLHB.

travaux de [Matthews and Baker, 2004] pour de plus amples détails. Cette transformation, introduite précédemment est noté : $W(x, y, p) = (W_x, W_y)$, (x, y) étant un pixel contenu dans le maillage de $\bar{S}_0$.

Pour 98% de variance conservée, nous obtenons 288 modes d'apparence, nombre bien plus conséquent par rapport à celui des modes de formes. Cela provient principalement de la grande diversité d'apparence contenue par les vêtements. Un sous-échantillon de ces modes d'apparence est représenté Figure 3.10b montrant la difficulté de modéliser l'apparence au niveau pixellique. Ce fait se confirme par notre expérimentation de l'algorithme d'alignement par composition inverse de [Matthews and Baker, 2004] qui délivre des résultats insatisfaisants. Soit l'algorithme ne parvient pas à converger vers le corps humain et la forme évolue de plus en plus sur le décor jusqu'à sortir de l'image, soit la convergence est atteinte, mais l'alignement reste éloigné.

C'est pourquoi nous proposons de modéliser autrement l'apparence complexe du corps humain en recourant à une machine de boosting.

3.5.2 Modèle d'apparence par GentleBoost

Nous nous sommes basés sur l'approche proposée par [Liu et al., 2008] qui construisent un modèle d'apparence avec une machine de type *GentleBoost* [Friedman et al., 1998]. La principale différence de cet algorithme avec *AdaBoost* (présenté Algorithme 1) est la fonction de sélection du classifieur

faible :

$$h_t = \underset{h_k}{\operatorname{argmin}} \sum_{n=1}^{N} w_n (h_k(I_n) - y_n)^2 \tag{3.3}$$

Cette formulation permet à h_t de fournir un score continu entre 1 et -1. Le score global de *GentleBoost* peut s'écrire de la façon suivante :

$$H(I,p) = \sum_{t=0}^{T} h_t(I,p) \tag{3.4}$$

pour T régresseurs faibles. Il donne une indication sur l'état d'alignement de la forme. Ainsi, pour aligner la forme à l'image, l'algorithme consiste à maximiser ce score par une descente de gradient par rapport aux paramètres p : $\frac{dH}{dp}$. Les paramètres p ainsi calculés sont mis à jour itérativement de façon additive :

$$p = p + \lambda \frac{dH}{dp} \tag{3.5}$$

avec λ une constante.

Concernant l'entraînement de *GentleBoost*, [Liu et al., 2008] associent à chaque échantillon de la base D un label y_n positif $y_n = 1$ lorsque la forme est alignée et négatif $y_n = -1$ lorsque la forme est désalignée par une perturbation des paramètres de la vérité terrain. Structurer la base de cette manière permet d'apprendre si l'alignement est correct ou non. Or, cela s'avère inefficace à la phase de test car le régresseur n'a pas été entraîné dans le but d'améliorer progressivement l'alignement. C'est pourquoi, nous proposons de guider l'alignement des régresseurs faibles grâce à une structuration spécifique de la base d'apprentissage.

3.5.2.1 Procédure d'apprentissage

Dans un premier temps, nous générons de nouveaux échantillons négatifs en perturbant progressivement et de façon linéaire les paramètres de $\hat{S}$ vers ceux d'une forme cible. Dans un second temps, nous nous basons sur la structuration proposée par [Wu and Nevatia, 2008], qui permet d'étendre *GentleBoost* à un problème de classement (ou *ranking* en anglais). Cette structuration consiste à former des paires d'échantillons consécutifs (à un rang de perturbation donné) et à leur attribuer un label positif dans le cas où elles sont arrangées vers la vérité terrain et un label négatif dans le sens contraire. Introduire cette notion de classement permet au modèle d'apprendre, non pas la justesse de l'alignement courant, mais comment se rapprocher de la forme correcte.

Nous notons D' la base restructurée contenant les échantillons remaniés :

$$D' = \begin{cases} \left[\{(I;p^{m+1});(I;p^{m})\}, y_+ = 1\right] \\ \\ \left[\{(I;p^{m});(I;p^{m+1})\}, y_- = -1\right] \end{cases} \tag{3.6}$$

en considérant que p^0 correspond aux paramètres de la forme vérité terrain. Le nombre d'échantillons de D' est $N \times (M-1) \times 2$ paires avec N le nombre d'images initiales et M le nombre de perturbations entre la vérité terrain et la forme cible. Une illustration de cette restructuration est donnée Figure 3.11.

Ainsi, la fonction de sélection du meilleur candidat donnée par l'Équation 3.3 devient :

$$h_t = \underset{h_k}{\operatorname{argmin}} \sum_{n=1}^{N} \sum_{m=1}^{M-1} w_{+nm}(y_{+nm} - f_+(n,m))^2 + w_{-nm}(y_{-nm} - f_-(n,m))^2 \tag{3.7}$$

$$\text{avec} \quad \begin{aligned} f_+(n,m) &= h_k(I_n; p_n^m) - h_k(I_n; p_n^{m+1}) \\ f_-(n,m) &= h_k(I_n; p_n^{m+1}) - h_k(I_n; p_n^m) \end{aligned}$$

w_{+nm} et w_{-nm} correspondant respectivement aux poids de la paire positive et négative de l'image n et de la perturbation classée m. Autrement formulé, le meilleur candidat h_k sera sélectionné sur sa capacité à fournir les meilleurs améliorations pondérées de score entre deux perturbations consécutives.

Nous définissons comme forme cible des perturbations $\bar{S}_0$. En conséquence, cette forme $\bar{S}_0$ sera utilisée comme initialisation en phase de test pour l'alignement. Ce processus présente l'avantage d'alléger la charge d'apprentissage du régresseur faible en réduisant l'espace d'évolution des paramètres de forme par rapport à une forme perturbée aléatoire. Pour favoriser la création d'un maximum du score sur la vérité terrain et éviter un dépassement non contrôlé, nous poursuivons la perturbation des paramètres après avoir atteint la vérité terrain sur quelques échantillons.

Par ailleurs, nous couplons l'approche de [Cristinacce and Cootes, 2007] à cette structuration, consistant à entraîner les régresseurs faibles sur les images labélisées par rapport à un classement progressif de perturbation m depuis la forme vérité terrain. Il est à noter que nous n'utilisons pas cette labélisation pour la sélection du candidat Équation 3.7 car le but premier n'est pas de donner un score défini à un avancement de l'alignement, mais d'apprendre comment améliorer l'alignement courant.

3.5.2.2 Caractéristiques

Il convient à présent de définir les régresseurs faibles h_k ainsi que les caractéristiques visuelles exploitées f_k. Nous utilisons les mêmes caractéristiques que [Liu et al., 2008], à savoir des blocs de 2x2 cellules de gradients orientés. Ces HOG locaux décrivent une localisation précise du fait de leur configuration en grille et s'avèrent plus robustes que l'intensité des pixels pour prendre en compte les bords d'une personne, qui sont déterminants pour l'alignement. Ces caractéristiques sont construites par la concaténation des histogrammes de gradients orientés, calculés via les *channels* de gradients présentés Section 2.4.1.1. Cependant, au lieu de normaliser directement le *channel* de gradient orienté par une zone centrée sur le pixel, nous utilisons la normalisation par bloc *L1-sqrt*, telle que présentée dans les travaux de [Dalal and Triggs, 2005]. Il est également envisageable d'employer d'autres caractéristiques comprenant des informations plus riches comme par exemple les caractéristiques SIFT.

Si l'on souhaite adapter le modèle d'apparence par rapport à la forme courante, il est nécessaire de définir une façon d'indexer à l'image ces caractéristique locales. Comme pour les travaux de [Cristinacce and Cootes, 2007; Liu et al., 2008; Valstar et al., 2010], nous proposons de les localiser grâce à la transformation affine par parties présentée précédemment. Puisque la caractéristique est référencée par rapport à $\bar{S}_0$, sa valeur peut s'écrire de la façon suivante : $f(I, W(f_x, f_y, p), f_w, f_h)$ où $W(f_x, f_y, p)$ représente ses coordonnées dans la forme courante (avec sa position référencée (f_x, f_y) et sa hauteur, largeur constante (f_w, f_h).

Cette indexation spatiale est plus coûteuse en calcul que celle utilisée par [Cao et al., 2013] qui s'appuient sur des coordonnées relatives par rapport aux amers les plus proches. Néanmoins, elle conserve sa cohérence par rapport à un modèle articulé comme le corps humain, comme illustrée par les Figures 3.12. Une limitation de cette indexation est que la caractéristique est automatiquement

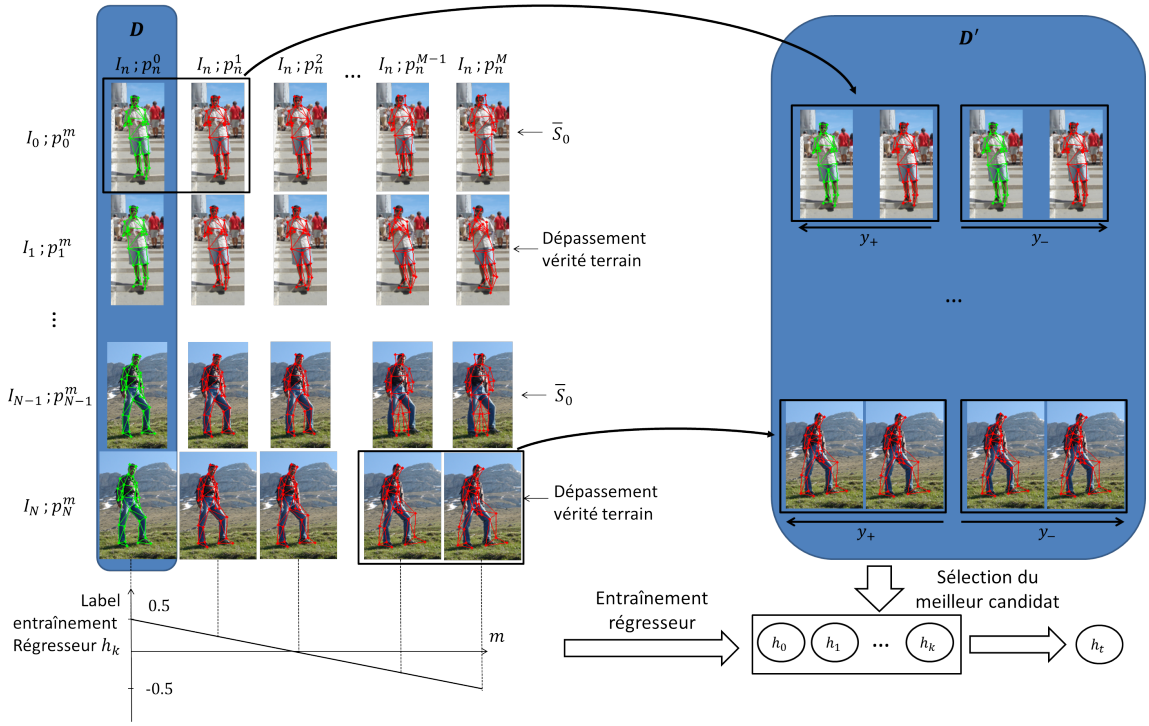


FIGURE 3.11 – Schéma illustrant la structuration de la base d'apprentissage pour entraîner *GentleBoost* sur une notion de classement.

rattachée à un triangle, et par conséquent forcément contenue à l'intérieur du maillage formé par le PDM. Nous proposerons, dans le cadre de notre second système d'alignement, d'étendre ce maillage afin d'avoir la possibilité de placer des caractéristiques en dehors de la forme courante.

(a) (b)

FIGURE 3.12 – Méthodes d'indexation à l'image des caractéristiques locales (a) Transformation affine par parties (b) Coordonnées relatives sur un référentiel non orienté de [Cao et al., 2013] présentant des incohérences spatiales pour un modèle articulé.

Par ailleurs pour améliorer la robustesse du système vis-à-vis de la pose de la personne, nous adaptons l'orientation de la caractéristique en la référençant à la forme courante. Nous cherchons à homogénéiser les valeurs que peuvent prendre les gradients face à des situations similaires relatives à la forme. Par exemple, le cas où une personne place ses bras le long du corps, le gradient possède une orientation horizontale. Si la personne lève ce bras, l'orientation de gradient devient verticale. L'objectif est d'adapter la structure de la caractéristique par rapport à l'orientation du triangle à laquelle elle appartient. Pour cela, nous proposons un calcul simple de l'orientation reposant sur l'angle formé par le centre de gravité du triangle. Soit les coordonnées des vertices du triangle j :

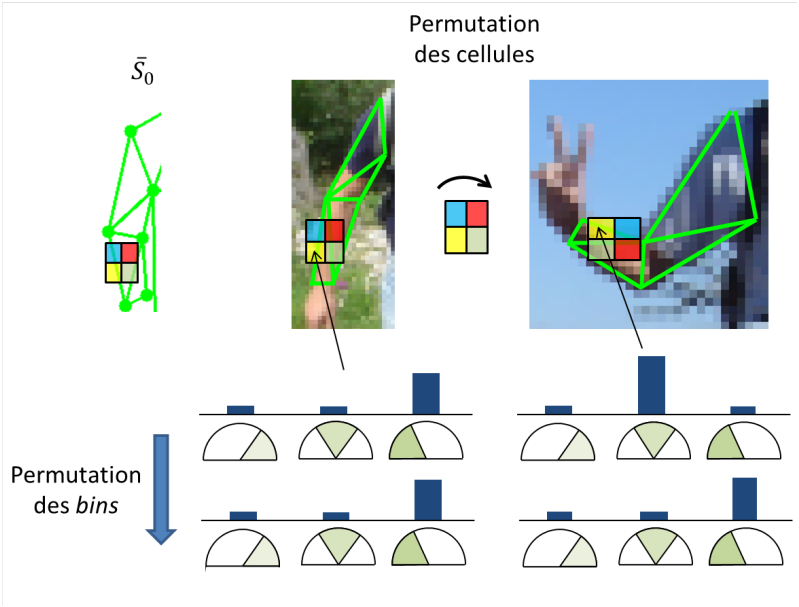


FIGURE 3.13 – Adaptation de l'orientation de la caractéristique locale HOG par rapport à la référence $\bar{S}_0$.

$(x_{o,j}, y_{o,j}; x_{1,j}, y_{1,j}; x_{2,j}, y_{2,j})$, nous calculons l'angle de la manière suivante :

$$\theta_j(p) = tan^{-1}\left(\frac{(y_{o,j}(p) + y_{1,j}(p) + y_{2,j}(p))/3 - y_{o,j}(p)}{(x_{o,j}(p) + x_{1,j}(p) + x_{2,j}(p))/3 - x_{o,j}(p)}\right) \tag{3.8}$$

Ensuite, l'orientation de la caractéristique, associée à celle du triangle, est obtenue par la différence entre l'angle calculé du PDM courant et $\bar{S}_0$. Nous permutons les classes d'orientation de l'histogramme par rapport à cet angle. De même, nous permutons les cellules du bloc comme illustré par la Figure 3.13 à chaque fois que l'orientation atteint : $(\theta_j(p=0) - \theta_j(p) \pm \frac{\pi}{4}) \pmod{\frac{\pi}{2}} = 0$

3.5.2.3 Régresseurs faibles

Les caractéristiques ayant été présentées, il s'agit de définir le type de régresseur qui va les utiliser. *GentleBoost* se compose de régresseurs capables de fournir un score compris entre -1 et 1. Dans le système original [Liu et al., 2008], les HOG sont projetés sur un hyperplan appris par une Analyse Linéaire Discriminante pondérée (en anglais *Weighted Linear Discriminant Analysis* ou WLDA), comparés à un seuil et normalisés avec arc tangente pour être ramenés entre -1 et 1. Deux inconvénients se dégagent de cette technique, le premier est que l'Analyse Discriminante Linéaire s'appuie sur une notion de classes, ce qui ne convient pas à la procédure d'apprentissage que nous avons avancée. Deuxièmement, leur formulation pour ramener le score entre -1 et 1 sous-entend que la valeur projetée de la caractéristique suive une loi en arc tangente, ce qui est faux.

Nous proposons de remplacer cette méthode par des réseaux de neurones artificiels, le perceptron multicouche [Rumelhart et al., 1986], prenant en entrée les valeurs de cette caractéristique. Puisque nous résolvons l'alignement par un algorithme du gradient, le modèle mathématique du régresseur doit être dérivable. Pour satisfaire cette exigence, nous choisissons comme fonction d'activation la sigmoïde symétrique, dérivable sur $\mathbb{R}$: $\sigma(x) = \frac{1-e^{-x}}{1+e^{-x}}$.

En s'appuyant sur la notation d'un perceptron de la Figure 3.14, nous écrivons sa sortie dans

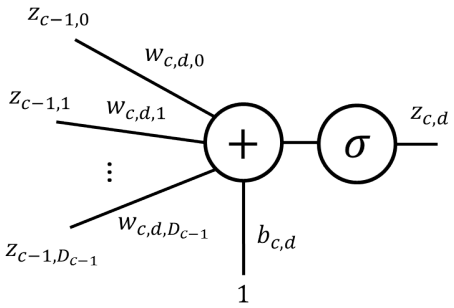


FIGURE 3.14 – Perceptron et notation adoptée.

un réseau multicouche comme étant égale à :

$$z_{c,d} = \sigma\left(\left(\sum_{n=0}^{D_{c-1}} w_{c,d,n}\, z_{c-1,n}\right) + b_{c,d}\right)_{c=1,\ldots,L} \tag{3.9}$$

$$z_{0,d} = f^{(d)}(I; W(f_x, f_y, p), f_w, f_h)$$

où $z_{c,d}$ correspond à la sortie du d-ième neurone de la c-ième couche, sachant que C représente le nombre de couches ($C >= 1$ car le réseau nécessite au moins une couche d'entrée et de sortie), D_c est le nombre de neurones contenus dans la c-ième couche, $w_{c,d,n}$ et $b_{c,d}$ sont respectivement les termes de poids et de biais. Enfin, $f^{(d)}$ correspond à la dimension d de la caractéristique HOG locale utilisée en entrée. Nous entraînons ce faible perceptron multicouche de sorte qu'il donne le meilleur score pour la vérité terrain et un score dégressif suivant l'intensité de perturbation de la forme. En s'appuyant sur cette cette notation, le score du *GentleBoost* donné Équation 3.4 peut s'écrire :

$$H(I,p) = \sum_{t=0}^{T} h_t(I,p) = \sum_{t=0}^{T} z_{L,0}^{t}(I,p) \tag{3.10}$$

3.5.2.4 Alignement par maximisation du score

La méthode d'alignement repose sur la maximisation du score de *GentleBoost* par un algorithme de gradient suivant les paramètres p :

$$\frac{\partial H}{\partial p} = \sum_{t=0}^{T} \frac{\partial z_{L,0}^{t}}{\partial p} \tag{3.11}$$

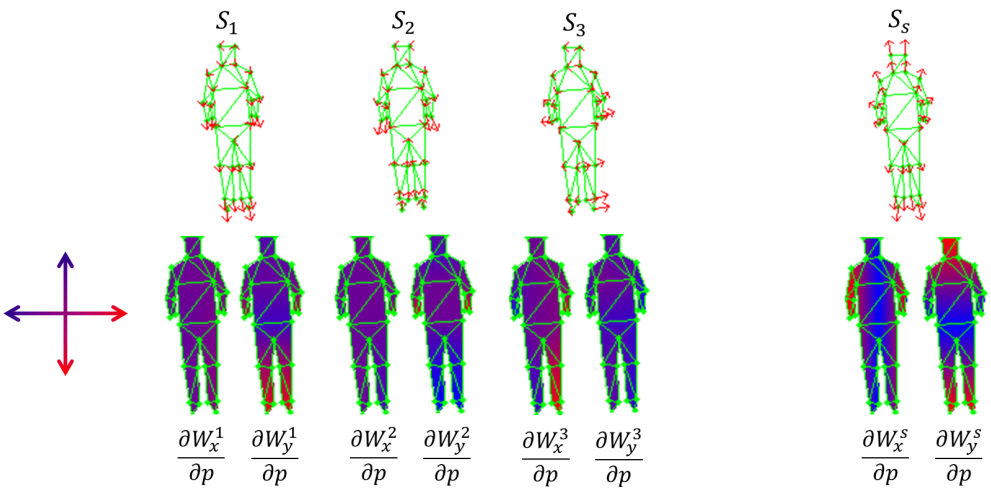
Nous calculons la forme dérivée d'un perceptron à partir de l'Équation 3.9 :

$$\frac{\partial z_{c,d}}{\partial p} = \sigma'\left(\left(\sum_{n=0}^{D_{c-1}} w_{c,d,n}\, z_{c-1,n}\right) + b_{i,j}\right) \sum_{n=0}^{D_{c-1}} w_{c,d,n} \frac{\partial z_{c-1,n}}{\partial p}_{c=1,\ldots,L} \tag{3.12}$$

$$\frac{\partial z_{0,d}}{\partial p} = \frac{\partial f^{(d)}(I; W(f_x, f_y), f_w, f_h)}{\partial p}$$

avec $\sigma'(x) = \frac{2e^{-x}}{(1+e^{-x})^2}$. Grâce au théorème de dérivation des fonctions composées, la dérivée de la caractéristique par rapport au paramètre p peut se décomposer de la manière suivante :

$$\frac{\partial f}{\partial p} = \frac{\partial f}{\partial W_x}\frac{\partial W_x}{\partial p} + \frac{\partial f}{\partial W_y}\frac{\partial W_y}{\partial p} \tag{3.13}$$



(a) Modes de forme (b) Échelle artificielle

FIGURE 3.15 – Représentation des termes de la matrice jacobienne de la fonction de transformation par parties par rapport à p.

Les dérivées des caractéristiques HOG par rapport à W_x et W_y sont approximées numériquement par le taux d'accroissement sur un pixel centré sur ces coordonnées transformées et calculées respectivement sur les *channels* de gradients orientés correspondants. Les termes $\begin{bmatrix} \frac{\partial W_x}{\partial p} & \frac{\partial W_y}{\partial p} \end{bmatrix}$ sont les composantes de la matrice jacobienne de la fonction de transformation affine par parties évaluée en $p = 0$ (Figure 3.15). Ils peuvent ainsi être pré-calculés sur la base des modes $\bar{S}_i$. Nous reprenons la méthode de calcul présentée par [Matthews and Baker, 2004].

Puisque l'approche est itérative, il convient de définir une condition d'arrêt. Elle s'obtient suite à la convergence de l'algorithme, c'est-à-dire lorsque la norme de la différence entre deux formes consécutives est inférieure à un seuil, ou lorsqu'une limite définie d'itérations est atteinte.

Par ailleurs, nous étendons cette technique d'alignement à l'échelle et aux translations. En effet, dans notre application, la phase de détection ne délivre pas nécessairement une zone parfaitement centrée et mise à l'échelle par rapport à la personne. Puisque l'alignement du modèle est fait itérativement, la forme initiale a une grande importance et un offset peut fausser la convergence. Pour résoudre cela, nous proposons une pré-phase d'alignement afin de gommer les erreurs de translation et d'échelle. Nous entraînons un second modèle, basé sur le même schéma d'apprentissage avec comme forme exclusive $\bar{S}_0$ et où les perturbations progressives de déformations sont remplacées par celles de translation et d'échelle. Nous générons ces échantillons de sorte à respecter le critère de recouvrement donné Équation 2.13 entre la boîte englobante contenant la forme vérité terrain et la boîte englobante perturbée. Puis, en phase d'alignement, nous mettons à jour simultanément le centre de la forme $\bar{S}_0$ et son échelle de la façon suivante :

$$\begin{bmatrix} \bar{x}_{\bar{S}_0} \\ \bar{y}_{\bar{S}_0} \end{bmatrix} = \begin{bmatrix} \bar{x}_{\bar{S}_0} + \lambda_t \frac{\partial H}{W_x} \\ \bar{y}_{\bar{S}_0} + \lambda_t \frac{\partial H}{W_y} \end{bmatrix} \tag{3.14}$$

$$s_{\bar{S}_0} = s_{\bar{S}_0} + \lambda_s \frac{\partial H}{p_s}$$

Concernant les termes de translation, il suffit de remplacer ∂p par ∂W_x et ∂W_y dans les Équa-

tions 3.11 et 3.12 et d'omettre les termes jacobiens dans l'Équation 3.13. En ce qui concerne le terme d'échelle $\frac{\partial H}{p_s}$, nous générons artificiellement un mode d'agrandissement $\bar{S}_s$ (illustré Figure 3.15b) de la forme pour pouvoir s'appuyer sur les mêmes calculs. Puisque nous proposions de gérer la rotation en phase de détection, nous n'avons pas considéré dans le cas présent, une correction de la rotation générale en phase d'alignement. Celle-ci demanderait des considérations réflexions par rapport à la résolution par l'algorithme de gradient, notamment au niveau de l'approximation des dérivées des caractéristiques vis-à-vis des coordonnées W_x et W_y.

3.5.3 Évaluations et validations

3.5.3.1 Métrique d'évaluation

Pour valider notre algorithme, nous nous basons sur la partie test de la base InriaLHB que nous avons introduite. Les critères d'évaluation doivent illustrer la distance entre la forme estimée et la forme vérité terrain, comme indiqué par l'Équation 3.1. Classiquement, les algorithmes d'alignement s'évaluent avec la mesure cumulée de la racine de l'erreur moyenne quadratique (en anglais *Root Mean Square Error* ou RMSE). Elle se calcule de la manière suivante :

$$RMSE = \sqrt{\frac{\sum_{k=1}^{K} (\hat{x_k} - x_k)^2 + (\hat{y_k} - y_k)^2}{2K}} \tag{3.15}$$

La RMSE peut également être normalisée selon une distance dans l'image afin de gagner une certaine invariance vis-à-vis de l'échelle. Dans le cas du visage, l'erreur est normalisée par la distance interoculaire donnée par les amers correspondants. Dans notre cas, nous utiliserons la distance entre le centre formé par les épaules et celui formé par les points du bassin :

$$D_{\text{norm}} = \sqrt{\left(\frac{\hat{x}_4 + \hat{x}_{28}}{2} - \frac{\hat{x}_{10} + \hat{x}_{22}}{2}\right)^2 + \left(\frac{\hat{y}_4 + \hat{y}_{28}}{2} - \frac{\hat{y}_{10} + \hat{y}_{22}}{2}\right)^2} \tag{3.16}$$

Néanmoins, puisque nous fournissons également la base sur les images normalisées en taille, nous représenterons nos résultats directement avec la RMSE non normalisée.

3.5.3.2 Validation de l'algorithme

Comme nous l'avons vu précédemment, le processus d'alignement nécessite une forme initiale. Nous établissons la validation des différents éléments de l'algorithme sur une position centrée sur la vérité terrain et dimensionnée correctement, correspondant à une détection parfaite. Cela permet de voir le comportement de notre approche par rapport aux déformations. Nous évaluons par la suite la robustesse sur une détection faussée en conservant les paramètres optimaux trouvés dans le cas des déformations seules. La méthode de [Liu et al., 2008] que nous avons implémentée nous servira également de *baseline*.

Paramètres

La détermination des différents paramètres de notre algorithme a été faite en se servant des résultats croisés sur la base de test. Pour illustrer le comportement individuel de chaque élément sur notre algorithme, nous faisons varier chaque paramètre de l'algorithme en fixant les autres avec les valeurs par défaut suivantes :

Structuration en classement de la base :
- $M = 7$ perturbations consécutives pour chaque échantillon de la vérité terrain vers $\bar{S}_0$
- $M = 7$ perturbations consécutives pour chaque échantillon en dépassement de la vérité terrain

Caractéristiques HOG locales :
- taille (en hauteur ou largeur) d'une cellule comprise entre 4 et 8 pixels, soit une taille totale de la caractéristique entre 8 et 16 pixels
- normalisation par rapport à l'histogramme complet de la caractéristique
- adaptation à l'orientation locale
- 6 orientations

Entraînement de *GentleBoost* :
- $T = 200$ régresseurs/classifieurs faibles
- 300 candidats pour chaque itération

Régresseur perceptron multicouche :
- $C = 2$ avec $D_1 = 30$, ce qui revient à un réseau avec 1 couche cachée et 30 neurones
- entraîné via l'algorithme RPROP [Riedmiller and Braun, 1993] (implémenté par la librairie OpenCV)

Alignement :
- λ est un paramètre à ajuster empiriquement pour chaque *GentleBoost*. Pour les réseaux neuronaux à 1 couche, nous utilisons $\lambda = 0.5$.
- Nombre maximal d'itérations fixé à 10

Structuration de la base d'apprentissage et régression de score de classement

Nous évaluons dans un premier temps notre principale amélioration par rapport à la *baseline*, à savoir la structuration en classement de la base d'apprentissage. Nous comparons les performances délivrées par la structuration originale qui s'appuie sur la classification par une WLDA, entre les positifs, vérité terrain, et les négatifs, échantillons ayant subi des perturbations aléatoires des paramètres p. La fonction de sélection d'un candidat ne s'appuie plus sur des paires d'échantillons perturbés consécutivement mais directement avec les labels utilisés pour construire la WLDA.

Nous évaluons également l'utilisation de la WLDA sur notre structuration en classement. Puisqu'elle se construit comme un classifieur, nous attribuons un label positif pour la moitié des échantillons les plus proches en terme de perturbations et un label négatif pour la seconde moitié. La sélection est néanmoins faîte sur les paires (Équation 3.7). La Figure 3.16a montre que la considération de l'alignement comme une fonction de classement permet d'obtenir de meilleurs résultats que ceux de la *baseline*. De plus, l'utilisation de réseaux neuronaux par rapport à la WLDA améliore la moyenne de RMSE de 0.74 (4.63 contre 5.37). Cela provient du fait que nous explicitons en phase d'apprentissage du régresseur faible la progression que doit suivre l'alignement.

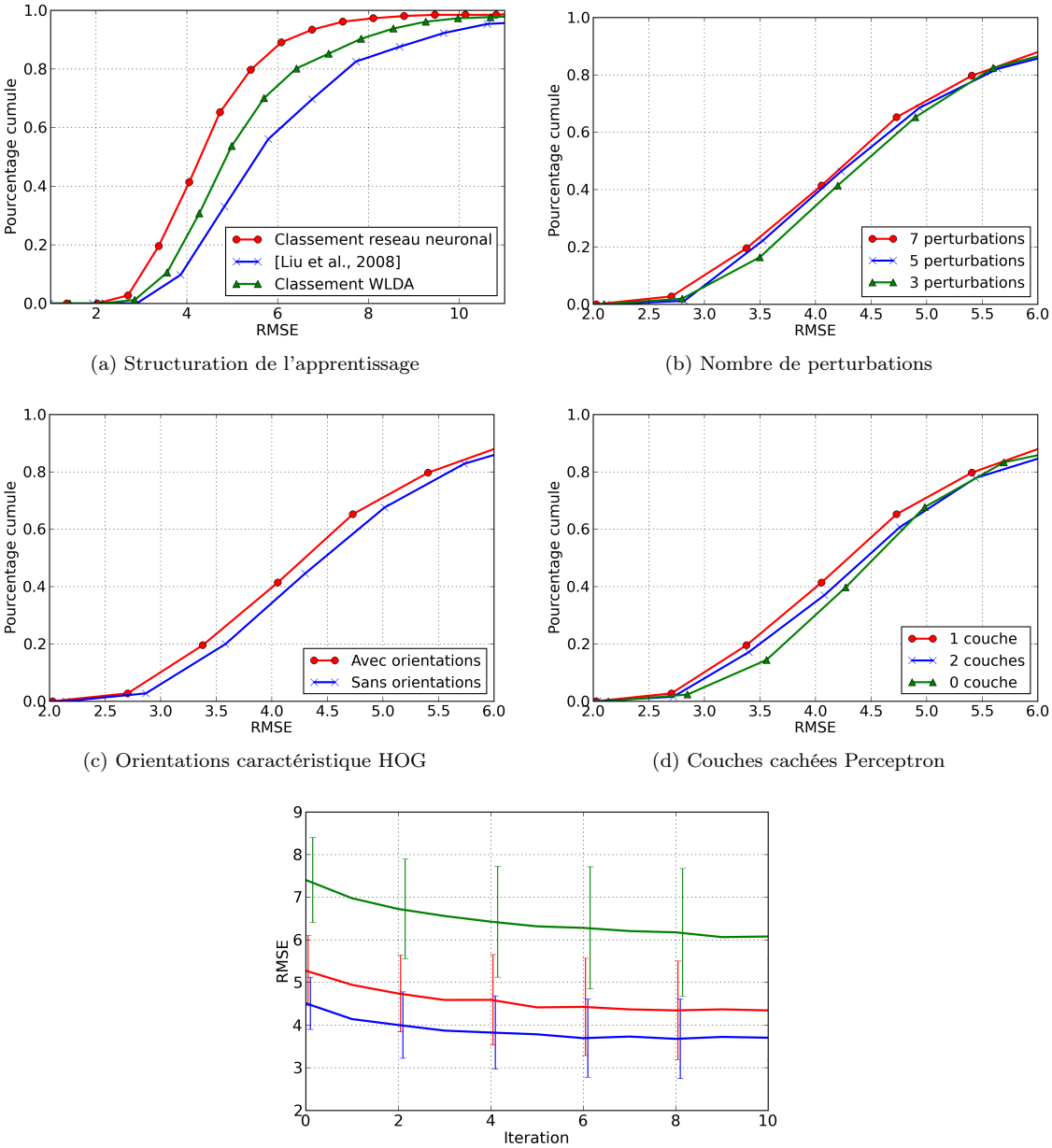
Nous étudions l'influence du nombre de perturbations M lors de la structuration de la base (Figure 3.16b). Sans surprise, une légère amélioration est apportée avec plus de perturbations, néanmoins la génération d'un trop grand nombre d'échantillons négatifs alourdit le temps d'apprentissage. Pour disposer de 7 perturbations entre $\hat{S}$ et $\bar{S}_0$, nous avons généré 11 424 échantillons.

Un exemple d'alignement ainsi que le score et la RMSE associée sont donnés Figure 3.17. On peut constater que le score évolue sur une plage plus importante pour les réseaux neuronaux, impliqué par une meilleure différenciation des différentes étapes de l'alignement lors de l'apprentissage. Nous notons également que le score fourni par *GentleBoost* de la *baseline* évolue très peu bien que nous ayons augmenté la valeur de λ à 50 (contre $\lambda = 0.5$ pour notre système et $\lambda = 20$ pour la WLDA construite sur un modèle de classement).

Caractéristiques et régresseurs

Nous montrons Figure 3.13 que la prise en compte de l'orientation permet d'obtenir une légère augmentation au niveau des résultats d'alignement. Néanmoins, à cause des difficultés listées ultérieurement qui empêchent le modèle de s'aligner correctement sur des bras levés à l'horizontal, cette configuration n'est pas entièrement exploitée. Il est aussi intéressant de voir les caractéristiques sélectionnées par *GentleBoost* Figure 3.18. Comme avancé par [Liu et al., 2008], cette machine de boosting va effectivement favoriser la sélection de caractéristiques placées sur les bords de la personne afin de s'appuyer sur les gradients découlant des contours.

Par ailleurs, nous avons cherché à voir l'influence du nombre de couches cachées des régresseurs faibles. La Figure 3.16d indique qu'utiliser des perceptrons à une couche cachée délivre les meilleurs résultats. Nos expérimentations montrent que le comportement d'alignement d'un réseau à deux couches cachées est beaucoup plus instable que celui à une couche, provenant probablement d'un phénomène de sur-apprentissage.



(a) Structuration de l'apprentissage

(b) Nombre de perturbations

(c) Orientations caractéristique HOG

(d) Couches cachées Perceptron

(e) Évolutions sur Translation/Échelle

FIGURE 3.16 – Évaluations et validations de notre algorithme sur la base de test InriaLHB. (a) (b) (c) (d) sont évalués sur la base des déformations avec une initialisation centrée tandis que (e) part d'une initialisation perturbée.

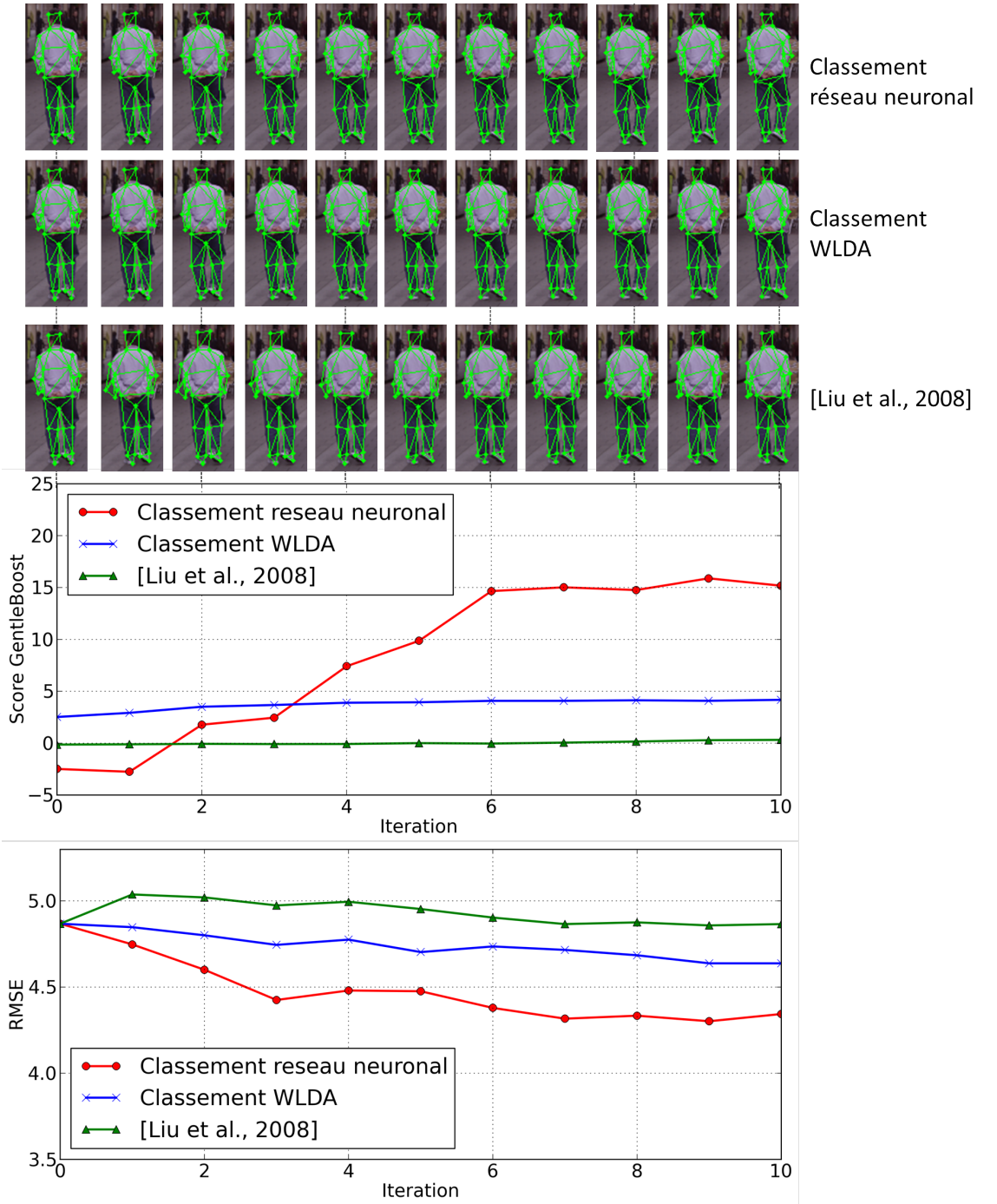
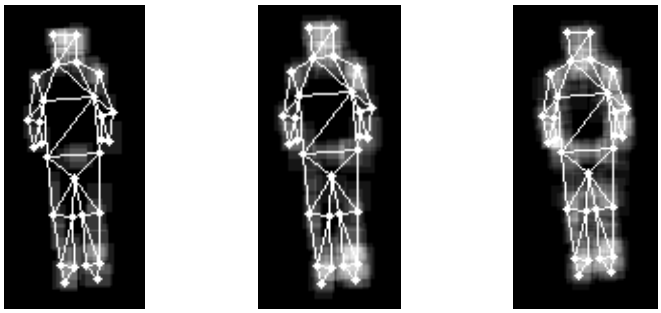


FIGURE 3.17 – Illustration d'un alignement en 10 itérations et correspondance de l'évolution du score de *GentleBoost* et de l'évolution de la RMSE.

Détection perturbée

Afin d'évaluer la capacité de notre algorithme à gérer le cas de détections imprécises, nous considérons la vérité terrain non plus comme étant $\hat{S}_n$ mais $\bar{S}_0$ à l'échelle $\hat{s}$ et centrée avec le facteur de translation $\hat{t}$, soit :



(a) 50 premières (b) 100 premières (c) 200

FIGURE 3.18 – Répartition des caractéristiques réparties sur la forme $\bar{S}_0$ et sélectionnées chronologiquement par *GentleBoost*.

$$\hat{S}_{n,0} = \hat{s}\bar{S}_0 + \hat{t} \tag{3.17}$$

Nous générons des situations initiales en perturbant $\hat{s}$ et $\hat{t}$ de sorte que la boîte englobante formée par ces initialisations perturbées respecte le critère de recouvrement pour une détection avec $\hat{S}_n$. L'algorithme consacré à cette tâche conserve les mêmes paramètres d'apprentissage et d'alignement listés auparavant. Nous représentons Figure 3.16e l'évolution de la RMSE à partir de plusieurs exemples d'initialisations. L'algorithme parvient à corriger globalement une certaine part de l'erreur engendrée par la détection approximative. Néanmoins, le fait que l'écart-type augmente au fur et à mesure de l'alignement (environ de 60%) montre un comportement hétérogène par rapport aux échantillons. Par ailleurs, l'erreur résiduelle se répercute directement sur l'alignement des déformations. En effet, l'apprentissage pour ces dernières a été réalisé dans des conditions où l'initialisation était parfaitement centrée. Une solution possible serait de poursuivre la correction de translation et d'échelle en même temps que l'alignement des déformations.

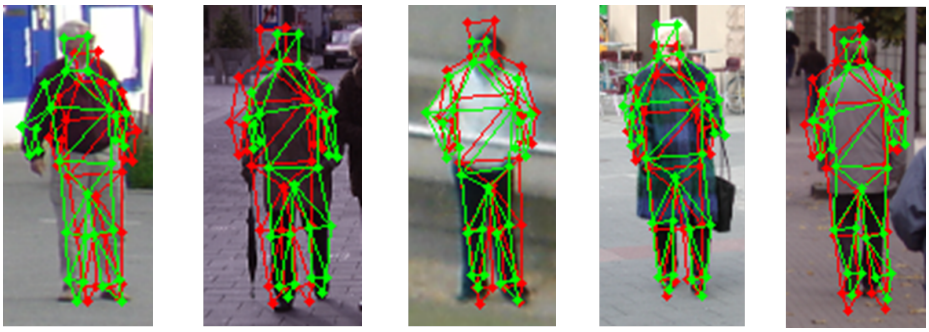
Temps de calcul

Les expérimentations ont été réalisées sur la même configuration PC que celle utilisée pour la détection. Nous moyennons le temps nécessaire à l'alignement sur l'ensemble de la base de test. Pour 10 itérations au maximum, notre implémentation fonctionne aux alentours de 12.2 FPS. Il faut compter une vitesse deux fois moindre si l'on intègre la phase de correction en translation et en échelle. En moyenne, l'alignement est atteint au bout de 8 itérations. Le calcul des *channels* de gradients orientés a été pris en compte dans la vitesse. Il peut néanmoins être économisé en s'appuyant sur les *channels* générés en phase de détection.

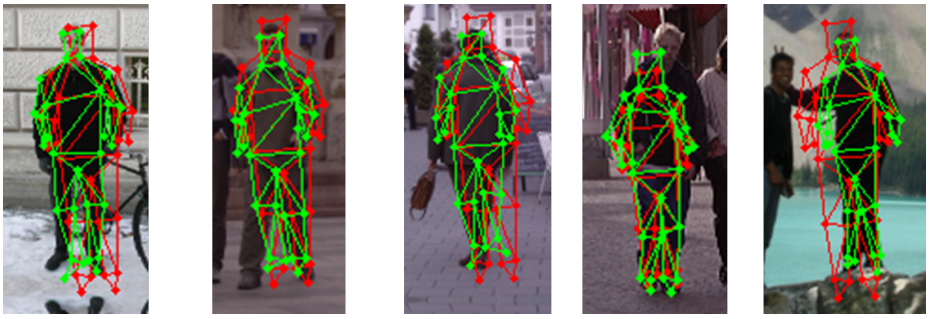
3.5.4 Limitations et discussions

Malgré l'amélioration que nous apportons à l'approche proposée par [Liu et al., 2008], les résultats globaux délivrés par ce premier système se révèlent décevants. En effet, nous jugeons pour le moment inexploitables la qualité de l'alignement fourni en vue d'une ré-identification. Globalement, la forme obtenue ne permet pas de segmenter correctement la personne du décor et l'algorithme échoue à s'aligner sur des poses plus complexes qui s'éloignent de la forme $\hat{S}_0$. Nous identifions trois raisons à ces résultats non conformes à nos attentes.

La première, principale, découle directement de la méthode de résolution de l'algorithme. En effet, comme pour toute descente de gradient, l'algorithme peut se bloquer dans des minima locaux (ici maxima locaux du score). Ce fait est d'autant plus vrai dans notre cas où le point d'initialisation peut être éloigné de la solution optimale. C'est pourquoi cette méthode peut fonctionner dans le cas



(a) Alignements corrects



(b) Alignements incorrects

FIGURE 3.19 – Exemples d'alignement du modèle d'apparence par boosting. La forme en rouge représente l'initialisation et celle en vert le résultat de l'alignement.

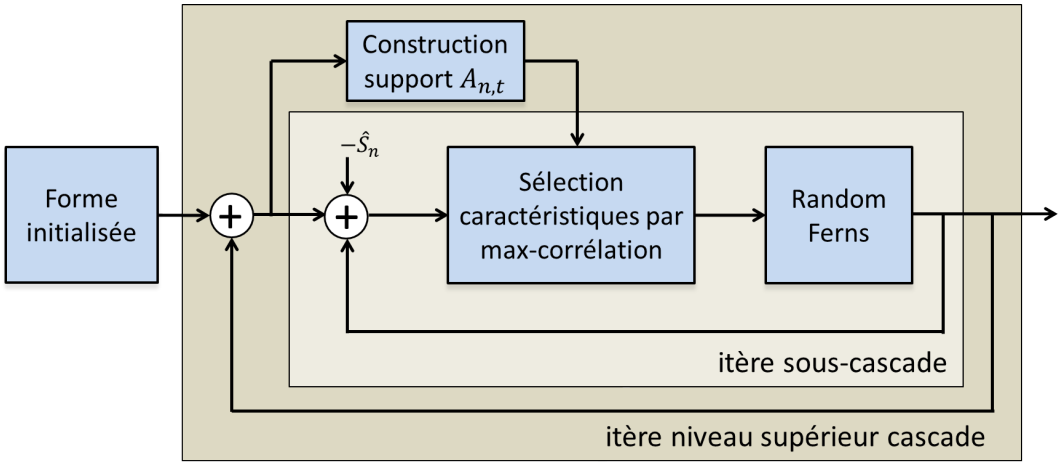
d'une initialisation perturbée non éloignée de la vérité terrain (comme l'évaluation proposée par [Liu et al., 2008]), mais aura des difficultés dans un cas réel où la vérité terrain est inconnue. De plus, la descente de gradient utilise une approximation sur un taux d'accroissement sur un pixel, induisant automatiquement une recherche localisée autour du point courant. Ceci implique qu'aligner un point virtuellement éloigné de l'objectif implique une difficulté dès lors que la distance dépasse la taille de la caractéristique locale.

La seconde raison, soulevée également par les récentes méthodes de régression [Cao et al., 2013; Xiong and De la Torre, 2013] provient des limitations imposées par un modèle paramétrique. En effet, trouver les paramètres p est sous-optimal pour répondre à l'objectif initial donné Équation 3.1. De plus, cette description paramétrique peut également ne pas s'adapter à une nouvelle forme non présente dans la base d'apprentissage.

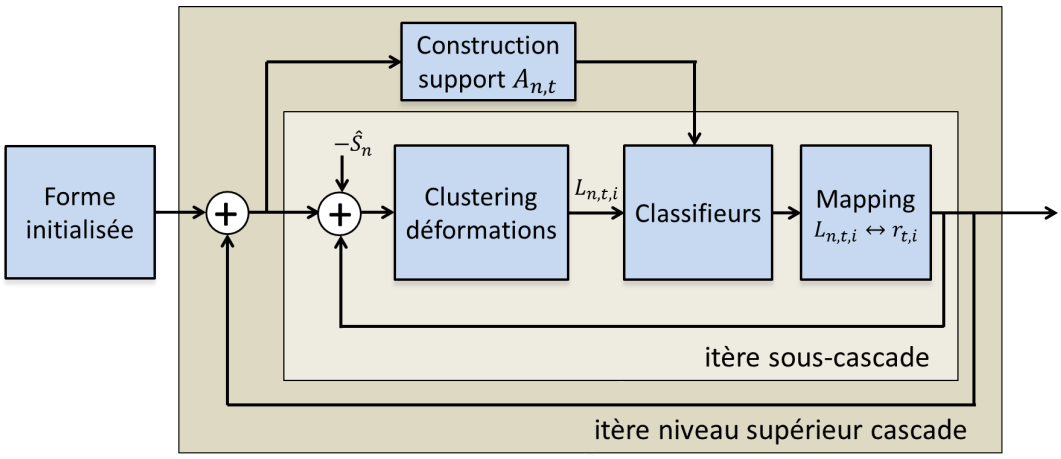
Enfin, la troisième raison provient de la puissance insuffisante des régresseurs faibles face à la tâche demandée. En effet, l'alignement est complexe et présente des situations variables au fur et à mesure de la convergence. Dans le cas présent, le même régresseur faible doit gérer à l'initialisation une distance éloignée par rapport à la vérité terrain et un affinement précis dans les dernières itérations (de grossier à fin).

Sur la base de ces trois raisons, nous avons développé un second algorithme d'alignement, s'appuyant sur une technique de cascade de régressions que nous présentons dans la section suivante.

3.6 Régression de forme par classification des déformations



(a) ESR



(b) Clustering des déformations

FIGURE 3.20 – Diagrammes de fonctionnement de la double cascade de régressions de forme proposée par [Cao et al., 2013] et la variante que nous proposons. La sélection de caractéristiques se fait sur la discrimination entre les déformations.

3.6.1 Cascade de régressions de forme

La cascade de régressions de forme est une technique récente montrant des résultats prometteurs [Dollár et al., 2010; Cao et al., 2013; Xiong and De la Torre, 2013]. Sa structure intègre naturellement les composantes utiles pour une tâche d'alignement comme les contraintes de forme ou une progression d'échelle grossière à fine. Nous proposons d'utiliser comme base l'algorithme *Explicit Shape Regression* ou ESR proposé par [Cao et al., 2013]. Nous décrivons dans un premier temps son fonctionnement avant d'aborder les améliorations que nous apportons. Une cascade de régressions de forme opère de façon similaire à une composition successive de régresseurs. Nous écrivons le régresseur de forme $R : \mathbb{R}^{m\times 1} \rightarrow \mathbb{R}^{2K\times 1}$ avec m la dimension de l'espace des caractéristiques utilisées et K le nombre de points rattachés au PDM.

Chaque régresseur prend en entrée la forme obtenue par le résultat du régresseur précédent.

Cette technique diffère de notre précédente approche puisque les mêmes régresseurs étaient utilisés tout au long de l'alignement. Si l'on note l'étape dans la cascade t, la forme courante de l'échantillon n est mise à jour de façon additive :

$$S_{n,t} = S_{n,t-1} + R_t(I_n, S_{n,t-1}) \tag{3.18}$$

Au niveau de l'apprentissage, la cascade est construite de façon récursive en entraînant chaque régresseur t sur l'erreur résiduelle de forme assimilée à la direction des déformations :

$$\Delta S_{n,t} = \hat{S}_n - S_{n,t-1} \tag{3.19}$$

La spécificité de l'ESR est d'introduire une sous-cascade qui décompose R_t en une série de I régresseurs dits primitifs. Ces derniers, que l'on notera $r_{t,i}$, calculent leur sortie de la même manière sur les déformations résiduelles fournies par l'étape précédente dans la sous-cascade i. Néanmoins, ils prennent en entrée un espace commun de caractéristiques sur l'étape t. Cet espace de caractéristiques s'obtient par le support d'apparence formé par l'image et la forme courante : $A_{n,t} = (I_n, S_{n,t-1})$. En général, il s'agit d'un ensemble de caractéristiques locales indexées à la forme courante. Nous pouvons écrire le régresseur primitif comme étant $r_{t,i}(A_{n,t}, S_{n,t-1,i-1})$. En l'injectant dans l'Équation 3.18, nous obtenons la règle de mise à jour suivante :

$$S_{n,t} = S_{n,t-1} + \sum_{i=1}^{I} r_{t,i}(A_{n,t}, S_{n,t-1,i-1}) \tag{3.20}$$

Cao et al. [2013] montrent que seulement mettre à jour la forme utilisée pour l'apparence dans la couche supérieure de la cascade stabilise la régression et améliore le processus d'alignement complet. Cela peut s'expliquer par le fait que le groupe des $r_{t,i}$ s'apparente à une méthode d'ensemble. Cette sous-cascade permet également d'économiser du temps de calcul en n'effectuant les opérations nécessaires à la construction de $A_{n,t}$ seulement sur la couche supérieure de la cascade. Dans le système original, les régresseurs primitifs utilisés sont des *random ferns* [Ozuysal et al., 2010], variante d'arbres décisionnels binaires arrangée structurellement que l'on peut évaluer par une séquence de tests binaires. Leur sortie est déterminée par la forme centroïde $\overline{\Delta S}_{m,t,i}$ de l'ensemble des échantillons tombant dans chaque feuille D_m :

$$r_{t,i} = \frac{1}{1 + \frac{\beta}{\text{taille}(D_m)}} * \overline{\Delta S}_{m,t,i} = \frac{1}{1 + \frac{\beta}{\text{taille}(D_m)}} * \frac{\sum_{n \in D_m} \hat{S}_n - S_{n,t-1,i-1}}{\text{taille}(D_m)} \tag{3.21}$$

$$\text{avec} \quad m = C_{t,i}$$

$C_{t,i}$ correspond à une fonction de partitionnement, en l'occurrence le *random fern*. Le premier terme est un facteur de rétrécissement dans le but d'atténuer le sur-apprentissage, avec β une constante à ajuster. Il permet d'attribuer plus de poids aux feuilles D_m regroupant de nombreux échantillons. Grâce à cette formulation, [Cao et al., 2013] montrent que les contraintes de forme sont implicitement encodées par les données d'apprentissage et que la forme courante est toujours une combinaison linéaire des formes d'entraînement et de la forme initiale. Les *random ferns* sont entraînés de sorte à seuiller aléatoirement des caractéristiques sélectionnées par le maximum de corrélation avec les déformations.

3.6.2 *Clustering* et classification des déformations

Notre apport principal par rapport à l'ESR est motivé par l'expression de sortie des régresseurs faibles donnée à l'Équation 3.21. Cette dernière est calculée par la moyenne d'un groupe d'échantillons, donc d'une direction générale des déformations. En cherchant à homogénéiser ce groupe de

sorte qu'il possède une variance minimale en terme de déformations, on augmente la signification et la confiance accordées à cette direction utilisée comme incrément. Si au contraire, des échantillons présentent des tendances contraires au sein d'un même groupe, ils se compenseront et la moyenne sera moins représentative.

L'idée est donc de grouper les déformations homogènes dans des partitions grâce à des méthodes de *clustering* statistiques. Nous considérerons ces partitions comme des classes de déformations, déterminées de manière non supervisée. Nous entraînons sur cette base des classifieurs de façon supervisée avec les labels fournis par l'étape de *clustering*. En introduisant une étape intermédiaire de classification à la régression, l'algorithme va chercher à trouver des caractéristiques communes à des déformations homogènes, tout en cherchant à discriminer ces groupes de déformations entre eux. Cela revient à minimiser le rapport des distances intra-classes sur celles inter-classes. Il s'agit d'un point très intéressant dans le cas de l'alignement sur un objet articulé et déformable comme le corps humain, étant donné le grand nombre de cas possibles pouvant être atteints dans l'espace des déformations. De plus, en procédant ainsi, nous apportons une structure et un découpage de l'espace d'apprentissage afin de guider l'apprentissage des régresseurs faibles. Cette idée présente des similitudes avec les *poselets* qui sont des parties du corps adoptant une certaine pose déterminées statistiquement par rapport à la base d'apprentissage. La différence, dans notre application, réside dans le fait qu'il s'agit de pose relative à la forme courante, évoluant au fur et à mesure de la cascade.

Par rapport à la formulation présentée précédemment Équation 3.21, $C_{t,i}$ représente un classifieur multiclasse associé au régresseur primitif $r_{t,i}$. Ce classifieur s'appuie, lors de sa phase d'entraînement, sur les données d'apprentissage formées par $(A_{n,t}, L_{n,t,i})$, $L_{n,t,i}$ étant un label résultant d'une fonction de *clustering* sur l'ensemble des résidus des formes $\Delta S_{n,t,i}$. Pour le partitionnement des données, nous utilisons une version modifiée du K-means : le K-means++ [Arthur and Vassilvitskii, 2007] qui optimise le placement des graines en initialisation de l'algorithme.

3.6.2.1 Simulation et généralisation des formes

Pour illustrer cette idée, nous procédons à des expérimentations en simulant le résultat du classifieur $C_{t,i}$. Cette simulation a également pour but de vérifier si notre approche permet à l'algorithme de généraliser de nouvelles formes. Au niveau des paramètres de cette simulation, nous fixons le nombre de *clusters* $M = 16$, le facteur de rétrécissement $\beta = 1000$ dans l'expression de l'incrément de formes, le nombre de régresseurs totaux à 500 (la notion de double cascade n'intervenant pas ici car nous n'utilisons pas les données d'apparence).

Dans un premier temps, nous séparons en deux la base d'entraînement initiale en une base d'apprentisage (4/5) et une base de validation (1/5).

Par la suite, nous construisons les *clusters* de déformations par l'algorithme du K-means avec uniquement les échantillons d'apprentissage. Chaque échantillon de validation est rattaché au *cluster* le plus proche, ramené à sa moyenne, en distance euclidienne.

Nous fixons ensuite une erreur de classification d'entraînement et de validation qui simule les performances du classifieur à entraîner. En fonction de cette erreur, nous attribuons à la proportion d'échantillons d'apprentissage et de validation la bonne classe/*cluster* ou une mauvaise classe tirée aléatoirement. Nous réitérons le processus sur toute la cascade. Nous représentons l'évolution de la RMSE moyenne sur la cascade dans différentes configurations d'erreurs d'apprentissage et de validation Figure 3.21a.

Nous constatons que l'erreur de classification impacte fortement les performances de l'alignement. En effet, le cas idéal d'un classifieur mène rapidement à une convergence correcte, tandis que celui d'un classifieur avec une mauvaise généralisation (0%/50%) donne des résultats bien moindres. La simulation illustre également la bonne capacité de l'algorithme à s'adapter à des formes non présentes dans la base d'apprentissage, sous réserve d'une bonne généralisation des classifieurs. Nous

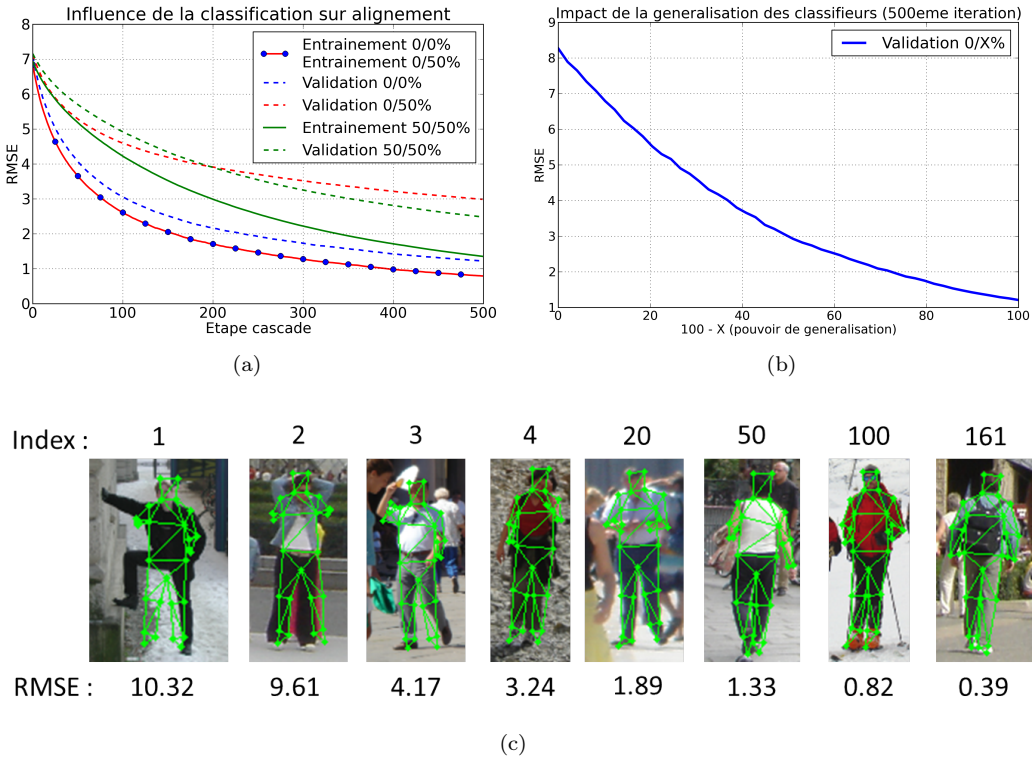


FIGURE 3.21 – (a)(b) Structuration de la légende : [Dataset correspondante] [Erreur simulée d'entraînement]/[Erreur simulée de validation]%. (a) Évolutions de la moyenne de la RMSE en fonction de 3 combinaisons d'erreur de classification d'apprentissage et de validation. (b) Évolution de la moyenne de la RMSE en fin de cascade pour une erreur d'apprentissage nulle. (c) Exemples d'alignement pour la généralisation de formes. Les 4 premiers exemples sont associés avec l'échantillon généré par une symétrie horizontale.

obtenons une RMSE moyenne finale de 0.84 pour l'entraînement et de 1.19 pour la validation (égales respectivement au départ à 6.84 et à 7.08). Néanmoins, les algorithmes basés sur les PDM sont uniquement orientés sur les données d'apprentissage pour définir les contraintes de forme. Par conséquent, certaines postures atypiques ou très peu représentées dans la base ne parviennent pas à être extrapolées par l'algorithme. Nous donnons une représentation de cet échec de généralisation Figure 3.21c. Une solution serait, à court terme, d'étayer la base en y représentant des poses plus complexes ou de définir un modèle plus explicite pour gérer les objets articulés.

Un autre point intéressant soulevé par cette simulation concerne le sur-apprentissage des classifieurs. Ce cas est illustré par les courbes possédant la même erreur de validation et les erreurs d'entraînement différentes : 0%/50% (sur-apprentissage) et 50%/50%. Nous constatons que la tendance d'alignement s'inverse lors de la cascade et que les classifieurs possédant une erreur d'entraînement non nulle fournissent de meilleurs résultats. Nous expliquons cela par la structure en cascade de l'algorithme et l'erreur résiduelle qui se propage. Dans un cas de sur-apprentissage, l'alignement va se faire très rapidement pour les échantillons d'entraînement. Or, les mauvais résultats du classifieur vont empêcher la partie validation de suivre ce rythme d'alignement. En conséquence, l'information d'alignement contenue dans les directions générales de déformations fournies par les *clusters* ne va pas être assimilée par les échantillons de validation, qui vont se retrouver en quelque sorte «bloqués» à un mauvais état d'alignement. Pour atténuer ce problème, une première solution est d'augmenter le facteur de rétrécissement β pour diminuer l'intensité de l'incrément. Néanmoins, cela nécessite alors d'augmenter la taille de la cascade et par conséquent d'alourdir les temps de calcul. Une autre

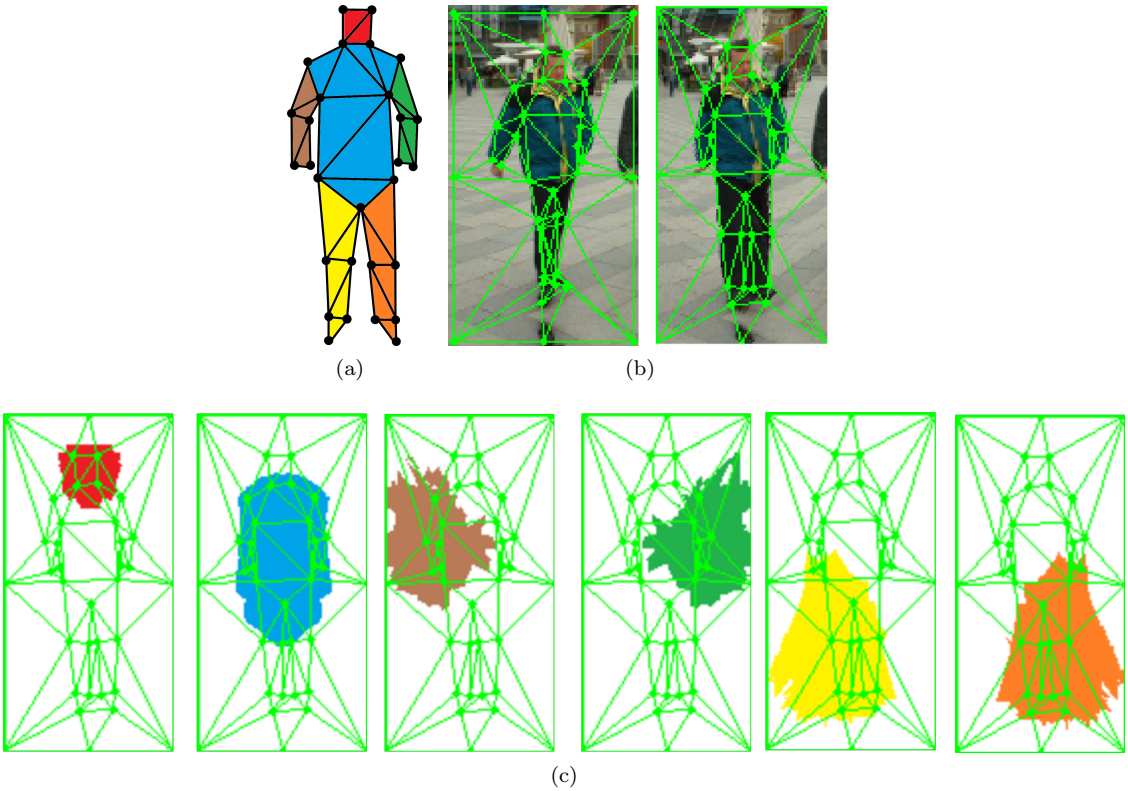


FIGURE 3.22 – (a) Considération par parties (b) Maillage complété pour englober l'apparence autour de la personne (c) Occupation potentielle des caractéristiques locales à la première itération de la cascade par rapport à chaque partie considérée, en se basant sur les données d'apprentissage.

solution est de perturber l'apprentissage, en injectant les échantillons d'apprentissage correctement classifiés dans des *clusters* différents. Enfin, une troisième solution est de réaliser l'apprentissage sur un sous-ensemble des données d'entrée mais de calculer les incréments sur l'ensemble de la base.

En ce qui concerne la correction de détections imparfaites, nous proposons la même stratégie que le système paramétrique. Il s'agit d'aligner la forme $\hat{S}_0$ à une position perturbée sur la position centrée. Nous consacrons les premières étapes de la cascade aux transformations d'échelle et de translation, puis nous modifions l'alignement pour qu'il se fasse sur les déformations.

3.6.2.2 Stratégie pour la classification

Jusqu'à présent, nous avons caractérisé les déformations sur l'ensemble des amers. Nous avons vu lors de la description de la base annotée que l'articulation du corps humain entraîne un très vaste champ potentiel de déformations. Pour simplifier la tâche effectuée par les classifieurs, nous proposons une considération par parties au niveau du *clustering* des déformations. À chaque étape, le *clustering*, qui va définir les classes de déformations, est fait uniquement sur un sous-ensemble d'amers. Cependant, nous calculons les incréments $r_{t,i}$ toujours sur le PDM entier de sorte à conserver les contraintes de forme induites par les données d'apprentissage. En procédant ainsi, nous spécialisons chaque étape de la cascade sur une partie du corps humain. Cette notion locale se retrouve dans les travaux de [Ren et al., 2014]. Concernant les sous-parties, nous les définissons comme étant les différents membres articulés du corps humain (6 au total : tête/torse et bassin/bras gauche/bras droit/jambe gauche/jambe droite). Nous les représentons Figure 3.22a.

Un défi important découlant de notre approche est le déséquilibre de données par classe de déformations. En effet, les poses classiques sont davantage représentées dans la base d'apprentissage que des poses atypiques. Par ailleurs, en raison de la structure en cascade, le nombre d'échantillons proches de la vérité terrain va naturellement croître. Si l'algorithme ne fait pas uniformément converger la base d'apprentissage, le résultat du *clustering* sera déséquilibré avec une grande classe pour les déformations de faibles amplitudes et des classes éparses de plus fortes déformations.

Le déséquilibre dans les données d'entraînement est une vaste problématique des machines d'apprentissage. La solution à adopter dépend en grande partie du problème. Dans notre cas, nous proposons l'approche suivante. Nous adoptons un *clustering* dichotomique *up-bottom*, similaire au partitionnement récursif proposé par [Strobl et al., 2009]. Nous construisons un arbre binaire où, pour chacun de ses nœuds, nous déterminons deux partitions de déformations. Procéder ainsi confère deux avantages. Dans un premier temps, cela facilite la tâche des classifieurs en les ramenant à un problème binaire. Nous utilisons comme classifieur binaire *AdaBoost* précédemment présenté Algorithme 1, qui permet de s'accommoder aux données déséquilibrées grâce à son approche pondérée. Dans un second temps, l'erreur de classification est atténuée dans une structure en arbre car les échantillons mal classifiés sont réévalués sur les étages inférieurs. Un inconvénient est que cela augmente le nombre de classifieurs à entraîner à $2^p - 1$, en se basant sur $M = 2^p$ *clusters*. La simulation Figure 3.21 montre que nous obtenons une convergence suffisamment rapide pour 16 *clusters*, suggérant qu'il est possible de se baser sur des arbres à $p = 4$ niveaux.

Une autre alternative pour gérer le déséquilibre des données est le *resampling*. Nous homogénéisons les proportions des *clusters* en les ramenant à la taille de N/M. Nous cédons l'excédent des échantillons des *clusters* dépassant N/M aux *clusters* petits. Pour cela, nous nous appuyons sur le caractère génératif des Modèles à Distribution de Points. Les formes des échantillons dont la classe est changée en m sont alors générées par rapport au centroïde des formes issu du *clustering* : $S_{n,t,i} \leftarrow \hat{S}_n - \overline{\Delta S}_{m,t,i}$. Sur cette base rééquilibrée, nous expérimentons le classifieur multiclasse Forêt Aléatoire, introduite par [Breiman, 2001]. Les Forêts Aléatoires sont un ensemble d'arbres aléatoires, chacun construit avec un sous-ensemble aléatoire d'échantillons et de caractéristiques.

3.6.2.3 Caractéristiques et indexations

Nous proposons d'utiliser comme caractéristiques visuelles les *Integral Channel Features* présentées Section 2.4.1. Nous pensons que l'efficacité de ces caractéristiques dans une tâche de détection peut être mise à profit pour l'alignement. Le système précédent utilisait l'apparence en bordure des caractéristiques HOG locales afin de guider l'alignement. Une faiblesse de cette approche survenait lorsque la partie à atteindre ne recoupait pas la zone occupée par la caractéristique. Pour garantir l'occupation de la zone d'intérêt par la caractéristique, nous employons une astuce consistant à compléter le maillage de base avec des points périphériques. Nous plaçons 8 points autour du rectangle défini par S : $\{\min x_k; \min y_k; \max x_k; \max y_k\}$ à une marge que nous fixons à 20 pixels. Une représentation de ce maillage complété est donnée Figure 3.22b.

En terme d'indexation par rapport à la forme, nous utilisons également pour ce système la transformation affine par parties : $W(x, y, S)$ (transformation affine par parties des pixels contenus dans la forme $\bar{S}_0$ vers S). Nous proposons, en plus de ramener les coordonnées de la caractéristique sur la forme courante, une deuxième approche basée sur la projection de l'image sur $\bar{S}_0$ en utilisant le maillage englobant comme support. Ceci revient à faire la transformation affine par parties inverse. L'image projetée est notée : $I(W^{-1}(x, y, S))$.

Cette opération rend possible un placement cohérent des caractéristiques sur des zones pouvant être potentiellement éloignées de la forme initiale, ce qui est souvent le cas sur les premières étapes de la cascade. Nous guidons ce placement par rapport aux données d'apprentissage. Nous évaluons la zone recouverte sur $\bar{S}_0$ par le sous-ensemble d'amers comme illustré Figure 3.22c. Les caractéristiques sont générées sur cette localisation pour être ensuite sélectionnées par *AdaBoost*.

Nous tirons également parti de la structure en double cascade. En effet, l'image projetée correspond au support d'apparence $A_{n,t}$. Par conséquent, son calcul n'intervient que sur l'étage supérieur de la cascade, permettant d'économiser du temps de calcul vis-à-vis de l'opération coûteuse qu'est la transformation affine par parties.

3.6.2.4 Évaluations et validations

Notre système reprend les mêmes critères d'évaluation que ceux présentés Section 3.5.3.1. Nous utilisons comme forme d'initialisation $\bar{S}_0$. Les modèles sont entraînés par défaut sur la base complète d'apprentissage, sans le découpage utilisé pour les simulations. Nous utilisons ce découpage dans les cas où une partie validation est nécessaire (comme, par exemple, pour la perturbation des *clusters* d'entraînement) ou lorsque nous évaluons l'algorithme sur sa phase d'apprentissage.

Paramètres

Nous utilisons les paramètres par défaut suivants :

Taille de la double cascade :
- Nombre de régresseurs de l'étage supérieur : $T = 5$
- Nombre de régresseurs primitifs de l'étage inférieur : $I = 100$

Caractéristiques *Integral Channel Features* :
- taille de 4 à 14 pixels (en hauteur et largeur)
- 6 *channels* de gradients orientés + 1 *channel* de magnitude + 3 *channels* CIELUV
- lues sur $I(W^{-1}(x, y, S))$

Clustering des déformations :
- Algorithme K-means++
- Considération par parties (Figure 3.22a)
- $M = 16$ *clusters* de déformations (donc un *clustering* dichotomique sur 4 niveaux)
- Coefficient de rétrécissement : $\beta = 1000$

Classifieur *AdaBoost* sur *Clustering* dichotomique :
- 100 classifieurs faibles de type arbre de décision à 2 niveaux
- 50 candidats
- Entraîné sur 4/5 des échantillons

Classifieur Forêts Aléatoires :
- 100 arbres aléatoires
- Profondeur maximale de chaque arbre fixée à 5
- 200 caractéristiques utilisées
- Entraîné sur base rééquilibrée

Classification des *clusters* de déformations

Dans un premier temps, nous évaluons l'apport général d'une discrimination entre les déformations par rapport à la *baseline* qui s'appuie sur la corrélation entre les caractéristiques et les déformations [Cao et al., 2013]. Nous augmentons la taille de la cascade de la *baseline* avec $I = 300$ régresseurs primitifs et $T = 5$ étapes, car nous constatons une convergence plus lente de l'erreur d'entraînement vers 0. La Figure 3.24a montre que notre algorithme obtient des résultats d'alignement supérieurs à l'ESR (moyenne RMSE de 3.36 contre 4.49), mais également à notre version précédente sur modèle paramétrique. Concernant l'algorithme ESR, il est intéressant d'observer la répartition dans les groupes de déformations formés par les *Random Ferns* (dont la profondeur est également fixée à 4). Nous utilisons comme caractéristiques les *Integral Channel Features*, sélectionnées par une corrélation maximale avec les déformations. En nous plaçant sur la première itération, nous obtenons pour l'ESR une moyenne de 4 feuilles (avec au moins plus d'un échantillon). Un déséquilibre important est observé puisque la feuille la plus grande contient environ 75 % de tous les échantillons. Au niveau de l'utilisation des Forêts Aléatoires et d'*AdaBoost*, nous obtenons bien les 16 *clusters* attendus de façon mieux répartie. Un exemple de cette répartition est donné Figure 3.23.

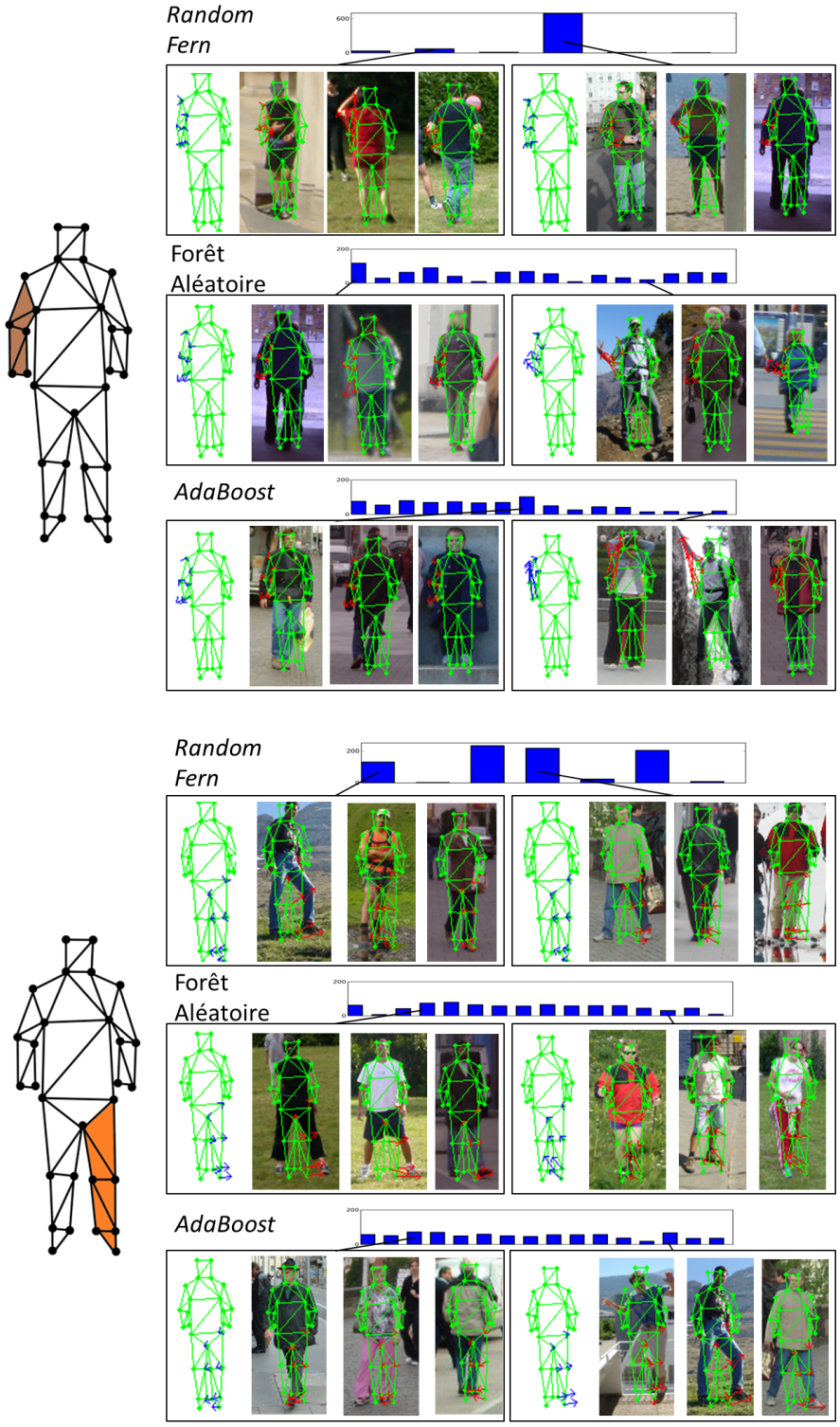
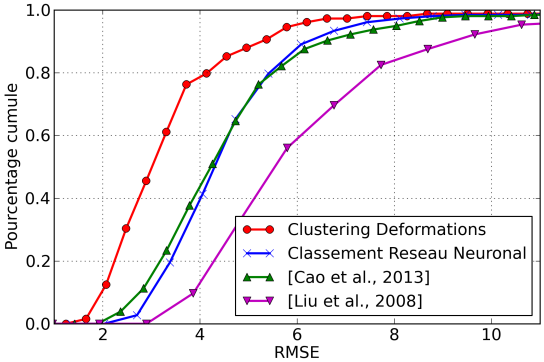
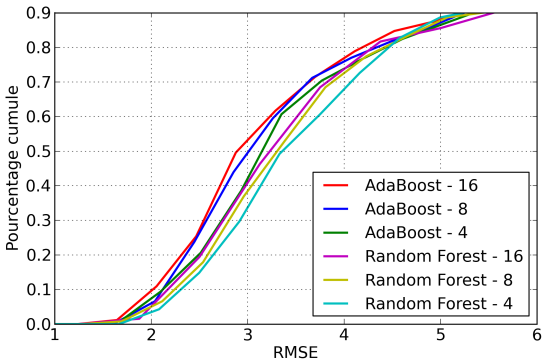


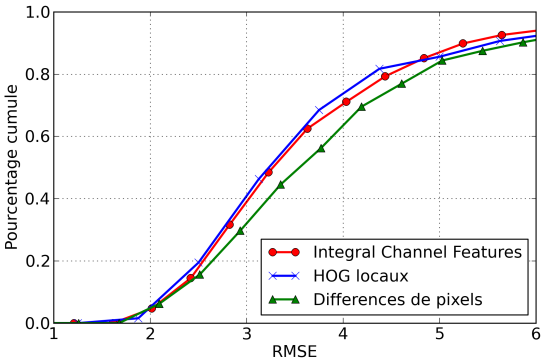
FIGURE 3.23 – Illustration des *clusters*/groupes de déformations obtenus à la première itération de l'algorithme pour le bras gauche et la jambe droite. Les flèches en bleu représentent la moyenne des déformations au sein du *cluster* $\overline{\Delta S}_{m,0,0}$. Les flèches en rouge représentent les déformations de chaque échantillon $\Delta S_{n,0,0}$. Pour plus de lisibilité, nous avons seulement représenté les flèches de la partie considérée et non de la forme globale.



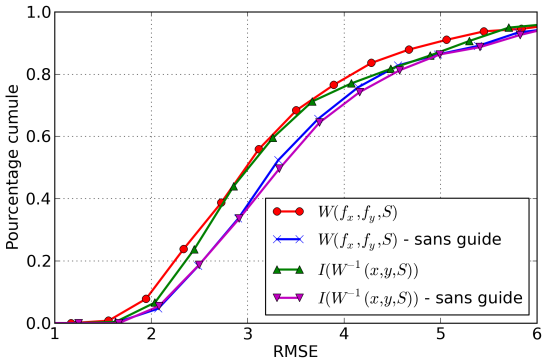
(a) Travaux existants



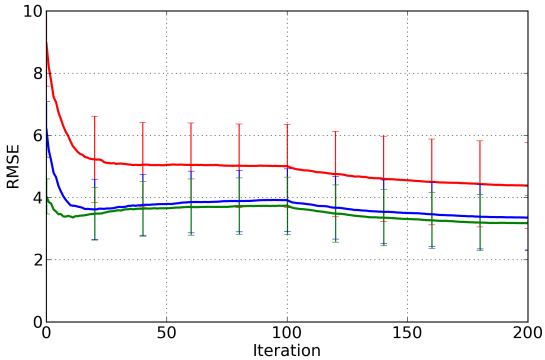
(b) Approche de classification et nombre de *clusters*



(c) Caractéristiques



(d) Lecture des caractéristiques



(e) Évolutions sur Translation/Échelle

FIGURE 3.24 – Évaluations de l'algorithme cascade de régressions de forme sur le *clustering* et la classification des déformations.

Nous proposons également d'évaluer la disparité des *clusters* obtenus grâce à la mesure WCSS (*within-cluster sum of squares*), qui est la fonction coût à minimiser du K-Means et qui donne une indication sur la compacité des *clusters* en terme de distances euclidiennes. Nous reportons les résultats Table 3.3 en omettant la considération par parties. Pour forcer la construction de *Random Ferns* équilibrés, nous appliquons sur chaque feuille le seuillage occupé par l'échantillon médian. Ce tableau montre que le cas des *Random Ferns* entraîne la formation de groupes plus disparates. La transformation du problème en classification de déformations permet bien l'obtention de groupes plus homogènes, délivrant un incrément cohérent de déformations. Nous illustrons la composition de ces groupes Figure 3.23. On peut voir une meilleure répartition dans le cas d'une classification des

groupes de déformations, ainsi qu'un incrément de plus forte amplitude. De plus, nous constatons une tendance générale des déformations qui suit celle donnée par l'incrément. En terme d'erreur de classification, les Forêts aléatoires possèdent sur la base équilibrée une erreur d'apprentissage de 0% et une *out-of bag error* de 75%. Leurs performances sur la véritable base sont une erreur de classification aux alentours de 20% et une erreur de validation de 81%. Les *AdaBoost* binaires obtiennent par nœud et sur les 3 premiers étages de l'arbre une erreur moyenne d'apprentissage de 18% (0% sur le sous-ensemble d'apprentissage) et 37% d'erreur de validation. Les résultats d'alignement liés à ces performances de classification recoupent les ordres de grandeur donnés par la simulation Section 3.6.2.1.

	Random Ferns	K-Means++	Forêts aléatoires	K-means dichotomique	*AdaBoost*
$\sum^{M}$ WCSS	1 306 163	830 283	1 034 135	777 616	1 039 453

Table 3.3 – Mesure de la compacité des groupes de déformations sur la première itération de l'algorithme. Plus l'indice WCSS est faible, plus les déformations au sein des groupes sont proches du centroïde.

Parmi les deux approches pour pouvoir gérer le déséquilibre des données, la gestion d'un *clustering* dichotomique en combinaison avec plusieurs *AdaBoost* binaires semble fournir des résultats légèrement meilleurs (Figure 3.24b). Au niveau du nombre de *clusters* considérés, 8 et 16 donnent des résultats similaires, tandis que nous constatons une dégradation dès lors que nous réduisons le nombre de *clusters* à 4. Par ailleurs, plus le nombre de *clusters* augmente, plus la convergence d'alignement sera rapide en terme d'itérations. Il faut cependant garantir un nombre suffisant d'échantillons à répartir entre les *clusters* pour la phase d'apprentissage. Sur la configuration *AdaBoost*, nous constatons une baisse de performance de -0.12 de RMSE en passant de 16 *clusters* à 32.

Nous évaluons également l'intérêt d'utiliser une approche par parties dans le but de spécialiser spatialement les classifieurs à chaque itération. Une considération sur la globalité des déformations réduit les performances de la RMSE moyenne de 0.36, (3.81 par rapport à 3.45 sur *AdaBoost* - 8 *clusters*).

	$T = 2$, $I = 250$	$T = 5$, $I = 100$	$T = 10$, $I = 50$	$T = 20$, $I = 25$
Moyenne RMSE	3.51	**3.45**	3.48	3.48

Table 3.4 – Performances en fonction du paramétrage de la double cascade sur *AdaBoost* - 8 *clusters*.

Comme indiqué par la Table 3.4, le paramétrage de la double cascade ne semble pas avoir une influence significative sur les performances globales du système. Néanmoins, paramétrer un étage supérieur avec T faible permet d'économiser du temps de calcul pour former $A_{n,t}$, c'est pourquoi nous fixons $T = 5$ et $I = 100$. La taille globale de la cascade (actuellement paramétrée sur 500 itérations) ainsi que le coefficient β ont été déterminés sur la base de la simulation qui permet d'obtenir une convergence suffisante d'alignement.

Caractéristiques

Nous poursuivons nos expérimentations en comparant les performances de trois types de caractéristiques : les *Integral Channel Features*, les HOG locaux utilisés dans le précédent système, et les différences de pixels sur l'image en niveau de gris. Cette comparaison est faîte avec les forêts aléatoires sur 16 *clusters*. Nous pouvons voir Figure 3.24c que les HOG et les *Channel Features* donnent des résultats équivalents, l'information exploitée étant similaire (gradients orientés). Nous préférons l'utilisation des *Channel Features* car ces dernières sont plus rapides à calculer. Sans grande surprise, les différences de pixels donnent des résultats inférieurs et ne représentent pas en l'état une

caractéristique viable pour le corps humain. Par rapport à son cadre d'application habituel qu'est le visage, le corps humain n'est pas un objet convexe. Ainsi, cette caractéristique va extraire des valeurs provenant du décor, revenant à s'appuyer sur du bruit. Cependant, elle présente l'avantage de pouvoir décrire des relations d'apparence entre deux points éloignés dans l'image, ce qui n'est pas le cas des caractéristiques locales de type blocs. Une piste à exploiter est la recherche de caractéristiques pouvant décrire des corrélations spatiales dans l'image tout en étant robustes face au décor.

Par la suite, nous évaluons quelle est la meilleure façon d'indexer les caractéristiques à la forme courante. Deux possibilités sont abordées. La première, jusqu'alors utilisée dans ces expérimentations, est la lecture des caractéristiques sur l'image transformée vers $\bar{S}_0$: $I(W^{-1}(x, y, S))$. Une autre possibilité est de transformer les coordonnées des caractéristiques, référencées sur $\bar{S}_0$, vers $S_{n,t,i}$, ce qui revient à l'opération $W(f_x, f_y, S)$ avec f_x et f_y les coordonnées de la caractéristique. Cette opération était utilisée pour notre premier système, sans toutefois l'étendre à un maillage complété par des points englobants. La Figure 3.24d montre que ce type d'indexation donne des résultats légèrement meilleurs : RMSE moyenne de 3.36 contre 3.45. Cela peut s'expliquer par une dégradation de l'information à cause des distorsions résultant de la transformation affine par parties, marquées par une discontinuité entre les triangles. Néanmoins, l'avantage d'une image projetée est de gagner une certaine robustesse vis-à-vis de l'échelle de la forme courante. En effet, l'opération redimensionne l'apparence contenue dans la forme à la taille $\bar{S}_0$, permettant de gagner une cohérence d'échelle lors de la lecture des caractéristiques. Ceci peut s'avérer avantageux dans le cas d'un résultat de détection à une échelle incorrecte. Par ailleurs, nous voyons que guider le placement des caractéristiques par rapport aux emplacement des parties de la base d'apprentissage améliore également l'alignement.

Détection imprécise

Nous évaluons la capacité de notre approche face à une détection imprécise de la même façon que le précédent système. La cascade est augmentée de 2 étages préliminaires de correction, composés de 100 régresseurs primitifs. Ces 2 étages considèrent comme vérité terrain la forme $\bar{S}_0$ centrée et à la bonne échelle. 3000 échantillons supplémentaires ont générés avec des positions et échelles perturbées. Puis, l'algorithme passe sur l'apprentissage d'alignement des déformations en s'appuyant sur les formes résultant de la phase de correction de translations et d'échelles. En comparant les courbes de la Figure 3.24e et celles de la Figure 3.16e, nous constatons que cette approche permet de nettement mieux corriger une détection imprécise. Par ailleurs, nous obtenons des résultats similaires avec l'alignement sur une phase intermédiaire ($\bar{S}_0$ centrée et à la bonne échelle) par rapport à un alignement direct vers $\hat{S}_n$ depuis une position perturbée.

Temps de calcul

Le temps de calcul va dépendre principalement de la taille de la double cascade, du nombre de fois pour lequel le support d'apparence doit être généré (T), du nombre de caractéristiques (et de leur lecture sur l'image) utilisées par chaque régresseur primitif, ainsi que du temps de prédiction de chaque régresseur. Avec un *AdaBoost* sur 8 *clusters* et la projection des coordonnées des caractéristiques sur la forme courante, l'alignement fonctionne aux alentours de 12.5 FPS sur PC. Avec l'ajout d'une phase de correction d'une détection perturbée, notre implémentation tourne à 9.5 FPS. S'il s'agit des coordonnées qui sont projetées, il est possible d'économiser la transformation de l'image en *channels* en réutilisant ceux générés lors de la phase de détection. Autrement, une réutilisation des *channels* n'est pas possible puisque ces derniers doivent s'appuyer sur $A_{n,t}$. Néanmoins, la conception en double cascade limite le nombre de générations des *channels* à effectuer en fixant T à une faible valeur. Pour améliorer les temps de calcul, une sorte d'élagage de la cascade est aussi à considérer.

3.6.3 Discussions

Ce deuxième système diminue l'erreur d'alignement d'environ 30% par rapport au premier proposé. Les poses pour lesquelles l'alignement est le meilleur sont celles associées à une attitude de marche piétonne. Par contre, le système ne parvient pas à aligner correctement le PDM dans le cas des poses atypiques, notamment pour les bras (par exemple quand les bras sont levés pour faire signe à la caméra). Ce genre de poses est illustré Figure 3.25b. Sachant que les algorithmes présentés reposent sur des apprentissages statistiques, cette faiblesse peut s'expliquer par la faible représentation de ce type de poses au sein de la base d'apprentissage. En effet, nous recensons seulement 24 images annotées, sur les 408 servant à l'apprentissage, soit 6%, possédant une pose que l'on peut qualifier de complexe. À titre d'exemple, les plus récentes bases de visages utilisées comme *300-W* sont composées de 3837 images annotées, que [Ren et al., 2014] divisent en 3148 images d'entraînement et 689 de test. En comparaison, notre base de personnes annotées ne contient que 537 images au total.

Néanmoins, malgré cette difficulté à gérer les poses complexes, le modèle appris demeure exploitable pour la ré-identification puisque les poses adoptées dans ce contexte sont naturellement associées à la marche.

(a) Alignements corrects

(b) Alignements incorrects

FIGURE 3.25 – Exemples d'alignement de l'approche par *clustering* et classification des déformations. La forme en rouge représente l'initialisation et celle en vert le résultat de l'alignement.

Chapitre 4

Ré-identification de personnes

Sommaire

4.1 Introduction **77**
4.2 État de l'art **78**
4.3 Renforcement de signatures basées sur la couleur **81**
4.3.1 Histogramme de couleurs HSV 81
4.3.2 Modèle d'ensemble de couleurs 81
4.3.3 Renforcement structurel 82
4.3.3.1 Évaluations 82
4.4 Signature de forme sur les amers d'un PDM **85**
4.4.1 Shape Context 85
4.4.2 Évaluations et discussions 86

4.1 Introduction

Ce chapitre est consacré à la dernière étape du système présenté dans cette thèse : le module de ré-identification de personnes. L'objectif de la ré-identification est de retrouver une correspondance entre l'acquisition actuelle d'une personne avec une ou plusieurs instances acquises d'elle, soit précédemment, soit à la volée (cas de caméras dont le champ de vision se recoupe). Son principal domaine d'application est la vidéo-surveillance au moyen d'un réseau de caméras fixes. Nous proposons, dans cette thèse, une extension de ce domaine d'application aux systèmes mobiles, sans l'appui classique de méthodes à base de soustraction d'environnements.

Les méthodes de ré-identification peuvent être classées en deux catégories. La première s'appuie sur des informations biométriques. La biométrie signifie littéralement la mesure du vivant, et désigne les systèmes de reconnaissance s'appuyant sur des caractéristiques possédées par une population, mais distinctes entre chaque individu. La reconnaissance faciale est une technique biométrique classique. Elle requiert une qualité d'acquisition suffisante. Il peut s'agir de la coopération des individus, du contrôle de l'environnement d'acquisition, ou d'images de résolutions suffisantes. L'émergence de capteurs spécifiques, tels que les caméras thermiques [Hermosilla et al., 2012], permet néanmoins de porter son application dans des environnements non contraints. Une autre donnée biométrique pouvant être exploitable est la reconnaissance des démarches. Cette dernière présente l'avantage de pouvoir s'accommoder des contraintes imposées par les systèmes de vidéo-surveillance (non intrusif et faible résolution) mais est sensible au changement de vêtements et aux points de vue de l'acquisition [Yu et al., 2006].

La seconde catégorie s'appuie sur l'apparence globale de la personne. Les systèmes de cette catégorie font l'hypothèse que la personne conserve les mêmes vêtements d'une acquisition à l'autre. Ces méthodes sont notamment utilisées pour le *tracking* de personnes dans les aéroports ou le métro. Un défi important de ces méthodes réside dans le fait que les caractéristiques visuelles utilisées ont un faible pouvoir distinctif pour établir des correspondances entre les personnes. L'apparence de

la même personne peut grandement varier selon les conditions d'acquisitions, que ce soit en terme d'illumination, de point de vue ou d'occultation (Figure 4.1).

Nous abordons un renforcement des techniques de ré-identification de la seconde catégorie avec le Modèle à Distribution de Points. Ce chapitre s'organise de la façon suivante. Nous présentons un état de l'art des techniques de ré-identification basées sur l'apparence globale des personnes, en abordant principalement les travaux s'appuyant sur un support structurel. Dans un second temps, nous présentons deux approches basées sur l'apparence renforcée par un PDM. Puis, nous expérimentons un descripteur de forme en nous basant uniquement sur les amers.

FIGURE 4.1 – Illustration des défis rencontrés lors de la ré-identification sur la base VIPeR [Gray and Tao, 2008]. Malgré une variation d'apparence significative, l'algorithme doit être capable de faire correspondre les paires d'images de la même personne entre elles (ligne du haut et du bas).

4.2 État de l'art

Les techniques de ré-identification suivent le schéma de fonctionnement global suivant : (1) l'extraction de caractéristiques robustes et discriminantes formant une signature (2) comparaison des signatures entre elles par une distance, potentiellement dans un espace métrique pouvant être l'objet d'un apprentissage.

De nombreux types de caractéristiques et descripteurs visuels ont été abordés pour la ré-identication. Comme énoncé précédemment, ils s'appuient en grande partie sur l'apparence des vêtements. Deux grandes familles de caractéristiques peuvent être distinguées : les caractéristiques locales et les caractéristiques globales.

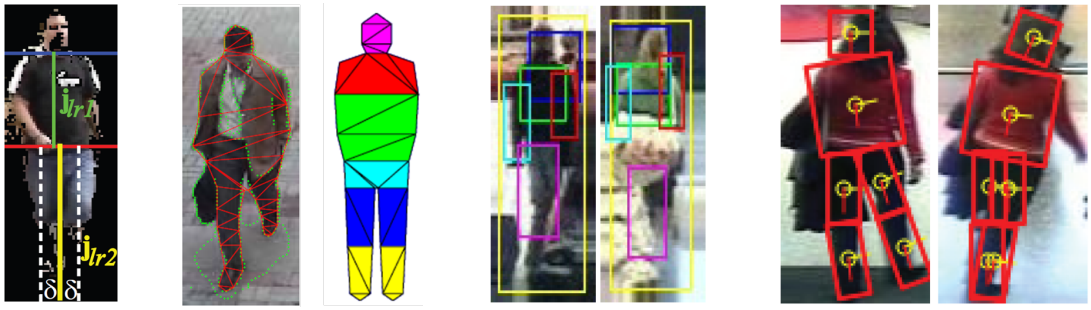
Les caractéristiques locales font référence à l'apparence d'une petite région de l'image. La détermination de cette région peut être faite par une subdivision dense de l'image, par un opérateur d'intérêt, ou par une sélection grâce à un algorithme d'extraction de caractéristiques. Les points d'intérêt sont des caractéristiques locales classiques. Les plus connus sont les points SIFT (*Scale Invariant Feature Transform*) [Lowe, 2004] et leurs variantes, telles que les points SURF (*Speeded-Up Robust Features*) [Bay et al., 2006]. Ces points sont décrits par un histogramme de bords orientés contenu dans une petite fenêtre centrée sur le dit-point. [Hamdoun et al., 2008] utilisent l'opérateur de points Camellia [Bdiri et al., 2009] et le descripteur SURF afin de générer une signature de la personne. Cette signature peut être renforcée sur plusieurs images successives, ce qui améliore les performances de reconnaissance. [Zhao et al., 2013] s'appuient quant à eux sur une subdivision en grille de l'image afin de générer un ensemble dense de caractéristiques SIFT et d'histogrammes de

couleurs. Leur objectif est de retrouver par la suite les correspondances spatiales des caractéristiques sur deux images grâce à un apprentissage non supervisé basé sur la saillance. [Khedher and Yacoubi, 2015] proposent une ré-identification en deux étapes basée sur de l'appariement entre points d'intérêt de type SURF. La première étape consiste en un filtre des points d'intérêt fiables grâce à un classifieur binaire entraîné pour discriminer les appariements corrects et incorrects de points. La seconde étape consiste à trouver une correspondance parmi les références de personnes grâce à une représentation clairsemée (*sparse*) sur un dictionnaire sélectionné dans le voisinage spatial de chaque point d'intérêt.

D'autres caractéristiques locales s'appuient sur des régions stables de couleur dans l'image : les *Maximally Stable Colour Regions* (MSCR) [Forssén, 2007]. Ces régions, stables par rapport à l'échelle et aux transformations affines, sont déterminées grâce à un *clustering* agglomérant des pixels voisins ayant des couleurs similaires. Les *Recurrent Highly-Structured Patches* (RHSP) sont, quant à elles, des caractéristiques locales s'appuyant sur la texture et cherchent à décrire les motifs et textures récurrents contenus dans l'apparence des vêtements. [Gray and Tao, 2008] ont défini un espace de caractéristiques et ont laissé *AdaBoost* choisir les plus discriminantes. Cet espace de caractéristiques locales (ELF) est constitué de *channels* (couleurs et textures), d'une région sur l'image et d'une classe d'histogramme. Un résultat intéressant est le placement des caractéristiques sélectionnées par *AdaBoost* qui se trouvent au 3/4 dans les *channels* de couleurs, démontrant leur importance dans le cadre de la ré-identification.

La seconde famille de caractéristiques est celle des caractéristiques globales. Elles sont évaluées sur le corps entier ou une grande région du corps. Les plus communes de ces caractéristiques sont les histogrammes de couleurs. Nous reviendrons en détail sur leurs constructions ultérieurement. Les histogrammes de couleurs présentent l'avantage d'être invariants par rapport à l'échelle, mais sont sensibles à la luminosité et aux réponses aux couleurs des caméras. Pour répondre à ces problèmes, [Porikli, 2003; Javed et al., 2008] estiment la fonction de transfert de luminosité (*Brightness Transfer Function*, BTF) afin de modéliser les changements d'apparence des objets entre deux caméras. Autrement, un moyen simple à mettre en œuvre s'appuie sur la normalisation des couleurs [van de Sande et al., 2010]. Il est également possible d'utiliser l'égalisation de l'histogramme proposée par [Finlayson et al., 2005], qui font l'hypothèse qu'un changement d'illumination préserve l'ordre dans les valeurs de l'histogramme. Plus récemment, [Kviatkovsky et al., 2012] ont proposé une autre approche pour gérer ces changements de couleurs. Au lieu de considérer les couleurs en valeur absolue, leur idée est d'exploiter la distribution relative des couleurs entre les parties haute et basse de la personne.

Outre ces problématiques de dissimilarités de couleurs, un inconvénient de ces histogrammes est qu'ils ne retiennent pas l'information de disposition spatiale des couleurs. Nous en venons au point sur lequel nous apportons notre contribution : l'utilisation d'un modèle structurel particulier afin d'établir les correspondances de régions. Une solution simple utilisée par [Park et al., 2006] est de découper l'image verticalement en 3 blocs et de calculer sur chacun de ces blocs l'histogramme de couleurs correspondant. Une structure un peu plus élaborée proposée par [Farenzena et al., 2010] est de considérer un axe de symétrie horizontale à partir duquel des caractéristiques locales sont pondérées suivant leur distance à cet axe. [Gheissari et al., 2006] appliquent un modèle déformable, consistant en un graphe triangle décomposable, qu'ils alignent sur les personnes. Ce modèle est aligné grâce à une fonction de coût s'appuyant sur la saillance dans l'image et un masque issu d'une segmentation spatio-temporelle faîte au préalable. Il est par la suite partitionné en 6 sous-parties et démontre des résultats supérieurs de ré-identification, en tant que support, face à la segmentation seule ($\sim$+25%). [Bak et al., 2010b] appliquent plusieurs détecteurs par parties et génèrent une signature basée sur un descripteur de covariance entre ces parties. [Cheng and Cristani, 2014] exploitent les *Pictorial Structures* dans le cadre de la ré-identification. Ils modifient leur formulation de façon à tirer parti de plusieurs images acquises de la personne afin de renforcer l'alignement du modèle. Par la suite, ils construisent pour chaque partie alignée, une signature basée sur les histogrammes de couleurs et sur les MSCR. Un aperçu de ces différentes considérations



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

FIGURE 4.2 – Illustration des principaux modèles structurels non figés utilisés dans la ré-identification. Les images sont tirées des papiers originaux. (a) Symétries verticales [Farenzena et al., 2010] (b) Graphe triangle [Gheissari et al., 2006] (c) Détecteur de parties [Bak et al., 2010b] (d) *Pictorial Structures* [Cheng and Cristani, 2014].

spatiales est donné Figure 4.2.

D'autres caractéristiques peuvent également être utilisées pour la ré-identification. [Layne et al., 2012] s'appuient sur des caractéristiques de plus haut niveau qui décrivent de façon sémantique les personnes (telle que la possession d'un sac, le sexe de la personne, de longs cheveux, etc.). Ces attributs sont déterminés par des caractéristiques de textures et de couleurs et sont appris avec une machine d'apprentissage SVM. [Wang et al., 2007] exploitent quant à eux la forme de la personne avec les caractéristiques *Shape Context* introduites par Belongie et al. [2002]. Ils segmentent la silhouette humaine en différentes parties grâce à un algorithme modifié de *Shape Context* basé sur un dictionnaire appris au préalable. Une représentation de type sac de mots est utilisée pour définir l'apparence en se basant sur des caractéristiques HOG calculées dans l'espace log-RGB. Puis, ils couplent l'apparence définie par le sac de mots et les résultats issus de la segmentation avec des matrices de co-occurrence. Cela leur permet de capturer les relations spatiales entre les différentes régions d'apparence. Enfin, les travaux récents de [Barbosa et al., 2012] mettent à profit l'émergence des capteurs de profondeur tels que la Microsoft Kinect pour générer une signature anthropométrique de la personne et affichent des résultats prometteurs. Néanmoins, l'utilisation d'un tel système dans un environnement non contraint peut poser des difficultés principalement sur l'extraction du squelette ou de la forme 3D de la personne.

La seconde problématique de la ré-identification est la technique de comparaison des signatures. Initialement, les signatures étaient comparées par des méthodes fixes, telles que la distance de Bhattacharyya pour mesurer la similarité entre deux histogrammes [Javed et al., 2005], ou encore des techniques de K plus proches voisins [Hahnel et al., 2004]. Un domaine émergent est l'apprentissage d'un espace métrique qui a permis d'améliorer significativement les résultats de reconnaissance. Il s'agit d'apprendre la distance appropriée maximisant les performances de reconnaissance du système sans s'attacher à la caractéristique considérée. Cette problématique revient à trouver une métrique pour laquelle les caractéristiques des objets appartenant à la même classe soient proches et celles des objets de différentes classes soient éloignées. On peut notamment citer comme technique la LMNN (*Large margin nearest neighbor*) [Weinberger et al., 2006], dérivant de la classification des plus proches voisins, la comparaison par distance relative [Zheng et al., 2013] ou encore la KISS métrique [Koestinger et al., 2012]. Les travaux de [Liu et al., 2015] utilisent cette dernière métrique et combinent de nombreuses caractéristiques de couleurs, leur permettant d'atteindre de très bons résultats de ré-identification.

4.3 Renforcement de signatures basées sur la couleur

Nous proposons dans un premier temps d'évaluer l'apport d'un PDM avec une signature basée sur la couleur, et plus particulièrement les histogrammes de couleurs. Nous anticipons une amélioration des résultats puisque ces derniers perdent intrinsèquement l'information spatiale lors de leur construction.

4.3.1 Histogramme de couleurs HSV

Nous expérimentons tout d'abord une signature classique et simple basée sur un seul type d'histogramme de couleurs, celui calculé dans l'espace de couleur HSV. Cet espace de couleur s'inspire de la façon dont le cerveau humain perçoit les couleurs. Les trois canaux correspondent à :

- la teinte (*Hue*) ou la tonalité de la couleur
- la saturation (*Saturation*) ou son «intensité»
- la valeur (*Value*) ou sa «brillance»

Il existe deux possibilités pour construire un histogramme de couleurs, soit de façon monodimensionnelle, soit de façon multidimensionnelle. La première s'obtient par une quantification des valeurs de pixels en considérant séparément chaque canal de couleurs. L'histogramme est ensuite obtenu par la concaténation des distributions calculées sur chaque canal. Les histogrammes multidimensionnels effectuent simultanément la quantification des valeurs de pixels sur l'ensemble des canaux de couleur. Ainsi, si l'on considère une quantification sur N classes, 3 canaux de couleurs et R sous-régions dans l'image, l'histogramme aura dans le premier cas la taille de $N \times 3 \times R$ et dans le second cas $N^3 \times R$. Les histogrammes en mono-dimension sont plus simples à calculer et prennent moins de place en mémoire. Nous proposons d'évaluer les performances délivrées dans les deux cas en fixant le nombre de classes $N = 10$. Dans le but de rendre cette signature plus robuste face aux variations de luminosité, nous égalisons au préalable l'histogramme du canal V. De façon similaire à [Javed et al., 2005], nous proposons d'utiliser la distance de Bhattacharyya afin de mesurer la similarité entre les signatures.

4.3.2 Modèle d'ensemble de couleurs

Nous expérimentons l'apport d'un appui structurel avec un second modèle d'apparence plus complet. Nous avons choisi le modèle d'ensemble de couleurs (*Ensemble Color Model* ou ECM) proposé par [Liu et al., 2015] qui figure actuellement parmi les plus performants de l'état de l'art et dont nous résumons succinctement le principe.

La spécificité de ce modèle est qu'il utilise de nombreuses informations provenant exclusivement de la couleur. Les auteurs mixent un ensemble de caractéristiques répondant à des propriétés globales. Ainsi, nous retrouvons les histogrammes RGB, HSV et Lab pour leurs différentes propriétés photométriques. Au niveau des caractéristiques conçues pour être invariantes par rapport à la luminosité, les auteurs proposent d'utiliser les caractéristiques suivantes : RGB normalisé, la teinte, les couleurs opposées et les moments des couleurs. Enfin, la troisième catégorie concerne la sémantique des couleurs (noms que nous donnons pour décrire les couleurs) et s'appuie sur l'étude de [van de Weijer et al., 2007] dont les noms de couleurs ont été appris depuis un grand nombre d'images. Ces histogrammes sont calculés sur 6 régions résultant d'un découpage vertical uniforme sur la zone définie par le masque issu d'une segmentation.

Afin d'améliorer la similarité entre les caractéristiques d'une même personne et de mieux discriminer celles de personnes différentes, les auteurs recourent à une technique d'apprentissage de distance. Ils utilisent la métrique proposée par [Koestinger et al., 2012] qui s'appuie sur la famille de distances de Mahalanobis. Cela leur permet d'obtenir un ensemble de mesures de similarités pour chaque type de caractéristiques de couleurs. Ensuite, au lieu de simplement concaténer ces mesures,

les auteurs proposent de définir les affinités entre les types d'apparences comme une combinaison linéaire. Les poids de cette dernière sont appris grâce à un SVM structuré [Joachims et al., 2009].

4.3.3 Renforcement structurel

Nous proposons de renforcer les deux types de modèles d'apparence que nous venons de présenter avec le support d'un PDM. Nous alignons dans un premier temps la forme sur l'image et projetons l'apparence sur la forme moyenne. Puis, nous calculons les histogrammes de couleurs à l'intérieur des régions définies par les triangles du maillage. Par rapport à un découpage vertical de l'image segmentée, l'utilisation de ces régions permet de ne pas mélanger l'apparence entre les différentes parties contigües du corps. Ce modèle permet par exemple de séparer proprement l'apparence du torse et celle du bas du corps.

En se basant sur la représentation que nous avons définie Figure 3.5a, nous expérimentons deux ensembles de sous-régions. Le premier est directement constitué des triangles composant le maillage, formant alors $R = 27$ sous-régions dans lesquelles nous générons les signatures d'apparence. Cependant, suivant la pose de la personne, l'alignement peut engendrer des triangles de très faibles tailles, ce qui va augmenter la sensibilité au bruit de cette sous-région. C'est pourquoi, nous proposons également de regrouper arbitrairement les triangles pour former 12 sous-régions de taille plus importante comme indiqué par la Figure 4.3.

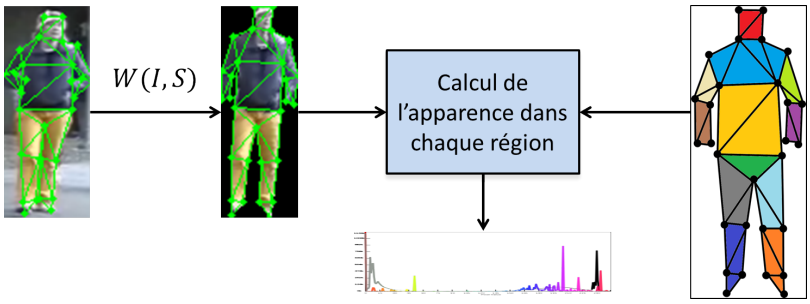
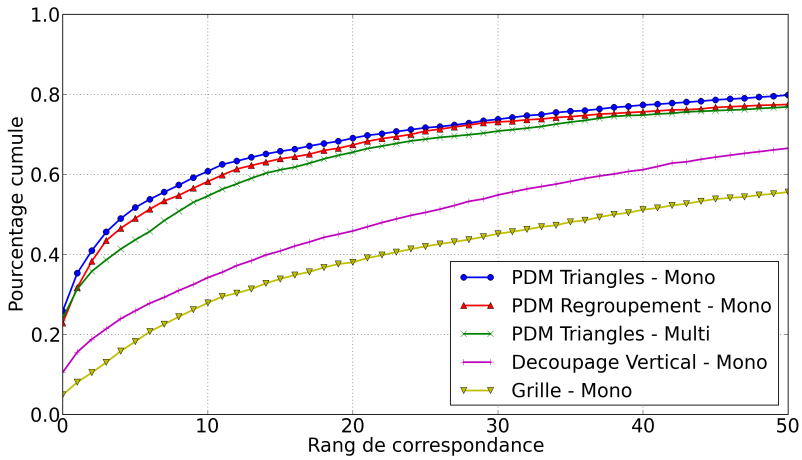


FIGURE 4.3 – Application du Modèle à Distribution de Points à la ré-identification. Chaque couleur du modèle de droite représente une région dans laquelle est calculée la signature d'apparence.

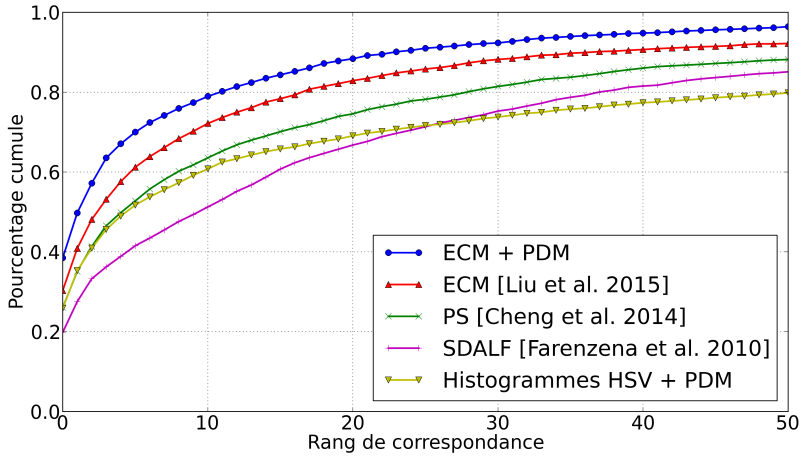
4.3.3.1 Évaluations

Pour évaluer les modèles d'apparence proposés, nous utilisons la base de ré-identification VIPeR [Gray and Tao, 2008]. Cette base, acquise dans des conditions naturelles, est connue pour représenter un important défi pour la ré-identification. En effet, les images sont de faibles résolutions et présentent des points de vue d'acquisitions et des conditions d'illumination variés. Par ailleurs, la ré-identification doit être faite dans un scénario *single-shot*, c'est-à-dire qu'elle ne peut s'appuyer que sur une seule image, et non pas, par exemple, sur une séquence vidéo. Cette base contient 632 personnes, ou identités, représentées par deux images.

Nous représentons nos résultats de reconnaissance avec la courbe de correspondance cumulée (*Cumulative Matching Characteristic* ou CMC) qui s'avère être une métrique d'évaluation performante [Gray and Tao, 2008]. Elle se déduit des correspondances correctes cumulées, obtenues grâce aux distances triées par ordre croissant entre les échantillons sondés (*probe* en anglais) et les échantillons enregistrés (*gallery*).



(a) Histogrammes HSV



(b) Comparaison état de l'art

FIGURE 4.4 – Moyenne des courbes CMC sur la base VIPeR.

L'évaluation sur la base VIPeR se fait habituellement avec la moyenne de 10 résultats sur un sous-ensemble de 316 personnes tirées aléatoirement. Nous expérimentons dans un premier temps le premier type de signature, basée sur les histogrammes HSV. Pour évaluer le potentiel du PDM, nous avons aligné de façon semi-automatique les formes de la base avec le modèle de type régression de forme. Nous comparons Figure 4.4a les différentes considérations structurelles proposées ainsi qu'un découpage uniforme en grille de l'image segmentée (2 colonnes et 5 lignes) et un découpage vertical similaire aux travaux de [Park et al., 2006] qui utilisent les proportions correspondantes (1/5, 3/5 et 1/5). La considération avec un PDM fournit de bien meilleurs résultats : +15% au rang 1 par rapport à un découpage vertical et +20% par rapport à la grille. Par ailleurs, regrouper les triangles pour former de plus grandes parties dégrade légèrement les résultats par rapport à une considération sur l'ensemble des triangles. Nous constatons que les histogrammes monodimensionnels sont plus performants que ceux multidimensionnels, ce qui présente un double avantage puisqu'ils sont également plus rapides à calculer.

Ces résultats vont forcément dépendre de la qualité de l'alignement. En effet, un alignement incorrect va par exemple capturer l'apparence du décor, ce que nous souhaitons éviter. Lors d'un processus entièrement automatique, la Table 4.1 montre une détérioration des résultats (-8% au rang 1). Cela provient de l'erreur issue de l'alignement, mais aussi du fait que nous n'utilisons que

le modèle de face/dos car cette base contient de multiples angles de vue des personnes. De futurs travaux, intégrant la vue de profil, pourront permettre d'améliorer ces résultats.

Sur la Figure 4.4b, nous comparons cette approche avec les travaux de l'état de l'art. Nous voyons que la signature utilisant les histogrammes HSV et renforcée par un PDM donne des résultats tout à fait convenables malgré la simplicité de l'apparence utilisée. Ces performances décroissent au fur et à mesure du rang, certainement ralenties par l'utilisation d'un seul type d'information, contrairement aux autres approches telles que celles de [Farenzena et al., 2010] qui s'appuient sur des signaux complémentaires de l'image.

Le deuxième modèle d'apparence que nous expérimentons est l'ECM. Nous utilisons l'implémentation fournie par [Liu et al., 2015] et la paramétrons de sorte que le décor ne soit pas pris en compte dans le calcul d'apparence afin d'effectuer une comparaison juste. En effet, les travaux originaux s'appuient sur cette information contextuelle, sachant que de nombreux individus de VIPeR ont été acquis dans le même environnement (mais sur un point de vue différent). Excepté ce point, nous conservons les paramètres originaux avancés. La Figure 4.4b indique que nous parvenons à dépasser les résultats obtenus par l'état de l'art. Nous obtenons une amélioration de 8% au rang 1 par rapport au modèle original, confirmant l'utilité de l'utilisation d'un PDM en tant que support structurel avec des modèles complexes d'apparence tels que l'ECM.

Nous avons porté le système de ré-identification par histogrammes HSV sur smartphone et obtenons un temps de calcul de 80 FPS, négligeable par rapport à celui des modules de détection et d'alignement. L'implémentation complète du *framework* sur smartphone, utilisant les modules de détection et d'alignement, tourne aux alentours de 1.5-2 FPS.

Rang	1	10	50
ELF [Gray and Tao, 2008]	12.0	41.5	81.0
KISSME [Koestinger et al., 2012]	27.0	70.0	95.0
PS [Cheng and Cristani, 2014]	26.0	61.8	88.0
SDALF [Farenzena et al., 2010]	19.9	49.4	84.8
ECM [Liu et al., 2015]	30.3	70.2	92.1
Histogrammes HSV + PDM	25.9	59.2	79.6
ECM + PDM	38.5	77.4	96.2

Table 4.1 – Taux de correspondances en fonction du rang sur la base VIPeR.

FIGURE 4.5 – Exemples des résultats de ré-identification classés par rang croissant. Les encadrés bleus correspondent aux images sondées et les encadrés rouges correspondent à la véritable correspondance.

4.4 Signature de forme sur les amers d'un PDM

Un tout autre type de signature que nous avons expérimenté se base sur la forme décrite par le PDM. Nous utilisons alors uniquement les coordonnées des amers sans prendre en compte l'apparence à l'intérieur du maillage.

4.4.1 Shape Context

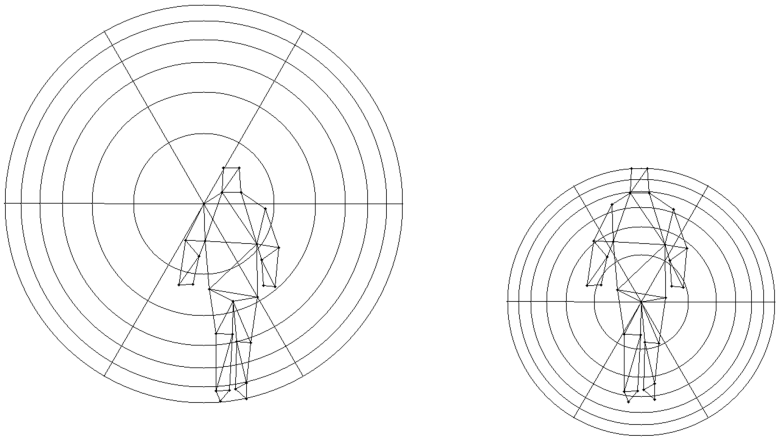


FIGURE 4.6 – Illustration sur deux points de la caractéristique de forme *Shape Context* avec adaptation de son rayon pour contenir la globalité du PDM.

Nous employons comme caractéristique de forme le *Shape Context* proposé par [Belongie et al., 2002]. Cette caractéristique est une caractéristique globale qui permet de décrire la distribution des

points composant une forme les uns par rapport aux autres. Puisque les Modèles à Distribution de Points varient fortement d'une pose à l'autre comme illustré Section 3.4.1, nous contraignons cette expérimentation à une vue seulement de face ou de dos.

Avant de calculer la signature, nous alignons toutes les formes de la base à évaluer grâce à la même analyse procustéenne présentée Algorithme 3. Cela permet de supprimer les différences d'échelle, de translation et d'orientation entre les formes. Nous conservons en mémoire l'ensemble de ces formes pour réaliser l'analyse procustéenne avec l'échantillon sondé lorsque l'algorithme marche en ligne.

La caractéristique *Shape Context* est calculée grâce à des histogrammes log-polaires centrés sur chacun des amers et comptabilisant les occurrences des points qui sont autour. Nous concaténons par la suite l'ensemble de ces histogrammes pour former la signature de la personne. Un histogramme log-polaire est défini par son rayon r, le nombre de classes angulaires C_θ ainsi que le nombre de classes le long de son rayon C_r. Ainsi la signature aura une taille de $C_\theta \times C_r \times K$, avec K le nombre de points du PDM. Pour gagner en robustesse vis-à-vis de la taille globale du PDM, nous choisissons d'adapter r suivant l'amer sur lequel l'histogramme log-polaire est centré. Nous le fixons de façon à ce qu'il soit égal à la distance avec le point le plus éloigné de l'amer considéré k : $r_k = \max_i \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}$. Ce faisant, nous tirons parti de l'avantage procuré par un PDM qui est que chaque point le composant possède un emplacement défini sur l'objet, ceci n'étant pas forcément le cas d'une forme ou d'un contour discrétisé en points.

Par la suite, nous appliquons une Analyse en Composante Principale pour diminuer la taille du descripteur. Cet algorithme permet également de supprimer les classes ne recevant jamais d'occurrences (comme, par exemple, pour la zone située au dessus de la tête lorsque l'histogramme log-polaire est centré sur cette dernière). Puis, nous comparons la signature obtenue avec la distance de Bhattacharyya.

4.4.2 Évaluations et discussions

Pour évaluer l'utilisation d'un descripteur de forme, nous avions besoin d'une base constituée de personnes de face ou de dos. Nous nous sommes tournés vers la base RGB-D Re-Identification Dataset (RGBD-ID) proposée par [Barbosa et al., 2012] qui satisfait ces besoins. Cette base a été conçue pour s'appuyer sur une ré-identification biométrique sur les distances corporelles de la personne grâce à des images de couleurs et de profondeur. Nous délaissons les images de profondeur pour nous concentrer sur les images RGB. Cette base se compose de 79 personnes, acquises dans 4 scénarios : de dos, de façon collaborative (bras levés et jambes apparentes), et deux prises séparées en train de marcher de face. Nous sélectionnons aléatoirement une image pour chaque prise de marche de face, et comme précédemment, y appliquons de façon semi-automatique l'alignement d'un Modèle à Distribution de Points.

Après validation croisée, nous trouvons comme paramètres optimaux pour l'histogramme log-polaire 6 classes sur la log-distance et 6 classes angulaires. L'application de l'Analyse en Composante Principale ramène la taille de ce descripteur $6 \times 6 \times 29 = 1044$ à 65. Nous donnons les résultats de ré-identification Figure 4.7 et les comparons à la méthode proposée par [Barbosa et al., 2012]. Cette méthode s'appuie sur des mesures anatomiques des personnes, comme par exemple, le ratio torse/jambes ou encore la distance euclidienne du sol à la tête. Ils combinent ces caractéristiques avec un jeu de poids déterminé soit uniformément, soit de façon à maximiser le score de ré-identification. Nous voyons que notre approche permet d'obtenir des performances correctes de ré-identification, en utilisant simplement les PDM alignés sur les images de couleurs. Nous obtenons en premier rang 17.7% de correspondances tandis que [Barbosa et al., 2012] obtiennent respectivement 12.7% et 8.9% pour les poids optimisés et uniformes.

Nous avons cherché à savoir si cette approche se base sur la morphométrie pour ré-identifier les personnes. Pour cela, nous avons réalisé une simple expérience en demandant la collaboration de 3 personnes à adopter une pose avec les bras levés et les jambes serrées. Nous les avons pris en photo

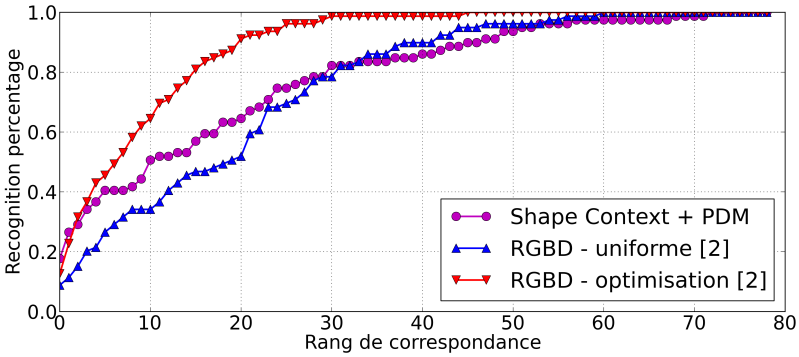


FIGURE 4.7 – Courbes CMC sur la base RGBD-ID.

sur deux vues différentes, en leur demandant de réadopter cette même pose. Nous voyons sur la Figure 4.8 après analyse procustéenne, que la différence des Modèles à Distribution de Points entre les personnes est minime et que le *Shape Context* paramétré de façon grossière ne peut pas parvenir à capturer les différences de formes. Cependant, un fait intéressant est que les personnes ont adopté une pose très proche entre les deux prises d'acquisitions et que les PDM semblent retranscrire ce fait (notamment pour le cas des formes de couleurs bleus au niveau des bras et des pieds).

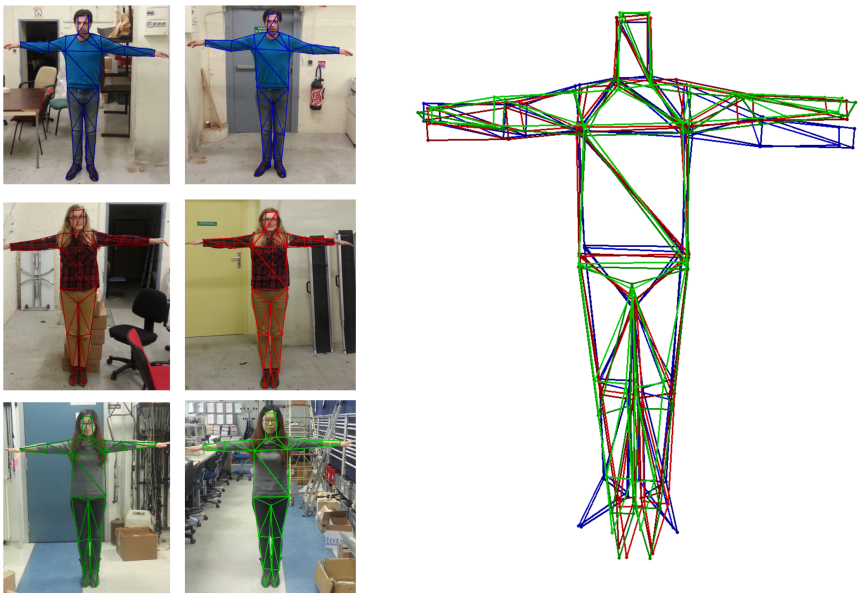


FIGURE 4.8 – Retranscription des proportions humaines par les Modèles à Distribution de Points sur 3 sujets. Les formes de droite correspondent à celles de gauche avec les mêmes couleurs, obtenues après une analyse procustéenne.

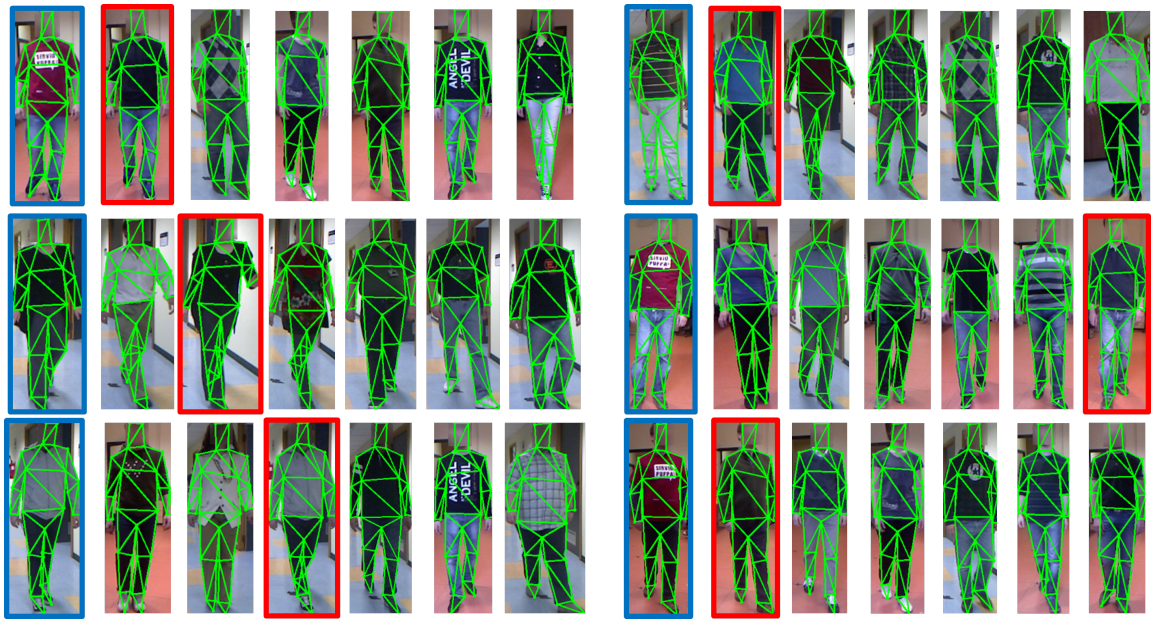


FIGURE 4.9 – Exemples de ré-identification sur RGBD-ID avec le descripteur *Shape Context*.

Nous nous référons à la Figure 4.9 pour comprendre l'information utilisée pour ré-identifier les personnes. Cette figure montre des exemples d'images sondées et les images en réponse classées en fonction de leur distance. Nous pouvons constater que les formes répondant à un rang proche de 1 présentent des similitudes vis-à-vis de la pose adoptée lors de la marche.

Ainsi, malgré une ré-identification en *single-shot*, ce modèle semble capable d'identifier, dans une certaine mesure, la pose de la personne en train de marcher. Cette expérimentation ouvre des perspectives sur une analyse comportementale telle que la reconnaissance de démarche en construisant une signature temporelle basée sur l'évolution spatiale des amers dans la séquence.

Chapitre 5

Conclusion et Perspectives

Sommaire

5.1 Résumé 89
5.2 Futurs travaux 90

5.1 Résumé

L'objectif de cette thèse est de concevoir un système complet de ré-identification basé sur la monovision et adapté à des applications embarquées. Nous proposons de délaisser les approches classiques du domaine qui s'appuient sur la soustraction de fond pour expérimenter l'alignement du Modèle à Distribution de Points (PDM) (ensemble d'amers placés à des positions clés) sur le corps humain.

Avant de procéder à cet alignement, la première étape consiste à localiser la région de l'image dans laquelle se trouve la personne. Il s'agit d'une tâche de détection de piétons. Nous présentons un état de l'art, d'une part des techniques existantes sur une approche de fenêtre glissante, d'autre part des diverses caractéristiques visuelles communément utilisées pour les piétons. Nous retenons une méthode performante et rapide basée sur les *Channel Features*, qui sont des caractéristiques locales simples, calculées sur des transformations de l'image et utilisées avec l'algorithme *AdaBoost*. Nous proposons d'améliorer certains points de cette approche tels que l'approximation des caractéristiques multi-échelles et procédons à son évaluation sur la base piétonne de l'INRIA. Le but est également de décrire les caractéristiques et méthodes d'apprentissage utilisées dans les modules suivants.

La seconde partie constitue la majeure contribution de cette thèse et concerne l'alignement du PDM sur la personne. Dans un premier temps, nous décrivons les différents modèles de représentation du corps humain utilisés en vision par ordinateur. Puis, nous faisons un état de l'art de l'alignement des PDM, jusqu'alors principalement appliqués aux visages. Par la suite, nous proposons une définition du PDM appliqué au corps humain.

Deux approches d'alignement sont exposées. La première s'appuie sur une représentation de la forme par un modèle paramétrique. Pour pouvoir retrouver les paramètres de forme de ce modèle, nous proposons d'utiliser une machine *GentleBoost*, entraînée sur une base structurée de façon à retranscrire un classement de l'alignement. Nous apprenons un score d'alignement à des régresseurs faibles de type perceptron multicouche. L'alignement est obtenu par maximisation du score de ces régresseurs faibles. La seconde approche d'alignement proposée emploie les cascades de régressions de forme. Elle tire parti du principe d'un alignement itératif pour simplifier la tâche des régresseurs faibles. À chaque itération, nous proposons de former des groupes homogènes de déformations (ou erreurs résiduelles d'alignement) grâce à une méthode de *clustering*. Puis, nous entraînons des classifieurs sur ces groupes. L'incrément de déformations gagne en cohérence grâce à l'homogénéité des déformations au sein de chaque *cluster*. Pour évaluer et valider ces deux approches, nous introduisons une nouvelle base d'apprentissage constituée d'annotations de formes sur des images de piétons. Nous mettons cette base à disponibilité de la communauté scientifique.

La dernière partie de cette thèse est consacrée à l'usage et l'apport des PDM pour la réidentification. Nous évaluons deux types de signature. Le premier s'appuie sur l'apparence de la personne, renforcée par la connaissance de l'emplacement des parties du corps grâce au PDM aligné. Nous montrons, grâce à nos évaluations, une amélioration des résultats par rapport aux méthodes de l'état de l'art. Le second type de signature s'appuie directement sur la forme décrite par les amers et démontre le potentiel de ce modèle pour de la reconnaissance de poses ou de démarches.

5.2 Futurs travaux

Cette thèse ouvre des perspectives de travaux sur les aspects de l'alignement et les usages que l'on peut faire d'un Modèle à Distribution de Points.

Une première tâche à réaliser serait d'annoter une nouvelle base d'entraînement avec des vues de profil. Un classifieur binaire déterminant si le piéton apparaît de profil ou de face/dos pourrait être appliqué entre la phase de détection et celle de l'alignement. Cet alignement se ferait ensuite avec la vue correspondante.

Par ailleurs, des améliorations pourraient être apportées à la phase de *clustering*. En effet, l'inconvénient du *K-Means* est qu'il opère dans l'espace euclidien. Or, considérer les déformations dans un espace topologique comme une variété pourrait permettre d'accentuer la séparabilité sous-jacente entre les *clusters*. Nous proposons notamment de considérer le *clustering* par un algorithme NMF (*Non-negative Matrix Factorization*) [Ding et al., 2005] et plus spécialement l'extension semi-NMF [Ding et al., 2010], les déformations pouvant être négatives. En définissant la matrice formée par l'ensemble des déformations de N échantillons : M de taille $2K \times N$, alors l'algorithme semi-NMF permet de décomposer M de façon à ce que : $M \approx WH$ avec W une matrice $2K \times R$ et H une matrice $R \times N$, R étant inférieur à $2K$ et à N. H est une matrice constituée d'éléments seulement positifs sur laquelle sont déterminés les *clusters* d'échantillons. Par ailleurs, ajouter une contrainte afin d'obtenir W clairsemée (*sparse*) reviendrait à automatiser la considération par parties spatiales à partir des données (définie actuellement arbitrairement avec la tête, le torse, les bras et les jambes). Les parties considérées seraient les R colonnes composant W.

Concernant la tâche de classification, il serait intéressant d'expérimenter l'extension d'*AdaBoost* à un problème multiclasse, *AdaBoost SAMME* [Zhu et al., 2009]. L'avantage de cet algorithme est son système de pondération à chaque échantillon lors de l'apprentissage, permettant de répondre, dans une certaine mesure, au problème de déséquilibre des classes.

Notre algorithme d'alignement fonctionne correctement sur des poses issues de la marche, mais a du mal à gérer des poses plus complexes. Une première solution serait d'étoffer la base. Autrement, pour mieux gérer l'articulation du corps humain, l'alignement pourrait être couplé avec un modèle explicite comme dans le cas des *Pictorial Structures*. C'est ce que propose [Antonakos et al., 2015] en étendant le modèle AAM [Cootes et al., 2001].

Concernant les usages du PDM, nous évoquons l'analyse du déplacement temporel des amers afin de procéder à une reconnaissance de démarches. L'avantage de ce modèle par rapport à la silhouette est qu'il prend en compte le chevauchement des parties (par exemple les bras devant le corps). De plus, l'emplacement des points à des localisations clés pourrait permettre de connaître le comportement local des personnes lors d'une marche (roulement des épaules, balancement des bras, etc.). Il serait également intéressant d'examiner le potentiel du PDM dans le cadre d'une reconnaissance de poses face à un modèle squelette.

Annexe

Publications

Sommaire

.1 Conférences internationales avec comité de relecture 91

.1 Conférences internationales avec comité de relecture

Person Re-identification Using the Silhouette Shape Described by a Point Distribution Model, Olivier Huynh, Bogdan Stanciulescu, WACV 2015, *IEEE Winter Conference on Applications of Computer Vision*

Clustering and classifying deformations for Shape Regression applied to the human body, Olivier Huynh, Bogdan Stanciulescu, ISPA 2015, *IEEE 9th International Symposium on Image and Signal Processing and Analysis*

Bibliographie

Andriluka, M., Roth, S., and Schiele, B. (2009). Pictorial structures revisited : People detection and articulated pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 1014–1021. (Cité en pages 35 et 36.)

Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., and Davis, J. (2005). Scape : Shape completion and animation of people. *ACM Transactions on Graphics*, 24(3) :408–416. (Cité en pages 35 et 36.)

Antonakos, E., Alabort-i Medina, J., and Zafeiriou, S. (2015). Active pictorial structures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*, pages 5435–5444. (Cité en pages 40, 43 et 90.)

Arthur, D. and Vassilvitskii, S. (2007). K-means++ : The advantages of careful seeding. In *Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA. (Cité en page 65.)

Bak, S., Corvee, E., Bremond, F., and Thonnat, M. (2010a). Person re-identification using haar-based and dcd-based signature. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS 2010)*, pages 1–8. (Cité en page 36.)

Bak, S., Corvee, E., Bremond, F., and Thonnat, M. (2010b). Person re-identification using spatial covariance regions of human body parts. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS 2010)*, pages 435–440. (Cité en pages 79 et 80.)

Baker, S. and Matthews, I. (2004). Lucas-kanade 20 years on : A unifying framework. *Int. J. Comput. Vision*, 56(3) :221–255. (Cité en page 39.)

Barbosa, I., Cristani, M., Del Bue, A., Bazzani, L., and Murino, V. (2012). Re-identification with rgb-d sensors. In Fusiello, A., Murino, V., and Cucchiara, R., editors, *European Conference on Computer Vision - Workshops and Demonstrations (ECCV 2012)*, pages 433–442. (Cité en pages 35, 80 et 86.)

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf : Speeded up robust features. In *European Conference on Computer Vision (ECCV 2006)*, volume 3951, pages 404–417. (Cité en page 78.)

Bdiri, T., Moutarde, F., and Steux, B. (2009). Visual object categorization with new keypoint-based adaboost features. *CoRR*, abs/0910.1294. (Cité en page 78.)

Belhumeur, P. N., Jacobs, D. W., Kriegman, D. J., and Kumar, N. (2013). Localizing parts of faces using a consensus of exemplars. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 35, pages 2930–2940. (Cité en pages 40, 41 et 43.)

Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(4) :509–522. (Cité en pages 9, 80 et 85.)

Benenson, R., Mathias, M., Timofte, R., and Van Gool, L. (2012). Pedestrian detection at 100 frames per second. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, pages 2903–2910. (Cité en pages 11 et 14.)

Benenson, R., Mathias, M., Tuytelaars, T., and Van Gool, L. (2013). Seeking the strongest rigid detector. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, pages 3666–3673. (Cité en pages 11 et 13.)

Benenson, R., Omran, M., Hosang, J., and Schiele, B. (2014). Ten years of pedestrian detection, what have we learned ? In Agapito, L., Bronstein, M. M., and Rother, C., editors, *European Conference on Computer Vision - Workshops (ECCV 2014)*, pages 613–627. (Cité en page 14.)

Bourdev, L. and Brandt, J. (2005). Robust object detection via soft cascade. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pages 236–243. (Cité en page 23.)

Bourdev, L. and Malik, J. (2009). Poselets : Body part detectors trained using 3d human pose annotations. In *IEEE International Conference on Computer Vision (ICCV 2009)*. (Cité en page 36.)

Boykov, Y. and Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary amp ; region segmentation of objects in n-d images. In *IEEE International Conference on Computer Vision (ICCV 2001)*, pages 105–112. (Cité en page 34.)

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1) :5–32. (Cité en pages 42 et 68.)

Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J.-P., and Osher, S. (2007). Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and Vision*, 28(2) :151–167. (Cité en page 35.)

Burgos-Artizzu, X. P., Perona, P., and Dollár, P. (2013). Robust face landmark estimation under occlusion. In *IEEE International Conference on Computer Vision (ICCV 2013)*, pages 1513–1520, Washington, DC, USA. (Cité en pages 41 et 43.)

Cao, X., Wei, Y., Wen, F., and Sun, J. (2013). Face alignment by explicit shape regression. *International Journal of Computer Vision*, 107(2) :177–190. (Cité en pages 41, 43, 52, 53, 62, 63, 64 et 69.)

Carreira-Perpindn, M. (2007). Gaussian mean-shift is an em algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(5) :767–776. (Cité en page 41.)

Chan, T. and Vese, L. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, 10(2) :266–277. (Cité en page 35.)

Cheng, D. S. and Cristani, M. (2014). *Person Re-identification by Articulated Appearance Matching*, pages 139–160. Springer London, London. (Cité en pages 36, 79, 80 et 84.)

Cootes, T., Edwards, G., and Taylor, C. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(6) :681–685. (Cité en pages 38, 43 et 90.)

Cootes, T., Ionita, M., Lindner, C., and Sauer, P. (2012). Robust and accurate shape model fitting using random forest regression voting. In *European Conference on Computer Vision (ECCV 2012)*, pages 278–291. (Cité en pages 42 et 43.)

Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1995). Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1) :38–59. (Cité en pages 37, 38, 43 et 44.)

Cootes, T. F., Taylor, C. J., et al. (2004). Statistical models of appearance for computer vision. (Cité en pages 37, 39 et 49.)

Cristinacce, D. and Cootes, T. (2008). Automatic feature localisation with constrained local models. *Pattern Recognition*, 41(10) :3054 – 3067. (Cité en pages 41 et 43.)

Cristinacce, D. and Cootes, T. F. (2007). Boosted regression active shape models. In *British Machine Vision Conference (BMVC 2007)*, pages 79.1–79.10. doi :10.5244/C.21.79. (Cité en pages 41, 43 et 52.)

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pages 886–893. (Cité en pages 9, 10, 12, 13, 14, 16, 24, 25, 29, 47 et 52.)

Dalal, N., Triggs, B., and Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In *European Conference on Computer Vision (ECCV 2006)*, pages 428–441. (Cité en page 11.)

Ding, C., He, X., and Simon, H. D. (2005). *On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering*, chapter 70, pages 606–610. (Cité en page 90.)

Ding, C. H. Q., Li, T., and Jordan, M. I. (2010). Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(1) :45–55. (Cité en page 90.)

Dollar, P., Appel, R., Belongie, S., and Perona, P. (2014). Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 36(8) :1532–1545. (Cité en pages 7, 11, 13, 14, 18, 19, 20 et 30.)

Dollar, P., Belongie, S., and Perona, P. (2010). The fastest pedestrian detector in the west. In *Proceedings of the British Machine Vision Conference*, pages 68.1–68.11. doi :10.5244/C.24.68. (Cité en pages 11, 13, 14 et 19.)

Dollar, P., Tu, Z., Perona, P., and Belongie, S. (2009). Integral channel features. In *British Machine Vision Conference (BMVC 2009)*, pages 91.1–91.11. doi :10.5244/C.23.91. (Cité en pages 11, 13, 14, 15, 21 et 27.)

Dollár, P., Welinder, P., and Perona, P. (2010). Cascaded pose regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*. (Cité en pages 41, 43 et 63.)

Dollár, P., Wojek, C., Schiele, B., and Perona, P. (2012). Pedestrian detection : An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34. (Cité en pages 7, 11, 12, 13, 19 et 26.)

Edwards, G., Cootes, T., and Taylor, C. (1998). Face recognition using active appearance models. In Burkhardt, H. and Neumann, B., editors, *European Conference on Computer Vision (ECCV 1998)*, pages 581–595. (Cité en page 39.)

Everingham, M., Van Gool, L., Williams, C., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2) :303–338. (Cité en page 26.)

Farenzena, M., Bazzani, L., Perina, A., Murino, V., and Cristani, M. (2010). Person re-identification by symmetry-driven accumulation of local features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, pages 2360–2367. (Cité en pages 35, 79, 80 et 84.)

Felzenszwalb, P., Girshick, R., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9) :1627–1645. (Cité en pages 12, 13 et 14.)

Felzenszwalb, P. and Huttenlocher, D. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1) :55–79. (Cité en page 35.)

Felzenszwalb, P., McAllester, D., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pages 1–8. (Cité en pages 12 et 13.)

Ferreira Da Costa, M. and Dai, W. (2016). Achieving Super-Resolution in Multi-Rate sampling systems via efficient semidefinite programming. In *IEEE Information Theory Workshop (ITW 2016)*. (Cité en page 14.)

Finlayson, G., Hordley, S., Schaefer, G., and Tian, G. Y. (2005). Illuminant and device invariant colour using histogram equalisation. *Pattern Recognition*, 38(2) :179 – 190. (Cité en page 79.)

Fischler, M. A. and Elschlager, R. (1973). The representation and matching of pictorial structures. *IEEE Transactions on Computers*, C-22(1) :67–92. (Cité en page 35.)

Forssén, P.-E. (2007). Maximally stable colour regions for recognition and matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, USA. (Cité en page 79.)

Freifeld, O., Weiss, A., Zuffi, S., and Black, M. J. (2010). Contour people : A parameterized model of 2D articulated human shape. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, pages 639–646. (Cité en pages 35 et 36.)

Freund, Y. and Schapire, R. E. (1997a). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1) :119 – 139. (Cité en page 9.)

Freund, Y. and Schapire, R. E. (1997b). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1) :119 – 139. (Cité en page 20.)

Friedman, J., Hastie, T., and Tibshirani, R. (1998). Additive logistic regression : a statistical view of boosting. *Annals of Statistics*, 28 :2000. (Cité en pages 40 et 50.)

Gheissari, N., Sebastian, T., and Hartley, R. (2006). Person reidentification using spatiotemporal appearance. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, pages 1528–1535. (Cité en pages 79 et 80.)

Gower, J. C. (1975). Generalized procrustes analysis. *Psychometrika*, 40(1). (Cité en pages 48 et 49.)

Gray, D. and Tao, H. (2008). *Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features*, pages 262–275. (Cité en pages 78, 79, 82 et 84.)

Hahnel, M., Klunder, D., and Kraiss, K. F. (2004). Color and texture features for person recognition. In *IEEE International Joint Conference on Neural Networks 2004*, page 652. (Cité en page 80.)

Hamdoun, O., Moutarde, F., Stanciulescu, B., and Steux, B. (2008). Person re-identification in multi-camera system by signature based on interest point descriptors collected on short video sequences. In *ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2008)*, pages 1–6. (Cité en page 78.)

Hermosilla, G., del Solar, J. R., Verschae, R., and Correa, M. (2012). A comparative study of thermal face recognition methods in unconstrained environments. *Pattern Recognition*, 45(7) :2445 – 2459. (Cité en page 77.)

Horbert, E., Rematas, K., and Leibe, B. (2011). Level-set person segmentation and tracking with multi-region appearance models and top-down shape information. In *IEEE International Conference on Computer Vision (ICCV 2011)*, pages 1871–1878. (Cité en pages 35 et 36.)

Hou, X., Li, S., Zhang, H., and Cheng, Q. (2001). Direct appearance models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, pages I–828–I–833. (Cité en page 39.)

Javed, O., Shafique, K., Rasheed, Z., and Shah, M. (2008). Modeling inter-camera space–time and appearance relationships for tracking across non-overlapping views. *Computer Vision and Image Understanding*, 109(2) :146 – 162. (Cité en page 79.)

Javed, O., Shafique, K., and Shah, M. (2005). Appearance modeling for tracking in multiple non-overlapping cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pages 26–33. (Cité en pages 80 et 81.)

Joachims, T., Finley, T., and Yu, C.-N. J. (2009). Cutting-plane training of structural svms. *Machine Learning*, 77(1) :27–59. (Cité en page 82.)

Jojic, N., Perina, A., Cristani, M., Murino, V., and Frey, B. (2009). Stel component analysis : Modeling spatial correlations in image class structure. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 2044–2051. (Cité en page 34.)

Ju, S., Black, M., and Yacoob, Y. (1996). Cardboard people : a parameterized model of articulated image motion. In *International Conference on Automatic Face and Gesture Recognition 1996*, pages 38–44. (Cité en page 45.)

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes : Active contour models. *International Journal of Computer Vision*, 1(4) :321–331. (Cité en page 35.)

Kazemi, V. and Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014)*. (Cité en pages 42 et 43.)

Khedher, M. I. and Yacoubi, M. A. (2015). *Local Sparse Representation Based Interest Point Matching for Person Re-identification*, pages 241–250. Cham. (Cité en page 79.)

Koestinger, M., Hirzer, M., Wohlhart, P., Roth, P. M., and Bischof, H. (2012). Large scale metric learning from equivalence constraints. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*. (Cité en pages 80, 81 et 84.)

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. (Cité en page 11.)

Kviatkovsky, I., Adam, A., and Rivlin, E. (2012). Color Invariants for Person Re-Identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, pages 1–15. (Cité en page 79.)

Layne, R., Hospedales, T. M., and Gong, S. (2012). Towards person identification and re-identification with attributes. In *European Conference on Computer Vision (ECCV 2012)*, pages 402–412, Berlin, Heidelberg. (Cité en page 80.)

Layne, R., Hospedales, T. M., and Gong, S. (2014). *Investigating Open-World Person Re-identification Using a Drone*, pages 225–240. (Cité en page 2.)

Liu, X. (2007). Generic face alignment using boosted appearance model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, pages 1–8. (Cité en pages 40 et 43.)

Liu, X., Wang, H., Wu, Y., Yang, J., and Yang, M. H. (2015). An ensemble color model for human re-identification. In *IEEE Winter Conference on Applications of Computer Vision (WACV 2015)*, pages 868–875. (Cité en pages 80, 81 et 84.)

Liu, X., Yu, T., Sebastian, T., and Tu, P. (2008). Boosted deformable model for human body alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pages 1–8. (Cité en pages 36, 37, 41, 42, 43, 44, 47, 50, 51, 52, 54, 57, 58, 61 et 62.)

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110. (Cité en pages 9 et 78.)

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679. (Cité en page 11.)

Matthews, I. and Baker, S. (2004). Active appearance models revisited. *International Journal of Computer Vision*, 60(2) :135–164. (Cité en pages 39, 43, 47, 49, 50 et 56.)

Moutarde, F., Stanciulescu, B., and Breheret, A. (2008). Real-time visual detection of vehicles and pedestrians with new efficient adaBoost features. In *IEEE International Conference on Intelligent Robots Systems (IROS 2008) - Workshop on Planning, Perception and Navigation for Intelligent Vehicles (PPNIV)*, Nice, France. (Cité en page 41.)

Mu, Y., Yan, S., Liu, Y., Huang, T., and Zhou, B. (2008). Discriminative local binary patterns for human detection in personal album. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pages 1–8. (Cité en page 10.)

Munaro, M., Basso, A., Fossati, A., Van Gool, L., and Menegatti, E. (2014). 3d reconstruction of freely moving persons for re-identification with a depth sensor. In *IEEE International Conference on Robotics and Automation (ICRA 2014)*, pages 4512–4519, Piscataway, N.J. (Cité en pages 35 et 36.)

Ojala, T., Pietikäinen, M., and Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1) :51 – 59. (Cité en page 10.)

Ott, P. and Everingham, M. (2009). Implicit color segmentation features for pedestrian and object detection. In *IEEE International Conference on Computer Vision (ICCV 2009)*, pages 723–730. (Cité en page 11.)

Ouyang, W. and Wang, X. (2013a). Joint deep learning for pedestrian detection. In *IEEE International Conference on Computer Vision (ICCV 2013)*, pages 2056–2063. (Cité en pages 11 et 13.)

Ouyang, W. and Wang, X. (2013b). Single-pedestrian detection aided by multi-pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, pages 3198–3205. (Cité en pages 12, 13 et 14.)

Ozuysal, M., Calonder, M., Lepetit, V., and Fua, P. (2010). Fast keypoint recognition using random ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(3) :448–461. (Cité en page 64.)

Papageorgiou, C. and Poggio, T. (2000). A trainable system for object detection. *International Journal of Computer Vision*, 38(1) :15–33. (Cité en page 9.)

Papandreou, G. and Maragos, P. (2008). Adaptive and constrained algorithms for inverse compositional active appearance model fitting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pages 1–8. (Cité en pages 40 et 43.)

Park, D., Ramanan, D., and Fowlkes, C. (2010). Multiresolution models for object detection. In *European Conference on Computer Vision (ECCV 2010)*, pages 241–254. (Cité en pages 12 et 13.)

Park, D., Zitnick, C. L., Ramanan, D., and Dollar, P. (2013). Exploring weak stabilization for motion feature extraction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, CVPR '13, pages 2882–2889, Washington, DC, USA. (Cité en pages 11 et 13.)

Park, U., Jain, A. K., Kitahara, I., Kogure, K., and Hagita, N. (2006). Vise : Visual search engine using multiple networked cameras. In *International Conference on Pattern Recognition (ICPR 2006)*, volume 3, pages 1204–1207. (Cité en pages 79 et 83.)

Pishchulin, L., Andriluka, M., Gehler, P., and Schiele, B. (2013). Poselet conditioned pictorial structures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013)*. (Cité en pages 36 et 45.)

Porikli, F. (2003). Inter-camera color calibration by correlation model function. In *International Conference on Image Processing (ICIP 2003)*, volume 2, pages II–133–6 vol.3. (Cité en page 79.)

Qu, C., Gao, H., Monari, E., Beyerer, J., and Thiran, J. P. (2015). Towards robust cascaded regression for face alignment in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition - Workshops (CVPRW 2015)*, pages 1–9. (Cité en pages 42 et 43.)

Ren, S., Cao, X., Wei, Y., and Sun, J. (2014). Face alignment at 3000 fps via regressing local binary features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014)*, pages 1685–1692. (Cité en pages 42, 43, 67 et 74.)

Riedmiller, M. and Braun, H. (1993). A direct adaptive method for faster backpropagation learning : the rprop algorithm. In *IEEE International Conference on Neural Networks 1993*, pages 586–591 vol.1. (Cité en page 58.)

Rother, C., Kolmogorov, V., and Blake, A. (2004). "grabcut" : Interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH 2004 Papers*, pages 309–314. (Cité en pages 34 et 36.)

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Parallel distributed processing : Explorations in the microstructure of cognition, vol. 1. chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press. (Cité en page 54.)

Sabzmeydani, P. and Mori, G. (2007). Detecting pedestrians by learning shapelet features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, pages 1–8. (Cité en pages 9, 10 et 13.)

Saragih, J. M., Lucey, S., and Cohn, J. F. (2010). Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91(2) :200–215. (Cité en pages 41 et 43.)

Sermanet, P., Kavukcuoglu, K., Chintala, S., and LeCun, Y. (2013). Pedestrian detection with unsupervised multi-stage feature learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, pages 3626–3633. (Cité en pages 11 et 13.)

Sève, R. (2009). *Science de la couleur : aspects physiques et perceptifs*. Chalagam éd. (Cité en page 17.)

Sheikh, Y., Javed, O., and Kanade, T. (2009). Background subtraction for freely moving cameras. In *IEEE International Conference on Computer Vision (ICCV 2009)*, pages 1219–1225. (Cité en page 6.)

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from a single depth image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*. (Cité en page 35.)

Smedt, F. D. and Goedemé, T. (2015). Fast rotation invariant object detection with gradient based detection models. In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications*, pages 400–407. (Cité en page 31.)

Sobral, A. and Vacavant, A. (2014). A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding*, 122 :4 – 21. (Cité en page 7.)

Strobl, C., Malley, J., and Tutz, G. (2009). An introduction to recursive partitioning : Rationale, application and characteristics of classification and regression trees, bagging and random forests. *Psychological methods*. (Cité en page 68.)

Sun, Y., Wang, X., and Tang, X. (2013). Deep convolutional network cascade for facial point detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, CVPR '13, pages 3476–3483, Washington, DC, USA. (Cité en pages 42 et 43.)

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface : Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014)*. (Cité en pages 2 et 37.)

Toshev, A. and Szegedy, C. (2013). Deeppose : Human pose estimation via deep neural networks. *CoRR*, abs/1312.4659. (Cité en pages 36 et 45.)

Tzimiropoulos, G., Alabort-i Medina, J., Zafeiriou, S., and Pantic, M. (2013). Generic active appearance models revisited. In Lee, K., Matsushita, Y., Rehg, J., and Hu, Z., editors, *Computer Vision – ACCV 2012*, pages 650–663. (Cité en pages 40 et 43.)

Valstar, M., Martinez, B., Binefa, X., and Pantic, M. (2010). Facial point detection using boosted regression and graph models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, pages 2729–2736. (Cité en pages 41, 43 et 52.)

van de Sande, K., Gevers, T., and Snoek, C. (2010). Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9) :1582–1596. (Cité en page 79.)

van de Weijer, J., Schmid, C., and Verbeek, J. (2007). Learning color names from real-world images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, pages 1–8. (Cité en page 81.)

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, pages I–511–I–518 vol.1. (Cité en pages 9, 17, 23 et 30.)

Viola, P. and Jones, M. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2) :137–154. (Cité en page 21.)

Viola, P., Jones, M., and Snow, D. (2003). Detecting pedestrians using patterns of motion and appearance. In *IEEE International Conference on Computer Vision (ICCV 2003)*, pages 734–741 vol.2. (Cité en pages 9, 11 et 13.)

Walk, S., Majer, N., Schindler, K., and Schiele, B. (2010). New features and insights for pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, pages 1030–1037. (Cité en pages 11 et 13.)

Wang, L., Tan, T., Ning, H., and Hu, W. (2003). Silhouette analysis-based gait recognition for human identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(12) :1505–1518. (Cité en page 35.)

Wang, X., Doretto, G., Sebastian, T., Rittscher, J., and Tu, P. (2007). Shape and Appearance Context Modeling. *IEEE International Conference on Computer Vision (ICCV 2007)*, pages 1–8. (Cité en page 80.)

Wang, X., Han, T., and Yan, S. (2009). An hog-lbp human detector with partial occlusion handling. In *IEEE International Conference on Computer Vision (ICCV 2009)*, pages 32–39. (Cité en pages 10, 12 et 13.)

Weinberger, K. Q., Blitzer, J., and Saul, L. K. (2006). Distance metric learning for large margin nearest neighbor classification. In Weiss, Y., Schölkopf, B., and Platt, J. C., editors, *Advances in Neural Information Processing Systems 18*, pages 1473–1480. (Cité en page 80.)

Wojek, C. and Schiele, B. (2008). A performance evaluation of single and multi-feature people detection. In *Pattern Recognition*, volume 5096, pages 82–91. (Cité en pages 9, 13 et 14.)

Wu, B. and Nevatia, R. (2005). Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*. (Cité en pages 10 et 12.)

Wu, B. and Nevatia, R. (2008). Optimizing Discrimination-Efficiency Tradeoff in Integrating Heterogeneous Local Features for Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*. (Cité en pages 10 et 51.)

Xiong, X. and De la Torre, F. (2013). Supervised descent method and its applications to face alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, pages 532–539. (Cité en pages 42, 43, 62 et 63.)

Yan, J., Zhang, X., Lei, Z., Liao, S., and Li, S. (2013). Robust multi-resolution pedestrian detection in traffic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, pages 3033–3040. (Cité en pages 12, 13 et 14.)

Yang, Y. and Ramanan, D. (2011). Articulated pose estimation with flexible mixtures-of-parts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, pages 1385–1392. (Cité en page 36.)

Yu, S., Tan, D., and Tan, T. (2006). A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. In *International Conference on Pattern Recognition (ICPR 2006)*, volume 4, pages 441–444. (Cité en page 77.)

Zhang, S., Bauckhage, C., and Cremers, A. (2014). Informed haar-like features improve pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014)*, pages 947–954. (Cité en pages 9 et 13.)

Zhao, R., Ouyang, W., and Wang, X. (2013). Unsupervised salience learning for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013)*. (Cité en page 78.)

Zheng, W. S., Gong, S., and Xiang, T. (2013). Reidentification by relative distance comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(3) :653–668. (Cité en page 80.)

Zhu, J., Zou, H., Rosset, S., and Hastie, T. (2009). Multi-class adaboost. (Cité en page 90.)

Zhu, Q., Yeh, M.-C., Cheng, K.-T., and Avidan, S. (2006). Fast human detection using a cascade of histograms of oriented gradients. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, pages 1491–1498. (Cité en page 9.)

Zhu, X. and Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, pages 2879–2886. (Cité en pages 40, 41 et 43.)

Zhuang, Y. and Chen, C. (2007). Efficient silhouette extraction with dynamic viewpoint. In *IEEE International Conference on Computer Vision (ICCV 2007)*, pages 1–8. (Cité en page 35.)

Zuffi, S., Freifeld, O., and Black, M. J. (2012). From pictorial structures to deformable structures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, pages 3546–3553. (Cité en page 36.)

Résumé

L'essor des systèmes mobiles pose de nouvelles problématiques dans le domaine de vision par ordinateur. Les techniques de ré-identification s'appuyant sur un réseau de caméras fixes doivent être repensées afin de s'adapter à un décor changeant. Pour répondre à ces besoins, cette thèse explore, dans le cadre du corps humain, l'utilisation d'un modèle structurel habituellement employé pour de la reconnaissance faciale. Il s'agit de l'alignement d'un modèle à distribution de points (*Point Distribution Model* ou PDM). L'objectif de ce pré-traitement avant la ré-identification est triple, segmenter la personne du décor, améliorer la robustesse vis-à-vis de sa pose et extraire des points clés spatiaux pour construire une signature basée sur son comportement.

Nous concevons et évaluons un système complet de ré-identification, découpé en trois modules mis en séquence. Le premier de ces modules correspond à la détection de personnes. Nous proposons de nous baser sur une méthode de l'état de l'art utilisant les *Channel Features* avec l'algorithme *AdaBoost*.

Le second module est l'alignement du PDM au sein de la boîte englobante fournie par la détection. Deux approches sont présentées dans cette thèse. La première s'appuie sur une formulation paramétrique du modèle de forme. L'alignement de ce modèle est guidé par la maximisation d'un score d'un modèle d'apparence *GentleBoost* utilisant des caractéristiques locales de type histogrammes de gradients orientés. La seconde approche exploite une technique de cascade de régressions de forme. L'idée principale est le regroupement de déformations homogènes en clusters et la classification de ces derniers dans le but d'aligner le PDM itérativement.

Enfin, le troisième module est celui de la ré-identification. Nous montrons que l'utilisation d'un PDM en support permet d'améliorer les résultats de ré-identification. Nos expérimentations portent sur des signatures d'apparence classique, les histogrammes de couleurs, et sur un descripteur de forme, le *Shape Context*. L'évaluation de ce dernier fournit des résultats encourageants pour une perspective d'utilisation des PDM au sein d'une reconnaissance de démarches.

Mots Clés

Modèle à Distribution de Points, Alignement, Régression de Forme, Corps humain, Ré-identification, Applications embarquées

Abstract

The emergence of mobile systems brings new problematics in computer vision. Static camera-based methods for re-identification need to be adapted in this new context. To deal with dynamical background, this thesis proposes to employ the well known Point Distribution Model (PDM), usually applied for face alignment, on the human body. Three advantages come from this pre-processing before re-identification, segment the person from background, enhance robustness to the person pose and extract spatial key points to build a behavioural-based signature.

We implement and evaluate a complete framework for re-identification, divided in three sequential modules. The first one corresponds to the pedestrian detection. We use an efficient method of the state of the art employing the Channel Features with the algorithm AdaBoost.

The second one is the PDM alignment within the bounding box provided by the detection step. Two distinct approaches are presented in this thesis. The first method relies on a parametric formulation to describe the shape, similar to the ASM or AAM. To fit this shape model, we maximize the score of an appearance model defined by GentleBoost, which employs local histograms of oriented gradients. The second approach is based on the cascade regression shape scheme. The main idea is the approximation for each step into a classification of homogeneous deformations, grouped by unsupervised clustering.

The third module is the re-identfication one. We show that employing a PDM as a structural support improves re-identification results. We experiment classic appearance-based signatures, color histograms and the shape descriptor Shape Context. The results are encouraging for application perspective of PDM for the gait recognition.

Keywords

Point Distribution Model, Alignment, Shape Regression, Human Body, Re-identification, Embedded applications