Machine Learning for Image Based Motion Capture

Ankur Agarwal

▶ To cite this version:

HAL Id: tel-00390301

https://theses.hal.science/tel-00390301

Submitted on 1 Jun 2009

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Numéro attribué par la bibliothèque

THÈSE

pour obtenir le grade de

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Spécialité : Imagerie, Vision et Robotique

Ecole Doctorale : Mathématiques, Sciences et Technologie de l'Information

présentée et soutenue publiquement

par

Ankur Agarwal

le 26 Avril 2006

Machine Learning for Image Based Motion Capture

Directeur de thèse : M. William Triggs

JURY

M. Roger Mohr	Président
M. Andrew Zisserman	Rapporteur
M. Pascal Fua	Rapporteur
M. William Triggs	Directeur de thèse
M. Philip Torr	Examinateur

Abstract

Image based motion capture is a problem that has recently gained a lot of attention in the domain of understanding human motion in computer vision. The problem involves estimating the 3D configurations of a human body from a set of images and has applications that include human computer interaction, smart surveillance, video analysis and animation. This thesis takes a machine learning based approach to reconstructing 3D pose and motion from monocular images or video. It makes use of a collection of images and motion capture data to derive mathematical models that allow the recovery of full body configurations directly from image features. The approach is completely data-driven and avoids the use of a human body model. This makes the inference extremely fast.

We formulate a class of regression based methods to distill a large training database of motion capture and image data into a compact model that generalizes to predicting pose from new images. The methods rely on using appropriately developed robust image descriptors, learning dynamical models of human motion, and kernelizing the input within a sparse regression framework. Firstly, it is shown how pose can effectively and efficiently be recovered from image silhouettes that are extracted using background subtraction. We exploit sparseness properties of the relevance vector machine for improved generalization and efficiency, and make use of a mixture of regressors for probabilistically handling ambiguities that are present in monocular silhouette based 3D reconstruction. The methods developed enable pose reconstruction from single images as well as tracking motion in video sequences. Secondly, the framework is extended to recover 3D pose from cluttered images by introducing a suitable image encoding that is resistant to changes in background. We show that non-negative matrix factorization can be used to suppress background features and allow the regression to selectively cue on features from the foreground human body. Finally, we study image encoding methods in a broader context and present a novel multi-level image encoding framework called 'hyperfeatures' that proves to be effective for object recognition and image classification tasks.

Acknowledgements

This thesis would never have existed without the guidance of my advisor Bill Triggs. I am grateful to him for the invaluable supervision I have received during the course of this work. From his knowledge of the most minute details in a variety of technical areas to his broad understanding and global viewpoint of challenging problems, Bill's thinking has played a major role in the development of the research I have described in this thesis. Interacting with Bill over a period of time has been a very wholesome learning experience for me and I am happy to say that a lot of what I have learned from him applies far beyond technical research.

I would like to express my gratitude to my external thesis committee members Andrew Zisserman, Pascal Fua and Phil Torr for the time they spent on reading and reviewing this work; and to Andrew Zisserman in particular for the several interesting discussions I have had with him. I am also thankful to Roger Mohr, Cordelia Schmid and all other members of the LEAR team for their support: Gyuri, Navneet and Guillaume for helping with pieces of code/software, Diane and Matthijs for helping with my French, and all my other friends at INRIA for making my stay in Grenoble an unforgettable experience.

Finally, though I cannot describe in a finite number of words the contribution of my family, I must mention that it was my parents who first encouraged me to take up the initiave of spending these few years in a place like France; and it is their love that, from a distance, had been the major driving force for me throughout the course of this work.

Contents

List of Figures

Apprentissage automatique pour l'estimation du mouvement humain

Résumé

L'estimation du mouvement 3D humain à partir d'images est un problème phare de la vision par ordinateur. Il y a des applications dans l'interaction homme-machine, la surveillance, l'animation et l'analyse des vidéos. Cette thèse propose une approche basée sur l'apprentissage automatique pour estimer la pose et le mouvement articulaire du corps humain à partir d'images et de séquences monoculaires. Partant d'une base d'images des personnes en mouvement annotée avec les configurations articulaire correspondantes issues de la capture de mouvement, nous déduisons des modèles qui permettent d'estimer directement la pose 3D du corps en fonction d'un vecteur de descripteurs de forme visuelle extrait de l'image. Entièrement basée sur les données observées, l'approche évite l'introduction d'un modèle explicit du corps humain et fournit ainsi une inférence directe et rapide.

Nous proposons notamment une classe de méthodes basée sur la régression qui sont en mesure de résumer une grande base de données d'apprentissage dans un modèle d'estimation de mouvement compact et performant. L'approche se base sur des descripteurs robustes d'image, sur la régression éparse basée sur une représentation noyau, et sur l'apprentissage d'un modèle dynamique du mouvement humain. Nous montrons d'abord comment encoder dans un vecteur de descripteurs la forme des silhouettes extraites de l'image par soustraction de fond. Afin d'estimer la pose 3D du corps à partir de ces descripteurs, nous introduisons une approche régressive, qui exploite la caractère éparse de la machine à vecteur de pertinence (*relevance vector machine*) pour améliorer la généralisation et réduire le coût du calcul. Ensuite nous généralisons l'approche à un mélange probabiliste de régresseurs afin de mieux caractériser les ambiguïtés du problème. Nos méthodes permettent l'estimation de la pose à partir d'images statiques et en plus le suivi du mouvement dans les séquences vidéo. Ensuite nous démontrons comment un codage d'image affiné permet de récupérer la pose même à partir d'images dont le fond est complexe et encombré. Cette méthode exploite la factorisation non-négative de matrice afin de supprimer la plupart des perturbations liées au fond.

Finalement nous changeons de contexte afin de présenter une méthode d'extraction d'indices d'image génériques et performantes pour la reconnaissance de classes visuelles – les « hyperfeatures », qui codent le contenu image à plusieurs niveaux d'abstraction par biais d'un processus récursif multi-échelle de caractérisation de co-occurrence d'indices.

Introduction

1. Contexte de la thèse

Un thème majeur de la vision par ordinateur est l'identification automatique du contenu des images et des vidéos. Parmi les verrous du domaine on peut citer la compréhension de scène, la

reconnaissance d'objets, la détection de personnes et l'interprétation de leurs activités. L'analyse d'images de personnes est un sous-domaine privilégié en raison de ses nombreuses applications potentielles. Par exemple des algorithmes efficaces pour le suivi et l'interprétation du mouvement humain permettraient l'interaction homme-machine plus naturelle, la surveillance domestique et de sécurité plus fiable, l'analyse améliorée du mouvement pour la diagnostique médicale et pour la formation sportive, ainsi que beaucoup d'autres applications.

On appelle souvent la technologie d'enregistrement du mouvement humain la « capture du mouvement » (*motion capture*). À l'origine elle était conçue pour l'analyse biomécanique et médicale, mais elle s'est vite imposée comme la source de données d'animation préférée pour la production de films et de jeux vidéo. Les systèmes les plus performants sont basés sur la photogrammétrie, mais ceci exige plusieurs appareils-vidéo spécialisés et soigneusement calibrés, l'illumination spéciale, et des costumes spéciaux munies de cibles réfléchissantes ou actives attachées aux l'articulations du corps. Il existe également des systèmes mécaniques qui s'attachent au corps, et des sondes magnétiques.

2. Présentation du problème

Cette thèse s'adresse au problème d'estimation de la *pose* humaine — la configuration des différents membres du corps — à partir d'images et de séquences vidéo monoculaires, sans l'utilisation de cibles marqueur (*markerless monocular motion capture*). Le but est d'estimer la configuration 3D du corps à partir des images seules, sans intervention manuelle. La configuration 3D est représenté par un vecteur numérique qui encode les positions et les orientations relatives des différents membres du corps — par exemple du même type que les vecteurs d'angle d'articulation qui sont issues des systèmes de capture de mouvement conventionnels. Le problème devient ainsi celui de l'estimation d'un état paramétrique de haute dimension (la configuration 3D du corps) à partir d'un signal complexe et parfois ambiguë (l'image).

Nous nous limitons au cas monoculaire, c-à-d une seule caméra est utilisée en entrée. Le problème multi-caméra est déjà difficile en raison de la grande variabilité de l'aspect humain et du nombreux paramètres à estimer (typiquement entre 30 et 60), et il devient plus difficile dans le cas monoculaire parce qu'une partie de l'information 3D est perdue — il n'y a plus de signal stéréo et la profondeur (la distance entre le membre du corps en question et la caméra) n'est pas directement observable.

3. Les approches générative et diagnostique

Le traitement visuel du mouvement humain est un secteur de recherche actif. Il existe de nombreux travaux sur la détection de personnes, le suivi de leurs mouvements dans les vidéos, l'estimation de la pose de leurs corps, et la modélisation de la dynamique de leurs mouvements. Ici nous parlerons uniquement des méthodes d'estimation de pose. Grossièrement, les approches peuvent être affectées en deux classes : génératives et diagnostiques. Une approche générative — on dit aussi descendante (*top-down*) ou *model based* — dispose d'un modèle plus ou moins explicite de la situation 3D qui devrait être ajusté pour coller au mieux aux observations. Ainsi, le problème devient celui de l'ajustement d'un modèle paramétrique complexe qui a vocation à « expliquer » (les éléments pertinents de) l'image. Inversement, une approche diagnostique — on dit aussi ascendante (*bottom-up*) ou régressive — ne dispose pas d'un modèle explicite et cherche uniquement à prédire la sortie voulue (ici la pose 3D) directement à partir des observations. Ne faisant ni explication ni ajustement, les approches diagnostiques sont typiquement plus légères et plus réactives mais en principe moins sûres et moins informatives que les approches génératives. Cependant, la phase d'ajustement générative se révèle être délicate et en pratique l'approche diagnostique est souvent plus sûre, quoique moins précise, que l'approche générative. N'ayant plus besoin d'un modèle explicite du corps, les méthodes diagnostiques se donnent naturellement à l'apprentissage

— l'exploitation d'une base d'exemples représentatifs et d'une méthode d'apprentissage statistique pour estimer la fonction qui prédit la pose à partir des observations.

L'approche d'apprentissage automatique

Nous avons déjà observé que l'exploitation d'un modèle géométrique explicite du corps humain peut être remplacée par l'apprentissage automatique d'un modèle effectif qui prédit la configuration 3D du corps directement à partir de l'image observée. La phase de l'apprentissage exige un ensemble d'images représentatives annotées avec leurs configurations 3D associées. Typiquement les images sont représentées par des indices pertinentes extraites de l'image et recueillées dans un vecteur de descripteurs de haute dimension. Cependant, la complexité du problème rend difficile l'identification manuelle d'un ensemble satisfaisant de descripteurs et il est utile d'exploiter l'apprentissage à cette étape aussi.

Cette thèse adopte systématiquement l'approche apprentissage. Elle ne demande pas la modélisation détaillée de la forme et de l'apparence du corps humain, ce qui (en principe et moyennant des descripteurs d'image et une base d'exemples d'apprentissage adéquate) lui permet d'être plus résistante à ces informations inessentiels. Plus signicativement, constituer la représentation sur la base de mouvements humains réels a pour effet de lui focaliser sur les poses humaines « typiques » — un ensemble de poses bien plus petit que celui de tout ce qui est en principe possible au plan cinématique. Quoique limitante, cette focalisation stabilise la solution et réduite significativement les ambiguïtés intrinsèques du problème en éliminant les configurations invraisemblables. La modélisation statistique de l'aspect et du mouvement humain permet également d'incorporer l'incertitude et ainsi de faire l'inférence probabiliste.

Nous introduisons plusieurs classes de méthodes basées sur la régression afin de résumer la base d'apprentissage dans un modèle compact qui passe directement des descripteurs image à la pose 3D. L'approche est entièrement diagnostique, sans modèle explicite du corps.

Contributions de la thèse

Le thème principal de la thèse est l'estimation de la pose et des mouvements 3D humains à partir d'images monoculaires avec des approches régressives basées sur l'apprentissage, qui se fait à partir d'une base d'exemples issus d'un système de capture de mouvement conventionnel. Il y a deux classes de contribution : les approches régressives pour la pose et le mouvement et la régression multi-valeurs ; et les représentations d'image, notamment la méthode factorisation matricielle non-négative pour réduire l'influence du fond et les indices « hyperfeatures » pour la reconnaissance.

En détail on peut citer les contributions suivantes :

Estimation de pose humain à partir d'une seule image. Le chapitre 3 démontre comment combiner une représentation robuste de la silhouette du sujet avec la régression à noyau afin d'estimer sa pose 3D à partir d'une seule image. Une machine à vecteur de pertinence (*relevance vector machine*) apprend une représentation creuse qui améliore la généralisation et l'économie de la régression. L'approche n'exige ni modèle explicite du corps ni identification antérieure des membres du corps dans l'image.

Suivi discriminatif mouvement humain. Le chapitre 4 présente un nouvel algorithme de suivi du mouvement humain dans les images. La méthode combine l'estimation de pose par régression avec un modèle dynamique des mouvements 3D. Elle est entièrement discriminative : elle va
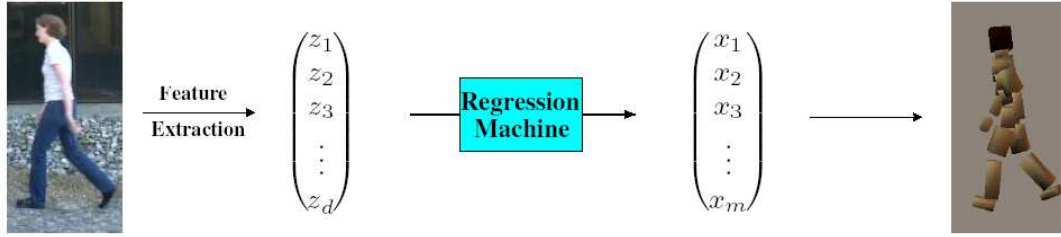
Figure 1: *La méthode basée sur la régression pour l'estimation de la pose humain: l'image d'entrée est résumée dans un vecteur de descripteurs* $\mathbf{z}$, *qui est envoyé à la machine de régression afin d'estimer le vecteur de pose* $\mathbf{x}$. *Ce vecteur encode la configuration du corps et permet la synthèse des images d'un animation du sujet.*

directement des observations au mouvement 3D sans passer par un calcul intermédiaire de la probabilité de l'image connaissant la pose.

Estimation de pose multivaleurs probabiliste. L'estimation de la pose 3D à partir d'une seule image est souvent ambiguë : plusieurs solutions sont admissibles. Le chapitre 5 développe une méthode qui reconstruit les solutions multiples aussi que leurs probabilités associées, basée sur la modélisation probabiliste conjointe de la pose et des descripteurs d'image. La méthode est exploitée pour le suivi du mouvement multi-hypothèses et pour la reconnaissance d'actions simples.

Codage sélective d'images encombrées. Le chapitre 6 développe une méthode de re-encodage des descripteurs d'image qui rehausse les contours humains utiles et réduite l'encombrement du fond. La factorisation non-négative de matrice permet d'apprendre une base locale pour l'image à cet effet, et ainsi d'étendre l'estimation régressive de la pose humaine aux images encombrées.

Un mélange de modèles dynamiques locales. Le chapitre 7 présente un modèle dynamique 2D du mouvement humain composé d'un mélange de processus auto-régressifs sur des représentations locales de dimension réduite. Il modélise de manière probabiliste les transitions entre les différents aspects visuels et classes de mouvement, et permet d'étendre le suivi aux images plus encombrées.

La représentation « hyperfeatures ». Le chapitre 8 présente une nouvelle représentation d'images pour la reconnaissance visuelle, les « hyperfeatures », qui encodent la co-occurence d'indices à plusieurs niveaux récursifs afin de représenter l'image à plusieurs niveaux d'abstraction.

L'estimation de la pose humain basée sur la régression

1. Approche

Nous décrivons la pose 3D du corps par un vecteur $\mathbf{x}$. N'importe quelle représentation peut être utilisé. Dans nos expériences les entrées sont soit les angles d'articulation — souvent au format de la sortie d'un système de capture de mouvement — soit les coordonnées 3D des centres d'articulations. L'image d'entrée est aussi représentée par un vecteur de descripteurs $\mathbf{z}$. Le choix de représentation d'image est délicat en raison de la difficulté du problème et nous adoptons une approche apprentissage afin de disposer des représentations qui codent les aspects qui sont les plus pertinents pour l'estimation de la pose 3D.

Nous avons étudié deux types de représentation. Quand la soustraction de fond d'image est disponible, nous adoptons une représentation basée sur la forme de la silhouette image du corps.
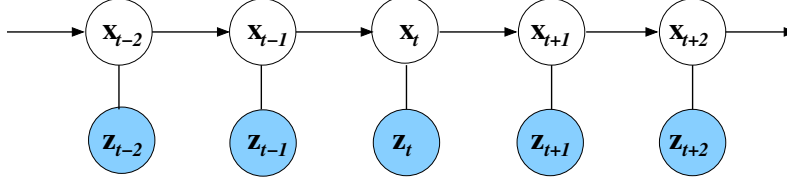
Figure 2: *L'estimation du mouvement humain dans une séquence vidéo peut être considérée comme un problème d'inférence sur une chaîne de Markov. A l'instant t, l'estimation de l'état $\mathbf{x}_t$ dépend directement de l'état précèdent $\mathbf{x}_{t-1}$ et de l'observation courante $\mathbf{z}_t$.*

Plus précisément, la représentation se base sur la quantification vectorielle de la distribution de descripteurs locaux *shape context* issu des contours de la silhouette. Elle capte la forme de la silhouette dans un vecteur 100D numérique d'une manière robuste aux occultations et aux erreurs de soustraction de fond. Quand la soustraction de fond n'est pas disponible, nous adoptons une représentation basée sur les histogrammes d'orientation de gradient similaire aux descripteurs *SIFT*, qui sont évalués dans une grille dense qui recouvre l'image entière du sujet. La factorisation non-négative de matrice permet de minimiser l'influence du fond. Ceci donne un vecteur de taille 720D pour chaque image. Dans les deux cas une base d'exemples représentatifs est utilisée pour apprendre les paramètres de la représentation.

Dans ces représentations, le problème d'estimation de pose est reporté à la régression du vecteur de pose $\mathbf{x}$ à partir du vecteur de descripteurs image $\mathbf{z}$. L'approche est illustrée sur la figure 1. Nous apprenons une fonction qui va de l'image d'entrée à la pose du corps à partir d'un ensemble d'images d'apprentissage et des poses associés $\{(\mathbf{z}_i, \mathbf{x}_i) \,|\, i = 1 \ldots n\}$. Dans notre cas la fonction est toujours une combinaison linéaire d'un ensemble de fonctions de base prédéfinies :

$$\mathbf{x} \;=\; \sum_{k=1}^{p} \mathbf{a}_k \, \phi_k(\mathbf{z}) + \boldsymbol{\epsilon} \;\equiv\; \mathbf{A}\,\mathbf{f}(\mathbf{z}) + \boldsymbol{\epsilon}$$

Ici, $\{\phi_k(\mathbf{z}) \,|\, k = 1 \ldots p\}$ sont les fonctions de base et $\mathbf{f}(\mathbf{z}) = \begin{pmatrix} \phi_1(\mathbf{z}) & \phi_2(\mathbf{z}) & \cdots & \phi_p(\mathbf{z}) \end{pmatrix}^{\top}$ est un vecteur qui les regroupe. $\mathbf{a}_k$ sont les paramètres à estimer, $\mathbf{A} \equiv \begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_p \end{pmatrix}$ est une matrice qui les regroupe et $\boldsymbol{\epsilon}$ est l'erreur. $\mathbf{A}$ est à estimer avec une méthode de régression. Nous en avons évaluée plusieurs, et notamment la méthode *relevance vector machine* qui à l'avantage de donner une solution éparse qui peut être évaluée rapidement.

Cette méthode s'applique aux images individuelles. Afin de lui étendre au suivi du mouvement dans les séquences vidéo, nous introduisons une prévision dynamique auto-régressive $\check{\mathbf{x}}_t$ dans une régression modifiée :

$$\check{\mathbf{x}}_t \;\equiv\; (\mathbf{I} + \mathbf{A})(2\hat{\mathbf{x}}_{t-1} - \hat{\mathbf{x}}_{t-2}) + \mathbf{B}\,\hat{\mathbf{x}}_{t-1}$$

$$\hat{\mathbf{x}}_t \;=\; \mathbf{C}\,\check{\mathbf{x}}_t + \sum_{k=1}^{p} \mathbf{d}_k \, \phi_k(\check{\mathbf{x}}_t, \mathbf{z}_t) \;\equiv\; \begin{pmatrix} \mathbf{C} & \mathbf{D} \end{pmatrix} \begin{pmatrix} \check{\mathbf{x}}_t \\ \mathbf{f}(\check{\mathbf{x}}_t, \mathbf{z}_t) \end{pmatrix}$$

À chaque étape t, une première estimation $\check{\mathbf{x}}_t$ de l'état $\mathbf{x}_t$ est obtenue à partir des deux vecteurs de pose précédents en utilisant un modèle dynamique auto-régressif. $\check{\mathbf{x}}_t$ entre dans le calcul des fonctions de base, qui prennent maintenant la forme $\{\phi_k(\check{\mathbf{x}}, \mathbf{z}) \,|\, k = 1 \ldots p\}$. Cette forme (nonlinéaire en $\check{\mathbf{x}}_t$ et $\mathbf{z}_t$) permet de lever les ambiguïtés dans le cas où il y a plusieurs reconstructions de pose possibles. Les paramètres $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ et $\mathbf{D}$ sont estimés par régression. Les expériences adoptent un noyau Gaussien comme fonction de base.
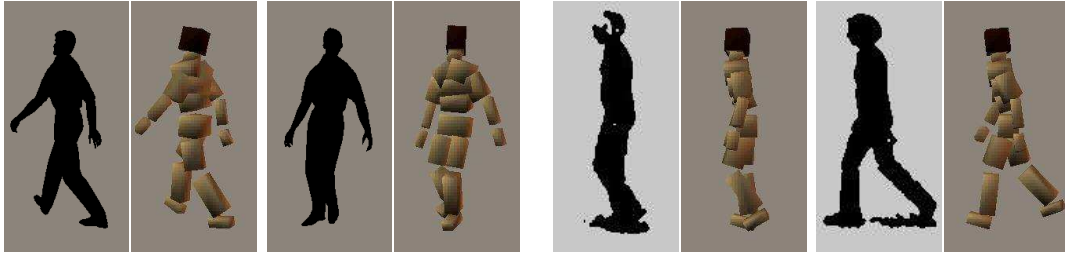
Figure 3: *Quelques exemples de la reconstruction de pose à partir de silhouettes avec un régresseur éparse basée sur le noyau gaussien. L'estimation est correcte dans environ 85% du temps, mais incorrecte dans l'autre 15% en raison des ambiguïtés de la représentation silhouette.*
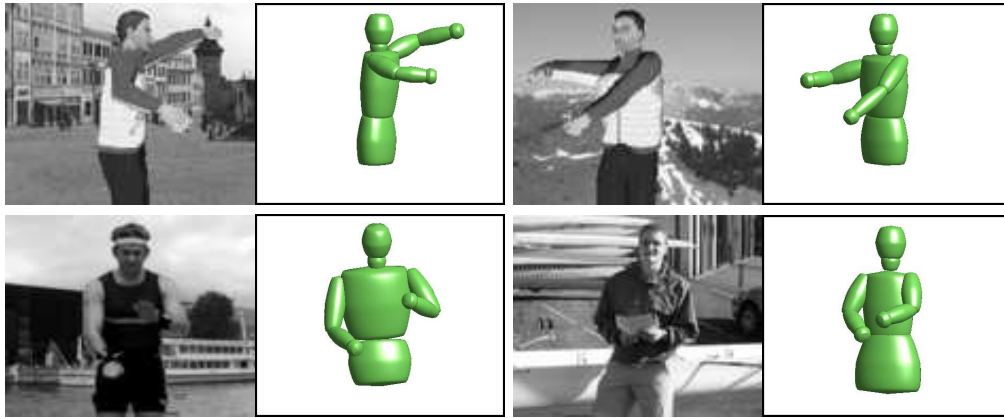
Figure 4: *L'estimation de la pose humaine à partir d'images encombrées. Les résultats sont obtenus par régression d'une représentation factorisation non-négative de matrice basée sur une grille de descripteurs de gradient orientée. La pose est estimée directement à partir des descripteurs d'entrée, sans segmentation préalable.*

2. Résultats expérimentaux

Nous avons comparé la performance de trois méthodes d'estimation de pose à partir des descripteurs: la *relevance vector machine*, la *support vector machine* et la *ridge regression*. Les résultats obtenus sont en effet très similaires. La *support vector machine* donne la meilleure précision mais la *relevance vector machine* donne des solutions très similaires qui sont beaucoup plus éparses. Par exemple, pour la régression du corps entier à partir des noyaux gaussiens, environ 6% des noyaux sont retenus. Ce qui veut dire que l'évaluation du modèle est 15 fois plus rapide. La figure 3 illustre quelques résultats obtenus avec la régression éparse de pose basée sur les silhouettes. En pratique la représentation s'est montré être assez robuste aux erreurs d'extraction de la silhouette. L'erreur d'estimation moyenne sur toutes les articulations pour une séquence typique est environ 6.0°.

Cependant, l'estimation de pose à partir d'une seule silhouette n'est pas entièrement satisfaisante. Quoiqu'elle donne la bonne solution dans environ 85% des cas, la solution est incorrecte dans l'autre 15% en raison des ambiguïtés de la représentation silhouette — plusieurs solutions sont valables et la méthode en choisit la mauvaise. Nous avons étudié deux façons de corriger ces erreurs. Pour les séquences d'images, l'introduction du suivi dynamique permet d'éviter la plupart de ces
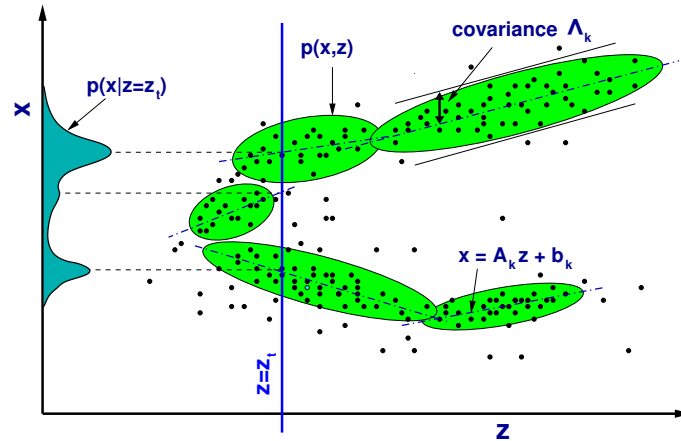
Figure 5: *Une illustration du modèle « mélange de régresseurs » pour l'estimation de la densité conditionnelle $p(\mathbf{x} \,|\, \mathbf{z})$.*

erreurs et semble donner des résultats satisfaisants en pratique. Quelques résultats sont montrés dans le chapitre 4. Pour les images individuelles, nous allons voir prochainement comment une modélisation probabiliste permet de prédire les plusieurs solutions possibles dans les cas ambïgués.

En ce qui concerne les images encombrées, la figure 4 montre quelques exemples d'estimation de pose à partir de la représentation factorisation non négative de matrice sur les descripteurs histogramme de gradients orientés, comme décrit ci-dessus.

L'estimation de pose multi-valeurs probabiliste

1. Approche

L'estimation du mouvement humaine 3D à partir d'images monoculaires doit souvent faire face aux solutions multiples en raison des ambiguïtés intrinsèques du problème. Le problème est plus fréquent avec la représentation silhouette. Dans le cas des séquences vidéo, le chapitre 4 résout ce problème par biais de l'incorporation d'un modèle dynamique. Le chapitre 5 présente une seconde approche qui génère une liste des hypothèses de pose possibles à partir d'une image statique. L'idée de base est d'apprendre plusieurs régresseurs, dont chacun corresponde à un ensemble réduit d'exemples qui est sans ambiguïté. Leurs réponses peuvent alors être proposées en tant de mélange probabiliste codée par une variable cachée, $\mathbf{l}$. Étant donné la valeur de $\mathbf{l}$, la prédiction de la pose est représentée comme une distribution gaussienne centrée sur la prévision du $\mathbf{l}$-ème régresseur $\mathbf{r_l}(\mathbf{z})$:

$$p(\mathbf{x} \,|\, \mathbf{z}, \mathbf{l}) = \mathcal{N}(\mathbf{r_l}(\mathbf{z}), \boldsymbol{\Lambda_l})$$

La distribution complète de la pose est alors obtenue par marginalisation sur les valeurs discrètes de la variable caché :

$$p(\mathbf{x} \,|\, \mathbf{z}) = \sum_{k=1}^{K} p(\mathbf{l} = k \,|\, \mathbf{z}) . \, \mathcal{N}(\mathbf{r}_k(\mathbf{z}), \boldsymbol{\Lambda}_k)$$

Ceci donne un mélange de regresseurs, appelé aussi *mélange d'experts*. Les paramètres du modèle sont estimés par l'*Expectation-Maximisation* avec un modèle de covariance adapté. Chaque regresseur est modélisé par une fonction linéaire du vecteur de descripteurs $\mathbf{z}$ (une représentation très non-linéaire de l'image) : $\mathbf{r}_k(\mathbf{z}) = \mathbf{A}_k \, \Psi(\mathbf{z}) + \mathbf{b}_k$.
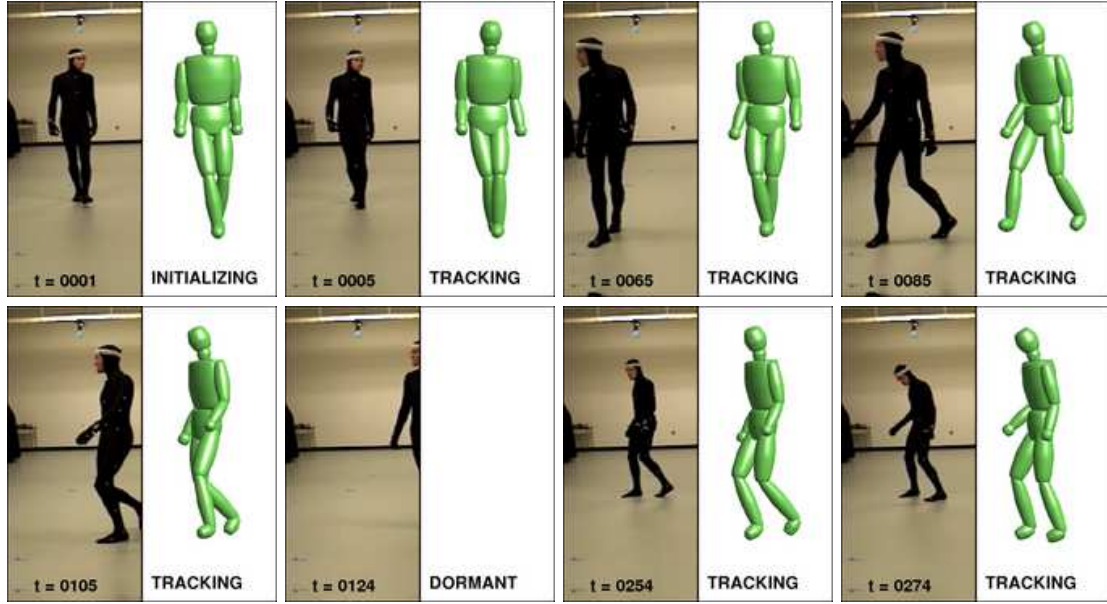
Figure 6: *Quelques exemples du suivi de mouvement dans l'approche hypothèses multiples. L'estimation directe et probabiliste de la pose permet l'initialisation automatique, aussi que la réinitialisation si la détection échoue ou le sujet sorte de l'image.*

2. Résultats expérimentaux

En pratique, nous trouvons que le modèle mélange de régresseurs retourne une seule solution probable pour environ 60% des images, deux solutions pour 30%, et trois solutions ou plus pour 10%. Cette information peut être exploitée par exemple par une méthode de suivi de mouvements probabiliste multi-hypothèses. Quelques exemples de la prédiction de la pose la plus probable sont illustrés sur la figure 6. Le suivi peut s'initialiser automatiquement parce que la propagation temporelle n'est plus nécessaire pour estimer les poses qui sont actuellement possibles. Ainsi, la méthode détecte des échecs du suivi de façon probabiliste et se réinitialise. Le mélange de régresseurs peut aussi être utilisé pour identifier la classe d'action en cours. Voir le chapitre 5 pour les détails.

La modélisation dynamique 2D

1. Approche

Les mouvements humains peuvent être assez complexes et il est souvent difficile de les suivre dans les images encombrées. Afin d'améliorer le suivi nous avons développé une approche qui caractérise mieux les mouvements basée sur un mélange de modèles dynamiques locaux. L'approche permet notamment de suivre mieux les transitions entre différentes classes de mouvement.

Afin d'exploiter les corrélations entre les mouvements des membres du corps et de stabiliser l'estimation, chaque modèle dynamique est appris dans son propre sous-espace de dimension réduite. Chaque modèle est un processus linéaire auto-régressif. Aux transitions entre les modèles, les prédictions sont combinés de façon linéaire avec des poids probabilistes :

$$\check{\mathbf{x}}_t^k = \sum_{i=1}^{p} \mathbf{A}_i^k \, \mathbf{x}_{t-i} + \mathbf{w}_t^k + \mathbf{v}_t^k, \qquad \hat{\mathbf{x}}_t = \sum_{k=1}^{K} p(k \,|\, \mathbf{x}_{t-1}) \, . \, \check{\mathbf{x}}_t^k$$
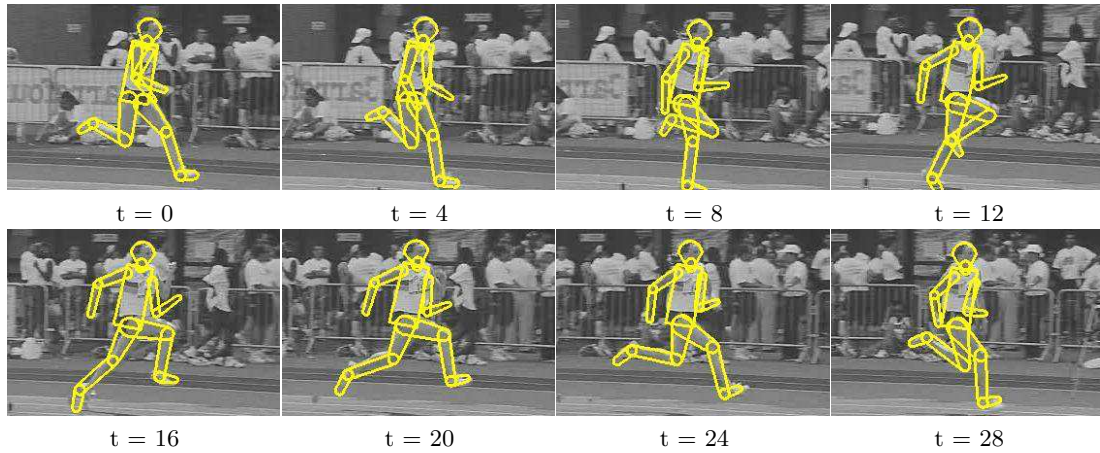
Figure 7: *Un exemple de suivi de mouvement sportif. Le modèle dynamique a été appris à partir d'une athlète différente effectuant un mouvement semblable. La connaissance préable représenté par le modèle dynamique permet de suivre les différents membres en présence d'un fond encombré.*

Les modèles sont appris à partir d'images de mouvement humain 2D qui ont été étiquetés à la main. L'algorithme d'apprentissage commence par un groupement initial de poses par k-means. Dans chaque groupe l'Analyse en Composantes Principales (*PCA*) projette les vecteurs de pose dans un sous-espace de dimension inférieure. Dans chaque groupe un modèle linéaire auto-régressif pour le vecteur de pose réduit est appris sachant les poses réduites des p images précédentes. En pratique, $p = 1$ ou $p = 2$ suffit. Ensuite, les exemples sont regroupées selon la précision du modèle local pour chaque point, tenant aussi en compte la continuité spatiale des points dans chaque modèle. Le processus est itéré jusqu'à la convergence selon la façon *Expectation-Maximisation*.

2. Résultats expérimentaux

Afin d'effecteur un suivi robuste du mouvement humain, le modèle dynamique est utilisé de façon probabiliste, pour tirer des échantillons selon la distribution prévisionnelle dans un cadre de suivi multi-hypothèses. Chaque prévision de pose est ensuite optimisée localement afin de maximiser sa correspondance à l'image. Dans les expériences décrites nous employons un modèle 2D « scaled prismatic model » du corps humain et la correspondance modèle-image est établie de façon séquentielle descendent, c-à-d chaque membre du corps est mis en correspondance avec l'image dans son tour, en descendant membre par membre l'arbre cinématique du corps. Nous observons que la mise en place de priors spécialisés pour les différentes régions de l'espace des poses aide le suivi du mouvement. Un exemple de suivi du cours d'un athlète basée sur ce modèle est montré dans la figure 7.

Représentation d'image basée sur la co-occurrence à niveaux multiples

La dernière contribution de cette thèse passe au problème général des représentations pour la reconnaissance d'objets. Nous développons les « hyperfeatures » – une nouvelle classe d'indices d'image basée sur la quantification à plusieurs niveaux des co-occurrences d'éléments. Nous démontrons l'efficacité de représentation pour la classification de scènes et pour la reconnaissance d'objets structurés.
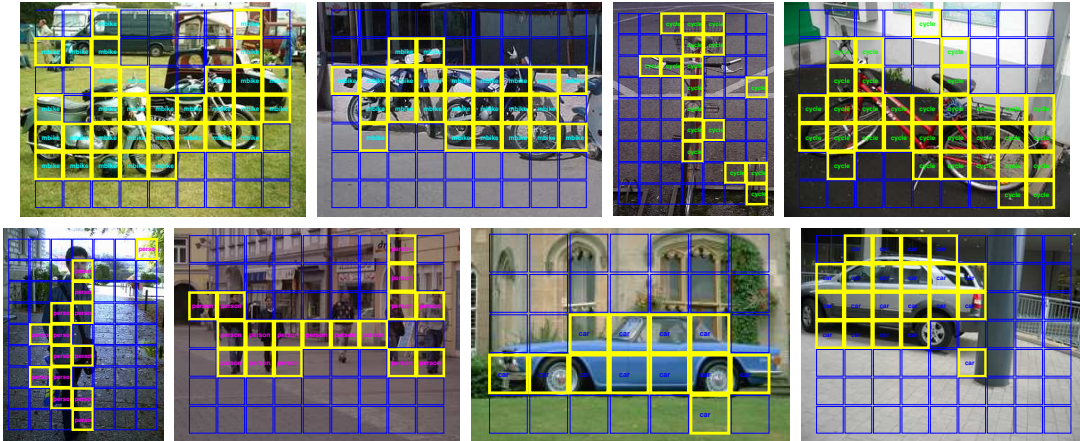
Figure 8: *La localisation d'objets avec les indices « hyperfeature ». Chaque région est représentée par l'ensemble de ses hyperfeatures et classée indépendamment en utilisant un classifieur linéaire pour chaque classe.*

1. Approche

Histogrammer les descripteurs locaux d'apparence quantifiés pourvoir une représentation simple et performante pour la reconnaissance visuelle. Ces représentations sont souvent assez discriminantes et elles sont résistent aux occultations locales et aux variations géométriques et photométriques, mais elles n'exploitent pas explicitement la forme spatiale des objets à reconnaître. Nous avons développé une nouvelle représentation visuelle, les *hyperfeatures*, qui incorporent plus d'information spatial locale en exploitant la coincidence spatiale à plusieurs niveaux. Le point de départ est l'idée familière de construire un modèle d'objet à partir d'une hiérarchie de parts locales et simples mais discriminantes. Afin de détecter des parties d'objet, il suffit souvent en pratique de détecter la co-occurrence de leurs fragments plus locaux. Le processus peut être formalisé comme la comparaison (par exemple par quantification vectorielle) des descripteurs d'image contre un « vocabulaire » de descripteurs connus, suivi par l'agrégation des statistiques locales de ces comparaisons. Il transforme une collection locale de vecteurs de descripteur d'image en un vecteur un peu moins locale d'histogrammes : un descripteur de plus haut niveau mais moins bien localisé dans l'image. Puisque le résultat est encore une fois un vecteur local de descripteurs, le processus peut être répété de façon récursive afin de coder de parties de plus en plus grandes de l'objet. À chaque itération le niveau d'abstraction monte et on peut espérer que les niveaux supérieurs représentent des propriétés sémantiques de l'image. Le chapitre 8 présente l'algorithme de construction des *hyperfeatures* et étude son comportement avec plusieurs différentes codages de base de l'image.

2. Résultats expérimentaux

Les nouveaux descripteurs ont été évalués dans le cadre de la classification d'images et de la localisation d'objets dans l'images. Nous avons comparé plusieurs algorithmes de codage — la quantification vectorielle et les mélanges de gaussiennes, avec ou sans le *Latent Dirichlet Allocation* (une méthode qui caractérise les aspects latents des données). Les résultats pour l'identification de diverses catégories d'objet (voitures, motocycles, vélos, images de texture) sont présentés dans le chapitre 8. Le codage hyperfeatures améliore les résultats notamment dans le cas d'objets à forte structure géométrique tels que les motocycles et les voitures. La représentation hyperfeatures est aussi utile pour classer les régions locales d'image selon les objets qui les contiennent. La figure 8 montre l'identification d'objets basée sur les hyperfeatures avec un SVM linéaire pour la classification.

Conclusions et perspectives

Cette thèse a abordée plusieurs aspects de l'interprétation d'images et en particulier de la reconnaissance du mouvement humaine. Nous avons montré qu'un système efficace pour la reconstruction de la pose humaine à partir d'images monoculaires peut être construit en conjuguant les idées de la vision par ordinateur, de la capture de mouvement, et de l'apprentissage automatique. Le système utilise une modélisation statistique basée sur une collection d'enregistrements issus de la capture de mouvement et les images correspondantes. Au delà des contributions directes, cette approche à des atouts qui fait d'elle une fondation intéressante pour la recherche à venir dans ce domaine.

La capture de mouvements de manière efficace. Dans les environnements connus où le fond est fixe ou peut être estimé, la capture basée sur les silhouettes s'est montrée efficace malgré la perte d'information liée à la utilisation des silhouettes. La régression éparse basée sur une représentation noyau permet l'estimation relativement précise de la pose et le suivi efficace du mouvement. Cette approche convient à plusieurs applications et nous avons démontré les résultats sur plusieurs types de mouvement de marche et aussi sur les gestes de bras.

Estimation de pose probabiliste à partir d'images statiques. L'approche mélange de régresseurs pourvoit une évaluation probabiliste des différentes poses 3D qui sont susceptibles à correspondre à l'image statique donnée. A notre connaissance, c'est la première fois qu'à été proposée un modèle probabiliste explicite des solutions multiples crées par les ambiguïtés de la reconstruction de la pose 3D à partir d'images monoculaires. La méthode a des applications directes sur le suivi robuste d'actions humains et sur l'identification de geste.

Reconstruction de la pose dans des images encombrées. Nous avons présenté une nouvelle représentation qui exploite la factorisation non-négative de matrice afin de supprimer l'influence du fond. Contrairement aux approches précédentes, la régression à partir de ces descripteurs permet l'évaluation de la pose humaine dans les images encombrées, sans modèle explicite du corps et sans exiger une segmentation antérieure du sujet.

Hyperfeatures. Nous avons présenté le modèle hyperfeatures, un nouveau méthode d'extraction d'indices d'image pour la reconnaissance basée sur la co-occurence à plusieurs niveaux. Nos expériences démontrent que l'introduction d'un ou plusieurs niveaux d'hyperfeatures améliore les résultats dans plusieurs problèmes de classification, notamment pour les classes qui ont une structure géométrique prononcée. La représentation peut aussi être utile pour trouver des personnes dans les images et pour estimer leurs poses.

Extensions possibles

Le travail présenté dans cette thèse résout plusieurs problèmes de l'estimation du mouvement humain à partir d'images et il ouvre la voie à plusieurs applications directes. Cependant, il reste un certain nombre de limitations et plusieurs extensions peuvent être envisagées. D'abord on peut mieux exploiter la structure du corps humain. Nos algorithmes actuels évitent toute utilisation d'un modèle explicite du corps humain et représentent les poses sous forme de simples vecteurs. Cette représentation ramene l'inférence à un ensemble d'opérations linéaires (sur une représentation non-linéaire d'image) et ainsi permet l'estimation rapide de la pose, mais elle est pour cette raison aveugle à la plupart de la structure de l'espace de poses. Il serait intéressant de voir si on ne pouvait pas faire mieux par biais d'une modélisation plus structurée, sans pourtant établir un modèle explicite et détaillé du corps. Par exemple, l'arbre cinématique du corps peut être déduit automatiquement en apprenant un modèle graphique structurel. En fait, la structure n'est pas limitée a un arbre : l'approche apprentissage automatique peut aussi apprendre un graphe général qui

exprime les dépendances non-locales telles que la coordination entre membres qui est fondamentale à beaucoup de mouvements humains.

Deuxièmement, nos méthodes actuelles n'atteignent pas la même précision que les systèmes commerciaux de capture de mouvement. Une grande partie de cette imprécision peut être attribuée aux difficultés intrinsèques de la capture de mouvement à partir d'images monoculaires. Cependant, faute de modèle explicite la distribution (nécessairement creuse) d'exemples d'apprentissage limite la précision atteignable et par conséquent la méthode n'arrive pas toujours a trouver un alignement parfait avec l'image d'entrée. L'introduction d'un modèle explicite du corps permettrait la reprojection directe de la pose reconstruite dans l'image, et par conséquent peut permettre l'ajustement plus fin de la solution.

Finalement, il y a de nombreuses façons d'apprendre une représentation qui code l'information exigée par la reconnaissance visuelle au sens large. Les hyperfeatures ne sont qu'une des approches possibles pour la représentation de classes visuelles qui ont de la structure géométrique, et beaucoup d'autres représentations peut être étudiées. Par exemple, une utilisation plus structurelle de l'extraction d'information sémantique latente à partir de descripteurs d'image de base semble être une voie prometteuse pour la reconnaissance de classes génériques d'objets.

1

Introduction

The first electronic computers were rare and bulky devices that were essentially used as calculators. Seven decades later, computers are in everyday use in many homes and offices, and they are frequently integrated with other technologies for applications in health care, communication, industrial automation and scientific research. As these machines have become more advanced and gained more and more capabilities, they have become an integral part of our lives. The next generation of computers will be yet more advanced, containing intelligent software and having many more capabilities. One of these capabilities, which has been the goal of computer vision research for many years, is that of *seeing*, *i.e.* performing automated processing of visual information captured by means of a camera.

Computer vision based systems are currently used in applications such as medical imaging, automated manufacturing and autonomous vehicles. Enabling their more widespread use in our day to day lives requires a number of research problems to be solved. A large part of current computer vision research deals with developing techniques for automatically recognizing the contents of images or videos *e.g.* understanding a scene, identifying the objects in it, detecting people and interpreting their actions. Methods for analyzing images of people have received a lot of attention and constitute a whole research area in their own right owing to the large number of potential applications. For instance, effective algorithms for interpreting the movements of people would allow more natural human machine interaction, smarter security, home and driver monitoring systems, improved sports training by analysis of training videos, and many other applications. However automatically inferring human movements from a signal as complex as a video stream remains a challenging problem.

The technology of recording human body movements is often called motion capture. It originally developed as an analysis tool in biomechanics and medical research, but it has grown increasingly important as a source of animation data for film and video game production. Currently the best systems are vision based, but they require multiple specialized cameras, carefully controlled lighting, and special costumes with reflectors or active markers attached to the body joints. Mechanical systems that strap onto the body and magnetic sensors also exist.

1.1 The Problem

This thesis addresses the problem of *markerless*, *monocular* image-based motion capture. It develops techniques for automatically inferring the subject's 3-dimensional body *pose* — the configuration of his or her limbs, trunk and head — from a single image or video stream. The pose is encoded by a set of numbers that quantify the relative positions and orientations of the different limbs, one
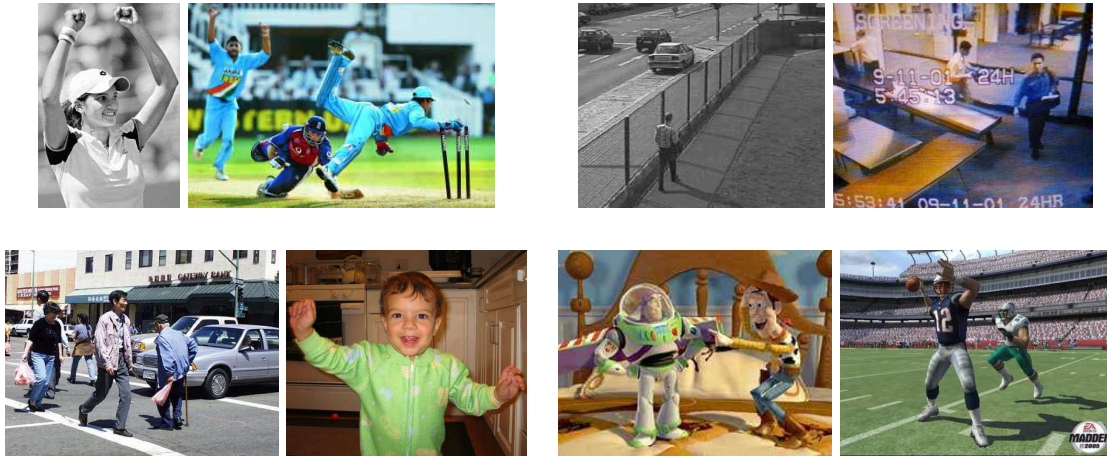
Figure 1.1: *Could an intelligent machine understand what is happening in these images? Automated understanding of human motion from images has applications ranging from analyzing the content of sports video to developing smart visual surveillance systems for areas of high security, monitoring everyday activities and allowing natural interaction between people and machines. One of the challenges involved is that of recovering the pose of the human body in these images,* i.e. *image-based motion capture. Recorded human movements from motion capture are currently used for motion synthesis in film special effects and computer games.*

possible encoding being the output of a conventional motion capture system. The reconstruction problem thus takes the form of estimating a high dimensional parametric state (the body pose, or for video sequences the temporal sequence of body poses) from a complex input signal (the image).

Estimating the pose of a person from images has several direct applications. In an online mode, it can be used for human motion analysis for automated visual surveillance or as an input device for human-computer interaction. In an offline mode, it is useful for the analysis and annotation of image and video content, and it would greatly simplify the setup and reduce the costs of motion capture for film and game production.

We deal specifically with the problem of reconstructing 3D pose and motion from monocular images or video, *i.e.* using input from a single camera rather than a stereo or multi-camera setup. Inference from monocular sequences is challenging because some of the 3D information is not available directly — in particular the depth (the distance from the camera to the body part in question) can not be estimated directly — but it greatly simplifies the process of data acquisition and it potentially allows both archived film and video footage and rushes shot directly with the final production camera to be used. Below we discuss some of the issues that must be addressed to solve this problem.

1.1.1   Issues Involved

Estimating human pose from monocular images involves several difficult issues that make it a challenging problem.

The foremost is handling very large variations in the image signal that arise as a result of variability in human appearance and in the conditions under which the images are taken. People have a range of physiques and wear different types of clothing, the deformability of which often adds to the problem. Changes of lighting and camera viewpoint cause effects like shadows and other variations
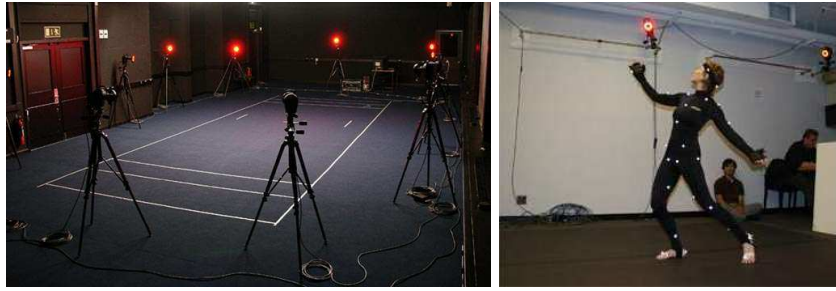
Figure 1.2: *A standard multicamera optical motion capture setup. (Left) Several cameras are used to record the position and motion of different joints on the body. (Right) A special costume with reflective markers located at the joint positions is used by an actor who performs the motion.*

in image appearance (*e.g.* see figure 1.1). All of these factors make it non-trivial to reliably identify image features that can be cued on to extract relevant information.

Secondly, the problem of obtaining a three-dimensional pose from monocular two-dimensional images is geometrically under-defined. Projecting a 3D scene onto a 2D image plane suppresses depth information and thus makes the 3D recovery ambiguous. Also, in monocular images, parts of the body are often unobservable due to self occlusions. As a result, there are often multiple pose solutions for a given image. Such ambiguities can be reduced by exploiting prior knowledge about typical body poses.

Thirdly, the human body has many degrees of freedom. A simplistic skeletal body model typically requires between 30 and 60 parameters to characterize pose in terms of limb positions and orientations, so inference must take place over a high dimensional space of possible 3D configurations, which makes the process yet more complicated.

1.2 A Brief Background

Traditionally, there has been a keen interest in human movement from a variety of disciplines. Human perception of motion and gestures has been studied in psychology. The field of biomechanics deals with understanding the mechanical functioning of the human body by studying the forces and torques involved in different movements, and such movements are mimicked in humanoid robotics. In the field of computer graphics, synthesis of human movements has applications in animation.

The analysis of human movements in images and video has been of interest to the computer vision community for many years and there exist a number of approaches to the problem [49]. For estimating human pose from images, many of the past methods make use of explicit models of the human body and attempt to match these to image features. Such models are typically skeletons built from kinematic chains of articulations fleshed out with elements that approximate body shape: volumetric surfaces in 3D approaches, and rectangles or blobs in 2D approaches. They thus recover pose as the locations and orientations of the limbs in the image plane or in 3D. Model based methods are effective in recovering pose from images, but the main limitations are that they involve expensive optimization of image likelihoods that are obtained by projecting or overlaying these models on to the image and they are hard to generalize across appearance variations between people. Moreover, they do not naturally incorporate prior information about typical human pose and motion patterns. An alternative approach eschews explicit body models and instead describes human movement directly or indirectly in terms of examples. This may

involve matching the image (or a region of interest) to a set of key images or 'exemplars' to deduce
a non-parametric form of pose. More recently, there has also been research on using motion capture
data to recover detailed parametric body pose. Motion capture is extremely useful in providing
human movement recordings that may be exploited for modeling typical poses and motion. In
its conventional form, however, it requires an expensive and complicated setup including multiple
cameras and special body suits (see figure 1.2). Combining machine learning with motion capture
technology and computer vision techniques can enable the recovery of human pose directly from
image observations, simplifying the setup and avoiding the matching of explicit shape or body
models to images.

1.3   Machine Learning and Motion Capture

Machine Learning uses collections of samples from a process to derive effective mathematical models
for the process. The model is often statistical and is thus useful in cases where the process is
difficult to model exactly. Even when partial models are available, learning techniques can play
an important role in verifying different models and estimating their parameters by fitting them to
the data.

In markerless motion capture, the need for accurate modeling and rendering of the human body can
be avoided by learning to predict 3D configurations directly from the observed images. This requires
a set of images and their associated configurations as training data. One way to produce such data
is by using existing motion capture technology to record the movements of subjects performing a
range of different activities. Such systems can usually provide the body configuration as a set of
angles between members at major body articulations, but the details of the representation vary
between systems. The corresponding images can be obtained either by using a conventional video
camera synchronized with the cameras of the motion capture system, or by rendering artificial
images based on the motion capture data as is done for motion synthesis in graphics. The machine
learning based methods then condense this data into mathematical models that can be used to
predict pose from new images.

An important issue in learning from images is how to represent their content. Typically some
kind of low-level visual features are extracted and gathered into a vector to create a *feature space*
representation for input to the learning method. As mentioned in § 1.1.1, there are a number
of factors that make it difficult to identify a set of features that is suitable for capturing human
appearance. However it turns out that we can learn effective representations by modeling the
statistics of images and basic feature responses on them, using a collection of images as training
data.

The data-driven machine learning approach has several advantages in our context. As explained
above, it can avoid the need for accurate 3D modeling and rendering, which are both computation-
ally expensive and difficult to generalize across appearance variations. Its use of training data from
real human motions means that it captures the set of typical human poses. This set is far smaller
than the set of all kinematically possible ones, which stabilizes the solution and helps to reduce the
intrinsic ambiguities of monocular pose estimation by ruling out implausible configurations. Statis-
tical models of human motion and appearance also allow uncertainty to be incorporated, making it
possible to perform probabilistic inference and estimate confidence measures on the output based
on what has been observed in the training data.

Another facet of human motion analysis that can benefit from learning based methods is dynamical
modeling. Building models of human motion analytically requires an understanding of biomechan-
ics and the complicated inter-dependencies of motion within various parts of the body. On the
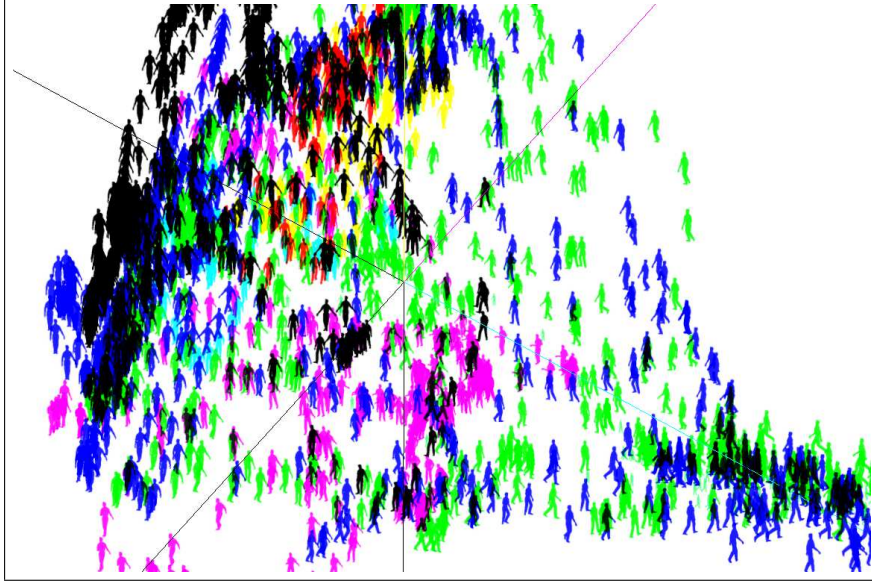
Figure 1.3*: A projection of the manifold of human silhouettes in a feature space that encodes silhouettes using robust shape descriptors. The encoding is used to map silhouettes into a high dimensional space where Euclidean distance measures the similarity between two silhouettes. Such an encoding allows 3D body pose to be recovered by using direct regression on the descriptors.*

other hand, statistical models can easily capture the temporal dependencies and the correlations between movements of different body parts. Machine learning technology has begun to be used to synthesize natural looking motion for human animation *e.g.* [86, 124, 41].

1.4 Overview of Approach

In this thesis, we take a learning-based approach to motion capture, using regression as a basic tool to distill a large training database of 3D poses and corresponding images into a compact model that has good generalization to unseen examples. We use a bottom-up approach in which the underlying pose is predicted directly from a feature-based image representation, without directly modeling the generative process of image formation from the body configuration. The method is purely data-driven and does not make use of any explicit human body model or prior labeling of body parts in the image.

We represent the target 3D body pose by a vector, denoted $\mathbf{x}$. In our experiments, we simply use native motion capture pose descriptions. These are in the form of joint angles or 3D coordinates of the body joints, but any other representation is applicable. The input image is also represented in a vectorized form denoted by $\mathbf{z}$. Given the high dimensionality and intrinsic ambiguity of the monocular pose estimation problem, active selection of appropriate image features is critical for success. We use the training images to learn suitable image representations specific to capturing human body shape and appearance. Two kinds of representation have been studied. In cases where foreground-background information is available, we use background subtraction based segmentation to obtain the human silhouette, and encode this in terms of the distribution of its softly vector quantized local shape context descriptors [16], with the vector quantization centres being learned from a representative set of human body shapes. This transforms each silhouette to a point in a
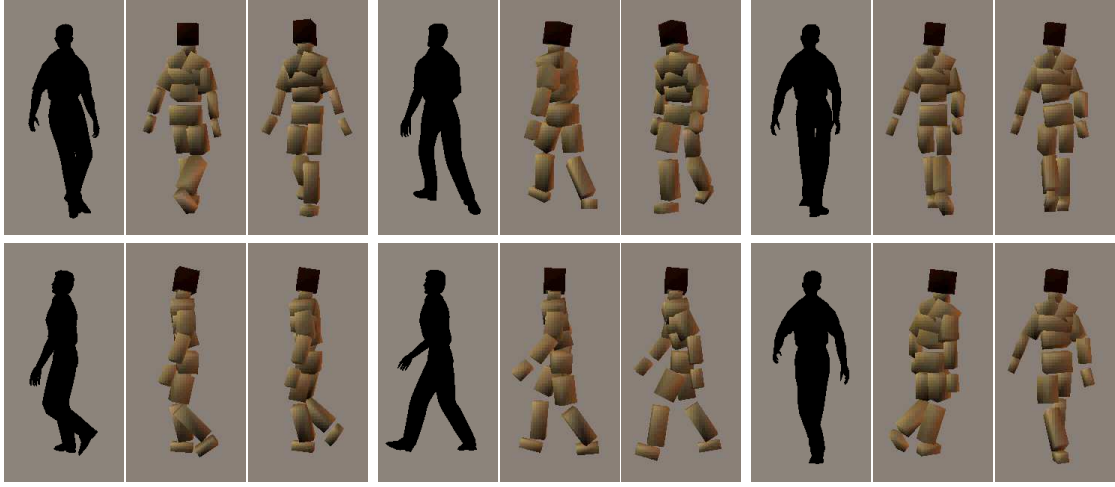
Figure 1.4: *Silhouettes are an effective representation for estimating body pose from an image, but add to the problem of ambiguities in the solution because left and right limbs are sometimes indistinguishable. Here we see some examples of multiple 3D pose solutions that are obtained from such confusing silhouettes using a mixture of regressors method developed in this thesis. Cases of forward/backward ambiguity, kinematic flipping of the legs and interchanging labels between them are seen here. In the last example, the method misestimates the pose in one of the solutions.*

100D space of characteristic silhouette shapes. A 3D projection of this space is shown in figure 1.3. In cases where background subtraction is not available, we use the input image directly, computing histograms of gradient orientations on local patches densely over the entire image (*c.f.* the SIFT [90] and HOG [32] descriptors). These are then re-encoded to suppress the contributions of background clutter using a basis learned using Non-negative Matrix Factorization [85] on training data. In our implementation this gives a 720D vector for the image.

The pose recovery problem reduces to estimating the pose $\mathbf{x}$ from the vectorized image representation $\mathbf{z}$. We formulate several different models of regression for this. Given a set of labeled training examples $\{(\mathbf{z}_i, \mathbf{x}_i) \mid i = 1 \ldots n\}$, we use the Relevance Vector Machine [151] to learn a smooth reconstruction function[1] $\mathbf{x} = \mathbf{r}(\mathbf{z})$, valid over the region spanned by the training points. The function is a weighted linear combination $\mathbf{r}(\mathbf{z}) \equiv \sum_k \mathbf{a}_k \, \phi_k(\mathbf{z})$ of a prespecified set of scalar basis functions $\{\phi_k(\mathbf{z}) \mid k = 1 \ldots p\}$.

When we use the method in a tracking framework, we can extend the functional form to incorporate an approximate preliminary pose estimate $\check{\mathbf{x}}$, $\mathbf{x} = \mathbf{r}(\check{\mathbf{x}}, \mathbf{z})$. This helps to maintain temporal continuity and to disambiguate pose in cases where there are several possible reconstructions. At each time step t, a state estimate $\check{\mathbf{x}}_t$ is obtained from the previous two pose vectors using an autoregressive dynamical model, and this is used to compute the basis functions, which now take the form $\{\phi_k(\check{\mathbf{x}}, \mathbf{z}) \mid k = 1 \ldots p\}$.

Our regression solutions are well-regularized in the sense that the weight vectors $\mathbf{a}_k$ are damped to control over-fitting, and sparse in the sense that many of them are zero. Sparsity is ensured by the use of the Relevance Vector Machine, a learning method that actively selects the most 'relevant' basis functions — the ones that really need to have nonzero coefficients for the successful

---

[1]*I.e.* a function that directly encodes the *inverse* mapping from image to body pose. The forward mapping from body pose to image observations can be more easily explained by projecting a human body model or learning image likelihoods. In this thesis, we avoid the use of such a forward mapping.
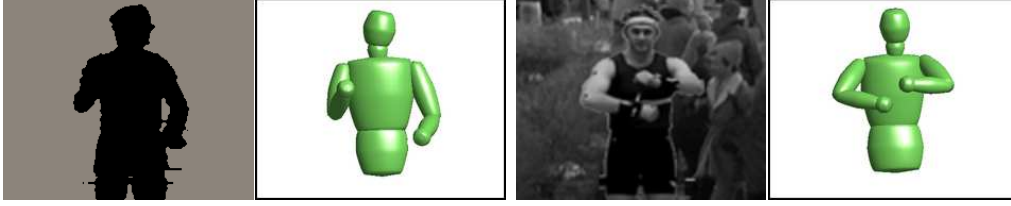
Figure 1.5: Examples of 3D pose reconstruction from single natural images. The subject is a person making arm gestures. The reconstructions are obtained using the regression based methods developed in this thesis which can take either a silhouette shape or a cluttered image an input.

completion of the regression. In a kernel based regression where $\phi_k(\mathbf{z}) \equiv K(\mathbf{z}, \mathbf{z}_k)$ for some kernel function $K(\mathbf{z}_i, \mathbf{z}_j)$ and centres $\mathbf{z}_k$, this selects the most relevant training examples, thus allowing us to prune extraneous examples from a large training dataset, retaining only the minimal subset needed for accurate predictions.

Recovering 3D pose from monocular images is ill-conditioned and subject to ambiguities in the solution. These are reduced significantly by our use of training data to characterize the range of typical body configurations, but there are nevertheless instances where multiple solutions are possible. These problems arise more frequently when the images are represented using silhouettes because distinguishing between left and right limbs and between forward-facing and backwards-facing views can sometimes be very difficult. To deal with this problem, we develop another model that uses a set of nonlinear regressors to predict the different possible pose solutions with their corresponding probabilities. The result in this case is a potentially multimodal posterior probability density $p(\mathbf{x} \,|\, \mathbf{z})$ for the pose, rather than a functional estimate. Some examples of multiple pose solutions obtained using this method are shown in figure 1.4. Figure 1.5 shows some sample results of the most likely pose estimates obtained from image silhouettes and cluttered images using the methods developed in this thesis.

1.5   Thesis Contributions

This thesis studies machine learning based approaches for directly recovering 3D human pose and movement from monocular images. The methods use data generated from conventional marker-based motion capture as their principal source of training examples. There are two main contributions:

1. The framework for efficient model-free tracking and recovery of 3D human pose from images and video sequences using regression based models. This includes algorithms for fast pose estimation from a single image, for completely bottom-up (discriminative) tracking, and for obtaining probabilistic multimodal pose estimates in cases of ambiguity.

2. Two novel image representation schemes are developed. The first shows how to selectively encode useful image information in the presence of background clutter and the second departs from the problem of human pose to introduce *hyperfeatures* — a multi-level visual coding scheme for generic object representation in visual recognition.

Fast bottom-up pose estimation. It is shown that by the use of a sparse kernel based regression and robust image representation, the complete 3D pose of the body may be estimated from a given

image in a very efficient manner. The approach requires neither an explicit body model nor prior labeling of body parts in the image and uses only a fraction of the training database for computing pose from a new image.

Discriminative tracking. We develop a new tracking algorithm that is based on a (kernel) regression model. The algorithm is completely bottom-up and avoids image likelihood computations. Furthermore, it exploits sparsity properties of the Relevance Vector Machine and involves computation that is tractable in real time.

Multimodal pose estimation. A probabilistic method is developed for reconstructing multiple pose solutions from a single image. This is very helpful for dealing with ambiguities involved in monocular pose estimation. The multivalued pose estimates are also used for multiple hypothesis tracking giving rise to a self-initializing and robust human pose tracker.

Selective image encoding in clutter. A new image feature coding mechanism is developed that selectively encodes foreground information in an image by learning to suppress feature components contributed by background clutter. This allows for completely model-free and bottom-up pose 3D estimation from images containing background clutter and is the first method of its kind.

Hyperfeatures. We introduce a new framework for multi-level visual representation that is based on the idea of capturing co-occurrence statistics of local image features at various levels of abstraction. It improves the performance of classical 'bag-of-features' based representations by robustly incorporating spatial information and is shown to give improved performance in image classification and object recognition tasks.

1.6   Thesis Outline

State of the Art. In the next chapter we briefly discuss the various methodologies that have been adopted for solving the human pose estimation problem and present a review of some of the major contributions under each category. The different approaches that exist in the literature are classified according to their top-down or bottom-up nature. We also briefly discuss the main techniques used to track motion in video sequences and to detect people in images.

Learning 3D Pose: Regression on Silhouettes. Chapter 3 introduces a regression framework for learning to recover 3D pose from image silhouettes and shows how sparse kernel based regression can effectively be employed for this problem. We describe an effective silhouette shape representation method and study the accuracy of different regressors in obtaining pose from single images. The conclusion is that the Relevance Vector Machine provides an effective trade-off between accuracy and computational complexity. This work was first published in the *IEEE International Conference on Computer Vision & Pattern Recognition, 2004* [1].

Tracking and Regression. In chapter 4, the preceeding kernel regression based method is extended to track pose through video sequences. The regressor is used to build a *discriminative tracker* that avoids the use of the generative models that are usually required in the visual tracking literature. We present tracking results on different image sequences and show that the approach alleviates the ambiguity problems associated with pose estimation from monocular silhouettes. This chapter is based on a paper that first appeared in the *International Conference on Machine Learning, 2004* [3]. A consolidated description of chapters 3 and 4 also appears in the *IEEE Transactions on Pattern Analysis & Machine Intelligence, 2006* [9].

A Mixture of Regressors. When used to estimate pose, a single regressor often fails in ambiguous cases in which there are several possible 3D pose solutions for a given image. In chapter 5, a probabilistic mixture model is developed that explicitly allows for multiple solutions by returning a mixture of Gaussians probability density for the pose, rather than just a single pose value. We show that the density can be used to construct a robust self-initializing tracker, and also used for gesture recognition. Part of this work is described in a paper presented at the *IEEE Workshop on Vision for Human-Computer Interaction, 2005* [6].

Estimating Pose in Cluttered Images. Chapters 3-5 use a silhouette based representation that requires background estimation and image segmentation as preprocessing steps before estimating the pose. Chapter 6 studies various ways of encoding image information when such segmentations are not available, so that the images input to the pose estimation stage contain significant amounts of background clutter. A selective encoding scheme based on Non-negative Matrix Factorization is adopted. We show that in a setting where the subject has been approximately localized in the image, efficient bottom-up 3D pose estimation based on regression remains possible in the presence of clutter. This chapter is based on a paper that appeared in the *Asian Conference on Computer Vision, 2006* [7].

Modeling Dynamics using Mixtures. In chapter 7, we present some work on tracking in cluttered images. The focus is on learning accurate motion priors for various kinds of human motion. A mixture of autoregressive processes over reduced subspaces is used to probabilistically model transitions between different motion categories. This chapter contains older work that uses a planar body model to track the configuration within the image plane without performing motion capture. It was published in the *European Conference on Computer Vision, 2004* [4].

Towards a Robust Image Representation. Finally, chapter 8 stands back from the problem of human modeling, taking the notion of effective image encoding one step further by describing a multilevel coding method, *hyperfeatures*, that serves as a robust and generic image representation. In contrast to the rest of this thesis, this chapter studies different image coding methods for object recognition and image classification. The proposed algorithm is shown to be effective in several such tasks. The main idea of hyperfeatures is described in a paper that appears in the *European Conference on Computer Vision, 2006* [8].

Conclusion and Perspectives. The thesis ends with a summary of the work and a discussion of our approach. We also discuss some of the shortcomings of our work, giving suggestions for possible future extensions, and highlight some of the open problems in the field of understanding human motion from images.

2

State of the Art

Developing automated techniques for understanding the motion of people in image sequences is a very active research area in computer vision and there exists a vast amount of literature on several aspects of it. Research in the area currently addresses many different issues including reliably detecting the presence of people in images, tracking them through sequences, estimating the complete pose of the human body from image data and modeling its dynamics for synthesizing seemingly natural motion. In this chapter, we start by reviewing state of the art methods for estimating and tracking human body pose in images, which is the focus of this thesis. This is followed by a review of some literature on tracking and modeling the dynamics of human motion and finally an overview of some existing methods for detecting people in images.

2.1 Estimating Human Pose

Estimating the full pose of a human body from images refers to determining the configuration of all major body segments (generally the torso, head & neck, arms and legs) given the image data. This typically involves estimating more than 30 parameters that are used to encode this pose, but different methods work at different levels of complexity which may range from approximately locating the main body segments in an image to reconstructing the complete set of joint-angles in three dimensions. Some methods employ a multiple camera setup for this purpose while others make use of monocular image sequences. We include all these methods in our discussion on pose estimation techniques and classify the methods into two categories depending on whether they take a *top-down* or *bottom-up* approach to the problem.

In the context of human pose estimation from images, a top-down methodology refers to using a (semantically) high-level description of the complete pose to explain a lower-level image signal. This normally involves using a geometrical body model to measure the likelihood of an image, given a description of the body pose. Bottom-up methods, on the other hand, start with lower-level image features and use these to predict higher-level pose information in the form of a prespecified set of parameters. Bottom-up methods may or may not involve the use of a body model and can be further classified as *model-based* or *model-free* on this basis.

This entire thesis is based on a strictly bottom-up approach to the problem, but before detailing the existing literature under this category, we briefly discuss work based on the top-down approach.

2.1.1 Top-down

Methods in this category are based on modeling a likelihood function that explains how likely an image observation is, given a particular pose estimate. An intermediate step involved in building

this likelihood function is actually defining a model of the human body that can be overlayed on the image to predict its appearance. A variety of 2D and 3D human body models have been proposed for this task. A very simple yet popular one is the 'cardboard person' model [67] that uses a collection of articulated planar patches to represent body segments. These patches are parametrized by their affine deformation parameters and are used to match an expected appearance of each segment with that observed in the image. A similar model called the 'scaled prismatic model' [101, 29] includes limb lengths in the parametrization to explicitly allow for limb foreshortening effects caused by 3D movements perpendicular to the image plane, and is hence often referred to as being '2.5D'. Both of these models allow easy appearance modeling as they can make use of separate patch templates for each segment. However, they are evidently restricted to the image plane in terms of representing pose. 3D body models use volumetric representations of body segments and include parameters for the complete motion of a kinematic chain in three-dimensional space. Some examples of shapes used to represent the segments include cylinders [24, 134], tapered cones [36, 162] and more complicated shapes such as superquadric ellipsoids [142] and generalized algebraic surfaces called 'metaballs' [110]. Such models use many more parameters than their 2D counterparts and can produce much more realistic renderings of the human body.

All these models of the human body are associated with a likelihood function which measures how well the image data is explained when the model is projected on the image. This can cue on different forms of image features: some authors match image edges with those of the human model contour predictions, *e.g.* [162, 36], while others use image texture under the model projection [24, 162, 29, 134, 142]. In the case of video sequences, optical flow information has also been exploited for this purpose [142]. In some cases, appearance models for different parts of the human body have been learned from image statistics and have been shown to perform better than hand-built appearance models [133]. Having defined a likelihood measure, an optimal configuration of the human body is estimated as the one that maximizes this measure. This estimation itself, however, is a nontrivial task as the parameter space involved is high-dimensional and the likelihood functions are often ill-behaved. Most top-down methods thus rely on some heuristic or prior knowledge based initialization of pose in the first image of a video sequence and make use of temporal continuity information for reducing the search space in consecutive frames, *e.g.* in a tracking framework. Several methods including Kalman filtering, Condensation and Hidden Markov Models have been used in the literature. Some of these are discussed in § 2.2. A few problem-specific optimization methods have also been developed for this task [142, 141], but other papers use expensive search methods such as Markov Chain Monte Carlo sampling *e.g.* [87].

One way to narrow down the search space is to use images from multiple calibrated cameras. Two calibrated images, in principle, contain complete 3D information and hence resolve the ambiguities associated with monocular image based reconstructions. While the exactly same image likelihood based approach described above can be applied for simultaneous optimization on images from multiple cameras, some papers adopt a different strategy and use multiple silhouette images of a person to reconstruct pose by fitting the body model to a 3D shape representation (visual hull) computed from the multiple views *e.g.* [30]. Such a method actually involves some elements of the bottom-up approach.

2.1.2 Bottom-up

As mentioned above, bottom-up methods for human pose estimation start with lower-level image features and use these to predict higher-level pose information in the form of a prespecified set of parameters. Unlike top-down methods that are very much based on a human body model, bottom-up methods may or may not involve the use of a body model. We call these two subclasses *model-based* and *model-free* bottom-up methods.

Model based

Model-based bottom-up methods use an explicit knowledge of body parts to predict the location of different limbs in an image, employing a *weak* body model to constrain these parts to valid configurations. The body models used here are weak in the sense that they are usually specified as a set of loose constraints or spatial priors between different body parts, and need not necessarily encode precise shapes of the body segments. So they can be quite different from the detailed body models of top-down methods. A number of commonly used models of this kind are based on what are known as 'pictorial structure' models [46, 43, 27] — graphs in which the body parts form nodes and part-interactions are represented using edges.

A recent and representative model of this kind is the 'loose-limbed' person [137] that uses relatively simple part detectors called 'shouters' to measure local evidence for body limbs and incorporates spatial priors to infer body pose using a non-parametric form of belief propagation [62]. In this work, the shouters use multi-scale edge and ridge filters as well as background subtraction information and their response is combined across multiple views. In other work, several other detectors have been developed for finding human body parts in images, *e.g.* Gaussian derivative filter responses and orientation features have been used to learn appearance models [120, 94]. Normally an exhaustive search is performed over image regions to classify each region as containing a valid limb or not. A number of labeled images are used to learn the classification rules (usually probabilistic) either in the form of occurrence statistics of the different features for positive and negative examples, or using discriminative learning methods such as the Support Vector Machine [159]. Some of the methods based on pictorial structure models actually use a bottom-up process only to obtain a first estimate of body part locations but make use of a generative model of the image appearance from the estimated limb locations to obtain complete pose. We can thus think of them as including a top-down inference step.

Explicit occlusion modeling of part appearances in pictorial structures is possible via the use of 'layered pictorial structures' [73]. This represents multiple layers in the image as probabilistic masks and appearance maps (referred to as 'sprites' [66]) and directly allows for handling articulated objects. In fact, layered pictorial structures have also been used to automatically learn object parts from video data by identifying rigidly moving regions in an image sequence [74]. This, in principle, allows a body model to be learned fully automatically — unlike the several existing papers that mostly use hand-built priors over part locations using the skeletal structure of the body, or otherwise prespecify a set of parts and then independently learn spatial (interaction) priors and appearance models for the different body parts.

Overall, model based methods appropriately account for the articulations and structure in the human body and allow for interesting inference methods. One of their main disadvantages, however, is the computational cost associated with their use. While top-down methods involve projecting precise body models and repeatedly computing image likelihoods for a large number of pose hypotheses, bottom-up methods are typically associated with costly inference algorithms over the body graph.

Model free

Model-free methods avoid the use of a human body model all together and directly infer pose parameters from image data. An algorithm based on such an approach has no notion of what the different pose parameters actually correspond to in a physical sense and as a result, there is very little scope for incorporating prior pose knowledge in this scenario. The pose vector is (mostly) simply treated as a point in some high-dimensional space. Current model-free methods rely on previously seen image-pose pairs to predict the pose of a new image and make use of large databases

of training data. These are usually obtained from motion capture or artificially synthesized using human model rendering packages such as Poser[1]. An advantage of using motion capture data is that it allows these methods to learn *typical* configurations of the human body. Methods that use such data can thus predict more natural poses as they explicitly or implicitly encode the statistics of observed human motion.

Most bottom-up model-free methods are example-based. They explicitly store a set of training examples whose 3D poses are known, and estimate pose by searching for training image(s) similar to the given input image and interpolating from their poses [14, 98, 132, 144, 117]. Searching for similar images requires effective matching amongst image observations, which in turn calls for suitable image representations. Human silhouettes are a good choice that have commonly been used in the literature *e.g.* [14, 98, 117]. Silhouettes have also been used for estimating 3D hand pose from images [144], where a tree based representation is used to facilitate efficient hierarchical search. As regards searching through large databases, an effective method based on using a parameter-sensitive hash function [132] has recently been proposed that uses a set of hashing functions to efficiently index examples, yielding approximate neighbours to a given example at a very low computational cost. This work makes use of multi-scale edge direction histograms over segmented images for representing the human appearance and thus captures more information that silhouette shape.

An alternative approach to searching through large training databases for estimating human pose is to directly learn a compactly representable mapping from image measurements to 3D pose. One of the first papers to employ such an approach models a dynamical manifold of human body configurations with a Hidden Markov Model and learns using entropy minimization [22]. Input images are again represented in the form of human silhouettes. Other papers that learn mappings between silhouettes and pose have made use of specialized maps [121] to learn many functions with different domains of applicability, and manifold embedding techniques to learn the mapping via an 'activity manifold' [40]. One of the contributions of this thesis (chapter 3) is in showing that a sparse nonlinear regression framework is a very effective alternative [1, 9]. Such a method makes use of kernel functions to implicitly encode locality and retain the advantage of example based methods, while at the same time obtains sparse solutions, ensuring that only a fraction of the training examples are actually used to compute pose. This allows the database of images to be pruned down to a bare minimum and avoids having to explicitly store the associated poses. The regression framework has also been generalized to infer pose as probabilistic multimodal distributions [6, 2] (described in detail in chapter 5) where the information contained monocular silhouettes is insufficient for precise inference, causing ambiguities in the solution. An interesting possibility with these methods that learn a mapping from image space to an underlying pose representation is that they can also make use of unlabeled data by using semisupervised learning methods. *E.g.* temporal correlation between frames has been exploited to learn a low dimensional manifolds of body pose [112]. Also, similar methods have been used to infer pose using multiple cameras. For instance, simultaneously observed silhouettes from multiple calibrated cameras have been used to infer body pose by fitting a mixture of probabilistic principal component analyzers to the density of multi-view shape and corresponding pose [52].

Within the framework of learning to predict 3D pose from image measurements, shape matching algorithms (*e.g.* [16, 51]) can also be used to first estimate the image locations of the centre of each body joint, and then recover 3D pose from them. This is not a strictly model-free approach, but nevertheless uses a set of training images with pre-labeled centres and avoids the construction of a detailed human body model. Many methods adopting this strategy have made use of a few hand-labeled key frames [145, 98] and others have used a training set of pose-center pairs obtained from resynthesized motion capture data [58]. Among the various possible methods that could be used for mapping a set of 2D points to a 3D geometry, human pose recovery has seen the use of

---

[1]Poser is a commercial software used for 3D human figure design, rendering and animation.

methods such as Bayesian learning [58] and recovering 3D via epipolar geometry of the camera [107]. These approaches show that using 2D joint centres as an intermediate representation can be an effective strategy. However, obtaining 3D body pose from 2D joint centres alone may be less robust and accurate than estimating pose directly from underlying image descriptors because it cannot cue on finer image details.

2.2   Tracking and Dynamics

The problem of pose estimation is very often addressed in the context of a sequence of images where temporal information is used to recover a consistent set of body poses over time. The most common framework used in this context is that of obtaining a *prediction* of the pose using previous estimates, followed by an *update* from the current observation.

Top down methods normally make use of their likelihood functions to update a pose hypothesis from the previous time step. The Kalman filter, extended Kalman filter and the particle filter (Condensation [63]) have widely been used for human tracking in a probabilistic framework by maintaining single or multiple hypotheses of body pose at each point of time [162, 36, 135]. Other trackers have made use of least-squares optimization [24, 110] and covariance scaled sampling [142] to compute the optimal body pose at each time step by explicitly maximizing the image likelihood.

Bottom-up methods, such as those which use tree or graph based representations of the human body usually rely on performing inference over multiple time steps [136, 61] or simply using appearance and spatial consistency across multiple frames to filter part proposals [113]. However, bottom-up tracking is also possible in the standard predict-and-update framework. In this thesis (chapter 4), we describe a discriminative tracking framework in which dynamical state predictions are fused directly with the image observations for estimating the body pose in a regression based setting [3]. This idea has also been extended to probabilistic tracking in which a discriminative state density is propagated in time, either by using multimodal pose estimates to weight particles sampled from a dynamical model as we describe in chapter 5 (*c.f.* [6]), or by analytically computing a mixture of Gaussians density by integration [138].

As regards modeling the dynamics of human motion, a variety of methods have been proposed. For regular motion such as walking, Hidden Markov Model based switching linear models [109] and the use of explicit knowledge of repeating 'cycles' [105] in human motion have been shown to be effective. Principal Component Analysis has been used in different ways to exploit the correlations in activities like walking and running motion [4, 157] and Gaussian processes [116] have been employed to model motion such as golf swing actions [156]. Another prediction approach is based on search *e.g.* plausible future states may be obtained by recovering training examples similar to the current state and looking up their successors [135]. This would, however, require a large amount of motion data. Motion capture is one common means of generating this. In fact, motion capture data has been used to model a variety of motion patterns in the vision-based human tracking literature, *e.g.* [23, 154, 9, 40]. In the graphics community, motion capture data is used very widely for motion synthesis — physically valid motion models have been constructed by incorporating physical constraints (*e.g.* ground contact) into an optimization approach [86, 124] and transitions between several motion segments have modeled for allowing interactive control of avatars in virtual environments [41].

2.3   Detecting People

Detecting people in images essentially means finding all instances of people in a given image. Depending on the approach adopted, the problem may be viewed as being totally opposite to that of pose estimation — an ideal human detector would detect people in all possible poses and thus being completely insensitive to pose information — or indeed very much the same because detecting a person in an image involves detecting his/her body segments and hence requires some sort of pose estimation. The difference in viewpoint depends on what resolution a person is detected at.

We first outline some approaches that detect people by detecting their parts. These are actually very similar in methodology to the model-based bottom-up pose estimation methods discussed in § 2.1.2. Separate detectors for different body parts are scanned over an image, generally at various scales and orientations, and their responses are post-processed with a spatial prior model *e.g.* [120, 94]. These part detectors normally use a variety of robust static image features, but motion and colour consistency information from several frames can also be exploited [113]. Enforcing temporal consistency among detected parts has actually been shown to be very effective and has recently also been used to detect people in unrestricted poses by starting with an easy-to-detect lateral walking pose — 'striking' a particular pose [114]. This allows the construction of a simple pose-specific detector (the particular one implemented being based on edges) followed by using a pictorial structure model to track the pose in successive video frames.

The other approach to human detection is to detect people without explicitly inferring part configurations, *e.g.* pedestrian detection [106, 96, 131, 161]. Here, the main challenge is handling the variability in pose and appearance of people, yet being discriminant from background. Several feature types are possible for this, *e.g.* wavelet based transforms have commonly been used [106, 131]. Recently a 'histogram of oriented gradients' [32] based representation has been shown to be very effective when used to learn a linear Support Vector Machine [159] based classifier. The framework has been used to detect people by scanning a detector window over the image at multiple scales. Although the classifier in this system learns to discriminate a person from background irrespective of pose, the dense orientation histogram representation itself is actually quite powerful even in capturing pose information. In chapter 6, it is shown that such features can successfully be used to regress 3D pose from images. *c.f.* [7].

Somewhere 'between' methods that detect pedestrians without pose information and those that look for precise pose are another class of methods which make use of weak shape priors or appearance exemplars for detection *e.g.* [152, 39]. These do not output body pose but nevertheless can often recognize the action or classify the kind of pose by comparison against a set of labeled images. A recent interesting approach that uses shape priors combines local and global cues to detect pedestrians via a probabilistic top-down segmentation [88]. Some other detection methods also give a detailed segmentation mask for images containing people at a reasonable resolution [100, 118]. These often make use of stick body models to exploit the articulated structure of the body during detection.

The human detection literature also has a large overlap with methods used in the field of object recognition. A major goal in both these areas is to develop robust and informative image representations that can discriminate people or objects from the background. We give a brief review of existing methods in this area in chapter 8.

3

Learning 3D Pose: Regression on Silhouettes

3.1 Introduction

This chapter describes a learning based method for recovering 3D human body pose from single images and monocular image sequences. Most existing methods in this domain (*example based methods*) explicitly store a set of training examples whose 3D poses are known, and estimate pose by searching for training image(s) similar to the given input image and interpolating from their poses [14, 98, 132, 144]. In contrast, the method developed here aims to learn a direct *mapping* from an image representation space to a human body 'pose space' and makes use of sparse nonlinear regression to distill a large training database into a single compact model that has good generalization to unseen examples. The regression framework optionally makes use of kernel functions to measure similarity between image pairs and implicitly encode locality. This allows the method to retain the advantage of example based methods. Despite the fact that full human pose recovery is very ill-conditioned and nonlinear, we find that the method obtains enough information for computing reasonably accurate pose information via regression.

Given the high dimensionality and intrinsic ambiguity of the monocular pose estimation problem, active selection of appropriate image features and good control of over-fitting is critical for success. We have chosen to base our system on taking image silhouettes as input, which we encode using robust silhouette shape descriptors. (Other image representations are discussed in chapters 6 and 8). To learn the mapping from silhouettes to human body pose, we take advantage of the sparsification and generalization properties of *Relevance Vector Machine (RVM)* [150] regression, allowing pose to be obtained from a new image using only a fraction of the training database. This avoids the need to store extremely large databases and allows for very fast pose estimation at run time.

3.1.1 Overview of the Approach

We represent 3D body pose by 55D vectors $\mathbf{x}$ including 3 joint angles for each of the 18 major body joints. Not all of these degrees of freedom are independent, but they correspond to the motion capture data that we use to train the system (see §3.3) and we retain this format so that our regression output is directly compatible with standard rendering packages for motion capture data. The input images are reduced to 100D observation vectors $\mathbf{z}$ that robustly encode the shape of a human image silhouette (§3.2). Given a set of labeled training examples $\{(\mathbf{z}_i, \mathbf{x}_i) \,|\, i = 1 \ldots n\}$,
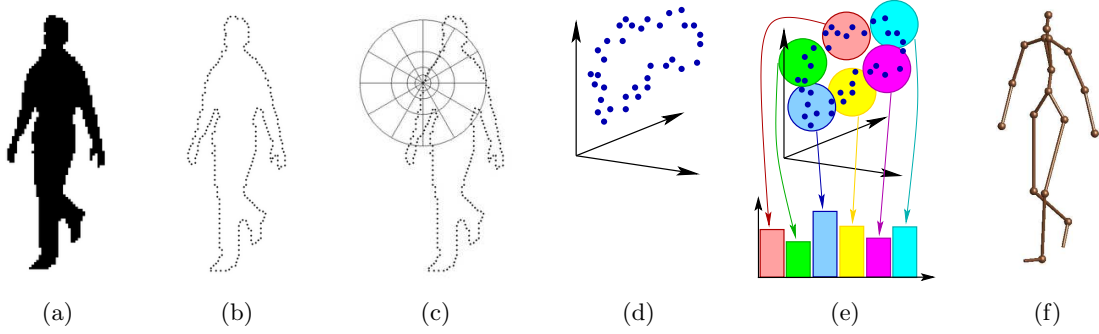
Figure 3.1: *A step by step illustration of our silhouette-to-pose regression method: (a) input silhouette extracted using background subtraction (b) sampled edge points (c) local shape contexts computed on edge points (d) distribution of these contexts in shape context space (e) soft vector quantization of the distribution to obtain a single histogram (f) 3D pose obtained by regressing on this histogram.*

the RVM learns a smooth reconstruction function $\mathbf{x} = \mathbf{r}(\mathbf{z}) = \sum_k \mathbf{a}_k \, \phi_k(\mathbf{z})$ that is valid over the region spanned by the training points. $\mathbf{r}(\mathbf{z})$ is a weighted linear combination of a prespecified set of scalar basis functions $\{\phi_k(\mathbf{z}) \,|\, k = 1 \ldots p\}$.

Our solutions are well-regularized in the sense that the weight vectors $\mathbf{a}_k$ are damped to control over-fitting, and sparse in the sense that many of them are zero. Sparsity occurs because the RVM actively selects only the 'most relevant' basis functions — the ones that really need to have nonzero coefficients to complete the regression successfully. For a linear basis ($\phi_k(\mathbf{z}) = k^{th}$ component of $\mathbf{z}$), the sparse solution obtained by the RVM allows the system to select relevant input *features* (components). For a kernel basis — $\phi_k(\mathbf{z}) \equiv K(\mathbf{z}, \mathbf{z}_k)$ for some kernel function $K(\mathbf{z}, \mathbf{z}')$ and centres $\mathbf{z}_k$ — relevant training *examples* are selected, allowing us to prune a large training dataset and retain only a minimal subset.

The complete process is illustrated in figure 3.1. We discuss our representations of the input and output spaces in §3.2 and §3.3; and the regression methods used in §3.4. The framework is applied to estimating pose from individual images in §3.5.

3.2   Image Descriptors

Directly regressing pose on input images requires a robust, compact and well-behaved representation of the observed image information. In this chapter, we use background subtraction to extract human silhouettes from an image and encode the observed image using robust descriptors of the shape of the subject's silhouette.

3.2.1   Silhouettes

Of the many different image descriptors that can be used for human pose estimation, image silhouettes are a popular choice and have often been used in the literature, *e.g.* [60, 13, 22, 40].

Silhouettes have three main advantages. *(i)* They can be extracted moderately reliably from images, at least when robust background- or motion-based segmentation is available and problems with shadows are avoided; *(ii)* they are insensitive to irrelevant surface attributes like clothing
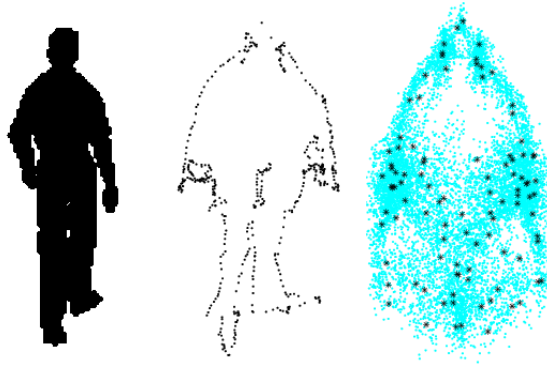
Figure 3.2*: Silhouette encoding using local shape context descriptors. Silhouette shape (left), is encoded as a fuzzy form in the shape context space (centre). The figure shows a projection on the first two principal components of the distribution of 60D shape context vectors computed on the edge points of this silhouette. (Right) The projection of all context vectors from a training data sequence. The average-over-human-silhouettes like form arises because (besides finer distinctions) the context vectors encode approximate spatial position on the silhouette: a context at the bottom left of the silhouette receives votes only in its upper right bins, etc. Also shown here are k-means centres that are used to vector quantize each silhouette's distribution into a single histogram.*

colour and texture; *(iii)* they encode a great deal of useful information about 3D pose without the need of any labeling information.

Two factors limit the performance attainable from silhouettes. *(i)* Artifacts such as shadow attachment and poor background segmentation tend to distort their local form. This often causes problems when global descriptors such as shape moments are used (as in [13, 22]), as every local error pollutes each component of the descriptor. To be robust, shape descriptors need to have good *locality*. *(ii)* Silhouettes leave several discrete and continuous degrees of freedom invisible or poorly visible. It is difficult to tell frontal views from back ones, whether a person seen from the side is stepping with the left leg or the right one, and what are the exact poses of arms or hands that fall within (are "occluded" by) the torso's silhouette. Including interior edge information within the silhouette [132] is likely to provide a useful degree of disambiguation in such cases, but is difficult to disambiguate from, *e.g.* markings on clothing.

3.2.2   Shape Context Distributions

To improve resistance to segmentation errors and occlusions, we need a robust silhouette representation[1]. The first requirement for robustness is *locality* — the presence of noise in some parts of a silhouette must not effect the entire representation. Histogramming edge information is a good way to encode local shape robustly [90, 16], so we begin by computing local edge histogram descriptors at regularly spaced points on the edge of the silhouette. About 400-500 points are used, which corresponds to a one pixel spacing on silhouettes of size 64×128 pixels such as those in our training set. We make of use shape context descriptors (histograms of edge pixels into log-polar bins [16]) to encode the local silhouette shape at a range of scales quasi-locally, over regions of

---

[1]We believe that any representation (Fourier coefficients, *etc.*) based on treating the silhouette shape as a continuous parametrized curve is inappropriate for this application: silhouettes frequently change topology (*e.g.* when a hand's silhouette touches the torso's one), so parametric curve-based encodings necessarily have discontinuities w.r.t. shape. *c.f.* [55] that explicitly deals with such discontinuities.
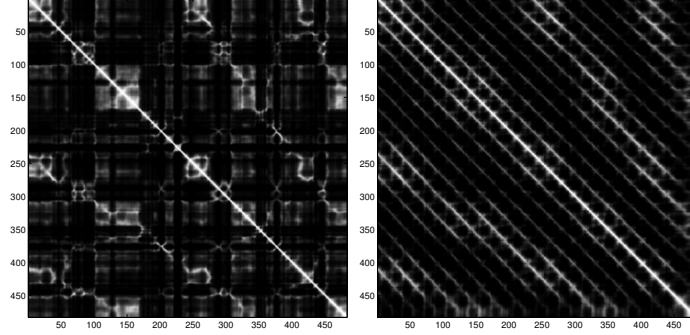
Figure 3.3: *Pairwise similarity matrices for (left) image silhouette descriptors and (right) true 3D poses, for a 483-frame sequence of a person walking in a decreasing spiral. The light off-diagonal bands that are visible in both matrices denote regions of comparative similarity linking corresponding poses on different cycles of the spiral. This indicates that our silhouette descriptors do indeed capture a significant amount of pose information. (The light anti-diagonal ripples in the 3D pose matrix arise because the standing-like poses at the middle of each stride have mid-range joint values, and hence are closer on average to other poses than the 'stepping' ones at the end of strides).*

diameter similar to the length of a limb[2]. The scale of the shape contexts is calculated as a function of the overall silhouette size, making the representation invariant to the overall scale of a silhouette. See figure 3.1(c). In our application we assume that the vertical is preserved, so to improve discrimination, we do not normalize contexts with respect to their dominant local orientations as originally proposed in [16]. Our shape contexts contain 12 angular × 5 radial bins, giving rise to 60 dimensional histograms. The silhouette shape is thus encoded as a 60D distribution (in fact, as a noisy multibranched curve, but we treat it as a distribution) in the shape context space.

Matching silhouettes is therefore reduced to matching distributions in shape context space. To implement this, a second level of histogramming is performed: we vector quantize the shape context space and use this to reduce the distribution of each silhouette to a 100D histogram (*c.f.* shapemes [97]). Silhouette comparison is thus finally reduced to a comparison of 100D histograms. The 100 centre codebook is learned once and for all by running k-means clustering on the combined set of context vectors of all of the training silhouettes. See figure 3.2. Other centre selection methods give similar results. For a given silhouette, a 100D histogram $\mathbf{z}$ is now built by allowing each of its 60D context vectors to vote softly into the few centre-classes nearest to it, and accumulating the scores of all of the silhouette's context vectors. The votes are computed by placing a Gaussian at each centre and computing the posterior probability for each shape context to belong to each centre/bin. We empirically set the common variance of the Gaussians such that each shape context has significant votes into 4-5 centres. This *soft* voting reduces the effects of spatial quantization, allowing us to compare histograms using simple Euclidean distance, rather than, say, Earth Movers Distance [123]. We also tested the Hellinger distance[3], with very similar results. The histogram-of-shape-contexts scheme gives us a reasonable degree of robustness to occlusions and local silhouette segmentation failures, and indeed captures a significant amount of pose information as shown in figure 3.3.

---

[2]Computing shape contexts descriptors with local support helps in attaining improved robustness to segmentation errors and local shape variations. This is different from the original use of shape contexts [98, 16] where they are globally computed over the entire shape.

[3]Hellinger distance, $H(\mathbf{z}_1, \mathbf{z}_2) \equiv \|\sqrt{\mathbf{z}_1} - \sqrt{\mathbf{z}_2}\|^2$ where $\|\,.\,\|$ is the L_2 norm.

3.3   Body Pose Representation

We recover 3D body pose (including orientation with respect to the camera) as a real 55D vector $\mathbf{x}$ that includes 3 joint angles for each of the 18 major body joints shown in figure 3.1(f). The subject's overall azimuth (compass heading angle) θ can wrap around through 360°. To maintain continuity, we actually regress $(a, b) = (\cos\theta, \sin\theta)$ rather than θ, using atan2(b, a) to recover θ from the not-necessarily-normalized vector returned by regression. So we have $3{\times}18{+}1 = 55$ parameters.

The regression framework developed here, however, is inherently model-free (*i.e.* it does not make use of any explicit body model) and is independent of the choice of this pose representation. The pose vector is simply treated as a point in some high dimensional space. Several different parametrizations of the human body are possible and the system can learn to predict pose in the form of any continuous set of parameters that the training data is specified in. The following chapters of this thesis, for example, will demonstrate results on recovering body pose in the form of a 44D vector of joint angles and a 24D vector of joint coordinates in 3D space. A discussion on various representations of human body pose is given in appendix B.

Also, since the algorithm has no explicit 'meaning' attached to the parameters that it learns to predict from silhouette data, we have not sought to learn a minimal representation of the true human pose degrees of freedom. We simply regress the training data its original motion capture format[4] — here in the form of Euler angles. Our regression method handles such redundant output representations without problems.

3.4   Regression Methods

This section describes the regression methods that have been evaluated for recovering 3D human body pose from the silhouette shape descriptors described in § 3.2. The output pose is written as a real vector $\mathbf{x} \in \mathbb{R}^m$ and the input shape as a descriptor vector $\mathbf{z} \in \mathbb{R}^d$.

Adopting a standard regression framework, $\mathbf{x}$ is expressed as a function of $\mathbf{z}$. Note that due to the ambiguities of pose recovery from monocular silhouettes (*i.e.* a given silhouette may actually be produced by more that one different underlying pose), the relationship between $\mathbf{z}$ and $\mathbf{x}$ may actually be non-functional. This issue, however, is postponed to chapters 4 and 5. For the moment, we assume that the relationship can be approximated functionally as a linear combination of a prespecified set of basis functions:

$$\mathbf{x} \;=\; \sum_{k=1}^{p} \mathbf{a}_k \, \phi_k(\mathbf{z}) + \boldsymbol{\epsilon} \;\equiv\; \mathbf{A}\,\mathbf{f}(\mathbf{z}) + \boldsymbol{\epsilon} \tag{3.1}$$

Here, $\{\phi_k(\mathbf{z}) \,|\, k = 1 \ldots p\}$ are the basis functions, $\mathbf{a}_k$ are $\mathbb{R}^m$-valued weight vectors, and $\boldsymbol{\epsilon}$ is a residual error vector. For compactness, we gather the weight vectors into an $m{\times}p$ weight matrix $\mathbf{A} \equiv (\mathbf{a}_1 \;\; \mathbf{a}_2 \;\; \cdots \;\; \mathbf{a}_p)$ and the basis functions into a $\mathbb{R}^p$-valued function $\mathbf{f}(\mathbf{z}) = (\phi_1(\mathbf{z}) \;\;\; \phi_2(\mathbf{z}) \;\; \cdots \;\; \phi_p(\mathbf{z}))^\top$. To allow for a constant offset $\mathbf{x} = \mathbf{A}\mathbf{f} + \mathbf{b}$, we can include $\phi(\mathbf{z}) \equiv 1$ in $\mathbf{f}$.

To train the model (estimate $\mathbf{A}$), we are given a set of training pairs $\{(\mathbf{x}_i, \mathbf{z}_i) \,|\, i = 1 \ldots n\}$. We use the Euclidean norm to measure $\mathbf{x}$-space prediction errors, so the estimation problem is of the following form:

---

[4]The motion capture data used in this chapter is in the 'BioVision Hierarchy' format and was taken from the public website www.ict.usc.edu/graphics/animWeb/humanoid.

$$\mathbf{A} := \arg\min_{\mathbf{A}} \left\{ \sum_{i=1}^{n} \|\mathbf{A}\,\mathbf{f}(\mathbf{z}_i) - \mathbf{x}_i\|^2 + R(\mathbf{A}) \right\} \tag{3.2}$$

where $R(-)$ is a regularizer on $\mathbf{A}$ that prevents overfitting. Gathering the training points into an $m \times n$ output matrix $\mathbf{X} \equiv (\mathbf{x}_1 \ \ \mathbf{x}_2 \ \cdots \ \mathbf{x}_n)$ and a $p \times n$ feature matrix $\mathbf{F} \equiv (\mathbf{f}(\mathbf{z}_1) \ \ \mathbf{f}(\mathbf{z}_2) \ \cdots \ \mathbf{f}(\mathbf{z}_n))$, the estimation problem takes the form:

$$\mathbf{A} := \arg\min_{\mathbf{A}} \left\{ \|\mathbf{A}\,\mathbf{F} - \mathbf{X}\|^2 + R(\mathbf{A}) \right\} \tag{3.3}$$

where $\| \, . \, \|$ denotes the Frobenius norm. Note that the dependence on $\{\phi_k(-)\}$ and $\{\mathbf{z}_i\}$ is encoded entirely in the numerical matrix $\mathbf{F}$.

3.4.1 Ridge Regression

Pose estimation is a high dimensional and intrinsically ill-conditioned problem, so simple least squares estimation — setting $R(\mathbf{A}) \equiv \mathbf{0}$ and solving for $\mathbf{A}$ in least squares — typically produces severe overfitting and hence poor generalization. To reduce this, we need to add a smoothness constraint on the learned mapping, for example by including a damping or regularization term $R(\mathbf{A})$ that penalizes large values in the coefficient matrix $\mathbf{A}$. Consider the simplest choice, $R(\mathbf{A}) \equiv \lambda \|\mathbf{A}\|^2$, where λ is a regularization parameter. This gives the *damped least squares* or *ridge* regressor which minimizes

$$\|\mathbf{A}\,\tilde{\mathbf{F}} - \tilde{\mathbf{X}}\|^2 := \|\mathbf{A}\,\mathbf{F} - \mathbf{X}\|^2 + \lambda \|\mathbf{A}\|^2 \tag{3.4}$$

where $\tilde{\mathbf{F}} \equiv (\mathbf{F} \ \ \lambda\,\mathbf{I})$ and $\tilde{\mathbf{X}} \equiv (\mathbf{X} \ \ \mathbf{0})$. The solution can be obtained by solving the linear system $\mathbf{A}\,\tilde{\mathbf{F}} = \tilde{\mathbf{X}}$ (*i.e.* $\tilde{\mathbf{F}}^\top \mathbf{A}^\top = \tilde{\mathbf{X}}^\top$) for $\mathbf{A}$ in least squares[5], using QR decomposition or the normal equations. Ridge solutions are not equivariant under relative scaling of input dimensions, so we usually scale the inputs to have unit variance before solving. λ must be set large enough to control ill-conditioning and overfitting, but not so large as to cause overdamping (forcing $\mathbf{A}$ towards $\mathbf{0}$ so that the regressor systematically underestimates the solution). In practice, a suitable value of λ is usually determined by cross validation.

3.4.2 Relevance Vector Regression

Relevance Vector Machines (RVMs) [150, 151] are a sparse Bayesian approach to classification and regression. They introduce Gaussian priors on each parameter or group of parameters, each prior being controlled by its own individual scale hyperparameter, and perform inference by integrating over the set of parameters $\mathbf{A}$. Here we keep to the estimation form of (3.3) and adopt an alternate (MAP) approach.

Integrating out the hyperpriors (which can be done analytically) gives singular, highly nonconvex total priors of the form $p(a) \sim \|a\|^{-\nu}$ for each parameter or parameter group a, where ν is a

---

[5]If a constant offset $\mathbf{x} = \mathbf{A}\mathbf{f} + \mathbf{b}$ is included, $\mathbf{b}$ must not be damped, so the system takes the form $(\mathbf{A} \ \ \mathbf{b})\,\tilde{\mathbf{F}} = \tilde{\mathbf{X}}$ where $\tilde{\mathbf{F}} \equiv \begin{pmatrix} \mathbf{F} & \lambda\,\mathbf{I} \\ \mathbf{1} & \mathbf{0} \end{pmatrix}$ and $\tilde{\mathbf{X}} \equiv (\mathbf{X} \ \ \mathbf{0})$.

(a)   (b)
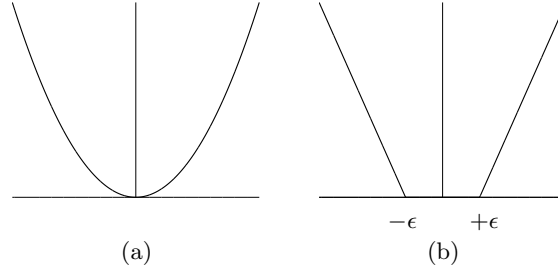
Figure 3.4: *(a) The quadratic loss function used by ridge regression and our RVM algorithm, and (b) the ϵ-insensitive linear loss function used by the SVM.*

hyperprior parameter. Taking log likelihoods gives an equivalent regularization penalty of the form $R(a) = \nu \log \|a\|$. Such a logarithmic form associates very high penalties with small values of a and has an effect of pushing unnecessary parameters to zero. The model produced is thus sparse and the RVM automatically selects the most 'relevant' basis functions to describe the problem.

To solve for the complete matrix $\mathbf{A}$, we minimize the functional form given by

$$\|\mathbf{A}\,\mathbf{F} - \mathbf{X}\|^2 + \nu \sum_k \log \|\mathbf{a}_k\| \tag{3.5}$$

where $\mathbf{a}_k$ are the columns of $\mathbf{A}$ and ν is a regularization parameter that also controls sparsity. The minimization algorithm that we use for this is based on successively approximating the logarithmic term with quadratics, in effect solving a series of linear systems. The details of the algorithm and a discussion on its sparseness properties is are given in appendix A. This is different from the original algorithm proposed in [151] and was not developled as a part of the work done in this thesis.

3.4.3   Support Vector Regression

A third method for regularized regression that we have tested uses the Support Vector Machine (SVM) [159], which is well known for its use in maximum-margin based classification.

The goal in support vector regression is to find a function that has at most ϵ deviation from the actually obtained targets x_i for all the training data, and at the same time, is as flat as possible. In its standard formulation, the SVM assumes scalar outputs and hence works on individual components x of the complete vector $\mathbf{x}$. As in ridge regression, flatness is ensured by minimizing the Euclidean norm of the weight matrix, but now separately for each *row* $\mathbf{a}$ of $\mathbf{A}$. Since the existence of an ϵ-precision function is not guaranteed and some errors must be allowed, extra *slack* variables are introduced to 'soften' the constraints. The final formulation, as stated in [159], takes the following form for each output component x:

$$
\begin{aligned}
&\text{minimize} \qquad \frac{1}{2}\|\mathbf{a}\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\
&\text{subject to} \qquad
\begin{cases}
x_i - \mathbf{a}\,\mathbf{f}(\mathbf{z}_i) & \leq \ \epsilon + \xi_i \\
\mathbf{a}\,\mathbf{f}(\mathbf{z}_i) - x_i & \leq \ \epsilon + \xi_i^* \\
\xi_i, \xi_i^* & \geq \ 0
\end{cases}
\end{aligned}
\tag{3.6}
$$

where $\mathbf{a}$ is the row corresponding to the scalar component x, ξ_i, ξ_i^* are the slack variables and the constant $C > 0$ determines the trade off between the function flatness and the amount up to which deviations larger than ϵ are tolerated. This formulation corresponds to dealing with a so called ϵ-insensitive loss function $|\xi|_\epsilon$ described by

$$|\xi|_\epsilon := \begin{cases} 0 & \text{if } |\xi| \leq \epsilon \\ |\xi| - \epsilon & \text{otherwise} \end{cases} \tag{3.7}$$

Figure 3.4 illustrates this function in comparison to the quadratic loss functions used by ridge regression and our approximated RVM algorithm. While points with a deviation in prediction less that ϵ do not contribute to the cost, deviations greater than ϵ are penalized in a linear fashion. This gives the SVM a greater degree of robustness to outliers than ridge regression and the RVM. The optimization problem (3.6) is mostly solved in its dual form. We make use of the standard algorithm, details of which are available in [143].

3.5 Estimating Pose from Static Images

For our experiments, we made use of a database of motion capture data for a 54 d.o.f. body model, represented using joint angles as described in § 3.3. The data was divided into a training set of several motion sequences ($\sim$2600 poses in all) that include various kinds of walking motion viewed from different directions, and a test set of $\sim$400 frame sequence of a person walking in a decreasing spiral. The silhouettes corresponding to all these poses were synthesized using a graphics package (Poser from Curious Labs).

The error between a pair of true and estimated joint angle vectors (in degrees) is measured as the mean over all 54 angles of the absolute differences:

$$D(\mathbf{x}, \mathbf{x}') = \frac{1}{m} \sum_{j=1}^{m} \left| (x_j - x_j') \bmod \pm 180° \right| \tag{3.8}$$

where $x \bmod \pm 180° \equiv (x + 180°) \bmod 360° - 180°$ reduces angles to the interval $[-180°, +180°]$. For the entire test set, an average error is computed as the mean over all angles of the RMS absolute differences.

3.5.1 Choice of Basis & Implicit Feature Selection

Recall that the form of the basis functions in (3.1) must be prespecified. We tested two kinds of regression bases $\mathbf{f}(\mathbf{z})$:

- *Linear bases*, $\mathbf{f}(\mathbf{z}) \equiv \mathbf{z}$, simply return the input vector, so the regressor is linear in $\mathbf{z}$ and the RVM selects relevant *features* (components of $\mathbf{z}$).

- *Kernel bases*, $\mathbf{f}(\mathbf{z}) = (K(\mathbf{z}, \mathbf{z}_1) \ \cdots \ K(\mathbf{z}, \mathbf{z}_n))^\top$, are based on a kernel function $K(\mathbf{z}, \mathbf{z}_i)$ instantiated at training examples $\mathbf{z}_i$, so the RVM effectively selects relevant *examples*.
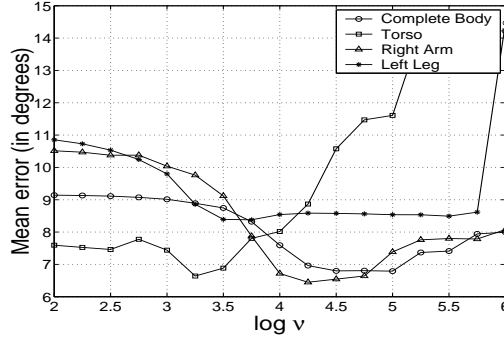
Figure 3.5: *The mean test-set fitting error for various body parts, versus the linear RVM sparseness parameter ν. The minima indicate the optimal sparsity / regularization settings for each part. Limb regressors are sparser than body or torso ones. The whole body regressor retains 23 features; the torso, 31; the right arm, 10; and the left leg, 7.*
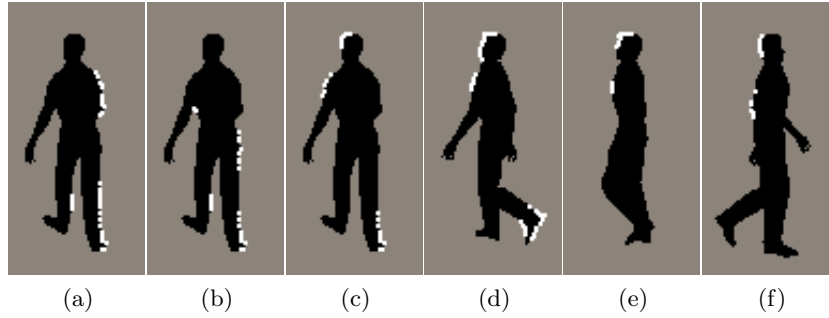
Figure 3.6: *Feature selection on silhouettes: the points whose shape context classes are retained by the RVM for regression on (a) left arm angles, (b) right leg angles, shown on a sample silhouette. (c-f): Silhouette points encoding torso & neck parameter values over different view points and poses. On average, about 10 features covering about 10% of the silhouette suffice to estimate the pose of each body part.*

For the experiments, we use a Gaussian kernel $K(\mathbf{z}, \mathbf{z}_i) = e^{-\beta \|\mathbf{z} - \mathbf{z}_i\|^2}$ with β estimated from the scatter matrix of the training data, but the form and parameters of the kernel have remarkably little influence. Other β values within a factor of 2 from this value give very similar results. Our experiments show that linear bases on our already highly non linear features work well, but that kernelization gives a small but useful improvement — about 0.8° per body angle, out of a total mean error of around 7°.

Linear RVM regression directly selects relevant components of $\mathbf{z}$. This is like an implicit feature selection as it reveals which of the original input features encode useful pose information. One might expect that, *e.g.* the pose of the arms was mainly encoded by (shape-context classes receiving contributions from) features on the arms, and so forth, so that the arms could be regressed from fewer features than the whole body, and could be regressed robustly even if the legs were occluded. To test this, we divided the body joints into five subsets — torso & neck, the two arms, and the two legs — and trained separate linear RVM regressors for each subset. Figure 3.5 shows that similar validation-set errors are attained for each part, but the optimal regularization level is significantly

smaller (there is less sparsity) for the torso than for the other parts. Figure 3.6 shows the silhouette points whose contexts contribute to the features (histogram classes) that were selected as relevant, for several parts and poses. The main observations are that the regressors are indeed sparse — only about 10% of the histogram bins were classed as relevant on average, and the points contributing to these tend to be well localized in important-looking regions of the silhouette — but that there is a good deal of non-locality between the points selected for making observations and the parts of the body being estimated. This nonlocality is interesting and is perhaps due to the extent to which the motions of different body segments are synchronized during natural walking motion. This suggests that — at least for such motion — the localized calculations of model-based pose recovery may actually miss a good deal of the information most relevant for pose.

3.5.2   Performance Analysis

In this section, we first compare the results of regressing body pose $\mathbf{x}$ (in the 55D representation of §3.3) against silhouette descriptors $\mathbf{z}$ (the 100D histograms of §3.2) using different regression methods. Figure 3.7 summarizes the test-set performance of the various regression methods studied — kernelized and linear basis versions of damped least squares regression (LSR), RVM and SVM regression, for the full body model and various subsets of it — at optimal regularizer settings computed using two-fold cross validation. All output parameters are normalized to have unit variance before regression and the tube width ϵ in the SVM is set to correspond to an error of $1°$ for each joint angle (the SVM results shown here use SVM-*Light* [65]). Kernelization brings a small advantage ($0.8°$ on an average) over purely linear regression against our (highly nonlinear) descriptor set. The regressors are all found to give their best results at similar optimal kernel parameters, which are more or less independent of the regularization prior strengths. Overall, the results obtained from the three different regressors are quite similar and this confirms that our representation and framework are independent of the exact method of regression used. The best performance in terms of reconstruction error is achieved by the SVM and we attribute this to the different form of its loss function. (This could be verified by trying to design an ϵ-insensitive loss RVM.) The RVM regression, on the other hand, gives very slightly higher errors than the other two regressors, but much more sparsity. For example, in the whole-body regression using kernel bases, the RVM selects just 156 (about 6%) of the 2636 training points as basis kernels, giving a mean test-set error of only $6.0°$. This ability to achieve very sparse solutions without compromising much on accuracy allows the RVM to be employed for extremely fast regression and makes it usable in a real time pose tracking system.

Figure 3.8 shows some sample pose estimation results using the RVM, on silhouettes from a spiral-walking motion capture sequence that was not included in the training set. The mean estimation error over all joints for Gaussian kernel bases in this test is $6.0°$. The RMS errors for individual body angles depends on the observability and on the ranges of variation of these angles and can vary quite a lot from angle to angle. For example, the errors (with the ranges in variation in parentheses) for some of the joints are as follows: body heading angle, $17°$ ($360°$); left shoulder angle, $7.5°$ ($51°$); and right hip angle, $4.2°$($47°$).

Figure 3.9 (top) plots the estimated and actual values of the overall body heading angle θ during the test sequence. Much of the error is seen in the form of occasional large errors. We refer to these as "glitches". They are associated with poses where the silhouette is ambiguous and might easily arise from any of several possible poses. As one diagnostic for this, recall that to allow for the $360°$ wrap around of the heading angle θ, we actually regress $(a, b) = (\cos\theta, \sin\theta)$ rather than θ. In ambiguous cases, the regressor tends to compromise between several possible solutions, and hence returns an (a, b) vector whose norm is significantly less than one. These events are strongly correlated with large estimation errors in θ, as illustrated in figure 3.9 (middle and bottom).

	LSR	RVM	SVM
Average error (in degrees)	5.95	6.02	5.91
% of support vectors retained	100	6	53
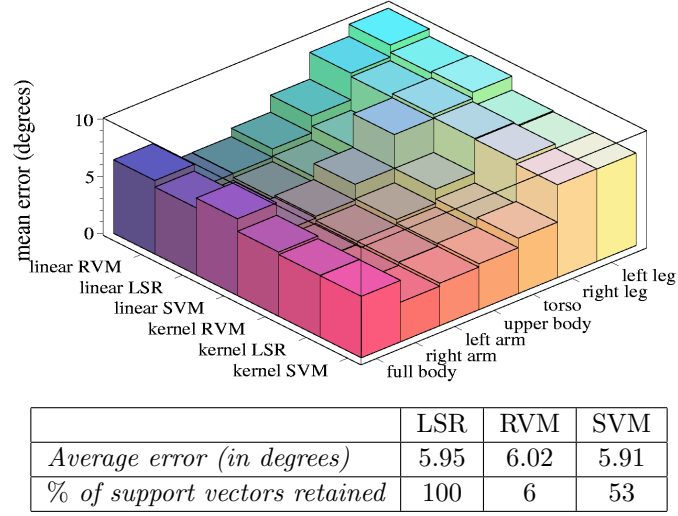
Figure 3.7: *(Top) A summary of the various regressors' performance on different combinations of body parts for the spiral walking test sequence. (Bottom) Error measures for the full body using Gaussian kernel bases with the corresponding number of support vectors retained.*

(a)   (b)   (c)   (d)   (e)   (f)
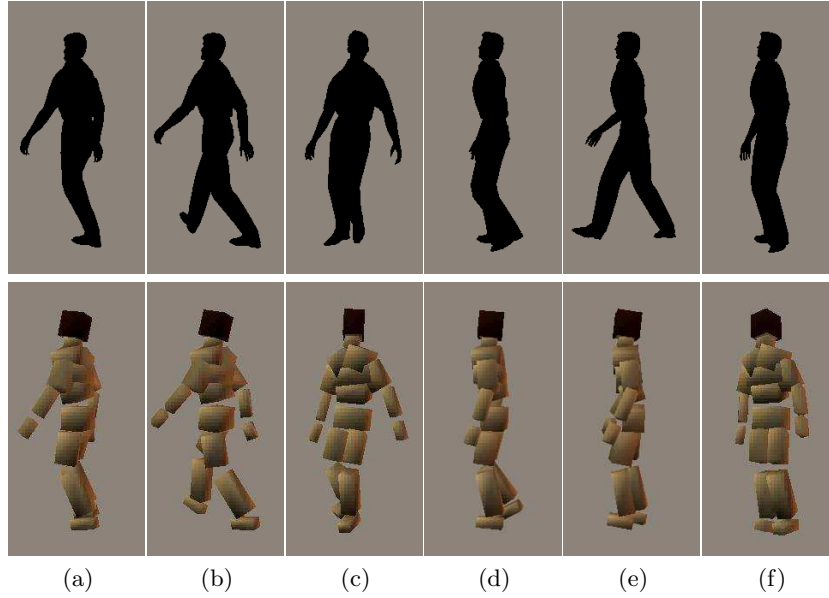
Figure 3.8: *Some sample pose reconstructions for a spiral walking sequence not included in the training data. The reconstructions were computed with a Gaussian kernel RVM, using only 156 of the 2636 training examples. The mean angular error per d.o.f. over the whole sequence is 6.0°. While (a-c) show accurate reconstructions, (d-f) are examples of misestimation: (d) illustrates a label confusion (the left and right legs have been interchanged), (e,f) are examples of compromised solutions where the regressor has averaged between two or more distinct possibilities. Using single images alone, we find ∼ 15% of our results are misestimated.*
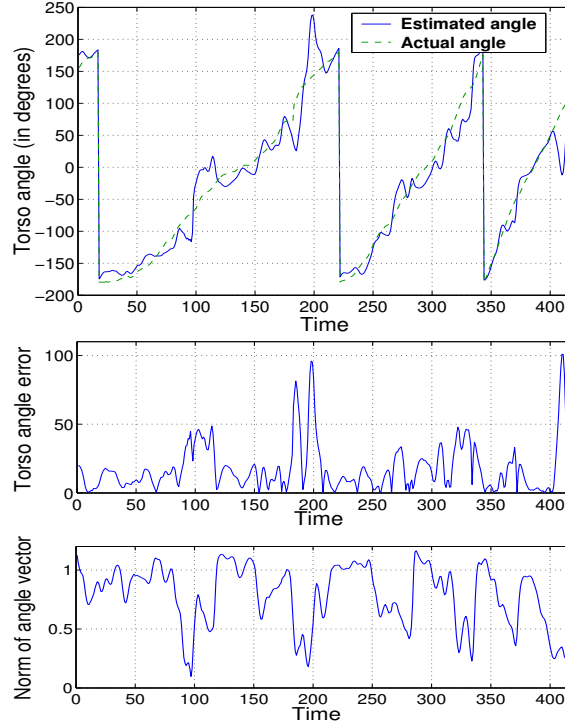
Figure 3.9: *(Top): The estimated body heading angle(azimuth θ) over 418 frames of the spiral walking test sequence, compared with its actual value from motion capture. (Middle, Bottom): Episodes of high estimation error are strongly correlated with periods when the norm of the $(\cos\theta, \sin\theta)$ vector that was regressed to estimate θ becomes small. These occur when similar silhouettes arise from very different poses, so that the regressor is forced into outputting a compromise solution.*

Figure 3.10 shows reconstruction results on some real images[6]. The silhouettes for this experiment are extracted by a simple background subtraction method that uses a probabilistic model of the (static) background and are relatively noisy, but this demonstrates the method's robustness to imperfect visual features. (The complete background subtraction method is described in appendix C.) The last example in the figure illustrates the problem of silhouette ambiguity: the method returns a pose with the left knee bent instead of the right one because the silhouette looks the same in the two cases, causing a glitch in the output pose.

3.6   Discussion

In this chapter, we have introduced a regression based method that recovers 3D human body pose from monocular silhouettes using robust histogram-of-shape-context silhouette shape descriptors. The advantages of the approach are that it requires no 3D body model, no labeling of image positions of body parts and no image likelihood computations. This makes the method easily adaptable to different people or appearances.

---

[6]These images are part of a sequence from www.nada.kth.se/~hedvig/data.html
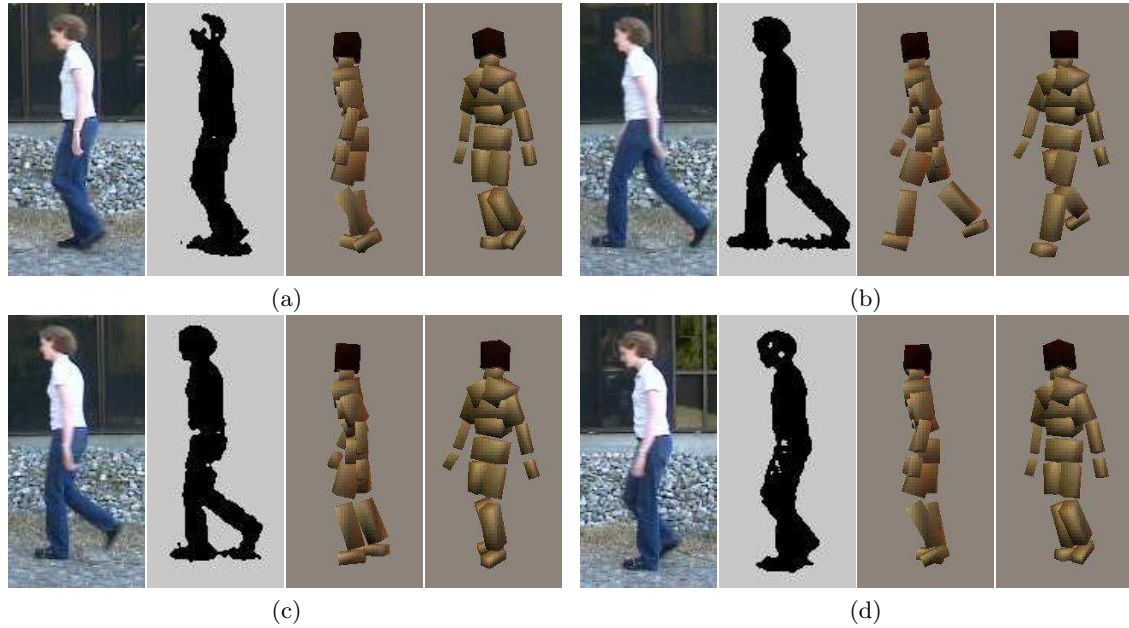
Figure 3.10: *3D poses reconstructed from some real test images using a single image for each reconstruction. The first and second reconstruction images in each set show the estimates from the original viewpoint and from a new one respectively. (a,b) show examples of accurate reconstructions. In (c), a noisy silhouette causes mis-estimation of the lower right leg, while the example in (d) demonstrates a case of left-right ambiguity in the silhouette.*

Amongst the various regression methods tested, we find that the Relevance Vector Machine gives very sparse solutions and allows accurate reconstruction of 3D pose using only a fraction of the training database. This makes the method capable of running in real time[7].

The silhouette descriptors developed here are shown to be quite robust in that they match real silhouettes with the artificially synthesized ones quite well. The representation as it is, however, does not currently support occlusions. We discuss some thoughts on this in chapter 9.

From both the experiments in this chapter, we see that regressing pose on single image silhouettes using the current framework gives very reasonable results in most cases but there are occasional glitches. These glitches occur when more than one solution is possible, causing the regressor to either 'select' the wrong solution, or to output a compromise between two different solutions. One possible way to reduce such errors would be to incorporate stronger features such as internal body edges within the silhouette, however the problem is bound to persist as important internal edges are often invisible and useful ones have to be distinguished from irrelevant clothing texture. Furthermore, even without these limb labeling ambiguities, depth related ambiguities (caused due to the loss of depth information in 2D images) exist and remain an issue. By keying on experimentally observed poses, our single image method already reduces this ambiguity significantly, but the subtle cues that human beings rely on to disambiguate multiple solutions remain inaccessible. An important source of information in this regard, which has not been exploited in this chapter, is temporal continuity. The next two chapters discuss two different extensions to the regression model described here, both of which make use of temporal information to reduce this ambiguity.

---

[7]In our existing implementation, images are pre-processed offline as the background subtraction and shape context routines are in matlab. Obtaining pose from image descriptors is a linear operation that runs at more than 30 fps.

4

Tracking and Regression

4.1 Introduction

Regressing human body pose on image descriptors is a powerful method for markerless motion capture. In chapter 3, we have seen that regression can be used to obtain reasonable pose estimates from single image silhouettes. However, it produces occasional errors/glitches when the information contained in a single silhouette is not sufficient to infer a unique or precise 3D pose from it. In this chapter, we consider the case when images are available in the form of a continuous video stream. We incorporate temporal constraints by modeling the dynamics of human motion and develop a discriminative tracking framework that works in a completely bottom-up manner, reconstructing the most likely 3D pose at each time step by fusing pose predictions from a learned dynamical model into our single-image regression framework.

The 3D pose can only be observed indirectly via ambiguous and noisy image measurements, so we start by considering the Bayesian tracking framework which represents our knowledge about the state (pose) $\mathbf{x}_t$ given the observations up to time t as a probability distribution, the posterior state density $p(\mathbf{x}_t \mid \mathbf{z}_t, \mathbf{z}_{t-1}, \ldots, \mathbf{z}_0)$[1]. Given an image observation $\mathbf{z}_t$ and a prior $p(\mathbf{x}_t)$ on the corresponding pose $\mathbf{x}_t$, the posterior likelihood for $\mathbf{x}_t$ is usually evaluated using Bayes' rule, $p(\mathbf{x}_t|\mathbf{z}_t) \propto p(\mathbf{z}_t|\mathbf{x}_t)\, p(\mathbf{x}_t)$, where $p(\mathbf{z}_t|\mathbf{x}_t)$ is an explicit 'generative' observation model that predicts $\mathbf{z}_t$ and its uncertainty given $\mathbf{x}_t$. Unfortunately, when tracking objects as complicated as the human body, the observations depend on a great many factors that are difficult to control, ranging from lighting and background to body shape, clothing style and texture, so any hand-built observation model is necessarily a gross oversimplification. One way around this would be to learn the generative model $p(\mathbf{z}|\mathbf{x})$ from examples, then to work backwards via its Jacobian to get a linearized state update, as in the extended Kalman filter. However, this approach is somewhat indirect, and it may waste a considerable amount of effort modeling appearance details that are irrelevant for predicting pose. Keeping in line with our choice of learning a 'diagnostic' regressor $\mathbf{x} = \mathbf{x}(\mathbf{z})$ rather than a generative predictor $\mathbf{z} = \mathbf{z}(\mathbf{x})$ for pose reconstruction, we prefer to learn a diagnostic model $p(\mathbf{x}|\mathbf{z})$ for the pose $\mathbf{x}$ given the observations $\mathbf{z}$. (*c.f.* the difference between generative and discriminative classifiers, and the regression based trackers of [68, 163].) However, as we have seen in the previous chapter, image projection suppresses most of the depth (camera-object distance) information and using silhouettes as image observations induces further ambiguities owing to the lack of limb labeling (see figure 4.1). So the state-to-observation mapping is always many-to-one. These ambiguities make learning to regress $\mathbf{x}$ from $\mathbf{z}$ difficult because the true mapping is actually

---

[1]This is different from the case of *filtering* in which all observations from the complete video stream are assumed to be accessible at any given time instant. Keeping in mind a real time tracking application, it is assumed here that only the image features $\{\mathbf{z}_t, \mathbf{z}_{t-1}, \ldots, \mathbf{z}_0\}$ are accessible at time t and no attempt is made to model any dependence on future observations.

Figure 4.1: *Examples of ambiguities in silhouettes: different 3D poses can have very similar image observations, causing the regression from image silhouettes to 3D pose to be inherently multi-valued. The legs and the arms are reversed in the first two images, for example.*

multi-valued. A single-valued regressor tends to either zig-zag erratically between different training poses, or (if highly damped) to reproduce their arithmetic mean [18], neither of which is desirable.

One approach to this is to learn a multivalued representation — a method of this type is discussed in chapter 5. In this chapter, we reduce the ambiguity by working incrementally from the previous few states[2] $(\mathbf{x}_{t-1}, \ldots)$. *c.f.* [38]. We adopt the working hypothesis that given a dynamics-based estimate $\mathbf{x}_t(\mathbf{x}_{t-1}, \ldots)$ — or any other rough initial estimate $\check{\mathbf{x}}_t$ for $\mathbf{x}_t$ — it will usually be the case that only one of the possible observation-based estimates $\mathbf{x}(\mathbf{z})$ lies near $\check{\mathbf{x}}$. Thus, we can use the $\check{\mathbf{x}}_t$ value to "select the correct solution" for the observation-based reconstruction $\mathbf{x}_t(\mathbf{z}_t)$. Formally this gives a regressor $\mathbf{x}_t = \mathbf{x}_t(\mathbf{z}_t, \check{\mathbf{x}}_t)$, where $\check{\mathbf{x}}_t$ serves mainly as a key to select which branch of the pose-from-observation space to use, not as a useful prediction of $\mathbf{x}_t$ in its own right. To work like this, the regressor must be local and hence nonlinear in $\check{\mathbf{x}}_t$. Taking this one step further, if $\check{\mathbf{x}}_t$ is actually a useful estimate of $\mathbf{x}_t$ (*e.g.* from a dynamical model), we can use a single regressor of the same form, $\mathbf{x}_t = \mathbf{x}_t(\mathbf{z}_t, \check{\mathbf{x}}_t)$, but now with a stronger dependence on $\check{\mathbf{x}}_t$, to capture the net effect of implicitly reconstructing an observation-estimate $\mathbf{x}_t(\mathbf{z}_t)$ and then fusing it with $\check{\mathbf{x}}_t$ to get a better estimate of $\mathbf{x}_t$.

4.2   Learning the Regression Models

The regression framework from the previous chapter that estimates 3D pose from silhouette shape descriptors is now extended to include a dynamical (autoregressive) model for handling ambiguities and producing smooth reconstructions over a stream of images. The new model, which we call *discriminative tracking* because it is fully conditional and avoids the construction of a generative model of image likelihoods, now involves two levels of regression — a dynamical model and an observation model.

---

[2]The ambiguities persist for several frames so regressing the pose $\mathbf{x}_t$ against a sequence of the last few silhouettes $(\mathbf{z}_t, \mathbf{z}_{t-1}, \ldots)$ does not suffice.

4.2.1 Dynamical (Prediction) Model

Dynamical models, as the name suggests, are used to model the dynamics of a system. They are widely used in a variety of domains involving time-varying systems [83, 103] and several forms of such models have been proposed in the human tracking literature, *e.g.* [135, 108]. In this work, we model human body dynamics with a second order linear autoregressive process which assumes that the current state (pose in this case) can be expressed as a linear function of the states from the two previous time steps[3]. The state at time t is modeled as $\mathbf{x}_t = \check{\mathbf{x}}_t + \boldsymbol{\epsilon}$, where $\check{\mathbf{x}}_t \equiv \tilde{\mathbf{A}}\,\mathbf{x}_{t-1} + \tilde{\mathbf{B}}\,\mathbf{x}_{t-2}$ is the second order dynamical estimate of $\mathbf{x}_t$ and $\boldsymbol{\epsilon}$ is a residual error vector. Learning the regression directly in this form with regularization on the parameters $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ (see § 3.4), however, forces $\check{\mathbf{x}}_t$ to $\mathbf{0}$ in the case of overdamping. To avoid such a situation, the solution can be forced to converge to a default linear prediction if the parameters are overdamped by learning the autoregression for $\check{\mathbf{x}}_t$ in the following form:

$$\check{\mathbf{x}}_t \; \equiv \; (\mathbf{I} + \mathbf{A})(2\mathbf{x}_{t-1} - \mathbf{x}_{t-2}) + \mathbf{B}\,\mathbf{x}_{t-1} \tag{4.1}$$

where $\mathbf{I}$ is the $m \times m$ identity matrix. $\mathbf{A}$ and $\mathbf{B}$ are estimated by regularized least squares regression against $\mathbf{x}_t$, minimizing $\|\boldsymbol{\epsilon}\|^2 + \lambda(\|\mathbf{A}\|^2_{\text{Frob}} + \|\mathbf{B}\|^2_{\text{Frob}})$ over the training set as described in section 3.4.1 and the regularization parameter λ is set by cross-validation to give a well-damped solution with good generalization.

4.2.2 Observation (Correction) Model

Now consider the observation model. As discussed in § 4.1, the underlying density $p(\mathbf{x}_t|\mathbf{z}_t)$ is highly multimodal owing to the pervasive ambiguities in reconstructing 3D pose from monocular images, so no single-valued regression function $\mathbf{x}_t = \mathbf{x}_t(\mathbf{z}_t)$ can give completely acceptable point estimates for $\mathbf{x}_t$. However, much of the 'glitchiness' and jitter observed in the static reconstructions can be removed by feeding $\check{\mathbf{x}}_t$ along with $\mathbf{z}_t$ into the regression model.

A combined regressor $\mathbf{x}_t = \mathbf{x}_t(\mathbf{z}_t, \check{\mathbf{x}}_t)$ could be formulated in several ways. Linearly combining $\check{\mathbf{x}}_t$ with an observation based estimate $\mathbf{x}_t(\mathbf{z}_t)$ such as that in (3.1) would only smooth the results, reducing jitter while still continuing to give wrong solutions when the original regressor returns a wrong estimate of $\mathbf{x}_t$. So we build a state sensitive observation update by including a non-linear dependence on $\check{\mathbf{x}}_t$ with $\mathbf{z}_t$ in the observation-based regressor *i.e.* we construct basis functions of the form $\phi(\check{\mathbf{x}}_t, \mathbf{z}_t)$. Our full regression model also includes an explicit linear $\check{\mathbf{x}}_t$ term to represent the direct contribution of the dynamics to the overall state estimate, so the final model becomes $\mathbf{x}_t \equiv \hat{\mathbf{x}}_t + \boldsymbol{\epsilon}'$ where $\boldsymbol{\epsilon}'$ is a residual error to be minimized, and:

$$\hat{\mathbf{x}}_t \; = \; \mathbf{C}\,\check{\mathbf{x}}_t + \sum_{k=1}^{p} \mathbf{d}_k\,\phi_k(\check{\mathbf{x}}_t, \mathbf{z}_t) \; \equiv \; \begin{pmatrix}\mathbf{C} & \mathbf{D}\end{pmatrix} \begin{pmatrix}\check{\mathbf{x}}_t \\ \mathbf{f}(\check{\mathbf{x}}_t, \mathbf{z}_t)\end{pmatrix} \tag{4.2}$$

Here, $\{\phi_k(\mathbf{x}, \mathbf{z}) \,|\, k = 1 \ldots p\}$ is a set of scalar-valued nonlinear basis functions for the regression, and $\mathbf{d}_k$ are the corresponding $\mathbb{R}^m$-valued weight vectors. For compactness, we gather these into an $\mathbb{R}^p$-valued feature vector $\mathbf{f}(\mathbf{x}, \mathbf{z}) \equiv (\phi_1(\mathbf{x}, \mathbf{z}), \ldots, \phi_p(\mathbf{x}, \mathbf{z}))^\top$ and an $m \times p$ weight matrix $\mathbf{D} \equiv (\mathbf{d}_1, \ldots, \mathbf{d}_p)$. $\mathbf{C}$ is an $m \times m$ coefficient matrix that controls the weight of the dynamical prediction term. The final minimization of $\boldsymbol{\epsilon}'$ involves regularization terms for both $\mathbf{C}$ and $\mathbf{D}$.

---

[3]We find that a global model in the form of a single second-order autoregressive process suffices for the kind of motions studied here. A more sophisticated dynamical model based on a mixture of such processes that is capable of tracking through changing aspects of motion and appearance is described in chapter 7.
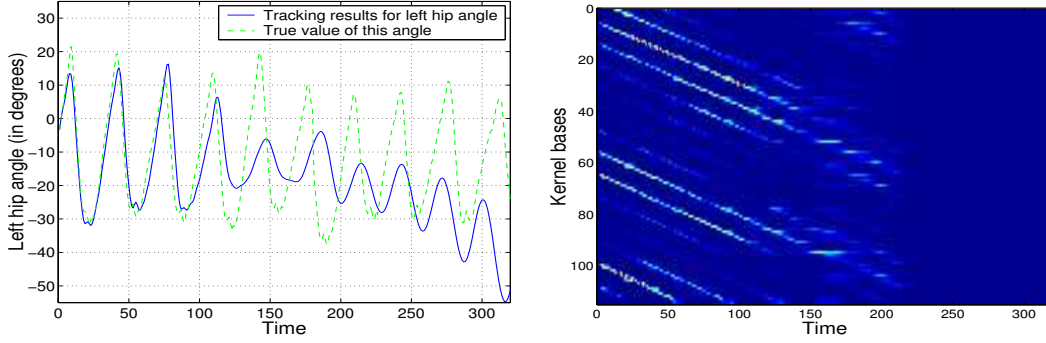
Figure 4.2: *An example of mistracking caused by an over-narrow pose kernel K_x. The kernel width is set to 1/10 of the optimal value, causing the tracker to lose track from about t=120, after which the state estimate drifts away from the training region and all kernels stop firing by about t=200. (Left) the variation of a left hip angle parameter for a test sequence of a person walking in a spiral. (Right) The temporal activity of the 120 kernels (training examples) during this track. The banded pattern occurs because the kernels are samples taken from along a similar 2.5 cycle spiral walking sequence, each circuit involving about 8 steps. The similarity between adjacent steps and between different circuits is clearly visible, showing that the regressor can locally still generalize well.*

For the experiments, we use instantiated-kernel bases that measure similarity in both components $\mathbf{x}$ and $\mathbf{z}$:

$$\phi_k(\mathbf{x}, \mathbf{z}) = K_x(\mathbf{x}, \mathbf{x}_k) \cdot K_z(\mathbf{z}, \mathbf{z}_k) \tag{4.3}$$

where $(\mathbf{x}_k, \mathbf{z}_k)$ is a training example and K_x, K_z are independent Gaussian kernels on $\mathbf{x}$-space and $\mathbf{z}$-space, $K_x(\mathbf{x}, \mathbf{x}_k) = e^{-\beta_x \|\mathbf{x} - \mathbf{x}_k\|^2}$ and $K_z(\mathbf{z}, \mathbf{z}_k) = e^{-\beta_z \|\mathbf{z} - \mathbf{z}_k\|^2}$. Using Gaussians kernels in the combined $(\mathbf{x}, \mathbf{z})$ space makes examples relevant only if they have similar image silhouettes *and* similar underlying poses to training examples. This overcomes the weakness of the original model (3.1) by preventing an ambiguous silhouette from being matched to a similar looking silhouette with a different underlying 3D pose, and thus is able to resolve the ambiguities.

4.2.3   Parameter Settings

The matrices $\mathbf{C}$ and $\mathbf{D}$ in the model (4.2) are normally estimated using ridge or Relevance Vector Machine regression and the kernel widths β_x and β_z in ϕ_k are empirically set using cross validation. The parameter β_x, however, is observed to have a very interesting influence on the system, leading to 'extinction' if set to too small a value. Also, an analysis of performance change with value of the dynamical model coefficient $\mathbf{C}$ gives useful insight into the role played by the linear dynamical term in the model. Both these cases are discussed individually below.

A. Mistracking due to extinction

Kernelization in joint $(\mathbf{x}, \mathbf{z})$ space allows the relevant branch of the inverse solution to be chosen, but it is essential to choose the relative widths of the kernels appropriately. If the $\mathbf{x}$-kernel is chosen too wide, the method tends to average over (or zig-zag between) several alternative pose-from-observation solutions, which defeats the purpose of including $\check{\mathbf{x}}$ in the observation regression.
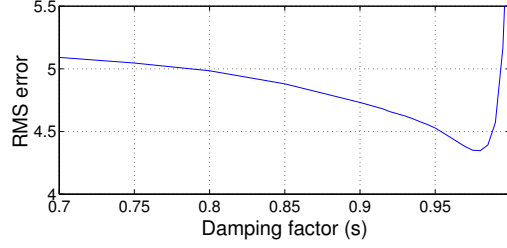
Figure 4.3: *The variation of the RMS test-set tracking error (in degrees) with damping factor* s *that is applied to the coefficient matrix* **C** *of the dynamical prediction term. See the text for a discussion.*

On the other hand, too much locality in **x** effectively 'switches off' the observation-based state corrections whenever the estimated state happens to wander too far from the observed training examples $\mathbf{x}_k$. So if the **x**-kernel is set too narrow, observation information is only incorporated sporadically and mistracking can easily occur. Figure 4.2 illustrates this effect, for an **x**-kernel a factor of 10 narrower than the optimum. After fixing good values by cross-validation on an independent sequence we observed accurate performance of the method over a scale range of about 2 on β_x and about 4 on β_z.

B. Neutral vs Damped Dynamics

The coefficient matrix **C** in (4.2) plays an interesting role. Setting $\mathbf{C} \equiv \mathbf{I}$ forces the correction model to act as a differential update on $\check{\mathbf{x}}_t$ (what we refer to as having a 'neutral' dynamical model). At the other extreme, $\mathbf{C} \equiv \mathbf{0}$ gives largely observation-based state estimates with little dependence on the dynamics. An intermediate setting with **C** near **I** turns out to give the best overall results. Damping the dynamics slightly ensures stability and controls drift — in particular, preventing the observations from disastrously 'switching off' because the state has drifted too far from the training examples — while still allowing a reasonable amount of dynamical smoothing. Usually we estimate the full (regularized) matrix **C** from the training data, but to get an idea of the trade-offs involved, we also studied the effect of explicitly setting $\mathbf{C} = s\mathbf{I}$ for $s \in [0, 1]$. We find that a small amount of damping, $s_{opt} \approx .98$ gives the best results overall, maintaining a good lock on the observations without losing too much dynamical smoothing. This is illustrated in figure 4.3. The simple heuristic setting of $\mathbf{C} = s_{opt}\mathbf{I}$ gives very similar results to the model obtained by learning the full matrix **C**.

4.3   A Condensation based viewpoint

The regressive (discriminative) approach described in § 4.2 can also be understood in the context of a Condensation [63] style Bayesian tracking framework.

Assuming the state information from the current observation is independent of state information from dynamics (which is a common assumption in the tracking literature) and applying Baye's rule, we obtain

$$p(\mathbf{x}_t \,|\, \mathbf{z}_t, \mathbf{x}_{t-1}, \ldots) \;=\; \frac{p(\mathbf{z}_t \,|\, \mathbf{x}_t)}{p(\mathbf{z}_t)}\, p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \ldots) \tag{4.4}$$

A dynamical model gives us $p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \ldots)$. We must now fuse in the information from $\mathbf{z}_t$. The way to do this is to multiply by the contrast $\frac{p(\mathbf{x}_t, \mathbf{z}_t)}{p(\mathbf{x}_t)p(\mathbf{z}_t)} = \frac{p(\mathbf{x}_t \,|\, \mathbf{z}_t)}{p(\mathbf{x}_t)} = \frac{p(\mathbf{z}_t \,|\, \mathbf{x}_t)}{p(\mathbf{z}_t)}$. Here $p(\mathbf{x}_t)$ or $p(\mathbf{z}_t)$ are vague priors assuming no knowledge of the previous state, so have little influence. Approximating the contrast term with the likelihood $p(\mathbf{z}_t \,|\, \mathbf{x}_t)$ gives rise to the standard Bayesian tracking framework, that involves building a generative model for the image observation, given the body pose. In the discriminative model developed here, we ignore the dependence on $p(\mathbf{x})$ and estimate a noise model for the regressor to directly model $p(\mathbf{x}_t \,|\, \mathbf{z}_t)$ as a Gaussian centered at $\mathbf{r}(\check{\mathbf{x}}_t, \mathbf{z}_t)$. The term $\sum_{k=1}^{p} \mathbf{d}_k \, \phi_k(\check{\mathbf{x}}_t, \mathbf{z}_t)$ in (4.2) can thus be interpreted as corresponding to the observation-based state density that replaces the likelihood term.

We implement a modified Condensation algorithm by using the dynamical model from section 4.2.1 to generate an estimate of the 3D pose distribution $p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \ldots)$ — as in the original algorithm — but assigning weights to the samples $(\check{\mathbf{x}}_t^i)$ from this distribution using a Gaussian model centered at the regressor output $\sum_{k=1}^{p} \mathbf{d}_k \, \phi_k(\check{\mathbf{x}}_t, \mathbf{z}_t)$ with covariance learned from the training data. In the following section, it is shown that such a scheme actually performs very similarly to the fully regression based model discussed in § 4.2.2.

4.4 Tracking Results

The regression model (4.2) is trained on our motion capture data using Relevance Vector regression as described in § 3.4.2. The experiments in this chapter use 8 different motion sequences totaling about 2000 instantaneous poses for training, and another two sequences of about 400 points each as validation and test sets. Errors are reported as mean (over all 54 angles) RMS absolute differences between the true and estimated joint angle vectors, as described by (3.8) in the previous chapter.

The dynamical model is learned from the training data using autoregression as described in §4.2.1 and is found to capture the motion patterns quite well. When training the observation model, however, we find that it sometimes fails to give reasonable corrections if the predicted state $\check{\mathbf{x}}_t$ is too far off from the actual state $\mathbf{x}_t$. We thus increase its coverage and capture radius by including a wider selection of $\check{\mathbf{x}}_t$ values than those produced by the dynamical predictions. So we train the model $\mathbf{x}_t = \mathbf{x}_t(\check{\mathbf{x}}_t, \mathbf{z}_t)$ using a combination of 'observed' samples $(\check{\mathbf{x}}_t, \mathbf{z}_t)$ (with $\check{\mathbf{x}}_t$ computed from (4.1)) and artificial samples that generate $\check{\mathbf{x}}_t$ by Gaussian sampling $\mathcal{N}(\mathbf{x}_t, \Sigma)$ around the training state $\mathbf{x}_t$. The unperturbed observation $\mathbf{z}_t$ corresponding to $\mathbf{x}_t$ is still used, forcing the observation based part of the regressor to rely mainly on the observations, *i.e.* on recovering $\mathbf{x}_t$ from $\mathbf{z}_t$, using $\check{\mathbf{x}}_t$ only as a hint about the inverse solution to choose. The covariance matrix Σ for this sampling is chosen to reflect the local scatter of the training example poses, but with increased variance along the tangent to the trajectory at each point so that the model will reliably correct any phase lag between the estimate and true state that builds up during tracking. (Such lags can occur when the observation signal is weak for a few time steps and the model is driven mainly by the dynamical component of the tracker.)

Figure 4.4 illustrates the relative contributions of the dynamics and observation terms in our model by plotting tracking results for a motion capture test sequence in which the subject walks in a decreasing spiral. The sequence was not included in the training set, although similar ones were. The purely dynamical model provides good estimates for a few time steps, but gradually damps and drifts out of phase. Such damped oscillations are characteristic of second order autoregressive systems, trained with enough regularization to ensure model stability. The results (from chapter 3) based on observations alone without any temporal information are included again here for comparison. These are obtained from (3.1), which is actually a special case of (4.2) where $\mathbf{C} = \mathbf{0}$ and $K_x = 1$. Panels (e) and (f) plot results from the combined model described by (4.2), showing that jointly regressing dynamics and observations gives a significant improvement in estimation
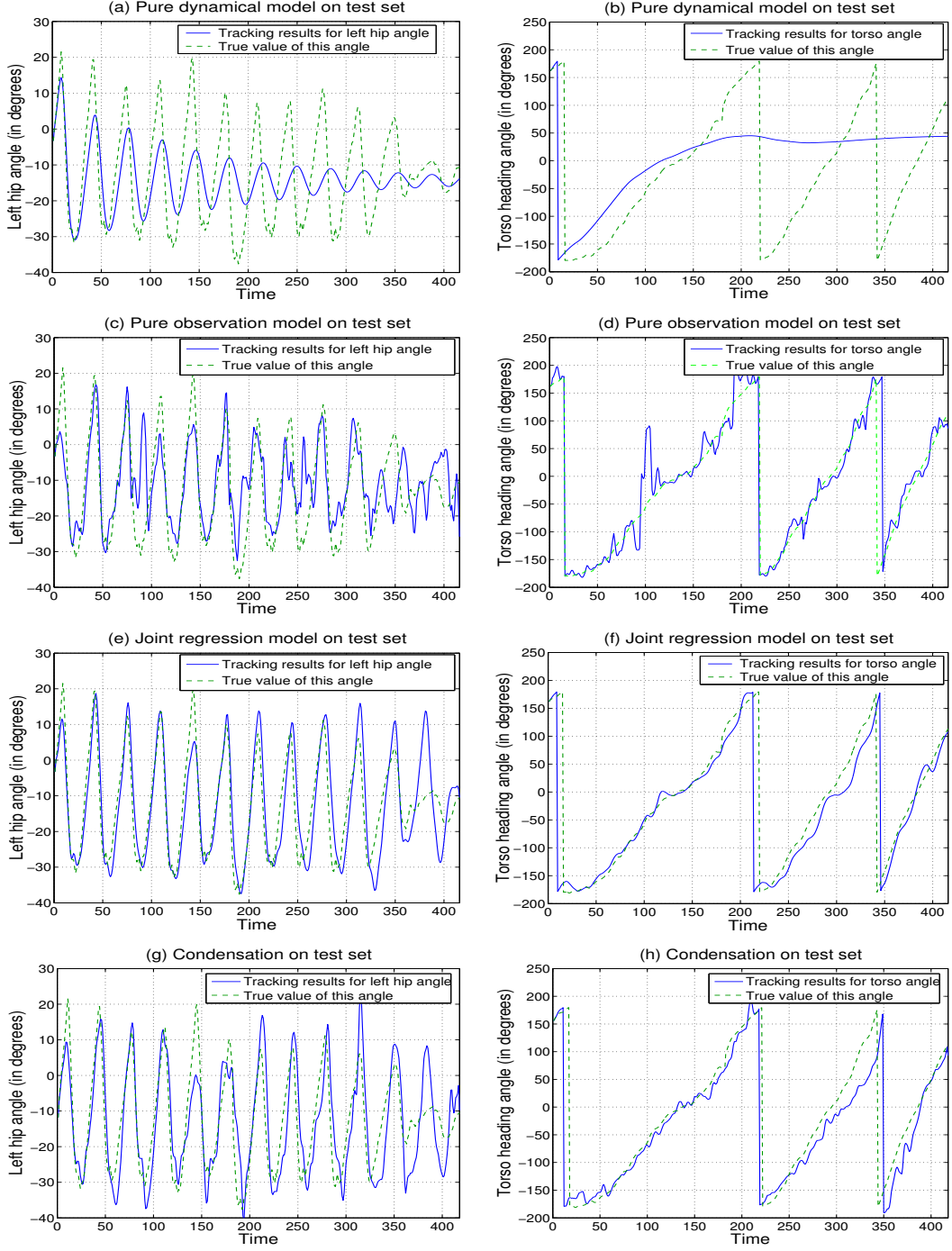
Figure 4.4: *Sample tracking results on a spiral walking test sequence. (a,b) Variation of the left hip-angle and overall body rotation parameters, as predicted by a pure dynamical model initialized at $t = \{0, 1\}$; (c,d) Estimated values of these angles from regression on observations alone (i.e. no initialization or temporal information); (e,f) Results from our novel tracker, obtained by combining dynamical and state+observation based regression models. (g,h) Condensation based tracking, showing a smoothed trajectory of the most likely particle at each time step. Note that the overall body rotation angle wraps around at $360°$, i.e. $\theta \simeq \theta \pm 360°$.*

t=001   t=060   t=120   t=180   t=240   t=300
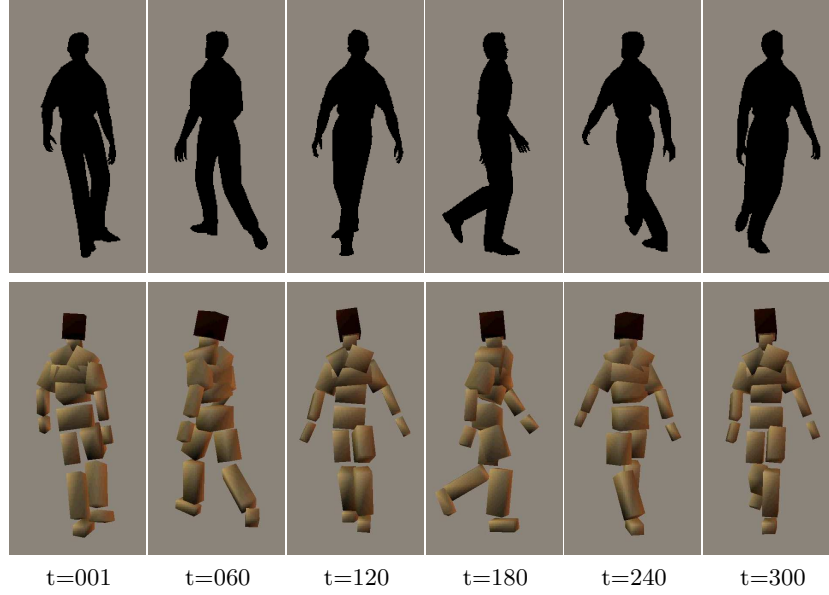
Figure 4.5: *Some sample pose reconstructions for the test spiral walking sequence using the discriminative tracker. These results correspond to the plots in figure 4.4(e) and (f). The reconstructions were computed with a Gaussian kernel RVM which retained only 18% of the training examples. The average RMS estimation error per d.o.f. over the whole sequence is 4.1°.*

quality, with smoother and stabler tracking. There is still some residual misestimation of the hip angle in (e) at around t=140 and t=380. At these points, the subject is walking directly towards the camera (heading angle $\theta \sim 0°$), so the only cue for hip angle is the position of the corresponding foot, which is sometimes occluded by the opposite leg. Humans also find it difficult to estimate this angle from the silhouette at these points. Results from the Condensation based tracker described in section 4.3 are shown in panels (g) and (h). They are very similar to those obtained using the joint regression, but not as smooth.

Figure 4.5 shows some silhouettes and the corresponding pose reconstructions for the same test sequence. The 3D poses for the first two time steps were set from ground truth to initialize the dynamical predictions. The average RMS estimation error over all joints using the Gaussian kernel based RVM regressor in this test is 4.1° and the regressor is sparse, involving only 348 (18%) of the 1927 training examples. Note that this solution is less sparse than the regression based on silhouette observations alone (§ 3.5.2). This shows that additional examples are required to resolve the ambiguities. Well-regularized least squares regression over the same basis gives similar errors, but has much higher storage requirements. Figure 4.6 shows reconstruction results on the lateral walking test video sequence that was used in the previous chapter. The incorrect solutions from ambiguities that were observed in figure 3.10 are no longer present here. However, the poor quality silhouettes do effect the output of the observation component, causing the dynamical estimate to *drive* the tracker at some time instants. As a result, the predicted motion is sometimes not synchronized with the observations (*e.g.* at $t = 14$, the arms overshoot because of domination of the dynamical model over the observation signal).

Figures 4.7 and 4.8 show the performance on further test video sequences. In the first sequence, the method tracks though a scale change by a factor of ~ 2, as the subject walks towards the camera. Note that as the silhouette representation is invariant to the scale/resolution of an image, no rescaling/downsampling of the test images is required — images and silhouettes in the figure

t=02   t=08

t=14   t=20

t=26   t=32
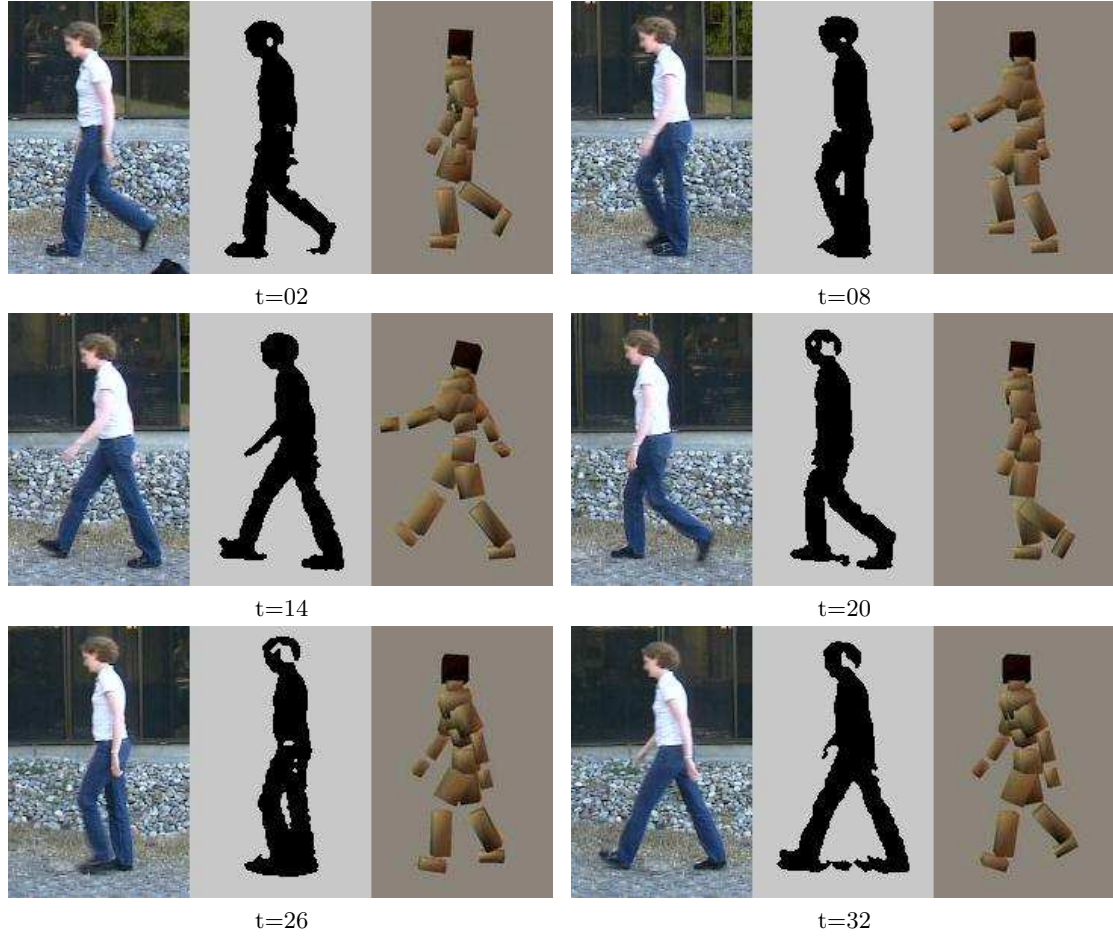
Figure 4.6: *3D poses reconstructed from a test video sequence. The presence of shadows and holes in the extracted silhouettes demonstrates the robustness of the shape descriptors — however, a weak or noisy observation signal sometimes causes failure to track accurately. E.g. at $t = 8, 14$, the pose estimates are dominated by the dynamical predictions, which ensure smooth and natural motion but cause slight mistracking of some parameters.*

have been normalized in scale only for display purposes. The second sequence is an example of a more complicated motion — the subject often changes heading angle, walking in several different directions. For this example, the system was trained on a somewhat similar sequence of the same person to ensure a wider coverage of his poses. Also, the motion capture data used for the training was in a different format[4], so we used a 44D joint angle representation in this experiment, again demonstrating that the methods developed here are independent of the body pose representation.

In terms of computation time, the final RVM regressor runs in real time in Matlab. Silhouette extraction and shape-context descriptor computations are performed offline in these experiments, but are feasible online in real time. The offline learning process takes about 2-3 minutes for the RVM with $\sim$2000 data points, and about 20 minutes for shape context extraction and clustering.

---

[4]The data used for this experiment was taken from http://mocap.cs.cmu.edu and is in the 'Acclaim Motion Capture' format. Note that although this database consists of both motion capture data (@120 fps) and the corresponding video recordings (@30 fps), the two are not synchronized. For all experiments in this thesis, the data has been approximately synchronized by hand.

Figure 4.7: *3D pose reconstructions on an example test video sequence. The scale invariant silhouette representation allows the method to successfully track through a scale change by a factor of ∼2 as the subject walks towards the camera. The images and silhouettes have been normalized in scale here for display purposes.*
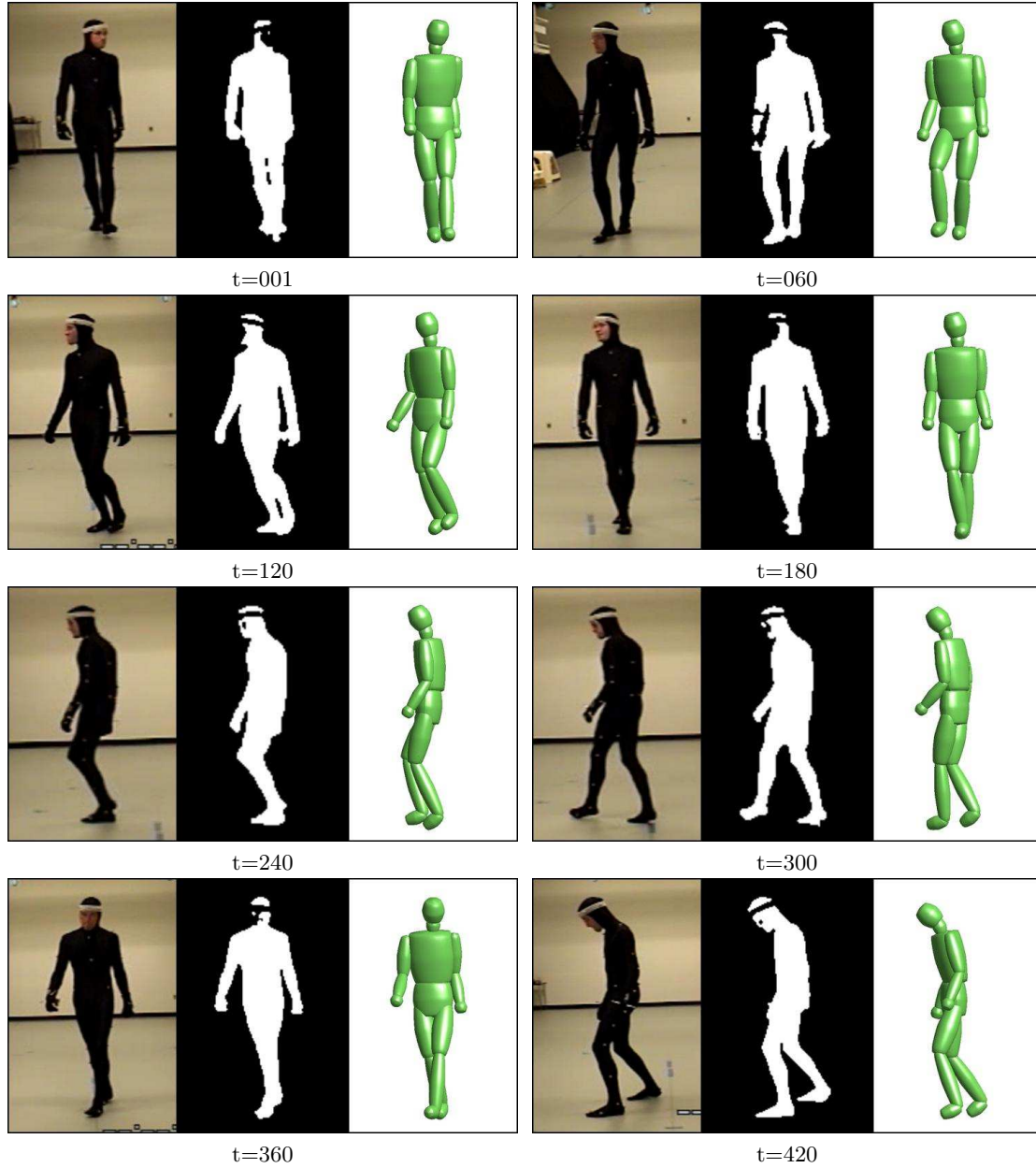
Figure 4.8: *3D pose reconstructions on another example test sequence on which the method was tested. The motion here is more complicated and the subject often changes heading angle in this sequence, walking randomly in different directions. The method successfully tracks through 600 frames.*

However this is for a highly unoptimized Matlab code and a considerable speed-up should be possible simply by an optimized reimplementation, *e.g.* in C/C++.

4.4.1   Automatic Initialization

The tracking framework described here explicitly makes use of pose estimates from the previous two time steps while computing the pose at any time instant. This is a problem for initializing the tracker. All the results shown here were obtained by initializing manually or from ground truth where available, but we also tested the effects of automatic (and hence potentially incorrect) initialization using the single image based regression method from chapter 3. The method is found to be reasonably robust to small initialization errors, but fails to track when the initialization is completely incorrect. In an experiment in which the tracker was initialized automatically at each of the time steps using the pure observation model, then tracked forwards and backwards using the dynamical tracker, the initialization led to successful tracking in 84% of the cases. The failures were the glitches where the observation model gave completely incorrect initializations.

4.5   Discussion

This chapter discusses a novel discriminative tracking framework that is based on regression and avoids expensive image likelihood computations. It is demonstrated that nonlinearly incorporating a dynamical model based pose estimate into the basis functions of a kernel regressor can successfully fuse information from dynamics and observations for smooth tracking. Sparsity properties of the Relevance Vector Machine are exploited for computational efficiency and better generalization — the tracker retains only 15-20% of the training examples, thus giving a considerable reduction in storage space compared to nearest neighbour methods which must retain the whole training database.

The main shortcoming of the method is that the tracker maintains a single hypothesis for pose at each time step. Although the observation model is designed to have a weak dependence on the previous state estimate and can recover from slight mis-estimations, any reasonable error will be propagated ahead in time, making the system susceptible to mistracking. Similarly, the tracker cannot deal with incorrect initialization. One way to resolve these issues is to maintain multiple hypotheses that could probabilistically be combined for more robust tracking. We formulate such a multiple hypothesis tracker in the next chapter.

5

A Mixture of Regressors

5.1   Introduction

3D human motion capture from monocular images often encounters the problem of multiple possible solutions due to ambiguities in the observation. In chapter 3, we saw that using a single regressor to map silhouettes to 3D pose gives very wrong estimates of the pose in about 15% of the cases. The previous chapter described one approach to resolve this problem — fusing dynamics into a regressor in order to use temporal information in a single hypothesis tracking framework. This chapter develops a method based on an alternate approach that explicitly calculates several possible pose hypotheses from a single image in order to compensate for ambiguities in the pose reconstruction problem. We show that these multimodal pose estimates can be used to build a robust and automatically initializing multiple hypothesis tracker.

The underlying idea is to handle the 'multi-valuedness' by learning multiple regressors, each spanning a small set of examples that is free of ambiguities. To handle nonlinearities effectively, the regressors are learned on a manifold within the silhouette descriptor space. A new input point, *i.e.* a silhouette descriptor, is then mapped, via its projection into this manifold, to (potentially) as many 3D poses as there are regressors. In practice, since each of these regressors is local, only a few of a solutions are actually probable. This information is encoded by a latent 'regressor class' variable that is associated with each input observation.

To learn the model, we use locality on the manifold in the input space and connectivity on a neighbourhood graph in the output space to identify regions of multi-valuedness in the mapping from silhouette to 3D pose. This is then used to initialize an iterative process that estimates a locally optimal set of regressors along with their regions of support. The resulting model also learns to predict the gating probabilities of each of these regressors for a given input, and hence is capable of predicting probability values associated with the multiple pose estimates. These multivalued pose estimates are very valuable when tracking motions through a sequence of images because they allow the system to deal directly with multiple hypotheses. In this chapter, we use the discriminative version of the particle filter as described in § 4.3 for multiple hypothesis tracking. Unlike the tracker developed in chapter 4, the framework developed here allows for automatic initialization in a video sequence as it does not explicitly rely on the previous time step to estimate pose from an image. The method also detects tracking failures and accordingly re-initializes itself.

5.2   Multimodal Pose Estimation

To begin with, let us introduce a latent variable l to account for the missing information in a silhouette. This will implicitly capture the limb labeling and kinematic-flipping [142] possibilities

of the given silhouette. The central assumption is that given the value of $\mathbf{l}$, the 3D pose $\mathbf{x}$ has a functional dependence on the observed silhouette:

$$\mathbf{x} \,|\, \mathbf{l} \;\simeq\; \mathbf{r_l(z)} + \boldsymbol{\epsilon_l} \tag{5.1}$$

where $\mathbf{z}$ is the observation (silhouette shape descriptor vector), $\mathbf{r_l}$ is a functional transformation from $\mathbf{z}$ to $\mathbf{x}$, and $\boldsymbol{\epsilon_l}$ is a noise vector. Modeling $\boldsymbol{\epsilon_l}$ as a Gaussian with zero mean and covariance $\boldsymbol{\Lambda_l}$, the conditional pose distribution is written as

$$p(\mathbf{x} \,|\, \mathbf{z}, \mathbf{l}) = \mathcal{N}(\mathbf{r_l(z)}, \boldsymbol{\Lambda_l}) \tag{5.2}$$

A simple model is to have a separate latent variable for each of the unknown factors associated with a silhouette. In the past, latent variables for disambiguating human silhouettes have been used in the form of explicit left/right limb labellings [60]. In principle, this could be extended to labeling the 3D kinematic flipping possibilities (motions towards/away from the camera that leave the image unchanged) that represent the main residual reconstruction ambiguity once the image limbs have been labeled. However, this would require an exponential number of labels to account for all of the flipping possibilities across all limbs. In practice, such a fine level of labeling is not really needed to disambiguate between the probable pose hypotheses — we find that there are typically only a hand-full of probable *modes* (3D pose solutions) that may correspond to a given silhouette. Furthermore, the multiplicity of the solutions usually persists over considerable subspaces within the silhouette space. A reasonable alternative is thus not to attach any physical meaning to the latent variable but rather learn its values automatically for different silhouettes so as to capture whatever information is required for disambiguating between typical human poses. Here, we model the latent variable $\mathbf{l}$ as belonging to a discrete set[1]: $\mathbf{l} \in \{1, 2 \ldots K\}$.

Marginalizing over all possible values of $\mathbf{l}$ in (5.2), we obtain the pose distribution for a given observation $\mathbf{z}$ as

$$p(\mathbf{x} \,|\, \mathbf{z}) = \sum_{k=1}^{K} p(\mathbf{l} = k \,|\, \mathbf{z}) . \, \mathcal{N}(\mathbf{r}_k(\mathbf{z}), \boldsymbol{\Lambda}_k) \tag{5.3}$$

The output state density is thus a linear combination of the different regressor responses in which the response of the kth regressor is weighted by its conditional gating probability $p(\mathbf{l} = k \,|\, \mathbf{z})$. This gives rise to a mixture of uncertain regressors, which is also known as a mixture of experts [64]. *c.f.* [19, 115].

5.3 Model Formulation

So far we have seen the form of the conditional density $p(\mathbf{x}|\mathbf{z})$ for multimodal pose estimation. To work in a completely probabilistic setting, we would also need to estimate the density for $p(\mathbf{z})$ to allow us to measure the reliability of an observation. In this section, we see how both of these can actually be modeled using a single density estimation algorithm.

---

[1]Besides the small number of typically possible reconstructions, there are other attributes that can potentially be captured by latent variables. For example, inter-person variations may also be considered to be discrete, in the form of a finite number of 'person classes'.
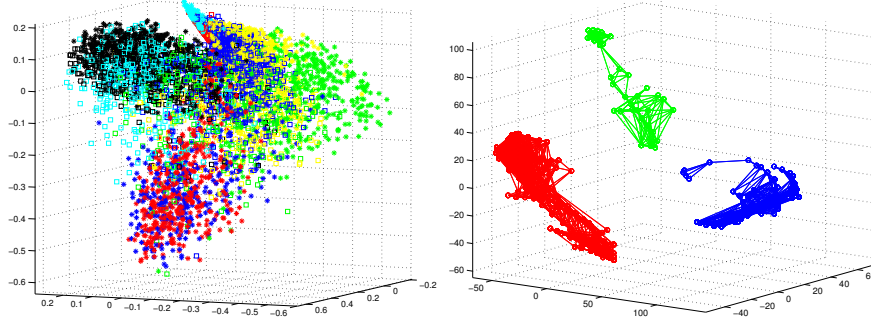
Figure 5.1*: (Left): Initial clusters in $\Psi(\mathbf{z})$ obtained by running k-means with k=12. The plot shows a projection on the first 3 kernel principal components, with the different clusters colour-coded. (Right): 3 connected components are obtained for one of these clusters, as seen on the neighbourhood graph of the corresponding points in $\mathbf{x}$. This cluster is thus split into 3 sub-clusters to separate the different pose subclasses that it contains. Of the 12 initial clusters in $\Psi(\mathbf{z})$, we find that 3 get split into 2 sub-clusters each and 2 into 3 sub-clusters each based on this connectivity analysis. A few of these merge into others during the EM process, giving a final model consisting $\sim$20 clusters.*

5.3.1   Manifold learning and Clustering

Given the nonlinearities in the mapping from $\mathbf{z}$ to $\mathbf{x}$, we first identify a reduced manifold within the the input feature space $\mathbf{z}$ on which the local mappings can be approximated with linear functions. Any nonlinear dimensionality reduction technique may be used for this purpose, *e.g.* [122, 148]. Here, we perform a kernel PCA [126] to obtain a reduced representation[2] $\Psi(\mathbf{z})$ for the input $\mathbf{z}$. We can imagine $\Psi(\mathbf{z})$ as the coordinates of the silhouette descriptor on a manifold that is folded over onto itself due to many-to-one projection mappings. To allow for multimodal output distributions, the mapping to the output space is now learned as a mixture of linear regressors on the reduced space $\Psi(\mathbf{z})$. Each of the regressors is thus modeled as

$$\mathbf{x} = \mathbf{r}_k(\mathbf{z}) + \boldsymbol{\epsilon}_k \;\; \equiv \;\; \mathbf{A}_k \, \Psi(\mathbf{z}) + \mathbf{b}_k + \boldsymbol{\epsilon}_k \tag{5.4}$$

where $\mathbf{A}_k$ and $\mathbf{b}_k$ are coefficients to be estimated and $\boldsymbol{\epsilon}_k$ is the uncertainty associated with the regressor, having a constant covariance $\boldsymbol{\Lambda}_k$ independent of $\mathbf{z}$.

The complete learning process takes place in an iterative framework based on the Expectation Maximization (EM) algorithm [35] which guarantees convergence to a local minimum but relies on good initialization for attaining a globally optimal solution. The key to successful learning is thus to clearly separate the ambiguous cases into different mixture components (clusters) at initialization. Otherwise the individual regressors tend to average over several possible solutions. For this, we first use k-means to divide the KPCA-reduced space $\Psi(\mathbf{z})$ into several clusters. (This corresponds to performing a spectral clustering in the original space $\mathbf{z}$ [102].) Each of these clusters is then split into sub-clusters by making use of the corresponding $\mathbf{x}$ values (which we assume to encode the *true* distance between points), exploiting the fact that silhouettes appearing similar in $\Psi(\mathbf{z})$ can be disambiguated based on the distance between their corresponding 3D poses. This is achieved by constructing a neighbourhood graph in $\mathbf{x}$ that has an edge between all points within

---

[2]The manifold projection shown in chapter1 (figure 1.3) was actually obtained by using KPCA to embed $\mathbf{z}$ into a 3-dimensional space. In practice, the dimensionality is much larger than 3, as will be seen later in this chapter.
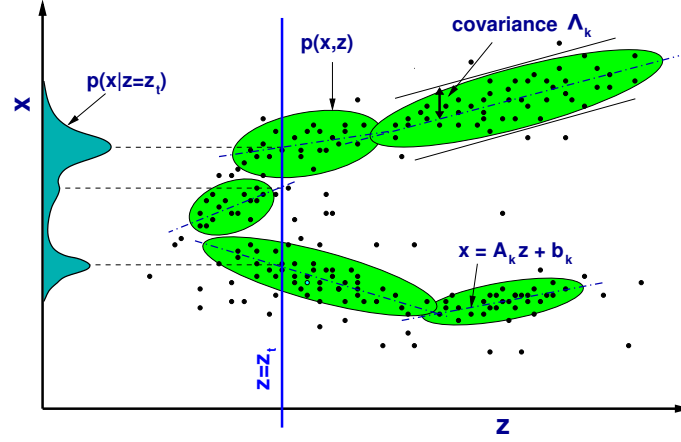
Figure 5.2: *An illustration of the density estimation / regression mixture model used to estimate the conditional density $p(\mathbf{x} \,|\, \mathbf{z})$.*

a thresholded distance from one another, and robustly identifying connected components in this graph for each cluster in $\Psi(\mathbf{z})$. An example illustrating the process is shown in figure 5.1. We find that this two-step clustering separates most ambiguous cases and gives better performance than the other initialization methods that we tested. For example, in terms of final reconstruction errors on a test set after EM based learning (see below), clustering in either $\mathbf{x}$ alone or jointly in $(\mathbf{x}, \Psi(\mathbf{z}))$ is found to give reconstruction errors higher by 0.3 degrees on average, while clustering in $\Psi(\mathbf{z})$ alone shows several instances of averaging across multiple solutions owing to the inability to resolve the ambiguities, also increasing the average error.

5.3.2   Expectation-Maximization

Having obtained a set of clusters, each of which are known to be free of multivaluedness, the individual regressors can directly be learned using the methods described in chapter 3, but in order that the output of these regressors may be combined probabilistically, there are several other components that need to be learned: the likelihood of a given observation $p(\mathbf{z})$, the probability $p(\mathbf{l} = k \,|\, \mathbf{z})$ that the solution from the kth regressor is correct, and also the uncertainty Λ_k associated with each regressor. All these are obtained by using the initial clusters to fit a mixture of Gaussians to the joint density of $(\Psi(\mathbf{z}), \mathbf{x})$:

$$\begin{pmatrix} \Psi(\mathbf{z}) \\ \mathbf{x} \end{pmatrix} \simeq \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Gamma}_k) \tag{5.5}$$

where π_k are the gating probabilities $p(\mathbf{l} = k)$ of the respective classes and $\boldsymbol{\mu}_k, \boldsymbol{\Gamma}_k$ are their means and covariances. Combining the regression model defined in (5.4) into this density model now gives the following relations for these quantities:

$$\boldsymbol{\mu}_k = \begin{pmatrix} \Psi(\bar{\mathbf{z}}_k) \\ \mathbf{r}_k(\bar{\mathbf{z}}_k) \end{pmatrix}, \boldsymbol{\Gamma}_k = \begin{pmatrix} \boldsymbol{\Sigma}_k & \boldsymbol{\Sigma}_k \mathbf{A}_k^\top \\ \mathbf{A}_k \boldsymbol{\Sigma}_k & \mathbf{A}_k \boldsymbol{\Sigma}_k \mathbf{A}_k^\top + \boldsymbol{\Lambda}_k \end{pmatrix} \tag{5.6}$$

The various components of the full model are shown in figure 5.2. Within each regressor, the model has a conditional covariance of $\mathbf{\Lambda}_k$ for $\mathbf{x}\,|\,\mathbf{z}$ (the vertical "thickness" of the classes in figure 5.2). To this, the uncertainty $\mathbf{A}_k\mathbf{\Sigma}_k\mathbf{A}_k^\top$ inherited from $\mathbf{z}$ via $\mathbf{A}$ is added to form the full covariance $(\mathbf{A}_k\mathbf{\Sigma}_k\mathbf{A}_k^\top + \mathbf{\Lambda}_k)$ for $\mathbf{x}$. To avoid overfitting, we assume the descriptor covariance matrices $\mathbf{\Sigma}_k$ and the residual noise covariances $\mathbf{\Lambda}_k$ to be diagonal.

The parameters of the mixture and the regressors are estimated using EM. The 'M' step consists of two parts: First, $\mathbf{A}_k, \mathbf{b}_k$ are estimated by weighted least squares regression (each example being weighted by its responsibility for the given class) using the linear model given in (5.4), followed by estimating the covariances $\mathbf{\Lambda}_k$ from the residual errors. Second, the statistics $\boldsymbol{\mu}_k, \mathbf{\Sigma}_k, \pi_k$ for each class are computed, given the class membership weights for each point (initialized from the clustering above). The 'E' step, as usual, involves re-estimating the membership weights (responsibilities) for each point given the statistics of each class. The process is iterated to convergence, which takes 30-40 iterations. Occasionally, a few of the clusters 'die out' as their points are merged with others. The EM process 'smooths' the initial clusters, giving better generalization in terms of test set performance by exploiting the global mapping between the $\mathbf{z}$ and $\mathbf{x}$ spaces. At the same time, the initial *structure* contained in the connectivity based clustering is retained, so ambiguous cases are handled by different regressors.

5.3.3   Inference

The inference step involves computing the likelihood of a new observation $p(\mathbf{z})$ and the conditional pose estimate $p(\mathbf{x}|\mathbf{z})$. We first compute $p(\mathbf{z})$ by marginalizing over all components:

$$p(\mathbf{z}) \;=\; \sum_{k=1}^{K} p(\mathbf{l}{=}k)\, p(\mathbf{z}\,|\,\mathbf{l}=k) \;=\; \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\Psi(\bar{\mathbf{z}}_k), \mathbf{\Sigma}_k)\,|\,{}_{\Psi(\mathbf{z})} \tag{5.7}$$

where $\mathcal{N}(\Psi(\bar{\mathbf{z}}_k), \mathbf{\Sigma}_k)\,|\,{}_{\Psi(\mathbf{z})}$ is the Gaussian function with mean $\Psi(\bar{\mathbf{z}}_k)$ and covariance $\mathbf{\Sigma}_k$, evaluated at the point $\Psi(\mathbf{z})$. Finally, Baye's rule can be used to derive the regressor gating probabilities:

$$p(\mathbf{l}{=}k\,|\,\mathbf{z}) \;=\; \frac{p(\mathbf{l}=k)\, p(\mathbf{z}\,|\,\mathbf{l}=k)}{p(\mathbf{z})} \;=\; \frac{\pi_k \cdot \mathcal{N}(\Psi(\bar{\mathbf{z}}_k), \mathbf{\Sigma}_k)\,|\,{}_{\Psi(\mathbf{z})}}{\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\Psi(\bar{\mathbf{z}}_k), \mathbf{\Sigma}_k)\,|\,{}_{\Psi(\mathbf{z})}} \tag{5.8}$$

The conditional probability $p(\mathbf{x}|\mathbf{z})$ is now estimated by directly substituting these gating probabilities in (5.3).

5.4   Analysis and Performance

In this section we analyze the accuracy of the mixture model in identifying ambiguities and estimating full 3D body pose from single silhouette images. We use motion capture data along with corresponding synchronized image sequences for training the system. A lot of the motion capture data, however, does not have associated images. For this part, we render each pose with several different human models (from Poser) to capture inter-person variations and also increase the amount of synthetic training data to ~ 8000 pose-image pairs (see figure 5.3). The body pose $\mathbf{x}$ is recovered as a vector of joint angles and the image descriptors $\mathbf{z}$ are computed using 100D histograms of local shape context descriptors as in the earlier chapters. The nonlinear embedding

Figure 5.3*: A sample pose rendered using different 'synthetic people'. To study the effect of inter-person variations in appearance, a part of the motion capture data is used to create multiple training images in this manner.*

	% of frames with m solutions			Error in the	Error in best of
	$m = 1$	$m = 2$	$m \geq 3$	top solution	top 4 solutions
(A)	62	28	10	6.14	4.84
(B)	65	28	6	7.40	5.37
(C)	72	23	5	6.14	4.55

Figure 5.4*: The numbers of solutions and the errors (RMS of joint angles in degrees) obtained when reconstructing three different datasets. To count the number of modes predicted, we consider only modes with $p > 0.1$. See text for explanation.*

$\Psi(\mathbf{z})$ is obtained by kernel PCA using a polynomial dot product kernel based on the Bhattacharya measure for histogram similarity: $K(\mathbf{z}_1, \mathbf{z}_2) = \langle \sqrt{\mathbf{z}_1}, \sqrt{\mathbf{z}_2} \rangle^p$. With p=6, the 100D vectors $\mathbf{z}$ are reduced to 23D vectors $\Psi(\mathbf{z})$[3].

To quantify performance, we first measure accuracy on two test sets consisting of the artificially synthesized silhouettes so that we can also study the robustness with respect to inter-person variations. The first test set (A) consists of ∼600 frames of a person not included in the training data, and the second test set (B) consists of ∼400 frames of a person in the training data but with a different motion sequence. For comparison, we also report errors on a subset (C) of ∼600 frames from the original training set. We find that the mixture generally outputs between 1 and 3 high probability solutions for each silhouette, with the highest probability solution often but not always being the correct one. Statistics of the multimodalities are summarized in figure 5.4. We also measure the accuracy of the solution from the regressor that is predicted to be the most probable (this is called the 'top solution') and the accuracy if we consider the best prediction from amongst the 4 most probable regressors, where ground truth is used to identify the best case. Errors in figure 5.4 are reported as average RMS deviations for each joint-angle in degrees. The table shows that most of the estimates actually have one or two high-probability solutions, and that on an average, the best solution is better than the top solution, which means that the errors are sometimes due to wrong gating probabilities. The better overall performance on test (A) than on test (B) suggests that the model generalizes better between different appearances than between different motion patterns.

Figure 5.5 shows sample reconstructions on some synthetic and real test silhouettes, none of which were included in the training data. Since ground truths were not available for all of these silhouettes,

---

[3]A linear PCA on the same data requires 91 dimensions to retain 99% variance in our data.
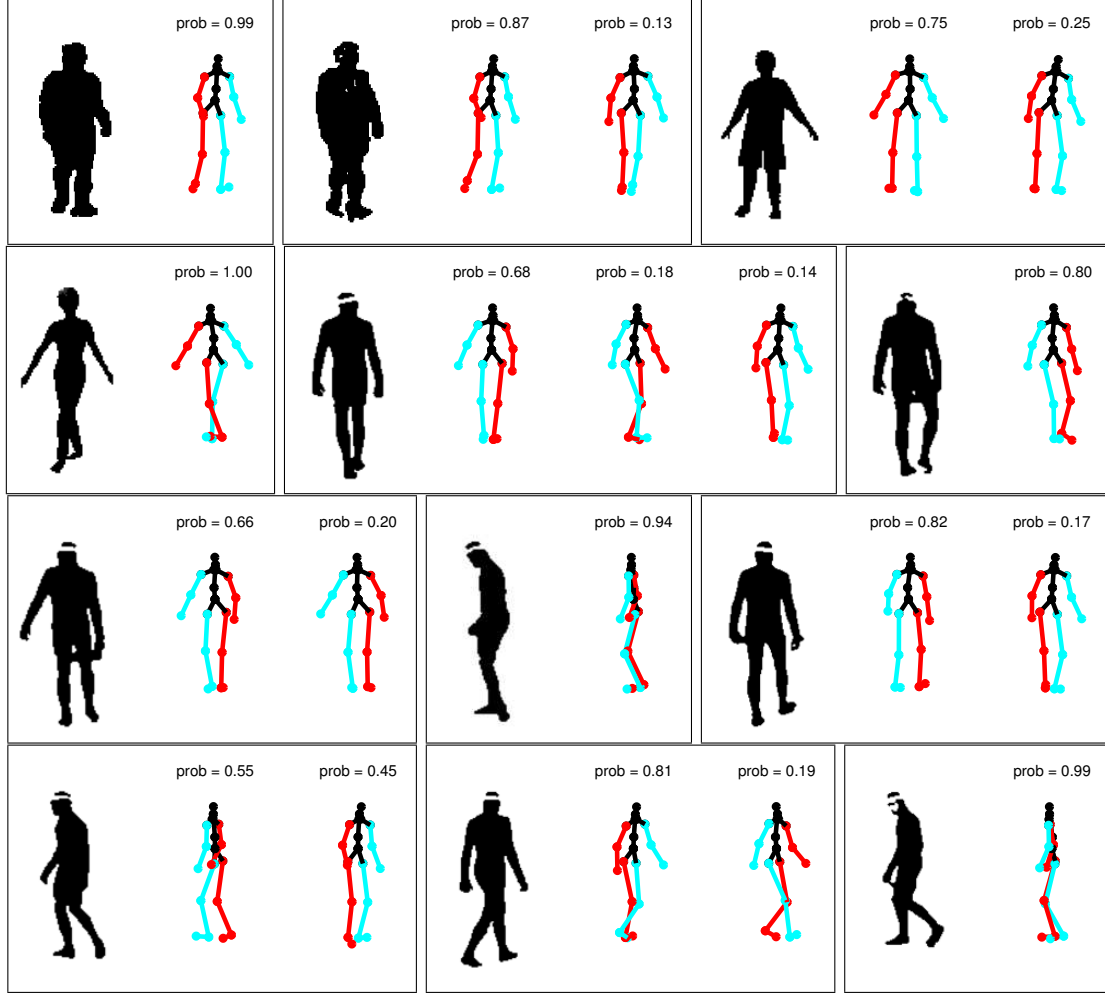
Figure 5.5: *3D pose estimates obtained on test silhouettes of people that are not present in the training set. The mean pose estimates from all modes with probability greater than 0.1 are displayed, with red (dark gray) denoting right and blue (light gray) denoting left limbs. The method mostly gives accurate pose reconstructions, identifying the different possible solutions when there are ambiguities. Some interesting and non-obvious forward-backward ambiguities in the silhouettes are revealed. For instance in the second-last example, a careful look at the silhouette will convince the reader that there are indeed two possible solutions — corresponding to the person walking into the image plane, 45° towards the right, or out of the image plane, again 45° to the right. This is correctly identified by the system. (Note that the arms are clearly interchanged in the 2 cases.)*
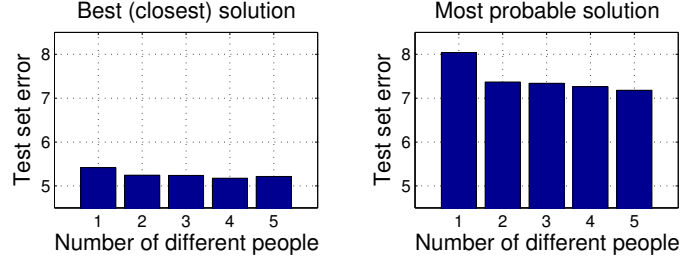
Figure 5.6: *Generalization performance across different people. By varying the number of people in the training set from one to five and testing on all of the others, we find that there is very little improvement in the performance of individual regressors, as seen by the accuracy of the closest (to ground truth) solution on the left, but a slightly more significant increase in the system's ability to select probable solutions, as seen on the right. The errors (shown in degrees) were obtained using k-fold cross validation in each case.*

we visually inspected the reconstructions on 300 frames of one of the real test sequences to quantify the quality of reconstructed modes. We find that in 47% of images, the highest ranked solution gives a good reconstruction. In respectively 24% and 13% of the images, the second, or one of the third or fourth modes give suitable reconstructions, while in the remaining 16%, the system fails to give the correct pose estimate within the top 4 modes. One place where this happens is at the points where multiple surfaces split or merge in the $\mathbf{z}-\mathbf{x}$ space. As seen in the last example in the figure, the regressors sometimes still average over multiple poses (in this case, the two possible leg labelings) in certain cases. This is however, a 'special case' where the two training examples are not only very similar in the observation space $\mathbf{z}$, but are also difficult to disambiguate based on distance in $\mathbf{x}$ space as their poses are not too far apart. The local connectivity information used in the learning process fails to disambiguate such regions because they often to belong to the same connected region in the pose neighbourhood graph.

We also tested the effect of training on appearances of different people. As expected, the generalization to appearance improves when more people are added to the training database. As shown in figure 5.6, the improvement is relatively minor as regards the accuracy of individual regressors, although the ability to select the correct solutions shows a slightly more significant improvement. Overall the method generalizes among people surprisingly well. We attribute this to the robust representation of the silhouette shape using shape context distribution histograms described in chapter 3. Here, the 100 centre codebook was learned by clustering shape contexts from the silhouettes of several people.

5.5   Self-Initialized 3D Tracking

The solutions obtained from the multiple hypothesis pose estimator can be used to obtain smooth reconstructions of 3D human body motion across a video sequence. Intuitively, this involves selecting the correct mode for each image with the extra constraint that temporal contiguity is maintained. In practice, the tracker probabilistically samples from a continuous state density. We use a modified Condensation based tracker similar to the one described in § 4.3, this time replacing the image likelihoods $p(\mathbf{z}|\mathbf{x})$ with the multimodal pose estimate densities $p(\mathbf{x}|\mathbf{z})$ that are returned by the mixture of regressors.

Since the mixture model returns multiple hypotheses from each image, the pose density from the first frame can be used to automatically initialize the tracker in a robust manner. Also, recall
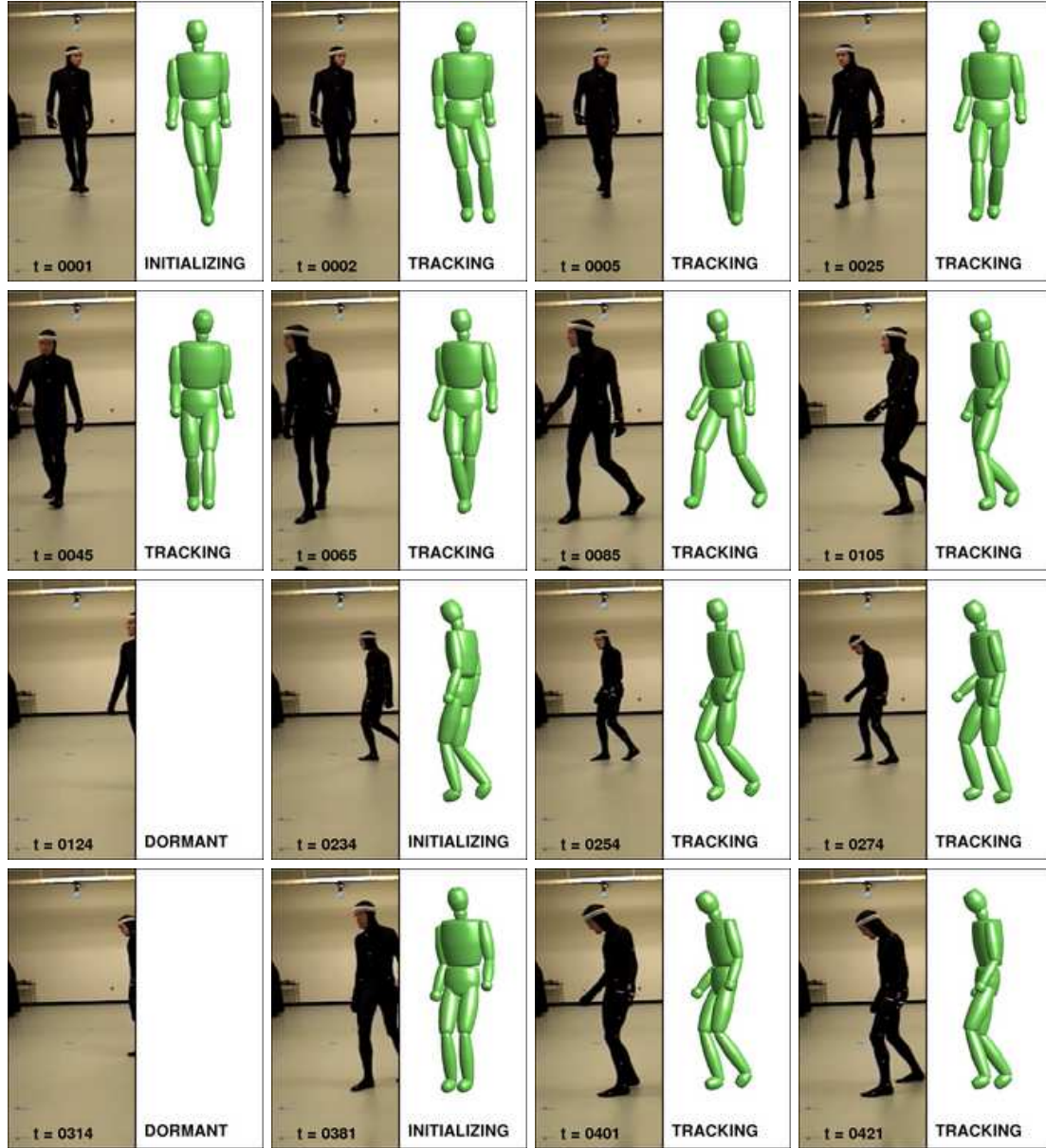
Figure 5.7: *Snapshots from multiple hypothesis tracking of a person across 500 frames. Our direct and probabilistic pose estimation from the image allows automatic initialization, and re-initialization on detecting tracking failure or absence of a person (see text). Maintaining multiple track hypotheses allows the tracker to recover from possibly inaccurate initializations, tracking stably through instances where the person is not observed. The overall error with time for this track is shown in figure 5.8.*
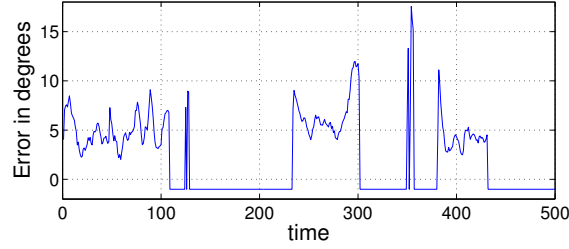
Figure 5.8: *Error of the tracker (RMS deviations for individual joint angles) across the sequence shown in figure 5.7. A value of -1 indicates that no person is detected and the tracker is in a 'dormant' state, waiting to automatically reinitialize itself when the observation probability $p(\mathbf{z})$ increases.*

that the model estimates $p(\mathbf{z})$ at each time step. This serves as a detection mechanism for a valid human-like shape being present in the image and allows the tracker to recover from failures that are typically associated with temporal continuity based trackers. (*c.f.* [163] where including detection at each frame is used to detect and recover from tracking failures). A special case of this is when the subject disappears from the field of view of the camera and $p(\mathbf{z})$ falls to zero. In such cases, the tracker detects images where the person is not observed and shifts into a 'dormant' state. Re-initialization automatically takes place when he/she re-enters the scene (*i.e.* $p(\mathbf{z})$ increases), allowing tracking to continue after failures. The implementation used here is based on a simple threshold on $p(\mathbf{z})$ in order to carry out this re-initialization but a probabilistic re-initialization scheme could easily be incorporated.

Figure 5.7 shows sample frames from the tracking of a real motion sequence in which the subject disappears from the field of view a couple of times. The body pose is successfully tracked through the 500 frames, the tracker being automatically (re)initialized at $t = 1$, 234 and 381. Although the initializations are not always perfect, multiple hypothesis tracking allows the correct modes to emerge after a few frames, giving a stable track. The overall error in pose estimation across the sequence is shown in figure 5.8. Rapid stabilization is seen in both cases of reinitialization at $t = 234$ and 381. Instances of false detection and initialization are visible at $t \sim 125$ and 350. The reconstructions in figure 5.7 show the most likely particle at any given instant, but do not necessarily reflect the optimal temporal sequence of state estimates. The latter may be obtained, *e.g.*, by back-tracing the particles that contributed highly likely tracks — a mechanism that also resolves the ambiguities present in individual images by exploiting temporal coherency.

5.6   Gesture recognition using the Mixture Components

An interesting by-product of using a mixture of regressors to track human motion is the posterior probability value of each regressor class for a each image, $p(\mathbf{l} = k \mid \mathbf{z})$. These values can be used to deduce the label of the mixture component that was used to reconstruct pose at any given time instant. Now the mixture components not only help to resolve ambiguities by separating regions of multivaluedness, but also softly partition the space into small regions having consistently similar appearance and pose. In this section, we exploit this fact to use the model for labeling different gestures in a video sequence.

The system is retrained on a set of images from of a sequence of well defined arm gestures as shown in figure 5.9. For this experiment, we use the algorithm defined in § 5.3, but initialize the clusters manually on the basis of the ground truth gesture label associated with each image-pose pair. The

| Traveling | Illegal dribble | Illegal defense | No score |
| Stop clock | Hand check | Technical foul | (Neutral) |

Figure 5.9: *Sample frames from 7 basketball referee signals that are used as training gestures for learning a mixture of regressors capable of inferring gesture along with reconstructing pose.*
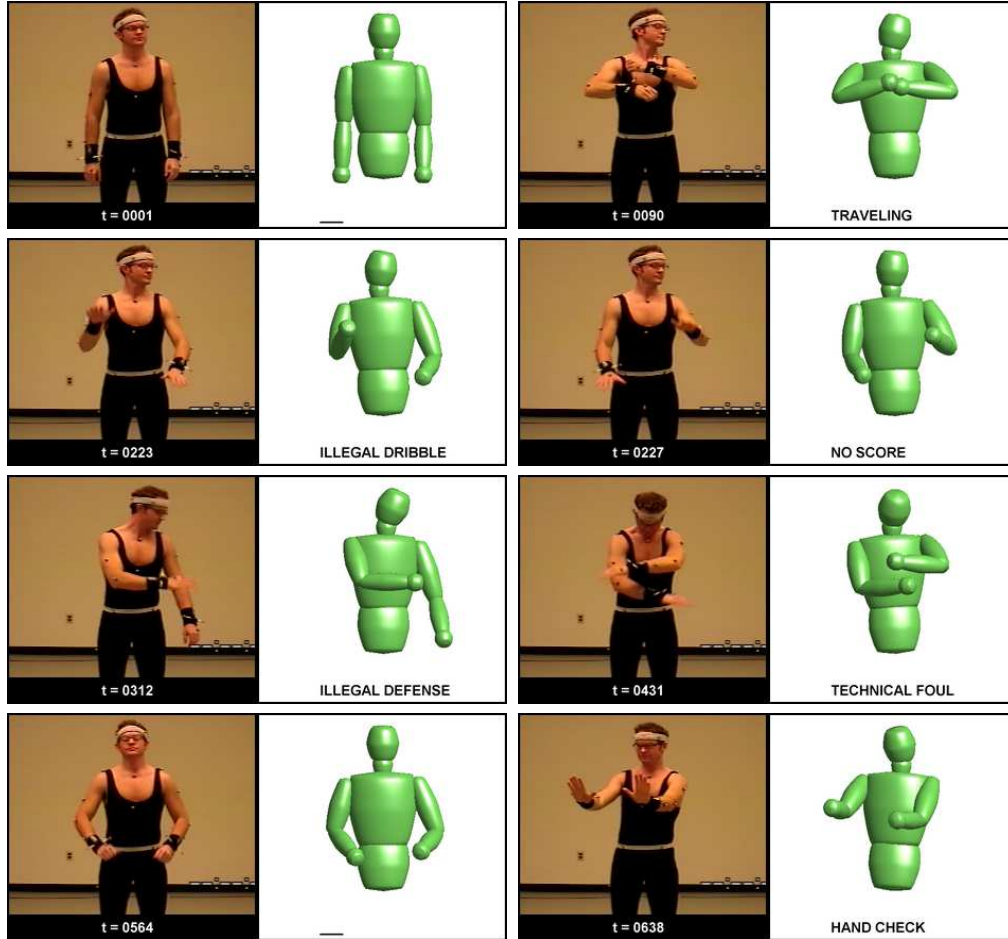
Figure 5.10: *Tracking 3D pose and recognizing gesture (from a predefined set of gestures) on a test sequence, using a mixture of regressors. Each state is associated with the gesture label corresponding to the class with the maximum posterior conditional likelihood at that point in time.*
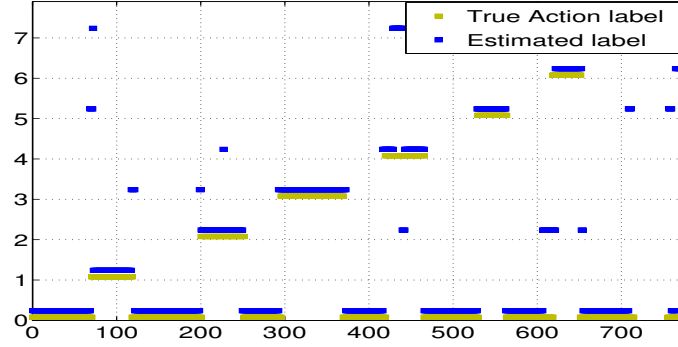
Figure 5.11: *A comparison of the estimated gesture labels with hand-labeled ground truth for the sequence shown in fig 5.10. The numbers 1 - 7 on the y axis correspond to the seven gestures shown in fig 5.9, with the output 0 corresponding to a neutral pose. The predicted labels are mostly the true ones, with errors occurring mostly at class transitions.*

final components obtained after EM thus contain probabilistic information of the gesture label associated with each data point.

On a test sequence of similar gestures, we now assign a gesture label k^* to each reconstructed pose by reading off the label of the component with maximum posterior probability:

$$k_t^* = \arg \max_k \{ p(\mathbf{l} = k \,|\, \mathbf{z}_t) \} \tag{5.9}$$

where $p(\mathbf{l} = k \,|\, \mathbf{z})$ is computed from (5.8). Figure 5.10 shows snapshots from a test video sequence where each reconstructed pose is labeled with its most likely gesture. No explicit smoothing is applied to this label across neighbouring images, but the condensation based tracking helps to ensure that the predicted gesture label is reasonably consistent with time. Except for a few cases of confusion (*e.g.* at $t = 227, 431$ where the algorithm outputs a wrong gesture label), the method recognizes gestures with a very high level of accuracy. A plot of the estimated gesture labels compared with ground truth for the complete sequence is shown in figure 5.11. We see that most of the errors actually occur at class transitions. One of the reasons of this is likely to be the fact that we simply take the maximum of the posterior probability — a more reasonable scheme, *e.g.* combining the probability values with an HHM over gesture transitions, may help in reducing these.

5.7   Discussion

We have developed a method for multiple hypothesis estimation of 3D human pose from silhouettes, based on mixtures of regressors. The mixture is learned as a generative model on the combined density of the input and output spaces and the regressors are estimated within an EM framework by constraining the covariance matrix to take a special structured form. Accurate pose reconstruction results that correctly identify the ambiguities are obtained on a variety of real unseen silhouettes, demonstrating the method's ability to generalize across inter-person variations and imperfect silhouette extraction. When used in a multiple hypothesis tracker, the method is capable of tracking stably over time with robustness to occasional tracking failures.

In contrast to the previous two chapters, the regression framework developed here is probabilistic and more robust, but compromises on the amount of computation required at runtime. While the regressors themselves are linear and fast to apply, projecting a silhouette to the KPCA-reduced manifold requires computing a kernel function based at each of the training points, thus losing the advantage of the sparse solutions obtained by the RVM in chapters 3 and 4. A possible way around this would be to compute a sparse approximation of the embedding and apply suitable priors on the regressor parameters to learn a sparse mixture of regressors. Another possibility is to extend the cost function of the Relevance Vector Machine to incorporate the latent variable within it and directly deal with multimodal output solutions that are computed sparsely. However, we leave these possibilities for future work and focus attention on using the regression models developed so far in cases where the images cannot be represented using silhouettes, *e.g.* in cases of unknown or cluttered backgrounds. The next chapter develops an image representation that allows for regression-based pose recognition in the presence of clutter.

6

Estimating Pose in Cluttered Images

6.1 Introduction

In a general scene, it is often not known apriori what the background consists of. Obtaining a human body silhouette representing the shape of the foreground may not be straight-forward in such a case. So reconstructing the pose of a person without segmentation information becomes a significantly harder problem — but on the other hand, it is not evident that a precise segmentation is actually needed to perform (pose) recognition. Several existing pose estimation methods work without any segmentation information and in the presence of background clutter, but most of these adopt a model based approach *i.e.* they rely on a predefined kinematic body model. Top-down methods obtain pose by minimizing the image projection error of an articulated model using techniques such as optimization [142] or by generating a large number of pose hypotheses [87]. So they automatically avoid clutter by only attempting to explain a part of the image that is covered by the hypothesized pose projection. This is an effective approach, but does not account for unexplained portions of the image. Moreover, both these techniques can be quite expensive due to repeated measures of the image likelihood involved. Bottom-up methods, on the other hand, use weak limb-detectors to find human body segments in the image (*e.g.* [120, 94]) and then combine independent detections from several detectors with spatial priors on the relative arrangement of the limbs to infer body pose (*e.g.* [137, 113]). Current bottom-up methods based on monocular images obtain very coarse level pose information that is usually not sufficient for motion capture. In fact, only a few of the existing methods in this category actually attempt to recover 3D pose.

In chapter 3, we introduced a very effective model-free approach that estimates 3D body pose by learning a regression-based mapping from monocular image observations to the space of body poses. The method is completely bottom-up and extends gracefully to incorporate temporal information and support multimodal estimates, as we have seen in chapters 4 and 5. Now in all these chapters, we have made use of a robust shape descriptor to encode the input image by segmenting out the human figure to obtain a silhouette. However, the regression framework itself remains a valid approach to inferring pose from any suitable representation of the input image. In the absence of a segmented shape, the regressor could be allowed to cue on other features in an image.

In this chapter we extend the regression based approach to work on general images containing cluttered backgrounds. This calls for an appropriate encoding of image features. Unlike in the case of top-down methods that need to explain only a part of the image that is covered by a projection of a body model for a particular pose hypothesis, a bottom up method requires either the ability to explicitly 'detect' body parts in an image, or otherwise a representation of the image

Figure 6.1: *The presence of background clutter in images makes the problem of pose estimation significantly harder. An important issue is to cue on useful parts of the image while not being confused by objects in the background.*

in a manner that would allow a learning algorithm to implicitly cue on relevant features that encode pose information while being robust to irrelevant clutter.

6.1.1 Overview of the Approach

We base our method on the model-free approach developed in the previous chapters and use a large collection of pose-labeled images (from motion capture) to learn a system that directly predicts a set of body pose parameters from image descriptors. We side-step the problem of detecting people in a scene and focus on extracting pose from image windows that are known to contain people. To encode the input, local gradient orientation histograms such as those in the underlying descriptor of the SIFT [90] are computed over a dense grid of patches on the image window to give a large vector $\mathbf{z}$ of concatenated descriptors. This is followed by a Non-negative Matrix Factorization step that learns a set of sparse bases for the descriptors at each of the grid locations. The basis vectors correspond to local features on the human body and allow the patches to be re-encoded to selectively retain human-like features (that occur frequently and consistently in the data) while suppressing background clutter (which is highly varied and inconsistent). This gives a representation $\phi(\mathbf{z})$ of the image which is reasonably invariant to changes in background. Pose is then recovered by direct regression, $\mathbf{x} = \mathbf{A}\,\phi(\mathbf{z}) + \boldsymbol{\epsilon}$. The complete sequence of steps involved is summarized in figure 6.2.

6.2 Image representation

Developing an effective image representation is a well-known problem in several areas of computer vision. Simple applications often make use of global representations such as colour histograms, but demanding tasks such as reconstructing pose or recognizing objects require more robust representations. A powerful way of achieving robustness is by combining the information contained in many *local* regions of an image in order to summarize its complete contents. Effective human pose estimation — as in many cases of object recognition — now relies on the ability of a method to key on only a subset of the constituent regions in an image and successfully identify the remaining image as being 'irrelevant' to the problem[1].

Limb detectors. Perhaps the most intuitive way to think of encoding an image of a person is using a collection of body limbs. These are natural constituent parts of the human body and its pose

---

[1]Note that in some situations, an understanding of the background content may actually be useful for providing *contextual information*, but accommodating for this is out of the scope of this thesis.

(a) (b) (c) (d) (e)
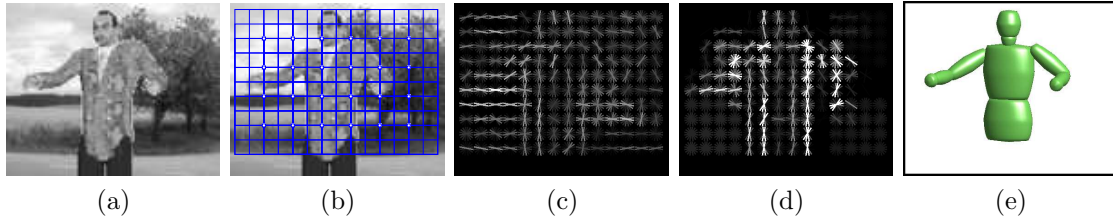
Figure 6.2: *An overview of our method of pose estimation from cluttered images. (a) original image, (b) a grid of fixed points where the descriptors are computed (each descriptor block covers an array of 4x4 cells, giving a 50% overlap with it's neighbouring blocks), (c) SIFT descriptors computed at these points, the intensity of each line representing the weight of the corresponding orientation bin in that cell, (d) Suppressing background using a sparse set of learned (NMF) bases encoding human-like parts, (e) final pose obtained by regression*

directly depends on the relative configuration of these parts. So trying to deduce pose from an image given the location and orientation of each body part or limb is one possible approach, even though finer details of limb shape and appearance are extremely important in many cases. Detecting the presence of these limbs in an image, however, is a hard problem involving learning a general appearance model for each body limb and is still an active area of research [120, 94, 137, 113]. The problem remains very difficult due to large appearance variations caused because of clothing, lighting and occlusions (among other factors) and most state of the art human part detectors have to be supplemented with a human body model to either constrain the search or refine the detections by incorporating spatial priors and inter-part interactions. In a model-free approach like ours, we prefer to explore lower level image features that might directly be used to predict pose.

Local patches. A very effective and widely used representation for images, a collection of image patches[2] can be used to include much more information than the locations and orientations of different limbs. Appropriately located patches at the right scale could encode the appearance of human body parts such as elbow joints, shoulder contours and possibly the outlines of parts like the head and hands — and all of these contain important information concerning pose. In many problems, the contents of an image have been well-summarized by describing the properties of a small set of patches centered at salient points of interest on the image [54, 70]. Several object recognition and scene classification methods, for instance, key on corners and blob-like regions for this purpose. Recently it has been shown that computing patch descriptors densely over an entire image rather than sparsely at these interest points actually provides a more powerful encoding (*e.g.* [69, 32]). The key to successfully taking advantage of such a representation, however, is to develop a learning method that would automatically identify the 'salient' patches (or features) from amongst this dense set. We usually make use of a large collection of labeled training data for this purpose.

6.2.1 Dense patches

Densely sampling patches from an image — for instance, every few pixels — will, in general, give quite a large number of patches on an image. At the first thought, this may seem to be a redundant representation, especially if these patches overlap (as is often the case). However, using such overlapping patches and robustly encoding the information in each patch with an appropriate descriptor has recently been shown to be an effective strategy .

---

[2]Representative patches are often called 'textons' from their original use in the texture literature.
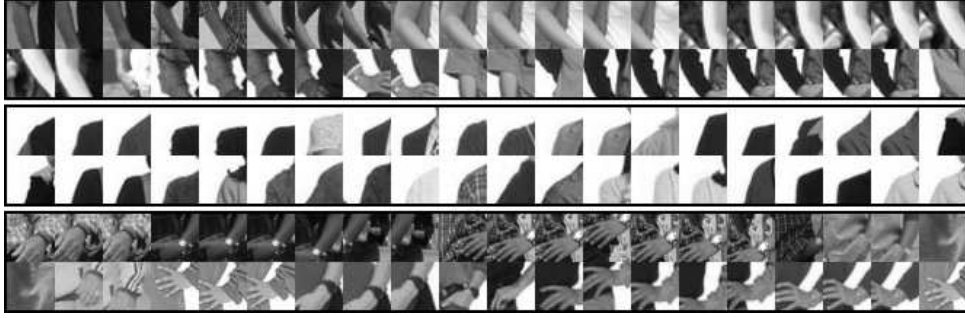
Figure 6.3: *Sample clusters obtained by a k-means clustering on patches represented as their 128D SIFT descriptors appended with suitably scaled image coordinates. Each cluster includes patches with similar appearance and pose information in a localized image region, but using the centers of several such clusters to encode human images as a 'bag of features' is found to perform poorly with respect to encoding 3D pose information.*

Patch information can be encoded in many different ways. Given the variability of clothing and the fact that we want to be able to use black and white images, we do not use colour information. To allow the method to key on important body contours, we base our representation on local image gradients and use histograms of gradient orientations for effective encoding. The underlying descriptor of the SIFT [90] proves to be a useful representation in this regard as it quantizes gradient orientations into discrete values in small spatial cells and normalizes these distributions over local blocks of cells to achieve insensitivity to illumination changes. The relative coarseness of the spatial coding provides some robustness to small position variations, while still capturing the essential spatial position and limb orientation information. Note that owing to loose clothing, the *positions* of limb contours do not in any case have a very precise relation to the pose, whereas *orientation* of body edges is a much more reliable cue. We thus use SIFT-like histograms to obtaining a 128D feature vector for each patch[3].

To retain the information about the image location of each patch that is indispensable for pose estimation, the descriptors are computed at fixed grid locations in the image window. This gives an array of 128D feature vectors for the image. Figure 6.2(c) shows the features extracted from a sample image where the value in each orientation bin in each cell is represented by the intensity of the line drawn in that cell at the corresponding orientation. (Overlapping cells contribute to the intensities of the same set of lines, but these are normalized accordingly.) We denote the descriptor vectors at each of these L locations on the grid as $\mathbf{v}^l, l \in \{1 \ldots L\}$, and simply raster-scan the array to represent the complete image as a large vector $\mathbf{z}$, a concatenation of the individual descriptors:

$$\mathbf{z} \equiv (\mathbf{v}^{1^\top}, \mathbf{v}^{2^\top}, \ldots \mathbf{v}^{L^\top})^\top \tag{6.1}$$

As an alternative to maintaining a large vector of the descriptor values, we also study a *bag of features* [31] style of image encoding. This is a common scheme in object recognition that involves identifying a representative set of parts (features) as a vocabulary (generally obtained by clustering the set of patch descriptors using k-means or some other similar algorithm), and then representing each image in terms of the statistics of the occurrence of each vocabulary part in that image. In an analogous manner, a human body image can be represented as a collection of representative

---

[3]Note that other similar descriptors could also possibly be useful for this purpose, *e.g.* the generalized shape context [99]. However, we have not tried these in this work.

patches encoding limb sections and other key body parts. Unlike the standard bag of features that completely ignores spatial information, however, we find that spatial information is absolutely critical for inferring body pose from an image. We thus include the image coordinates as two extra dimensions[4] in the patch descriptors while clustering to obtain the part-dictionary. Samples of clusters obtained in this fashion are shown in figure 6.3. In our experiments, however, quantizing patches using these centers proves to be incapable of capturing sufficient information to successfully regress pose.

For a comparison with our NMF based encoding (described below), we also independently cluster patches at each of the L locations on the images to identify representative configurations of the body parts that are seen in these locations. Each image patch is then represented by softly vector quantizing the SIFT descriptor by voting into each of its corresponding k-means centers, *i.e.* as a sparse vector of similarity weights computed from each cluster center using a Gaussian kernel. Again, such a representation gives poor predictions of pose. Below we describe an alternate encoding that proves to be much more effective. Results from the different representations are presented in § 6.4.

6.3 Building tolerance to clutter

The dense representation provides a rich set of features that are robust to lighting, slight positional variations and also compactly encode the contents of each patch. However, features from both the foreground (from which pose is to be estimated) and the background (that is assumed not to contain any useful pose information) are represented in a similar manner. Ideally a single learning algorithm would learn to key on the relevant features while not being confused by the clutter in the background to successfully recover 3D pose from such as incompletely specified representation *e.g.* we have seen that the Relevance Vector Machine, to some extent, is capable of achieving this in the form of implicit feature selection (§ 3.5.1). Here, we do this in a separate phase by re-encoding the patches to explicitly remove irrelevant components from their descriptor vectors. We find that given a training set of foreground-background labeled images, Non-negative Matrix Factorization can be usefully exploited for this purpose.

6.3.1 Non-negative Matrix Factorization

Non-negative matrix factorization (NMF) is a recent method that has been used to exploit latent structure in data to find part based representations [85, 59]. NMF factorizes a non-negative data matrix $\mathbf{V}$ as a product of two lower-rank matrices $\mathbf{W}$ and $\mathbf{H}$, both of which are constrained to be non-negative:

$$\mathbf{V}_{d \times n} \approx \mathbf{W}_{d \times p} \mathbf{H}_{p \times n} \qquad p \leq d, n \quad \mathbf{V}_{ij}, \mathbf{W}_{ij}, \mathbf{H}_{ij} \geq 0 \qquad (6.2)$$

If the columns of $\mathbf{V}$ consist of feature vectors, $\mathbf{W}$ can be interpreted as a set of basis vectors, and $\mathbf{H}$ as corresponding coefficients needed to reconstruct the original data. Each entry of $\mathbf{V}$ is thus represented as $\mathbf{v}_i = \sum_j \mathbf{w}_j h_{ji}$. Unlike other linear decompositions such as PCA or ICA [158], this purely additive representation (there is no subtraction) tends to pull out local fragments that occur consistently in the data, giving a sparse set of basis vectors.

---

[4]These two extra dimensions must usually be appropriately centred and scaled by some weight to balance their effect with respect to the original descriptor. A suitable weight is generally obtained by empirically trying different values.
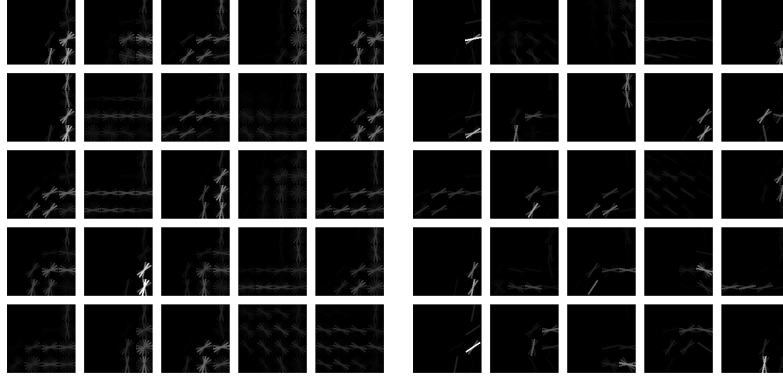
Figure 6.4: *Exemplars, or basis vectors, extracted from SIFT descriptors over 4000 image patches located close to the right shoulder. The corresponding block is shown in figure 6.5. (Left) Representative examples selected by k-means. (Right) Much sparser basis vectors obtained by non-negative matrix factorization. These capture important contours encoding a shoulder, unlike the denser examples given by k-means.*

In an attempt to identify the meaningful components of the descriptors at each patch location, we collect all the descriptors $\mathbf{v}_i^l$ from a given location l in the training set into a matrix $\mathbf{V}^l$ and decompose it using NMF. The results of applying NMF to the 128D descriptor space at a given patch location are shown in figure 6.4. Besides capturing the local edges representative of human contours, the NMF bases allow us to compactly code each 128D SIFT descriptor directly by its corresponding vector of basis coefficients, denoted here by $\mathbf{h}(\mathbf{v}^l)$, giving a significant reduction in dimensionality. This serves as a nonlinear image coding that retains good locality for each patch, and the image is now represented by concatenating the coefficient vectors for the descriptors at all grid locations:

$$\phi(\mathbf{z}) \equiv (\mathbf{h}(\mathbf{v}^1)^\top, \mathbf{h}(\mathbf{v}^2)^\top, \dots \mathbf{h}(\mathbf{v}^L)^\top)^\top \tag{6.3}$$

Having once estimated the basis $\mathbf{W}$ (for each image location) from a training set, we keep it fixed when we compute the coefficients for test images. In our case, we find that the performance tends to saturate at about 30-40 basis elements per grid patch.

An interesting advantage of using NMF to represent image patches is its ability to selectively encode the components of a descriptor that are contributed by the foreground, hence effectively rejecting background. We find that by learning the bases $\mathbf{W}$ from a set of clean images (containing no background clutter), and using these only additively (with NMF) to reconstruct images with clutter, only the edge features corresponding to the foreground are reconstructed, while suppressing features in unexpected parts of the image. This happens because constructing the bases from a large number of background-free images of people forces them to consist of components of the descriptors corresponding to consistently occurring human parts. Now when used to reconstruct patches from cluttered images, these basis elements can add up to, at best, reconstruct the foreground components. Some examples illustrating this phenomenon are shown in figure 6.5.
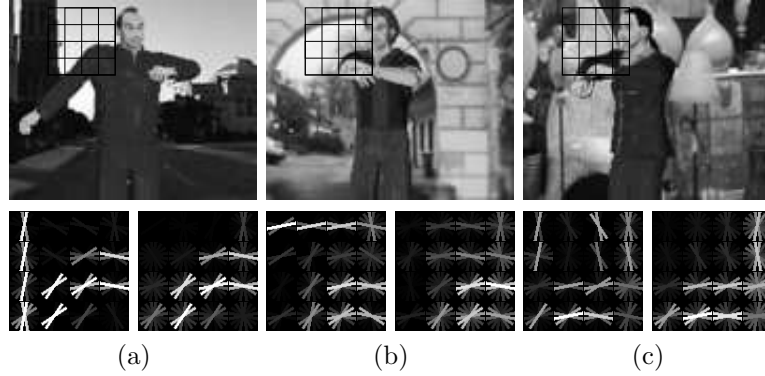
Figure 6.5: *To selectively encode foreground features and suppress unwanted background, we use NMF bases learned on clean images (with no clutter) to reconstruct the cluttered image patches. For each image, the original SIFT feature and its representation using the bases extracted using NMF are shown for the patch marked. Features corresponding to background edges such as those of the building on the left in (a) and the arch in (b) are clearly suppressed, while background clutter in (c) is down-weighted.*

6.4 Experimental Performance

This section tests the effectiveness of the image encoding schemes discussed above in the context of recovering 3D human body pose from images. To evaluate reconstruction performance independently of the issue of ambiguities (see chapter 5), we restrict ourselves to frontal body poses, taking upper body arm gestures as a test case.

In these experiments, (upper) body pose is represented by a 24D vector $\mathbf{x}$, encoding the 3D locations of 8 key upper body joint centres. We use methods described in chapter 3 to regress this pose vector on the input representation $\phi(\mathbf{z})$ as given by (6.3):

$$\mathbf{x} \;=\; \mathbf{A}\,\phi(\mathbf{z}) + \boldsymbol{\epsilon} \tag{6.4}$$

where $\mathbf{A}$ is a matrix of weight vectors to be estimated, and $\boldsymbol{\epsilon}$ is a residual error vector. Errors are reported as the average deviation, in centimeters, of each body joint for a human body of average adult size.

For descriptor computation, we quantized gradient orientations into 8 orientation bins (in $[0, \pi]$) in 4×4 spatial cells, as described in [90], using blocks 32 pixels across. Our images are centered and re-sized to 118×95 pixels. The descriptor histograms are computed on a 4×6 grid of 24 uniformly spaced overlapping blocks on each image, giving rise to 3072D image descriptor vectors $\mathbf{x}$. NMF is performed using available software [59] to reduce each of the 128D descriptors to 30D vectors of basis coefficients. So we finally have a 720D feature vector $\phi(\mathbf{z})$ for each image.

We train and evaluate the methods on two different databases of human pose examples. The first is a set of randomly generated human poses[5] that are rendered using the human model rendering package POSER. The second dataset[6] contains motion capture data from human recordings of

---

[5]This dataset was introduced in [132]

[6]This dataset was obtained from http://mocap.cs.cmu.edu. Since the two datasets use different motion capture formats, we use a compatible set of 3D joint coordinates and not joint angles for representing pose for these experiments. *c.f.* appendix B.
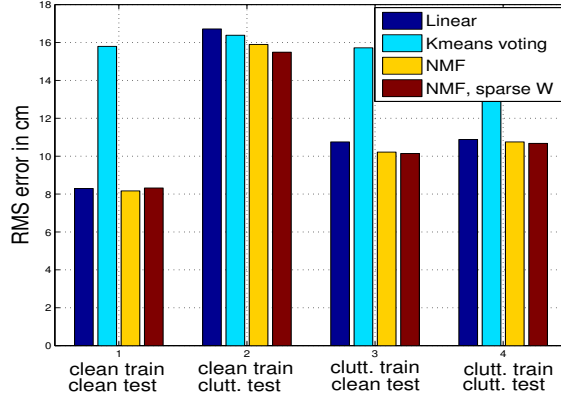
Figure 6.6: *A comparison of the performance of different feature encodings in regressing 3D pose, over different combinations of training and testing on clean and cluttered data. See text.*

several sets of arm movements along with corresponding real images. In order to analyze the effects of cluttered backgrounds and the robustness of the image encoding, two versions of each of the image sets are prepared — a *clean* version with a simple or no background, and a *cluttered* version consisting of randomly added backgrounds to the clean images.

6.4.1   Effect of Image encoding

Figure 6.6 shows the performance of different feature encodings over all combinations of training and testing on clean and cluttered images. The regularization parameter of the regressor was optimized using cross validation. These figures are reported for 4000 training and 1000 test points from the Poser dataset. The errors reported indicate, in centimeters, the average RMS deviations for the 3D locations of 8 points located at the shoulders, elbows, wrists, neck and pelvis. The best performance, as expected, is obtained by training and testing on clean, background-free images, irrespective of the descriptor encoding used. Training on clean images does not suffice for generalization to clutter. Using cluttered images for training provides reasonably good generalization to unseen backgrounds, but the resulting errors are larger by 2-3 cms on both clean and cluttered test sets than the best case. Surprisingly, a linear regressor on the vector $\mathbf{x}$ performs very well despite the clutter. An examination of the elements of the weight matrix $\mathbf{A}$ reveals this is due to automatic down-weighting of descriptor elements that usually contain only background — an implicit feature selection by the regressor. On average, the k-means based representation performs the worst of all and the NMF-based representation gives the best performance. To study the space of encodings 'between' an extreme exemplar based k-means representation and the set of basis vectors obtained by NMF, we tested NMF with constraints on the sparsity level of the basis vectors and coefficients [59]. Varying the sparsity of the basis vectors $\mathbf{W}$ has very little effect on the performance, while varying the sparsity of the coefficients $\mathbf{H}$ gives results spanning the range of performances from k-means to unconstrained NMF. As the sparsity prior on $\mathbf{H}$ is increased to a maximum, NMF is forced to use only a few basis vectors for each training example, in the extreme case giving a solution very similar to k-means.

6.4.2   Pose reconstruction results

Figure 6.7 shows some examples of pose estimation on the cluttered test set from the experiment described above. The reconstructions are visually quite appealing, though there are instances of
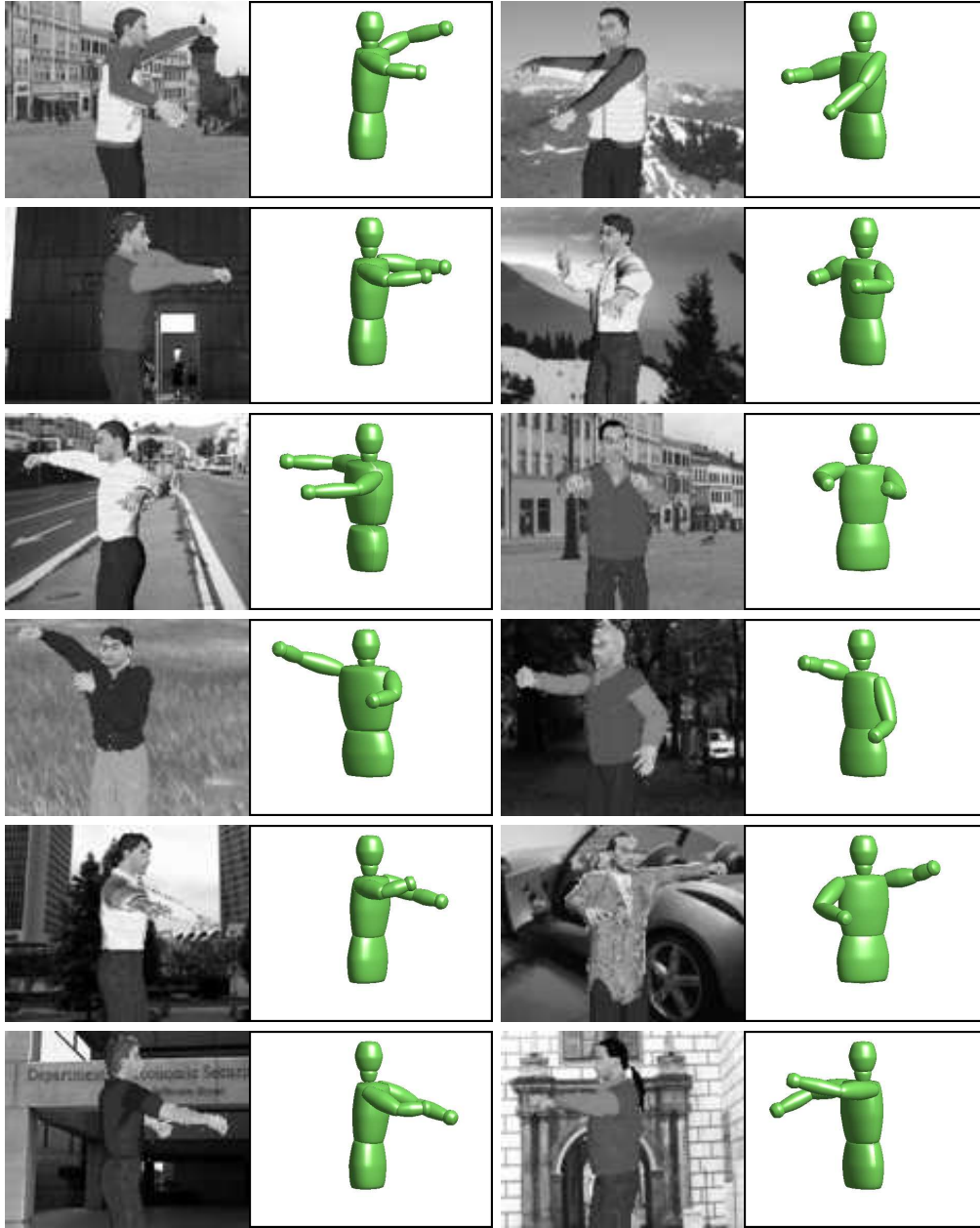
Figure 6.7: *Sample 3D pose estimates from a test set of 1000 images containing cluttered backgrounds. No knowledge of segmentation is used in the process. The images are encoded using sparse NMF bases computed from a dense grid of SIFT descriptors on the image and pose is obtained by regressing the coordinates of body joints on this representation.*

Figure 6.8: *Pose reconstructions on real unseen images. The images on the left are taken from a test sequence in our motion capture dataset which includes similar gestures made by another person, while the ones on the right are example images obtained using Google image search.*

mis-estimation and confusion between limb labels *e.g.* in the bottom left example, the two arms are interchanged. To see the effect of depth ambiguities on these results, we computed errors separately in the x and y coordinates corresponding to the image plane and z, corresponding to depth. We find that errors in depth estimation are a little higher than those in lateral displacement. *E.g.*, of the 10.88 cm of error obtained in the experiment on cluttered images, 9.65 cm comes from x and y, while 12.97 cm from errors in z. In the absence of clutter, we obtain errors of $\sim$8 cm. This is similar to the performance reported in [132] on this dataset (when transformed into the angle based error measure used in that paper), showing that regression based methods can match the performance of nearest-neighbourhood based ones, while avoiding having to store and search through excessive amounts of training data.

For our second set of experiments, we use $\sim$1600 images from 9 video sequences of motion capture data. Performance on a test set of 300 images from a 10th sequence gives an error of 7.4 cm in the presence of clutter. We attribute this slightly improved performance to the similarity of the gestures performed in the test set to those in the training sequences, although we emphasize that in the test set they were performed by a different subject. Figure 6.8 shows sample reconstructions over test examples from the second database and from some natural images obtained using Google image search. We find that training on the second dataset also gives qualitatively better performance on a set of randomly selected real images. This suggests that it is important to include more 'natural', human-like poses in the training set, which are not covered by randomly sampling over the space of possible poses. Notice that although the results on the real images are not very accurate, they do capture the general appearance of the subject's gestures fairly well. In figure 6.9, we overlay the reconstructions on the original images for a test set with a new set of backgrounds. This reveals that although the poses are reconstructed very well, the arm estimates often do not lock onto the corresponding edges in the image and this is responsible for a large part of the error. Part of

Figure 6.9: *Results of the 3D pose estimation in the presence of different backgrounds. Overlaying the estimated pose over the original images shows that a large portion of the error is due to the fact that the arms do not precisely lock onto the edges in the image. See text for a discussion on this.*

the reason for this is the coarse encoding of edge locations within the SIFT descriptors[7]. While loosely encoding positional information allows the descriptors to generalize from several examples and be more robust, missing the fine details could lead to a loss of precision. Better adapted descriptors based on a finer coding may be useful in reducing this effect. However, unlike model based methods which can explicitly force these edges to match with those of the human body model, no such information is used in the model-free method developed here. Slightly less precision for a considerable gain with respect to computation time is one of the trade-offs involved in the use of model-free approaches *vs.* model based ones. Some perspectives on a possible combination of the two approaches are given in chapter 9.

6.5 Discussion

This chapter presents an approach for completely bottom-up human pose recovery from cluttered images. Given an image window roughly centered at a person, the method uses a robust, background tolerant image encoding to obtain 3D pose by regression. A novel application of Nonnegative Matrix Factorization is demonstrated for this purpose: its ability to selectively encode information (from learning on labeled data) can be used to 'project' cluttered image features onto a space of more relevant features that encode human appearance. This is likely to prove useful in other applications including segmentation and recognition.

The approach developed here directly links image based human motion understanding and motion capture with the broader problem of image representations and encodings, by trying to understand

---

[7]Another source of this error is the fact that the experimental setup is not perfect — calibration error between the optical and motion capture cameras used to record training images is not accounted for.

what features are suitable for encoding human pose information. Given that this is a largely unexplored area, there certainly remains plenty of scope for further work. The method, as it is, can be extended to include descriptors computed at multiple scales, which will code more information (*e.g.* multi-scale edge histograms are used in [132]). A more basic question, however, is regarding the use of descriptors that are tied to the reference frame of the image window being processed[8]. This is certainly very practical given that a person detector could be used to locate such a window. Nevertheless, it does involve going via a two-step process, detection followed by pose estimation. A more 'elegant' representation would be robust to the scale and location of this detection window.

We have found that the bag-of-features representation has not been very effective here. A possibility is to build richer forms of such a representation that might loosely encode geometry *e.g.* via the modeling of spatial relationships between the parts. Probabilistic methods that build on NMF to extract latent semantics from data have also been used for image encoding recently (*e.g.* [42, 45, 21]). These may prove helpful for human pose estimation. We shall see more on image encoding in chapter 8 which presents a multilevel local encoding method, 'hyperfeatures', for a generic and robust image representation. Meanwhile, in the next chapter, we describe some work on tracking human body pose in cluttered images.

---

[8]Recall that the silhouette based representation used in the earlier chapters, on the other hand, is completely invariant to scale and translation.

7

Modeling Dynamics using Mixtures

7.1 Introduction

Human motion is highly nonlinear and time-varying. Tracking this in video sequences containing cluttered scenes is a challenging problem. In this chapter we focus on modeling the dynamics of human motion in order to use knowledge of different motion patterns during the tracking process. The main issue that we address is allowing for transitions between different kinds of motions.

We describe a mixture-based approach to the problem that applies statistical methods to example motion trajectories in order to capture characteristic patterns that exist in typical human motion. The method developed here is based on learning a collection of local motion models by automatically partitioning the parameter space into regions with similar dynamical characteristics. Each motion model is a Gaussian autoregressive process and the learning takes place as a probabilistic mixture where each training vector has a finite responsibility towards each motion model. To exploit the correlations between different body part movements during various activities and to stabilize the otherwise high-dimensional estimation problem, each dynamical model in learned on a low dimensional manifold. Transitions between these models occur probabilistically based on their support regions in body pose space.

Unlike all the other chapters of this thesis the work in this chapter makes use of a top down methodology. A planar human body is registered onto the images by matching appearance templates for each body part and the human body configuration is tracked in the image plane without performing motion capture. We find that besides providing accurate motion priors for different activities, using multiple motion models also helps in tracking through changing camera viewpoint in this case as the dynamics are dependent on viewpoint.

Most multi-class models of dynamics use discrete Markov processes (often HMMs) to describe the switching dynamics for transition between classes (*e.g.* [103, 109]). This decouples the switching dynamics from the continuous dynamics, which is somewhat artificial. In the method described below, we propose a smoother switching scheme based on the continuous state of the system that ensures a smooth motion between class transitions.

7.1.1 Overview of the Approach

The mixture of dynamical models is built from a set of hand-labeled training sequences as follows: *(i)* the state vectors representing human body configurations in the image plane are clustered using
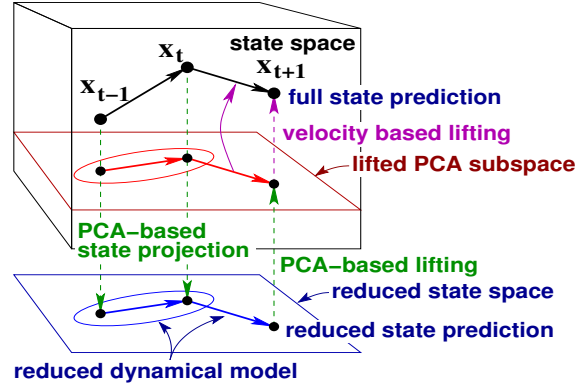
Figure 7.1: *Using a reduced dynamical model to predict states in a high-dimensional space. A given state is projected onto a low-dimensional space using PCA, within which a linear autoregressive progress is used to predict a current (reduced) state. This is then lifted back into full state space to estimate a noise model in the high-dimensional space. To prevent the state from being continually squashed into the PCA subspace, we lift the velocity prediction and not the state prediction.*

k-means and projected to lower dimensional subspaces using PCA to stabilize the subsequent estimation process; *(ii)* a local linear autoregression for the state given the p previous reduced states is learned for each cluster ($p = 1,2$ in practice); *(iii)* the data is re-clustered using a criterion that takes into account the accuracy of the local model for the given point, as well as the spatial contiguity of points in each model; *(iv)* the models are refitted to the new clusters, and the process is iterated to convergence.

The tracking framework adopted in this chapter is similar to Covariance Scaled Sampling [140]. At each time step, random samples are drawn from the dynamical prior using an appropriately scaled covariance matrix. Each sample is then locally optimized by maximizing its image likelihood.

7.2   Modeling the Local Dynamics

Despite the complexity of human dynamics, the local motion within each region is usually well described by a linear Auto-Regressive Process (ARP):

$$\mathbf{x}_t = \sum_{i=1}^{p} \mathbf{A}_i \, \mathbf{x}_{t-i} + \mathbf{w}_t + \mathbf{v}_t \tag{7.1}$$

Here, $\mathbf{x}_t \in \mathbb{R}^m$ is the pose vector[1] at time t, p is the model order (number of previous states used). $\mathbf{A}_i$ are $m \times m$ matrices giving the influence of $\mathbf{x}_{t-i}$ on $\mathbf{x}_t$, $\mathbf{w}_t \in \mathbb{R}^m$ is a drift/offset term, and $\mathbf{v}_t$ is a random noise vector (here assumed white and Gaussian, $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$).

The choice of the ARP order is strongly dependent on the nature of the motions exhibited by the system[2]. In practice, experiments on different kinds of motion show that a second order ARP usually suffices for human tracking:

---

[1]A typical representation of human pose could involve anywhere between 30 and 60 parameters, so $\mathbf{x}$ is normally a high dimensional quantity consisting of joint angles and possibly limb dimensions.

[2]A possible automatic selection procedure for determining the number of previous states required in the ARP could be implemented using the Relevance Vector Machine.

$$\mathbf{x}_t = \mathbf{A}_1\,\mathbf{x}_{t-1} + \mathbf{A}_2\,\mathbf{x}_{t-2} + \mathbf{v}_t \tag{7.2}$$

This models the local motion as a mass-spring system (set of coupled damped harmonic oscillators) and can also be written in differential form:

$$\ddot{\mathbf{x}}_t = \mathbf{B}_1\,\dot{\mathbf{x}}_t + \mathbf{B}_2\,\mathbf{x}_t + \mathbf{v}_t \tag{7.3}$$

There are standard ways of learning ARP models from training data [83], however the high dimensionality of our pose vector $\mathbf{x}$ prohibits the use of these techniques without requiring excessive amounts of training data. (Recall that a separate dynamical model must be learned for each of the local regions.) Given that human motion usually consists of fairly correlated body movements, we choose to learn the dynamics with respect to a reduced set of degrees of freedom. The trajectories are thus locally projected into a lower dimensional subspace within each class using PCA before learning the dynamics. The ARP model learned in this reduced space is then "lifted" to the full state space using the PCA injection and the resulting model is cross-validated to choose the PCA dimension (about 5 in practice). The scheme is illustrated in figure 7.1, and the complete reduction and lifting algorithm for a given local region (class) is described below:

1. Reduce the vectors in the class to a lower dimensional space by:

    (a) Centering them and assembling them into a matrix (by columns):
    $\mathbf{X} = [\ (\mathbf{x}_{p_1} - \mathbf{c})\quad (\mathbf{x}_{p_2} - \mathbf{c})\quad \cdots\quad (\mathbf{x}_{p_m} - \mathbf{c})\ ]$, where $p_1 \ldots p_m$ are the indices of the points in the class and $\mathbf{c}$ is the class mean.

    (b) Performing a Singular Value Decomposition of the matrix to project out the dominant directions: $\mathbf{X} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^T$.

    (c) Projecting each vector into the dominant subspace: each $\mathbf{x}_i \in \mathbb{R}^m$ is represented as a reduced vector $\mathbf{q}_i = \tilde{\mathbf{U}}^T(\mathbf{x}_i - \mathbf{c})$ in $\mathbb{R}^{m'}$ ($m' < m$), where $\tilde{\mathbf{U}}$ is the matrix consisting of the first m' columns of $\mathbf{U}$.

2. Build an autoregressive model, $\hat{\mathbf{q}} = \sum_{i=1}^{p} \mathbf{A}_i\,\mathbf{q}_{t-i}$, and estimate $\mathbf{A}_i$ by writing this in the form of a linear regression:

$$\mathbf{q}_t = \tilde{\mathbf{A}}\,\tilde{\mathbf{q}}_{t-1}, \quad t = t_{p_1}, t_{p_2}, \ldots t_{p_n} \tag{7.4}$$

    where

$$\tilde{\mathbf{A}} = (\ \mathbf{A}_1\quad \mathbf{A}_2\quad \cdots\quad \mathbf{A}_p\ ), \quad \tilde{\mathbf{q}}_{t-1} = \begin{pmatrix} \mathbf{q}_{t-1} \\ \mathbf{q}_{t-2} \\ \vdots \\ \mathbf{q}_{t-p} \end{pmatrix}$$

3. Estimate the error covariance $\mathbf{Q}$ from the residual between $\{\hat{\mathbf{x}}_i\}$ and $\{\mathbf{x}_i\}$ by lifting $\hat{\mathbf{q}}_t$ back into m dimensions:

$$\hat{\mathbf{x}}_t = \mathbf{c} + \tilde{\mathbf{U}}\hat{\mathbf{q}}_t \tag{7.5}$$

7.3 Global Estimation with EM

Learning the complete model involves estimating the ARP parameters $\{\mathbf{A}_1^k, \mathbf{A}_2^k, \ldots \mathbf{A}_p^k, \mathbf{Q}^k\}$ for each class $k = 1 \ldots K$ and the extents of the class regions. An initial set of classes is obtained

by clustering on the unstructured collection of pose vectors $\mathbf{x}_i$, using k-means on Mahalanobis distances. The class regions formed are found to more-or-less cut the state trajectories into short sections, all sections in a given partition having similar dynamics. Parameters of the dynamical model within each class are estimated as described above in § 7.2.

The k-means based partitions are then revised by assigning training points to the dynamical model that predicts their true motion best, and the dynamical models are re-learned by using the new class memberships to project each region into a PCA subspace . This Expectation-Maximization / relaxation procedure is iterated to convergence. In practice, using the dynamical prediction error as the sole fitting criterion gives erratic results, as models sometimes "capture" quite distant points. So we include a spatial smoothing term by minimizing:

$$\sum_{\text{training points}} (\text{prediction error}) + \gamma \cdot (\text{number of inter-class neighbors})$$

where γ is a relative weighting term, and the number of inter-class neighbors is the number of edges in a neighborhood graph that have their two vertices in different classes (*i.e.* a measure of the lack of contiguity of a class).

7.3.1   Inter-class Transitions

Many example-based trackers use discrete state HMMs (transition probability matrices) to model inter-cluster transitions [152, 149]. This is unavoidable when there is no state space model at all (*e.g.* in exemplars [152]), and it is also effective when modeling time series that are known to be well approximated by a set of piecewise linear regimes [50]. Its use has been extended to multi-class linear dynamical systems exhibiting continuous behaviour [109], but we believe that this is not the best strategy as the discrete transitions ignore the location-within-partition information encoded by the continuous state, which strongly influences inter-class transition probabilities. To work around this, quite small regions have to be used, which breaks up the natural structure of the dynamics and greatly inflates the number of parameters to be learned. In fact, in modeling human motion, the current continuous state already contains a great deal of information about the likely future evolution, and often this alone is rich enough to characterize human motion classes, without the need for the separate hidden discrete state labels of HMM based models.

We thus model the inter-class transitions based on the class membership probability of the dynamical prediction of a point. Given the continuous state of the system, a soft class membership is obtained from the Gaussian mixture model based at the class centres, and the dynamical predictions from all probable classes are linearly combined according to their membership weights. This allows for greater uncertainty in the class label when the continuous state is close to the 'boundary' of a particular class. Figure 7.2 compares the two schemes in graphical form. Modeling the class-label to be conditional on continuous state ensures a smooth flow from one dynamical model class to the next, avoiding erratic jumps between classes.

7.4   Model-based Tracking

The mixture of dynamical models described above is used in a tracking framework to overlay a 33 d.o.f. planar articulated human body model[3] on images from video sequences of human action. A

---

[3]This is a modified Scaled Prismatic Model [101] representation of the human body that encodes the body as a set of 2D chains of articulated limb segments, each represented by rounded trapezoidal image templates defined by their end widths. Body poses are parametrized by vectors of their joint angles and apparent (projected) limb lengths. Details of the parametrization are given in [4].
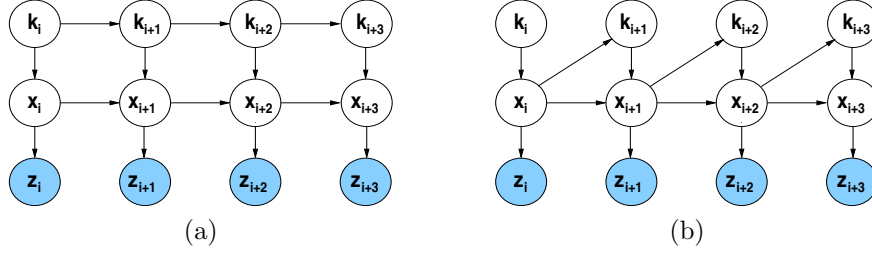
Figure 7.2: *Graphical models for inter-class transitions of a dynamical system. (a) An HMM-like mixed-state model, and (b) our inter-class transition model ($\mathbf{z}_i$: observation, $\mathbf{x}_i$: continuous state, k_i: discrete class). Transitions in an HMM are learned as a fixed transition probability matrix, while our model allows location-sensitive estimation of the class label by exploiting continuous state information.*

great challenge in modeling the dynamics of a planar body model is dealing with aspect change (change in viewpoint of the camera). We find that the EM based learning automatically converges on the different aspects. For instance, in an example where a person walking in one direction turns around and starts walking in the opposite direction, three mixture components are correctly identified and a local dynamical model is learned for each component.

The tracking framework used here is similar to Covariance Scaled Sampling [140]. For each mode of $\mathbf{x}_{t-1}$, the distribution $\mathcal{N}(\hat{\mathbf{x}}_t, \mathbf{Q})$ estimated by the dynamical model (7.1) is sampled, and the image likelihood is locally optimized at each mode. State probabilities are propagated over time using Bayes' rule. The probability of the tracker being in state (pose) $\mathbf{x}_t$ at time t given the sequence of observations $\mathcal{Z}_t = \{\mathbf{z}_t, \mathbf{z}_{t-1} \ldots \mathbf{z}_0\}$ is:

$$p(\mathbf{x}_t \,|\, \mathcal{Z}_t) \;=\; p(\mathbf{x}_t \,|\, \mathbf{z}_t, \mathcal{Z}_{t-1}) \;\propto\; p(\mathbf{z}_t \,|\, \mathbf{x}_t)\, p(\mathbf{x}_t \,|\, \mathcal{Z}_{t-1}) \tag{7.6}$$

where $\mathcal{X}_t$ is the sequence of poses $\{\mathbf{x}_i\}$ up to time t and

$$p(\mathbf{x}_t \,|\, \mathcal{Z}_{t-1}) \;=\; \int p(\mathbf{x}_t \,|\, \mathcal{X}_{t-1})\, p(\mathcal{X}_{t-1} \,|\, \mathcal{Z}_{t-1})\, d\mathcal{X}_{t-1} \tag{7.7}$$

The likelihood $p(\mathbf{z}_t \,|\, \mathbf{x}_t)$ of observing image $\mathbf{z}_t$ given model pose $\mathbf{x}_t$ is computed based on the image-model matching error, here measured using fixed templates of *r-g-b* pixel values for each limb with the use of support maps to handle self-occlusions[4]. The temporal prior $P(\mathbf{x}_t \,|\, \mathcal{X}_{t-1})$ is computed from the learned dynamics. The choice of discrete class label k_t is determined by the current region in state space, which in the implementation depends only on the previous pose $\mathbf{x}_{t-1}$, enabling us to express the probability as

$$p(\mathbf{x}_t \,|\, \mathcal{X}_{t-1}) \;\approx\; p(\mathbf{x}_t \,|\, \mathcal{X}_{t-1}, k_t)\, p(k_t \,|\, \mathbf{x}_{t-1}) \tag{7.8}$$

Note that with a second order ARP model, $p(\mathbf{x}_t \,|\, \mathcal{X}_{t-1}) = p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \mathbf{x}_{t-2})$.

Three additional parameters are used during tracking, two for the image location of the body centre and one for overall scale. We learn translation and scale independently of limb movements, so these parameters are not part of the learned dynamical model — they are are modeled respectively as first and zeroth order random walks and learned online during tracking. This allows the method to track sequences without assuming either static or fixating cameras.

---

[4]Support maps make use of information on limb layers to label each template pixel as being visible or occluded by another limb. An efficient matching scheme for image templates is described in [53].
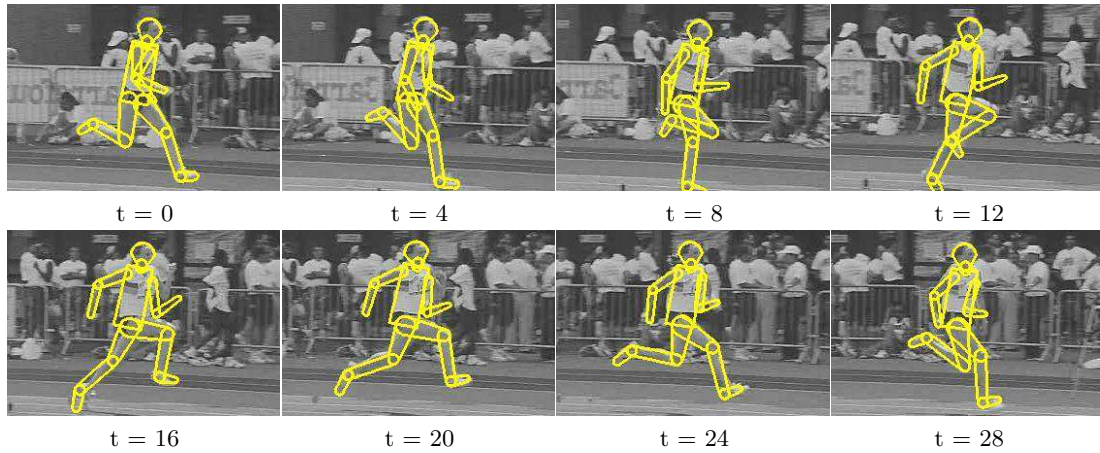
Figure 7.3: *Results from tracking athletic motion. The tracker was trained on a different athlete performing a similar motion. Strong priors from the dynamical model allow individual limbs to be tracked in the presence of a confusing background. Note that the left arm is not tracked accurately. This is due to the fact that it was occluded in the initial image and hence no information about its appearance was captured in the template. However, the dynamics continue to give a good estimate of its position.*
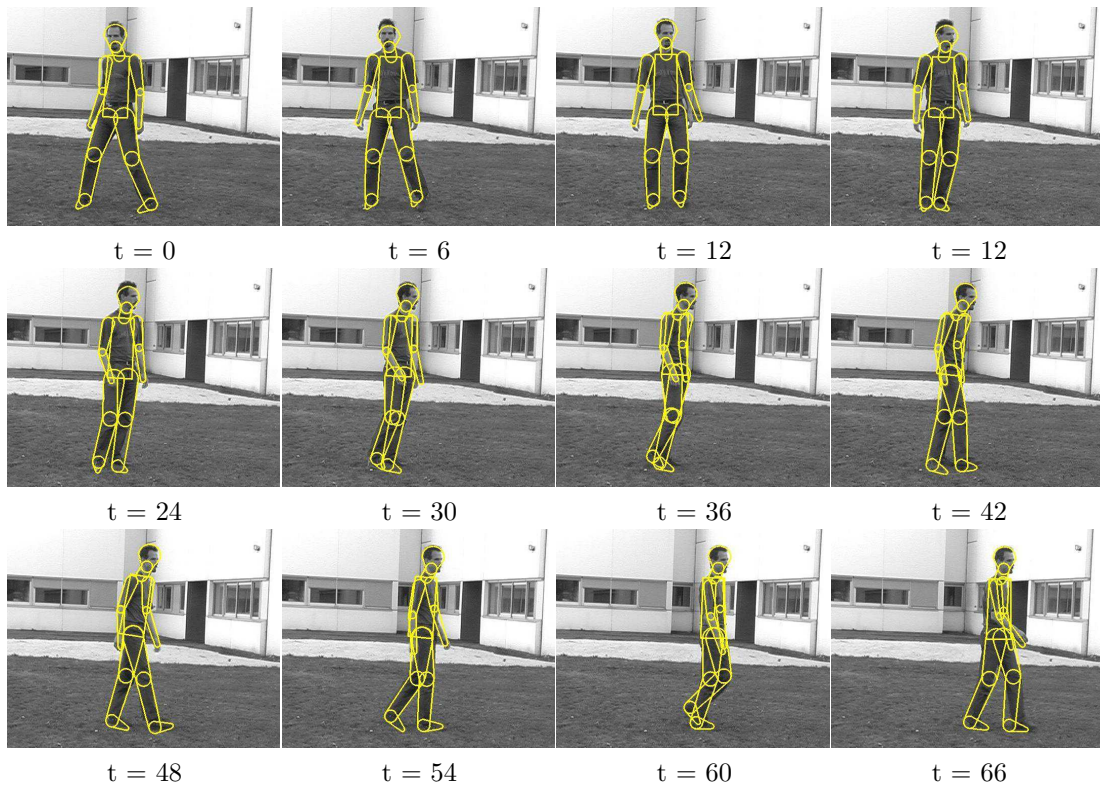
Figure 7.4: *Using a mixture of dynamical models to track through a turning motion in which the planar dynamics of the subject change significantly between the frontal and side views. The corresponding class memberships show a smooth transition between the turning and walking models.*

Figure 7.3 shows results from using the learned dynamics to track fast athletic motion. This is a case where traditional methods typically fail due to high motion blur. A hand-labeled sequence covering a few running cycles is used to train a model and this is used to track a different person performing a similar motion. For a given viewing direction, we find that a single 2nd order autoregressive process in five dimensions suffices to capture the dynamics of such running motions.

Figure 7.4 shows results on tracking through a transition from a turning motion to a walking motion on a test sequence. This example illustrates the effectiveness of the inter-class transition model. A 300-frame sequence consisting of walking in different directions and turning motion is used as training data and the learning algorithm correctly identifies 3 motion patterns corresponding to two different walking directions and turning between them. The tracker is initialized manually in both the examples.

7.5 Discussion

The focus of this chapter was modeling the dynamics of high degree-of-freedom systems such as the human body. We have described a method that makes use of a mixture of autoregressive processes to track through different classes of motion with smooth transitions between classes.

This chapter uses an explicit body model and hence is not in line with the bottom-up and discriminative methods which are developed in the remainder of this thesis. Nevertheless, it is shown that dynamics can effectively be modeled on a set of reduced subspaces and a mixture of dynamical models can be learned motion data. This would be useful for extending the trackers developed in previous chapters to track through multiple motion categories, which is essential for many practical applications.

8

Towards a Generic Image Representation

Histograms of local appearance descriptors are a popular representation for visual recognition. They are highly discriminant and have good resistance to local occlusions and to geometric and photometric variations, but they are not able to exploit spatial co-occurrence statistics at scales larger than their local input patches. In this chapter we present a new multilevel visual representation, 'hyperfeatures', that is designed to remedy this. The starting point is the familiar notion that to detect object parts, in practice it often suffices to detect co-occurrences of more local object fragments — a process that can be formalized as comparison (*e.g.* vector quantization) of image patches against a codebook of known fragments, followed by local aggregation of the resulting codebook membership vectors to detect co-occurrences. This process converts local collections of image descriptor vectors into somewhat less local histogram vectors — higher-level but spatially coarser descriptors. We observe that as the output is again a local descriptor vector, the process can be iterated, and that doing so captures and codes ever larger assemblies of object parts and increasingly abstract or 'semantic' image properties. We formulate the hyperfeatures model and study its performance under several different image coding methods including clustering based Vector Quantization, Gaussian Mixtures, and combinations of these with Latent Dirichlet Allocation. We find that the resulting high-level features provide improved performance in several object image and texture image classification tasks.

8.1   Introduction

Local codings of image appearance based on invariant descriptors are a popular representation for visual recognition [127, 125, 10, 90, 37, 81, 82, 31, 104, 69, 42]. The image is treated as a loose collection of quasi-independent local patches, robust visual descriptors are extracted from these, and a statistical summarization or aggregation process is used to capture the statistics of the resulting set of descriptor vectors and hence quantify the image appearance. There are many variants. Patches can be selected at one or at many scales, and either densely, at random, or sparsely according to local informativeness criteria [54, 70]. There are many kinds of local descriptors, which can incorporate various degrees of resistance to common perturbations such as viewpoint changes, geometric deformations, and photometric transformations [130, 90, 125, 93, 95]. Aggregation can be done in different ways, either over local regions to make higher-level local descriptors, or globally to make whole-image descriptors.

The simplest example is the 'texton' or 'bag-of-features' approach. This was initially developed for texture analysis (*e.g.* [92, 89]), but turns out to give surprisingly good performance in many image

classification and object recognition tasks [160, 37, 31, 104, 69, 42]. Local image patches or their feature vectors are coded using vector quantization against a fixed codebook, and the votes for each codebook centre are tallied to produce a histogram characterizing the distribution of patches over the image or local region. Codebooks are typically constructed by running clustering algorithms such as k-means over large sets of training patches. Soft voting into several nearby centres can be used to reduce aliasing effects. More generally, EM can be used to learn a mixture distribution or a deeper latent model in descriptor space, coding each patch by its vector of posterior mixture-component membership probabilities or latent variable values.

8.1.1 Hyperfeatures

The main limitation of local coding approaches is that they capture only the first order statistics of the set of patches (within-patch statistics and their aggregates such as means, histograms, *etc.*), thus ignoring the fact that inter-patch statistics such as co-occurrences are important for many recognition tasks. To alleviate this, several authors have proposed methods for incorporating an additional level of representation that captures pairwise or neighbourhood co-occurrences of coded patches [111, 128, 129, 10, 81].

This paper takes the notion of an additional level of representation one step further, generalizing it to a generic method for creating multi-level hierarchical codings. The basic intuition is that image content should be coded at several levels of abstraction, with the higher levels being spatially coarser but (hopefully) semantically more informative. Our approach is based on the local histogram model (*e.g.* [111, 129]). At each level, the image is divided into local regions with each region being characterized by a descriptor vector. The base level contains raw image descriptors. At higher levels, each vector is produced by coding (*e.g.* vector quantizing) and locally pooling the finer-grained descriptor vectors from the preceding level. For instance, suppose that the regions at a particular level consist of a regular grid of overlapping patches that uniformly cover the image. Given an input descriptor vector for each member of this grid, the descriptors are vector quantized and their resulting codes are used to build local histograms of code values over (say) 5×5 blocks of input patches. These histograms are evaluated at each point on a coarser grid, so the resulting upper level output is again a grid of descriptor vectors (local histograms). The same process can be repeated at higher levels, at each stage taking a local set of descriptor vectors from the preceding level and returning its coded local histogram vector. We call the resulting higher-level features **hyperfeatures**. The codebooks are learned in the usual way, using the descriptor vectors of the corresponding level from a set of training images. To promote scale-invariant recognition, the whole process also runs at each layer of a conventional multi-scale image pyramid, so there is actually a pyramid, not a grid of descriptor vectors at each level of the hyperfeature hierarchy[1]. The hyperfeature construction process is illustrated in figure 8.1.

Our main claim is that hyperfeature based coding is a natural feature extraction framework for visual recognition. In particular, the use of vector quantization coding followed by local his-togramming of membership votes provides an effective means of integrating higher order spatial relationships into texton style image representations. The resulting spatial model is somewhat 'loose' — it only codes nearby co-occurrences rather than precise geometry — but for this reason it is robust to spatial misalignments and deformations and to partial occlusions, and it fits well with the "spatially weak / strong in appearance" philosophy of texton representations. The basic intuition is that despite their geometric weakness, *in practice* simple co-occurrences of characteristic object fragments are often sufficient cues to deduce the presence of larger object parts, so that

---

[1]Terminology: 'layer' denotes a standard image pyramid layer, i.e. the same image at a coarser scale; 'level' denotes the number of folds of hyperfeature (quantize-and-histogram) local coding that have been applied, with each transformation producing a different, higher-level 'image' or 'pyramid'.
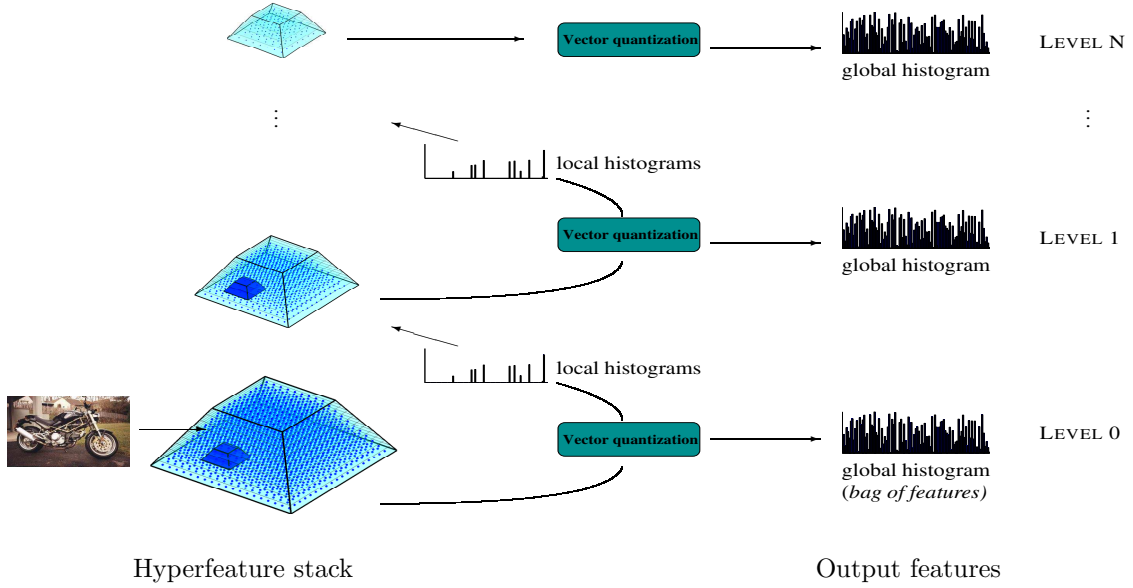
Figure 8.1: *Constructing a hyperfeature stack. The 'level 0' (base feature) pyramid is constructed by calculating a local image descriptor vector for each patch in a multiscale pyramid of overlapping image patches. These vectors are vector quantized according to the level 0 codebook, and local histograms of codebook memberships are accumulated over local position-scale neighbourhoods (the smaller darkened regions) to make the level 1 feature vectors. The process simply repeats itself at higher levels. The level l to l+1 coding is also used to generate the level l output vectors — global histograms over the whole level-l pyramid. The collected output features are fed to a learning machine and used to classify the (local or global) image region.*

as one moves up the hyperfeature hierarchy, larger and larger assemblies of parts are coded until ultimately one codes the entire object. Owing to their loose, agglomerative nature, hyperfeature stacks are naturally robust to occlusions and feature extraction failures. Even when the top level object is not coded successfully, substantial parts of it are captured by the lower levels of the hierarchy and the system can still cue recognition on these.

8.1.2 Previous Work

The hyperfeature representation has several precursors. Classical 'texton' or 'bag of features' representations are global histograms over quantized image descriptors — 'level 0' of the hyperfeature representation [92, 89]. Histograms of quantized 'level 1' features have also been used to classify textures and to recognize regularly textured objects [111, 129] and a hierarchical feature-matching framework for simple second level features has been developed [79].

Hyperfeature stacks also have analogies with multilevel neural models such as the neocognitron [48], Convolutional Neural Networks (CNN) [84] and HMAX [119]. These are all multilayer networks with alternating stages of linear filtering (banks of learned convolution filters for CNN's and of learned 'simple cells' for HMAX and the neocognitron) and nonlinear rectify-and-pool operations. The neocognitron activates a higher level cell if atleast one associated lower level cell is active. In CNN's the rectified signals are pooled linearly, while in HMAX a max-like operation ('complex cell') is used so that only the dominant input is passed through to the next stage. The neocognitron and

HMAX lay claims to biological plausibility whereas CNN is more of an engineering solution, but all are convolution based and typically trained discriminatively. In contrast, although hyperfeatures are still bottom-up, they are essentially a descriptive statistics model not a discriminative one: training is completely unsupervised and there are no convolution weights to learn for hyperfeature extraction, although the object classes can still influence the coding indirectly via the choice of codebook. The basic nonlinearity is also different: exemplar comparison by nearest neighbour lookup — or more generally nonlinear codings based on membership probabilities of latent patch classes — followed by a comparatively linear accumulate-and-normalize process for hyperfeatures, *versus* linear convolution filtering followed by simple rectification for the neural models.

The term 'hyperfeatures' itself has been used to describe combinations of feature position with appearance [44]. This is very different from its meaning here.

8.2   Base Features and Image Coding

The hyperfeature framework can be used with a large class of underlying image coding schemes. This section discusses the schemes that we have tested so far. For simplicity we describe them in the context of the base level (level 0).

8.2.1   Image Features

The 'level 0' input to the hyperfeature coder is a base set of local image descriptors. In our case these are computed on a dense grid — in fact a multiscale pyramid — of image patches. As patch descriptors we use SIFT-like gradient orientation histograms, computed in a manner similar to [90] but using a normalization that is more resistant to image noise in nearly empty patches. (SIFT was not originally designed to handle patches that may be empty). The normalization provides good resistance to photometric transformations, and the spatial quantization within SIFT provides a pixel or two of robustness to spatial shifts. The input to the hyperfeature coder is thus a pyramid of 128-D SIFT descriptor vectors. But other descriptors could also be used (*e.g.* [99, 17]).

Hyperfeature models based on sparse (*e.g.* keypoint based [37, 31, 81, 95]) feature sets would also be possible but they are not considered here, in part for simplicity and space reasons and in part because recent work (*e.g.* [69]) suggests that dense representations will outperform sparse ones.

8.2.2   Vector Quantization and Gaussian Mixtures

Vector quantization is a simple and widely-used method of characterizing the content of image patches [89]. Each patch is coded by finding the most similar patch in a dictionary of reference patches and using the index of this patch as a label. Here we use nearest neighbour coding based on Euclidean distance between SIFT descriptors, with a vocabulary learned from a training set using a clustering algorithm similar to the mean shift based on-line clusterer of [69]. The histograms have a bin for each centre (dictionary element) that counts the number of patches assigned to the centre. In the implementation, a sparse vector representation is used for efficiency.

Although vector quantization turns out to be very effective, abrupt quantization into discrete bins does cause some aliasing. This can be reduced by **soft vector quantization** — softly voting into the centers that lie close to the patch, *e.g.* with Gaussian weights. Taking this one step further, we can fit a probabilistic **mixture model** to the distribution of training patches in descriptor space, subsequently coding new patches by their vectors of posterior mixture-component membership probabilities. This gives centres that are qualitatively very different from those obtained by
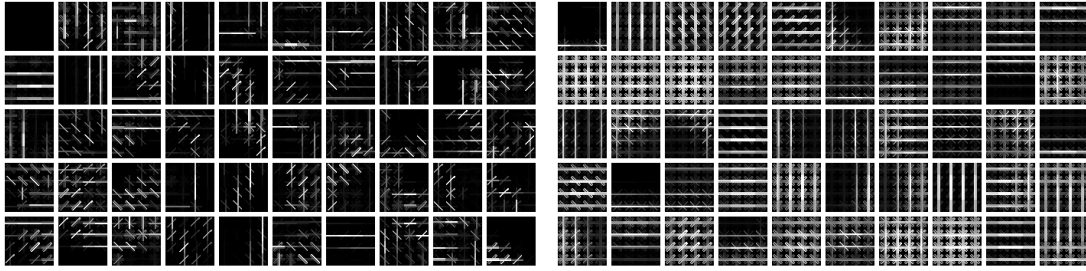
Figure 8.2: *Codebook centers obtained for SIFT descriptors from a dataset of 684 images from 4 object categories (the PASCAL dataset). The intensity of each line represents the weight of the corresponding orientation bin in that cell. (Left) Vector quantization cluster centers that are obtained using the mean-shift based clustering algorithm of [69], and (Right) Gaussian mixture centres. The two codebooks clearly code information very differently — VQ picks sparse 'natural features' while the GM tends to converge to denser, more averaged out features corresponding to structures such as vertical/horizontal lines, textures etc. The blank patch occurs very frequently in this particular dataset, mostly from uncluttered parts of background, and is hence almost always prominent amongst the centres.*

clustering, as shown in figure 8.2. In §8.4 we test hard vector quantization (VQ) and diagonal-covariance Gaussian mixtures (GM) fitted using Expectation-Maximization. The GM codings turn out to be more effective.

8.2.3 Latent Dirichlet Allocation

VQ and mixture models are flexible coding methods, but capturing fine distinctions often requires a great many centres. This brings the risk of fragmentation, with the patches of an object class becoming scattered over so many label classes that it is difficult to learn an effective recognition model for it. 'Bag of words' text representations face the same problem – there are many ways to express a given underlying 'meaning' in either words or images. To counter this, one can attempt to learn deeper latent structure models that capture the underlying semantic "topics" that generated the text or image elements. This improves learning because each topic label summarizes the 'meaning' of many different 'word' labels.

The simplest latent model is Principal Components Analysis ('Latent Semantic Analysis' *i.e.* linear factor analysis), but in practice statistically-motivated nonlinear approaches such as Probabilistic Latent Semantic Analysis (pLSA) [56] perform better. There are many variants on pLSA, typically adding further layers of latent structure and/or sparsifying priors that ensure crisper distinctions [26, 28, 72, 25]. Here we use **Latent Dirichlet Allocation (LDA)** [20]. LDA models document words as samples from sparse mixtures of topics, where each topic is a mixture over word classes. More precisely: the gamut of possible topics is characterized by a learned matrix $\boldsymbol{\beta}$ of probabilities for each topic to generate each word class; for each new document a palette of topics (a sparse multinomial distribution) is generated from a Dirichlet prior; and for each word in the document a topic is sampled from the palette and a word class is sampled from the topic. Giving each word its own topic allows more variety than sharing a single fixed mixture of topics across all words would, while still maintaining the underlying coherence of the topic-based structure. In practice the learned values of the Dirichlet parameter α are small, ensuring that the sampled topic palette is sparse for most documents.

1. $\forall(i, x, y, s)$, $\mathcal{F}^{(0)}_{ixys} \leftarrow$ base feature at point (x, y), scale s in image i.

2. For $l = 0, \ldots, N$:

   - If learning, cluster $\{\mathcal{F}^{(l)}_{ixys} \mid \forall(i, x, y, s)\}$ to obtain a codebook of $d^{(l)}$ centres in this feature space.

   - $\forall i$:

      - If $l < N$, $\forall(x, y, s)$ calculate $\mathcal{F}^{(l+1)}_{ixys}$ as a $d^{(l)}$ dimensional local histogram by accumulating votes from $\mathcal{F}^{(l)}_{ix'y's'}$ over neighbourhood $\mathcal{N}^{(l+1)}(x, y, s)$.

      - If global descriptors need to be output, code $\mathcal{F}^{(l)}_{i\ldots}$ as a $d^{(l)}$ dimensional histogram $\mathcal{H}^{(l)}_i$ by globally accumulating votes for the $d^{(l)}$ centers from all (x, y, s).

3. Return $\{\mathcal{H}^{(l)}_i \mid \forall i, l\}$.

Figure 8.3: *The hyperfeature coding algorithm.*

In our case – both during learning and use – the visual 'words' are represented by VQ or GM code vectors and LDA functions essentially as a locally adaptive nonlinear dimensionality reduction method, re-coding each word (VQ or GM vector) as a vector of posterior latent topic probabilities, conditioned on the local 'document' model (topic palette). The LDA 'documents' can be either complete images or the local regions over which hyperfeature coding is occurring. Below we use local regions, which is slower but more discriminant. Henceforth, "coding" refers to either VQ or GM coding, optionally followed by LDA reduction.

8.3   Constructing Hyperfeatures

The hyperfeature construction process is illustrated in figure 8.1. At level 0, the image (more precisely the image pyramid) is divided into overlapping local neighbourhoods, with each neighbourhood containing a number of image patches. The co-occurrence statistics within each local neighbourhood $\mathcal{N}$ are captured by vector quantizing or otherwise nonlinearly coding its patches and histogramming the results over the neighbourhood. This process converts local patch-level descriptor vectors (image features) to spatially coarser but higher-level neighbourhood-level descriptor vectors (local histograms). It works for any kind of descriptor vector. In particular, it can be repeated recursively over higher and higher order neighbourhoods to obtain a series of increasingly high level but spatially coarse descriptor vectors.

Let $\mathcal{F}^{(l)}$ denote the hyperfeature pyramid at level l, (x, y, s) denote position-scale coordinates within a feature pyramid, $d^{(l)}$ denote the feature or codebook/histogram dimension at a level l, and $\mathcal{F}^{(l)}_{ixys}$ denote the level-l descriptor vector at (x, y, s) in image i. During training, a codebook or coding model is learned from all features (all i, x, y, s) at level l. In use, the level-l codebook is used to code the level-l features in some image i, and these are pooled spatially over local neighbourhoods $\mathcal{N}^{(l+1)}(x, y, s)$ to make the hyperfeatures $\mathcal{F}^{(l+1)}_{ixys}$. The complete algorithm for VQ coding on N levels is summarized in figure 8.3.

For vector quantization, coding involves a single global clustering for learning, followed by local histogramming of class labels within each neighbourhood for use. For GM, a global mixture model is learned using EM, and in use the mixture component membership probability vectors of the neighbourhood's patches are summed to get the code vector. If LDA is used, its parameters $\alpha, \boldsymbol{\beta}$ are estimated once over all training images, and then used to infer topic distributions over each
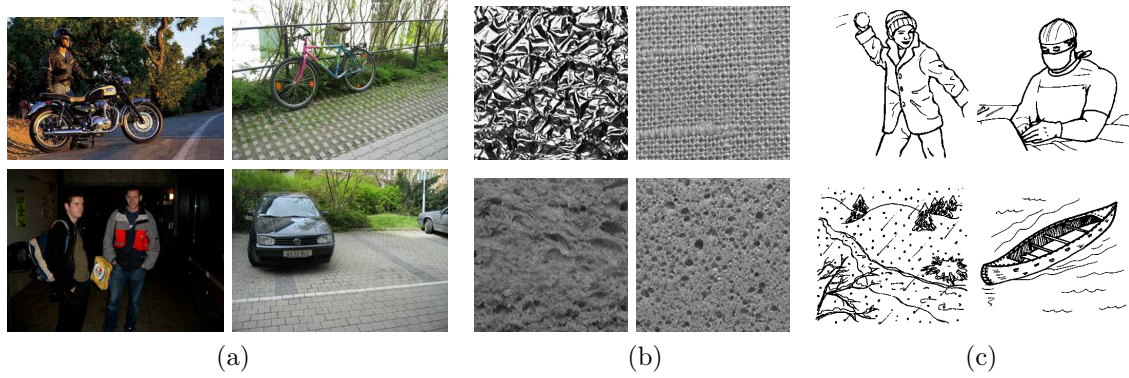
Figure 8.4: *Some typical images from the datasets that are used to evaluate hyperfeature based coding for image classification. (a) The PASCAL object dataset contains 4 classes: motorbikes, bicycles, people and cars; (b) 4 of the 10 different textures in the KTH-TIPS dataset; and (c) the CRL-IPNP dataset includes drawings of people as well as objects and scenes.*

neighbourhood independently, *i.e.* each neighbourhood is a separate 'document' with its own LDA context.

In all of these schemes, the histogram dimension is the size of the codebook or GM/LDA basis. The neighbourhoods are implemented as small trapezoids in scale space, as shown in figure 8.1. This shape maintains scale invariance and helps to minimize boundary losses, which cause the pyramids to shrink in size with increasing level. The size of the pooling region at each level is a parameter. The effective region size should grow with the level – otherwise the same information is re-encoded each time, which tends to cause rapid saturation and suboptimal performance.

8.4 Experiments on Image Classification

To illustrate the discriminative capabilities of hyperfeatures, we present image classification experiments on three datasets: a 4 class object dataset based on the "Caltech 7" and "Graz" datasets that was used for the European network PASCAL's "Visual Object Classes Challenge"; the 10 class KTH-TIPS texture dataset[2]; and the CRL-IPNP dataset of line sketches used for picture naming in language research[3]. The PASCAL dataset contains 684 training and 689 test images, which we scale to a maximum resolution of 320×240 pixels. The texture dataset contains 450 training and 360 test images over 10 texture classes, mostly 200×200 pixels. The CRL-IPNP dataset consists of 360 images of 300×300 pixels which we divide into two classes, images of people and others. As base level features we used the underlying descriptor of Lowe's SIFT method – local histograms of oriented image gradients calculated over 4×4 blocks of 4×4 pixel cells [90][4]. The input pyramid had a scale range of 8:1 with a spacing of 1/3 octave and patches sampled at 8 pixel intervals, giving a total of 2500-3000 descriptors per image. For the pooling neighbourhoods $\mathcal{N}$, we took volumes of 3×3×3 patches in (x, y, s) by default, increasing these in effective size by a factor of $2^{1/3}$ (one pyramid layer) at each hyperfeature level.

---

[2]The PASCAL object recognition database collection is available at www.pascal-network.org/challenges/VOC and the KTH-TIPS texture dataset is available at www.nada.kth.se/cvap/databases/kth-tips

[3]This dataset is a part of the International Picture Naming Project at the Centre of Research in Language, and is available at http://crl.ucsd.edu/∼aszekely/ipnp.

[4]But note that this is tiled densely over the image with no orientation normalization, not applied sparsely at keypoints and rotated to the dominant local orientation as in [90].
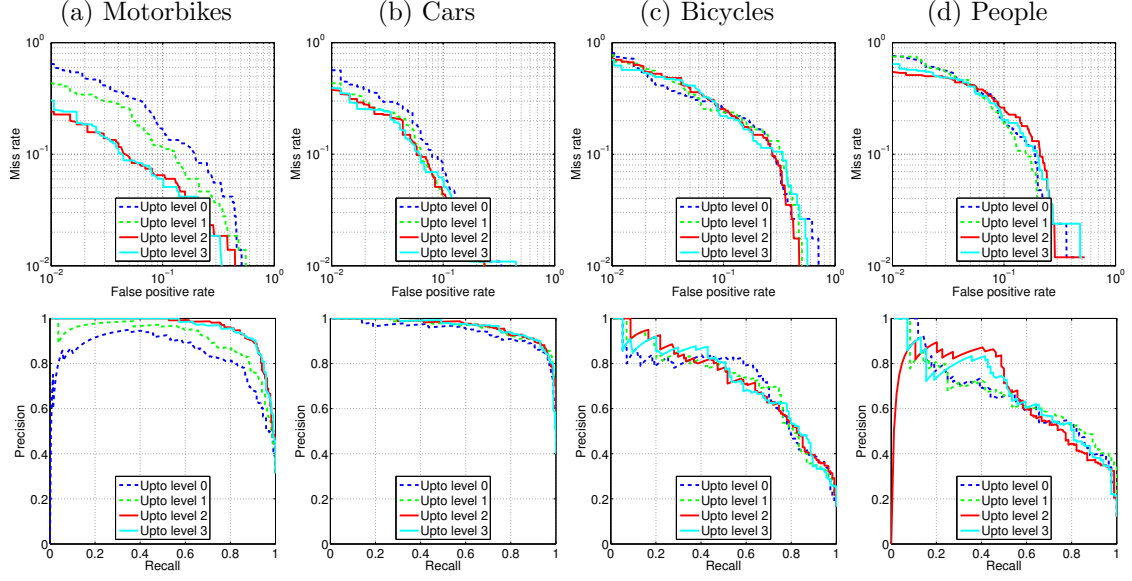
Figure 8.5: *Detection Error Trade-off and Recall-Precision curves for the classes of the PASCAL dataset. Up to a certain level, including additional levels of hyperfeatures improves the classification performance. For the motorbike, car and bicycle classes the best performance is at level 3, while for the person class it is at level 1 (one level above the base features). The large gain on the motorbike (a $5\times$ reduction in false positives at fixed miss rate) and car classes suggests that local co-occurrence structure is quite informative, and is captured well by hyperfeatures.*
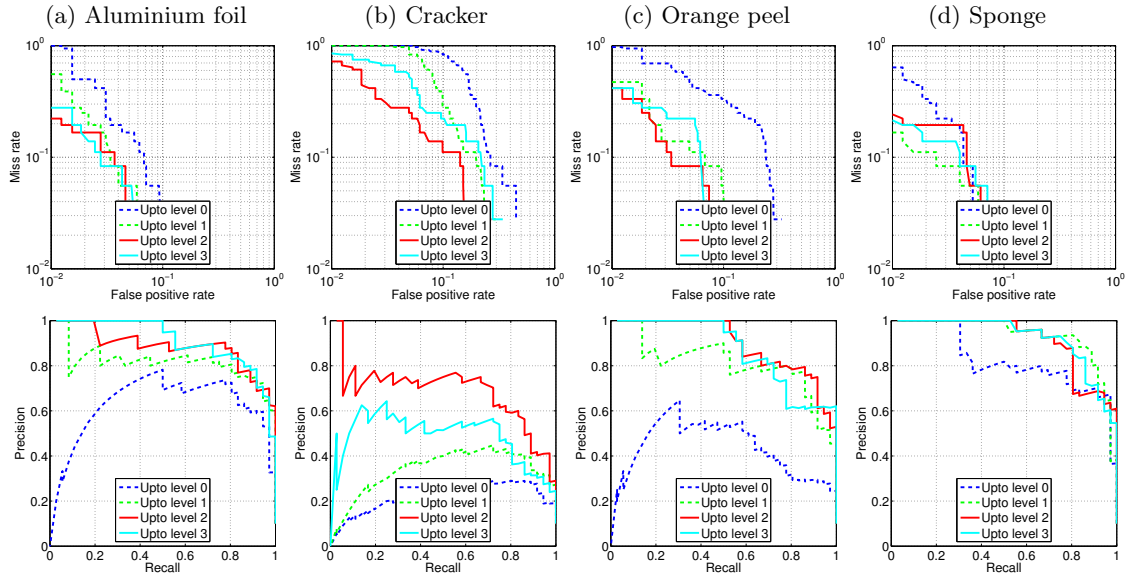
Figure 8.6: *Detection Error Trade-off curves for 4 of the 10 classes from the KTH-TIPS dataset, using a mixture of 100 Gaussians at each level. Including hyperfeatures improves the classification performance for every texture that is poorly classified at level 0, without hurting that for well-classified textures. The aluminium and sponge classes are best classified by including 3 levels of hyperfeatures, and cracker and orange peel by using 2 levels.*

	Al. foil	Bread	Corduroy	Cotton	Cracker	Linen	Orange peel	Sandpaper	Sponge	Styrofoam
VQ	97.2	88.1	100	86.1	94.4	77.8	94.4	83.3	91.7	88.9
GM	100	88.9	100	88.9	91.6	86.1	94.4	83.3	91.7	91.7

Figure 8.7: *One-vs-rest classification performance (hit rate) at the equal error point for the 10 classes of the texture dataset, using hard vector quantization (VQ) and a diagonal Gaussian mixture model learned by EM (GM). Each class uses its optimal number of hyperfeature levels. GM performs best on average.*
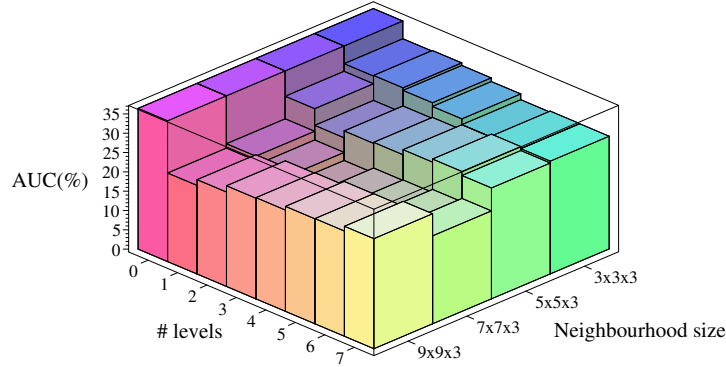
Figure 8.8: *Performance on the CRL-IPNP dataset: average miss rates on the positive class for different pooling neighbourhood sizes and different numbers of hyperfeature levels. For a 3x3x3 neighbourhood (in x, y, s), 5 levels of hyperfeatures are best, but the best overall performance is achieved by 7x7x3 neighbourhoods with 3 levels of hyperfeatures.*

The final image classifications were produced by training soft linear one-against-all SVM classifiers independently for each class over the global output histograms collected from the active hyperfeature levels, using SVM-light [65] with default settings.

Effect of multiple levels: Figure 8.5 presents DET[5] and recall-precision curves showing the influence of hyperfeature levels on classification performance for the PASCAL dataset. We used GM coding with a 200 center codebook at the base level and 100 center ones at higher levels. Including higher levels gives significant gains for 'cars' and especially 'motorbikes', but little improvement for 'bicycles' and 'people'. The results improve up to level 3 (*i.e.* using the hyperfeatures from all levels 0–3 for classification), except for 'people' where level 1 is best. Beyond this there is overfitting – subsequent levels introduce more noise than information. We believe that the difference in behaviour between classes can be attributed to their differing amounts of *structure*. The large appearance variations in the 'person' class leave little in the way of regular co-occurrence statistics for the hyperfeature coding to key on, whereas the more regular geometries of cars and motorbikes are captured well, as seen in figure 8.5(a) and (b). Different coding methods and codebook sizes have qualitatively similar evolutions the absolute numbers can be quite different (see below).

The results on the KTH-TIPS texture dataset in figure 8.6 lead to similar conclusions. For 4 of the 10 classes the level 0 performance is already near perfect and adding hyperfeatures makes little difference, while for the remaining 6 there are gains (often substantial ones) up to hyperfeature level 3. The texture classification performance at equal error rates for VQ[6] and GM coding is

---

[5]DET curves plot miss rate *vs.* false positive rate on a log-log scale – the same information as a ROC curve in more visible form. Lower values are better.

[6]At the base level of the texture dataset, we needed to make a manual correction to the SIFT VQ codebook to work around a weakness of codebook creation. Certain textures are homogeneous enough to cause all bins of the

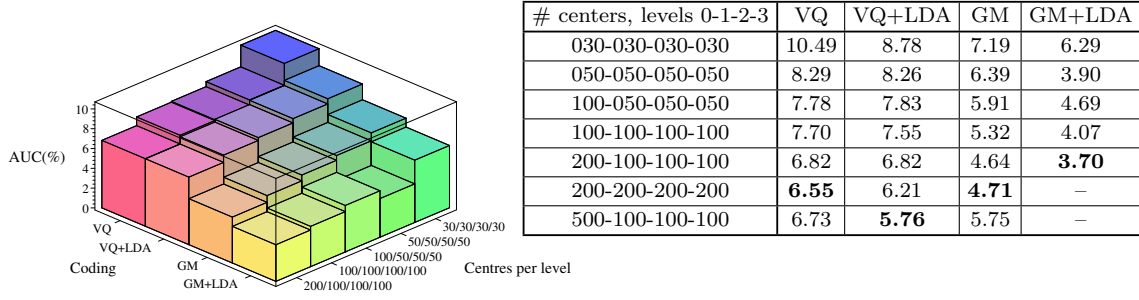| # centers, levels 0-1-2-3	VQ	VQ+LDA	GM	GM+LDA
030-030-030-030	10.49	8.78	7.19	6.29
050-050-050-050	8.29	8.26	6.39	3.90
100-050-050-050	7.78	7.83	5.91	4.69
100-100-100-100	7.70	7.55	5.32	4.07
200-100-100-100	6.82	6.82	4.64	**3.70**
200-200-200-200	**6.55**	6.21	**4.71**	–
500-100-100-100	6.73	**5.76**	5.75	–

Figure 8.9: *Average miss rates on the PASCAL objects test set. Miss rates for different codebook sizes and coding methods. Larger codebooks always give better performance. GM coding outperforms VQ coding even with significantly fewer centres, and adding LDA consistently improves the results. The LDA experiments use the same number of topics as VQ/GM codebook centres, so they do not change the dimensionality of the code, but they do make it sparser.*

# centers at levels 0-1-2-3	Train	Test
600-000-000-000	7.46	8.35
300-300-000-000	5.30	7.77
400-200-000-000	4.06	7.41
200-200-200-000	4.24	6.57
150-150-150-150	4.35	7.30
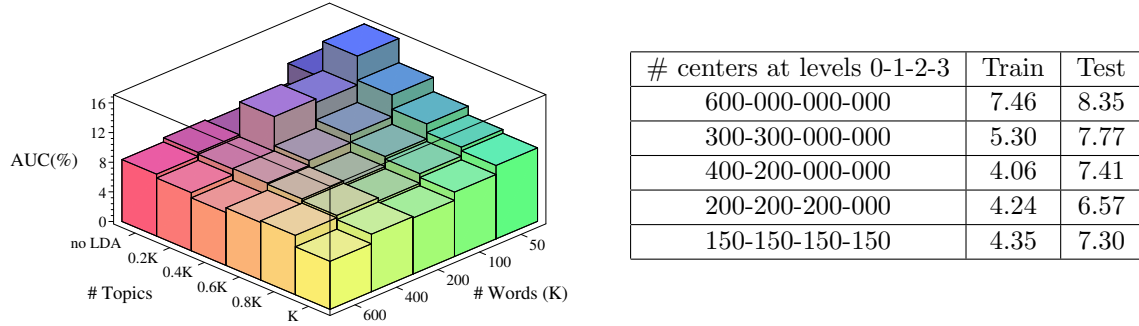
Figure 8.10: *(Left) The effect of LDA on average classification performance on test images: average miss rates for the PASCAL objects testset. Performance improves systematically as both code centres (here VQ) and LDA topics are added. (Right) For a fixed total number of centers (here VQ ones), performance improves if they are distributed relatively evenly across several levels (here 3 levels, with the inclusion of a 4th reducing the performance). I.e. adding higher level information is more useful than adding finer-grained low level information.*

shown in figure 8.7. GM is better on average. Overall, its mean hit rate of 91.7% at equal error is slightly better than the 90.6% achieved by the bank of filters approach in [47] – a good result considering that in these experiments relatively few centres, widely spaced samples and only a linear SVM were used. (Performance improves systematically with each of these factors).

On the CRL-IPNP dataset, we find that 4 or 5 levels of hyperfeatures give the best performance, depending on the size of the pooling regions used. See figure 8.8. Experiments on this dataset all use a 100-centre codebook at each level.

Coding methods and hyperfeatures: Figure 8.9 shows average miss rates[7] on the PASCAL dataset, for different coding methods and numbers of centers. The overall performance depends considerably on both the coding method used and the codebook size (number of clusters / mixture components / latent topics), with GM coding dominating VQ, the addition of LDA always im-

---

SIFT descriptor to fire about equally, giving rise to a very heavily populated "uniform noise" centre in the middle of SIFT space. For some textures this centre receives nearly all of the votes, significantly weakening the base level coding and thus damaging the performance at all levels. The issue can be resolved by simply deleting the rogue centre (stop word removal). It does not occur either at higher levels or for GM coding.

[7]miss rate = (1 - Area Under ROC Curve)

proving the results, and performance increasing whenever the codebook at any level is expanded. On the negative side, learning large codebooks is computationally expensive, especially for GM and LDA. GM gives much smoother codings than VQ as there are no aliasing artifacts, and its partition of the descriptor space is also qualitatively very different – the Gaussians overlap heavily and inter-component differences are determined more by covariance differences than by centre differences. LDA seems to be able to capture canonical neighbourhood structures more crisply than VQ or GM, presumably because it codes them by selecting a sparse palette of topics rather than an arbitrary vector of codes. If used to reduce dimensionality, LDA may also help simply by reducing noise or overfitting associated with large VQ or GM codebooks, but this can not be the whole story as LDA performance continues to improve even when there are more topics than input centres. *c.f.* figure 8.10 (left).

Given that performance always improves with codebook size, one could argue that rather than adding hyperfeature levels, it may be better to include additional base level features. To study this we fixed the total coding complexity at 600 centres and distributed the centres in different ways across levels. Figure 8.10 (right) shows that spreading centres relatively evenly across levels (here up to level 3) improves the results, confirming the importance of higher levels of abstraction.

8.5 Object Localization

One advantage of hyperfeatures is that they offer a controllable tradeoff between locality and level of abstraction: higher level features accumulate information from larger image regions and thus have less locality but potentially more representational power. However, even quite high-level hyperfeatures are still local enough to provide useful object-region level image labeling. Here we use this for bottom-up localization of possible objects of interest. The image pyramid is tiled with regions and in each region we build a "mini-pyramid" containing the region's hyperfeatures (*i.e.* the hyperfeatures of all levels, positions and scales whose support lies entirely within the region). The resulting region-level hyperfeature histograms are then used to learn a local region-level classifier for each class of interest. Our goal here is simply to demonstrate the representational power of hyperfeatures, not to build a complete framework for object recognition, so the experiments below classify regions individually without any attempt to include top-down or spatial contiguity information.

The experiments shown here use the bounding boxes provided with the PASCAL dataset as object masks for foreground labeling[8]. The foreground labels are used to train linear SVM classifiers over the region histograms, one for each class with all background and other-class regions being treated as negatives. Figure 8.11 shows results obtained by using these one-against-all classifiers individually on the test images. Even though each patch is treated independently, the final labellings are coherent enough to allow the objects to be loosely localized in the images. The average accuracy in classifying local regions over all classes is 69%. This is significantly lower than the performance for classifying images as a whole, but still good enough to be useful as a bottom-up input to higher-level visual routines. Hyperfeatures again add discriminative power to the base level features, giving an average gain of 4–5% in classification performance (see [5] for details). Figure 8.12 shows the key entries of the combined and the two-class confusion matrices, with negatives being further broken down into true background patches and patches from the three remaining classes.

---

[8]This labeling is not perfect. For many training objects, the bounding rectangles contain substantial areas of background, which are thus effectively labeled as foreground. Objects of one class also occur unlabeled in the backgrounds of other classes and, *e.g.*, instances of people sitting on motorbikes are labeled as 'motorbike' not 'person'. In the experiments, these imperfections lead to some visible 'leakage' of labels. We would expect a more consistent foreground labeling to reduce this significantly.
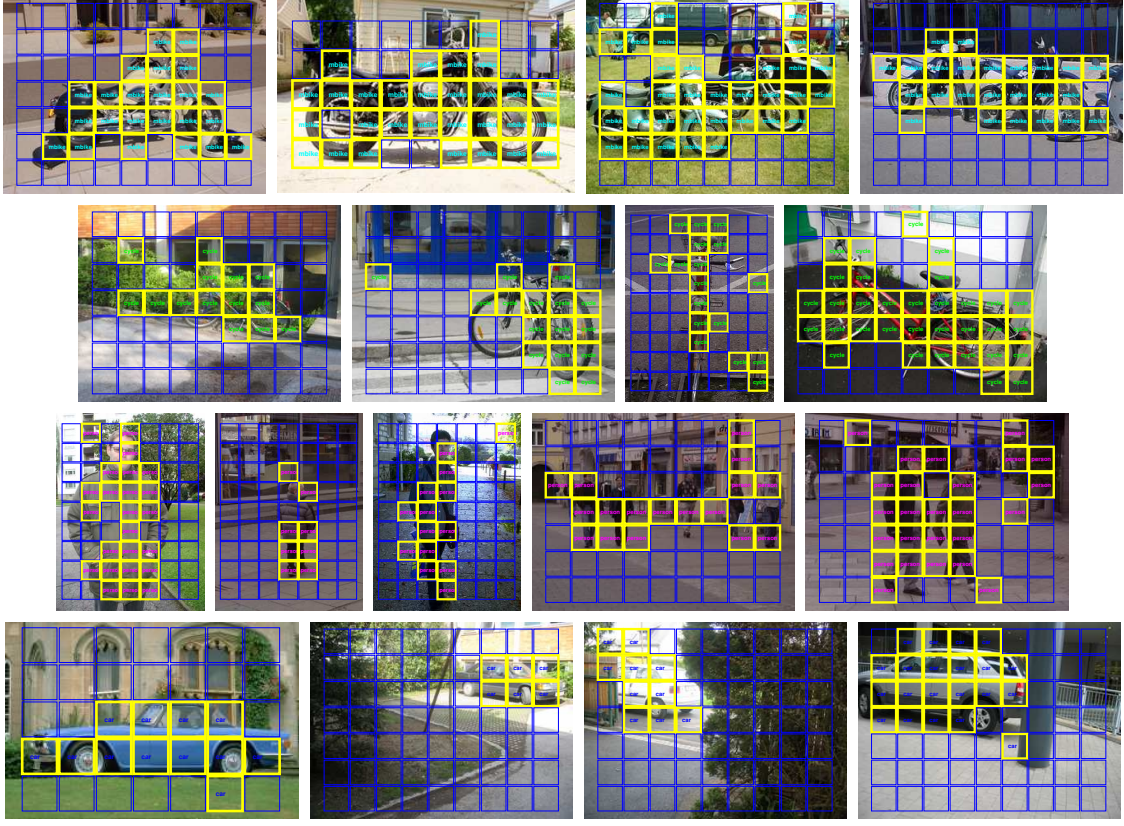
Figure 8.11: *Object localization in the PASCAL dataset by classifying local image regions using hyperfeatures. Each row shows examples of results using one of the four independent classifiers, each being trained to classify foreground regions of its own class against the combined set of all other regions – background regions and foregrounds from other classes. An image region is labeled as belonging to the object class if the corresponding SVM returns a positive score. Each region is classified independently – there is no attempt to enforce spatial coherence.*

true \ estimated	motorbike	cycle	person	car	background
motorbike	**41.02**	17.58	10.03	18.02	13.34
cycle	20.17	**42.21**	14.66	6.51	16.45
person	9.81	13.67	**55.71**	6.43	14.39
car	18.32	4.56	6.19	**63.00**	7.93
background	7.48	13.66	15.99	19.09	**43.78**
true proportion	20.62	9.50	3.52	4.71	61.65

true \ est.	motorbike	cycle	person	car
motorbike	**69.34**	45.17	19.79	35.76
cycle	49.82	**63.56**	26.08	14.43
person	27.01	35.37	**65.84**	19.54
car	52.43	12.43	10.39	**77.30**
background	16.36	19.81	19.46	23.46
negative	**22.98**	**25.81**	**19.74**	**25.07**

Figure 8.12: *Confusion matrices for region level labeling. Four two-class linear SVM region classifiers are trained independently, each treating regions from the background and from other classes as negatives. Left: A classical confusion matrix for the classifiers in winner-takes-all mode with negative best scores counting as background. The final row gives the population proportions, i.e. the score for a random classifier. Right: Each column gives entries from the pairwise confusion matrix of the corresponding classifier used alone (independently of the others), with the negative true-class scores (final row) broken down into scores on each other class and on the background. (NB: in this mode, the assigned class labels are not mutually exclusive).*

8.6 Discussion

We have introduced 'hyperfeatures', a new multilevel nonlinear image coding mechanism that generalizes – or more precisely, iterates – the quantize-and-vote process used to create local histograms in texton / bag-of-feature style approaches. Unlike previous multilevel representations such as convolutional neural networks and HMAX, hyperfeatures are optimized for capturing and coding local appearance patches and their co-occurrence statistics. Our experiments show that the introduction of one or more levels of hyperfeatures improves the performance in many classification tasks, especially for object classes that have distinctive geometric or co-occurrence structures.

The hyperfeature idea is applicable to a wide range of problems involving part-based representations. In this chapter the hyperfeature codebooks have been trained bottom-up by unsupervised clustering, but more discriminative training methods should be a fruitful area for future investigation. For example image class labels could usefully be incorporated into the learning of latent topics. Also, more general LDA like methods could be investigated that use local context while training. One way to do this is to formally introduce a "region" (or "subdocument") level in the word–topic–document hierarchy. Such models should allow us to model contextual information at several different levels of support, which may be useful for object detection.

9

Conclusion and Perspectives

This thesis has touched on several aspects of image understanding and in particular of recognizing and reconstructing human actions by combining ideas from computer vision, motion capture, and machine learning. We have shown that an effective vision-based human pose recognition system can be constructed using example based statistical models of a collection of motion capture recordings with their corresponding images. Beyond our direct contributions, this approach will provide a useful framework for further research in the area.

9.1 Key Contributions

- **Efficient markerless motion capture in a controlled environment.** In a controlled environment where the lighting and background are fixed or can be estimated, extracting silhouettes from the images has proved to be an effective strategy. Despite loss of some information in reducing images to silhouettes, we have found that this is a very practical approach and the kernel based regression methods developed in this thesis allow reasonably accurate pose estimation and efficient tracking. This is suitable for several tasks. We have demonstrated results on several kinds of walking motion as well as arm gestures.

- **Multimodal pose density from a single image.** The mixture of regressors allows for a probabilistic estimate of the various 3D pose solutions that may correspond to a single image. To the best of our knowledge, this is the first explicit probabilistic model of the multimodal posteriors generated by the ambiguities of monocular image based 3D human pose reconstruction. The method also has direct applications to building self initializing human trackers and to recognizing gestures.

- **Bottom-up pose recognition in clutter.** By introducing a coding method that suppresses features contributed by clutter, we have developed a purely bottom-up and model-free approach to human pose estimation in cluttered images that requires no prior segmentation of the subject. Previous approaches have required a prior segmentation.

- **Hyperfeatures.** We have introduced hyperfeatures, a new multilevel nonlinear image coding mechanism that generalizes the quantize-and-vote process used to create local histograms in texton / bag-of-feature style approaches. Hyperfeatures are optimized for capturing and coding local appearance patches and their co-occurrence statistics. Our experiments show that the introduction of one or more levels of hyperfeatures improves the performance in many classification tasks, especially for object classes that have distinctive geometric or appearance based co-occurrence structures.

9.2   Possible Extensions

The work presented in this thesis provides solutions for many different cases of the target problem and has several direct applications. Nevertheless, there are a number of shortcomings. While some of these can be addressed by extensions within the proposed framework, it is less obvious how to overcome others. Below we suggest several potentially promising extensions of the methods developed here. Open problems are discussed in the next section.

Exploiting structure in the output space

The algorithms developed in this work are model-free. Poses are encoded as simple vectors and predicted using essentially generic regression methods. The system has no explicit knowledge of the fact that the output represents the pose or motion of a structured human body. This representation allows very fast pose estimation by reducing inference to a set of linear operations (on a nonlinearly encoded image representation), but it necessarily ignores a lot of the structure present in the output space. It would be interesting to exploit this structure without explicitly building a human body model. For example, the underlying kinematic tree of the body could be inferred automatically by learning a tree-structured graphical model [15] that 'explains' the dependencies implicit in the observed training data. A number of inference methods for multiple dependent output variables and structured output spaces have been developed in the context of classification or discrete labeling, *e.g.* [71, 153, 75, 76, 11], and even for time varying data [146]. These can perhaps be extended to infer continuous human body pose using an underlying tree structure. In fact, the structure is not restricted to be a tree: one advantage of learning it automatically from data is that by learning a general graph, one can capture dependencies such as the coordinated movements of limbs that are observed in many common motions, and not just the kinematic constraints that are inherited from the body's skeletal structure. For example, in a related context, graphical models have recently been used to augment the tree structure of the body with latent variables that can account for coordinations between limbs [78].

Combination with model-based methods

Our current methods do not attain the same precision as commercially available motion capture systems. In large part this is due to the inherent difficulties of monocular and markerless motion capture, but model-free, example-based approaches are limited by the ranges of their training examples and hence do not always achieve perfect alignment with the input image. The introduction of an explicit body model would allow direct projection of the reconstructed pose into the image, and hence explicit adjustment of the pose to optimize image likelihoods. Such top-down optimization is computationally expensive, but (given an appropriate likelihood model) it would be very likely to improve accuracy. This suggests that initializing the pose with a model-free bottom-up estimate and then optimizing it with a top-down model would be an effective strategy. *c.f.* [139].

One could even use the top down model to provide feedback on the likelihoods of the image features used to predict the pose and thus initiate a cycle of bottom-up and top-down steps. For example, in a silhouette based method the segmentation itself could be refined by projecting the estimated pose onto the image using a 3D body model and this refined silhouette could then be used to re-estimate the pose. Such iteration would be computationally expensive but it is likely to give significantly higher accuracy. Figure B.2(b) shows a 'unified' body model of the kind that could be used for such an algorithm. It consists of a *loose* graphical structure for bottom-up inference and a renderable 3D surface for top-down refinement.

Extended motion categories

One relatively straightforward extension to the tracking algorithms presented in chapters 4 and 5 would be to extend the dynamical models in order to support motion capture over a wider class of

movements. This work used relatively weak motion priors based on a single linear autoregressive process. We find that this suffices for simple walking and arm-gesture motions, but tracking through several different activities could probably be handled better by using a mixture of dynamical models. One such framework is presented in chapter 7. Other methods that use a collection of motion models have been developed for motion synthesis and shown to be effective in dealing with transitions between different motion categories [86, 12]. Using such methods over motion capture data from various activities would be useful for constructing models that could track many more kinds of motion.

Using alternate descriptors for recognizing pose

We have seen that the robust shape descriptors developed in chapter 3 for representing human silhouettes are very effective at recovering full body pose from a variety of images when segmentation is possible. In the context of natural images containing clutter, however, the method developed in chapter 6 is restricted to reconstructing pose from image windows localized at a person. Reliably detecting the parts of people remains a challenge and there is a lot of scope for improved image descriptors in this context. Several directions are possible. Firstly, although very effective, the SIFT features that we have used were originally designed for feature matching with some degree of invariance to changes of position, not for encoding the detailed positions of human limbs. The spatial quantization of SIFT may be too coarse for accurate position estimation and it would perhaps be better to use 'fine coding' ideas from psychophysics, *i.e.* redundant coding with overlapping, smoothly windowed spatial cells. Another approach would be to learn combinations of image filters that are optimized for coding human pose. Higher-level 'features' such as face and limb detector responses could also be combined with the image representations used here.

Secondly, video data can be used to include motion information within the input features. It is well known that observations are not independent across time and several types of motion features have been developed for human detection and tracking tasks, *e.g.* [161, 80, 33]. Such methods could be a useful complement to the static image features that are used in the regressors developed in this thesis.

Thirdly, latent semantic analysis methods such as those used in chapter 8 could be developed to provide more semantic-level features for human pose. As a starting point, we would like to see how pLSA or LDA encoded image patches compare with the NMF ones of chapter 6. Also, hyperfeatures have been shown to be an effective encoding technique for image recognition but their application to pose estimation is yet to be studied.

Going beyond regression

Our current methods for estimating human pose and motion from images are essentially all based on least squares regression. This is true of the static method, the tracking method incorporating dynamical predictions, and even the probabilistic mixture model for posterior pose. Several other possibilities could be explored. In a broad sense the problem is essentially one of inferring a continuous and high dimensional time varying state from a set of observed signals. Statistical methods such as linear or kernel Canonical Correlation Analysis [77] could prove helpful in understanding the relationship between different components of the pose vectors and image features. It is also important to model the uncertainty of the predictions and Gaussian processes [116] are known to give more reasonable uncertainty measures than simple regressors such as the Relevance Vector Machine, particularly for data outside the span of the training database. Several models based on these (*e.g.* [147]) might prove useful for human pose recovery. *c.f.* [155]. Other possibilities include inference on structured output spaces as discussed above.

9.3   Open Problems

A complete understanding of human motion from images still eludes us. It is associated with a number of technically challenging open problems that also raise to philosophical issues, perhaps emphasizing the need for a deeper understanding of biological visual perception. We now mention some of these.

Multiple people and occlusions

Many real-life applications such as interpreting and interacting with natural environments and day-to-day scenes require a system that can handle multiple and partly occluded people. This problem goes well beyond the techniques addressed in this thesis and a complete solution is likely to require the cooperation of several interacting components. One possibility would be to use a human detector, running with a pose recognition system like the one from chapter 6 on top of each detection. However it is unclear whether these stages can be treated independently. In particular, people in a scene often occlude and interact with one another and any pose recognition algorithm should probably take this into account.

There are some interesting recent advances in the problem of simultaneous detection and segmentation of multiple people, *e.g.* [88, 164]. Precise 3D pose is not required for some applications and these papers are a step in that direction. However other applications, for example systems such as robots that interact physically with people, need a detailed activity analysis at the level of 3D body pose. How these detection-segmentation methods could interact with a pose estimation algorithm such as the one developed in this thesis is an interesting direction to explore.

Another problem posed by crowded or otherwise cluttered environments is that of occlusions. Parts of the body may not be visible in an image due to other objects or people. Self occlusions can be modeled by relying on having seen similar situations in the training data, but handling more general occlusions requires some sort of inference about which parts of the subjects are actually visible in the image. It might be possible to formulate probabilistic graphical models with extra latent variables that could "switch off" inference on parts that are not visible. It remains to be investigated whether such local inference suffices, or whether a more global understanding of which sections of the body are occluded is required. With silhouettes, for example, if a partial silhouette is available the global statistics of local descriptors on it might be compared with those of complete silhouettes from the training database, using KL divergence to measure image similarity in a framework similar to that of chapter 3. This might give reasonable partial pose estimates but it would not directly reveal which limbs are actually occluded. This may seem strange because we might expect certain limbs to be labelled as occluded and ignored, with no further attempt to infer their configuration from the image. On the other hand it could be argued that it is useful to 'guess' the configuration of the occluded parts by exploiting prior knowledge. The usefulness of this is likely to depend very much on the problem context and the problem remains open in general.

The role of context

Each method that we have discussed applies a given pose estimation algorithm irrespective of the image and scene context. The dynamical model does adapt according to the type of motion observed, but it still uses only features from the person to infer pose. In contrast, a human looking at the problem would often use contextual cues to understand what the people in the image are doing. For example, certain activities are more likely to occur in certain kinds of environments, or after certain other actions. It is an open problem how best to incorporate an understanding of the global context into pose estimation. Using a rough estimate of scene geometry from an image

has been shown to be effective in the problem of object detection [57], but the extension of such contextual approaches to pose recovery or action labelling remains to be investigated.

Online learning

Another limitation of the methods developed in this thesis is that the inferential model cannot *adapt* over time. A training phase is performed offline, and after this the system never changes. Chapter 1 mentioned that one advantage of learning based methods is that their reliance on training data implicitly encodes behaviours that are typical and avoids modelling ones that are atypical. In fact, there is work that explicitly constrains the output poses to lie within a space of 'valid' poses extracted from the training data [34]. Such methods undoubtedly stabilize the pose estimates, but they are necessarily limited to tracking the types of motion that they have been trained on.

It is debatable to what extent, and how, these algorithms should be allowed to learn online and thus modify their initial inference models. There is a danger both of forgetting established results and of learning incorrect new ones. Regarding algorithms, the ability to incorporate partial supervision would be useful, but the ideal is a method that can learn on-line in an unsupervised manner, continuously improving itself as it observes more and more human motion. Developing such algorithms is challenging and in our opinion remains an open problem.

A

A MAP Approach to the Relevance Vector Machine

The Relevance Vector Machine (RVM) was originally proposed in [150, 151], where it is formulated under a general Bayesian framework for obtaining sparse solutions for problems of regression and classification. The problem addressed by it is that of learning a model to predict the value of an output $\mathbf{x}$ (which in general may be discrete/continuous and scalar/vectorial) from an input measurement $\mathbf{z}$ by making use of a training set of labeled examples $\{(\mathbf{z}_i, \mathbf{x}_i) | i = 1 \ldots n\}$. This normally involves estimating the parameters of some function defined over the input space. A common form of such a function is:

$$\mathbf{x} \;=\; \sum_{k=1}^{p} \mathbf{a}_k \, \phi_k(\mathbf{z}) + \boldsymbol{\epsilon} \;\equiv\; \mathbf{A}\,\mathbf{f}(\mathbf{z}) + \boldsymbol{\epsilon} \tag{A.1}$$

where $\{\phi_k(\mathbf{z}) \,|\, k = 1 \ldots p\}$ are the basis functions, $\mathbf{a}_k$ are $\mathbb{R}^m$-valued weight vectors (the parameters to be estimated), and $\boldsymbol{\epsilon}$ is a residual error vector. For compactness, the weight vectors can be gathered into an $m{\times}p$ weight matrix $\mathbf{A} \equiv (\mathbf{a}_1 \;\; \mathbf{a}_2 \;\; \cdots \;\; \mathbf{a}_p)$ and the basis functions into a $\mathbb{R}^p$-valued function $\mathbf{f}(\mathbf{z}) = (\phi_1(\mathbf{z}) \;\; \phi_2(\mathbf{z}) \;\; \cdots \;\; \phi_p(\mathbf{z}))^\top$. To allow for a constant offset $\mathbf{x} = \mathbf{A}\mathbf{f} + \mathbf{b}$, we can include $\phi(\mathbf{z}) \equiv 1$ in $\mathbf{f}$. Often, the unknown parameter matrix $\mathbf{A}$ is solved for by minimizing some predefined $\mathbf{x}$-space prediction error over the training set, combined with a *regularizer* function $R(\mathbf{A})$ to control over-fitting, *e.g.* using a Euclidean error measure gives

$$\mathbf{A} \;=\; \arg\min_{\mathbf{A}} \left\{ \sum_{i=1}^{n} \|\mathbf{A}\,\mathbf{f}(\mathbf{z}_i) - \mathbf{x}_i\|^2 \;+\; R(\mathbf{A}) \right\}$$

$$\equiv\; \arg\min_{\mathbf{A}} \left\{ \|\mathbf{A}\,\mathbf{F} - \mathbf{X}\|^2 + R(\mathbf{A}) \right\} \tag{A.2}$$

The RVM, in its original form [151], does not explicitly solve for $\mathbf{A}$ in this fashion. It takes a Bayesian approach by introducing Gaussian priors on each parameter or group of parameters, with each prior being controlled by its own individual scale hyperparameter, and follows a type-II maximum likelihood approach by optimizing the hyper-parameters while integrating out the parameters $\mathbf{A}$. This appendix describes an alternative algorithm that is based on a maximum-aposteriori (MAP) approach and hence not strictly Bayesian. A good discussion of the differences between these two philosophies is given in [91].
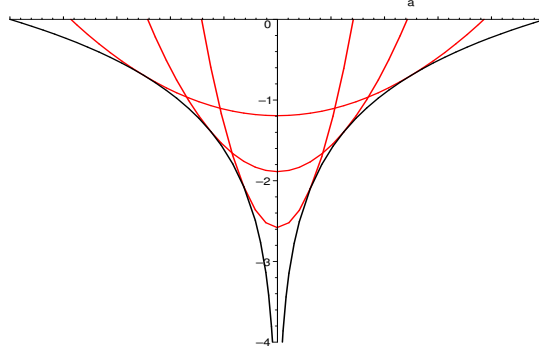
Figure A.1: *"Quadratic bridge" approximations to the $\nu \log \|a\|$ regularizers. These are introduced to prevent parameters from prematurely becoming trapped at zero.*

Following a MAP approach, the hyperpriors, which obey a power law distribution, are integrated out analytically to give singular, highly non-convex total priors of the form $p(a) \sim \|a\|^{-\nu}$ for each parameter or parameter group a, where ν is a hyperprior parameter. Taking log likelihoods gives an equivalent regularization penalty of the form $R(a) = \nu \log \|a\|$. The effect of this penalty is worth noting: if $\|a\|$ is large, the 'regularizing force' $dR/da \sim \nu/\|a\|$ is small so the prior has little effect on a. But the smaller $\|a\|$ becomes, the greater the regularizing force. At a certain point, the data term no longer suffices to hold the parameter at a nonzero value against this force, and the parameter rapidly converges to zero. Hence, the fitted model is sparse — the RVM automatically selects a subset of 'relevant' basis functions that suffices to describe the problem. The regularizing effect is invariant to rescalings of $\mathbf{f}()$ or $\mathbf{X}$. (*E.g.* scaling $\mathbf{f} \to \alpha \mathbf{f}$ forces a rescaling $\mathbf{A} \to \mathbf{A}/\alpha$ with no change in residual error, so the regularization forces $1/\|a\| \propto \alpha$ track the data-term gradient $\mathbf{A}\,\mathbf{F}\,\mathbf{F}^\top \propto \alpha$ correctly). ν serves both as a sparsity parameter and as a sparsity based scale-free regularization parameter. The complete RVM model is highly nonconvex with many local minima and optimizing it can be problematic because relevant parameters can easily become accidentally 'trapped' in the singularity at zero, but this does not prevent RVMs from giving useful results in practice. Setting ν to optimize the estimation error on a validation set, one typically finds that this form of the RVM gives sparse regressors with performance very similar to the much denser ones from analogous methods with milder priors.

A.1   RVM Training Algorithm

To train the RVM we use a continuation method based on successively approximating the $\nu \log \|a\|$ regularizers with quadratic "bridges" $\nu \left(\|a\|/a_{\text{scale}}\right)^2$ chosen to match the prior gradient at a_{scale}, a running scale estimate for a (see fig. A.1). The bridging changes the apparent curvature of the cost surfaces, allowing parameters to pass through zero if they need to with less risk of premature trapping. The complete algorithm is described below:

1. Initialize $\mathbf{A}$ with ridge regression. Initialize the running scale estimates $a_{\text{scale}} = \|\mathbf{a}\|$ for the components or vectors $\mathbf{a}$.

2. Approximate the $\nu \log \|\mathbf{a}\|$ penalty terms with "quadratic bridges", the gradients of which match at a_{scale} *i.e.* the penalty terms take the form $\frac{\nu}{2}\left(\mathbf{a}/a_{\text{scale}}\right)^2 + \text{const}[1]$.

---

[1]One can set $\text{const} = \nu(\log\|a_{\text{scale}}\| - \frac{1}{2})$ to match the function values at a_{scale} as shown in figure A.1, but this value is irrelevant for the least squares minimization.

3. Solve the resulting linear least squares problem in $\mathbf{A}$.

4. Remove any components $\mathbf{a}$ that have become zero, update the scale estimates $a_{\text{scale}} = \|\mathbf{a}\|$, and continue from 2 until convergence.

We have tested both *componentwise* priors, $R(\mathbf{A}) = \nu \sum_{jk} \log |\mathbf{A}_{jk}|$, which effectively allow a different set of relevant basis functions to be selected for each dimension of $\mathbf{x}$, and *columnwise* ones, $R(\mathbf{A}) = \nu \sum_k \log \|\mathbf{a}_k\|$ where $\mathbf{a}_k$ is the kth column of $\mathbf{A}$, which select a common set of relevant basis functions for all components of $\mathbf{x}$. The two priors give similar results, but one of the main advantages of sparsity is in reducing the number of basis functions (support features or examples) that need to be evaluated, so in the experiments we use columnwise priors, minimizing the expression

$$\|\mathbf{A}\,\mathbf{F} - \mathbf{X}\|^2 + \nu \sum_k \log \|\mathbf{a}_k\| \tag{A.3}$$

B

Representing 3D Human Body Poses

In this thesis, the pose of a person refers to the configuration of his/her body limbs in 3-dimensional space at any given time instant. This is generally specified in the form of a vector of m numbers, the values of which uniquely define the associated pose in a *fixed reference frame i.e.* it includes the orientation of the body with respect to the camera. However, translation (location of the person in the 3D world) and scale of the person are, by definition, not included in the pose vector. The value of m itself varies with different representations of the human body (*e.g.* depending on the amount of detail encoded) and also the choice of parameters, several options for which exist.

The human body is often treated as a series of rigid segments that are linked via joints and the underlying skeletal structure is very conveniently represented as a tree, an example of which is shown in figure B.1. The ratios of the segment lengths normally have fixed ranges but can vary slightly with body physique. For all experiments in this thesis, we have used dimensions obtained from motion capture.

B.1   Parametrization Options

Several different forms of parametrization of human body pose are possible. Most motion capture formats specify the relative angles at major body joints[1] along with the skeletal structure and segment lengths. An action sequence is thus represented as a time-dependent vector of angles and a fixed skeletal structure. For processing, however, these may represented directly as Euler angles or converted into quaternions or 3D coordinates.

Euler Joint Angles. A set of joint angles is a very convenient choice for pose representation as the camera view point can easily be factored out, the representation is inherently independent of scale and it is helpful in modeling motion dynamics and studying biomechanics. Also, joint angles can easily be used to transfer motion from one body (skeleton) to another. On the other hand, a common problem with angle based representations is that they wrap around at 360° and operations on them are not commutative. Unconstrained Euler angles suffer from gimbal lock problems when more than one set of rotation angles can define the same configuration, thus causing discontinuities in the representation[2]. With respect to measuring similarity between poses and their relationship

---

[1]The exact set of joints generally depends on the level of detail required by the application at hand.

[2]In practice, this is usually tackled by imposing constraints on the angles by allowing only one of the three to cover the full 360° range. In this work, we constrain θ_x and θ_z to $\pm 90°$ and allow θ_y to vary between $\pm 180°$ when computing rotations in the *x-y-z* order.
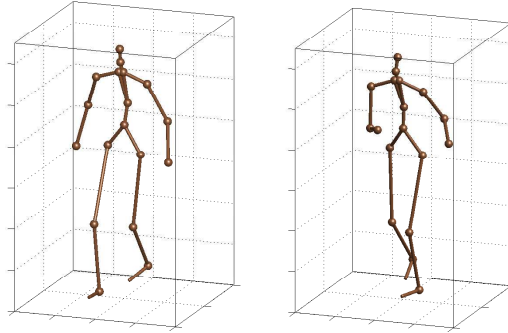
Figure B.1*: The underlying skeletal structure of the human body (here shown for a set of 18 body joints) is conveniently represented as a tree. For representing different configurations (body poses), the root of this tree is often defined as the origin of the 3D coordinate system so that the representation is invariant to translation and the limb lengths are usually normalized for invariance to scale.*

with image appearance, vectors in the form of a collection of joint angles suffer from two major disadvantages. Firstly, they must be supplemented with a notion of joint *hierarchy* (which is usually defined based on the body tree) because some joint angles have more influence on pose than others, *e.g.* a 90° rotation at a wrist joint is quite different from an equivalent rotation at a shoulder joint. Secondly, any similarity measure defined between two pose vectors must ideally take into account the instantaneous poses because the amount of change in appearance associated with a change in pose is sometimes dependent on the pose itself. For instance, consider two people standing straight, one with his arms vertically down and the other with arms stretched out horizontally. If both turn (along the vertical axis) by 90°, the change produced in the appearance is very different in the two cases, for the same change measured in pose.

Joint Angles as Quaternions. These are an alternative to using Euler angles for representing a set of joint angles. Quaternion based representations are associated with most of the advantages and disadvantages of joint angle representations discussed above. They are, however, two main advantages of using quaternions as opposed to Euler angles: *(i)* quaternions do not suffer from gimbal lock problems, and *(ii)* computing similarity two pose vectors is a much faster operation as it simply involves taking the dot product of two pose vectors.

3D Joint Coordinates. A very popular choice of representing the human body is as a vector of x-y-z locations of a set of joints. Given the skeletal structure, a set of rotation angles can be converted to this representation by performing a series of rotational transformations on the kinematic tree, starting from the 'neutral' pose. A joint coordinate based representation is extremely simple to use as the total Euclidean distance between corresponding points from two collections of joint coordinates directly captures pose difference, and correlates well with the corresponding difference in appearance (unlike angle based representations). Extracting viewpoint from a pose vector is, however, less obvious in this case. Also, unlike joint angles, 3D coordinates include scale information, so must be normalized for scale invariance. The main disadvantage of using 3D joint coordinates is that any operation on them has to make sure that the limb lengths are maintained — a set of 3D coordinates corresponding to the joint positions, in general, will not be a valid pose unless the length constraints are satisfied. In practice, these constraints are usually relaxed slightly to account for image noise and inter-person variations. Hard constraints on the limb lengths in such a representation can thus be replaced by *soft interactions* between the joints. This motivates a graphical model based representation of the body.
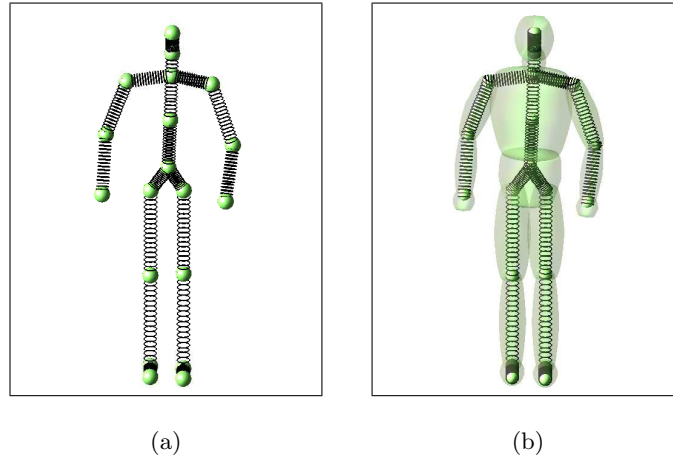
(a)   (b)

Figure B.2: *(a) The human body can be conveniently represented as a graph in which soft priors are imposed between body joints locations via the edges, here shown as springs (b) A 'unified' body model consisting of a graphical structure for bottom-up pose inference and a renderable surface that can be used for measuring image likelihoods in a top-down manner.*

Body as a Graph. A very generic representation of the human body that has recently started being explored is one using a graphical model. The joints could correspond to the nodes of the graph and edges may be used to model interaction between these joints, as shown in figure B.2(a). This enforces soft priors on the relative positions of neighbouring joints and allows for probabilistic inference on the position of each node, naturally allowing for variations in body segment ratios from person to person. The dual graph is also possible in which body limbs are considered as nodes and the articulations are modeled as interactions between the limbs (*e.g.* see [137]). In such a representation, each node is associated with position and orientation parameters. In the context of pose inference from images, this model has a very close resemblance to several part-based (pictorial structure) models used in object recognition since the appearance of each node (or limb) may be modeled independently, allowing for direct forms of conditional inference. An interesting advantage of a body graph is that it need not be restricted to the tree-like skeletal structure of the body but can also be used to model interactions between nodes that are not directly linked via the kinematic structure. This allows for the correlations between different body parts to be accommodated in a very natural manner. A short discussion on the use of a graphical representation of the human body for pose inference from images is given in chapter 9.

B.2   Rendering Pose

Given that this thesis takes a model-free approach to the pose estimation problem, pose rendering is not a part of the estimation algorithm[3]. However, we have developed a rendering routine for visualizing the results of our experiments. For rendering pose, we use a representation based on 3D joint coordinates which are usually obtained from the underlying joint angle based representation so that limb lengths are automatically read from a pre-defined skeletal structure. This choice was made so that the rendering routine may be independent of the motion capture format used — all joint angle based representations can be converted uniquely to 3D coordinates, whereas the

---

[3]In fact, the algorithm itself does not even make use of the skeletal structure associated with the joints.
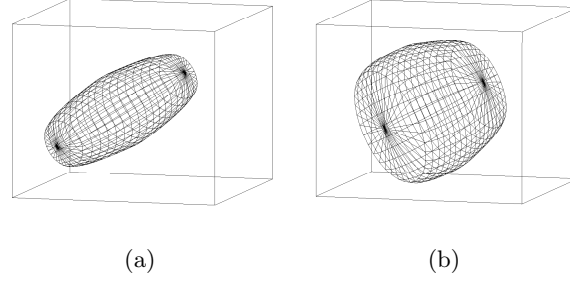
Figure B.3: *Examples of 3D surfaces that are used to render different body parts. These are formed by using exponential functions of the limb end-points along with other prespecified parameters. (a) Axially symmetric surfaces are used to render the arms, legs, neck & head, (b) non-symmetric shapes are used to render the torso parts.*
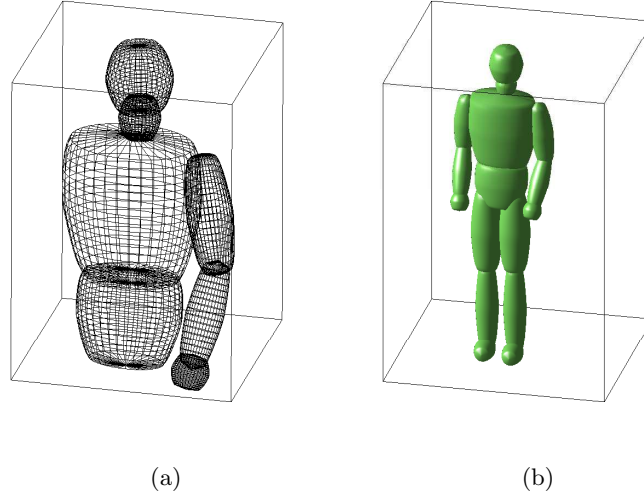

Figure B.4: *Rendering all the body parts in the common reference frame. (a) Overlaps in the individual surfaces help to create more realistic shape at the body joints (b) The complete human body model used to render estimated poses for visualizing our results.*

opposite is not true. We use a set of 18 basic joints[4] on the full body that can be extracted from both the different motion capture formats we have worked with.

The orientation and location of each body part is now defined by the two joint coordinates on either end — except for the hands that are approximated as 'directionless' spheres, and the torso parts that make use of hip and shoulder points as well. (Note that the feet make use of an additional point at the toe which is obtained using the angle at the ankle.) Exponential curves are used to define a closed surface for each limb (examples of which are shown in figure B.3 and these are all independently rendered in the common frame of reference to form the complete body model as shown in figure B.4.

---

[4]These comprise of one point each in the pelvis, sternum, neck and head, and two in each of the clavicles, shoulders, elbows, wrists, hips, knees and ankles — see figure B.1.

C

Background Subtraction

In this thesis, we perform background subtraction in video sequences to extract the foreground object in the form of a human silhouette shape (chapters 3-5). All the sequences used for this purpose have static backgrounds and hence a very basic method is used for the subtraction. Video frames containing the static background without any foreground are used to build a statistical model of the background pixel values. Each pixel is assumed to be independently sampled from a Gaussian distribution in the 3-dimensional r-g-b space of colour values and images from several frames of the background are used to compute the mean and variance of these distributions for each pixel.

Given an new image that contains an unknown foreground object, the probability of each pixel belonging to the background is computed from the Gaussian distribution for the background colour for that pixel, and a foreground probability map for the whole image is constructed by subtracting each of these quantities from unity. A foreground mask is then obtained by simply thresholding this probability map and applying a median filter for smoothing.

C.1 Shadow Removal

In most cases, we find that this simple procedure suffices to give silhouettes of reasonable quality. However, images of moving people in an indoor scene are sometimes associated with shadows, which will be marked as foreground using the method described above. To handle this we build a second foreground probability map by using normalized r-g-b values (normalized to unit L2 norm), thus removing *intensity* information and retaining only the *colour* of each pixel. Areas under shadow are characterized by a change in only intensity and not colour, so this second foreground probability map correctly assigns high background probabilities to pixels in such regions. For the final result, we multiply the two probability maps before the thresholding and median filtering steps.

Two sample silhouettes obtained with and without shadow removal are shown in figure C.1. Note however the difference is not always as pronounced as in this example since it depends on the lighting conditions. Often, shadow effects are negligible and an explicit shadow removal step is not required.
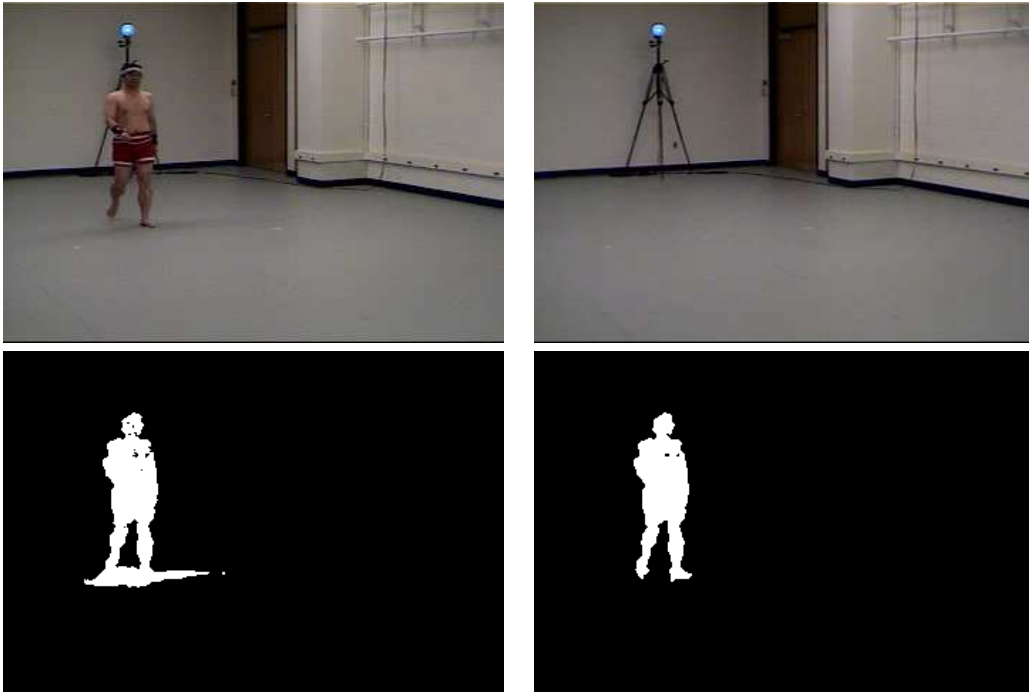
Figure C.1: *An illustration of the background subtraction process. (a) Original image (b) mean background image obtained from 30 static images over time (c) Silhouette obtained by thresholding foreground probability map (d) Silhouette obtained after shadow removal.*

Bibliography

[1] A. Agarwal and B. Triggs. 3D Human Pose from Silhouettes by Relevance Vector Regression. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2004.

[2] A. Agarwal and B. Triggs. Learning Methods for Recovering 3D Human Pose from Monocular Images. Technical report, INRIA Rhône Alpes, 2004.

[3] A. Agarwal and B. Triggs. Learning to Track 3D Human Motion from Silhouettes. In *Proceedings of the International Conference on Machine Learning*, 2004.

[4] A. Agarwal and B. Triggs. Tracking Articulated Motion using a Mixture of Autoregressive Models. In *Proceedings of the European Conference on Computer Vision*, 2004.

[5] A. Agarwal and B. Triggs. Hyperfeatures – Multilevel Local Coding for Visual Recognition. Technical report, INRIA Rhône Alpes, 2005.

[6] A. Agarwal and B. Triggs. Monocular Human Motion Capture with a Mixture of Regressors. In *IEEE Workshop on Vision for Human-Computer Interaction*, 2005.

[7] A. Agarwal and B. Triggs. A Local Basis Representation for Estimating Human Pose from Cluttered Images. In *Proceedings of the Asian Conference on Computer Vision*, 2006.

[8] A. Agarwal and B. Triggs. Hyperfeatures - Multilevel Local Coding for Visual Recognition. In *Proceedings of the European Conference on Computer Vision*, 2006.

[9] A. Agarwal and B. Triggs. Recovering 3D Human Pose from Monocular Images. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 28(1):44–58, 2006.

[10] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 26(11):1475–1490, 2004.

[11] Y. Altun, T. Hofmann, and A. Smola. Gaussian Process Classification for Segmenting and Annotating Sequences. In *Proceedings of the International Conference on Machine Learning*, 2004.

[12] O. Arikan and D. Forsyth. Interactive Motion Generation from Examples. In *Proceedings of SIGGRAPH*, 2002.

[13] V. Athitsos and S. Sclaroff. Inferring Body Pose without Tracking Body Parts. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2000.

[14] V. Athitsos and S. Sclaroff. Estimating 3D Hand Pose From a Cluttered Image. In *Proceedings of the International Conference on Computer Vision*, 2003.

[15] F. Bach and M. Jordan. Beyond Independent Components: Trees and Clusters. *Journal on Machine Learning Research*, 4:1205–1233, 2003.

[16] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition using Shape Contexts. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 24(4):509–522, 2002.

[17] A. Berg and J. Malik. Geometric Blur for Template Matching. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2001.

[18] C. Bishop. *Neural Networks for Pattern Recognition*, chapter 6. Oxford University Press, 1995.

[19] C. Bishop and M. Svensén. Bayesian Hierarchical Mixtures of Experts. In *Proceedings of Uncertainty in Artificial Intelligence*, 2003.

[20] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet Allocation. *Journal on Machine Learning Research*, 3:993–1022, 2003.

[21] A. Bosch, A. Zisserman, and X. Muñoz. Scene Classification via pLSA. In *Proceedings of the European Conference on Computer Vision*, 2006.

[22] M. Brand. Shadow Puppetry. In *Proceedings of the International Conference on Computer Vision*, pages 1237–1244, 1999.

[23] M. Brand and A. Hertzmann. Style Machines. In *Siggraph 2000, Computer Graphics Proceedings*, pages 183–192, 2000.

[24] C. Bregler and J. Malik. Tracking People with Twists and Exponential Maps. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, pages 8–15, 1998.

[25] W. Buntine and A. Jakaulin. Discrete Principal Component Analysis. Technical report, HIIT, July 2005.

[26] W. Buntine and S. Perttu. Is Multinomial PCA Multi-faceted Clustering or Dimensionality Reduction? *AI and Statistics*, 2003.

[27] M. Burl, M. Weber, and P. Perona. A Probabilistic Approach to Object Recognition Using Local Photometry and Global Geometry. In *Proceedings of the European Conference on Computer Vision*, 1998.

[28] J. Canny. Gap: A factor model for discrete data. In *ACM Conference on Information Retrieval (SIGIR)*, Sheffield, U.K., 2004.

[29] T. Cham and J. Rehg. A Multiple Hypothesis Approach to Figure Tracking. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 239–245, 1999.

[30] G. Cheung, S. Baker, and T. Kanade. Shape-From-Silhouette of Articulated Objects and its Use for Human Body Kinematics Estimation and Motion Capture. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2003.

[31] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *Proceedings of the European Conference on Computer Vision*, 2004.

[32] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2005.

[33] N. Dalal, B. Triggs, and C. Schmid. Human Detection Using Oriented Histograms of Flow and Appearance. In *Proceedings of the European Conference on Computer Vision*, 2006.

[34] D. Demirdjian, T. Ko, and T. Darrell. Constraining Human Body Tracking. In *Proceedings of the International Conference on Computer Vision*, 2003.

[35] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[36] J. Deutscher, A. Blake, and I. Reid. Articulated Body Motion Capture by Annealed Particle Filtering. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2000.

[37] G. Dorko and C. Schmid. Object Class Recognition using Discriminative Local Features. Technical report, INRIA Rhône Alpes, 2005.

[38] A. D'Souza, S. Vijayakumar, and S. Schaal. Learning Inverse Kinematics. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2001.

[39] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing Action at a Distance. In *Proceedings of the International Conference on Computer Vision*, 2003.

[40] A. Elgammal and C. Lee. Inferring 3D Body Pose from Silhouettes using Activity Manifold Learning. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2004.

[41] A. Fang and N. Pollard. Efficient Synthesis of Physically Valid Human Motion. *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, 22(3):417–426, 2003.

[42] L. Fei-Fei and P. Perona. A Bayesian Hierarchical Model for Learning Natural Scene Categories. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2005.

[43] P. Felzenszwalb and D. Huttenlocher. Pictorial Structures for Object Recognition. *International Journal of Computer Vision*, 61 (1), 2005.

[44] A. Ferencz, E. Learned-Miller, and J. Malik. Learning Hyper-Features for Visual Identification. In *Proceedings of Advances in Neural Information Processing Systems*, 2004.

[45] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning Object Categories from Google's Image Search. In *Proceedings of the International Conference on Computer Vision*, 2005.

[46] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE. Trans. Computers*, C-22(1), 1973.

[47] M. Fritz, E. Hayman, B. Caputo, and J.-O. Eklundh. On the Significance of Real-World Conditions for Material Classification. In *Proceedings of the European Conference on Computer Vision*, 2004.

[48] K. Fukushima. Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980.

[49] D. M. Gavrila. The Visual Analysis of Human Movement: A Survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

[50] Z. Ghahramani and G. Hinton. Switching State-Space Models. Technical report, Department of Computer Science, University of Toronto, 1998.

[51] K. Grauman and T. Darrell. Fast Contour Matching Using Approximate Earth Mover's Distance. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2004.

[52] K. Grauman, G. Shakhnarovich, and T. Darrell. Inferring 3D Structure with a Statistical Image-Based Shape Model. In *Proceedings of the International Conference on Computer Vision*, pages 641–648, 2003.

[53] G. Hager and P. Belhumeur. Efficient Region Tracking with Parametric Models of Geometry and Illumination. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 20(10):1025–1039, 1998.

[54] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Alvey Vision Conference*, pages 147–151, 1988.

[55] T. Heap and D. Hogg. Wormholes in Shape Space: Tracking Through Discontinuous Changes in Shape. In *Proceedings of the International Conference on Computer Vision*, pages 344–349, 1998.

[56] T. Hofmann. Probabilistic Latent Semantic Analysis. In *Proceedings of Uncertainty in Artificial Intelligence*, Stockholm, 1999.

[57] D. Hoiem, A. Efros, and M. Herbet. Geometric Context form a Single Image. In *Proceedings of the International Conference on Computer Vision*, 2005.

[58] N. Howe, M. Leventon, and W. Freeman. Bayesian Reconstruction of 3D Human Motion from Single-Camera Video. In *Proceedings of Advances in Neural Information Processing Systems*, 1999.

[59] P. Hoyer. Non-negative Matrix Factorization with Sparseness Constraints. *Journal on Machine Learning Research*, 5:1457–1469, 2004.

[60] I.Haritaoglu, D.Harwood, and L.Davis. Ghost: A Human Body Part Labeling System Using Silhouettes. In *International Conference on Pattern Recognition*, 1998.

[61] S. Ioffe and D. Forsyth. Human Tracking with Mixtures of Trees. In *Proceedings of the International Conference on Computer Vision*, pages 690–695, 2001.

[62] M. Isard. PAMPAS: Real-valued Graphical Models for Computer Vision. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2003.

[63] M. Isard and A. Blake. CONDENSATION – Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[64] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton. Adaptive Mixtures of Local Experts. *Neural Computation*, 3(1):79–87, 1991.

[65] T. Joachims. Making large-Scale SVM Learning Practical. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.

[66] N. Jojic and B. Frey. Learning Flexible Sprites in Video Layers. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2001.

[67] S. Ju, M. Black, and Y. Yacoob. Cardboard People: A Parameterized Model of Articulated Motion. In *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recongnition*, pages 38–44, 1996.

[68] F. Jurie and M. Dhome. Hyperplane Approximation for Template Matching. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 24(7):996–1000, 2002.

[69] F. Jurie and B. Triggs. Creating Efficient Codebooks for Visual Recognition. In *Proceedings of the International Conference on Computer Vision*, 2005.

[70] T. Kadir and M. Brady. Saliency, Scale and Image Description. *International Journal of Computer Vision*, 45(2):83–105, 2001.

[71] H. Kashima and Y. Tsuboi. Kernel-based Discriminative Learning Algorithms for Labeling Sequences, Trees and Graphs. In *Proceedings of the International Conference on Machine Learning*, 2004.

[72] M. Keller and S. Bengio. Theme-Topic Mixture Model: A Graphical Model for Document Representation. In *PASCAL Workshop on Learning Methods for Text Understanding and Mining*, 2004.

[73] M. P. Kumar, P. Torr, and A. Zisserman. Learning Layered Pictorial Structures from Video. In *Proceedings of the Indian Conference on Computer Vision, Graphics & Image Processing*, pages 148–153, 2004.

[74] M. P. Kumar, P. Torr, and A. Zisserman. Learning Layered Motion Segmentations of Video. In *Proceedings of the International Conference on Computer Vision*, 2005.

[75] J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic Models for Segmentation and Labeling Sequence Data. In *Proceedings of the International Conference on Machine Learning*, 2001.

[76] J. Lafferty, X. Zhu, and Y. Liu. Kernel Conditional Random Fields: Representation and Clique Selection. In *Proceedings of the International Conference on Machine Learning*, 2004.

[77] P. Lai and C. Fyfe. Kernel and Nonlinear Canonical Correlation Analysis. *Proceedings of the International Journal of Neural Systems*, 10(5):365–377, 2000.

[78] X. Lan and D. Huttenlocher. Beyond Trees: Common-Factor Models for 2D Human Pose Recovery. In *Proceedings of the International Conference on Computer Vision*, 2005.

[79] G. Lang and P. Seitz. Robust Classification of Arbitrary Object Classes Based on Hierarchical Spatial Feature-Matching. *Machine Vision and Applications*, 10(3):123–135, 1997.

[80] I. Laptev and T. Lindeberg. Space-Time Interest Points. In *Proceedings of the International Conference on Computer Vision*, pages 432–439, 2003.

[81] S. Lazebnik, C. Schmid, and J. Ponce. Affine-Invariant Local Descriptors and Neighborhood Statistics for Texture Recognition. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 649–655, 2003.

[82] S. Lazebnik, C. Schmid, and J. Ponce. Semi-local Affine Parts for Object Recognition. In *Proceedings of the British Machine Vision Conference*, volume 2, pages 779–788, 2004.

[83] H. Lütkepohl. *Introduction to Multiple Time Series Analysis*. Springer-Verlag, Berlin, Germany, second edition, 1993.

[84] Y. LeCun, F. Huang, and L. Bottou. Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2004.

[85] D. D. Lee and H. S. Seung. Learning the Parts of Objects by Non–negative Matrix Factorization. *Nature*, 401:788–791, 1999.

[86] J. Lee, J. Chai, P. Reitsma, J. Hodgins, and N. Pollard. Interactive Control of Avatars Animated with Human Motion Data. *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, 21(3):491–500, 2002.

[87] M. Lee and I. Cohen. Human Upper Body Pose Estimation in Static Images. In *European Conference on Computer Vision*, 2004.

[88] B. Leibe, E. Seemann, and B. Schiele. Pedestrian Detection in Crowded Scenes. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2005.

[89] T. Leung and J. Malik. Recognizing Surfaces Using Three-Dimensional Textons. In *Proceedings of the International Conference on Computer Vision*, 1999.

[90] D. Lowe. Distinctive Image Features from Scale-invariant Keypoints. *International Journal of Computer Vision*, 60, 2:91–110, 2004.

[91] D. MacKay. Comparison of Approximate Methods for Handling Hyperparameters. *Neural Computation*, 11(5):1035–1068, 1999.

[92] J. Malik and P. Perona. Preattentive Texture Discrimination with Early Vision Mechanisms. *J. Optical Society of America*, A 7(5):923–932, May 1990.

[93] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10), 2005.

[94] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human Detection based on a Probabilistic Assembly of Robust Part Detectors. In *Proceedings of the European Conference on Computer Vision*, volume I, pages 69–81, 2004.

[95] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2), 2005.

[96] A. Mohan, C. Papageorgiou, and T. Poggio. Example-Based Object Detection in Images by Components. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 23(4):349–361, 2001.

[97] G. Mori, S. Belongie, and J. Malik. Shape Contexts Enable Efficient Retrieval of Similar Shapes. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2001.

[98] G. Mori and J. Malik. Estimating Human Body Configurations Using Shape Context Matching. In *Proceedings of the European Conference on Computer Vision*, volume 3, pages 666–680, 2002.

[99] G. Mori and J. Malik. Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2003.

[100] G. Mori, X. Ren, A. Efros, and J. Malik. Recovering Human Body Configurations: Combining Segmentation and Recognition. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2004.

[101] D. Morris and J. Rehg. Singularity Analysis for Articulated Object Tracking. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, pages 289–296, 1998.

[102] A. Ng, M. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an algorithm. In *Proceedings of Advances in Neural Information Processing Systems*, 2001.

[103] B. North, A. Blake, M. Isard, and J. Rittscher. Learning and Classification of Complex Dynamics. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 22(9):1016–1034, 2000.

[104] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *Proceedings of the European Conference on Computer Vision*, 2004.

[105] D. Ormoneit, H. Sidenbladh, M. Black, and T. Hastie. Learning and Tracking Cyclic Human Motion. In *Proceedings of Advances in Neural Information Processing Systems*, pages 894–900, 2000.

[106] C. Papageorgiou and T. Poggio. A Trainable System for Object Detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.

[107] V. Parameswaram and R. Chellappa. View Independent Body Pose Estimation from a Single Perspective Image. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2004.

[108] V. Pavlovic, J. Rehg, T. Cham, and K. Murphy. A Dynamic Bayesian Network Approach to Figure Tracking using Learned Dynamic Models. In *Proceedings of the International Conference on Computer Vision*, pages 94–101, 1999.

[109] V. Pavlovic, J. Rehg, and J. MacCormick. Learning Switching Linear Models of Human Motion. In *Proceedings of Advances in Neural Information Processing Systems*, pages 981–987, 2000.

[110] R. Plänkers and P. Fua. Articulated Soft Objects for Video-Based Body Modeling. In *Proceedings of the International Conference on Computer Vision*, pages 394–401, 2001.

[111] J. Puzicha, T. Hofmann, and J. Buhmann. Histogram Clustering for Unsupervised Segmentation and Image Retrieval. *Pattern Recognition Letters*, 20:899–909, 1999.

[112] A. Rahimi, B. Recht, and T. Darrell. Learning Appearance Manifolds from Video. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2005.

[113] D. Ramanan and D. Forsyth. Finding and Tracking People from the Bottom Up. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2003.

[114] D. Ramanan, D. Forsyth, and A. Zisserman. Strike a Pose: Tracking People by Finding Stylized Poses. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, pages 271–278, 2005.

[115] A. Rao, D. Miller, K. Rose, and A. Gersho. Mixture of Experts Regression Modeling by Deterministic Annealing. *IEEE Transactions on Signal Processing*, 45(11):2811–2820, 1997.

[116] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[117] L. Ren, G. Shakhnarovich, J. Hodgins, H. Pfister, and P. Viola. Learning Silhouette Features for Control of Human Motion. *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, 24(4):1303–1331, 2005.

[118] X. Ren, A. Berg, and J. Malik. Recovering Human Body Configurations using Pairwise Constraints between Parts. In *Proceedings of the International Conference on Computer Vision*, 2005.

[119] M. Riesenhuber, T., and Poggio. Hierarchical Models of Object Recognition in Cortex. *Nature Neuroscience*, 2:1019–1025, 1999.

[120] R. Ronfard, C. Schmid, and B. Triggs. Learning to Parse Pictures of People. In *Proceedings of the European Conference on Computer Vision*, pages 700–714, 2002.

[121] R. Rosales and S. Sclaroff. Learning Body Pose via Specialized Maps. In *Proceedings of Advances in Neural Information Processing Systems*, 2001.

[122] S. Roweis and L. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290:2323–2326, 2000.

[123] Y. Rubner, C. Tomasi, and L.J. Guibas. A Metric for Distributions with Applications to Image Databases. In *Proceedings of the International Conference on Computer Vision*, Bombay, 1998.

[124] A. Safonova, J. Hodgins, and N. Pollard. Synthesizing Physically Realistic Human Motion in Low-dimensional, Behavior-specific Spaces. *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, 23(3):514–521, 2004.

[125] F. Schaffalitzky and A. Zisserman. Viewpoint invariant texture matching and wide baseline stereo. In *Proceedings of the International Conference on Computer Vision*, pages 636–643, Vancouver, 2001.

[126] B. Schölkpof, A. Smola, and K. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10:1299–1319, 1998.

[127] B. Schiele and J. Crowley. Recognition without Correspondence using Multidimensional Receptive Field Histograms. *International Journal of Computer Vision*, 36(1):31–50, 2000.

[128] B. Schiele and A. Pentland. Probabilistic Object Recognition and Localization. In *Proceedings of the International Conference on Computer Vision*, 1999.

[129] C. Schmid. Weakly supervised learning of visual models and its application to content-based retrieval. *International Journal of Computer Vision*, 56(1):7–16, 2004.

[130] C. Schmid and R. Mohr. Local Grayvalue Invariants for Image Retrieval. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 19(5):530–534, 1997.

[131] H. Schneiderman and T. Kanade. Object Detection Using the Statistics of Parts. *International Journal of Computer Vision*, 56(3):151–177, 2004.

[132] G. Shakhnarovich, P. Viola, and T. Darrell. Fast Pose Estimation with Parameter Sensitive Hashing. In *Proceedings of the International Conference on Computer Vision*, 2003.

[133] H. Sidenbladh and M. Black. Learning the Statistics of People in Images and Video. *International Journal of Computer Vision*, 54(1-3):181–207, 2003.

[134] H. Sidenbladh, M. Black, and D. Fleet. Stochastic Tracking of 3D Human Figures Using 2D Image Motion. In *Proceedings of the European Conference on Computer Vision*, volume 2, pages 702–718, 2000.

[135] H. Sidenbladh, M. Black, and L. Sigal. Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. In *Proceedings of the European Conference on Computer Vision*, volume 1, 2002.

[136] L. Sigal, S. Bhatia, S. Roth, M. Black, and M. Isard. Tracking Loose-Limbed People. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2004.

[137] L. Sigal, M. Isard, B. Sigelman, and M. Black. Attractive People: Assembling Loose-limbed Models using Non-Parametric Belief Propagation. In *Proceedings of Advances in Neural Information Processing Systems*, 2003.

[138] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative Density Propagation for 3D Human Motion Estimation. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2005.

[139] C. Sminchisescu, A. Kanaujia, and D. Metaxas. Top-down and Bottom-up Processing for Robust Visual Inference. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2006.

[140] C. Sminchisescu and B. Triggs. Covariance Scaled Sampling for Monocular 3D Body Tracking. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2001.

[141] C. Sminchisescu and B. Triggs. Mapping Minima and Transitions in Visual Models. *International Journal of Computer Vision*, 61(1), 2005.

[142] C. Sminchisescu and Bill Triggs. Estimating Articulated Human Motion with Covariance Scaled Sampling. *International Journal of Robotics Research*, 22(6):371–391, 2003.

[143] A. Smola and B. Schölkpof. A Tutorial on Support Vector Regression. *NeuroCOLT2 Technical Report NC2-TR-1998-030*, 1998.

[144] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. Filtering Using a Tree-Based Estimator. In *Proceedings of the International Conference on Computer Vision*, 2003.

[145] J. Sullivan and S. Carlsson. Recognizing and Tracking Human Action. In *Proceedings of the European Conference on Computer Vision*, 2002.

[146] C. Sutton, K. Rohanimanesh, and A. McCallum. Dynamic Conditional Random Fields: Factorized Probabilistic Models for Labeling and Segmenting Sequence Data. In *Proceedings of the International Conference on Machine Learning*, 2004.

[147] Y. W. Teh, M. Seeger, and M. Jordan. Semiparametric Latent Factor Models. *AI and Statistics*, 2004.

[148] J. Tenenbaum, V. Silva, and J. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290:2319–2323, 2000.

[149] A. Thayananthan, B. Stenger, P. Torr, and R. Cipolla. Learning a kinematic prior for tree-based filtering. In *Proceedings of the British Machine Vision Conference*, volume 2, pages 589–598, 2003.

[150] M. Tipping. The Relevance Vector Machine. In *Proceedings of Advances in Neural Information Processing Systems*, 2000.

[151] M. Tipping. Sparse Bayesian Learning and the Relevance Vector Machine. *Journal on Machine Learning Research*, 1:211–244, 2001.

[152] K. Toyama and A. Blake. Probabilistic Tracking in a Metric Space. In *Proceedings of the International Conference on Computer Vision*, pages 50–59, 2001.

[153] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support Vector Machine Learning for Interdependent and Structured Output Spaces. In *Proceedings of the International Conference on Machine Learning*, 2004.

[154] R. Urtasuna, D. Fleet, and P. Fua. Monocular 3-D Tracking of the Golf Swing. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2005.

[155] R. Urtasuna, D. Fleet, and P. Fua. 3D People Tracking with Gaussian Process Dynamical Models. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2006.

[156] R. Urtasuna, D. Fleet, A. Hertzmann, and P. Fua. Priors for People Tracking from Small Training Sets. In *Proceedings of the International Conference on Computer Vision*, 2005.

[157] R. Urtasuna and P. Fua. 3-D Human Body Tracking using Deterministic Temporal Motion Models. In *Proceedings of the European Conference on Computer Vision*, 2004.

[158] J. van Haateran and A. vander Schaaf. Independent Component Filters of Natural Images Compared with Simple Cells in Primary Visual Cortex. *Proc. Royal Soc. Lond.*, B 265:359–366, 1998.

[159] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[160] M. Varma and A. Zisserman. Texture Classification: Are filter banks necessary? In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition*, 2003.

[161] P. Viola, M. Jones, and D. Snow. Detecting Pedestrians Using Patterns of Motion and Appearance. In *Proceedings of the International Conference on Computer Vision*, 2003.

[162] S. Wachter and H. Nagel. Tracking Persons in Monocular Image Sequences. *Computer Vision and Image Understanding*, 74(3):174–192, 1999.

[163] O. Williams, A. Blake, and R. Cipolla. A Sparse Probabilistic Learning Algorithm for Real-Time Tracking. In *Proceedings of the International Conference on Computer Vision*, 2003.

[164] L. Zhao and L. Davis. Closely Coupled Object Detection and Segmentation. In *Proceedings of the International Conference on Computer Vision*, 2005.