TESIS DE LA UNIVERSIDAD DE ZARAGOZA

2012

5

Grégory Rogez

Advances in Monocular Exemplar-based Human Body Pose Analysis: Modeling, Detection and Tracking

Departamento

Ingeniería Electrónica y Comunicaciones

Director/es

Orrite Uruñuela, Carlos

ISSN 2254-7606

Prensas de la Universidad
Universidad Zaragoza

Tesis Doctoral

ADVANCES IN MONOCULAR EXEMPLAR-BASED HUMAN BODY POSE ANALYSIS: MODELING, DETECTION AND TRACKING

Autor

Grégory Rogez

Director/es

Orrite Uruñuela, Carlos

UNIVERSIDAD DE ZARAGOZA

Ingeniería Electrónica y Comunicaciones

2012

Instituto Universitario de Investigación
en Ingeniería de Aragón
Universidad Zaragoza
1542

Advances in Monocular Exemplar-based Human Body Pose Analysis:

Modeling, Detection and Tracking

Grégory Rogez

Ph.D. Dissertation

Advisor
Dr. Carlos Orrite Uruñuela

June 2012

Advances in Monocular Exemplar-based Human Body Pose Analysis:

Modeling, Detection and Tracking

Grégory Rogez

Ph.D. Thesis

PhD Committee Members:

Dr. D. Armando Roy Yarza	Universidad de Zaragoza, España
Dr. D. José Mara Martínez Montiel	Universidad de Zaragoza, España
Dr. D. Dariu M. Gavrila	University of Amsterdam, The Netherlands
Dr. D. Dimitrios Makris	Kingston University London, UK
Dr. D. Nicolás Pérez de la Blanca Capilla	Universidad de Granada, España

External European Reviewers:

Dr. D. Philip H. S. Torr	Oxford Brookes University, UK
Dr. D. Antonis Argyros	University of Creete, Greece

Acknowledgements

This dissertation would not have been possible without the guidance and the help of several individuals who, in one way or another, contributed in the preparation and completion of this thesis.

First and foremost, my gratitude goes to my supervisor and friend, Dr. Carlos Orrite, whose guidance, support and encouragements made this research possible. Thanks for giving me freedom to shape my research path and the time to finally submit the thesis I had in mind. Next, I would like to thank Prof. Armando Roy. This thesis would not have been possible without the FPU grant that I obtained with his help.

During the preparation of the thesis, I have had the chance to collaborate with many people. I want to thank the members of the CVLab in Zaragoza, more concretely Dr J. Elias Herrero and Dr Jesús Martínez whose earlier phd work have been the basis of the first part of this thesis. I want to thank Dr Jose J. Guerrero for his help and advices on the elaboration of the second part of the thesis.

The research work of the third part of the thesis has been done when I was visiting Oxford Brookes University. During the time I have spent working at Oxford Brookes, I have had the honour and the pleasure of meeting and working with many very talented and enthusiastic people. I want to thank Professor Philip Torr for giving me the opportunity to work in a great environment, and for his valuable advice, enthusiasm and support. I want to thank Dr Jon Rihan for the great collaboration. I will always remember our endless brainstorming sessions.

I want to thank the external reviewers of the thesis Professor Antonis Argyros and Professor Philip Torr (bis) and the member of the PhD committee for their valuable observations and suggestions. I also want to thank the anonymous reviewers who helped to improve the quality of our conference and journal papers and consequently the quality of this thesis.

Last but not least, I wish to thank my family and friends for their support, especially Dr Ruben Martínez, Dr Antonio Miguel and Dr Ignacio Martínez for their advices and help, and Jesús Senar for his continuous encouragements. And most of all, thank you Marga for your patience and for the love you gave me. The completion of my dissertation has been a long journey. It would not have been possible without you.

The author would like thank the I3A and the Department of Electronics and Communications Engineering, University of Zaragoza, for their confidence. He also wants to acknowledge support provided by: “Departamento de Ciencia, Tecnología y Universidad del Gobierno de Aragón”, “Fondo Social Europeo” and “Ministerio de Ciencia e Innovación (TIN2010-20177)”. Part of this work was also supported by the Spanish Ministry of Education under the FPU grant AP2003-2257.

Abstract

This thesis brings some contributions to one of the most active research areas in computer vision: the analysis of human body pose from monocular images. It has a broad range of potential applications in different fields such as human computer interfaces (gaming), safety (surveillance, biometrics) and biomedical (sport, motion analysis). Exemplar based techniques have been very successful for human body pose analysis. However, their accuracy strongly depends on the similarity of both camera viewing angle and scene properties between training and testing images. Given a typical training dataset captured from a small number of fixed cameras parallel to the ground, three types of testing environments with increasing level of difficulty have been identified and studied in this thesis: 1) a static camera with a similar viewing angle observing only one individual, 2) a fixed surveillance camera with a considerably different viewing angle and multiple targets and 3) a moving camera sequence or just a single static image of an unknown scene.

Each environment raises different problems that we have considered separately. Therefore, we have structured the thesis in three main parts corresponding to these three testing conditions. In the first part, we use a common static background subtraction algorithm to perform foreground detection and propose a model-based approach associating the body pose and the 2D silhouette to jointly segment and recover the pose of the subject observed in the scene. To cope with viewpoint and out-of plane rotation, local spatio-temporal models corresponding to several views and steps of the same action are trained, concatenated and sorted in a global framework. Temporal and spatial constraints are then considered to select the most probable models at each time step. The experiments carried out on indoor and outdoor sequences have demonstrated the ability of this approach to adequately segment walking pedestrians and estimate their poses independently of the direction of motion.

In the second part, we present a methodology for view-invariant monocular 3D body pose tracking in man-made environments. First, we model 3D body poses and camera viewpoint with a low dimensional manifold and learn a generative model of the silhouette from this manifold to the training views. During the online stage, 3D body poses are tracked using a recursive Bayesian sampling conducted jointly over the scene's ground plane and the pose-viewpoint manifold. For each sample, the homography relating training plane to the image points is calculated using the dominant 3D directions of the scene and used to project the regressed silhouette in the image in order to estimate its likelihood. In our experimental evaluation, we demonstrate the significant improvements of this homographic matching over a commonly used similarity transformation and provide quantitative 3D pose tracking results for monocular sequences with high perspective effect.

In the third part, we address human detection and pose estimation by formulating it as a classification problem. Our main contribution is a multi-class pose detector that uses the best components of state-of-the-art classifiers including hierarchical trees, cascades of rejectors as well as randomized forests. First, we define a set of classes by discretizing camera viewpoint and pose space. A bottom-up approach is then followed to build a hierarchical tree by recursively clustering and merging the classes at each level. For each branch of this decision tree, we take advantage of the alignment of training images to build a list of potentially discriminative HOG (Histograms of Orientated Gradients) features. We then select the HOG blocks that show the best rejection performances. We finally grow an ensemble of cascades by randomly sampling one of these HOG-based rejectors at each branch of the tree. The resulting multi-class classifier is then used to scan images in a sliding window scheme. Our approach, when compared to other pose classifiers, gives fast and efficient detection performances with both fixed and moving cameras as well as with static images. We present results using different publicly available training and testing data sets.

Resumen

Esta tesis establece una serie de contribuciones en una de las líneas de investigación más relevantes en el campo de la visión por computador: el análisis de la postura del cuerpo humano a partir de secuencias de imágenes adquiridas con una sola cámara. Esta temática presenta un amplio rango de potenciales aplicaciones entre las que se encuentran el desarrollo de nuevas interfaces persona-computador (videojuegos), sistemas avanzados de seguridad (videovigilancia, biometría) y aplicaciones biomédicas (análisis deportivo, análisis biomecánico del movimiento humano). Hasta la fecha, las técnicas basadas en patrones han tenido bastante éxito en el análisis de la postura humana. Sin embargo, su precisión depende en gran medida de la similitud del punto de vista de la cámara y de las propiedades de la escena entre las imágenes de entrenamiento y las de prueba. Teniendo en cuenta un típico conjunto de datos de entrenamiento capturado mediante un número reducido de cámaras fijas, todas ellas paralelas al suelo, en esta tesis hemos identificado y analizado tres posibles escenarios con creciente nivel de dificultad: 1) una cámara fija paralela al suelo observando un único individuo, 2) localización y seguimiento de múltiples sujetos mediante una cámara de vigilancia fija con un ángulo de visión considerablemente diferente al utilizado para capturar los datos de entrenamiento, y 3) una secuencia de vídeo capturada con una cámara en movimiento o simplemente una sola imagen estática de una escena desconocida.

Cada escenario plantea diferentes problemas que hemos considerado por separado. Por ello, hemos estructurado la tesis en tres partes que corresponden a estos tres entornos de prueba. En la primera parte, se utiliza un simple algoritmo de substracción respecto a un fondo estático para la detección del sujeto y se propone un modelo que asocia la postura 2D del cuerpo y la silueta 2D para conseguir conjuntamente la segmentación y la estimación de la postura del sujeto observado en la escena. Para hacer frente al problema del punto de vista de la cámara, y las posibles rotaciones de sujeto, se entrenan varios modelos espacio-temporales locales correspondientes a varios puntos de vista y a los diferentes movimientos básicos de la misma acción. Posteriormente, aplicamos un conjunto de restricciones temporales y espaciales con objeto de seleccionar en cada instante los modelos más probables. Los experimentos llevados a cabo tanto en secuencias de interiores como de exteriores, han demostrado la capacidad de este método para segmentar y estimar la postura de peatones independientemente de la dirección del movimiento.

En la segunda parte de esta memoria, se presenta una metodología para el seguimiento de posturas 3D invariante al punto de vista de la cámara en entornos creados por el hombre. En primer lugar, se modela la postura 3D del cuerpo y el punto de vista de la cámara en un subespacio de proyección de reducidas dimensiones. Posteriormente, se obtiene un modelo de la silueta a partir de este espacio, considerando los diferentes puntos de vista de entrenamiento. Durante el procesamiento, se realiza un seguimiento de la postura 3D mediante un muestreo bayesiano recursivo, considerando la localización del sujeto en el plano del suelo de la escena, conjuntamente con este subespacio que asocia punto de vista y

postura. Para cada muestra, se calcula la homografía que relaciona el plano de entrenamiento con los puntos de la imagen utilizando las direcciones 3D dominantes de la escena. Esta homografía se utiliza para proyectar la silueta estimada en la imagen con el fin de estimar su probabilidad de aparición. Los resultados experimentales demuestran la mejoría significativa de este emparejamiento homográfico respecto a una transformación de similaridad, comúnmente empleada, proporcionando resultados cuantitativos de seguimiento de posturas 3D en secuencias monoculares con efecto perspectivo.

En la tercera parte, abordamos al problema conjunto de detección de personas y estimación de su postura formulándolo como un problema de clasificación. La principal aportación realizada es un detector multi-clases que combina los mejores componentes de los clasificadores existentes, incluyendo árboles jerárquicos, cascadas o bosques aleatorios. En primer lugar, definimos un conjunto de clases discretizando el punto de vista de la cámara y el espacio de posturas. A continuación, se construye un árbol jerárquico agrupando y fusionando de forma recursiva las clases a cada nivel. Para cada rama de este árbol de decisión aprovechamos el alineamiento de las imágenes de entrenamiento para construir una lista de características basadas en Histogramas de Gradientes Orientados (HOGs) potencialmente discriminantes. Finalmente, creamos un conjunto de cascadas mediante un muestreo aleatorio de estos HOGs en cada rama del árbol. El clasificador multi-clases resultante se utiliza para escanear imágenes con una ventana deslizante. En comparación con otros clasificadores, nuestro enfoque permite una detección rápida y eficiente con cámaras fijas y móviles, así como en imágenes estáticas. Todos los resultados obtenidos se han llevado a cabo mediante el uso de bases de datos públicas tanto en la fase de entrenamiento como de prueba.

Conclusiones

En esta tesis se han introducido varias técnicas y algoritmos que han resultado ser muy eficaces para el análisis de posturas en diferentes escenarios. Para cada uno de ellos, hemos seguido una metodología común consistente en la revisión de trabajos anteriores, la justificación y descripción de la metodología propuesta y la evaluación experimental.

En las siguientes secciones repasaremos las contribuciones y el trabajo realizado, introduciendo las líneas futuras de investigación que de él se desprenden. En esta tesis hemos propuesto considerar un conjunto típico de datos de entrenamiento capturado mediante un número reducido de cámaras fijas, todas ellas paralelas al suelo. A continuación, se han identificado y analizado tres escenarios posibles con creciente nivel de dificultad: 1) una cámara estática observando un único individuo con un ángulo de visión similar en todas las muestras de entrenamiento y test, 2) localización y seguimiento de múltiples sujetos mediante una cámara de vigilancia fija con un ángulo de visión considerablemente diferente al utilizado en el modelado de la figura humana, y 3) una secuencia de vídeo capturada con una cámara en movimiento o simplemente una sola imagen estática de una escena desconocida.

A continuación explicamos cómo hemos cumplido los objetivos fijados y presentamos las soluciones alcanzadas. Posteriormente, analizaremos las principales aportaciones de cada una de las partes en las que hemos estructurado esta tesis y presentaremos algunas de las posibles líneas futuras de investigación.

Objetivos Fijados y Soluciones Alcanzadas

Los principales objetivos de esta tesis son el analizar y encontrar soluciones a los problemas de 1) el modelado de la postura humana y su apariencia, 2) la detección y localización de las personas presentes en la escena y 3) el seguimiento del sujeto tanto en el espacio de postura como de la imagen.

Modelado de la postura humana y de su apariencia

En esta tesis hemos seguido un método consistente en discretizar el punto de vista de la cámara alrededor del sujeto, considerando para el entrenamiento del modelo un conjunto de vistas paralelas al suelo. Para ello, se ha escogido la base de datos MoBo, considerando en total 8 puntos de vista de entrenamiento, tres de los cuales se obtuvieron mediante la duplicación de los datos de vistas simétricamente opuestas. Con objeto de modelar conjuntamente el punto de vista de la cámara y la postura humana, hemos introducido en la primera parte de esta tesis una transformación de los datos de entrada a un nuevo subespacio con forma geométrica de toroide, que ha sido empleado en posteriores capítulos. Esta representación toroidal ha demostrado tener también propiedades interesantes de visualización. En la segunda parte, hemos demostrado que podemos aprovechar la geometría proyectiva cuando el eje de la cámara no es paralelo al plano del suelo: la transformación de similitud 2D habitual entre el plano de la imagen y el modelo puede ser sustituido por una alineación basada en homografía. Los resultados experimentales han demostrado que la incorporación de esta corrección de la perspectiva en un sistema de seguimiento de posturas 3D, proporciona una mayor tasa de seguimiento y permite una mejor estimación de la postura del cuerpo con importantes variaciones del punto de vista.

Hemos considerado representaciones 2D y 3D para la postura humana que ha sido asociada al descriptor de apariencia. En primer lugar, la pose 2D y la forma 2D de la silueta han sido encapsulados en un modelo de distribución de punto (PDM), permitiendo conjuntamente la segmentación y la estimación de la postura del sujeto observado en la escena. Se ha tratado el problema de la no linealidad ajustando un modelo multi-vista basado en mezcla de gaussianas (GMM) al conjunto de datos de entrenamiento. Los modelos espacio-temporales 2D resultantes han sido posteriormente ordenados sobre la superficie del Toroide.

Para el modelado de la postura 3D hemos considerado, en el capítulo 2, el análisis de componentes principales (PCA) para la reducción de la dimensionalidad. Posteriormente, en el capítulo 5, hemos empleado un método de aprendizaje más sofisticado para mapear, de forma supervisada, el Toroide con las posturas 3D de entrenamiento y las siluetas correspondientes a los puntos de vista de entrenamiento, utilizando regresores que se han entrenado mediante una máquina de vectores de relevancia (RVM). Dado un punto en la superficie del Toroide, el modelo resultante puede generar la postura y la silueta correspondientes.

En la tercera parte de la tesis, el Toroide ha sido utilizado para definir un conjunto de clases discretizando su superficie. Cada clase se compone de imágenes de entrenamiento y posturas 2D y 3D asociadas. Hemos propuesto apoyarnos en la perfecta alineación de las imágenes de entrenamiento para construir una jerarquía de clases. En cada rama de la jerarquía, el algoritmo de entrenamiento permite seleccionar, de un espacio mucho más grande, un pequeño subconjunto de características relevantes de la clase especifica considerada. Esto hace que este planteamiento sea computacionalmente eficiente y escalable.

Detección y localización de las personas presentes en la escena

En las dos primeras partes de la tesis la detección de movimiento se ha realizado mediante un algoritmo de sustracción respecto al fondo, considerando que la cámara no se mueve y se dispone de un fondo estático. Mientras en la primera parte (capítulos 2 y 3), procesamos únicamente vídeos con un solo sujeto, en la segunda parte (capítulos 4 y 5), hemos considerado varios individuos y hemos utilizado la cabeza para detectar a las personas y resolver el problema de oclusiones y desplazamiento en grupo.

En la tercera parte de la tesis hemos estudiado los casos donde el cómputo de una imagen de fondo y, en consecuencia, la segmentación de los sujetos no es trivial. Hemos examinado el problema de la simultánea detección de la persona y la estimación de su postura. Hemos seguido un enfoque basado en una ventana deslizante para localizar y clasificar posturas humanas mediante un rápido clasificador multi-clase, que utiliza características de bordes para clasificar cada ventana testeada. Para representar dicha información de bordes se han elegido los descriptores HOG (histogramas de gradientes orientados) . El clasificador propuesto combina los mejores componentes de otros clasificadores ya existentes, tales como: árboles jerárquicos, cascadas o bosques aleatorios. Los clasificadores en cascada están específicamente diseñados para rechazar rápidamente una gran mayoría de candidatos negativos y centrarse en las regiones más prometedoras. Merced a esta propiedad hemos entrenado un conjunto de jerarquías de cascadas. Además, por muestreo aleatorio de estas características cada cascada utiliza diferentes conjuntos de características para votar, lo que añade cierta robustez al ruido y ayuda a prevenir el sobre-aprendizaje. La aleatorización y selección del número de cascadas pueden realizarse on-line sin ningún coste extra, por lo tanto, la clasificación de cada ventana puede realizarse con un conjunto diferente de cascadas. Este esquema de clasificación adaptable permite una aceleración considerable y una detección de posturas aún más eficiente que simplemente utilizando el mismo conjunto de tamaño fijo sobre toda la imagen. Cada cascada puede votar por una o más clases por lo que el conjunto genera una distribución que puede ser útil para resolver ambigüedades entre posturas.

Seguimiento de personas

En esta tesis hemos abordado tanto el problema de seguimiento del sujeto en la imagen, como el del seguimiento en el espacio de posturas.

La versión discretizada del Toroide ha sido empleada para limitar el espacio de modelos plausibles en el capítulo 3 o las clases plausibles en el capítulo 7, mediante simples restricciones espacio-temporales. La versión continua del Toroide ha sido utilizada en el capítulo 5 para muestrear posibles candidatos de postura-vista sobre la superficie del espacio basado en el filtro de partículas. Este último método ha demostrado ser más robusto a la hora de resolver ambigüedades en la postura, ya que permite mantener múltiples hipótesis a lo largo del tiempo.

El seguimiento en la imagen se ha realizado mediante un simple filtro de Kalman sobre la posición en la imagen, la escala y el ángulo en el capítulo 3 y capítulo 7, con casos fáciles donde sólo un sujeto es seguido. En la segunda parte de esta tesis, el problema ha sido simplificado mediante la introducción de la calibración de la cámara respecto a la escena ya que el seguimiento se aplica en el plano del suelo, explorando así un espacio 2D en lugar de un espacio de 4 dimensiones (posición, escala y ángulo). En el capítulo 5, hemos propuesto un sistema eficiente de filtro de partículas para seguimiento de posturas 3D en escenas de video vigilancia calibradas: el seguimiento se realizó conjuntamente en el plano del suelo y en la superficie de este subespacio Toroide que asocia punto de vista y postura. Así, sólo cuatro dimensiones necesitan ser exploradas para realizar el seguimiento de posturas de personas caminando en el espacio 3D.

Aportaciones y Líneas Futuras de Investigación

A continuación, vamos a resumir las principales aportaciones de cada sección de la tesis y enumerar algunas de las posibles líneas futuras de investigación.

Parte I: estimación de la postura de un único individuo con una cámara estática paralela al suelo

En la primera parte de la tesis hemos presentado un conjunto de modelos 2D espacio-temporales para análisis de movimiento humano. Para hacer frente a la restricción respecto al punto de vista, se han entrenado diferentes modelos locales 2D espacio-temporales correspondientes a varias vistas de la misma secuencia. Posteriormente, se han concatenado todos ellos ordenándolos en un espacio global. Al procesar una secuencia se han considerado las limitaciones temporales y espaciales para construir la matriz de transición probabilística (PTM) que da la predicción de un fotograma a otro de los modelos más probables del conjunto. Los experimentos llevados a cabo en secuencias tanto de interiores como de exteriores, han demostrado la capacidad de este método para segmentar y estimar la postura de peatones, independientemente de la dirección del movimiento. También han demostrado que el método responde muy adecuadamente a cualquier cambio de dirección durante la secuencia.

A pesar de que ha sido probado únicamente con el movimiento de andar, el enfoque presentado es genérico y puede aplicarse a cualquier otra acción. Para ello, sería necesario disponer de una base de datos más amplia, con más movimientos y mediante un modelo gráfico 3D se podría sintetizar automáticamente un conjunto de entrenamiento de representaciones 2D y 3D. En esta tesis se ha proporcionado una forma de transición entre varias vistas de una misma acción. De igual modo, podrían considerarse transiciones entre sub-espacios de diferentes actividades.

Parte II: localización y seguimiento de múltiples sujetos mediante una cámara de vigilancia fija

En la segunda parte de la tesis hemos combinado los mejores componentes de sistemas de seguimiento de posturas humanas y nos hemos aprovechado de la geometría proyectiva para desarrollar, en escenas calibradas de vigilancia, un eficiente sistema de seguimiento de posturas 3D basado en el filtro de partículas . Por medio de la geometría proyectiva hemos reemplazado la transformación de similitud 2D, comúnmente empleada para relacionar los planos de la imagen y del modelo, por una alineación basada en homografía. Asimismo, hemos propuesto un eficiente cálculo de la probabilidad de aparición de cada partícula basándonos únicamente en los bordes de la imagen y la sustracción respecto al fondo, resultando en un emparejamiento rápido de la silueta humana. También hemos introducido un nuevo estimador de Estado a partir del conjunto de partículas. La eficiencia de nuestro algoritmo ha sido demostrada mediante el procesamiento de un conjunto de vídeos de vigilancia particularmente difíciles presentando una evaluación numérica para 2784 posturas manualmente etiquetadas que serán puestas a disposición de la comunidad científica para futuras investigaciones. Los experimentos muestran que nuestro sistema es capaz de seguir correctamente varios peatones y estimar sus posturas 3D en casos de movimiento en grupo, con oclusiones y sombras.

Como trabajo futuro planteamos las siguientes líneas:

1. Una vez calibrada la cámara respecto a la escena, la cámara no puede moverse, lo que supone una limitación de la propuesta. Se podría considerar un método automático de detección de los puntos de fuga que permitiera calcular las homografías de forma completamente automática.

2. Para tratar el tema del seguimiento de múltiples sujetos interactuando hemos reponderado y reducido la influencia de las muestras usando un simple enfoque de ocupación 3D, que ha demostrado ser eficaz con los vídeos procesados en esta tesis. El problema del tema del seguimiento de múltiples sujetos en situaciones más complejas no entraba dentro de los objetivos de esta tesis y el uso de un filtro para varios objetos o un modelado más adecuado de las interacciones se plantean como trabajo futuro.

3. Aún que todos los experimentos son específicos para la actividad de caminar (debido a la mayor disponibilidad conjuntos de datos de entrenamiento y evaluación), nuestro sistema es suficientemente general para extenderse a otras actividades. La baja dimensionalidad del espacio de búsqueda, combinada con un limitado número de vistas de entrenamiento, hacen que nuestro trabajo sea fácilmente ampliable a más acciones y hace más factible el desarrollo de software de reconocimiento de actividades para aplicaciones de vigilancia reales. Para diferentes acciones podría aprenderse un modelo de baja dimensión, utilizando un mapeo para modelar las conmutaciones entre actividades.

4. De cara a encontrar la solución óptima en cada instante de tiempo en un modelo basado en el filtro de partículas, se podrían emplear técnicas de búsqueda por gradiente o un estudio

más complejo de la distribución posterior. La adaptación de nuestro método proyectivo para entornos no-calibrados ofrece otra línea interesante para futuras investigaciones.

Parte III: localización y estimación de posturas en secuencia de vídeo capturada con una cámara en movimiento o simplemente en una sola imagen estática

En la tercera parte de esta tesis, hemos abordado el problema conjunto de detección de personas y estimación de su postura formulándolo como un problema de clasificación. Hemos seguido una técnica de ventana deslizante para localizar y clasificar posturas humanas mediante un rápido clasificador multi-clase que combina los mejores componentes de clasificadores existentes incluyendo árboles jerárquicos, cascadas o bosques aleatorios.

Hemos validado nuestro método con una evaluación numérica para 3 niveles diferentes de análisis: detección y localización de personas en imágenes, clasificación de posturas humanas y estimación de la postura (con localización de las articulaciones del cuerpo). Si el espacio de búsqueda (ubicación en la imagen y escala) puede reducirse, por ejemplo, usando un algoritmo de seguimiento o limitando la distancia a la cámara en una interfaz hombre-máquina, nuestro método puede actuar en tiempo real.

Para mejorar el algoritmo planteamos varias alternativas:

1. El método actual de selección de características requiere una gran cantidad de imágenes de entrenamiento alineadas. El trabajo futuro debe centrarse en desarrollar un algoritmo de aprendizaje que pueda manejar imágenes débilmente etiquetadas o pequeños conjuntos de entrenamiento.

2. Una implementación computacionalmente eficiente del descriptor HOG, utilizando por ejemplo una implementación para GPUs, podría acelerar la detección aún más.

3. Nuestro algoritmo busca una distribución sobre las clases, una dirección interesante de trabajo futuro sería combinar esta búsqueda con algún tipo de regresión con objeto de encontrar un método computacionalmente eficiente y más preciso en términos de estimación de postura.

4. Finalmente, este trabajo abre varias otras líneas interesantes de trabajo futuro: por ejemplo, se podría intentar combinar diferentes tipos de características (color, profundidad, etc.) dentro de nuestro clasificador, extender el algoritmo para una más amplia gama de posturas y acciones o aplicar el algoritmo a otros problemas más generales de aprendizaje automático.

Contents

1 Introduction **1**

1.1 Human Pose Analysis and its Applications . . . 1

1.1.1 Surveillance Applications . . . 2

1.1.2 Control Applications . . . 3

1.1.3 Analysis Applications . . . 4

1.2 The Problem and its Difficulties . . . 5

1.2.1 The Problem . . . 5

1.2.2 The Difficulties . . . 5

1.3 State of the Art . . . 7

1.3.1 Detection . . . 8

1.3.2 Pose Estimation . . . 9

1.3.3 Tracking . . . 11

1.4 Goals and Hypotheses . . . 13

1.5 Contributions and Organization . . . 13

1.5.1 Part I: Segmentation and Pose Estimation with a Static Camera . . . 14

1.5.2 Part II: Pose Tracking in Video-Surveillance Environments . . . 15

1.5.3 Part III: Pose Estimation with a Moving Camera or in Static Images . . . 16

1.5.4 Part IV . . . 16

1.6 List of Relevant Publications . . . 16

1.6.1 Part I . . . 16

1.6.2 Part II . . . 17

1.6.3 Part III . . . 17

1.6.4 Miscellaneous: . . . 17

I Segmentation and Pose Estimation with a Static Camera **19**

2 Dealing with Non-linearities in Shape Modeling **21**

2.1 Introduction . . . 21

2.1.1 Related Work . . . 21

2.1.2 Overview . . . 23

2.2 Shape-Skeleton Training Database . . . 23

2.2.1 Training Database Construction . . . 23

2.2.2 Shapes Normalization . . . 24

2.2.3 Shape-Skeleton Eigenspace - PCA Model . . . 24

2.3 View-based Shape-Skeleton Gaussian Mixture Models . . . 25

2.3.1 Structural clustering . . . 25

2.3.2 View-based Non-Linear Models . . . 29

2.3.3 Joint Estimation of Shape and Skeleton 29
2.4 Non-Linear Models Testing 30
2.5 Conclusions 32

3 A Framework of Spatio-temporal Models 33
3.1 Introduction 33
3.1.1 Overview of the work 33
3.2 Framework Construction 35
3.3 Constraint-based search 37
3.3.1 Markov Chain for Modeling Temporal Constraint 37
3.3.2 Modeling Spatial Constraint 37
3.3.3 Combining Spatial and Temporal Constraints 38
3.4 Joint Segmentation and Pose Estimation 39
3.5 Experiments 40
3.5.1 Framework Validation 41
3.5.2 Numerical Evaluation with HumanEva dataset 45
3.6 Summary 46

II Pose Tracking in Video-Surveillance Environments 49

4 View-invariant Motion Analysis using View-based Models 51
4.1 Introduction 51
4.1.1 Related Work 52
4.1.2 Motivation and Overview of the Approach. 54
4.2 Geometrical Considerations in Man-Made Environments 55
4.2.1 Notations 55
4.2.2 Camera and Scene Calibration 56
4.3 Projection Image-Training View Through a Vertical Plane 57
4.3.1 Qualitative Results using Ground Truth Data 62
4.4 View-invariant Pose Analysis 63
4.5 Experiments 65
4.5.1 Numerical Evaluation of the Effect of Noise 65
4.6 Conclusions 67

5 View-invariant 3D Pose Tracking 69
5.1 Introduction 69
5.1.1 Related Work 69
5.1.2 Overview 70
5.2 Torus Manifold for Pose and Appearance Modeling 71
5.2.1 Body Pose Modeling 71
5.2.2 Appearance Modeling 72
5.3 Recursive Bayesian Sampling 73
5.3.1 Formulation. 73
5.3.2 Dynamic Model. 74
5.3.3 Image Measurements - Observation Model. 74
5.3.4 Tracking Multiple Pedestrians. 77
5.4 Experimental Results 80
5.4.1 Settings and Parameters. 81
5.4.2 Experiments. 81

5.5 Conclusions . . . 88

III Pose Estimation with a Moving Camera or in Static Images 91

6 Multi-class Pose Classifier 93
6.1 Introduction . . . 93
6.1.1 Related Previous Work . . . 93
6.1.2 Motivation and Overview of the Approach . . . 94
6.2 Sampling of Discriminative HOGs . . . 96
6.2.1 Formulation . . . 97
6.3 Randomized Cascades of Rejectors . . . 100
6.3.1 Bottom-up Hierarchical Tree Construction . . . 100
6.3.2 Discriminative HOG Blocks Selection . . . 102
6.3.3 Randomization . . . 105
6.4 Class Definition . . . 107
6.4.1 HumanEVA Dataset . . . 107
6.4.2 MoBo Dataset . . . 109
6.5 Experiments using MoBo Dataset . . . 112
6.6 Conclusions . . . 116

7 Human Localization and Pose Estimation 119
7.1 Introduction . . . 119
7.2 Human Pose Detection . . . 119
7.3 Properties of the Random Cascades Classifiers for Pose Detection . . . 121
7.4 Experiments . . . 123
7.4.1 Localization Results using Mobo Dataset for Training . . . 123
7.4.2 Pose Estimation in Video Sequences . . . 134
7.5 Conclusions . . . 136

IV Discussions and Conclusions 139

8 Conclusions 141
8.1 Introduction . . . 141
8.2 Conclusions . . . 141
8.2.1 Modeling . . . 141
8.2.2 Detection . . . 142
8.2.3 Tracking . . . 143
8.3 Main Contributions and Future Lines of Work . . . 143
8.3.1 Part I . . . 144
8.3.2 Part II . . . 144
8.3.3 Part III . . . 145

V Appendix 147

A Projection to a Vertical Plane 149

List of Figures

1.1 Surveillance applications . . . 2
1.2 Human computer interfaces . . . 3
1.3 Pose analysis applications . . . 4
1.4 Examples of human pose and appearance modeling . . . 10
1.5 Testing environments studied in this thesis . . . 13
1.6 Viewpoint parameterization and training views . . . 14

2.1 MoBo training database . . . 24
2.2 Gait cycles in pose eigenspace . . . 26
2.3 Structural clustering in pose eigenspace . . . 27
2.4 Gait cycle phases . . . 28
2.5 View-based mixtures of PCA . . . 29
2.6 Principal modes of variation of the gaussian models for each view-based GMM . 30
2.7 Non-linear models testing . . . 31
2.8 Caviar sequences selected for testing . . . 31

3.1 Framework of spatio-temporal shape-skeleton models . . . 34
3.2 Training views considered for the framework construction . . . 35
3.3 Multi-view Gaussian mixture model represented in the pose-silhouette eigenspace 36
3.4 Toroidal transition matrix . . . 37
3.5 Shape model fitting . . . 39
3.6 Examples of sequences processed with the framework of pose-silhouette models . 41
3.7 Challenging frames processed with the framework of pose-silhouette models . . . 41
3.8 Results obtained for the outdoor “Walkcircle” sequence . . . 42
3.9 Results obtained for the indoor “Elevator” sequence . . . 43
3.10 Segmentation results . . . 44
3.11 Pose estimation results . . . 45
3.12 Segmentation and 2D pose estimation obtained for the 4 HumanEva testing sequences . . . 46
3.13 Numerical results obtained for the 4 HumanEva testing sequences . . . 47

4.1 Training and testing video sequences considered in the chapter . . . 52
4.2 Viewpoint parameterization . . . 53
4.3 Tracking system diagram . . . 55
4.4 camera and scene calibration . . . 57
4.5 Examples of projection on vertical planes . . . 58
4.6 Schematical representation of the transformation between training and testing images through a vertical plane . . . 59

4.7 Projection to vertical plane . . . 61
4.8 Examples of projections to training planes . . . 62
4.9 View invariant tracking based on head detector . . . 64
4.10 View-based shape registration and pose estimation . . . 65
4.11 Example of a processed sequence . . . 66
4.12 Effect of noise on 2D pose estimation . . . 68

5.1 Tracking system flowchart . . . 71
5.2 Training 3D pose and shape and the pose-viewpoint torus manifold . . . 72
5.3 Likelihood function as a Laplacian distribution over a distance . . . 76
5.4 Foreground likelihood . . . 76
5.5 Likelihood values in 2 sequences . . . 77
5.6 Visualization of the likelihood for the sample set in several frames . . . 78
5.7 Percentage of lost tracks vs number of particles for similarity and homographic alignment . . . 82
5.8 Tracking results: percentage of valid localizations and 2D pose errors . . . 84
5.9 Detailed performances w.r.t. the distance to the camera . . . 85
5.10 Average 2D pose error varying the state estimator . . . 86
5.11 Qualitative 3D pose tracking results . . . 87
5.12 Qualitative 3D pose tracking results for sequences with two interacting subjects . 88
5.13 Qualitative 3D pose tracking results for sequences with three interacting subjects 88

6.1 Random Forest preliminary results . . . 95
6.2 Log-likelihood ratio for human pose . . . 96
6.3 Log-likelihood ratio for face expressions . . . 98
6.4 Selection of discriminative HOG for facial expressions classification . . . 99
6.5 Bottom-up hierarchical tree construction . . . 100
6.6 Example of a selected HOG block . . . 103
6.7 Rejector branch decision . . . 105
6.8 Diagram of image classification using an ensemble of n_c randomized cascades . . 107
6.9 Alignment of training images . . . 108
6.10 Class definition - HumanEVA . . . 109
6.11 Image classification using an ensemble of randomized cascades . . . 110
6.12 Modified MoBo dataset . . . 111
6.13 Class definition for MoBo dataset . . . 112
6.14 Set of classes obtained for the MoBo dataset . . . 113
6.15 Pose classification results on MoBo dataset for PSH, Random Forest (1000 trees) and SVMs classifiers . . . 114
6.16 Pose classification rates on MoBo datset for 4 different ensembles . . . 114
6.17 Pose classification experiments on MoBo using randomized cascades . . . 116

7.1 Cascade classifier response for a dense scan using different ensembles . . . 120
7.2 Detection score with on-line randomization . . . 122
7.3 Localization dataset . . . 123
7.4 HOG feature scaling . . . 124
7.5 Detection results on the localization dataset using different classifiers . . . 125
7.6 Detection results for different configurations of the cascades classifiers with a 3^{rd} pass of hard negative retraining . . . 127
7.7 Detection time for different configurations of the cascades classifiers after a 3^{rd} pass of hard negative retraining . . . 127

7.8 On-line randomization and filtering . . . 128
7.9 Detection results for different configurations of the cascades classifiers with a 3^{rd} pass of hard negative retraining and on-line randomization . . . 129
7.10 Selection of the best cascade classifier . . . 130
7.11 Detection rate of the selected cascade classifier . . . 131
7.12 Examples of multiple multi-scale detections in the same image . . . 131
7.13 Cascades detection errors - False Positives . . . 132
7.14 Cascades detection errors - False Negatives . . . 132
7.15 Cascades pose detections - True Positives . . . 133
7.16 Pose detection results on HumanEva II Subject S4-Camera C2 . . . 134
7.17 Pose detection results on HumanEva II dataset Subject S2-Camera C1 . . . 135
7.18 Scanning a frame from a moving camera . . . 136
7.19 Pose detection result with a moving camera . . . 137

A.1 Horizontal vanishing point localization . . . 150

List of Tables

3.1 Segmentation Results for Outdoor “Walkcircle” Sequence. 44
3.2 Segmentation Results for Indoor “Elevator” Sequence. 45
3.3 2D Pose Average Error on HumanEVA data set. 46

4.1 MoBo training angles 60

5.1 Caviar sequences considered for testing 80
5.2 Percentage of tracking failure with a similarity transformation 83
5.3 Percentage of tracking failure with the proposed homographic projection 83
5.4 Average 2D pose error 86

6.1 Classifiers training time 115

7.1 Detection times after hard negative retraining 124
7.2 Detection rates 126
7.3 2D pose estimation error on HumanEva II dataset 135

8.1 Comparison with state-of-the-art work. 143

List of Algorithms

1 $(\mathbf{s}, \mathbf{k}, B, \theta, index) = SegmentationPoseEstimation\ (B, I, \theta, index, nIter)$ 39

2 blobsList = $AdaptiveBackgroundSubtraction(B, I, \mathbf{s}, n, nIter)$ 40

3 Projection Image to Training View. 62

4 Human tracking based on head detector 64

5 Particle Filter Algorithm 79

6 Class Hierarchy Construction. 101

7 HOG Blocks Selection 104

1

Introduction

> *"Computer vision is the science and technology of machines that see."*

Computer vision refers to a recent but broad field of research whose goal is to make the computers understand what they see. At the crossroads between computer sciences, electrical engineering and mathematics, it is closely related to other research areas such as pattern recognition or machine learning, and can be seen as a branch of the Artificial Intelligence (AI) field.

The main concern of computer vision is to address the theory and technology for building artificial systems that can obtain information from images. The frames of a video sequence, the views from several cameras, or the multi-dimensional information from a medical scanner are some examples of the multiple forms that image data can take.

Over the last decades, the rapid progresses in imaging sensor technologies, the advances in data storage and transmission, and the exponential growth of computational power have all contributed to convert video into an ubiquitous and unavoidable media in modern life. The increasing quantity of available data has led to the recent interest for computer vision because of the possibilities offered by automatic video analysis. Indeed, computer vision systems should be able to automatically and quickly extract information from an image, or a sequence of images, and generate a description of the objects and actions observed in the scene.

Among all the possible objects observable in a video sequence, humans are of special interest since they play a major role in many activities. The part of computer vision dedicated to humans is usually known as human motion analysis. It intends to extract information such like presence/absence, position, posture, behaviors or activities from a single or multiple images. The monocular analysis of human motion presents a wide range of potential applications and is consequently one of the most active research areas in the field.

1.1 Human Pose Analysis and its Applications

Full-body human pose analysis from monocular images constitutes one of the fundamental problems in Computer Vision as shown by the recent special issue of the International Journal of Computer Vision [Sigal and Black, 2010]. It has a wide range of potential applications which can be classified into 3 main categories: surveillance, control and analysis.

1.1.1 Surveillance Applications

Surveillance applications are certainly the first applications that come to mind when talking about human motion analysis. In recent years, the number of cameras deployed for surveillance and safety in urban environments, such as streets, airports, subways, train stations or commercial center, has increased considerably. The main reasons are the potential terrorist menaces and, certainly, their falling cost.

While a constant and effective human monitoring of these numerous cameras seems difficult to achieve, automatic video understanding systems could enable a single operator to monitor many cameras and control wide areas more reliably. Such applications could, for instance, detect abnormal activities (fights, thefts or left-luggage) and provide timely alarm. They could also help to track eventual suspects, count people or analyze crowd flow. Some people even proposed to use gait analysis as biometric for people identification. A computer vision application can be considered to automate the acquisition and analysis of consumer shopping behavior in commercial centers. Taking full advantage of these video-surveillance networks, the processing of hours and hours of consumer behavior could provide a valuable feedback to the retailers with scarcely any additional costs. Finally, some researchers have been working on home care applications for elderly people. Population aging in developed countries causes changes in living arrangements resulting in increasing number of older people living alone. Yet vision-based home care systems have tended to focus on those elderly living alone, the main goal being to detect abnormal situations - mainly fall detection - and call for assistance when required. As the reader will notice soon, this last example provides a good transition with the next category of applications.

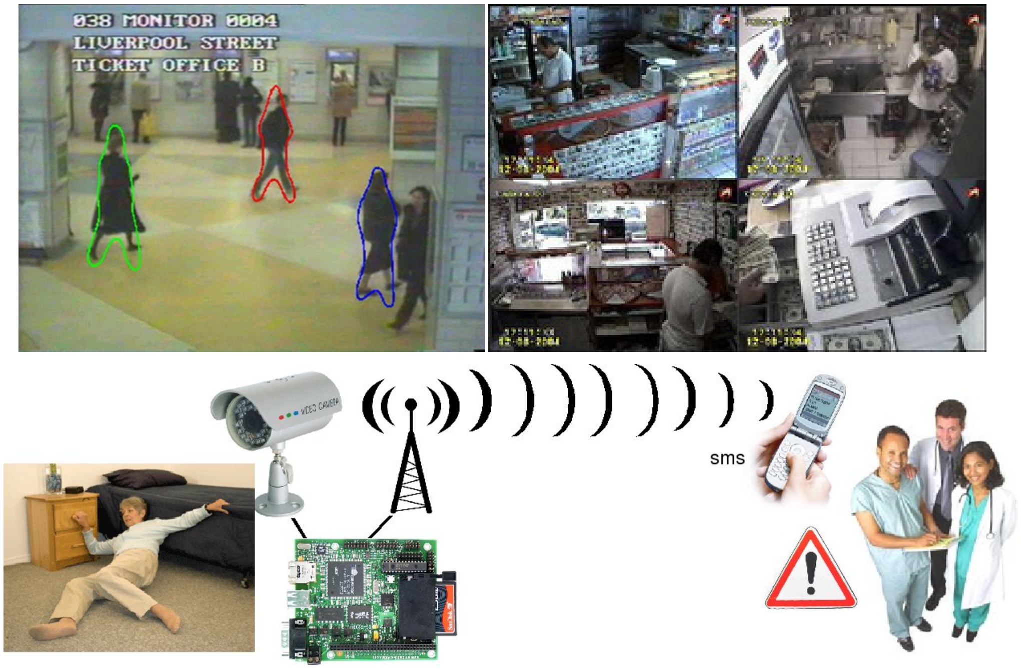


Figure 1.1: Surveillance applications: *(upperleft)* results from research project ADVISOR (Annotated Digital Video for Surveillance and Optimised Retrieval) in which is developed an integrated visual surveillance and behaviour analysis system based on the people tracker from [Baumberg, 1995]. *(upper right)* Multi-camera surveillance system in a shop. *(bottom)* A camera system for fall detection of elderly people living alone in their own private homes (image source: http://www.mobilab-khk.be).

1.1.2 Control Applications

These applications refer to the Human-Computer Interfaces, or Human-Computer Interaction (HCI), where the user interacts with a machine and controls it through particular gestures and movements. Video-games like EyeToy[1] or Kinect[2] and virtual reality are some of the few examples that one can encounter nowadays but new paradigms for interacting with computers are being investigated. They will enable in the future to communicate and interact effortlessly and intuitively with the machines.

To interact seamlessly with people, HCI systems will need to understand their environment through vision and auditory sensors, and should learn how to adapt themselves and intelligently respond depending on the context. One can remember the HAL 9000 Computer, the non-human and central character in the futuristic film by Stanley Kubrick and Arthur C. Clarke - 2001: A Space Odyssey. Its purpose is to watch over the space craft "Discovery I" by controlling all of the relevant ship's functions and by monitoring each event happening on the ship. HAL is able to localize the craft's members, identify them, know their emotions and recognize their activities. This constitutes a good example of what an Intelligent Environment could be even if, in this case, perhaps HAL becomes a bit too smart...

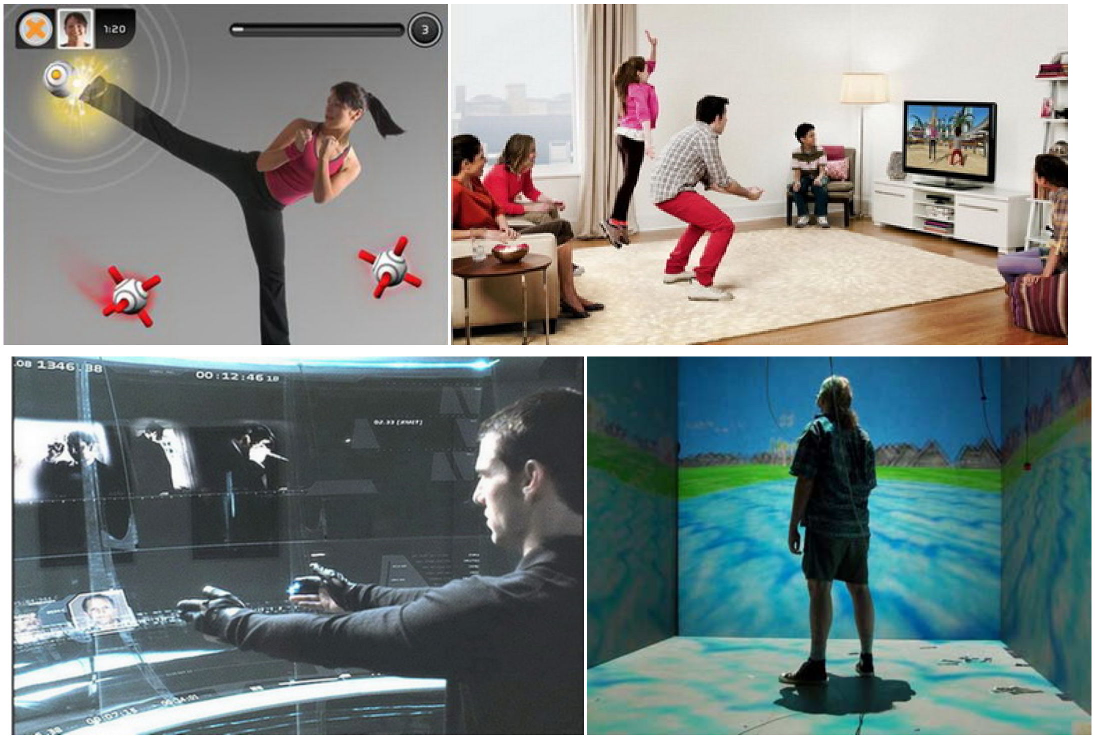


Figure 1.2: Human-Computer Interfaces: *(upper left)* Sony Eyetoy video game. *(upper right)* Family playing a MicroSoft Kinect video game. *(bottom left)* Example of a human computer interface using gesture recognition in the science fiction movie Minority Report. *(bottom right)* Virtual reality application.

[1] Sony's EyeToy is a small digital camera that sits on top of the TV and plugs into the Sony's PlayStation. The motion sensitive camera films the player as he stands in front of his TV, putting his image on screen in the middle of the action. The player can then use any part of his body to play the games.

[2] Microsoft Kinect, released in late 2010, is an interactive game which can track the joints of up to two people simultaneously using a 3D time-of-flight camera.

1.1.3 Analysis Applications

This part basically regroups all the applications that do not belong to any of the previous two categories. First, there are a non-negligible number of existing applications which require motion capture data [3]. These applications include gait analysis and diagnostic of orthopedic patients, study of athletes's movements and performances, or character animation and special effect making for the film industry. Until now, the estimation of the motion has been performed by motion capture systems that provide the 3D position/orientation of a set of markers using mechanical, electro-magnetic, acoustic or optical features. One can easily imagine how these applications could perform, in a similar way, using motion capture data from computer vision systems. Such video-based marker-less systems would present the advantage to be much cheaper and less invasive than the current marker based ones. Moreover, no specific hardware would be required since video from ordinary cameras could be used as input.

Some "content-based" applications also appear in this same category. They basically require the extraction and interpretation of the information present in the images for later processes. Automatic video annotation/indexing for fast content-based retrieval and image/video compression for efficient data storage and transmission are some clear examples.

Finally, research is being done by the car industry for safety applications: inside the car, by controlling the driver behaviors (sleeping detection, attention control or hand position on the steering wheel), and outside the car to prevent eventual accidents by detecting pedestrians crossing the street.

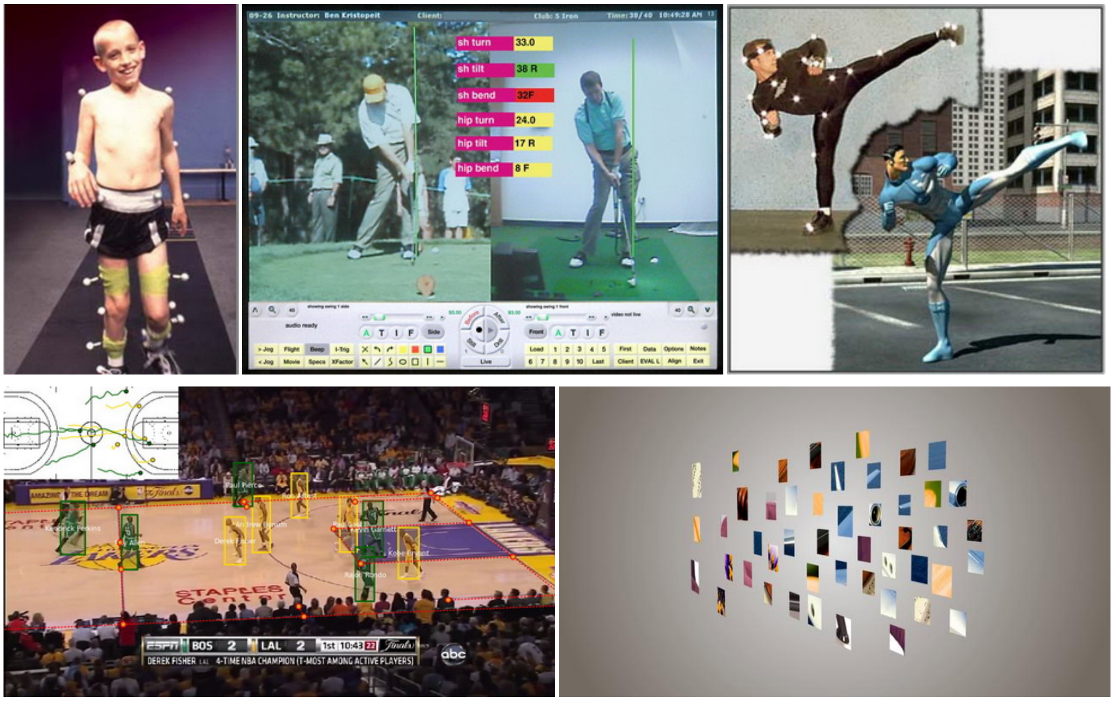


Figure 1.3: Pose analysis applications: *(upper left)* Diagnostic of an orthopedic patient with an optical motion capture system. *(upper center)* Performance analysis in golf. *(upper right)* Character animation. *(bottom left)* Sport event analysis (image source: http://www.cs.ubc.ca) *(bottom right)* Automatic video annotation.

[3] Motion capture is the process of recording real life movement of a subject and converting it into usable mathematical terms by tracking a number of key points in 3D space over time.

1.2 The Problem and its Difficulties

As well as presenting a wide spectrum of potential applications, the monocular analysis of human motion also constitutes one of the fundamental but unsolved problems of computer vision. Even if great advances have been achieved with the recently released MicroSoft Kinect, it is still an open problem as this videogame assumes a front-facing subject at a restricted distance from the TV. Moreover, the use of a 3D time-of-flight camera makes the system very sensitive to adverse lighting conditions.

In this thesis, we focus on monocular sequences captured by a standard 2D video camera. By restricting the problem to a monocular observation - supposing that only one camera view is available - we make it more attractive both in terms of research and development. Indeed, extracting 3D information from 2D data is a challenging theme of research while, for practical consideration, it is often the case where only a single camera is available.

In this section, a brief overview of the problem will be given as well as a detailed description of the involved difficulties.

1.2.1 The Problem

The problem addressed in this thesis is the consecutive detection, pose estimation and tracking of humans in monocular sequences, letting for future work the eventual recognition and interpretation of the observed activities.

The first step in almost every system of vision-based human motion analysis is the human detection. It aims at localizing the possible humans present in the scene. The goal of the segmentation step is to extract the regions corresponding to detected people from the rest of the image. These two first steps can be associated in a foreground segmentation algorithm that basically detects the region belonging to some eventual moving objects. The pose estimation task then relies on recovering the joints position (or angles) from image features while the tracking stage brings some temporal consistency by establishing coherent relations between consecutive frames of a video sequence. Location in the image, position on a map or even the posture of the observed subject are some possible features that can be tracked over time. When an estimation can not be made robustly from direct evaluation of image features (e.g. noisy measurement), the tracking stage can help to solve pose ambiguities by using the estimates from the previous frames.

Although the work is focused on sequence processing, the frames are considered to be available one at a time, in contrast to batch approaches that estimate human state at any given time using all the available images, prior and posterior to that time step. Such hypothesis constrains the tracking in some sense but allows a real-time implementation of the algorithms.

In [Zhang, 2006], the author divides the human body analysis into 3 sub-categories depending on the relative distance between camera and subject. It is in fact relative to the size of the humans in the image and the quantity of information available. The three sub-categories are far, medium and near fields. In this thesis, only an extended medium field will be considered. Depending on the application, i.e. typical video-surveillance or motion capture scenario, the full body of the observed humans will be from 30 to 300 pixels tall. Obviously, the expected accuracy in terms of segmentation, pose estimation or tracking will strongly depends on the application and the available image resolution.

1.2.2 The Difficulties

The monocular analysis of human motion involves a series of difficulties that are listed and discussed below. Some are due to the hypothesis of a monocular observation, others are inherent to the problem of observing a human and others are somehow generic to any vision-based problem.

2D-3D Projection and Depth Information: the projection of the 3D world into a 2D image suppresses the depth information. This is one of the fundamental problem of computer vision. Part of the problem can be solved by considering various camera views adequately located. In monocular case, the depth information lost during the 2D projection must be replaced by learned knowledge information about the 3D scene and the objects (their shape, structure and dynamics). Indeed, when a human looks at a 2D picture or only sees with one eye, he/she is still able to interpret the scene and recognize millions of objects and can even accurately estimate their 3D position and 3D pose from this 2D projection. This is made possible thanks to the learned knowledge his/her brain has accumulated over the years. As stated in [Bowden, 1999], providing a similar knowledge of a small subset of objects to a computer is the premise of *model based vision*.

High Variability in Shape, Appearance and Pose: because of its articulated nature, the human body can take a very high number of different poses that directly introduce a high variability in people shape and appearance. The considerable variability of both shape and appearance observable in images also results from other parameters that must be taken into account:

- shape and appearance may vary dramatically from person to person, depending on the **morphology** (short vs tall, fat vs slim) and **clothing type** (loose vs fitted clothes, dress vs pants).
- shape and appearance of a same subject can also vary over time because of **clothing and illumination** changes.
- finally, the shape and appearance of a same subject in the same pose usually present some remarkable differences when observed from different **camera viewpoints**.

Dimensionality. Human motion analysis, and more particularly human pose estimation and tracking, raises the problem of what really needs to be estimated. What is the optimum representation of a human body that a computer can learn and accurately estimate? The simplest 3D modeling of the human body relies on representing the limbs as rigid elements connected to each other at the joints that are then parameterized by their angles or 3D position. Such limited representation explains reasonably well the human motion but in any case allows to recover the real flexibility of the body. Even so, it still requires a minimum of 30 parameters!

The previous remark means that pose estimation has to be computed in a parameter space whose dimension is equal or higher than 30. Additionally, the previously emphasized high variability in shape and appearance introduces more complexity in modeling and estimation that rapidly make the problem computationally unsolvable. Some compromises must be made when representing the human body - appearance, pose and shape - to balance modeling complexity and computational feasibility.

Non-linearity. As demonstrated before, a feature space resulting from human pose, shape and/or appearance would present an extremely high dimensionality. It would also show a relatively high nonlinearity, mainly caused by the complex rotational deformations inherent to the articulated structure of the human body. In such conditions, modeling, estimation and tracking are much more challenging than with a linear problem.

Occlusions are most of the time a direct consequence of the monocular observation. Two types of occlusions may occur along a sequence. The first ones correspond to the occlusions of the observed subject, complete or local, provoked by some other elements of the scene that are located between the subject and the camera. Such occluding objects can be immobile - a wall, a tree or a post - or mobile, like a moving vehicle, an interacting subject or a simple passer by. The second category of occlusions refers to the self-occlusions. Frequently observed, they are inherent to the articulated nature of the human body. In those cases, some parts of the body are basically not visible in the image because they are occluded by some other parts of the body itself. Note that, additionally, permanent occlusions can be due to the type of clothes the observed subject is wearing: for instance, one can easily imagine how the legs can be hidden by a long coat or a long dress. Once more, thanks to the learned knowledge about human body shape and structure, *model based vision* should be able to solve part of the occlusion issue.

Image clutter, shadows and motion blur can be seen as the noise introduced by the environment settings, respectively the background, the lighting sources and the camera type.

First, image clutter refers to the distracting objects and structures in the background that can present some similarities with part of the subject in terms of shape or appearance. For instance, the position and pose of the subject's legs would be much more difficult to estimate if the color of the pants is similar to the color of the background or if there is an object with a similar shape. Those distracters can introduce uncertainty during the different steps of the motion analysis process.

Cast shadows, reflections or lighting changes are some of the possible artifacts that the lighting conditions can provoke. They can be interpreted as moving objects and can also distract the algorithms from their target.

Finally, motion blur appears when the shutter time of the camera is too long compared to the velocity of the captured motion. It introduces an additional difficulty when analyzing relatively fast motions.

In controlled environment, the different noise effects described before can be minimized by selecting an adequate background (cf blue screening technique employed by the film industry [4]), some optimal light sources and a camera adapted to the problem. In that case, the problem is substantially easier to solve since the researchers can focus their efforts on the remaining difficulties. But when the environment is imposed and not controlled by the operator or user, as in many video-surveillance or video game scenarios, some vision based solutions must be found to deal with the resulting noise.

1.3 State of the Art

The large number of related papers in journals and conferences, the number of special issues in journals and the numerous workshops dedicated to human motion capture and analysis demonstrate that it is a very active area of research in computer vision.

In this thesis, various different aspects of the human pose analysis will be dealt with and our work is closely related to several very active lines of research in human motion analysis such as shape-based analysis (part I and II), view-invariant understanding (part II), monocular 3D body pose tracking (part II), human detection, tracking-by-detection (part III). There has been a significant number of interesting papers dedicated to each one of them. Listing and explaining in this chapter all these publications would not be very useful to the reader.

Instead, we will give now a brief overview of the current state of the art, explaining only the few main methodologies for each step of the human motion capture and pose analysis. We will present the global strategies, their advantages and drawbacks, and group the efforts that consider similar underlying assumptions. Then, at the beginning of each chapter, a short but detailed section will be dedicated to the previous works directly related to the line of research

[4] Blue-screening is the process of removing blue from a scene. An actor stands in front of a blue background. Then the blue is replaced with another scene or picture.

investigated in that chapter. In this way, we aim at helping shed light on previous research, topic by topic, as well as positioning our own work in the context of related approaches.

Readers interested in seeing an extensive description of all the different relevant works dedicated to human motion capture and analysis are invited to read the numerous review papers including [Gavrila, 1999, Moeslund and Granum, 2001, Wang et al., 2003, Moeslund et al., 2006, Poppe, 2007] or the recent book by [Moeslund et al., 2011].

1.3.1 Detection

As mentioned above, the detection stage aims at localizing the possible humans present in the scene. A reliable and robust detection is essential since all the following steps strictly depend on it. An efficient human detector is very helpful for providing reliable inputs to both tracking and pose estimation algorithms. Indeed, it makes plausible the numerous pose estimation works which just consider that the bounding box of the subject is provided. In our opinion, most of the human detection techniques can be grouped into two main categories. In the first class, we can find the techniques that separate the foreground from the background and classify the resulting detected objects as human or non-human. The other principal category corresponds to the approaches which basically extract multiple windows of pixels from the image (varying location, scale and eventually rotation) and classify them as human or not.

1.3.1.1 Foreground Detection and Classification

Many previous works proposed to use a foreground segmentation and object classification to detect eventual humans in the scene. [Baumberg and Hogg, 1994, Haritaoglu et al., 2000, Isard and MacCormick, 2001, Elgammal et al., 2002, Siebel and Maybank, 2002a, Zhao and Nevatia, 2004, Orrite-Uruñuela et al., 2004] are some few examples.

Most of the time, a background model is used to estimate the foreground regions: pixels that do not correspond to the background model are considered as foreground. The resulting "blobs" are then classified as human or not, based on different criterions, mainly their size and shape. This method consequently supposes that the scene and the possible size of the humans in the image are known. In controlled environment, this method provides a very good estimation of the human segmentation. Moreover, because of its relative ease of implementation and its quick computation time, it is usually preferred for real-time applications such as video-surveillance or gaming (Eyetoy). Even so, this method suffers from serious drawbacks. First of all, it is extremely sensitive to illumination artifacts like shadows, lighting changes and reflections. It also requires a fixed camera and is not directly applicable to moving camera applications (or static isolated images). Finally, the binary classification alone - background vs foreground - is not always a sufficient feature for solving partial occlusions between multiple moving objects. In other words, when two subjects are too close, their segmentations are merged into a unique "blob" that can be difficult to analyze.

1.3.1.2 Scanning for Humans

There is an extensive and recent literature on this second type of human detection, including [Dalal and Triggs, 2005, Viola et al., 2005, Wu et al., 2005, Zhu et al., 2006, Dimitrijevic et al., 2006, Wu and Yu, 2006, Gavrila, 2007, Enzweiler and Gavrila, 2009, Breitenstein et al., 2011, Gall et al., 2011, Sabzmeydani and Mori, 2007]. These methods usually scan the image, classifying each extracted box as human or non-human based on a learnt knowledge of the human appearance. Sometimes, a coarse to fine search is considered to run the algorithm in real time as in [Gavrila, 2007]. These cited works mainly focus on pedestrian detection, i.e.

standing and walking people. Recently, Zhang et al proposed in [Zhang et al., 2007b] a novel model-based approach to detect humans performing different activities and showing complex postures.

All these detection techniques present some non-negligible advantages. For instance, they can work with isolated images as well as with sequences and do not require any previous knowledge about the scene. They can also work with moving cameras [Dimitrijevic et al., 2006] or with a camera mounted on a moving vehicle [Gavrila, 2007]. Even if recent work has focused on training these classifiers with weakly labelled data, their main drawback is the large data sets of manually labelled images that are needed to train them. Additionally, complex and long learning phases are usually required to select the relevant features that will allow to discriminate humans from cluttered background. Another drawback is that these techniques need to be very quick as they usually have to classify thousands of windows for each classified image. In [Gavrila, 2007], Gavrila presents a probabilistic approach to hierarchical, exemplar-based shape matching. This method achieves a very good detection rate and real time performance.

1.3.2 Pose Estimation

There are hundreds of papers dedicated to human pose estimation and motion capture, but there are basically two main strategies: methods which employ a datase of training exemplars, i.e. training images and the corresponding 2D/3D poses, and approaches that use a hand-made kinematic model of the human body.

1.3.2.1 Kinematic Models

Many efficient systems are based on the use of a model which is, most of the time, a representation of the human body. The selection of the appropriate model is a critical issue and the use of an explicit body model is not simple, given the numerous degrees of freedom (DOF) of the human body.

Kinematic Models attempt to describe the human body structure as a kinematic chain of segments (the limbs) connected by joints, each joint being parameterized by a series of degrees of freedom (translation or rotation). These models can be represented in 2D or 3D. To match the pose with the image, the individual limbs have been modelled as layered patches in the 2D image plane [Yacoob and Black, 1999, Yacoob and Davis, 2000, Agarwal and Triggs, 2004], or in the 3D world as stick figure [Taylor, 2000], cylinders [Sidenbladh et al., 2000, Bregler et al., 2004, Sigal et al., 2004], truncated cones [Deutscher et al., 2000, Deutscher and Reid, 2005], superquadrics [Gavrila and Davis, 1996, Sminchisescu and Triggs, 2003] or individually deformable shape models [Kakadiaris and Metaxas, 2000, Plankers and Fua, 2001].

In all these works, the authors make use of generative models for pose tracking, modeling the human kinematics more than the human appearance. Those methods do not explain the image evidence and do not always fit the image accurately which sometimes causes artifacts on joints estimation. Nowadays, exemplar based approaches are usually considered as they better represent the appearance of the human and closer match to image observation than kinematic models alone.

1.3.2.2 Exemplar based approaches

Exemplar-based approaches have been very successful for human pose estimation and tracking. Some consist of comparing the observed image with a data base of stored samples as in [Shakhnarovich et al., 2003, Mori and Malik, 2006, Ong et al., 2006]. In some other cases, the training examples are used to learn a mapping between image feature space and 3D pose

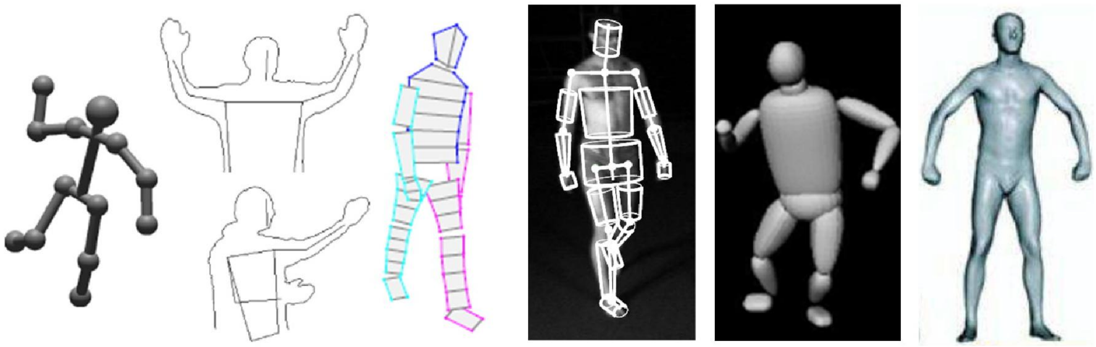


Figure 1.4: Examples of human pose and appearance modeling: *(from left to right)* 3D stick figure from [Taylor, 2000], 2D shape and 3D skeletal structure PDM from [Bowden et al., 2000], articulated 2D shape from [Zhang et al., 2005a], 3D truncated cones from [Deutscher et al., 2000], superquadrics from [Sminchisescu and Triggs, 2003] and 3D mesh model from [Balan et al., 2007].

space [Agarwal and Triggs, 2006, Elgammal and Lee, 2009, Lee and Elgammal, 2010, Jaeggli et al., 2009]. Such mappings can be used in a *bottom-up* discriminative way [Sminchisescu et al., 2007] to directly infer a pose from an appearance descriptor or in a *top-down* generative manner [Jaeggli et al., 2009] through a framework (e.g. a particle filter) where pose hypotheses are made and their appearances aligned with the image to evaluate the corresponding observation likelihood or cost function. The exemplars can also be used to train multi-class pose classifiers [Okada and Soatto, 2008, Andriluka et al., 2010] or part-based detectors [Ramanan et al., 2007, Wu and Nevatia, 2009, Bourdev et al., 2010, Felzenszwalb et al., 2010b, Lin and Davis, 2010] that are later employed to scan images.

Nearest Neighbor Search techniques have been very successful for pose recognition. This method consists in comparing the observed image with a data base of samples as in [Athitsos and Sclaroff, 2003, Orrite-Uruñuela et al., 2004, Mori and Malik, 2006, Ong et al., 2006] and find the most similar exemplar in the training set. However, in a scenario involving a wide range of viewpoints and poses, a large number of exemplars would be required. As a result, the computational time would be very high to recognize individual poses. One approach, based on efficient nearest neighbors search using histogram of gradient features, addressed the problem of quick retrieval in large set of exemplars by using Parameters Sensitive Hashing (PSH) [Shakhnarovich et al., 2003], a variant of the original Locality Sensitive Hashing algorithm (LSH) [Datar et al., 2004]. The final pose estimate is produced by applying locally-weighted regression to the neighbors found by PSH.

In [Grauman et al., 2004], the authors show how Earth Movers Distance (EMD) and Locality-Sensitive Hashing (LSH) can be used for quick contour-based shape retrievals. In [Toyama and Blake, 2002], an exemplar-based approach with dynamics is proposed for tracking pedestrians. In [Dimitrijevic et al., 2006], the authors present a template-based pose detector and solve the problem of large data set by detecting only human silhouette in a characteristic postures (sideways opened-legs walking postures in this case). Gavrila [2007] presents a probabilistic approach to hierarchical shape matching. These last four works basically look for the training human shape that best matches the input image but they do not infer any pose representations. Similar to Gavrila [2007] in spirit, Stenger [2004] uses a hierarchical Bayesian filter for real-time articulated hand tracking. Even if some techniques have been found for quick retrieval, the main drawback of all these exemplars based methods remains the large amount of memory needed to store the large data set.

Learning Based Approaches. Instead of storing and performing a nearest neighbor search for exemplars, Agarwal and Triggs [2006] use an efficient non-linear kernel-based regression of joint angles against shape descriptor vectors to distill a large training database into a compact model using a hopefully sparse subset of the exemplars learnt by the Relevance Vector Machines (RVM). Their method has the main disadvantage that it is silhouette based and that it can not model ambiguity in pose as the regression is uni-modal. Other earlier learning based approaches consider a mapping from 2D image space to 3D pose space. For example, Brand [1999] used a hidden Markov model (HMM) to represent this mapping between 2D silhouettes and 3D poses while Rosales et al. [Rosales et al., 2001] learn a similar mapping using neural network. Elgammal and Lee [Elgammal and Lee, 2004] learn view-based activity manifolds and subsequently, learn some mapping functions from these manifolds to both image inputs and 3D poses. The manifolds then allow to infer the 3D body pose from silhouette for specific actions and viewpoints. All these approaches have shown some very interesting results. However, most of them rely on the presence of clean segmented silhouettes.

Other techniques learn a mapping from a different feature space to 3D pose space. For instance, in [Sminchisescu et al., 2005], the authors model this complex multi-valued mapping between image observations and 3D poses with a mixture of experts models capable of representing multimodal conditionals. All these learning based algorithms have shown some very interesting results. However, most of them are trained to recognize specific poses for specific camera viewpoints.

Part-based Detectors The individual body parts can be detected in a *bottom-up* way, and then probabilistically assembled to estimate the 2D pose as in Ramanan's works [Ramanan and Forsyth, 2003, Ramanan, 2006, Ramanan et al., 2007]. The poselet work of [Bourdev and Malik, 2009] presents a two-layer classification/regression model for detecting people and localizing body components. The first layer consists of poselet classifiers trained to detect local patterns in the image. The second layer combines the output of the classifiers in a max-margin framework. Ferrari et al. [Ferrari et al., 2008] use an upper-body detector to localize a human in an image, find a rough segmentation using a foreground and background model calculated using the detection window location, and then apply a pictorial structure model [Felzenszwalb and Huttenlocher, 2005] in regions of interest. Such part-based methods are another successful approach to simultaneous human detection and pose estimation [Roberts et al., 2004, Navaratnam et al., 2005, Andriluka et al., 2009]. Typically however they require multiple classifiers or appearance models to represent each of the body parts.

1.3.3 Tracking

In the field of human motion analysis, the notion of *tracking* can refer to two different processes, depending on the space where the tracking is applied. In both cases, the *tracking* aims at establishing temporal correspondences between consecutive frames.

The first process is the tracking in the "image space" where the whole body's location, shape, size or the different body parts - in the image - are some of the possible parameters that can be tracked over time. Note that if the homography that relates image coordinates and coordinates on the ground plane is calculated, a tracking in the real world can be applied, providing trajectories on the ground plane.

The second type of tracking is the process of inferring a pose at one time instant given state information from previous time steps. This tracking is applied in the pose space and the resulting temporal trajectories then define the different actions - series of consecutive postures - such as walking, running, jumping, etc.

Relatively few papers tackle both problems together in real and complex environments such as monocular video surveillance footages or crowded street scene [Jaeggli et al., 2009, Okada and Soatto, 2008, Andriluka et al., 2010]. The two types of tracking face different problems:

the first one must deal with occlusions, image clutter and multiple interacting subjects while the second one has to tackle with the dimensionality and non-linearity of the state space. In the last years, multiple hypotheses tracking have been introduced to mitigate accumulation of error and ambiguities with both types of tracking, while learnt models of human motion have been proposed to constrain the search in high-dimensional pose spaces.

1.3.3.1 Multiple Hypotheses Tracking

Single hypothesis tracking algorithms, such as the early Kalman filter [Kalman, 1960], have been widely used for linear tracking. Unfortunately, they are restricted to unimodal probability distributions of the state parameter. Consequently, they badly perform in complex and ambiguous cases, such as tracking with the presence of occlusions and cluttered background, or pose tracking with severe self-occlusions ambiguities and non-linear motions. Maintaining multiple hypotheses over time has appeared as an effective solution to overcome those problems. In sampling-based approaches (particle filtering or Condensation [Isard and Blake, 1998]), a number of particles is propagated in time using a model of dynamics. Each particle is assigned a weight, known as importance weight, that is updated according to a cost function.

A well known disadvantage of particle filtering methods for pose tracking is that they are typically very slow. This is mainly due to the fact that the number of required particles grows up exponentially as the number of dimensions of parameters spaces does. Unfortunately, fewer samples decrease the performance of the filters. To overcome this problem, different modified particle filters have been developed for search in high dimensional configuration spaces. For example, the *annealed particle filter* from Deutscher et al [Deutscher et al., 2000, Deutscher and Reid, 2005] and the Covariance Scale Sampling from Sminchisescu and Triggs [Sminchisescu and Triggs, 2003] are two methods that have been introduced to guide the particles. MacCormick and Isard proposed in [MacCormick and Isard, 2000] to partition the pose space into lower dimensional subspaces.

1.3.3.2 Human Motion Models

As mentioned before, pose tracking is computationally difficult because of the high dimensionality of the involved space. Researches have investigated the use of learnt models of human motion to constraint the search in state space by providing strong priors on motion [Sidenbladh et al., 2002, Sigal et al., 2004, Ning et al., 2004b, Urtasun et al., 2005, 2006a].

In [Sidenbladh et al., 2002], Sidenbladh et al. combine stochastic sampling with a strong learned prior of walking. An exemplar-based approach is used where a database of motion capture examples is indexed to obtain possible movement directions. They retrieve motion samples and use their dynamics to propagate the particles. In [Ning et al., 2004b], the authors propose a similar method but consider physical motion constraints to restrict the propagation of the particles. Urtasun et al. [2006a] explore an approach to 3D people tracking with learned motion models and deterministic optimization.

Recent approaches have focused on the problem of dimensionality reduction for pose tracking and propose to use low dimensional embedding of human motion data like Gaussian process latent variable model (GPLVM) [Urtasun et al., 2006b, Ek et al., 2008, Andriluka et al., 2010], Locally Linear Embedding (LLE) [Jaeggli et al., 2009] or supervised manifold learning [Elgammal and Lee, 2009, Lee and Elgammal, 2010] .

1.4 Goals and Hypotheses

The goal of this thesis is to analyze the human pose in realistic scenarios with as less constraints as possible, and find some solutions to the following difficulties:

- the *modeling* of the human pose and appearance taking into account non-linearities, dimensionality and variability issues.

- the *detection* of the individuals present in the scene which needs to be accurate and fast.

- the *tracking* in pose and image spaces which imply to face the problem of the dimensionality of the searched space.

As shown in the previous section, exemplar based techniques have been very successful for human body pose analysis. However, their accuracy strongly depends on the similarity of both camera viewing angle and scene properties between training and testing images. Ideally, for an optimal result, the testing images should be taken in an environment (scene and camera) similar to the one that has been considered for the capture of the training exemplars. This is not always possible and we believe the trend is to consider as few training data as possible and develop algorithms that can be applied in any possible environment.

In this thesis, we focus on specific motion sequences (walking) due to the higher availability of training and evaluation datasets, walking being the most observed action in real sequences. We consider a limited training set captured from a small number of fixed cameras parallel to the ground and distributed around the subject. See examples in Fig. 1.6. Then, three types of testing environments with increasing level of difficulty have been identified and studied: 1) a static camera with a similar viewing angle observing only one individual, 2) a fixed surveillance camera with a considerably different viewing angle and multiple targets and 3) a moving camera sequence or just a single static image of an unknown scene. See examples in Fig. 1.5.

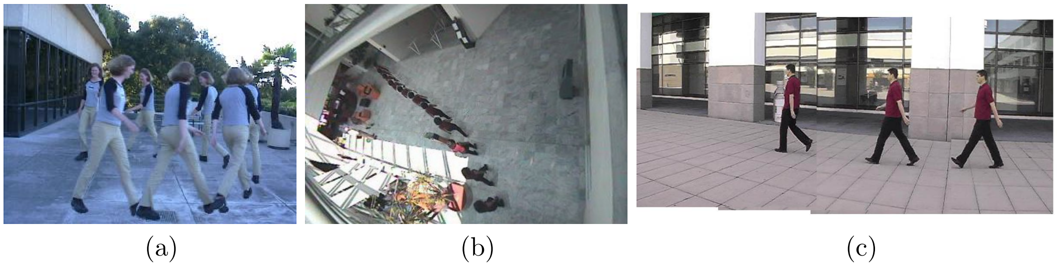


Figure 1.5: Testing environments with increasing level of difficulty studied in this thesis: (a) a static camera parallel to the ground observing only one individual, (b) a fixed surveillance camera with a considerable elevation angle and (c) a moving camera sequence.

Although all the experiments described in this thesis are specific to the walking activity, the frameworks and proposed algorithms are general enough to extend to other activities.

1.5 Contributions and Organization

Each testing environment raises different problems that we have considered separately. Therefore, we have structured the thesis in three main parts corresponding to these three testing conditions.

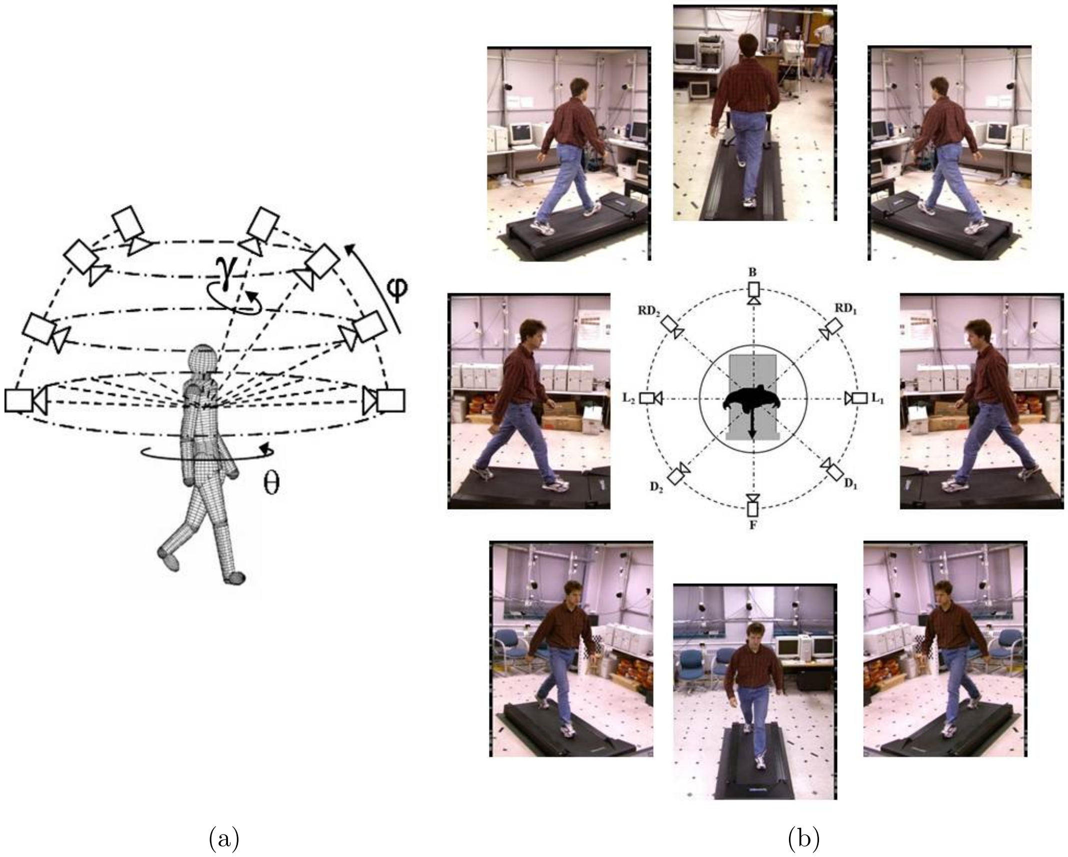


Figure 1.6: Viewpoint parameterization and training views: (a) Viewing hemisphere: the position of the camera with respect to the observed subject (the view) can be parameterized as the combination of two angles: the *elevation* $\varphi \in \left[0, \frac{\pi}{2}\right]$ (also called latitude or tilt angle) and the *azimuth* $\theta \in [-\pi, \pi]$ (also called longitude). A third angle $\gamma \in [-\pi, \pi]$ can be considered to parameterize the rotation around the viewing axis. (b) Viewpoint discretization for training: in this work, we use the MoBo dataset [Gross and Shi, 2001] and discretize the viewing hemisphere into 8 locations where θ is uniformly distributed around the subject. Only Front (F), Rear-Diagonal 2 (RD2), Lateral 1 (L1), Diagonal 1 (D1) and Back views belong to the original dataset. The other 3 views, L2, D2 and RD1 are obtained by mirroring L1, D1 and RD2 about the vertical axis.

1.5.1 Part I: Segmentation and Pose Estimation with a Static Camera

In the first part, we consider a captured with a camera parallel to the ground. We take advantage of the fact that the camera is fixed and use a common static background subtraction algorithm to perform foreground detection. We then propose a model-based approach associating the 2D body pose and the 2D silhouette shape.

1.5.1.1 chapter 2: Dealing with Non-linearities in Shape Modeling

In chapter 2, we address the problem of non-linearity in 2D shape modeling of articulated objects like the human body. This issue is partially resolved by applying a different Point Distribution Model (PDM) depending on the viewpoint. The remaining non-linearity is solved by using Gaussian Mixture Models (GMM). A dynamic-based clustering is proposed and carried out in the Pose Eigenspace. A fundamental question when clustering is to determine the optimal number of clusters. From our point of view, the main aspect to be evaluated is the mean Gaussianity. This partitioning is then used to fit a GMM to each one of the view-based PDM, derived from a database of silhouettes and 2D poses. Dynamic correspondences are then obtained between Gaussian models of the mixtures. Finally, we compare this approach with other two methods we previously developed to cope with non-linearity: Nearest Neighbor (NN) Classifier and Independent Component Analysis (ICA).

1.5.1.2 chapter 3: A Framework of Spatio-temporal Models

In chapter 3, we propose to use a model-based approach, where the 2D shape is associated to the corresponding 2D stick figure, thus allowing the joint segmentation and pose recovery of the subject observed in the scene. The main disadvantage of 2D-models is their restriction to the viewpoint. To cope with this limitation, we propose to train local spatio-temporal 2D-models corresponding to several views of the same sequences. A multi-view Gaussian Mixture Model (GMM) is then fitted to a feature space made of shapes and stick figures manually labelled. The resulting spatio-temporal models is concatenated and sorted in a global framework. Temporal and spatial constraints are considered to build a probabilistic transition matrix that gives a frame to frame estimation of the most probable local models to use during the fitting procedure, thus limiting the search space. This approach takes advantage of 3D information avoiding the use of a 3D human model.

1.5.2 Part II: Pose Tracking in Video-Surveillance Environments

In the second part, we present a methodology for view-invariant monocular body pose tracking in man-made environments with a calibrated camera. The framework proposed in the first part performs well when the camera is parallel to the ground. However, in presence of perspective effect, the distortion will cause the parts of the subject that are closer to the lens to appear abnormally large, thus deforming the shape of the human contour in ways that can prevent a correct analysis. We propose to exploit projective geometry to find viewpoint invariance.

1.5.2.1 chapter 4: View-invariant Motion Analysis using View-based Models

In chapter 4, we consider the problem of view dependency of 2D-models and present a solution for man-made environments. During the online stage, the Homography that relates the image points to the closest training plane is calculated using the dominant 3D directions. The input image is then be warped to this training view and processed using the corresponding view-based model in our framework. After model fitting, the inverse transformation can be performed on the resulting human features obtaining a segmented silhouette and a 2D pose in the original input image.

1.5.2.2 chapter 5: View-invariant 3D Pose Tracking

In chapter 5, we model 3D body poses and camera viewpoints with a low dimensional manifold and learn a generative model of the silhouette from this manifold to the set of training views. During the online stage, 3D body poses are tracked using recursive Bayesian sampling conducted jointly over the scene's ground plane and the pose-viewpoint manifold. For each sample, the homography that relates the corresponding training plane to the image points is calculated using the dominant 3D directions of the scene, the sampled location on the ground plane and the sampled camera view. Each regressed silhouette shape is projected using this homographic transformation and matched in the image to estimate its likelihood.

1.5.3 Part III: Pose Estimation with a Moving Camera or in Static Images

The techniques employed in the previous part therefore separate the foreground from the background and estimate the pose of the human fitting a human model on the resulting blob or silhouette. Such methods are very helpful when a relatively clean background image can be computed which is not always the case, depending on the settings and applications: for example if the goal is to detect humans in an isolated static image (not from a video sequence) or in a moving camera sequence, the computation of a background image and consequently the segmentation of the subject are not trivial. In the third part, we learn a fast pose classifier which is used in a sliding window framework to quickly estimate the presence and pose of the humans.

1.5.3.1 chapter 6: Multi-class Human Pose Classifier

The contribution in chapter 6 is a multi-class pose classifier that uses the best components of state-of-the-art classifiers including hierarchical trees, cascades of rejectors as well as randomized forests. Given a database of images with corresponding human poses, we define a set of classes by discretizing camera viewpoint and pose space. A bottom-up approach is first followed to build a class hierarchy by recursively clustering and merging the classes at each level. For each branch of this hierarchical decision tree, we take advantage of the alignment of training images to build a list of potentially discriminative HOG (Histograms of Orientated Gradients) features. We then select the HOG blocks that show the best rejection performances. We finally grow an ensemble of cascades by randomly sampling one of these HOG-based rejectors at each branch of the tree.

1.5.3.2 chapter 7: Human Localization and Pose Estimation

This chapter addresses human detection and pose estimation from monocular images by formulating it as a classification problem. The multi-class classifier presented in the previous chapter is thus used to scan images in a sliding window scheme.

Each hierarchical cascade in the ensemble can make a decision and efficiently reject negative candidates by only sampling a few features of the available feature space. This makes our classifier more suitable for sliding window detectors than state-of-the-art classifiers.

1.5.4 Part IV

1.5.4.1 chapter 8

The last chapter of the thesis summarized the conclusions of the work and some possible future lines.

1.6 List of Relevant Publications

The research described in this thesis is based on material from the following publications:

1.6.1 Part I

- Rogez, G., Orrite, C., Martínez, J., and Jaraba, J. E. H. (2006b). Probabilistic spatio-temporal 2d-model for pedestrian motion analysis in monocular sequences. In *Proc. of* *the 4th International Conference on Articulated Motion and Deformable Objects (AMDO)*, pages 175–184

- Rogez, G., Martínez, J., and Orrite, C. (2007b). Dealing with non-linearity in shape modelling of articulated objects. In *Proc. of the Third Iberian Conference on Pattern Recognition and Image Analysis (IbPria)*, pages 63–71
- Rogez, G., Orrite, C., and Martínez, J. (2008a). A spatio-temporal 2d-models framework for human pose recovery in monocular sequences. *Pattern Recognition*, 41(9):2926–2944

1.6.2 Part II

- Rogez, G., Guerrero, J., Martínez, J., and Orrite, C. (2006a). Viewpoint independent human motion analysis in man-made environments. In *Proc. of the 17th British Machine Vision Conference (BMVC)*, volume 2, pages 659–668, Edinburgh, UK
- Rogez, G., Guerrero, J. J., and Orrite, C. (2007a). View-invariant human feature extraction for video-surveillance applications. In *Proc. of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 324–329
- Rogez, G., Orrite, C., Rihan, J., Guerrero, J. J., and Torr, P. H. (under review). View-invariant shape-based 3d human pose tracking in monocular surveillance videos. *submitted to the International Journal of Computer Vision*

1.6.3 Part III

- Rogez, G., Rihan, J., Ramalingam, S., Orrite, C., and Torr, P. H. (2008b). Randomized trees for human pose detection. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*
- Rogez, G., Rihan, J., Orrite, C., and Torr, P. H. (2012). Fast human pose detection using randomized hierarchical cascades of rejectors. *International Journal of Computer Vision*, 99(1):25–52

1.6.4 Miscellaneous:

Several follow-on papers are still under preparation but some of the ideas developed in this thesis have already been used in the following publications:

- Rogez, G., Rius, I., Martínez, J., and Orrite, C. (2007c). Exploiting spatio-temporal constraints for robust 2d pose tracking. In *Proc. of the Second Workshop of Human Motion - Understanding, Modeling, Capture and Animation*, pages 58–73
- Martínez, J., Orrite-Uruñuela, C., and Rogez, G. (2007). Rao-blackwellized particle filter for human appearance and position tracking. In *Proc. of the Third Iberian Conference on Pattern Recognition and Image Analysis (IbPria)*, pages 201–208
- Ek, C. H., Rihan, J., Torr, P. H. S., Rogez, G., and Lawrence, N. D. (2008). Ambiguity modeling in latent spaces. In *MLMI*, pages 62–73
- Orrite, C., Gañán, A., and Rogez, G. (2009). Hog-based decision tree for facial expression classification. In *IbPRIA*, pages 176–183

Part I

Segmentation and Pose Estimation with a Static Camera

2

Dealing with Non-linearities in Shape Modeling

2.1 Introduction

In the first part of this thesis, we consider that the camera is fixed and that an estimate of the foreground silhouette can be obtained using a common static background subtraction algorithm. People are able to deduce the pose of a known articulated object (e.g. a person) from a simple binary silhouette. The possible ambiguities can be solved from dynamics when the object is moving. Following this statement, we propose in this first chapter to construct a human model that encapsulates within a Point Distribution Model (PDM) [Cootes and Taylor, 1997] both body silhouette information provided by the 2D shape and structural information given by the 2D skeleton joints. In that way, the 2D pose could be inferred from the silhouette and vice versa. Due to the high non-linearity of the resulting feature space, mainly caused by the rotational deformations inherent to the articulated structure of the human body, the use of non-linear statistical models will be considered in this chapter. This approach will be compared to other two methods previously tested for solving non-linearity issue. Such non-linear statistical models have been previously proposed by Bowden [Bowden et al., 2000] that demonstrated how the 3D structure of an object can be reconstructed from a single view of its outline. While Bowden only considered the upper human body and the frontal view, in this work the complete body will be modelled and different viewpoint will be taken into account.

2.1.1 Related Work

Even though many new types of image features have recently been developed, silhouette-based approaches are still receiving much attention. These approaches focus on the use of the binary silhouette of the human body as a feature for detection [Gavrila, 2007, Lin and Davis, 2010], tracking [Baumberg and Hogg, 1994, Siebel and Maybank, 2002b, Giebel et al., 2004, Toyama and Blake, 2002, Cremers, 2006], pose estimation [Jaeggli et al., 2009, Elgammal and Lee, 2009, Lee and Elgammal, 2010, Li et al., 2010] or action recognition [Weinland et al., 2007, Abdelkader et al., 2011] to cite a few. They rely on the observation that most human gestures can be recognized using only the outline shape of the body silhouette. The most important advantage of these features is their ease of extraction from raw video frames using low-level processing tasks like background subtraction or edge detection algorithms.

2.1.1.1 Human shape models

Human shape models have appeared to be powerful tools for human motion analysis. When a 2D representation is employed, the outline of the silhouette is usually parameterized by a series of 2D landmarks. Baumberg and Hogg [1994] used active shape models to track pedestrians from a fixed camera. The same active shape tracker was considered by Siebel and Maybank [Siebel and Maybank, 2002a] that extended it by a head detector and a region tracker, all integrated in the visual surveillance system ADVISOR. In [Fan et al., 2003], Fan et al. presented a compound structural and textural image model for pedestrian registration. In [Veeraraghavan et al., 2005], the authors even exploit the shape deformations of a person's silhouette as a discriminative feature for gait recognition, indicating that methods based on shape perform better than methods based on kinematics alone. Wang et al. [2003] also propose an efficient gait recognition algorithm using view-dependent silhouette analysis. They concluded that the lack of generality of viewing angle is a limitation to most gait recognition algorithms. Munder et al. [2008] proposed a Bayesian framework for tracking pedestrians from a moving vehicle: a method for learning spatio-temporal shape representations from examples was outlined, involving a set of distinct linear subspaces models. The main problem with 2D models is their dependency to the viewpoint. Indeed, the accuracy strongly depends on the similarity between training and testing viewpoints. Recent approaches proposed the use of detailed 3D mesh models learnt from 3D laser scans [Corazza et al., 2006, Rosenhahn et al., 2006].

All these approaches represent the shape of the human but they ignore the rigid deformations inherent to the kinematic of the human body. One exception can be found in [Zhang et al., 2005a] where the authors introduced a statistical shape representation of non-rigid and articulated body contours. To accommodate large viewpoint changes, the authors proposed to employ a mixture of a finite number of view-dependent models. Even so, all those works on shape models present the same common drawback that, even if they give an idea of the pose, no real regression to 2D/3D joints is considered.

2.1.1.2 Shape+Structure models.

Some few works have attempted to model the shape together with the structure of the human body. It is not a simple task since both types of feature belong to different metric spaces - pixels vs angles or pixels vs 3D position.

Grauman et al. [Grauman et al., 2003] inferred 3D structure from multi-view contour using a probabilistic "shape+structure" model. This idea was first introduced by Bowden [Bowden et al., 2000] that demonstrated how the 3D structure of an object can be reconstructed from a single view of its outline using a model of movement and shape. In this work, 2D shape and 3D skeletal structure were encapsulated within a non-linear Point Distribution Model (PDM). Some very encouraging results have been shown by these two papers in controlled environment and with relatively "good" input silhouette. However, both suffer from the same disadvantage than 2D shape models: the need of manually segmented data.

More recently, Balan et al. [Balan et al., 2007] also proposed something in between the two different approaches - kinematic models vs shape model - considering a parametric model of 3D shape and pose-dependent deformations to represent the body via SCAPE mesh model. One must say that the results are really impressive. They represent both articulated and non-rigid deformations of the human body as well as body shape variability. But this method requires 3D body scan of naked people. Another drawback is the computational cost required to fit the thousands of triangles of the mesh model.

2.1.2 Overview

Thanks to the structural knowledge, people are able to deduce the pose of an articulated object (e.g. a person) from a simple binary silhouette. Following this statement, our idea is to construct a human model encapsulating within a point distribution model (PDM) body silhouette information given by the 2D shape (landmarks located along the contour) and the structural information given by the 2D skeleton joints. In that way, the 2D pose could be inferred from the silhouette. Due to the high non-linearity of the feature space, mainly caused by the rotational deformations inherent to the articulated structure of the human body, we consider in this work the necessity to use non-linear statistical models. They have been previously proposed by Bowden [Bowden et al., 2000] that demonstrated how the 3D structure of an object can be reconstructed from a single view of its outline.

In a previous work [Orrite-Uruñuela et al., 2004], the problem of non-linear principal component analysis was partially resolved by applying a different PDM depending on previous pose estimation (4 views were considered: frontal, lateral, diagonal and back views) and the same procedure will be followed in this work. Additionally, results were obtained from measurement by selecting the closest shape from the training set by means of a Nearest Neighbour (NN) classifier. However, to cope with the remaining non-linearity, we consider in this thesis the use of Gaussian Mixture Models (GMM) as in [Enzweiler and Gavrila, 2008]. A dynamic-based clustering is considered by partitioning the 3D pose space. The 4 view-based GMM are then built in their respective shape space using this same labelling. Dynamic correspondences are then obtained between gaussian models of the 4 mixtures. In [Rogez et al., 2005], we proposed to use Independent Component Analysis (ICA) to cope with the problem of non-linearity in human shape modeling. Results obtained with our GMM will be compared with results from ICA and NN modeling.

In Sect. 2.2, we introduce the training database construction. In Sect. 2.3, we detail the construction of our view-based GMM. Some results are presented in Sect. 2.4 and some conclusions are finally drawn in Sect. 2.5.

2.2 Shape-Skeleton Training Database

The goal is to construct a statistical model which represents a human body and the possible ways in which it can deform. Point distribution models (PDM) are used to associate silhouettes (shapes) and the corresponding skeletal structures.

2.2.1 Training Database Construction

The generation of the 2D deformable model follows a procedure similar to [Koschan et al., 2003]. The CMU MoBo database [Gross and Shi, 2001] is considered for the training stage: good training shapes are extracted manually trying to get accurate and detailed approximations of human contours. Simultaneously, 13 fundamental points corresponding to a stick model are extracted: head center, shoulders, elbows, wrists, hips, knees and ankles. The skeleton vectors are then defined as:

$$\mathbf{k}_\mathrm{i} = [\mathrm{x}_{\mathrm{k}1}, ..., \mathrm{x}_{\mathrm{k}13}, \mathrm{y}_{\mathrm{k}1}, ..., \mathrm{y}_{\mathrm{k}13}]^\top \in \mathbb{R}^{26}, \tag{2.1}$$

with $i = 1...N_v$, N_v being the number of training vectors. Two gait cycles (low and high speed) and 4 viewpoints (frontal, lateral, diagonal and back views) are considered for each one of the 15 selected subjects. This manual process leads to the generation of a very precise database but without shape-to-shape landmarks correspondences.

2.2.2 Shapes Normalization

The good results obtained by a PDM depend critically on the way the data set has been normalized and on the correspondences that have been established between its members [Davies et al., 2003]. Human silhouette is a very difficult case since people can take a large number of different poses that affect the contour appearance. A big difficulty relies on establishing correspondences between landmarks and normalizing all the possible human shapes with the same number of points. In this chapter, we propose to use a large number of points for defining all the contours and "superpose" the points that are not useful (see Fig. 2.1).

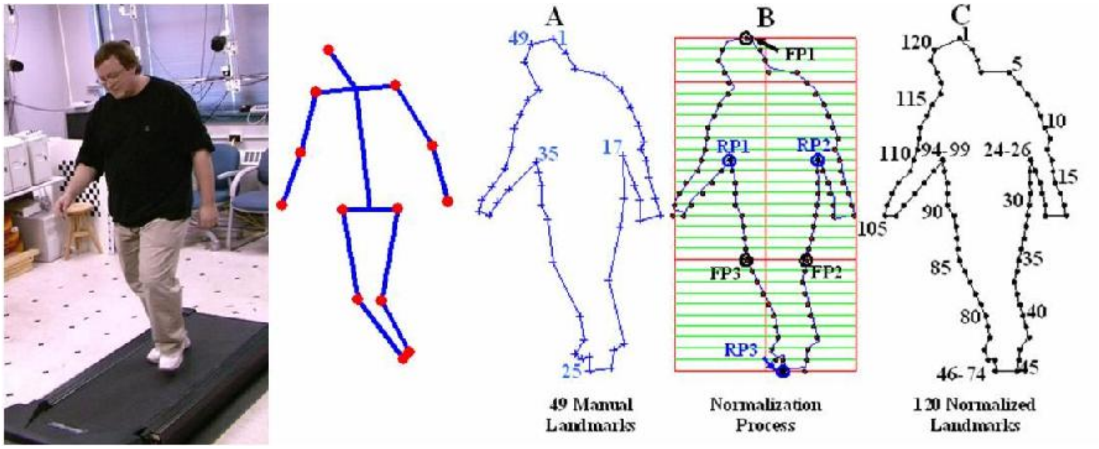


Figure 2.1: Training database. From *left* to *right*: MoBo Image, 2D skeleton and shape normalization: hand-labelled landmarks (A), rectangular grid (B), 120 normalized landmarks (C), part of them grouped at "repository points": 24-26 at RP2, 46-74 at RP3 and 94-99 at RP1.

A rectangular grid with horizontal lines equally spaced is applied to the contours database. This idea appears as a solution to the global verticality of the shapes and the global horizontality of the motion. The intersections between contours and grid are then considered. The shapes are then divided into 3 different zones delimited by three fixed points: the higher point of the head (FP1) and the intersections with a line located at 1/3 of the height (FP2 and FP3). A number of landmarks is thus assigned to each segment and a repository point (RP) is selected to concentrate all the points that have not been used. In this chapter, all the training shapes are made of 120 normalized landmarks:

$$\mathbf{s}_{\mathrm{i}} = [\mathrm{x}_{\mathrm{s}1}, ..., \mathrm{x}_{\mathrm{s}120}, \mathrm{y}_{\mathrm{s}1}, ..., \mathrm{y}_{\mathrm{s}120}]^{\top} \in \mathbb{R}^{240}, \tag{2.2}$$

with $i = 1...N_v$.

2.2.3 Shape-Skeleton Eigenspace - PCA Model

Shapes and Skeletons are now concatenated into Shape-Skeleton vectors:

$$\mathbf{v}_{\mathrm{i}} = [\mathbf{s}_{\mathrm{i}}^{\top} \quad \mathbf{k}_{\mathrm{i}}^{\top}]^{\top} \in \mathbb{R}^{266}, \tag{2.3}$$

with $i = 1...N_v$. This training set is aligned using Procrustes analysis (each view being aligned independently) and Principal Component Analysis (PCA) is applied [Cootes and Taylor, 1997] for dimensionality reduction on the 4 view-based training sets. In that way, 4 view-dependent Shape-Skeleton models are constructed by extracting the mean vector and the variation modes:

$$\mathbf{v}_\mathrm{i} \simeq \bar{\mathbf{v}}_\theta + \mathbf{\Phi}_\theta \mathbf{b}_\mathrm{i} \tag{2.4}$$

where $\bar{\mathbf{v}}_\theta$ and $\mathbf{\Phi}_\theta$ are respectively the mean Shape-Skeleton vector and the matrix of Eigenvectors for the training viewpoint θ. $\mathbf{b}$ is the projection of $\mathbf{v}_i$ in the corresponding Eigenspace i.e. a vector of weights $\mathbf{b}_i = [b_1, b_2...b_n]^\top$. The main problem with this approach is that PCA assumes a Gaussian distribution of the input data. This supposition fails because of the inherent non-linearity of the feature space and leads to a wrong description of the data: the resulting model can consider as valid some implausible Shape-Skeleton combinations. Therefore, other approaches have to be taken into account to generate the "Shape-Skeleton" model and adequately represent the training set.

2.3 View-based Shape-Skeleton Gaussian Mixture Models

Many researchers have proposed approaches to non-linear PDM [Cootes and Taylor, 1997, Bowden et al., 2000]. The use of Gaussian Mixture Model (GMM) was first proposed by Cootes and Taylor [Cootes and Taylor, 1997]. They suggested modeling non-linear data sets using a GMM fitted to the data using the Expectation Maximization (EM) algorithm. This provides a more reliable model since the feature space is limited by the bounds of each Gaussian that appear to be more precise local constraints.

$$\mathrm{p}_\mathrm{mix}(\mathbf{b}) = \sum_{\mathrm{j}=1}^{\mathrm{m}} \omega_\mathrm{j}\, \mathcal{N}(\mathbf{b} : \bar{\mathbf{b}}_\mathrm{j}, \mathbf{S}_\mathrm{j}), \tag{2.5}$$

where $\mathcal{N}(\mathbf{b} : \bar{\mathbf{b}}, \mathbf{S})$ is the p.d.f. of a gaussian with mean $\bar{\mathbf{b}}$ and covariance $\mathbf{S}$.

Bowden [Bowden et al., 2000] proposed first to compute linear Principal Component Analysis (PCA) and to project all shapes on PCA basis. Then do cluster analysis on projections and select an optimum number of clusters. Each data point is assigned to a cluster and separate local PCA are performed independently on each cluster. This results in the identification of local model's modes of variation inside each Gaussian distribution of the mixture: $\mathbf{b} \simeq \bar{\mathbf{b}}_j + \mathbf{\Phi}_j \mathbf{r}$ (see Eq.2.4). Thus, a more complex model is built to represent the statistical variations. Given the promising results described in [Bowden et al., 2000], a similar procedure is followed in this work, the main difference relying on the way the feature space is clustered: the proposed methodology consists in partitioning the complete Shape-Skeleton feature space using only the dynamical information provided by the pose parameters. The contour parameters are not taken into account for clustering since they do not provide any additional information on dynamics and can lead to ambiguities as stated in [Agarwal and Triggs, 2006].

2.3.1 Structural clustering

While in [Munder et al., 2008], the clustering of the shape feature space was based on a similarity measure derived from the registration procedure, here it is proposed to use the structural information provided by the pose to cluster both shape and skeleton training sets, thus establishing dynamical correspondences between view-based data.

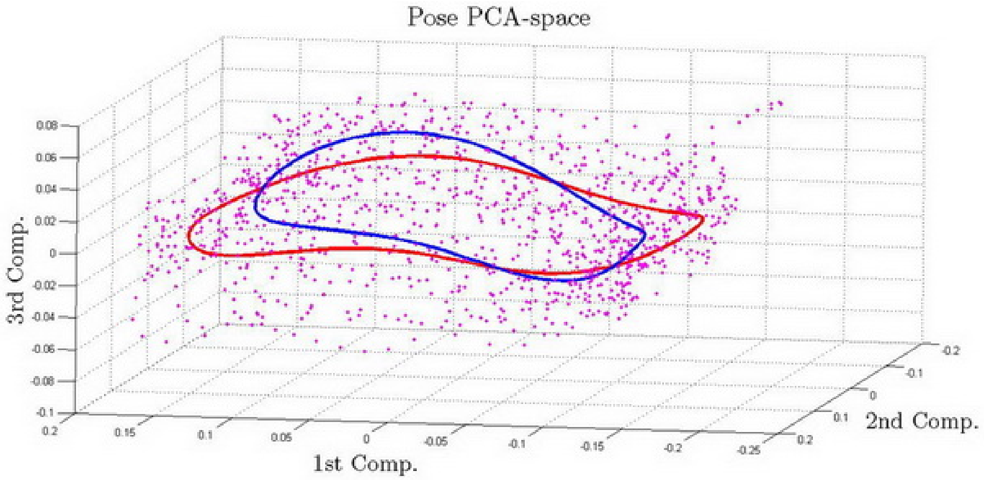


Figure 2.2: Low and high speed gait cycles represented on the 3 first modes of the Pose Eigenspace.

2.3.1.1 Pose Eigenspace for Clustering

The information provided by the 3D poses is used for clustering: for each snapshot of the training set, the 3D skeleton is built from the corresponding 2D poses $\mathbf{k}_i$ of the 4 views, by reconstructing the 3D position of the joints using the 4 2D-projections and Tsai's algorithm [Tsai, 1986]. The resulting set of 3D poses is then aligned using Procrustes and reduced by PCA obtaining the Pose Eigenspace (Fig. 2.2) where the dynamic-based clustering will be operated.

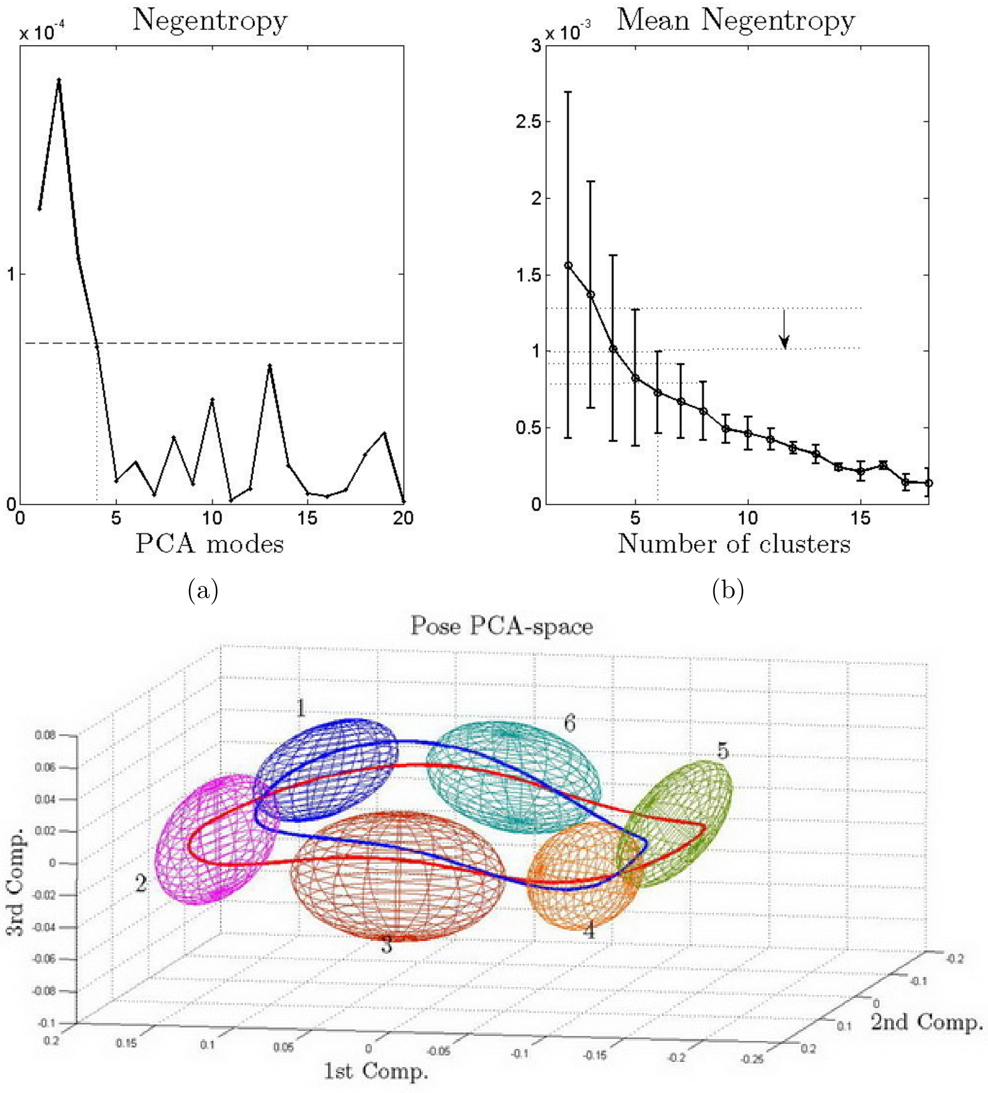
2.3.1.2 Principal components selection

The non-linearities of the training set are mainly localized in the first components of the PCA which capture the dynamics, as shown in Fig. 2.2. These components are really influential during the partitioning step while the last ones, more linear, only model local variations (details) and do not provide so much information for clustering. Only the first non-linear components are thus selected to perform the clustering of the data in a lower dimensional space. For components selection, the non-gaussianity of the data is measured on each component. There are different methodologies to test whether the assumed normal probability distribution accurately characterizes the observed data or not. Skewness and kurtosis, are two classical measures of non-gaussianity.

A more robust measure is given by the Negentropy, the classic information theory's measure of non-Gaussianity, whose value is zero for Gaussian distribution [Hyvaerinen et al., 2001]. Fig. 2.3a shows how the Negentropy converges to 0 and oscillates when considering lower modes. This oscillation between 0 and 0.75×10^{-4} starts from the 4th mode. It can be observed how the first 3 modes present a much higher Negentropy compared to the other modes. According to this analysis, we select the 3 first components for clustering.

2.3.1.3 Determining the number of clusters.

K-means algorithm is used fairly frequently as a result of its ease of implementation. K-means clustering splits a set of objects into a selected number of groups by maximizing between variations relative to within variations. The main disadvantage of this algorithm is its extreme sensitivity to the initial seeds. A solution could be found by applying k-means several times,



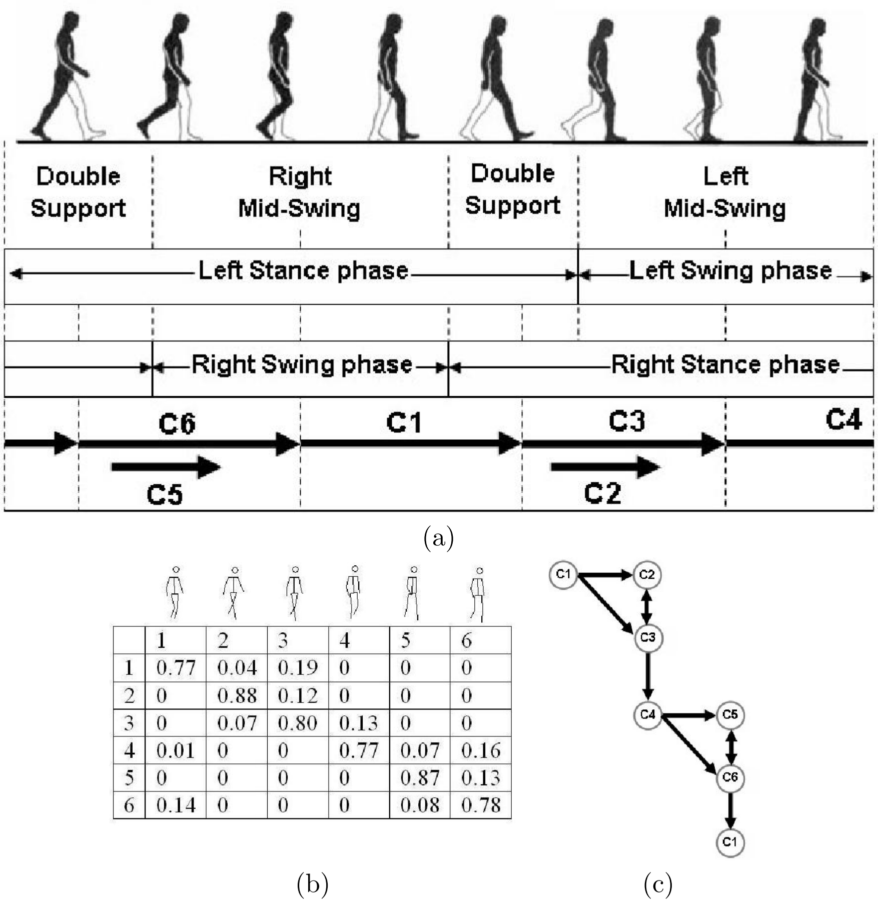
(a)

(b)

(c)

Figure 2.3: Structural clustering: (a) Negentropy of the 20 first modes of the Pose Eigenspace. (b) Negentropy of the GMM (mean & st.dev.) vs. number of clusters. (c) Resulting GMM for k=6, represented in the Pose Eigenspace together with the gait cycles.

starting with different initial conditions and then choosing the best solution. But this supposes a supervision that makes the process more ad-hoc. To make the clustering independent from the initial seeds, the K-means algorithm is ran many times and the total results are clustered as in [Rogez et al., 2005]. For each case ($K = 2 \cdots N$), a GMM is fitted to the Pose Eigenspace using the Expectation Maximization (EM) and a local PCA's is applied on each cluster. Since local modes of variation inside each Gaussian distribution of the mixture are expected, one of the aspects that should be evaluated when determining the optimal number of cluster is the global gaussianity of the GMM. All the points of the training set are then projected onto the corresponding local PCA space and the Negentropy is computed for each cluster.



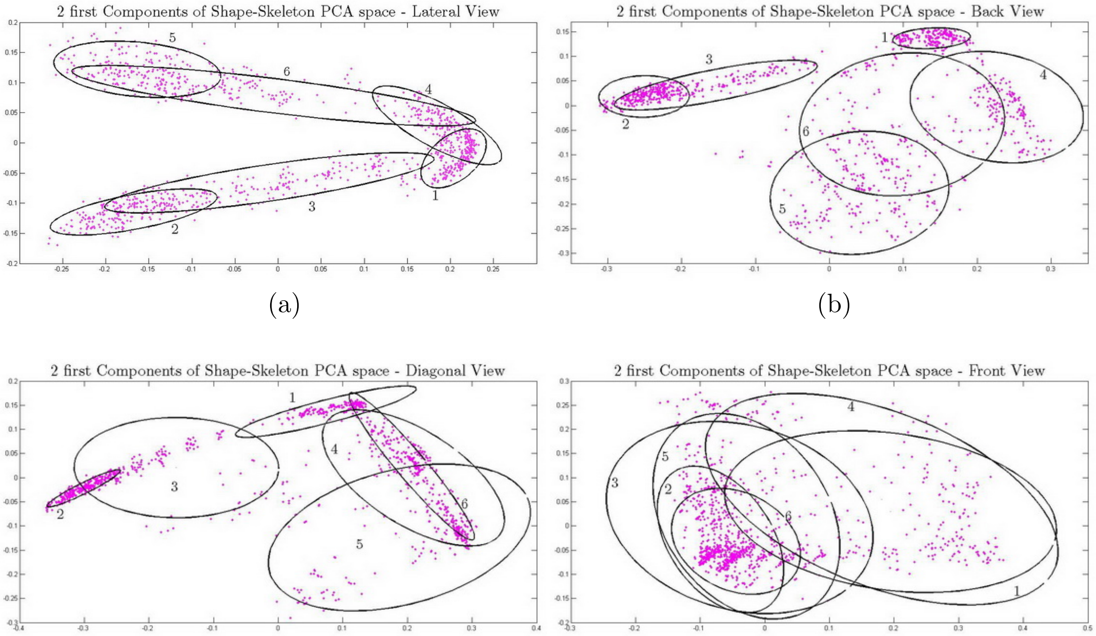
	1	2	3	4	5	6
1	0.77	0.04	0.19	0	0	0
2	0	0.88	0.12	0	0	0
3	0	0.07	0.80	0.13	0	0
4	0.01	0	0	0.77	0.07	0.16
5	0	0	0	0	0.87	0.13
6	0.14	0	0	0	0.08	0.78

(b) (c)

Figure 2.4: Gait cycle phases: (a) Correspondences between Gait cycle and the 6 clusters obtained. (b) Markov State Transition Matrix.(b) State Diagram.

In Fig. 2.3b, the evolution of the mean Negentropy can be observed for K varying from 2 to 18. The curve decreases and converges logically to 0. It is desired to create as few clusters as possible and obtain some clusters as gaussian as possible. A good compromise between number of clusters and gaussianity is reached at $K = 6$ where the standard deviation of the Negentropy substantially decreases compared to the one at $K = 5$. Fig. 2.3c shows the GMM obtained with $K = 6$, represented in the Pose Eigenspace. This graphical representation shows the accuracy of GMM only by simple visual criteria: comparing with Fig. 2.2, it can be observed how well the GMM limits the feature space.

This leads to the recognition of basic gait cycle phases [Inman et al., 1981], as illustrated by Fig. 2.4, in an unsupervised way. The patches are ordered according to the logic of the cyclic motion: C1 starts with the Right Mid-Swing and ends with the double support phase, then C3 starts until the Left Mid-Swing. C4 follows until the second double support of the cycle which ends with C6. C2 and C5 complete C3 and C6 phases in case of a higher speed gait with larger step. A Markov State Transition Matrix (STM) [Heap and Hogg, 1998] is then constructed (Fig. 2.4b), associating each sample to one of the 6 patches. Each temporal cluster corresponds to a state in the Markov chain. This gives the state transition probabilities, valid for the 4 sets (views) of SS-vectors.



(a) (b)

(c) (d)

Figure 2.5: View-based mixtures of PCA: GMM represented on the 2 first components of the Shape-Skeleton Eigenspace for the lateral (a), back (b), diagonal (c) and front (d) views.

2.3.2 View-based Non-Linear Models

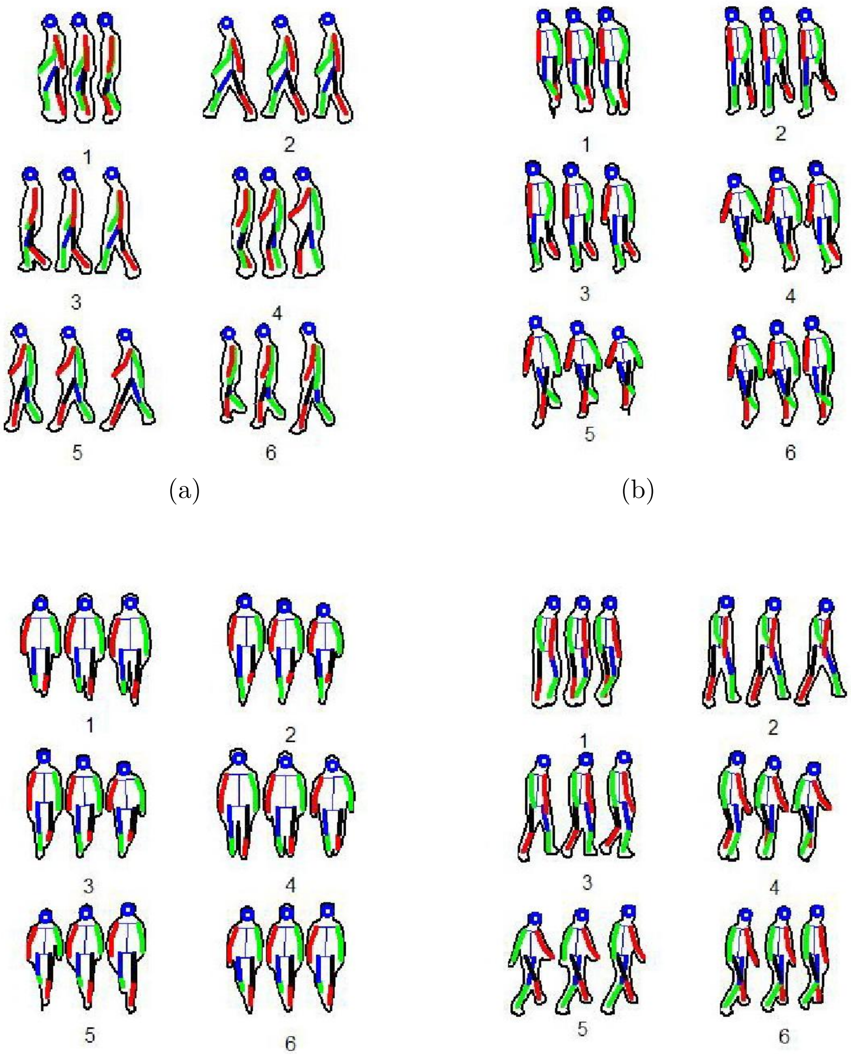
A view-based mixture of PCA is now fitted to the 4 Shape-Skeleton Eigenspaces, using the structure-based clustering obtained before. Figure 2.5 shows how the different mixtures limit the feature spaces: the clustering imposes a particular location of the gaussian distribution (represented as ellipsoids) that consequently treats some unseen data as valid by interpolating. Fig. 2.6 shows how both shape and skeleton deform linearly in each one of the clusters of the view-based GMM. Dynamic correspondences are obtained between gaussian models of the 4 mixtures, each cluster corresponding to one of 6 basic gait phases.

2.3.3 Joint Estimation of Shape and Skeleton

In [Grauman et al., 2003], Grauman inferred 3D structure from multi-view contour. Following the same idea, when presented a new shape, the unknown 2D structure (structural parameters) is treated as missing variables in a SS-vector. The corresponding $\mathbf{b}^*$ is then computed from Eq.2.4 and the nearest cluster, defined by Eigenvectors $\mathbf{\Phi} = [\Phi_1, ...\Phi_t..., \Phi_T]$ and Eigenvalues λ_t, is selected. Thus the closest allowable SS-vector from the model is constructed by finding $\mathbf{r}$ so that:

$$\mathbf{r} = \Phi^{-1}(\mathbf{b}^* - \bar{\mathbf{b}}) \quad \text{and} \quad -\beta\sqrt{\lambda_\mathrm{t}} \leq \mathrm{r_t} \leq \beta\sqrt{\lambda_\mathrm{t}}. \tag{2.6}$$

To ensure a valid SS-vector generation, the weight vector $\mathbf{r}$ is constrained to lie in the hyperellipsoid representing the linear subspace model [Koschan et al., 2003]. This leads to a model-based estimation of both shape and skeleton (cf Fig. 2.8).



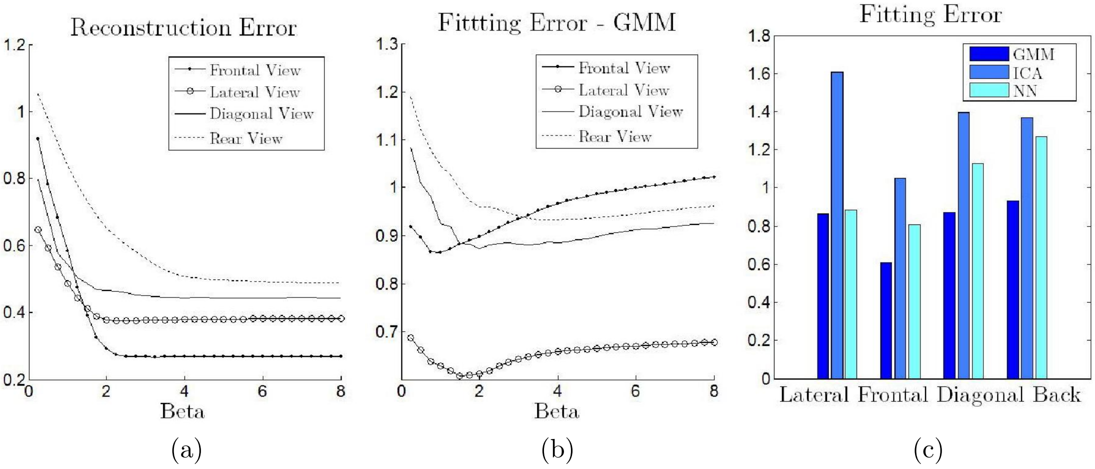
(a) (b) (c) (d)

Figure 2.6: Principal modes of variation of the 6 corresponding gaussian models for the 4 view-based GMMs: lateral (a), diagonal (b), frontal (c) and back (d) views.

2.4 Non-Linear Models Testing

The first approach we followed to cope with the non-linearity of the Eigenspace was to select the closest allowable shape from the training set by means of a Nearest Neighbor (NN) classifier [Orrite-Uruñuela et al., 2004]. This technique always warranties a valid contour but is imperfect because it can not generate new shapes absent from the training data. Moreover, the computational cost makes this approach infeasible with a very large database. In a previous work [Rogez et al., 2005], we proposed to use Independent Component Analysis (ICA) for human shape modeling. The dynamic-based GMM developed in this chapter will be compared to both methods.

For the evaluation of the view-based models, 4 gait sequences whose viewpoints correspond



(a) (b) (c)

Figure 2.7: Non-Linear Models Testing: Reconstruction (a) & Fitting (b) Error obtained applying our GMM on the 4 Caviar sequences. (c) Comparative results for the NN, ICA and GMM.

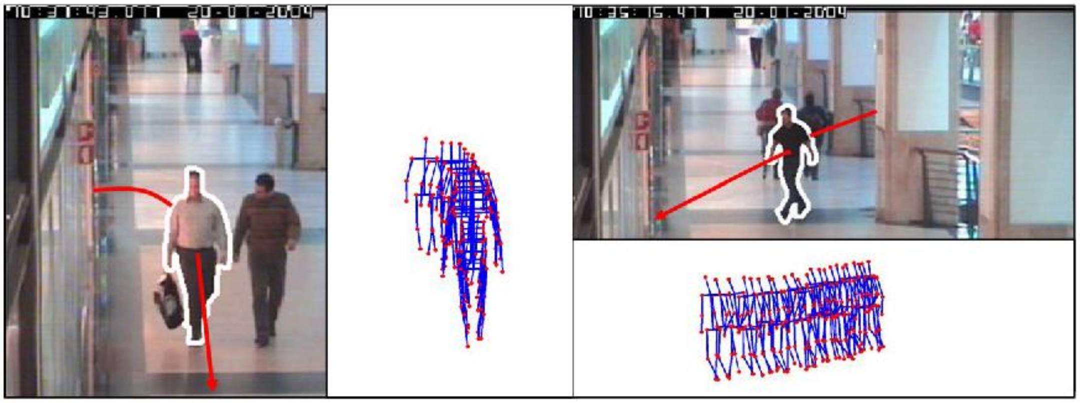


Figure 2.8: Selected Caviar sequences for testing frontal (*left*) and diagonal (*right*) views-based GMM. For each of the 2 sequences, a frame with fitted shape is presented as well as the 2D poses automatically generated when applying the SS-model along the sequence.

more or less to the 4 training views (cf Fig. 2.8) are selected from the Caviar database [Caviar, 2004]. On the one hand, groundtruth data are constructed by manually extracting the silhouettes of selected people appearing in the scene and on the other hand, human blobs are calculated by motion detection. Errors will be calculated as Euclidean distances between groundtruth and estimated shapes.

Two kinds of errors can be estimated: Reconstruction and Fitting Errors. The first one is calculated by projecting and reconstructing a groundtruth shape with the model: this error characterizes the ability of the model to generate new silhouettes. The Reconstruction Error decreases and converges logically for the 4 models when augmenting parameter β from Eq.2.6 (see Fig. 2.7a). The Fitting Error is calculated by correcting the shape extracted from the human blob with the model: this error characterizes the ability of the model to correct bad shapes. On Fig. 2.7b, it can be observed how the Reconstruction Error decreases until a minimum value and then starts increasing for the 4 models when augmenting β. This allows us to determine the optimal value of β for every View-based GMM. On Fig. 2.7c, Fitting

Errors obtained when applying GMMs, NN and ICA are compared for the 4 views (4 Caviar sequences).

GMM exhibits best results than both ICA and NN methods, and shows a better capability to reconstruct unseen shapes. Moreover computational cost of GMM mainly appears during the off-line stage (model construction) while the NN method requires an online comparison to the training exemplars. This makes this approach much more feasible for real-time applications with large databases of different poses and motions.

2.5 Conclusions

In this chapter, we have presented a statistical model for human silhouette and the corresponding skeletal structure. This model can be used to estimate human shape and pose along a sequence. The problem of non-linearity is solved by fitting a different Gaussian Mixture Model (GMM) to several training views. Since shape variations of articulated objects are closely linked to the pose evolution along time, we have clustered the total training set using only the structural information projected in the pose Eigenspace. In order to simplify the problem, we have selected only the most non-linear components to perform the clustering of the data in a lower dimensional space. The optimal number of clusters has been determined by considering the mean gaussianity of the GMM.

Finally we have compared this approach to other two methods developed to cope with shape models non-linearity: GMM exhibits best results than both ICA and NN methods, and shows a better capability to reconstruct unseen shapes. Moreover computational cost of GMM mainly appears during the off-line stage (model construction) while the NN method requires an online comparison to the training exemplars. This makes this approach much more feasible for real-time applications with large databases of different poses and motions.

The 4 training views considered in this chapter are obviously not sufficient to model all the possible orientations of the subject w.r.t the camera in real cases. A more complete model must be built. A possibility is to increase the feature data base considering other camera viewpoints. In the next chapter, these models will be included in a global multi-view 2D models framework. The dynamical correspondences we have established between view-based GMM will be taken into account to manage the eventual viewpoint changes along sequences.

3

A Framework of Spatio-temporal Models

3.1 Introduction

One of the difficulties when employing 2D-models relies on dealing with the viewpoint issue. During many years, most of the related work has been based on the fundamental assumption of "in-plane" motion or only presented results obtained from data satisfying such condition [Zhang et al., 2004, Ning et al., 2004a]. Relatively few papers considered motion-in-depth and out-of-plane rotation of the tracked people. In the last years, some authors have proposed a common approach consisting in discretizing the camera viewpoint and considering a series of view-based 2D models [Zhang et al., 2005b, Lee and Elgammal, 2006, Lan and Huttenlocher, 2004]. This method gives some good results, but there are two main problems that need to be addressed: spatial discontinuities due to the viewpoint discretization and temporal discontinuities due to the difficulties of maintaining the dynamics of the motion when the view is switched. It appears as a challenging problem to establish motion correspondences between viewpoints without considering a mapping to a complex 3D model. Therefore, the goal of the work presented in this chapter is to construct 2D dynamical models 1) that can perform independently of the orientation of the person with respect to the camera and 2) that can respond robustly to any change of direction during the sequence.

3.1.1 Overview of the work

We extend the model presented in the previous chapter by considering additional training viewpoints and complete the ring of possible viewpoints around the subject varying the azimuth angle of the camera. Again, we consider the walking action, but the methodology can be extended to any type of action.

As in chapter 2, 2D pose and 2D appearance parameters are first extracted from training images of the same gait sequences observed from the considered viewpoints. The resulting set of 2D shape and skeletons is then clustered following both spatial and temporal criteria, the spatial clustering being directly provided by the training views and the temporal clustering resulting from motion-based partitioning, i.e. the steps of the gait cycle. A *spatio-temporal clustering* is thus obtained in the global Shape-Skeleton eigenspace: the different clusters correspond in terms of dynamic (temporal clusters) or viewpoint (spatial clusters).

A local 2D-model is then built for each spatio-temporal cluster, generalizing well for a particular training viewpoint and state of the considered action. All those models are concatenated and sorted, what leads directly to the construction of the global Spatio-Temporal 2D-Models Framework (STMF) presented in Fig. 3.1.

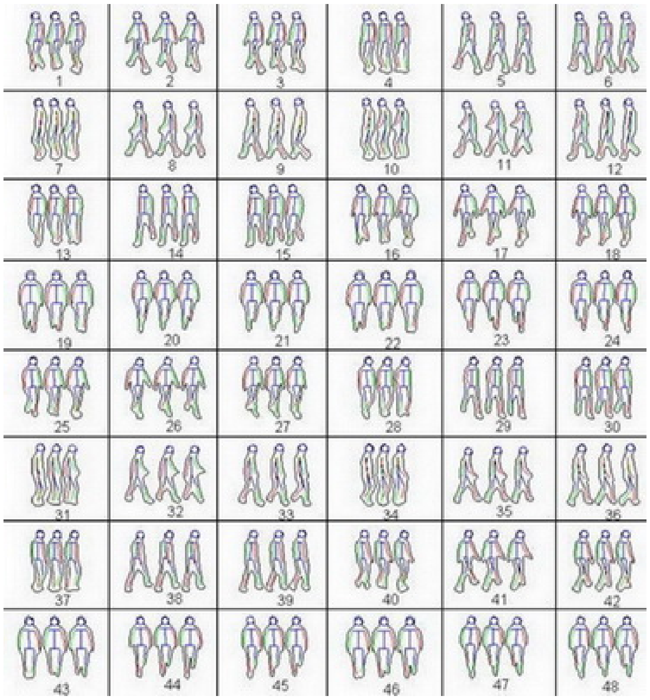


Figure 3.1: Spatio-temporal Shape-Skeleton Models Framework: 1st Variation Modes of the 48 local Models that compone the framework. The columns of this matix correspond to the gait steps (temporal clusters) while the rows represent the 8 camera views (spatial clusters).

These spatio-temporal models generalize well for a particular viewpoint and state of the tracked action but some spatio-temporal discontinuities can appear along a sequence, as a direct consequence of the discretization. Additionaly, an efficient search method is required to guide the exploration of a large feature space. To overcome these problems, we propose to consider *temporal* and *spatial constraints* and build a Probabilistic Transition Matrix (PTM). This matrix limits the search in the feature space by giving a frame to frame estimation of the most probable local models to be considered during the fitting procedure. Our approach is similar to [Lan and Huttenlocher, 2004] in that we integrate spatial and temporal models into a common framework, but differs in that we consider a combined transition that takes into account simultaneous state and viewpoint changes. This constraint-based search is described in Section 3.3.

Once the model has been generated (off-line), it can be applied (on-line) to real sequences. Given an input human blob provided by a background subtraction algorithm, the model is fitted to jointly segment the body silhouette and infer the posture. This model fitting is explained in Section 3.4.

Experiments are presented in Section 3.5 where both segmentation and 2D pose estimation are tested. The main goal of this part is to test the robustness of the approach w.r.t. the viewpoint changes with realistic conditions: indoor, outdoor, cluttered background, shadows etc. In that way, the following hypothesis will be considered to select the different testing sequences: only one walking pedestrian per sequence, with no occlusions but with important viewpoint changes. Note that both training and testing sets comprise of hand-labelled data. The

CMU MoBo database [Gross and Shi, 2001] will be used for training and real video-surveillance sequences for testing [Caviar, 2004]. The HumanEVA dataset, recently introduced by Sigal and Black [Sigal et al., 2010], will be considered for numerical evaluation of the pose estimation.

Section 3.6 finally concludes with some discussions.

3.2 Framework Construction

Recently, some authors have proposed a common approach consisting in discretizing the space considering a series of view-based 2D models [Zhang et al., 2005a, Lan and Huttenlocher, 2004]. In the same way, 8 different viewpoints will be considered, uniformly distributed between 0 and 2π, thus discretizing the frontal view (vertical image plane) into 8 sectors. For each sequence, the 4 training viewpoints used up to that point are now completed by a 5th supplementary back view that is also manually labelled. Finally, the 3 last missing views are interpolated using the periodicity and symmetry of human walking. By this process, a complete training database is generated encompassing more than 20000 Shape-Skeleton vectors, SS-vector (more than 2500 vectors per viewpoint). The resulting 8 view-based Shape-Skeleton associations for a particular snapshot of the CMU MoBo database are presented in Fig. 3.2.

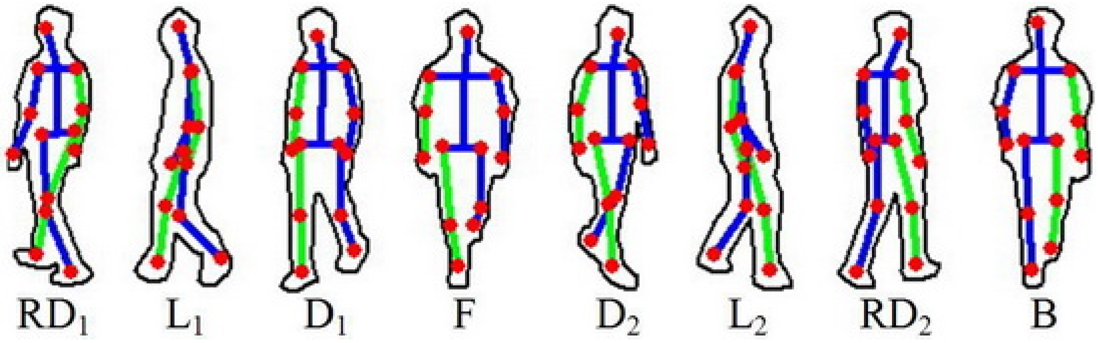


Figure 3.2: Training views considered for Framework Construction.

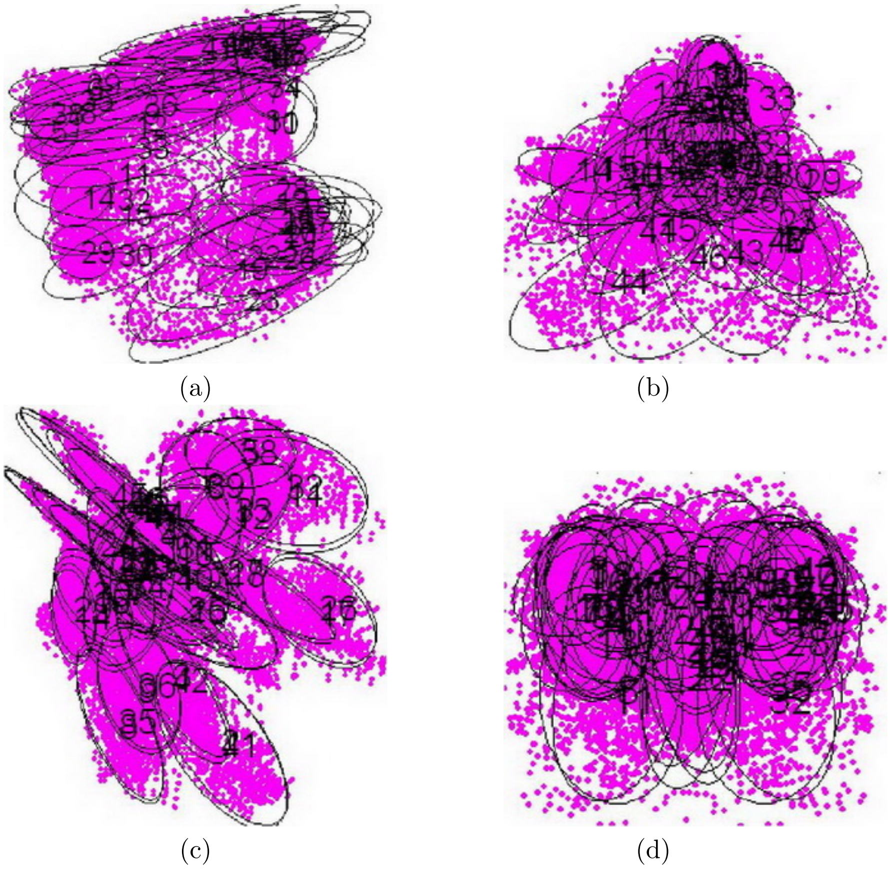
The complete set of SS-vectors is concatenated in a common space (the 8 views together) whose dimensionality is reduced using PCA, obtaining:

$$\mathbf{v}_{\mathrm{i}} \simeq \bar{\mathbf{v}} + \Phi_{\mathrm{v}} \mathbf{a}_{\mathrm{i}}, \tag{3.1}$$

where $\bar{\mathbf{v}}$ is the mean SS-vector, $\mathbf{\Phi}$ is the matrix of Eigenvectors and $\mathbf{a}_i$ is the new vector represented in the Eigenspace. Let us call $\mathcal{A}$ the Shape-Skeleton Eigenspace $\{\mathbf{a}_i\}$.

A series of local dynamic motion models has been learnt by clustering the structural parameters subspace. As mentioned in the Section 2.3.1, the gait cycle is divided into 6 basic steps, providing the temporal clusters C_j, while the 8 training views directly provide the spatial clustering (clusters R_r). The different clusters correspond in terms of dynamics or viewpoint. Using this structure-based partitioning and the correspondences between training viewpoints, 48 spatio-temporal clusters $\{\{T_{j,r} = C_j \cap R_r\}_{j=1}^{6}\}_{r=1}^{8}$ are obtained in the global shape-skeleton feature space where all the views considered are projected together.

Thus, following [Bowden et al., 2000], a local linear model is learnt for each spatio-temporal cluster $T_{j,r}$ and a mixture of PCA is fitted to the clustered $\mathcal{A}$ space, obtaining a new Spatio-Temporal 2D Models Framework (STMF). For each cluster, the local PCA leads to the extraction of local modes of variation, in which both shape and skeleton simultaneously deform


(a)

(b)

(c)

(d)

Figure 3.3: Multi-view Gaussian mixture model represented in the pose-silhouette eigenspace. The 48-clusters Gaussian Mixture Model is represented together with the training data and projected onto the planes defined by (a) 1st and 2nd, (b) 3rd and 4th, (c) 5th and 6th and (d) 7th and 8th components of the Shape-Skeleton Eigenspace.

(see Fig. 3.4). Parameters for the 48 Gaussian mixture models components are determined using EM algorithm. The prior Shape-Skeleton model probability is then expressed as:

$$\mathrm{p}_{mix}(\mathbf{a}) = \sum_{j,r} \omega_{j,r}\, \mathcal{N}(\mathbf{a} : \bar{\mathbf{a}}_{j,r}, \sigma_{j,r}), \tag{3.2}$$

where $\mathbf{a}$ is the eigen-decomposition of the Shape-Skeleton vector, $\mathcal{N}(\mathbf{a} : \bar{\mathbf{a}}, \sigma)$ is the p.d.f. of a gaussian with mean $\bar{\mathbf{a}}$ and covariance σ and $\omega_{j,r}$ is the mixing parameter corresponding to $T_{j,r}$.

The figure 3.3 shows the mixture projected onto various planes of the Eigenspace space $\mathcal{A}$. The 48 hyper-ellipsoids corresponding to the 48 local spatio-temporal models are also plotted. It can be appreciated how well the GMM delimits the subspace of valid SS-vectors.

Given this huge amount of data, an efficient search method is required. In that way, temporal and spatial constraints will be considered to constrain the evolution through the STMF along a sequence and limit the feature space only to the most probable models of the framework.

3.3 Constraint-based search

The total space has been clustered following temporal approach (clusters C_j) as well as spatial approach (clusters R_j) as described in the previous Section. The first one partitions the dynamics of the motion, and the second one, the viewpoint i.e. the direction of motion in the image. The purpose of the following probabilistic modeling is to obtain a transition matrix combining both spatial and temporal constraints.

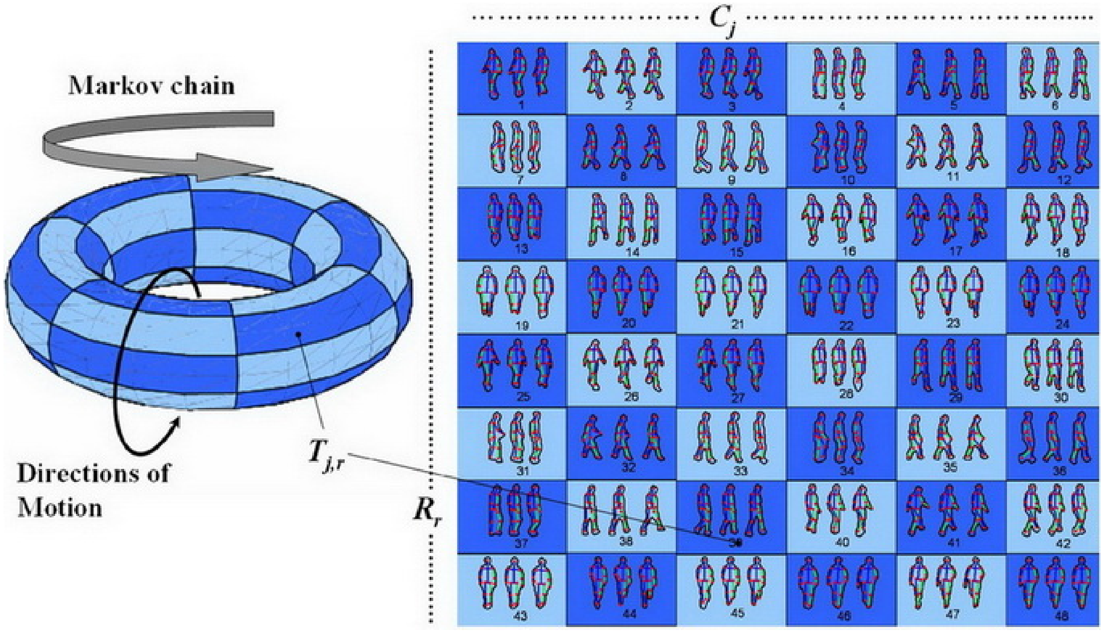


Figure 3.4: 3D (*left*) and 2D (*right*) representations of the toroidal Probabilistic Transition Matrix (PTM). The 1st Variation Modes of the 48 local Models that compone the framework are superposed with the 2D representation of the PTM: the 6 columns correspond to the 6 temporal clusters C_i while the 8 rows represent the 8 spatial clusters R_i.

3.3.1 Markov Chain for Modeling Temporal Constraint

Following the standard formulation of probabilistic motion model [Sidenbladh et al., 2002], the temporal prior $p(S_t|S_{t-1})$ satisfies a first-order Markov assumption where the choice of the present state S_t is made upon the basis of the previous state S_{t-1}. In the same way, if this state space is partitioned into N clusters $\mathcal{C} = \{C_1, ..., C_N\}$, the conditional probability mass function defined as $p(C_j^t|C_k^{t-1})$ corresponds to the probability of being in cluster j at time t conditional on being in cluster k at time *t-1* [Bowden et al., 2000]. The NxN State Transition Matrix (STM) computed in the previous Section points out the probabilities density function (pdf).

3.3.2 Modeling Spatial Constraint

In this chapter, a novel spatial prior $p(D_t|D_{t-1,...t-\mathrm{m}})$ is introduced for modeling spatial constraint. It expresses the statement that D_t (the present direction of motion of the observed pedestrian in the image) can be predicted given its m previous directions of motion $(D_{t-1}, D_{t-2}, ..., D_{t-\mathrm{m}})$. In this approach, the continuous values of all possible camera viewpoints are discretized. Consequently, the direction of motion in the image plane D_t takes a fixed set of values corresponding to the discrete set of M training viewpoints and M clusters $\mathcal{R} = \{R_{1}, ..., R_{M}\}$ in the feature space.

Let $\Delta_t = [R^{t}_{k_0}, R^{t\text{-}1}_{k_1}, ..., R^{t\text{-}m}_{k_m}]$ be the $m{+}1$-dimensional vector representing the sequence of the $m{+}1$ cluster labels (denoted by k_i) up to and containing the one at time t. It has to be noted that some of these k_i labels might be the same. Consequently, $p(R^t_j|\Delta_{t\text{-}1})$ is the probability of being in R_j at time t, conditional on being in R_{k_1} at time t-1, in R_{k_2} at time t-2, etc. (i.e. conditional on the m preceding clusters). In this chapter, a reasonable assumption is made that this direction of motion follows a normal distribution, with expected value equal to the local mean trajectory angle $\overline{\theta}_t$ and, variance calculated as a function of the sampling rate:

$$p(R^t_j|\Delta_{t\text{-}1}) = p(R^t_j|R^{t\text{-}1}_{k_1}, R^{t\text{-}2}_{k_2}, ..., R^{t\text{-}m}_{k_m}) \sim \mathcal{N}(\overline{\theta}_t, \sigma), \tag{3.3}$$

where $\overline{\theta}_t = \frac{1}{m+1}\sum_{\mathrm{i}=t}^{t\text{-}m} \theta_i$, being m a function of the sampling frequency.

3.3.3 Combining Spatial and Temporal Constraints

Let T be the NxM matrix, whose columns represent the N temporal clusters and rows correspond to the M spatial clusters. Thus the probability $p(C^t_j \cap R^t_r) = p(T^t_{j,r})$ denotes the unconditional probability of being in C_j and in R_r at time t.

The conditional spatio temporal transition probability is therefore defined as $p(T^t_{j,r}|C^{t-1}_k, \Delta_{t\text{-}1})$, the probability of being in C_j and in R_r at time t conditional on being in temporal cluster k at time t-1 and conditional on the m preceding spatial clusters. In this thesis, the assumption is made that the two considered events, state and direction changes, are independent, even if it is not strictly true. Some comments about this assumption will be made in Sections 3.5 and 3.6. This leads to the following simplified equation:

$$p_{j,r} = p(T^t_{j,r}|C^{t-1}_k, \Delta_{t\text{-}1}) \propto p(C^t_j|C^{t-1}_k).p(R^t_r|\Delta_{t\text{-}1}). \tag{3.4}$$

The resulting NxM matrix is the Probabilistic Transition Matrix (PTM) that gives, at each time step, the probability density function that limits the region of interest in the STMF to the most probable models.

The matrix is in fact a 2D manifold representation for viewpoint and action where the action (consecutive temporal clusters) is represented by a 1D manifold and the viewpoint by another 1D manifold. Because of the cyclic nature of the viewpoint parameter (circular distribution of the training viewpoints), if it is modeled with a circle the resulting manifold is in fact a cylindrical one. When the action is cyclic too, as in our case with gait, the resulting 2D manifold lies on a "closed cylinder" topologically equivalent to a torus. The resulting PTM is thus a toroidal matrix (Fig. 3.4) whose lines correspond to the M training view-based gait manifolds. Its 3D and 2D representations are illustrated in Fig. 3.4. All the different models can be ordered and classified according to their direction of motion and state, thus putting in evidence the correspondences with the PTM as shown in Fig. 3.4. Spatial and temporal relationships can be appreciated between local models from adjacent cells.

The content of the PTM can be visualized by converting it to grey scale image as will be shown in next sections. To compute this PTM and constrain the evolution through the STMF along a sequence, only previous viewpoint and previous state are required at each time step. Note that our approach shares some similarities with the one proposed by Lv and Nevatia [Lv and Nevatia, 2007]. In that paper, the authors model an action as a series of 2D Poses rendered from a wide range of viewpoints and represent the constraints on transition by a graph model where they assume a uniform transitional probability for each link.

3.4 Joint Segmentation and Pose Estimation

A discriminative detector as the one proposed in [Dimitrijevic et al., 2006] could be used to initialize the shape model-driven algorithm presented next. In this work, scene context information is considered to roughly limit the feature space only to the "logical" 2D-models from the framework. For example, if an object appears in the scene from the right side (right-to-left direction of motion), only the 3 first lines of the PTM will be considered.

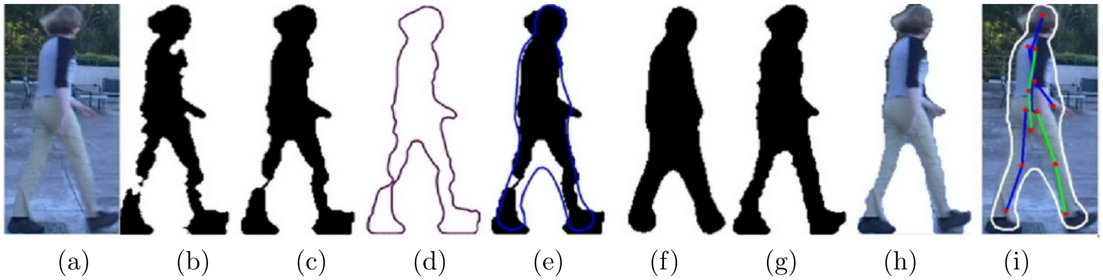


Figure 3.5: Model Fitting: (a) original input image. (b) Silhouette resulting from background subtraction. (c) Silhouette after being processed by *BlobsProcessing*. (d) Contour extracted by silhouette erosion. (e) Corrected shape represented on the Silhouette and corresponding mask (f) used for finer background subtraction. (g) Resulting Silhouette after 6 iterations and (h) corresponding segmentation. (i) Resulting shape and pose plotted on the original input image.

Once the system has been initialized, each frame of the sequence is processed individually by applying *SegmentationPoseEstimation* (Algorithm 1), taking advantage of previous information (Trajectory angle θ, State *index*, Background B) that is used to treat the current frame.

Algorithm 1: ($\mathbf{s}$, $\mathbf{k}$, B, θ, $index$) = *SegmentationPoseEstimation* (B, I, θ, $index$, $nIter$)

```
    $m[i]$ = $ModelsSelection(\theta, index)$;
    initialize shape $\mathbf{s} \leftarrow 0$;
    initialize pose $\mathbf{k} \leftarrow 0$;
    for $n \leftarrow 1$ to $nIter$ do
        blobsList = $AdaptiveBackgroundSubtraction(B, I, \mathbf{s}, n, nIter)$;
        Silhouette = $BlobsProcessing$(blobsList);
        $\mathbf{s}$ = $ContourExtraction$(Silhouette);
        [$\mathbf{s}$,$\mathbf{k}$,index]= $ShapeSkeletonCorrection(\mathbf{s}, \mathbf{k}, m[i])$;
    end for
    $B$ = $BackgroundUpdate(B, I, Silhouette)$;
```

The prediction of the most probable models from the GMM is estimated in *ModelsSelection* by means of the PTM. It allows a substantial reduction in computational cost and can solve some possible ambiguities by considering a limited number of models.

In *ShapeSkeletonCorrection*, the extracted shape $\mathbf{s}$ and an estimate for the skeleton are concatenated in $\mathbf{v} = [\mathbf{s}^\top \overline{\mathbf{k}}^\top]^\top$ and projected into the SS-Eigenspace obtaining $\mathbf{a}$. Then, for each one of the most probable clusters given by the PTM $p_{j,r}$, we update the parameters to best fit the "local model" defined by its mean, eigenvectors and eigenvalues, as done in Sect. 2.3.3, obtaining $\mathbf{a}^*$. The distance between extracted and corrected shapes is then calculated for each one of the estimations in order to select the best estimation. We then project the vector $\mathbf{a}^*$ back to the feature space obtaining $\mathbf{v}^*$ which contains a new estimation of both shape $\mathbf{s}^*$ and skeleton $\mathbf{k}^*$: $\mathbf{v}^* = [\mathbf{s}^{*\top} \mathbf{k}^{*\top}]^\top$.

Algorithm 2: blobsList = $AdaptiveBackgroundSubtraction(B,\, I,\, \mathbf{s},\, n,\, nIter)$

```
if ($n == 1$) then
  D = $BackgroundSubtraction(B,\, I,\, thr)$;
else
  Mask = $ShapeToMask(\mathbf{s})$;
  $t_{low} = DecreaseThreshold(thr,\, n,\, nIter\ )$;
  $D_{low} = BackgroundSubtraction(B,\, I,\, t_{low})$;
  $t_{high} = IncreaseThreshold(thr,\, n,\, nIter)$;
  $D_{high} = BackgroundSubtraction(B,\, I,\, t_{high})$;
  D = $D_{low} \times Mask + D_{high} \times \overline{Mask}$;
end if
blobsList = $BlobsLabelling$(D);
```

Aside from the models and the constraint-based search proposed in this work, some novelties appear in the segmentation algorithm. The first one refers to the shape extraction task (*ContourExtraction* in Alg. 1): while it is usually extracted from the blob looking along straight lines through each model point as in [Baumberg and Hogg, 1994], here the shape is directly obtained by eroding the human blob and normalizing the resulting contour following the shape normalization proposed in Sect. 2.2.2. This allows a direct, precise and faster registration of the shape in the image. The only drawback of this shape registration is that it requires an entire and non-fragmented silhouette. The *BlobsProcessing* function thus previously applies some common morphological operations to the result of *AdaptiveBackgroundSubtraction* (Alg. 2) and connect the possible fragments.

Another novelty of the fitting process appears in *AdaptiveBackgroundSubtraction* (Alg. 2) that aims at reconstructing the binary silhouette resulting from the background subtraction using the "corrected" shape returned by *ShapeSkeletonCorrection*. It is achieved by decreasing/increasing the detection threshold inside/outside the shape. This leads to an accurate silhouette segmentation, improving considerably the results specially when there is no significant difference between background and foreground pixels.

The last novelty relies on the way the background is updated: the final segmented silhouette, the foreground, is used to actualize the Background more finely, eliminating shadows from the foreground and improving the segmentation in next frames.

The different steps of the process are depicted in Fig. 3.5 for a particular frame.

3.5 Experiments

The model is now evaluated with a series of testing sequences illustrating different situations which may occur in the analysis of pedestrian motion: straight line walking, changes of direction, of speed, etc. Only model fitting and pose estimation will be tested in the first set of experiments, and not the tracking in the image, the system is then provided with the bounding-box taken from groundtruth avoiding the possible problems due to the tracking. In the PTM matrices from Fig. 3.6 (as well as from Fig. 3.8 and 3.9), the colored cells represent the probability $p_{j,r}$ from (Eq. 3.4). The obscured cell is the "winning one": the local model that best fits the silhouette. For each frame, the row of the "winning" model in the PTM indicates the orientation of the pedestrian with respect to the camera. Additionally, both trajectory and previous states are respectively plotted in the image/matrix with a white line.

As illustrated in Fig. 3.6 (*up*), the resultant vectors from a pedestrian crossing the scene

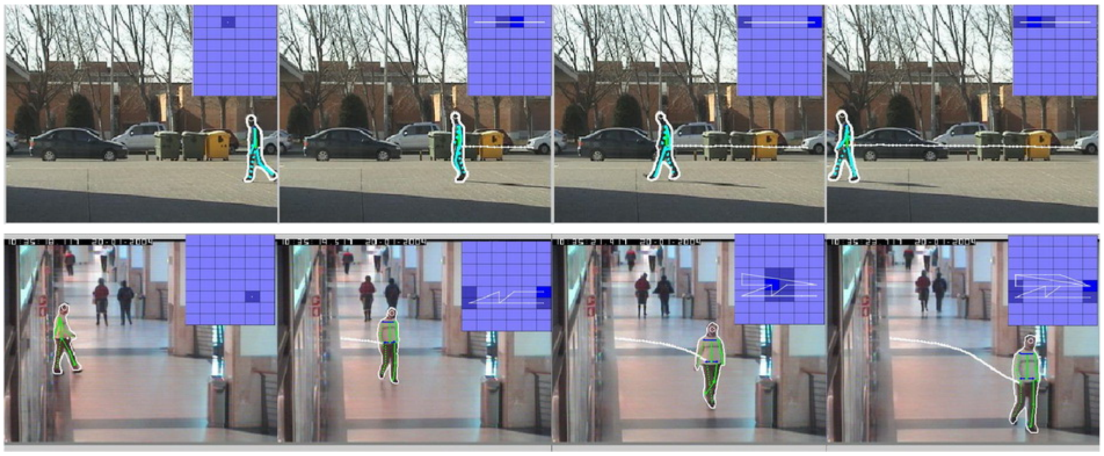


Figure 3.6: Examples of sequences processed with the framework of pose-silhouette models: (*up*) Outdoor straight line walking sequence at constant speed and (*down*) Caviar sequence with slight bend trajectory.

straight ahead without stopping or turning towards anything all belong to models from the same row of the PTM. Any change of direction is observed as a progressive change of row (See Fig. 3.6 (*down*)).

In Figure 3.7, the results obtained with two challenging frames are presented: in (a), the pedestrian is carrying a bag and in (b) he is partially occluded by the wall. In both cases, a plausible estimation is made of both shape and pose.

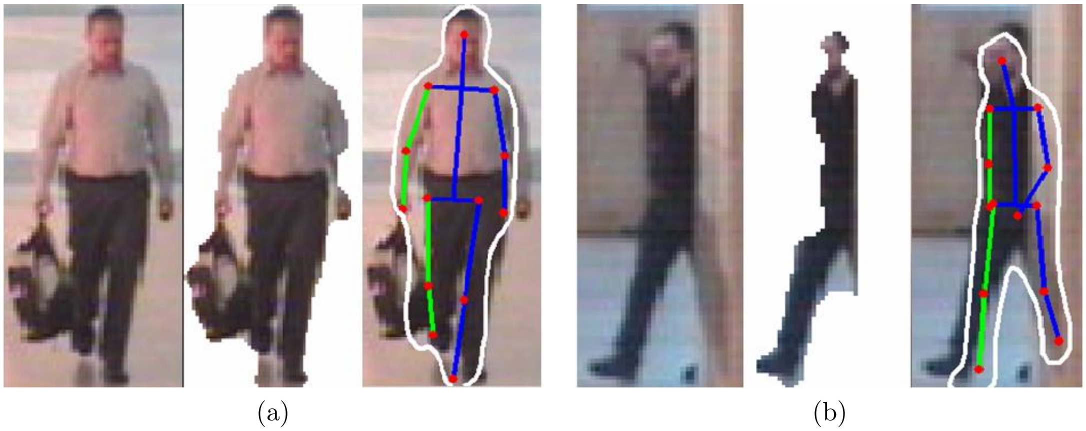


Figure 3.7: Results obtained with two challenging frames. For each of the two examples, original image (*left*), segmentation (*centre*) and resulting pose and shape are represented (*right*).

3.5.1 Framework Validation

To validate the Framework, the 2D poses and 2D shapes of 3 different sequences with different characteristics of interest are hand-labelled: an outdoor straight-line walking sequences at constant speed (Fig. 3.6*up*), an outdoor "Walkcircle" sequences with constant speed and constant viewpoint & scale evolution (Fig. 3.8) and an indoor sequence with viewpoint & speed variations (Fig. 3.9). Note that the subjects turn and move "in depth" so that both apparent scale and viewpoint vary. A top-down estimation of depth is directly provided by the "winning" model that points out the motion direction in 3D space (see Fig. 3.6 and later Fig. 3.8 and 3.9).

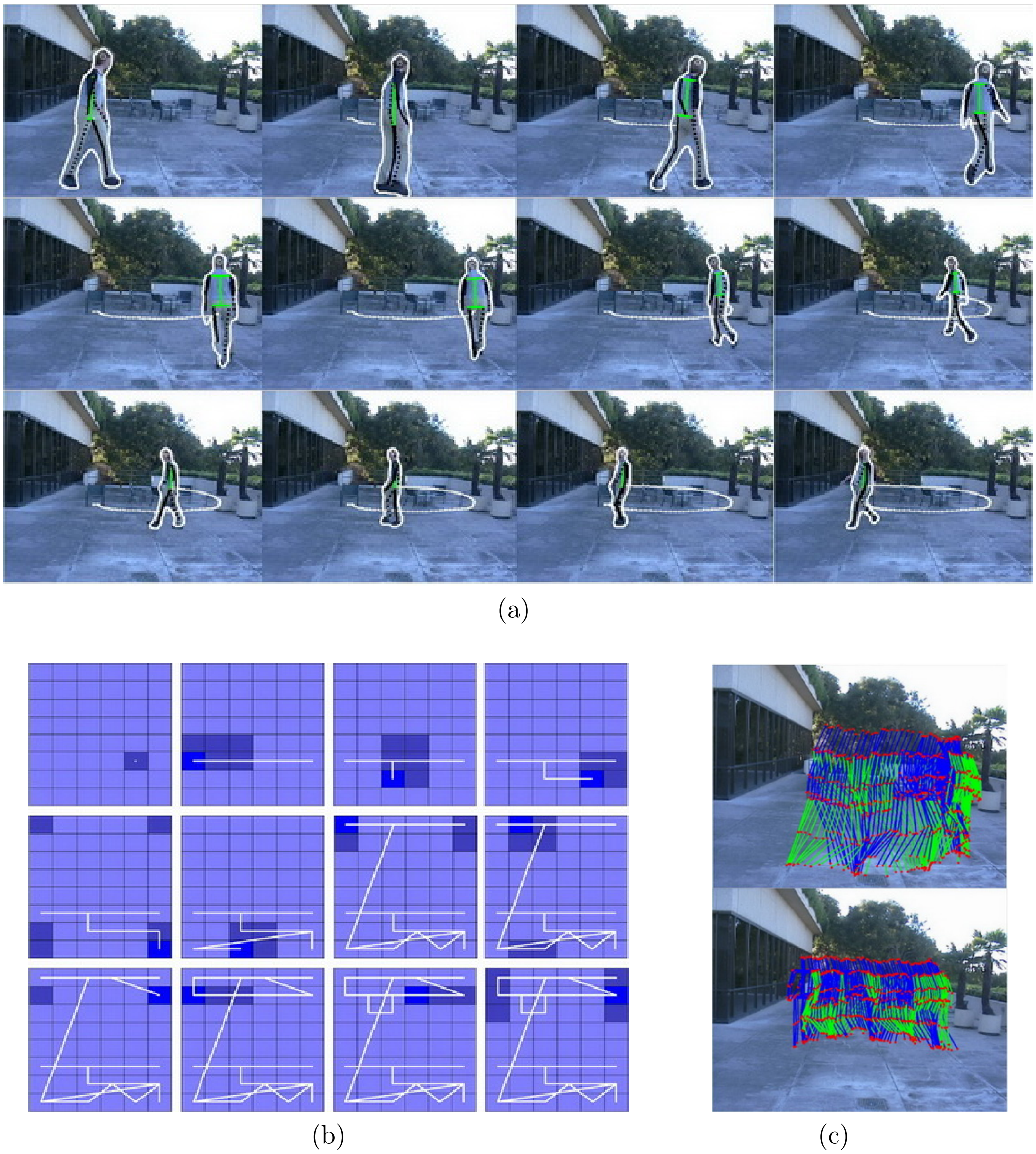
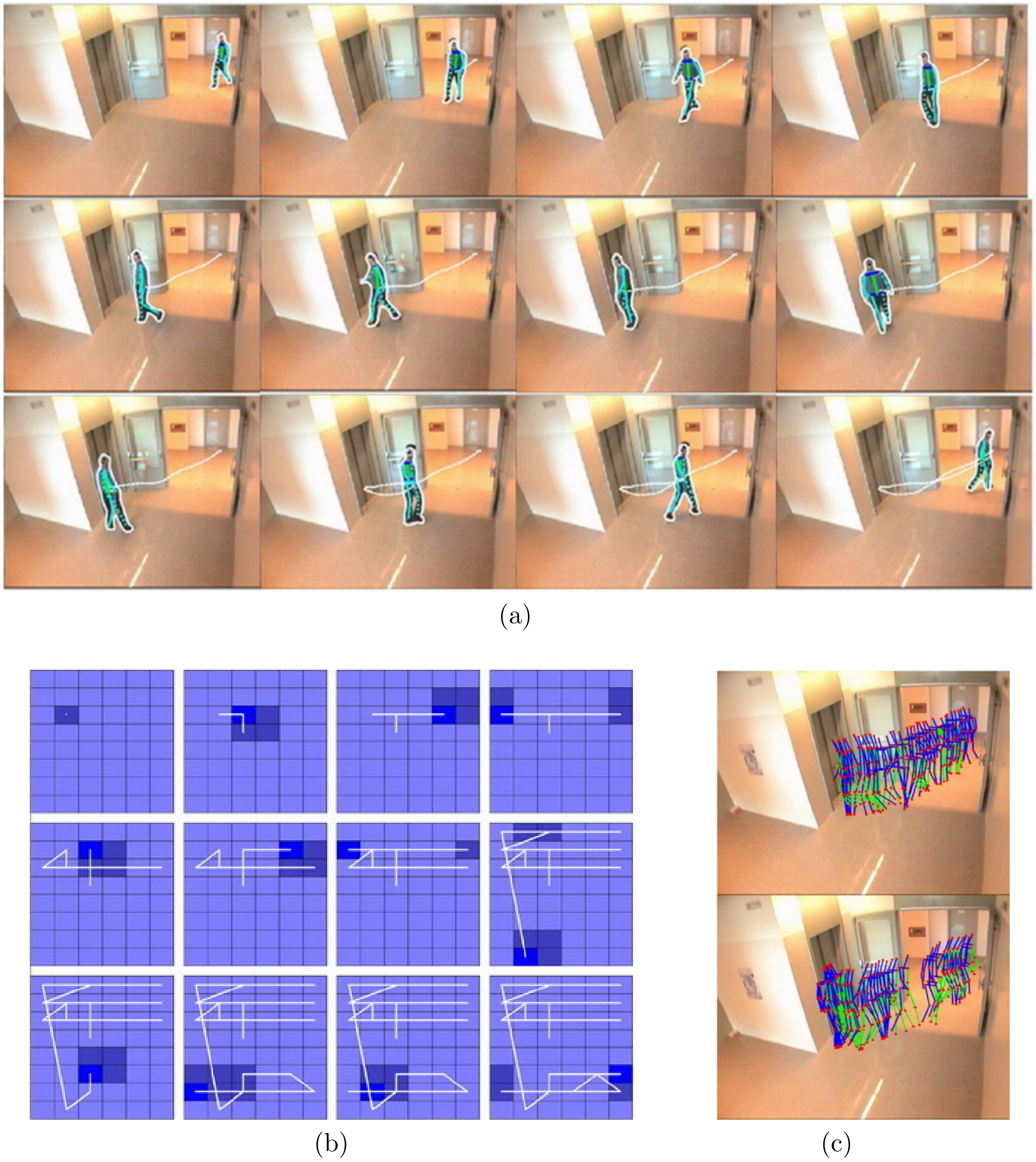


Figure 3.8: Results obtained for the outdoor "Walkcircle" sequence with constant speed and constant viewpoint & scale changes : (a) Estimated shapes and poses represented on the original image for frames 1, 15, 25, 40, 50, 60, 75 90, 100, 115, 130 and 140. (b) 12 corresponding PTM matrices and (c) 2D Poses estimated along the complete sequence.

3.5.1.1 Segmentation

Quantitative validation is performed by comparing with manually segmented solutions, both the segmentation obtained by simple background subtraction and the one resulting from the



(a)

(b)

(c)

Figure 3.9: Results obtained for the indoor "Elevator" sequence with viewpoint & speed changes: (a) Estimated shapes and Poses represented on the original image for frames 1, 18, 30, 38, 48, 58, 68, 88, 100, 122, 128 and 142. (b) 12 corresponding PTM matrices and (c) 2D Poses estimated along the complete sequence.

proposed model-based approach. Denote the manual segmentation in the images as $S_{groundtruth}$, and the results as $S_{estimated}$. We define the false negative ratio (FN) to indicate the fraction of silhouette that is included in the groundtruth segmentation but missed by the automatic method:

$$FN = \frac{|S_{groundtruth} - S_{estimated}|}{S_{groundtruth}}. \qquad (3.5)$$

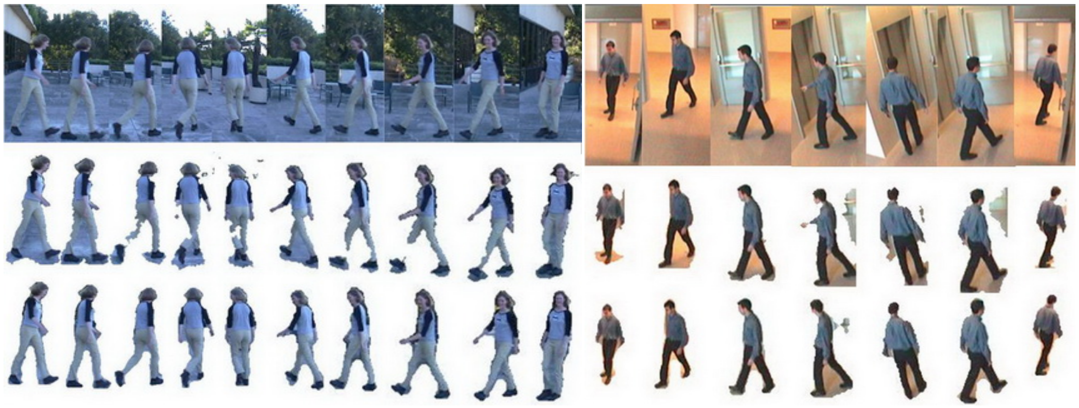


Figure 3.10: Segmentation results for (*left*)“Walkcircle” outdoor sequence and (*right*) for “elevator” indoor sequence. (*up*) the original image, (*centre*) the result obtained by simple background subtraction and (*down*) the result obtained by applying the proposed model-based algorithm are represented for each example.

The false positive ratio (FP) indicates the amount of foreground falsely identified by the algorithm as a fraction of the total silhouette in the groundtruth segmentation:

$$FP = \frac{|S_{estimated} - S_{groundtruth}|}{S_{groundtruth}}. \tag{3.6}$$

The true positive (TP) describes the fraction of the total silhouette in the true segmentation that is overlapped with the proposed method:

$$TP = \frac{|S_{estimated} \cap S_{groundtruth}|}{S_{groundtruth}}. \tag{3.7}$$

Example segmentation results are shown in Fig. 3.10 and average statistics compiled in Tables 3.1 and 3.2. On the outdoor sequence, the segmentation results produce the following: FN ratio is improved by 3.8%, FP by 14.48% and TP by 3.8%. On the indoor sequence, only FP ratio is improved by 7.03% while FN and TP stay unchanged. In both cases, we can observe how part of the shadow is eliminated with the proposed method what leads directly to the False Positive ratio FP improvement.

Table 3.1: Segmentation Results for Outdoor “Walkcircle” Sequence.

	Background Subt.	Model-based Segm.
FN	9.31%	5.51%
FP	27.92%	13.44%
TP	90.69%	94.49%

3.5.1.2 Pose Estimation

Fig. 3.11 shows the pose estimation results for the 3 tested sequences. The mean position error (in pixels) is calculated as the feet-distance between the skeleton estimated by the algorithm

Table 3.2: Segmentation Results for Indoor "Elevator" Sequence.

	Background Subt.	Model-based Segm.
FN	6.79%	6.80%
FP	20.83%	13.80%
TP	93.21%	93.20%

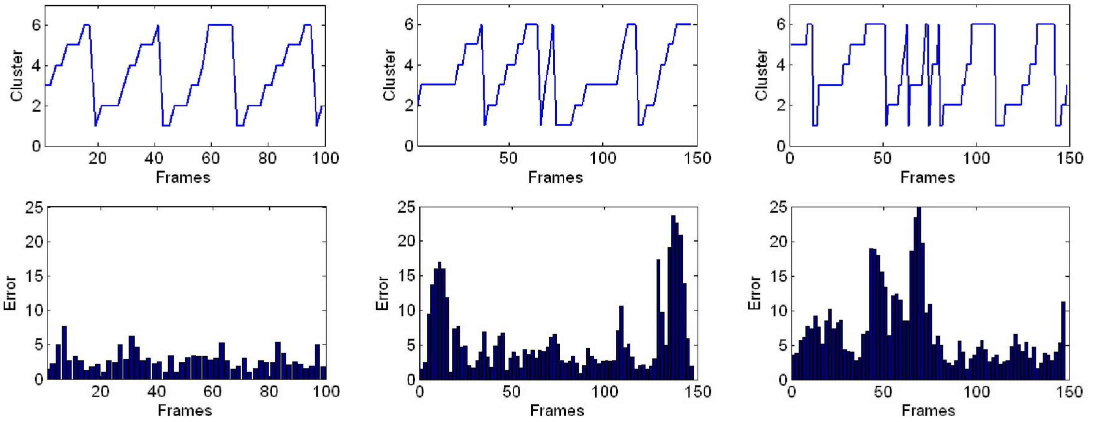


Figure 3.11: Pose estimation results: feet position error in pixels (bottom) and temporal clusters (top) - given by the column of the PTM corresponding to the "winning" model - of the Straight line walking (left), Indoor (centre) and "Walk-circle" (right) sequences.

and the hand-labelled one. Some peaks can be noticed in this figure. For instance, in the indoor sequence (center) the model failed because of the excessive difference of viewpoint-angle between training and input images, when the subject goes in and out of the scene. In the "Walkcircle" sequences (right) the model fails because of the stationary behavior of the tracking that stays stuck in a cluster during too many frames and then can hardly get out of it. It needs to wait until the next cycle to recuperate the dynamic behavior of the input motion. This is due to the very low shape variability in the back view where it is very complicated to distinguish a state from another. For the rest of the frames, the results are globally very satisfactory which means that the model is conveniently tuned to the suitable viewpoints and that the assumption of independency of spatial and temporal event, made in Section 3.3.3, is reasonable.

3.5.2 Numerical Evaluation with HumanEva dataset

For numerical evaluation of the Framework, the 4 walking sequences of the HumanEva-II dataset [Sigal et al., 2010] are considered: subjects S2 and S4 observed from camera C1 and camera C2.

Note that the groundtruth is not available for these sequences and that for each frame, the boundingbox is estimated using a simple Kalman filter. The good results obtained with such setting demonstrate that the method behaves quite well even if it is not provided with the exact boundingbox taken from groundtruth.

Segmentation and estimated 2D poses resulting from the proposed model-based approach are presented together in Fig. 3.12 while numerical evaluation is given in Fig. 3.13. This evaluation has been obtained using the on-line evaluation system and the metrics provided for 2D pose estimation i.e. the average distance in pixels over all the 13 2D key-points of the Stick model. For each sequence, this error is shown for all the processed frames in Fig. 3.13 and the average error per sequence (over all the frames) is given in Table 3.3.

In the 4 sequences, the human body was segmented and tracked successfully as can be seen in Fig. 3.12, maintaining the sequentiality of the motion even if some pics can be observed in the error curve. However, the average difference is quite high in all the frames even when the result is shown to be visually accurate in Fig. 3.12. This can be explained by the differences in defining the joint centers in the proposed skeleton model (constructed from hand labelled data) and in the marker-based system, that causes an offset clearly observable in Fig. 3.13.

Table 3.3: 2D Pose Average Error on HumanEVA data set.

Subject	Camera	Start	End	Mean Error
S2	C1	1	350	16,96 *pix*
S2	C2	1	350	18,53 *pix*
S4	C1	1	290	16,36 *pix*
S4	C2	1	290	14,88 *pix*

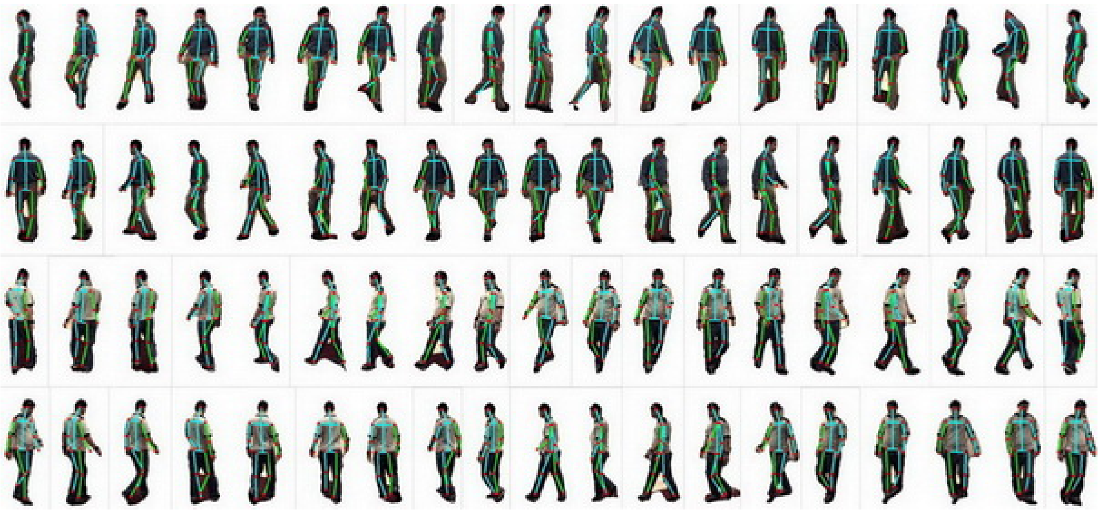


Figure 3.12: Segmentation and 2D pose estimation obtained for the 4 HumanEva testing sequences. From up to down: Subject S2, camera views C1 (1^{st}*line*) and C2 (2^{nd}*line*), and Subject S4, camera views C1 (3^{rd}*line*) and C2 (4^{th}*line*). For each sequence, frames 1, 20, 40, 60, 80 ... 300, 320, 340 and 350 are represented.

3.6 Summary

In this chapter we have presented a novel probabilistic spatio-temporal 2D-models framework for human motion analysis. In this approach, the 2D shape of the entire body has been associated to the corresponding stick figure (skeleton) allowing the joint segmentation and pose recovery of the subject observed in the scene along a sequence.

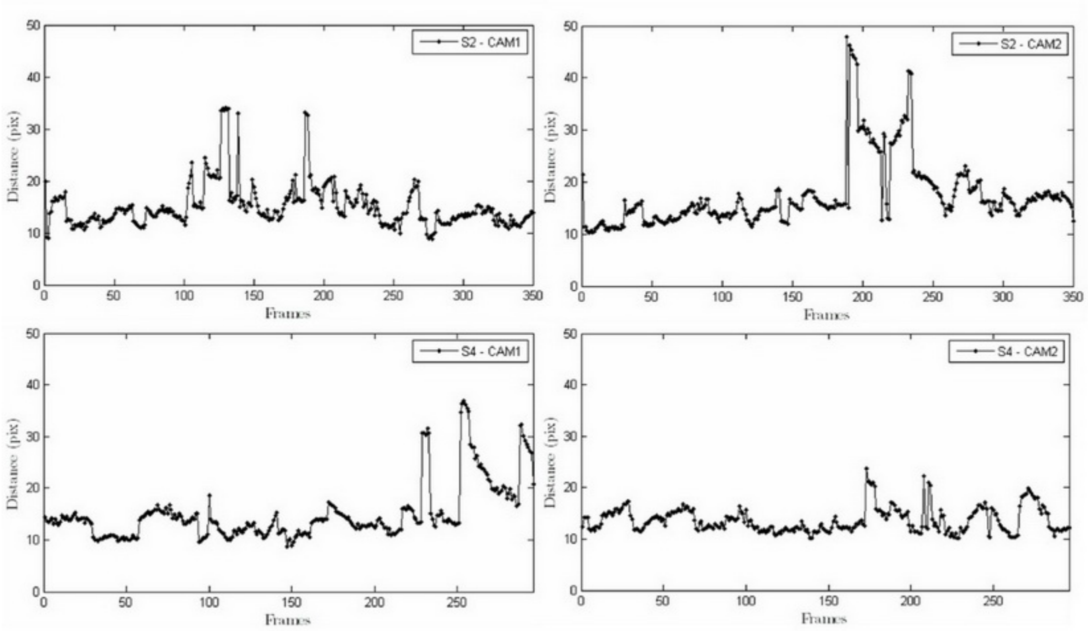


Figure 3.13: Numerical Results obtained for the 4 HumanEva testing sequences: for Subject S2 (*up*) and S4 (*down*). In both cases, the average error of 2D pose reconstruction is given for camera views C1 (*left*) and C2 (*right*).

To cope with the restriction to the viewpoint, local spatio-temporal 2D-models corresponding to many views of the same sequences were trained, concatenated and sorted in a global framework (a multi-view Gaussian mixture model). When processing a sequence, temporal and spatial constraints are considered to build the Probabilistic Transition Matrix (PTM) that gives the frame to frame prediction of the most probable models from the framework. The proposed fitting algorithm, combined with the new probabilistic models, allows a more reliable estimation of both pedestrian silhouette and 2D pose in real monocular sequences. The experiments carried out on both indoor and outdoor sequences have demonstrated the ability of this approach to adequately segment the pedestrians and estimate their postures independently of the direction of motion during the sequence. They have also demonstrated that the method responds quite robustly to any change of direction during the sequence.

In this part of the thesis, only one value has been considered for the elevation angle. A possibility to handle large viewpoint changes (when using roof-top cameras for example) could be to train the model with several values of this tilt angle as in [Lv and Nevatia, 2007]. The supplementary angle variation could then be represented by an additional third dimension in the Toroidal Transition Matrix in order to keep the spatial continuity between viewpoints of connected cells. Another possibility to deal with different tilt angles and decrease the effect of perspective distortion is to consider a projective transformation between training and testing images as we proposed in the next part of this thesis.

Part II

Pose Tracking in Video-Surveillance Environments

4

View-invariant Motion Analysis using View-based Models

4.1 Introduction

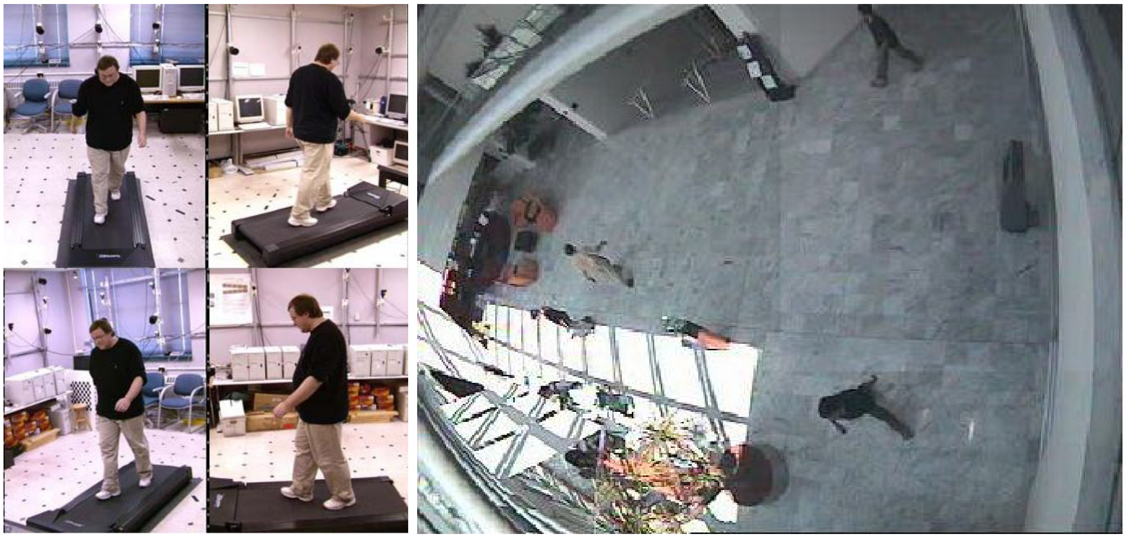
The second part of this thesis is dedicated to the problem of pose estimation and tracking in video-surveillance scenarios. In recent years, the number of cameras deployed for surveillance and safety in urban environments has increased considerably in part due to their falling cost. The potential benefit of an automatic video understanding system in surveillance applications has stimulated much research in computer vision, especially in the areas related to human motion analysis. The hope is that an automatic video understanding system would enable a single operator to monitor many cameras over wide areas more reliably.

As demonstrated in the introduction chapter, exemplar-based approaches have been very successful in the different stages of human motion analysis: detection, pose estimation and tracking. The main disadvantage of the techniques based on training exemplars is their direct dependence on the point of view: the accuracy of the result strongly depends on the similarity of the camera viewpoint between testing and training images. Ideally, to deal with viewpoint dependency, one could generate training data from infinitely many camera viewpoints, ensuring that any possible camera viewpoint could be handled. Unfortunately, this set-up is physically impossible and makes the use of real data infeasible. It could, however, be simulated by using synthetic data, but using a large number of views would drastically increase the size of the training data. This would make the analysis much more complicated; furthermore, the problem is exacerbated when considering more actions.

In practice, roof-top cameras are widely used for video surveillance applications and are usually placed at a significant angle from the floor, which is different from typical training viewpoints as shown in the example in Fig. 4.1. Perspective effects can deform the human appearance (e.g. silhouette features) in ways that prevent traditional techniques from being applied correctly. Freeing algorithms from the viewpoint dependency and solving the problem of perspective deformations is an urgent requirement for further practical applications in video-surveillance.

The goal of the work presented in this chapter is to track and estimate the pose of multiple walking people independently of the point of view from which the scene is observed (see Fig. 4.2a), even in cases of high tilt angles and perspective distortion.

We have seen that our framework of view-based models performs decently when the camera viewing direction is parallel to the ground ($\varphi \simeq 0$) and an estimate of both 2D pose and camera viewpoint can be made in spite of the discretization of the training viewpoint. But when using



(a) (b)

Figure 4.1: Video sequences considered in the chapter: (a) CMU Mobo database [Gross and Shi, 2001] for training and (b) video-surveillance Caviar database [Caviar, 2004] for testing. The CMU snapshot in (a) is represented from 4 different viewing angles *(clockwise from upper left)*: frontal, diagonal-rear, lateral and diagonal. The difference of viewing angle can be observed between training and testing sequences.

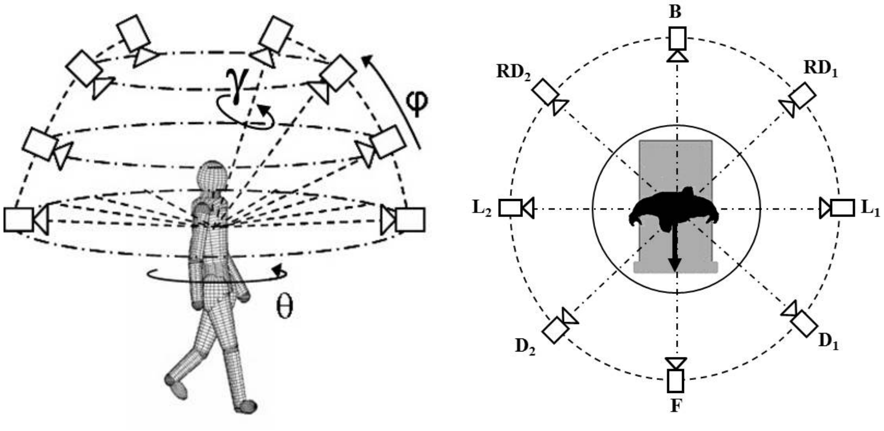
roof-top camera sequences, a pre-processing of the input image is necessary for perspective correction and correct view alignment.

The challenge is then to make use of those models successfully on any possible sequence taken from a single fixed camera with an arbitrary viewing angle. A solution is proposed to the paradigm of *"View-insensitive process using view-based tools"* for video-surveillance applications in man-made environments: supposing that the observed person walks on a planar ground in a calibrated environment, we propose to compute the homography relating the image points to the training plane of the selected viewpoint. The input image is then warped to this training view and a pose is estimated using the corresponding view-based models.

4.1.1 Related Work

In addition to the problem of viewpoint dependency of the model, we will have to overcome the classical difficulties that appear when tracking people in complex, but real, surveillance video-sequences. These difficulties are quite common: people moving in groups, occlusions, shadows/reflections on the ground/wall, low image resolution and especially the small size of the subjects that makes the pose estimation much more complicated.

Many surveillance systems can be found in the literature: for example, W^4 [Haritaoglu et al., 2000], BraMBLe [Isard and MacCormick, 2001] and ADVISOR [Siebel and Maybank, 2002a]. But those systems only consider data where multiple people are distributed horizontally in the image. Promising tracking results were presented by Zhao and Nevatia [Zhao and Nevatia, 2004] in conditions similar to those we propose: perspective images with camera model known and the assumption that people walk on a known ground plane. They first locate people by detecting the head as in [Haritaoglu et al., 2000], then use a coarse 3D shape model (an ellipsoid) for global motion tracking as done by Isard with a cylinder [Isard and MacCormick, 2001]. Finally they employ a locomotion model to infer the 3D human posture. Even though



(a) (b)

Figure 4.2: (a) Viewing hemisphere: the position of the camera with respect to the observed subject (the view) can be parameterized as the combination of two angles: the *elevation* $\varphi \in \left[0, \frac{\pi}{2}\right]$ (also called latitude or tilt angle) and the *azimuth* $\theta \in [-\pi, \pi]$ (also called longitude). A third angle $\gamma \in [-\pi, \pi]$ can be considered to parameterize the rotation around the viewing axis. (b) Viewpoint discretization for training: in this work, we use the MoBo dataset [Gross and Shi, 2001] and discretize the viewing hemisphere into 8 locations where θ is uniformly distributed around the subject. An example is given in Fig. 4.1.a for front (F), rear-diagonal (RD2), lateral (L1) and diagonal (D1) views.

our work shares similarity with [Zhao and Nevatia, 2004], there are two major differences: 1) in our work, segmentation, tracking and pose estimation will be done all together using a more detailed silhouette-pose model and 2) we take into account the possibility of a very large tilt angles.

Viewpoint dependence has been one of the bottlenecks for research development of human motion analysis as indicated in a recent survey [Ji and Liu, 2010]. Some work has been done on solving the problem of viewpoint dependency. In [Cucchiara et al., 2005], a calibrated approach is used in order to avoid perspective distortion of the extracted features. Farhadi and Tabrizi [2008] propose a method to build features that are highly stable under change of camera viewpoint and recognize action from new views. Recently, Gong and Medioni [2011] achieved view-invariant action recognition on videos by associating a few motion capture examples using a novel Dynamic Manifold Warping (DMW) alignment algorithm. In [Parameswaran and Chellappa, 2004], the authors present a method to calculate the 3D positions of various body landmarks given an uncalibrated perspective image and point correspondences in the image of the body landmarks. They also address the problem of view-invariance for action recognition in [Parameswaran and Chellappa, 2006]. Grauman et al. [2004] propose a solution for inferring a 3D shape from a single input silhouette with an unknown camera viewpoint. The model is learnt by collecting multi-view silhouette examples from a calibrated camera ring and the visual hull inference consists in finding the shape hypotheses most likely to have generated the observed 2D contour. The concept of "virtual cameras", introduced in [Rosales et al., 2001], allows for the reconstruction of synthetic 2D features from any camera location. The joint log-likelihood of body pose and camera parameters is maximized and results in an estimate of the 3D body pose. In [Kale et al., 2003], a method is proposed for view invariant gait recognition: considering a person walking along a straight line (making a constant angle with the image plane), a side-view is synthesized using a homography. Recently, Datta et al. [2011] described a motion estimation algorithm for projective cameras that explicitly enforces articulation constraints and presented pose tracking results for binocular sequences. In [Bouchrika et al., 2009] the authors propose a reconstruction method to rectify and normalize gait features recorded from different viewpoints into the side-view plane, exploiting such data for human recognition. The rectification method is based on the anthropometric properties of human limbs and the characteristics of the gait action [Goffredo et al., 2008]. In [Rogez et al., 2006a] we proposed an algorithm that uses a projective transformation between training and testing images to find viewpoint invariance. This paper will be discussed in details in the next section. In the same spirit, Li et al. [2008] later employed a homographic transformation to improve human detection in images presenting perspective distortion. They reported an improvement in detection rate from 38.3% to 87.2% using 3D scene information instead of scanning over 2D (plus in-plane rotation) on the same testing dataset [Caviar, 2004] we consider in this chapter.

4.1.2 Motivation and Overview of the Approach.

As discussed in [Riklin-Raviv et al., 2007], in presence of perspective effect, the distortion will cause the parts of the subject that are closer to the lens to appear abnormally large, thus deforming the shape of the human contour in ways that can prevent a correct analysis.

The basic idea is that projective geometry could be exploited when camera viewpoints in training and testing images are too different. In [Rogez et al., 2006a], we numerically demonstrated that the use of a projective transformation for shape registration, projecting both model and image in a canonical vertical view, improves silhouette-based pose estimation. In that case, the parameters required to estimate the homography, i.e. the subject's location on the ground plane (X, Y) and the viewpoint θ (Fig. 4.2a), were taken from ground truth data.

In this chapter, a classical process of Detection-Segmentation-Tracking (see Fig.4.3up) is considered during the processing of a video-surveillance sequence with arbitrary viewpoint. The pedestrians are tracked using a Kalman filter to automatically estimate X, Y and θ: the tracking is applied on the ground floor and the orientation with respect to the camera (the viewpoint) is estimated by approximating it by the direction of walk. In a calibrated environment, a good estimation of the ground plane position can be obtained by projecting vertically the head location. An advantage of sequences taken by a rooftop camera, is that the head is usually less likely to be occluded and appears as the best feature to track. We thus develop a view-invariant head tracker to estimate X, Y and θ.

For each frame, the nearest training view is selected and the homography that relates the image points to the corresponding training plane is considered. Supposing the person walks on a planar ground in a structured man-made environment, this homography can be computed using the dominant 3D directions of the scene in both training and input images. The projection of the input image onto the corresponding training image plane then compensates for the effect of both discretization along θ and variations along φ. It also removes part of the perspective effect.

Once the input image has been warped, the pose can then be estimated employing the view-based models corresponding to the selected training view from the framework introduced in the first Part of this thesis. The resulting silhouette and 2D pose can then be back-projected onto the original input image plane.

The rest of the chapter is organized as follows. First, some geometrical considerations are explained in Section 4.2. The computation of the projective transformation is then described in Section 4.3 while the tracking framework is depicted in Section 4.4. Experiments and quantitative evaluation are presented in Section 4.5 and conclusions are drawn in Section 4.6.

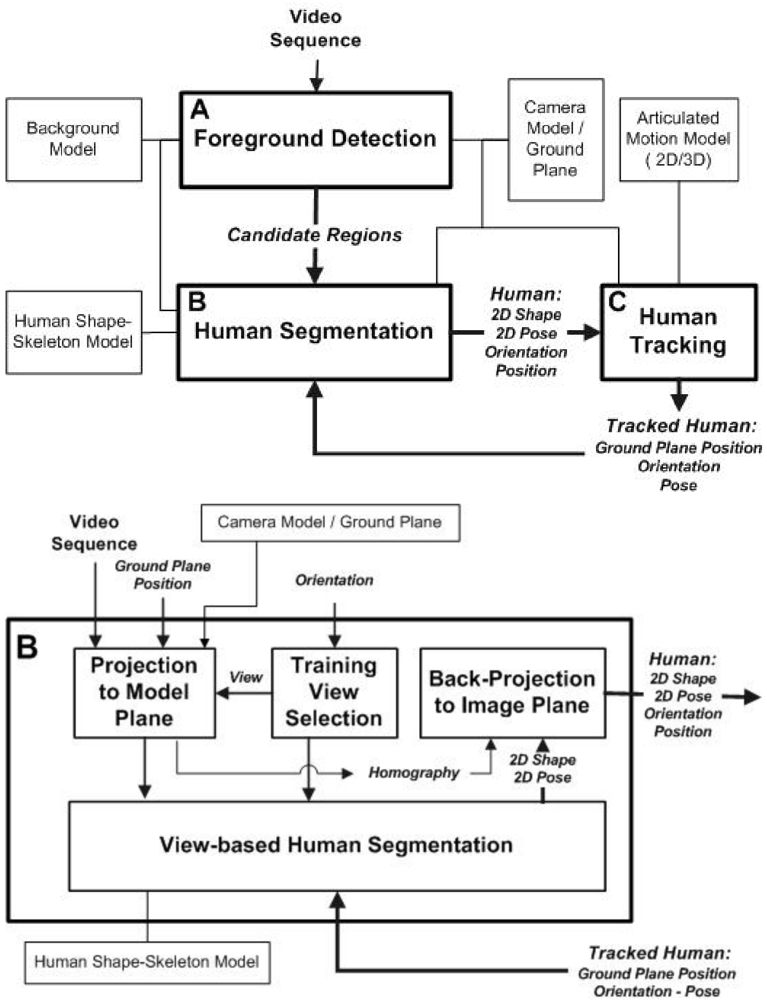

Video Sequence
Background Model
A
Foreground Detection
Camera Model / Ground Plane
Articulated Motion Model (2D/3D)
Candidate Regions
Human Shape-Skeleton Model
B
Human Segmentation
Human: 2D Shape 2D Pose Orientation Position
C
Human Tracking
Tracked Human: Ground Plane Position Orientation Pose
Video Sequence
Camera Model / Ground Plane
Ground Plane Position
Orientation
B
Projection to Model Plane
View
Training View Selection
Back-Projection to Image Plane
Human: 2D Shape 2D Pose Orientation Position
Homography
2D Shape 2D Pose
View-based Human Segmentation
Human Shape-Skeleton Model
Tracked Human: Ground Plane Position Orientation - Pose

Figure 4.3: (up) The proposed tracking system diagram comprises 3 main blocks: (A) Detection, (B) Pose Estimation&Segmentation and (C) Tracking. The main contribution we make in this chapter appears in the Segmentation-Pose Estimation block detailed below.

4.2 Geometrical Considerations in Man-Made Environments

We propose to exploit camera and scene knowledge when working in a man-made environments which is the case of most video-surveillance scenarios.

4.2.1 Notations

In the following sections upper case letters, e.g. $\mathbf{X}$ or X, will be used to indicate quantities in space whereas image quantities will be indicated with lower case letters, e.g. $\mathbf{x}$ or x. Euclidean vectors are denoted with upright boldface letters, e.g $\mathbf{x}$ or $\mathbf{X}$, while slanted letters denote their cartesian coordinates, e.g. (x, y, z) or (X, Y).

Following notations used in [Sola et al., 2012], underlined fonts $\underline{\bullet}$ indicate homogeneous coordinates in projective spaces, e.g $\underline{\mathbf{x}}$ or $\underline{\mathbf{X}}$. A homogeneous point $\underline{\mathbf{x}} \in \mathbb{P}^n$ is composed of a vector $\mathbf{m} \in \mathbb{R}^n$ and a scalar ρ (usually referred to as the homogeneous part):

$$\underline{\mathbf{x}} = \begin{bmatrix} \mathbf{m} \\ \rho \end{bmatrix} \in \mathbb{P}^n \subset \mathbb{R}^{n+1}, \tag{4.1}$$

where the choice $\rho = 1$ is the original Euclidean point representation while $\rho = 0$ defines the points at infinity. The homogeneous point $\underline{\mathbf{x}}$ thus refers to the Euclidean point $\mathbf{x} \in \mathbb{R}^n$:

$$\mathbf{x} = \mathbf{m}/\rho. \tag{4.2}$$

By definition, all the homogeneous points $\{[\rho\mathbf{x}^T, \rho]^T\}_{\rho\in\mathbb{R}^*}$ represents the same Euclidean point $\mathbf{x}$ (see [Hartley and Zisserman, 2004]) and, for homogeneous coordinates, "=" means an assignment or an equivalence up to a non-zero scale factor.

4.2.2 Camera and Scene Calibration

Supposing observed humans are walking on a planar ground floor with a vertical posture, camera model and ground plane assumptions provide useful geometric constraints that help reducing the search space as in [Lin and Davis, 2010, Zhao et al., 2008, Zhao and Nevatia, 2004], instead of searching for all scales, all orientations and all positions. During the scene calibration two 3×3 homography matrices are calculated: $\mathbf{H}_{\mathrm{g}}$ which characterizes the mapping between the ground plane in the image and the real world ground plane Π_{gd} and $\mathbf{H}_{\mathrm{h}}$ relating the head plane in the image with Π_{gd}. In this work, the homography matrices are estimated by the least-squares method using four or more pairs of manually preannotated points in several frames. The 2 homography mappings are illustrated in Fig. 4.4. Note that when surveillance cameras with a high field of view are used (as with [Caviar, 2004]), a previous lens calibration is required to correct the optical distortion.

Given an estimate of the subject's location (X, Y) on the world ground plane Π_{gd}, the planar homographies $\mathbf{H}_{\mathrm{g}}$ and $\mathbf{H}_{\mathrm{h}}$ are used to evaluate the location of the subject's head $\mathbf{x}_H$ and "feet" $\mathbf{x}_F$ in the image I:

$$\underline{\mathbf{x}}_H = \mathbf{H}_{\mathrm{h}} \cdot [X, Y, 1]^T, \tag{4.3}$$

$$\underline{\mathbf{x}}_F = \mathbf{H}_{\mathrm{g}} \cdot [X, Y, 1]^T, \tag{4.4}$$

where points in the projective space $\mathbb{P}^2$ are expressed in homogeneous coordinates.

In this work, we want to compensate for the difference of camera view between input and training images using the dominant 3D directions of the scenes. We suppose that the camera model is known and people walk in a structured man-made environment where straight lines and planar walls are plentiful. The transformation matrices introduced in the next section are calculated online using the vanishing points [1] evaluated in an off-line stage: the positions of the vertical vanishing point $\mathbf{v}_z$ and $\mathbf{l}$, the vanishing line of the ground plane, are directly obtained after a manual annotation of the parallel lines (on the ground and walls) in the image. An example of vertical vanishing point localization is given in Fig. 4.4.

This method makes sense only for man-made environments because of the presence of numerous easy-to-detect straight lines. Previous work for vanishing points detection [Lutton et al., 1994] could be used to automate the process.

Once we have calibrated the camera in the scene, the camera cannot be moved, which is a limitation of the proposal. In practice, the orientation of the camera could change, for example,

[1] A vanishing point is the intersection of the projections in the image of a set of parallel world lines. Any set of parallel lines on a plane define a vanishing point and the union of all these vanishing points is the vanishing line of that plane [Criminisi et al., 2000].

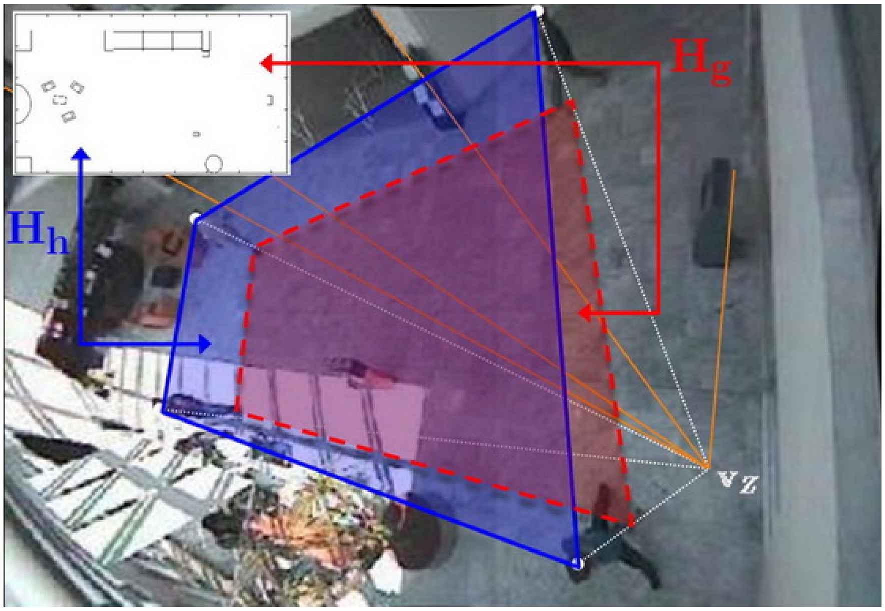


Figure 4.4: Camera and Scene Calibration: 2 homography matrices are calculated from manual annotations: $\mathbf{H_g}$ characterizing the mapping between the ground plane in the image (*red dashed line*) and the real world ground plane Π_{gd} (*upper left*) and $\mathbf{H_h}$ relating the head plane in the image (*blue solid line*) with Π_{gd}. The vertical vanishing point $\mathbf{v}_Z$ and the horizontal vanishing line are also computed using the straight lines from walls and floor observed in the scene.

due to the lack of stability of the camera support. A little change in orientation has a great influence in the image coordinates, and therefore invalidates previous calibration. However, if the camera is not changed in position, or position change is small with respect to the depth of the observed scene, the homography can easily be re-calibrated automatically. An automatic method to compute homographies and line matching between image pairs like the one presented in [Guerrero and Sagüés, 2003] can then be used. At the moment, however, this has not been included in our system..

4.3 Projection Image-Training View Through a Vertical Plane

As demonstrated in [Kale et al., 2003], for objects far enough from the camera, we can approximate the actual 3D object as being represented by a planar object. In other words, a person can be approximated by a planar object if he or she is far enough from the camera [2]. As shown in [Riklin-Raviv et al., 2007], in the presence of perspective distortion neither similarity nor affine model provide reasonable approximation for the transformation between a prior shape and a shape to segment. Riklin-Raviv et al. [2007] demonstrate that a planar projective transformation is a better approximation even though the object shape contour is roughly planar. Following these two observations, we propose to find a projective transformation, i.e. a homography, between training and testing camera views to compensate for the effect of both discretization along θ and variations along φ, thus alleviating the effect of perspective distortion

[2] This hypothesis is obviously not strictly true as it does not depend solely on the distance to the camera but also on the pose and orientation of the person w.r.t. the camera

on silhouette-based human motion analysis.

The 3×3 transformation $\mathbf{P}_{I_2 \Pi I_1}$ between two images I_1 and I_2 through a vertical plane Π observed in both images can be obtained as the product of 2 homographies defined up to a rotational ambiguity. The first one, $\mathbf{H}_{\Pi \leftarrow I_1}$, projects the 2D image points in I_1 to the vertical plane Π and the other one, $\mathbf{H}_{I_2 \leftarrow \Pi}$, relates this vertical plane to the image I_2. We thus obtain the following equation that relates the points $\mathbf{x}_1$ from I_1 with image points $\mathbf{x}_2$ from I_2:

$$\underline{\mathbf{x}}_2 = \mathbf{P}_{I_2 \Pi I_1} \cdot \underline{\mathbf{x}}_1, \tag{4.5}$$

where $\underline{\mathbf{x}}_1, \underline{\mathbf{x}}_2 \in \mathbb{P}^2$ and with:

$$\begin{aligned} \mathbf{P}_{I_2 \Pi I_1} &= \mathbf{H}_{I_2 \leftarrow \Pi} \cdot \mathbf{H}_{\Pi \leftarrow I_1} \\ &= \mathbf{H}_{I_2 \leftarrow \Pi} \cdot (\mathbf{H}_{I_1 \leftarrow \Pi})^{-1}. \end{aligned} \tag{4.6}$$

The two homographies $\mathbf{H}_{I_1 \leftarrow \Pi}$ and $\mathbf{H}_{I_2 \leftarrow \Pi}$ can be computed from the vanishing points of the 3D directions spanning the vertical plane Π i.e. the vertical Z-axis and a reference horizontal line $\mathbf{G} = \Pi \wedge \Pi_{\text{gd}}$, intersection of Π and ground plane Π_{gd}:

$$\mathbf{H}_{I \leftarrow \Pi} = [\underline{\mathbf{v}}_G \quad \alpha \underline{\mathbf{v}}_Z \quad \mathbf{o}], \tag{4.7}$$

where $\mathbf{v}_G$ and $\mathbf{v}_Z$ are the vanishing point along the horizontal and vertical axis in I, $\mathbf{o}$ is the origin of the coordinate system and α is a scale factor (see Appendix A).

In the same way, we now want to relate 2 images, e.g. training and testing images, observing two different calibrated scenes with 2 different subjects performing the same action from two similar viewing angles. These images can potentially be related through a vertical plane centered in the human body following Eq. 4.5 . The problem is to select the vertical plane that will optimize the 2D shape correspondence between the 2 images. We choose to select this vertical plane in the training image, where the azimuth angle θ is known and the camera is in an approximately horizontal position (i.e. elevation angle $\varphi \approx 0$), and consider the closest vertical plane centered on the human body: if a camera view Φ is defined by its azimuth and elevation angles (θ, φ) on the viewing hemisphere (Fig. 4.2a), the closest vertical plane Π is the plane defined as $(\theta, 0)$.

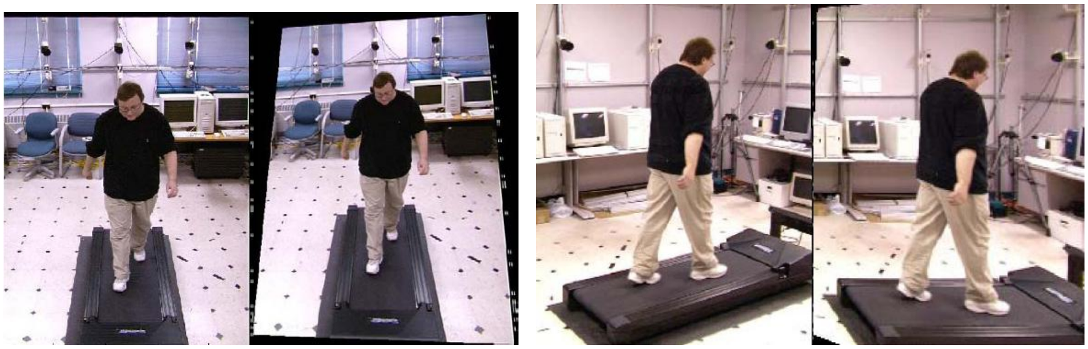

(a) (b)

Figure 4.5: Projection on the vertical plane: examples of original and warped images resulting from applying the homography $\mathbf{H}_{\Pi_v \leftarrow \Phi_v}$ for frontal (a) and "rear-diagonal" (b) views of the Mobo dataset.

Thus, considering a set of training views $\{\Phi_v\}_{v=1}^{N_v}$, the associated homographies $\{\mathbf{H}_{\Phi_v \leftarrow \Pi_v}\}_{v=1}^{N_v}$ relating each view and its closest vertical plane Π_v centered on the human body are computed during the off-line stage (following Eq. 4.7) and stored for online use[3]. Each vertical plane Π_v is spanned by the vertical Z-axis and a reference horizontal vector $\mathbf{G}_v \in (\Pi_v \wedge \Pi_{\mathrm{gd}})$. Examples of projection on a vertical plane are given for 2 of the 8 Mobo training views in Fig. 4.5. The perspective distortion, particularly severe in the front view (large head and short legs), is corrected: the image appears distorted but the global figure recovers real morphological proportions in the front view (Fig. 4.5a) while we can observe how the transformation tends to place the feet at the same vertical position and remove the perspective effect for the rear-diagonal (Fig. 4.5b) view.

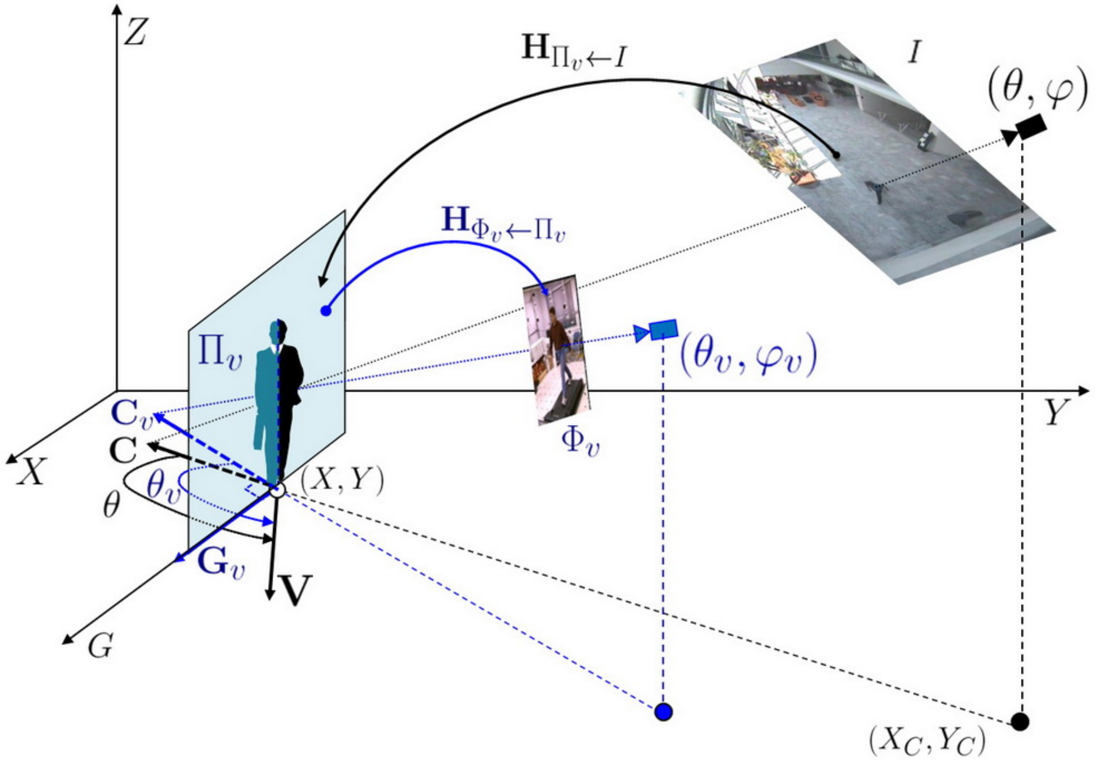


Figure 4.6: Schematical representation of the transformation between 2 images through a vertical plane: testing image I and training image plane Φ_v can be related through a vertical plane Π_v. The transformation $\mathbf{P}_{\Phi_v \Pi_v I}$ is obtained as the product of $\mathbf{H}_{\Phi_v \leftarrow \Pi_v}$ and $\mathbf{H}_{\Pi_v \leftarrow I}$ while the inverse projection $\mathbf{P}_{I \Pi_v \Phi_v}$ can be obtained as the product of $\mathbf{H}_{I \leftarrow \Pi_v} = (\mathbf{H}_{\Pi_v \leftarrow I})^{-1}$ and $\mathbf{H}_{\Pi_v \leftarrow \Phi_v} = (\mathbf{H}_{\Phi_v \leftarrow \Pi_v})^{-1}$.

Given a testing image I with an observed human at location (X, Y) on the ground plane Π_{gd}, the azimuth $\theta \in [-\pi, \pi]$ (i.e. camera viewpoint or the subject's orientation w.r.t. the camera) is defined on the ground as:

$$\theta = \widehat{\mathbf{CV}}, \tag{4.8}$$

where vectors $\mathbf{C}$ and $\mathbf{V} \in \mathbb{R}^2$ are the projections on the ground plane Π_{gd} of the camera viewing direction and the orientation vector respectively[4]. The viewing direction is defined as the line connecting the subject and camera (originating from the camera center) and the orientation

[3] The training views considered in this work are not exactly frontal explaining why $\mathbf{H}_{\Pi_v \leftarrow \Phi_v}$ are taken into account.

[4] The angle θ is π when the subject is facing the camera and θ is 0 when facing away.

direction is a vector perpendicular to the shoulder line of the subject pointing in the direction he or she is facing (see Fig. 4.6). Note that $\mathbf{C}$ is easily evaluated as:

$$\mathbf{C} = \begin{bmatrix} X - X_C \\ Y - Y_C \end{bmatrix}, \tag{4.9}$$

where (X_C, Y_C) is the projection on the ground plane of the camera center [5]. The direction of $\mathbf{V}$ can be found by rotating $\mathbf{C}$ around the Z-axis if θ is known:

$$\mathbf{V} \propto \mathbf{R}(\theta) \cdot \mathbf{C}, \tag{4.10}$$

where $\mathbf{R}(.)$ denotes a 2×2 rotation matrix.

Table 4.1: Azimuth $\theta_v = \widehat{\mathbf{C}_v\mathbf{V}}$ and $\widehat{\mathbf{VG}_v}$ angle defining the vertical plane Π_v for the 8 training viewpoints of the MoBo dataset (Fig. 4.2): lateral (L_1 & L_2), diagonal (D_1 & D_2), rear-diagonal (RD_1 & RD_2), front (F) and back (B) views.

	View							
	RD_1	L_1	D_1	F	D_2	L_2	RD_2	B
θ_v	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$\frac{3\pi}{4}$	π	$-\frac{3\pi}{4}$	$-\frac{\pi}{2}$	$-\frac{\pi}{4}$	0
$\widehat{\mathbf{VG}_v}$	$\frac{\pi}{4}$	0	$-\frac{\pi}{4}$	$-\frac{\pi}{2}$	$-\frac{3\pi}{4}$	π	$\frac{3\pi}{4}$	$\frac{\pi}{2}$

Given $\{\theta_v = \widehat{\mathbf{C}_v\mathbf{V}}\}_{v=1}^{N_v}$ the N_v training values for θ (cf Tab. 4.1) and given an estimation of θ for the observed subject, a training view Φ_v is selected so that:

$$v = \arg \min_{v \in \{1, N_v\}} |\theta - \theta_v|. \tag{4.11}$$

The transformation $\mathbf{P}_{\Phi_v \Pi_v I}$ (illustrated in Fig. 4.6) between input image I and Φ_v through the vertical plane Π_v can then potentially be obtained as the product:

$$\mathbf{P}_{\Phi_v \Pi_v I} = \mathbf{H}_{\Phi_v \leftarrow \Pi_v} \cdot \mathbf{H}_{\Pi_v \leftarrow I}, \tag{4.12}$$

up to a rotational ambiguity. The problem now consists of finding the plane Π_v in the image I, i.e. the vanishing points of the 3D directions, and compute $\mathbf{H}_{\Pi_v \leftarrow I} = (\mathbf{H}_{I \leftarrow \Pi_v})^{-1}$ from Eq. 4.7. The plane Π_v is spanned by the vertical Z-axis and a horizontal axis $\mathbf{G} = \mathbf{G}_v$ which can be found in the real 3D world by rotating $\mathbf{V}$ about the Z-axis:

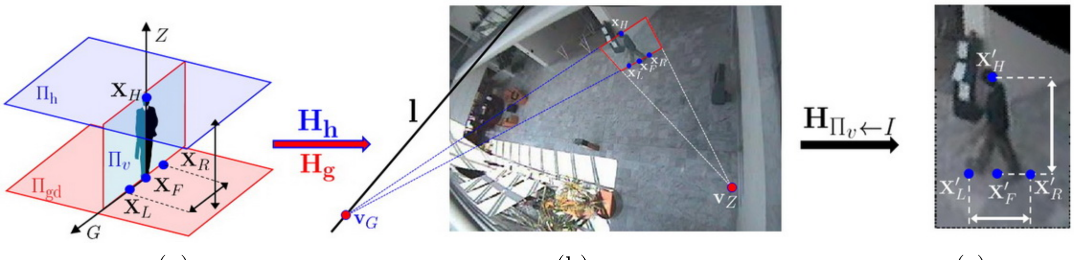
$$\mathbf{G} \propto \mathbf{R}(\widehat{\mathbf{VG}_v}) \cdot \mathbf{V}. \tag{4.13}$$

The training values for $\widehat{\mathbf{VG}_v}$ are given in Tab. 4.1. Two real world 3D points $\mathbf{X}_L$, $\mathbf{X}_R$ are then selected on the ground floor along the G-axis at each side of the subject (see Fig. 4.7a). In practice, we select 2 points at 50 cm from the subject. $\mathbf{X}_L$ and $\mathbf{X}_R$ are then reprojected in the image I obtaining $\underline{\mathbf{x}}_L = \mathbf{H}_{\mathrm{g}}\underline{\mathbf{X}}_L$ and $\underline{\mathbf{x}}_R = \mathbf{H}_{\mathrm{g}}\underline{\mathbf{X}}_R$, where $\underline{\mathbf{X}}_L, \underline{\mathbf{X}}_R \in \mathbb{P}^2$ are expressed in the ground plane coordinates. These two image points can be used to localize the vanishing point $\mathbf{v}_{\scriptscriptstyle G}$ along real-world G-axis in the image (Fig. 4.7b) as follows:

$$\underline{\mathbf{v}}_{\scriptscriptstyle G} = (\underline{\mathbf{x}}_L \times \underline{\mathbf{x}}_R) \times \mathbf{l}, \tag{4.14}$$

where × represents the vector product, and $\mathbf{l} \in \mathbb{P}^2$ is the vanishing line of the ground plane (see [Hartley and Zisserman, 2004] for details).

[5] As indicated in [Hartley and Zisserman, 2004], the vanishing point is the image of the vertical "footprint" of the camera centre on the ground plane, i.e. : $\underline{\mathbf{X}}_C = (\mathbf{H}_{\mathrm{g}})^{-1} \cdot \underline{\mathbf{v}}_Z$ with $\mathbf{X}_C = (X_C, Y_C)$.



(a) (b) (c)

Figure 4.7: Projection to Vertical Plane. Four real world coplanar points are selected on Π_v : $\mathbf{X}_L$, $\mathbf{X}_R$, $\mathbf{X}_F$ on the ground plane Π_{gd} along the G-axis and $\mathbf{X}_H$ the center of the head (a). The four points are reprojected in the image I obtaining $\underline{\mathbf{x}}_L = \mathbf{H}_g\underline{\mathbf{X}}_L$, $\underline{\mathbf{x}}_R = \mathbf{H}_g\underline{\mathbf{X}}_R$, $\underline{\mathbf{x}}_F = \mathbf{H}_g\underline{\mathbf{X}}_F$ and $\underline{\mathbf{x}}_H = \mathbf{H}_h\underline{\mathbf{X}}_H$ (b). The image points along the G-axis are then used to localize the vanishing point $\mathbf{v}_G$. The homography $\mathbf{H}_{\Pi_v \leftarrow I}$ relating the input image with the selected vertical plane is then obtained from vanishing points $\mathbf{v}_Z$ and $\mathbf{v}_G$ following Eq. 4.7. The scale factor α in $\mathbf{H}_{\Pi_v \leftarrow I}$ is then computed so that the height to width ratio stays constant between the set of reprojected points $\{\mathbf{X}'_L, \mathbf{X}'_R, \mathbf{X}'_F, \mathbf{X}'_H\}$ in (c) and the original real-world points $\{\mathbf{X}_L, \mathbf{X}_R, \mathbf{X}_F, \mathbf{X}_H\}$.

The computation of $\mathbf{H}_{\Pi_v \leftarrow I}$ relating the input image with the selected vertical plane is then obtained following Eq. 4.7. The scale factor α in Eq. 4.7 is evaluated using four known coplanar points[6] in the real-world vertical plane Π_v: $\mathbf{X}_L$, $\mathbf{X}_R$ (from above), the subject's ground floor location $\mathbf{X}_F$ and $\mathbf{X}_H$, the center of the subject's head, i.e. the vertical projection on the head plane Π_h of the ground floor location (see Fig. 4.7a). The images $\underline{\mathbf{x}}_L = \mathbf{H}_g\underline{\mathbf{X}}_L$, $\underline{\mathbf{x}}_R = \mathbf{H}_g\underline{\mathbf{X}}_R$, $\underline{\mathbf{x}}_F = \mathbf{H}_g\underline{\mathbf{X}}_F$ and $\underline{\mathbf{x}}_H = \mathbf{H}_h\underline{\mathbf{X}}_H$ of these four points in I (Fig. 4.7b) are reprojected in the plane Π_v using $\mathbf{H}_{\Pi_v \leftarrow I}$ obtaining $\mathbf{X}'_L$, $\mathbf{X}'_R$, $\mathbf{X}'_F$ and $\mathbf{X}'_H \in \mathbb{R}^2$ (Fig. 4.7c). The scale factor α in $\mathbf{H}_{\Pi_v \leftarrow I}$ is then computed so that the height to width ratio stays constant between the set of reprojected points $\{\mathbf{X}'_L, \mathbf{X}'_R, \mathbf{X}'_F, \mathbf{X}'_H\}$ and the original real-world points $\{\mathbf{X}_L, \mathbf{X}_R, \mathbf{X}_F, \mathbf{X}_H\}$, i.e.:

$$\frac{||\mathbf{X}_H - \mathbf{X}_F||}{||\mathbf{X}_R - \mathbf{X}_L||} = \frac{||\mathbf{X}'_H - \mathbf{X}'_F||}{||\mathbf{X}'_R - \mathbf{X}'_L||} = \frac{||h_\alpha(\mathbf{x}'_H) - h_\alpha(\mathbf{x}'_F)||}{||h_\alpha(\mathbf{x}'_R) - h_\alpha(\mathbf{x}'_L)||}, \tag{4.15}$$

where, for ease of notation, we define the one-to-one mapping function $h_\alpha : \mathbb{R}^2 \mapsto \mathbb{R}^2$ which transforms image points to plane Π_v using the homography $\mathbf{H}_{\Pi_v \leftarrow I}$: $\mathbf{X} = h_\alpha(\mathbf{x}) \Leftrightarrow \underline{\mathbf{X}} = \mathbf{H}_{\Pi_v \leftarrow I} \cdot \underline{\mathbf{x}}$. In our case, we assume the head is at 170 cm from the floor (average human height) and select $\mathbf{X}_L$ and $\mathbf{X}_R$ to be 100 cm apart, we thus have to find α which minimizes:

$$E(\alpha) \triangleq \left| 1.7 - \frac{||h_\alpha(\mathbf{x}'_H) - h_\alpha(\mathbf{x}'_F)||}{||h_\alpha(\mathbf{x}'_R) - h_\alpha(\mathbf{x}'_L)||} \right|, \tag{4.16}$$

i.e. a convex optimization problem which is easily solved by gradient descent search.

Finally, once $\mathbf{H}_{\Pi_v \leftarrow I}$ has been calculated, $\mathbf{P}_{\Phi_v \Pi_v I}$ can be computed using Eq. 4.12. The rotational ambiguity in choosing the coordinate system is resolved using the same four points and checking that the vectors $\mathbf{U}, \mathbf{U}' \in \mathbb{R}^3$ resulting from the two cross products $\mathbf{U} = \langle \mathbf{X}_L \mathbf{X}_R \rangle \times \langle \mathbf{X}_F \mathbf{X}_H \rangle$ and $\mathbf{U}' = \langle \mathbf{X}'_L \mathbf{X}'_R \rangle \times \langle \mathbf{X}'_F \mathbf{X}'_H \rangle$ point in the same direction, otherwise the G-axis is flipped in matrix $\mathbf{H}_{\Pi_v \leftarrow I}$. Eq. 4.12 becomes:

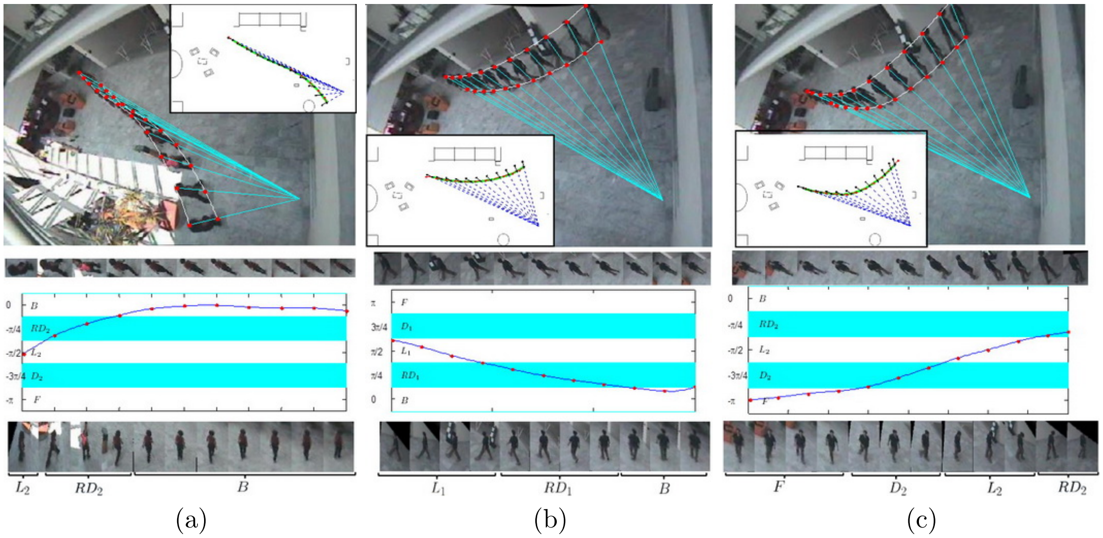
$$\mathbf{P}_{\Phi_v \Pi_v I} = \begin{cases} \mathbf{H}_{\Phi_v \leftarrow \Pi_v} \cdot \mathbf{H}_{\Pi_v \leftarrow I} & \text{if } \mathbf{U} \cdot \mathbf{U}' \geq 0, \\ \mathbf{H}_{\Phi_v \leftarrow \Pi_v} \cdot \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \mathbf{H}_{\Pi_v \leftarrow I} & \text{otherwise.} \end{cases} \tag{4.17}$$

[6] Note that even if four points have been considered in our implementation, three points would be sufficient.

```
Algorithm 3: Projection Image to Training View.
  input  : Triplet $(X, Y, \theta)$.
  output: Projective Transformation $\mathbf{P}_{\Phi_v \Pi_v I}$.
    • Select the training view $\Phi_v$ (cf. Eq. 4.11);
    • Compute camera viewing direction $\mathbf{C}$ (cf. Eq. 4.9) ;
    • Find orientation vector $\mathbf{V}$ (cf. Eq. 4.10);
    • Find the real-world $G$-axis defining $\Pi_\text{v}$ (cf. Eq. 4.13);
    • Localize the vanishing point $\mathbf{v}_G$ using Eq. 4.14;
    • Calculate $\mathbf{H}_{\Pi_v \leftarrow I} = (\mathbf{H}_{I \leftarrow \Pi_v})^{-1}$ using Eq. 4.7;
    • Compute the scale factor $\alpha$ (cf. Eq. 4.16);
    • Calculate $\mathbf{P}_{\Phi_v \Pi_v I}$ using Eq. 4.17;
```

The entire process leading to the computation of the projective transformation $\mathbf{P}_{\Phi_v \Pi_v I}$ is summarized in Alg. 3.

4.3.1 Qualitative Results using Ground Truth Data



(a) (b) (c)

Figure 4.8: Examples of projections to training planes for *Walk1* (a) and *Walk3* (b and c) sequences [Caviar, 2004]. The homographies are computed using "ground truth" locations (X, Y) and viewpoints θ which are estimated from the manual labelling of head location in consecutive frames, the angle θ being estimated from the direction of motion. For each sequence, we show (*from top to bottom*): head and feet trajectories in the image I and corresponding trajectory (X, Y) on the floor with vectors $\mathbf{C}$ and $\mathbf{V}$, the regions of interest, the viewpoint θ and selected training view Φ_v considered to compute $\mathbf{P}_{\Phi_v \Pi_v I}$, and finally the warped images $I_{\theta,X,Y}$ for frames $1, 20, \ldots 160, 200$ in (a), frames $1, 15, \ldots, 150$ in (b) and frames $1, 15, \ldots, 150, 160$ in (c). See dataset details in Tab. 5.1.

First, we conduct a qualitative evaluation of the proposed projective transformation employing manually labelled head locations in several sequences to generate ground truth data for triplets (X, Y, θ).

A series of gait sequences have thus been selected from the Caviar project database [Caviar, 2004]: in these sequences people are walking in various directions and the changing perspective effect can be observed. For each sequence, the trajectory $\{X_t, Y_t\}_{t=1}^{N_t}$ on the ground floor is directly recovered from the manual labelling using $\mathbf{H}_\mathrm{h}$ which relates the head plane in the image with the ground plane Π_gd. Supposing that the subject is facing in the direction of motion, we estimate the direction $\mathbf{V}_t$ and consequently the viewpoint angle θ_t at time t from the trajectory $\{X_t, Y_t\}_{t=1}^{N_t}$:

$$\theta_t = \arccos(\frac{\mathbf{C}_t \cdot \mathbf{V}_t}{||\mathbf{C}_t|| \cdot ||\mathbf{V}_t||}), \tag{4.18}$$

with $\mathbf{V}_t = [X_t - X_{t-1}, Y_t - Y_{t-1}]^T$ and $\mathbf{C}_t$ from Eq. 4.9. Projections on training plane obtained using the resulting data $\{(X_t, Y_t, \theta_t)\}_{t=1}^{N_t}$ are given in Fig. 4.8. For each presented sequence, we show (from top to bottom) the trajectory in the image and its projection on the real-world ground plane $\{X_t, Y_t\}_{t=1}^{N_t}$, the extracted subimages, the viewpoints $\{\theta_t\}_{t=1}^{N_t}$ with corresponding training views and, finally, the transformed sub-images $I_{\theta,X,Y}$ for several selected frames. We can observe the smoothness of the different trajectories and how the viewpoint θ slowly changes along the sequences. The regions of interest around the subjects are normalized and projected onto the adequate model plane and the perspective distortion seems corrected.

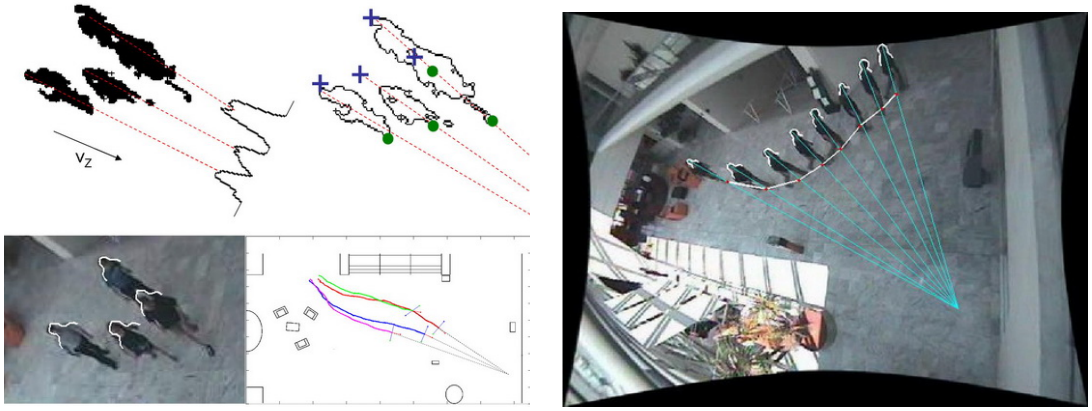
4.4 View-invariant Pose Analysis

The resulting warped images can be processed to estimate a pose using the silhouette-based model framework presented in the previous part of this thesis. In this section, the viewpoint θ is also estimated from the trajectory on the floor but, to ensure an automatic and reliable estimation of ground plane position, we employ a head-tracker based on Kalman filter to estimate the head location in consecutive frames.

A simple but effective way of locating people in an image relies on detecting their head. First of all, the head is the easiest human feature to detect because of the low variability of its shape and its top position in the body. Moreover, in a sequence taken by a rooftop camera, the head is the most visible feature since it is less likely to be occluded. Finally, in a calibrated environment, a good estimation of the ground plane position (X, Y) can be obtained by projecting vertically the head position.

Many papers propose computing the vertical histogram of the foreground blob and scanning it, searching for peaks as possible head candidates [Siebel and Maybank, 2002a, Zhao and Nevatia, 2004]. The problem with this method is that it cannot detect heads in the interior of the blob as shown in Fig. 4.9.a-1. In [Zhao and Nevatia, 2002], the authors extend this head candidates search by using a head-shoulder model. Following a similar approach, we train such a model by considering only the upper landmarks of our training shapes and learn a mixture of linear head shape models.

When given a selected blob (filtered w.r.t its size, position and area), we compute the possible head candidates by searching for local peaks (local maxima) in the direction towards the vertical vanishing point $\mathbf{v}_Z$. We also compute the feet candidates (local minima) and the corresponding probable head location (see Fig. 4.9.a-2). The head shape model is then applied to all the selected head candidates and the confidence weight of each hypothesis is evaluated by edge matching error. An example is given in Fig. 4.9.a-3. In this way, non-human blobs resulting from shadows and reflections are dismissed.



(a)

(b)

Figure 4.9: (a) View invariant head detector: An example with multiple pedestrians from *MeetCrowd* sequence [Caviar, 2004] is proposed *(clockwise from upper left)*: (1) vertical histogram of the foreground blob, (2) head (crosses) and feet (dots) candidates computed using distance to the vertical vanishing point, (3) detected heads and (4) corresponding trajectories on the ground floor. (b) View invariant tracking based on head detector: Ground Plane trajectory of *Walk3* sequence [Caviar, 2004] extracted using the head-based tracker.

Algorithm 4: Human tracking based on head detector

```
1. $(\widehat{X}, \widehat{Y})$ is predicted by the Kalman filter.
2. $\widehat{\underline{\mathbf{x}}_H} = \mathbf{H}_\mathrm{h} \cdot [\widehat{X}, \widehat{Y}, 1]^T$.
3. Head shape model fitted around $\widehat{\mathbf{x}_H}$ obtaining $\mathbf{x}'_H$ .
4. Filter parameters updated using $[X, Y, 1]^T = {\mathbf{H}_\mathrm{h}}^{-1} \cdot \underline{\mathbf{x}}'_H$ and $\theta$ evaluation from the
   ground plane trajectory following Eq. 4.18.
5. $\theta$, $X$ and $Y$ sent to the diagram block B in Fig. 4.3.
```

The system is initialized in the first frames, estimating $\mathbf{x}_H$ by a rough fitting of our model as in [Zhao and Nevatia, 2002]. A tracking is then applied, the state of the each pedestrian (ground plane position (X, Y)) being estimated at each time step using a Kalman filter as in [Zhao and Nevatia, 2004]. The tracking process is detailed in Algorithm 4. Figure 4.9.b shows an example of head-based tracking for an entire sequence. Our view-invariant head tracker has shown to be robust, even with difficult cases such as people moving in groups and partial occlusions.

Ground plane location (X, Y) and viewpoint θ are then estimated frame to frame, allowing the selection of a training view and the computation of the projective transformation $\mathbf{P}_{\Phi_v \Pi_v I}$. The input image is projected and later processed in the view-based human segmentation diagram block in Fig. 4.3. Two examples are presented in Fig. 4.10 for view RD_1 (up) and F (down): the candidate Region of Interest in the image I_{ROI} (Fig. 4.10a) is warped to the correct model plane obtaining $I_{\theta,X,Y}$ (Fig. 4.10b). The foreground information (Fig. 4.10c) is then used to estimate both 2D shape and 2D pose (Fig. 4.10d) using the view-based shape-skeleton models from the previous chapter. The 2D features can be back-projected to the original image plane (Fig. 4.10e).

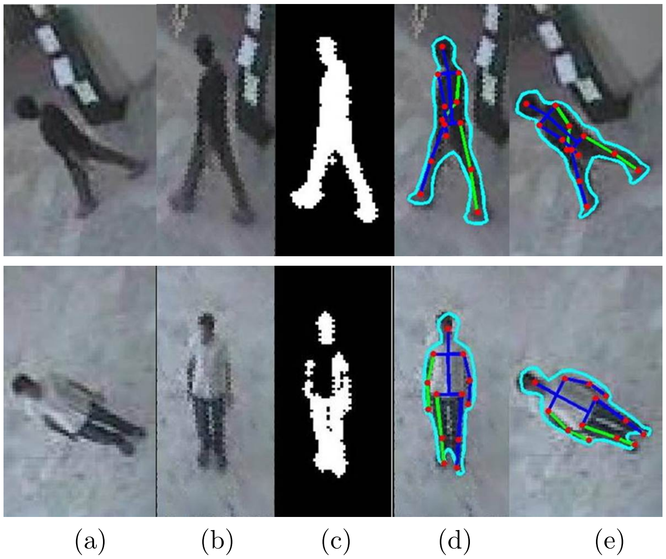


Figure 4.10: Two examples of view-based shape registration and pose estimation are presented for view RD_1 (up) and F (down). In both cases, the original image (a) is warped to the corresponding plane (b). The foreground information (c) is used to apply the view-based pose-silhouette model leading to the estimation of both 2D shape and 2D skeleton in the projected image (d). Shape and 2D pose can then be back-projected to the original image plane (e).

4.5 Experiments

For the evaluation of the framework, the gait sequences from Sec. 4.3.1 are processed. As we expected, since the orientation is estimated from the direction of motion, the system fails with stationary cases.

In Fig.4.11, we present the silhouettes and 2D poses that have been extracted from the sequence presented in the *Walk3*: we can observe how the direction of motion slowly changes along the sequence and how the images are projected on the selected model plane. The resulting shapes are not perfect but, given the complexity of the task (low resolution and high perspective effect), we find them reasonably good. However, while the 2D pose is well estimated during most of the presented sequence (when the viewpoint is lateral or diagonal), we can see that the sequentiality of the motion is lost in the last third of the sequence. As in Sect. 3.5.1.2 where we observed a similar effect with the “WalkiCircle” sequence, this is due to the very low shape variability in the back view where it is very complicated to distinguish a state from another.

We observe that acceptable results are obtained with single walking subjects. However, the reliability of the warping, and consequently the accuracy of the silhouette and pose estimate, seem to strongly depend on the precision with which both ground plane position (X, Y) and orientation θ are estimated.

4.5.1 Numerical Evaluation of the Effect of Noise

To numerically evaluate this dependence, we conduct a series of simulations using a set of testing ground truth poses $\{\mathbf{k}_1^{GT} \cdots \mathbf{k}_{N_{GT}}^{GT}\}$ and a set of sampled training poses $\{\{\mathbf{k}_i^v\}_{i=1}^{N_T}\}_{v=1}^{N_v}$ (i.e. N_T poses for each training view Φ_v). Each pose is made of 13 hand-labelled 2D joints: $\mathbf{k} = [\mathbf{x}_{k_1}, ..., \mathbf{x}_{k_{13}}] \ \in \mathbb{R}^{2\times 13}$. For each tested frame $t \in \{1, N_{GT}\}$, we compute the projective transformation $\mathbf{P}_{I\,\Pi_v \Phi_v}$ using ground truth location (X_t, Y_t) and viewpoint θ_t from above with

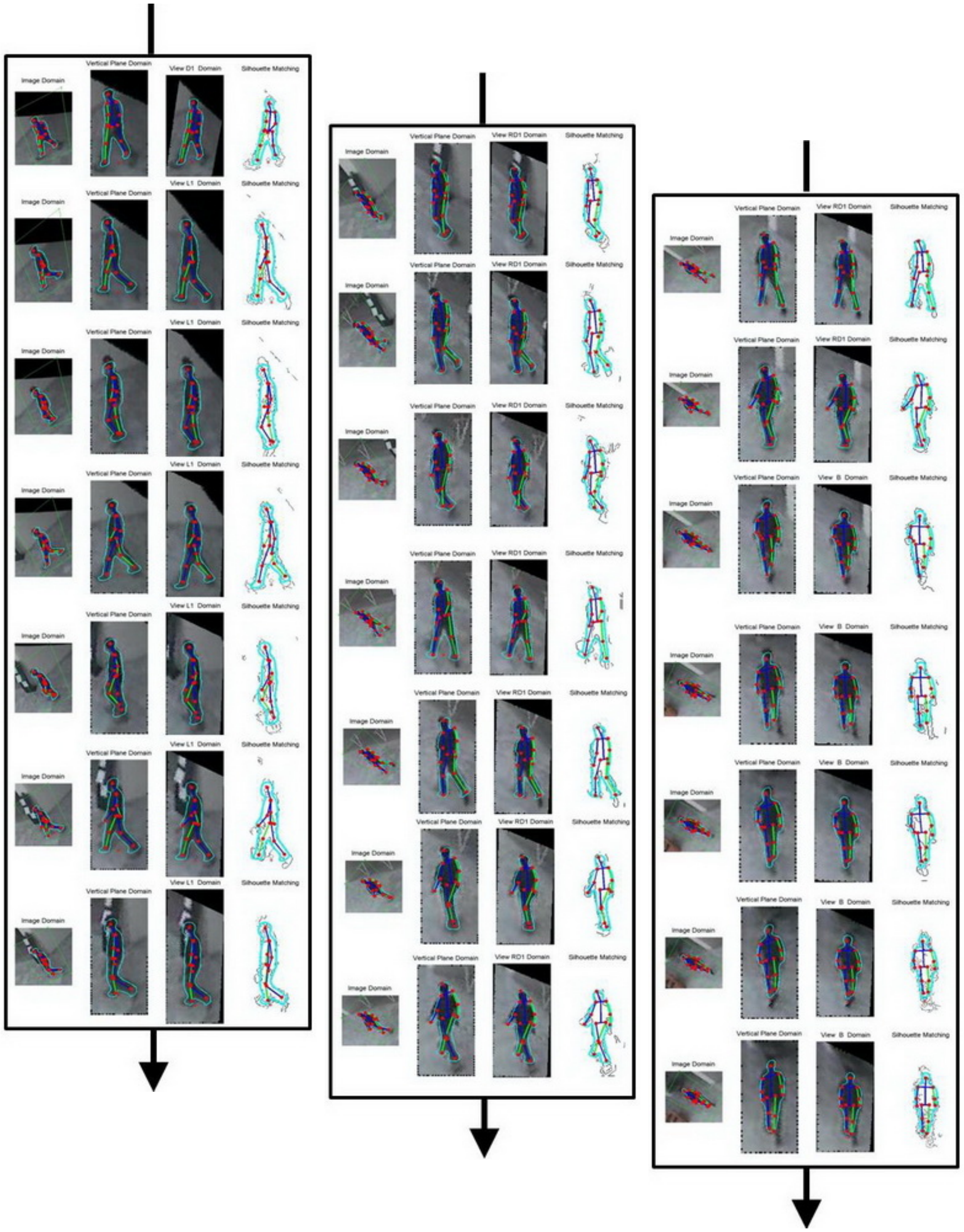


Figure 4.11: Shape and 2D pose estimated using our tracking framework for the *Walk3* sequence. For each presented frame, we show (*from right to left*): foreground image projected on the training plane and processed with our view-based model, image projected on the training plane with estimated shape and 2D pose, image projected on the vertical plane with estimated shape and 2D pose and representation of shape and 2D pose backprojected in the original image.

additive Gaussian white noises (η_{XY} and η_θ of variance σ^2_{XY} and σ^2_θ respectively) and align the N_T training poses $\{\mathbf{k}^v_1 \cdots \mathbf{k}^v_{N_T}\}$ from the selected viewpoint Φ_v, obtaining $\{\mathbf{k}^{Hom}_{1,t} \cdots \mathbf{k}^{Hom}_{N_T,t}\}$ with $\forall i \in \{1, N_T\}$:

$$\underline{\mathbf{x}}^{Hom}_{k_j,i,t} = \mathbf{P}_{I\,\Pi_v\Phi_v} \cdot \underline{\mathbf{x}}_{k_j,i,t},\ \forall j \in \{1,13\}. \tag{4.19}$$

We then compute the average pose error over the testing set taking the closest aligned pose for each frame t:

$$\epsilon^{\text{Hom}} = \frac{1}{N_{GT}} \sum_{t=1}^{N_{GT}} \min_{i\in\{1,N_T\}} d_k(\mathbf{k}^{GT}_t, \mathbf{k}^{Hom}_{i,t}), \tag{4.20}$$

where d_k, defined as:

$$d_k(\mathbf{k}, \mathbf{k}') \triangleq \frac{1}{13} \sum_{j=1}^{13} ||\mathbf{x}_{k_j} - \mathbf{x}'_{k_j}|| \tag{4.21}$$

is the average Root Mean Square Error over the 13 2D-joints (called RMS 2D Pose Error from now on).

We repeat the same operation considering a simple Euclidean 2D similarity transformation $\mathbf{T}$ to align training poses to the tested images and compute:

$$\epsilon^{\text{Sim}} = \frac{1}{N_{GT}} \sum_{t=1}^{N_{GT}} \min_{i\in\{1,N_T\}} d_k(\mathbf{k}^{GT}_t, \mathbf{k}^{Sim}_{i,t}), \tag{4.22}$$

where $\mathbf{k}^{Sim} = [\mathbf{x}^{Sim}_{k_1}, ..., \mathbf{x}^{Sim}_{k_{13}}]\ \in \mathbb{R}^{2\times 13}$ with:

$$\mathbf{x}^{Sim}_{k_j} = \mathbf{T} \cdot \mathbf{x}_{k_j},\ \forall j \in \{1,13\}. \tag{4.23}$$

The similarity is defined as:

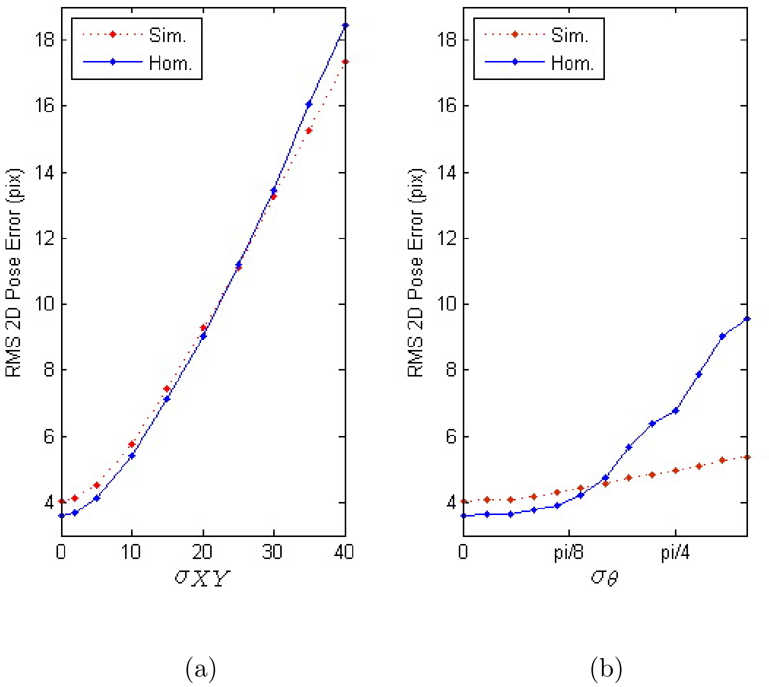
$$\mathbf{T} \cdot \mathbf{x} = \mathbf{u} + s\mathbf{R}$$

$(\gamma)\cdot\mathbf{x},\ \forall\mathbf{x}\ \in \mathbb{R}^2,$ (4.24)in which $(\mathbf{u}, \gamma, s)$ are offset, rotation angle and scaling factor respectively. These parameters are readily calculated using head center $\mathbf{x}_H$ and "feet" location on the ground floor $\mathbf{x}_F$ in training and testing images.

The results obtained when varying σ_{XY} and σ_θ are given in Fig. 4.12. The average pose error almost linearly increases with increasing localization noise η_{XY} for both alignment methods, slightly more for the proposed homographic alignment (Fig. 4.12a). A slight noise in the viewpoint estimation $\sigma_\theta \leq \frac{\pi}{16}$ does not seem to affect any of the 2 alignment methods (Fig. 4.12b). However, while the error with similarity seems to linearly increase with increasing viewpoint noise η_θ for higher noise levels, the effect is much more pronounced for the projective alignment. By augmenting σ_θ, we slowly increase the possibility of picking the wrong view which has more important consequences when a homography is employed between training and testing view planes instead of a simple Euclidean transformation. The benefit of using a homographic alignment rapidly decreases with the amount of added noise in viewpoint estimation $\sigma_\theta \geq \frac{\pi}{8}$ and the error even gets larger than the one obtained with a similarity transformation for $\sigma_\theta \geq \frac{\pi}{6}$.

4.6 Conclusions

The view-invariant approach we have proposed in this chapter can be applied to any type of 2D model or exemplar-based technique. The viewing angle is discretized into a finite number of training viewpoints and a framework of view-based models is constructed. Then, when processing a sequence, the adequate training view is selected by estimating the orientation on



(a) (b)

Figure 4.12: Effect of noise on 2D pose estimation: the average RMS 2D pose error (in pixels) is computed over a set of manually labelled testing poses and a set of training poses aligned using homographic (Hom) and similarity (Sim) alignments. The results are obtained varying the variance of the additive Gaussian white noise which has been added to (a) the ground truth location (X, Y) (in cm) and (b) the viewpoint angle θ (in radians).

the ground plane. The viewpoint correspondence is thus established by projecting the input image onto this training plane and finally the selected view-based model is employed for feature extraction in the warped image. To ensure a reliable estimation of both ground plane position and motion direction, indispensable for obtaining the right warping, we have developed a view-invariant head-based tracker.

Acceptable results have been obtained for sequences with a single walking subject but we identified two main drawbacks: 1) the employed model does not handle pose ambiguities and does not recover from drifting and, 2) more importantly, the result greatly depends on the accuracy achieved when estimating both location (X, Y) and orientation θ. We have numerically evaluated the effect of noise on pose estimation. Our framework performs sufficiently well when an accurate estimation of both ground plane location and orientation (i.e. viewpoint) can be made but, with high levels of noise, the effect on pose estimation is much more pronounced for our proposed projective alignment. These results explain why the estimation of the viewpoint θ from the ground plane trajectory is not satisfactory for our purpose. Even if it gives interesting results in case of constant speed motions, this method is not accurate enough and too sensitive to noisy measurements (e.g. with partial occlusions). Moreover, the estimation of the orientation from the direction of motion does not allow working with stationary cases. Therefore, in the next chapter, we will propose a tracking framework with a stochastic approach for estimating both location and viewpoint, and search for the optimum projective transformation by sampling multiple possible values for θ at multiple locations (X, Y).

5

View-invariant 3D Pose Tracking

5.1 Introduction

The goal in this chapter is to track and estimate the 3D pose of multiple walking people by means of view-based models independently of the point of view from which the scene is observed. As in the previous chapter, we will consider a discrete set of training views and exploit projective geometry to find view-invariance. We propose a stochastic approach for estimating both ground plane location and camera viewpoint, and improve the search of the optimum projective transformation for pose recognition by sampling multiple possible values for θ at multiple locations (X, Y). Applying a different projective transformation to the input image for each sampled triplet (X, Y, θ) and processing each resulting warped image in a *bottom-up* manner as we did in the previous chapter would be computationally inefficient. We instead consider a stochastic *top-down* approach for body pose (and associated appearance descriptor). The idea is to learn mappings between a body pose manifold and the 2D silhouette features (shape). For each triplet, a pose is then sampled on this manifold and the corresponding shape is evaluated in the original input image using the inverse homographic transformation. This approach will make the system more robust to possible drifting compared to the model employed previously in this thesis as it can maintain multiple hypotheses through time.

5.1.1 Related Work

3D pose tracking. Stochastic models have come to be the dominant way of approaching the problem of articulated 3D human body tracking: an approximate inference technique, usually a particle filtering, is used to tractably estimate the high-dimensional posture space [Deutscher and Reid, 2005, Canton-Ferrer et al., 2011, Li et al., 2010, Chang and Lin, 2010, Elgammal and Lee, 2009, Lee and Elgammal, 2010, Jaeggli et al., 2009]. Particle filtering allows modeling non-Gaussian multi-modal distributions and can maintain multiple hypotheses through time. However, the number of particles required to achieve an acceptable result considerably increases with the dimensionality of the search space. The number of degrees-of-freedom (generally more than 30) and the high dimensionality of the state space (i.e. valid poses) make the tracking problem computationally difficult. The search space gets even larger when the tracking algorithm also has to estimate the location, orientation and scale of the subject in the image or in the scene as in [Jaeggli et al., 2009]. Some work has investigated the use of learnt models of human motion to constrain the search in state space by providing strong priors on motion [Ning et al., 2004b, Urtasun et al., 2006a]. Others have focused their research on the problem of dimensionality reduction for pose tracking and proposed to use low dimensional embedding of human motion data: Gaussian process latent variable model (GPLVM) [Urtasun et al., 2006b, Ek et al., 2008, Andriluka et al., 2010], Locally Linear Embedding (LLE) [Jaeggli et al., 2009], supervised manifold learning [Elgammal and Lee, 2009, Lee and Elgammal, 2010] or coordinated mixture of factor analysers [Li et al., 2010] are some examples.

Most existing systems [Elgammal and Lee, 2009, Jaeggli et al., 2009, Andriluka et al., 2010] typically assume that the camera axis is parallel to the ground i.e. elevation angle $\varphi = 0$ (see Fig. 4.2a for angle definition) and that the observed people are vertically displayed i.e. rotation angle $\gamma = 0$. They discretize the viewpoint in a circle around the subjects, selecting a set of values for the azimuth θ: 36 orientations in [Jaeggli et al., 2009], 16 in [Rosales and Sclaroff, 2006], 12 in [Elgammal and Lee, 2009] and 8 in [Zhang et al., 2005b, Lan and Huttenlocher, 2004, Andriluka et al., 2010]. Results have been presented using different testing datasets in laboratory environments like HumanEva [Sigal et al., 2010] or challenging street views as in [Andriluka et al., 2010, Jaeggli et al., 2009], but generally training and testing images are captured from a similar environment or with a similar camera tilt angle. Very few present numerical evaluation of human pose tracking on surveillance scenario with low resolution and high perspective distortion, and few pose tracking algorithms exploit the key constraints provided by scene calibration which is available in a large number of real surveillance system. Zhao et al. [2008] presented tracking results in crowded video-surveillance sequences using a coarse 3D model but no body pose was estimated.

5.1.2 Overview

We tackle the problem of view-invariant 3D body pose tracking and explore the use of projective shape matching in a particle filtering framework which jointly explores a low dimensional pose-viewpoint manifold and the real world ground plane. Our approach is motivated by the encouraging preliminary results for view-invariant human motion analysis obtained in the previous chapter and in our earlier work [Rogez et al., 2006a] and the recent advances in low-dimensional manifold learning for human pose tracking [Jaeggli et al., 2009, Elgammal and Lee, 2009]. Given its proven effectiveness, we choose to model 3D walking poses using a low dimensional torus manifold for camera viewpoint and pose as in [Elgammal and Lee, 2009]. We map this manifold to our view-based silhouette manifolds using kernel-based regressors, which are learnt using a Relevance Vector Machine (RVM). Given a point on the surface of the torus, the resulting generative model can regress the corresponding pose and view-based silhouette as illustrated in Fig. 5.2b.

During the online stage, 3D body poses are thus tracked using a recursive Bayesian sampling conducted jointly over the scene's ground plane and this pose-viewpoint torus manifold. For each sample, the homography that relates the corresponding training plane to the image points can be calculated using the dominant 3D directions of the scene, the sampled location on the ground plane and the sampled camera view as explained in the previous chapter. Each regressed silhouette shape is then projected using the projective transformation and matched in the image to estimate its observation likelihood. Our tracking framework is depicted in Fig. 5.1.

The rest of the chapter is organized as follows: in Sect. 5.2, we introduce the torus manifold for pose and appearance modeling. In Sect. 5.3, we detail our tracking framework. Experimentations with qualitative and quantitative evaluations are presented in Sect. 5.4 and some conclusions are finally drawn in Sect. 5.5.

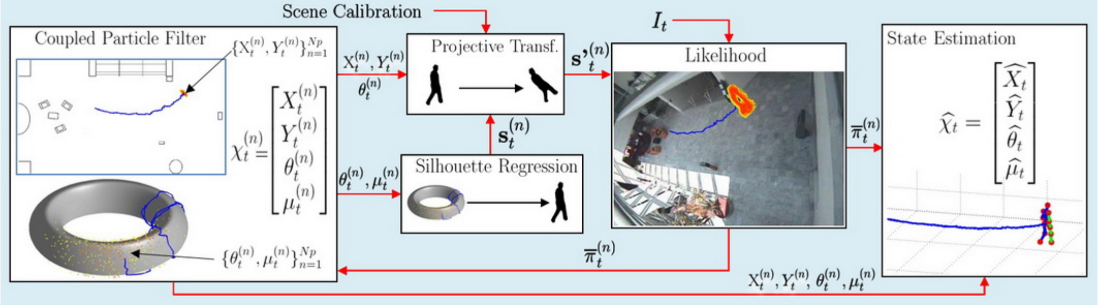

Scene Calibration
Coupled Particle Filter
$\{X_t^{(n)}, Y_t^{(n)}\}_{n=1}^{Np}$
$\chi_t^{(n)} = [X_t^{(n)}, Y_t^{(n)}, \theta_t^{(n)}, \mu_t^{(n)}]$
$\{\theta_t^{(n)}, \mu_t^{(n)}\}_{n=1}^{Np}$
$X_t^{(n)}, Y_t^{(n)}$
$\theta_t^{(n)}$
Projective Transf.
$\mathbf{s}'^{(n)}_t$
$\mathbf{s}_t^{(n)}$
$\theta_t^{(n)}, \mu_t^{(n)}$
Silhouette Regression
I_t
Likelihood
$\overline{\pi}_t^{(n)}$
State Estimation
$\widehat{\chi}_t = [\widehat{X}_t, \widehat{Y}_t, \widehat{\theta}_t, \widehat{\mu}_t]$
$X_t^{(n)}, Y_t^{(n)}, \theta_t^{(n)}, \mu_t^{(n)}$

Figure 5.1: System Flowchart: the 3D body poses are tracked using a recursive Bayesian sampling conducted jointly over the scene's ground plane (X, Y) and the pose-viewpoint (θ, μ) torus manifold ([Elgammal and Lee, 2009]). For each sample n, a projective transformation relating the corresponding training plane and the image points is calculated using the dominant 3D directions of the scene, the sampled location on the ground plane $(X_t^{(n)}, Y_t^{(n)})$ and the sampled camera view $\theta_t^{(n)}$. Each regressed silhouette shape $\mathbf{s}_t^{(n)}$ is projected using this homographic transformation obtaining $\mathbf{s}_t'^{(n)}$ which is later matched in the image to estimate its likelihood and consequently the importance weight. A state, i.e. an oriented 3D pose in 3D scene, is then estimated from the sample set.

5.2 Torus Manifold for Pose and Appearance Modeling

Full body pose configurations are necessarily high dimensional; in our case, we use 13 3D-joint locations in a human-centered coordinate system for our representation which results in a 39-dimensional pose configuration. To reduce the problem of high dimensionality in the learning stages, a dimensionality reduction step is needed to identify a low-dimensional embedding of the pose space. As we focus on the walking action which is cyclic, we consider a low-dimensional manifold embedding both camera viewing angle and body pose together and jointly model them by means of a torus manifold. Elgammal and Lee [2009] numerically demonstrated with experimental evaluation that the supervised torus embedding shows much better performances than unsupervised manifold representations (LLE, Isomap, GPLVM).

If $\mu \in [0,\ 1)$ is the body pose configuration on the torus and $\theta \in [-\pi, \pi]$ is the viewing angle [1], then the torus manifold illustrated in Fig. 5.2b can be defined parametrically in Euclidean space by:

$$\begin{aligned} x &= (R + r\cos 2\pi\mu)\cos\theta, \\ y &= (R + r\cos 2\pi\mu)\sin\theta, \\ z &= r\sin 2\pi\mu \end{aligned} \tag{5.1}$$

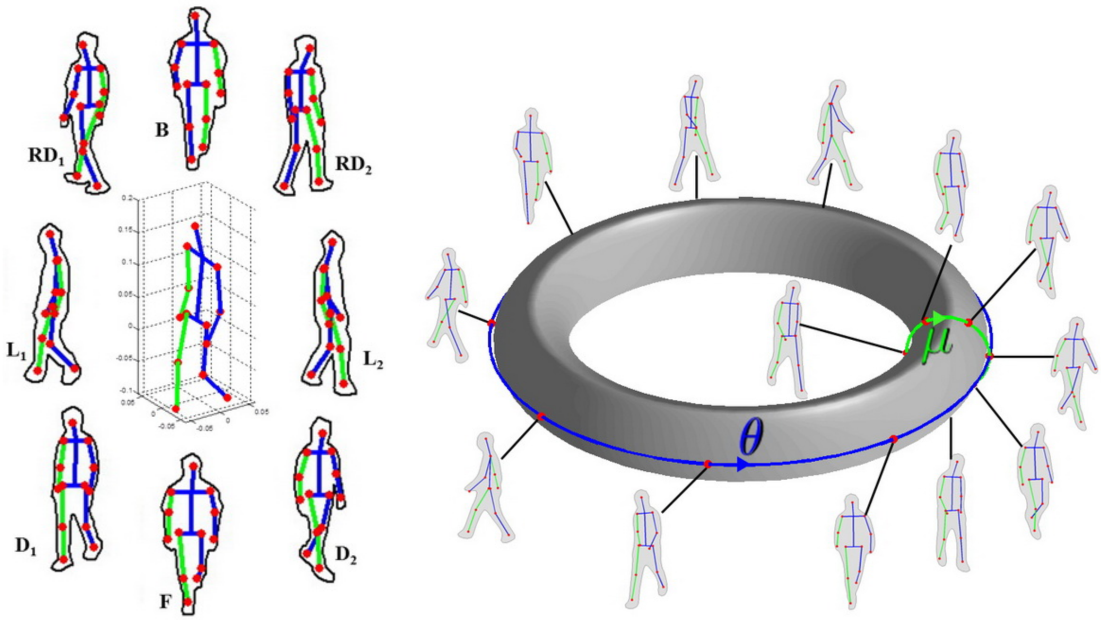
where R is the "major radius", i.e. the distance from the center of the tube to the center of the torus, and r is the "minor radius", i.e. the radius of the tube.

5.2.1 Body Pose Modeling

The training sequences are mapped onto the surface of the torus. We refer the reader to [Elgammal and Lee, 2009] for details. We learn the mapping back to the original data space from the torus manifold with a kernel regressor:

$$\mathbf{K} = f_p(\mu, \theta) = \mathbf{W}_p \mathbf{\Phi}_p(x, y, z), \tag{5.2}$$

[1] Note that for consistency with previous sections, we keep the viewpoint parameter as an angle while Elgammal and Lee [2009] define both viewpoint and action parameter in [0, 1) space.



(a) (b)

Figure 5.2: (a) Data used in this chapter: example of a training 3D pose and its 8 view-based 2D silhouettes and 2D poses extracted from the MoBo dataset. (b) Pose-viewpoint torus manifold (adapted from [Elgammal and Lee, 2009]) learned using the Mobo dataset. The 2 dimensions of the surface represent gait cycle and camera viewpoint. We represent 8 different views of a same pose (blue circle), and 6 different poses from a same viewpoint (green circle).

where $\mathbf{K} \in \mathbb{R}^{3\times13}$ is the orientated body pose configuration in the original 3D pose training space, $\mathbf{\Phi}_p$ is a vector of kernel functions and $\mathbf{W}_p$ is a matrix of weights[2]. The matrix $\mathbf{W}_p$ is learnt using a Relevance Vector Machine (RVM). We use radial basis functions as the kernel functions in $\mathbf{\Phi}_p$ computed at the training data locations. Any point $(\mu, \theta) \in [0,\ 1) \times [-\pi, \pi]$, on the surface of the torus, can be directly mapped to an oriented 3D pose.

5.2.2 Appearance Modeling

Different shape representations have been used for human silhouettes in recent literature, including parametric B-splines [Isard and Blake, 1998], shape context [Belongie et al., 2002, Mori and Malik, 2006], level-sets [Cremers, 2006], pose-adaptive shape descriptors [Lin and Davis, 2010] and distance transform [Jaeggli et al., 2009, Elgammal and Lee, 2009]. Again, we select the landmark parameterization [Baumberg and Hogg, 1994, Siebel and Maybank, 2002b, Giebel et al., 2004], i.e. a set of N_l 2D-landmarks, to represent the silhouette:

$$\mathbf{s} = [\mathbf{x}_{s_1}, ..., \mathbf{x}_{s_{N_l}}] \in \mathbb{R}^{2\times N_l}. \tag{5.3}$$

Although non-linearity and normalization issues can appear during the training phase as we discussed in chapter 2, landmark-based shape representations are lower dimensional and much simpler to manipulate and transform. They also facilitate a very quick matching with the image making them ideal in a top down particle filtering framework.

[2] As done in [Elgammal and Lee, 2009], we map from the Euclidean space where the torus lives and not from the coordinate system (μ, θ) since this coordinate system is not continuous at the boundary.

We now model the generative mapping from embedded pose μ to silhouette descriptors $\mathbf{s}$ that allows us to predict image appearance given an hypothesis for the pose μ and for the body orientation or camera viewpoint θ. In this work, the viewing hemisphere is discretized into a finite number N_v of training viewpoints $\{\theta_v\}_{v=1}^{N_v}$ varying the azimuth angle (see example in Fig. 5.2a). For each training viewpoint a mapping is learnt from the torus manifolds to the corresponding view-based silhouette manifold, which are learnt using a Relevance Vector Machine (RVM):

$$\mathbf{s} = f_s(\mu, \theta), \; \forall \mu \in [0,\ 1)\,,\; \forall \theta \in \{\theta_v\}_{v=1}^{N_v}, \tag{5.4}$$

with

$$f_s(\mu, \theta) = \mathbf{W}_s \mathbf{\Phi}_s(x, y, z). \tag{5.5}$$

Once again, the mapping $f_s(\mu, \theta)$ is learnt using RVM with weights $\mathbf{W}_s$ and kernel functions $\mathbf{\Phi}_s(\mu, \theta)$. Given a point $(\mu, \theta) \in [0,\ 1) \times \{\theta_1, \cdots, \theta_{N_v}\}$ on the torus manifold, the resulting generative model can generate the corresponding view-based silhouette. Note that in this work, the shape descriptor $\mathbf{s}$ is augmented with the 13 2D-joints $\mathbf{k} = [\mathbf{x}_{k_1}, ..., \mathbf{x}_{k_{13}}] \; \in \mathbb{R}^{2\times 13}$ to facilitate the estimation of a 2D pose error in the experiment Section.

5.3 Recursive Bayesian Sampling

5.3.1 Formulation.

At each time step, we simultaneously perform body pose estimation and image localisation since both processes can benefit from the coupling of the posture and image location as demonstrated in [Jaeggli et al., 2009]. As explained in Sect. 4.2.2, the advantage of assuming a calibrated environment and a planar ground plane is the considerable reduction of the search space as image location, scale and rotation can be recovered from the real world ground plane location. Thus, we define the state vector of the target as:

$$\chi_t = [X_t \; Y_t \; \theta_t \; \mu_t \,]\,, \tag{5.6}$$

consisting of the real-world ground plane location (X_t, Y_t) and the embedding coordinates on the torus surface $(\mu_t, \theta_t) \in [0,\ 1) \times [-\pi, \pi]$. The calibration of the scene and the torus embedding help us to face a much more tractable problem as the search has to be performed in a 4-dimensional state space while, for instance, Jaeggli et al. [2009] explore a 10-dimensional space.

We formulate the tracking problem as a Bayesian inference task, where the state of the tracked subject is recursively estimated at each time step given the evidence (image data) up to that moment. Formally, within the Bayesian filtering framework, we formulate the computation of the *posterior* distribution $p(\chi_t|\mathbf{I}_t)$ of our model parameters χ_t over time as follows:

$$p(\chi_t|\mathbf{I}_t) \propto p(I_t|\chi_t)\, p(\chi_t|\mathbf{I}_{t-1}), \tag{5.7}$$

where $\mathbf{I}_t$ is the image sequence up to time t and $p(I_t|\chi_t)$ is the *likelihood* of observing the image I_t given the parameterization χ_t of our model at time t, in other words the *observation density*. Finally $p(\chi_t|\mathbf{I}_{t-1})$ is the *a priori density*, which is the result of applying the *dynamic model* $p(\chi_t|\chi_{t-1})$ to the *a posteriori density* $p(\chi_{t-1}|\mathbf{I}_{t-1})$ of the previous time step:

$$p(\chi_t|\mathbf{I}_{t-1}) = \int p(\chi_t|\chi_{t-1})\, p(\chi_{t-1}|\mathbf{I}_{t-1})\, d\chi_{t-1}. \tag{5.8}$$

Unfortunately, when the involved distributions are non-Gaussian, Eq. (5.7) cannot be solved analytically. Instead, we use a particle filter [Isard and Blake, 1998, Sidenbladh et al., 2002, Deutscher and Reid, 2005] in order to approximate the true *posterior* pdf $p(\chi_t|\mathbf{I}_t)$ by means of a discrete weighted set of samples $\{\chi_t^{(n)}, \pi_t^{(n)}\}_{n=1}^N$:

$$p(\chi_t|\mathbf{I}_t) \approx \sum_{n=1}^{N} \pi_t^{(n)} \delta(\chi_t^{(n)}), \tag{5.9}$$

where for each particle, δ denotes the Dirac delta and $\pi_t^{(n)}$ is the normalized importance weight which is directly derived from measurement likelihood:

$$\pi_t^{(n)} = \frac{p(I_t|\chi_t^{(n)})}{\sum_{n'=1}^{N} p(I_t|\chi_t^{(n')})}, \tag{5.10}$$

as defined in [Isard and Blake, 1998].Hence, whilst the likelihood function decides which particles are worth propagating, the dynamic model is responsible for guiding the exploration through the state space.

5.3.2 Dynamic Model.

Since a static camera is being considered in this work, and assuming the people face along the direction of motion, we model the dependence of viewpoint on ground plane location while we assume statistical independence between the remaining state variables[3]. The dynamic model $p(\chi_t|\chi_{t-1})$ is thus a product of four dynamic models, i.e. $p(\chi_t|\chi_{t-1}) = p(X_t|X_{t-1}, \theta_{t-1})p(Y_t|Y_{t-1}, \theta_{t-1})p(\theta_t|\theta_{t-1})p(\mu_t|\mu_{t-1})$.

Therefore, our state model has the following form on the torus manifold:

$$\theta_t = \theta_{t-1} + n_\theta, \tag{5.11}$$

$$\begin{bmatrix} \mu_t \\ \dot{\mu}_t \end{bmatrix} = \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_{t-1} \\ \dot{\mu}_{t-1} \end{bmatrix} + \begin{bmatrix} n_\mu \\ n_{\dot{\mu}} \end{bmatrix}, \tag{5.12}$$

where n_θ, n_μ and $n_{\dot{\mu}}$ are zero mean white Gaussian noises (whose variances are set to $\sigma_\theta = \frac{\pi}{10}$, $\sigma_\mu = 0.075$ and $\sigma_{\dot{\mu}} = 0.0125$ respectively) and δt the time interval between successive frames, and on the ground plane:

$$\begin{bmatrix} X_t \\ Y_t \end{bmatrix} = \begin{bmatrix} X_{t-1} \\ Y_{t-1} \end{bmatrix} + n_V \begin{bmatrix} V_X \\ V_Y \end{bmatrix} + n_{XY} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{5.13}$$

where n_{XY} and n_V are zero mean white Gaussian noise (with variance set to $\sigma_{XY} = 1$cm and $\sigma_V = 10$cm in our experiments) and $[V_X, V_Y]^T = \mathbf{V}/||\mathbf{V}||$ is the unit orientation vector relating θ and the camera viewing direction $\mathbf{C}$ (see Eq. 4.8 and Eq. 4.10). In this way, we model the fact that pedestrians are more likely to move in the facing direction and, after a stationary phase, we can predict in which way the subject is going to move based on his body orientation (i.e. viewpoint angle θ).

5.3.3 Image Measurements - Observation Model.

The *likelihood* function $p(I|\chi)$ computes how likely it is to observe the image I given the unknown state χ. Given the viewpoint θ and the location on the ground plane (X, Y), a

[3] We choose not to model the dependencies between the gait parameter μ and the ground plane location because stride length depends on the subject morphology and walking style.

training view Φ_v with angle θ_v is selected following Tab. 4.1 and Eq. 4.11. The transformation $\mathbf{P}_{I\,\Pi_v\Phi_v} = (\mathbf{P}_{\Phi_v\Pi_v I})^{-1}$ that relates the training plane Φ_v to the image points is then calculated using the X, Y, θ and the dominant 3D directions of the scene following Alg. 3 in Sect. 4.3. The regressed silhouette descriptor $\mathbf{s} = f_s(\mu, \theta_v)$ is then projected on the image I obtaining $\mathbf{s}' = [\mathbf{x}'_{s_1}, ..., \mathbf{x}'_{s_{N_l}}] \in \mathbb{R}^{2\times N_l}$. For each landmark $l \in \{1, N_l\}$, $\mathbf{x}'_{s_l} = (x'_{s_l},\ y'_{s_l})$ is obtained following:

$$\underline{\mathbf{x}}'_{s_l} = \mathbf{P}_{I\,\Pi_v\Phi_v} \cdot \underline{\mathbf{x}}_{s_l},\ \forall l \in \{1, N_l\}. \tag{5.14}$$

The *likelihood* $p(I|\chi)$ is now estimated using this projected silhouette $\mathbf{s}'$. To keep the required time for computing the likelihood of each sample as low as possible, we chose to employ only low-level processing tasks like background subtraction or edge detection algorithms. Thus, the observations are based on the edge map I_{edges} of the image, as well as the binary foreground detection mask I_{fgd}. The pixel color values are not considered in this work. The joint likelihood is approximated as:

$$p(I|\chi) = p(I_{edges}|\chi)p(I_{fgd}|\chi). \tag{5.15}$$

The projected silhouette shape $\mathbf{s}'$ is used to compute the first likelihood term $p(I_{edges}|\chi)$ using a Chamfer distance function as in [Stenger et al., 2006]. Both silhouette $\mathbf{s}'$ and edge map I_{edges} are first decomposed into a number N_γ of separate orientation channels according to gradient orientation. The elements of $\mathbf{s}'$ are thus decomposed into N_γ lists of landmark indexes $\{\Gamma_\gamma\}_{\gamma=1}^{N_\gamma}$, i.e. $\forall l \in \{1, N_l\}$, $\exists!\ \gamma \in \{1, N_\gamma\} : l \in \Gamma_\gamma$. A Distance Transform (DT) of the edge image I_{edges} is then computed separately for each channel obtaining $\{\mathcal{D}_\gamma\}_{\gamma=1}^{N_\gamma}$ and a Chamfer distance $d_{Ch} \in [0, 1]$ is computed [4] as:

$$d_{Ch}(\mathbf{s}', I_{edges}) = \frac{1}{\tau.N_l}\sum_{\gamma=1}^{N_\gamma}\sum_{l\in\Gamma_\gamma}\mathcal{D}_\gamma(x'_{s_l}, y'_{s_l}), \tag{5.16}$$

where τ, the upper bound on the distance to the edge, is used to threshold the DT image and increase robustness toward partial occlusion as indicated in [Stenger et al., 2006]. We consider $\tau = 5$ pixels and $N_\gamma = 4$ orientation bins. Note that the elements in $\mathbf{s}'$ are rounded off before the computation of the Chamfer distance, i.e. $\mathbf{s}' \in \mathbb{N}^{2N_l}$. The edge based likelihood function is then defined as:

$$\begin{aligned} p(I_{edges}|\chi) &= p(I_{edges}|\mathbf{s}'), \\ &\propto \exp(-\lambda_e\ d_{Ch}(\mathbf{s}', I_{edges})), \end{aligned} \tag{5.17}$$

i.e. a Laplacian distribution over the distance d_{Ch} as in [Stenger et al., 2006]. In this work, we select $\lambda_e = 4$ (see Fig. 5.3).

The second likelihood term $p(I_{fgd}|\chi)$ aims at comparing two binary silhouettes: $\mathbf{s}'$ and the detection blob from I_{fgd}, obtained by state-of-the-art background subtraction. In surveillance videos, the problem is not as straightforward as it appears: occlusions, shadows, cluttered background, motion blur and low image resolution lead to low quality foreground silhouettes. Some approaches thus consider the foreground image as a binary mask to select foreground edges [Hofmann and Gavrila, 2012]. To cope with low quality foreground silhouettes, we introduce a new way to measure the fitness of a shape in a binary image. From $\mathbf{s}'$, we define two sets of 2D-landmarks constituting two new shapes $\mathbf{s}^{in}, \mathbf{s}^{out} \in \mathbb{N}^{2N_l}$, the inner boundary points and outer boundary points so that $\forall l \in \{1, N_l\}$:

$$\begin{bmatrix} x^{in}_{s_l} \\ y^{in}_{s_l} \end{bmatrix} \triangleq \begin{bmatrix} x'_{s_l} \\ y'_{s_l} \end{bmatrix} - \delta_s \mathbf{u}^{\perp}_{s_l} \text{ and } \begin{bmatrix} x^{out}_{s_l} \\ y^{out}_{s_l} \end{bmatrix} \triangleq \begin{bmatrix} x'_{s_l} \\ y'_{s_l} \end{bmatrix} + \delta_s \mathbf{u}^{\perp}_{s_l}, \tag{5.18}$$

[4] The DT image takes the set of feature points as input and assigns each location the distance to its nearest feature.

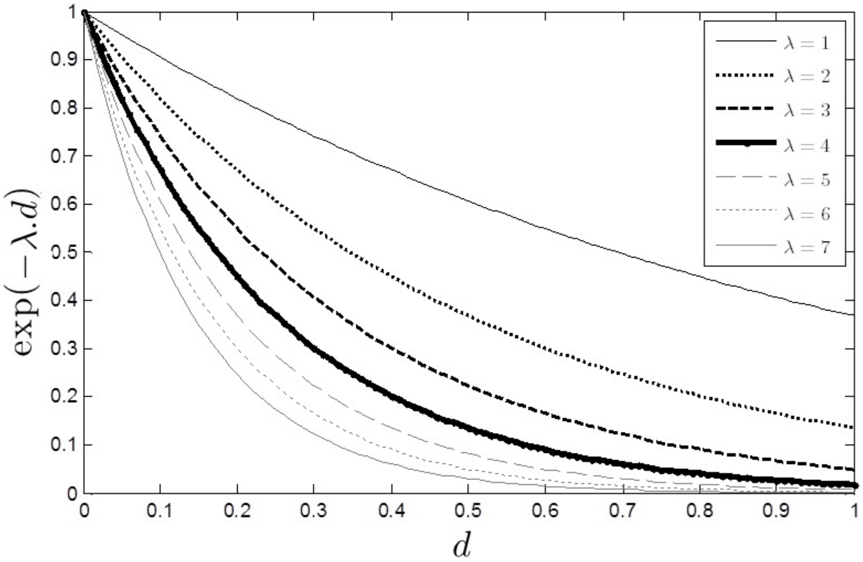
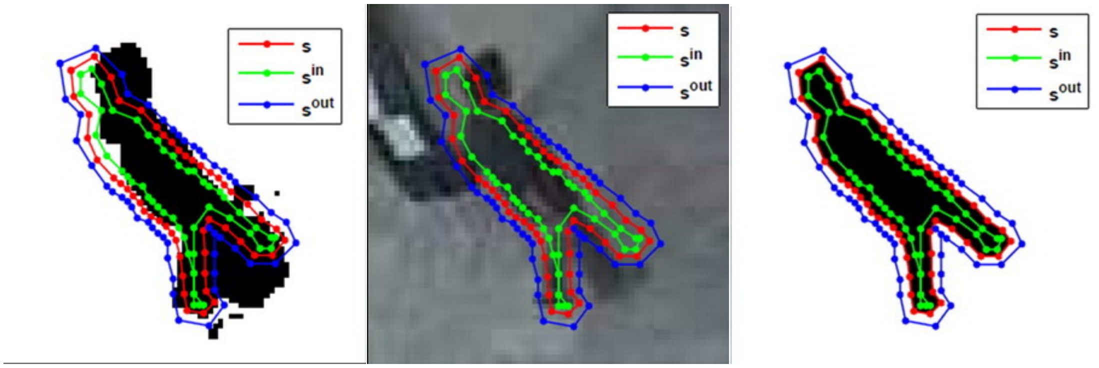


Figure 5.3: Likelihood function as a Laplacian distribution $\exp(-\lambda\ d)$ over the distance $d \in [0,1]$ for several values of the parameter λ. In this work, we consider an acceptable Likelihood function is obtained with $\lambda = 4$.



(a) (b) (c)

Figure 5.4: Foreground Likelihood: (a) The shape $\mathbf{s}'$ is matched on an input foreground blob. The new shapes $\mathbf{s}^{in}$ and $\mathbf{s}^{out}$ (in green and blue respectively) are the inner and outer boundary points, 2 pixels away from the original landmarks in $\mathbf{s}'$. The resulting distance $d_{Fgd}(\mathbf{s}', I_{fgd}) = 0.44$. The same shapes are represented on top of the original input image in (b), while in (c) we visualize the same 3 shapes on top of a foreground blob returning $d_{Fgd}(\mathbf{s}', I_{fgd}) = 0$, i.e. a perfect likelihood $p(I_{fgd}|\chi) = 1$.

where $\mathbf{u}_{s_l}^{\perp}$ is a unit vector passing through the landmark l and perpendicular to the shape, pointing outside of the shape. In this work, we consider $\delta_s = 2$ pixels (see examples in Fig. 5.4). We define $\mathcal{S}_{in}, \mathcal{S}_{out} \ \in [0,1]$:

$$\begin{aligned} \mathcal{S}_{in} &\triangleq \tfrac{1}{N_l} \textstyle\sum_{l=1}^{N_l} I_{fgd}(x_{s_l}^{in}, y_{s_l}^{in}), \\ \mathcal{S}_{out} &\triangleq 1 - \tfrac{1}{N_l} \textstyle\sum_{l=1}^{N_l} I_{fgd}(x_{s_l}^{out}, y_{s_l}^{out}) \end{aligned} \tag{5.19}$$

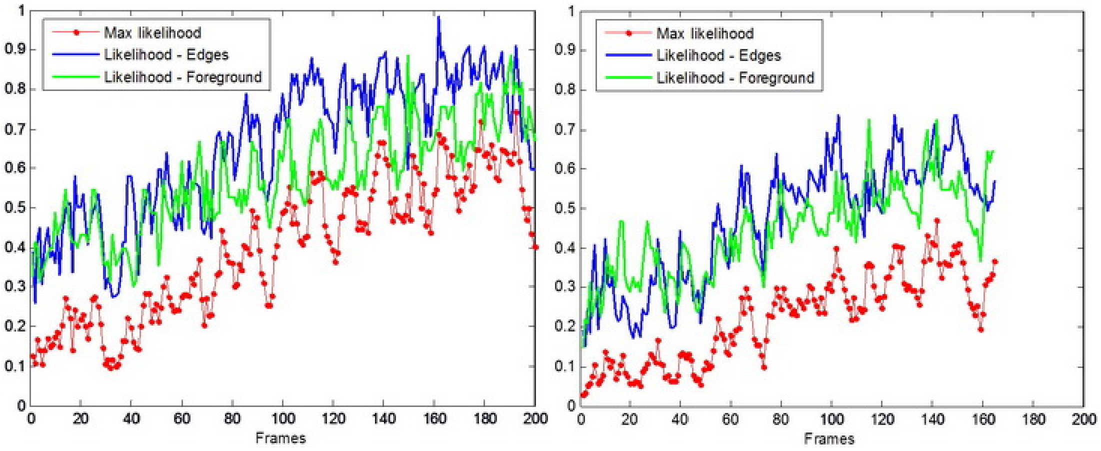
as the shape-to-foreground and shape-to-background similarities respectively. $\mathcal{S}_{in}$ indicates the amount of foreground pixels inside the shape $\mathbf{s}'$ while $\mathcal{S}_{out}$ informs on the quantity of background outside $\mathbf{s}'$. Finally the resulting distance is defined as:

$$d_{Fgd}(\mathbf{s}', I_{fgd}) \triangleq 1 - \frac{\mathcal{S}_{in} + \mathcal{S}_{out}}{2}. \tag{5.20}$$

The rationale behind this definition of d_{Fgd} is that the distance metric should be high with noisy segmentation (occlusions and shadows) and zero with perfect matches. The likelihood is modeled, again, as a Laplacian distribution over this new distance measure:

$$p(I_{fgd}|\chi) \propto \exp(-\lambda_f \; d_{Fgd}(\mathbf{s}', I_{fgd})), \tag{5.21}$$

where λ_f is chosen so that the two likelihood terms have the same importance in Eq. 5.15, i.e. $\lambda_f = \lambda_e = 4$. See examples in Fig. 5.5. On a state-of-the-art laptop with an Intel Core @ 1.73GHz, the average computation time of the likelihood is about 2 ms per sample (0.4 ms for $p(I_{edges}|\chi)$ and 1.6 ms for $p(I_{fgd}|\chi)$ in unoptimized Matlab code and considering $N_l = 50$ landmark points to parameterize the shapes. In Fig. 5.6, the likelihood of the entire sample set can be visualized for four frames of the Walk2 sequence.



(a) (b)

Figure 5.5: Likelihood Examples: for Walk1 (a) and Walk2 (b) sequences. For each frame, we give the largest likelihood value (*red*) over a sample set of 500 particles, and the corresponding foreground $p(I_{fgd}|\chi)$ (*green*) and edges $p(I_{edges}|\chi)$ (*blue*) likelihood terms. In both sequences, the subject follows a similar path but, in Walk1 (a) the subject wears dark clothes while in Walk2 (b) the subject wears pale clothing against a pale background, explaining the highest likelihood values in (a) compared to (b). See dataset details in Tab. 5.1.

5.3.4 Tracking Multiple Pedestrians.

There is an extensive literature on particle filtering for tracking multiple interacting targets with a single calibrated camera [MacCormick and Blake, 2000, Smith et al., 2005, Isard and MacCormick, 2001, Zhao and Nevatia, 2004]. Visual interactions among targets can be exploited when defining the likelihood term of a multiple-object filter as done in the BraMBLe system [Isard and MacCormick, 2001]. Dealing with occlusion is simplified when using the 3D positions of the multiple targets and even more when a roof-top surveillance camera is used. With such cameras, the heads are almost always visible and complete occlusions rarely occur. We thus choose to simply instantiate several independent 1-subject trackers and follow a simple but effective approach to avoid the coalescence of the trackers onto the best-fitting target (e.g.

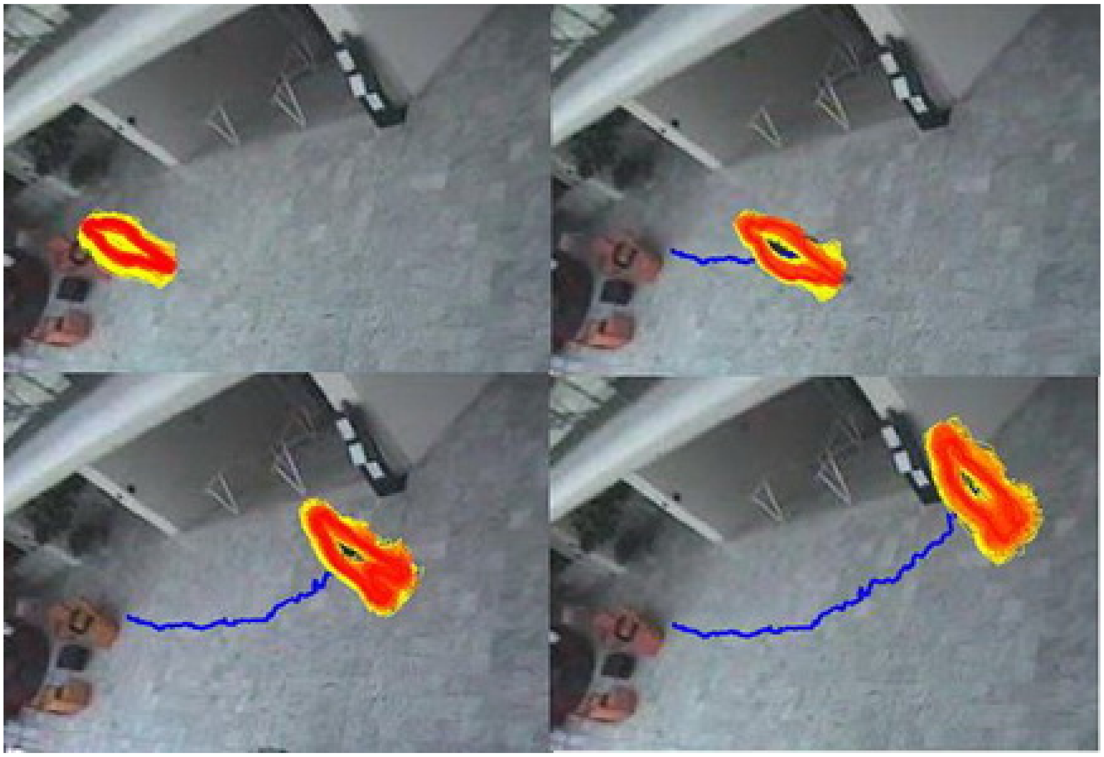


Figure 5.6: Visualization of the likelihood for the entire sample set in several frames of the Walk2 sequence. For each frame, we plot the set of sampled and aligned shapes: darker colors indicate a higher likelihood.

subject wearing dark clothes with a pale background): we model each subject's 3D occupancy on the ground floor with a Gaussian probability function centered on the subject's estimated location which is then employed to downweight the particles from the other targets. Considering N_s subjects/targets with N_s individual trackers, the samples of each tracker are reweighted accordingly and the normalized importance weight of the n^{th} particle of subject s at time t thus becomes:

$$\pi_{t,s}^{(n)} \propto p(I_t|\chi_{t,s}^{(n)}) \prod_{\substack{s'=1 \\ s' \neq s}}^{N_s} (1 - \lambda_o \ \exp(-\frac{||\widehat{\chi}_{t-1,s'} - \chi_{t,s}^{(n)}||_{gd}^2}{\sigma_o^2}))^{\eta_o}, \tag{5.22}$$

with $\sum_{n=1}^{N} \pi_{t,s}^{(n)} = 1$ and where $||.||_{gd}$ is the Euclidean distance on the ground floor and σ_o, λ_o and η_o are defined empirically. Results provided in the next section are obtained with $\sigma_o = 50$ cm, $\lambda_o = 1$ and $\eta_o = 3$.

The approach we follow to deal with multiple targets may seem simplistic at first sight. However, it performs sufficiently well for the cases we have considered in this work, i.e. no severe occlusions and basic interactions where a few people meet, chat and walk together. The problem of multiple target tracking in more complex situations is out-of-scope for this chapter and we leave for future work the use of a multiple-object filter or a more adequate modeling of the interactions. The complete tracking algorithm is summarized in Alg. 5.

The state $\widehat{\chi}_t$ at a particular time step is usually estimated using a Monte Carlo approximation of the expectation of the posterior pdf, i.e. a weighted sum over the set of samples: $\widehat{\chi}_t^{MC} = \mathcal{E}[\chi_t] = \sum_{n=1}^{N} \pi_t^n \chi_t^n$.

Algorithm 5: Particle Filter Algorithm

```
Initialize a sample set $\{\chi_{0,s}^{(n)}, \frac{1}{N}\}_{n=1}^{N}$ for each subject $s$ according to prior distribution $p(\chi_0)$;
for each time step $t$ do
    for each subject $s$ do
        for each particle $n$ do
            Resample $\{\chi_{t-1,s}^{(n)}, \pi_{t-1,s}^{(n)}\}_{n=1}^{N}$ to obtain a new sample $\{\chi_{t-1,s}^{\prime(n)}, 1\}$;
            Propagate $\chi_{t-1,s}^{\prime(n)}$ using the dynamic model $p(\chi_t|\chi_{t-1})$ to obtain $\chi_{t,s}^{(n)}$;
            Compute likelihood $p(I_t|\chi_{t,s}^{(n)})$ from Eq. 5.15;
            Update weight $\pi_{t,s}^{(n)}$ using Eq. 5.22;
        Normalize $N$ weights $\pi_{t,s}^{(n')} = \pi_{t,s}^{(n)} / \sum_{n=1}^{N} \pi_{t,s}^{(n)}$;
        Estimate the state $\widehat{\chi}_{t,s}$ ;
```

An estimation of the state can also be made by selecting one of the particles. For instance, the maximum a posteriori (MAP) estimate $\widehat{\chi_t}^{MAP}$ given by the particle with the largest normalized weight has been broadly considered, especially for human pose estimation problems [Elgammal and Lee, 2009]. The Viterbi path finding algorithm can also be considered to choose one of the samples at each time t and form a trajectory through time and state space that best satisfies both observation likelihood and temporal prior as in [Jaeggli et al., 2009]. Viterbi estimate $\widehat{\chi_t}^{Vit}$ takes into account temporal consistency and can solve possible ambiguities and multi-modal distributions, which often happen in articulated body tracking. The particle filter will usually be able to concentrate particles in the main mode of the likelihood function. However, multiple modes of similar size in the likelihood function might bias MC estimation $\widehat{\chi_t}^{MC}$. MAP and Viterbi based methods require a large number of particles to reach the optimal position precisely leading to high computational cost. Moreover it is not guaranteed that the optimal position is necessarily sampled, even when a large number of particles are employed.

In this work, we propose a new hybrid way of estimating the state at each time step which is derived based on the discrete approximation of the posterior but also takes advantage of the temporal consistency of a Viterbi based estimate. We first define $\mathcal{N}_t^{Vit}$ a neighborhood around $\widehat{\chi_t}^{Vit}$, the sample selected by Viterbi, and consider a local weighted sum of the particles belonging to that neighborhood:

$$\widehat{\chi}_t = \frac{\sum_{n \in \mathcal{N}_t^{Vit}} \pi_t^{(n)} \chi_t^{(n)}}{\sum_{n \in \mathcal{N}_t^{Vit}} \pi_t^{(n)}}, \tag{5.23}$$

where the neighborhood $\mathcal{N}_t^{Vit}$ is defined in the coupled ground floor-torus state space with a circular region around the Viterbi estimate $\widehat{\chi_t}^{Vit}$ on both ground floor and torus surface:

$$\mathcal{N}_t^{Vit} \triangleq \left\{ n \ : \ ||\widehat{\chi}_t^{Vit} - \chi_t^{(n)}||_{gd} \leq \rho_{gd} \ \wedge \ ||\widehat{\chi}_t^{Vit} - \chi_t^{(n)}||_{tor} \leq \rho_{tor} \right\}, \tag{5.24}$$

where $||.||_{gd}$ and $||.||_{tor}$ are the Euclidean distance on ground floor and torus manifold respectively while ρ_{gd} and ρ_{tor} are the two radii defining the neighborhood on the ground floor and torus manifold respectively. In our experiments, we set $\rho_{gd} = 10$ cm and $\rho_{tor} = 0.1$.

5.4 Experimental Results

The comparison with state-of-the-art work is not straightforward for several reasons. First, standard testing data sets for pose estimation (e.g. HumanEva [Sigal et al., 2010]) do not consider perspective distortion and can not be used in this chapter to offer a quantitative comparison. We will instead employ the Caviar dataset [Caviar, 2004] that presents very challenging sequences with perspective distortion but, as far as we know, no pose estimation results (apart from our work) have been published on this dataset. The ground truth labelled for this paper on the Caviar dataset will be made publicly available to the scientific community for further research and comparison. We present in Tab. 5.1 the selected sequences.

Table 5.1: Testing dataset considered in the chapter. The selected sequences belong to the Caviar dataset [Caviar, 2004]. Eleven tracks of walking people have been considered for manual ground truth annotation (i.e. manual localization of the 13 2D-joints defining a 2D pose). For each track, we indicate the selected frames, the subject ID, the number of available ground truth 2D poses and a short description with possible difficulties.

Sequence	**Track**	**Subj. ID**	**Frames**	**No. Poses**	**No. Frames**	**Description Difficulty**
Walk1	1	1	260-459	200	200	Dark clothing - Good segmentation
Walk2	2	2	0304-0468	165	289	Pale clothing - Bad segmentation
	3		0931-1054	124		
Walk3	4	3	0500-0649	150	310	Dark clothing - Good segmentation
	5		1200-1359	160		
LeftBag_PickedUp	6	4	0314-0413	100	100	The subject carries a bag
Meet_WalkTogether2	7	2	190-509	320	320	Varied clothing color - Occlusions
	8	1	209-507	298		
Meet_Split_3rdGuy	9	5	077-742	666	666	Varied clothing color - Occlusions
	10	3	189-506	318		
	11	6	332-614	283		
Total		**6**		**2784**	**1885**	

Many papers consider 3D body pose estimation or localization in real-world images separately. Few papers [Jaeggli et al., 2009, Andriluka et al., 2010, Okada and Soatto, 2008] tackle both problems simultaneously as we do, but they pay no attention to the problem of perspective distortion (they consider a camera elevation angle $\varphi = 0$) and do not include scene knowledge in their frameworks. Since our system is more complete than state-of-the-art methods and takes into account a calibration of the camera w.r.t the scene, running these algorithms on the proposed testing dataset and making a comparison with our results would be unfair.

Nevertheless, we will compare the performances of our complete framework based on projective geometry with a simpler solution considering a shape alignment based on similarity (Eq. 4.5.1) and keeping the rest of the framework unchanged. Existing methods implicitly [Jaeggli et al., 2009, Andriluka et al., 2010, Okada and Soatto, 2008] or explicitly [Toyama and Blake, 2002] apply a similarity transformation between their models and the processed images, most of the time with only scale and translation elements but without any rotation in the image plane (i.e. $\gamma = 0$ in Eq. 4.5.1)[5]. Thus, a comparison of the performances of our framework replacing the projective transformation in Eq. 5.14 by a similarity transformation will provide a quantitative evaluation of the improvement achieved by our proposal w.r.t. state-of-the-art [6].

5.4.1 Settings and Parameters.

We use training silhouettes and 2D/3D poses extracted from the MoBo dataset [Gross and Shi, 2001] illustrated in Fig. 5.2a: for each one of the 8 training views, 15 walking cycles corresponding to 15 different subjects are temporally aligned, subsampled and averaged to compute a mean walking cycle made of 100 silhouettes and 2D poses. Thus, 800 silhouettes and associated 2D poses are used to learn the mapping between the torus manifold and the original data space. The same operation is performed with the 3D poses which are rotated around the Z-axis (azimuth θ) to cover the entire torus manifold. For each training view, we localize the horizontal and vertical vanishing points and compute the 8 homographies $\{\mathbf{H}_{\Phi_v \leftarrow \Pi_v}\}_{v=1}^{8}$.

We remove lens distortion from Caviar testing images and calibrate the camera w.r.t. the scene by localizing the vertical vanishing point and the horizontal vanishing line, and compute the homographies $\mathbf{H}_{\mathrm{g}}$ and $\mathbf{H}_{\mathrm{h}}$ from manual annotations.

In this work, we do not address the detection problem and take the ground plane location in the first frame from ground truth data, but the detector from [Li et al., 2008] would perfectly suit our framework as it can deal with perspective distortion and has shown significantly improved detection performance on the Caviar dataset. When a subject appears in the scene, we initialize a tracker by sampling in the entire space of possible poses and probable viewpoints. Supposing that the subject is facing in the direction of motion, the most probable viewpoints can defined based on the location in the first frame and the scene knowledge: e.g. if the subject is entering the scene by the right side, the viewpoint is most likely to be L_1, D_1 or RD_1 and viewpoint should be sampled in the corresponding part of the torus.

Note that, during tracking, the particles which fall in non-valid areas of the ground plane (such like walls, plants, etc) are assigned a 0 likelihood. When a subject is not moving, the likelihood is computed using only the shape landmarks corresponding to the upper part of the body. Since we model walking poses, our framework is not supposed to recognise standing poses. When motion is detected after a stationary phase, the tracker is reset by sampling in the entire space of possible poses.

5.4.2 Experiments.

We ran a series of tests on the selected Caviar sequences varying the number of particles in the filter. Since randomness is involved in the re-sampling of the particles, to gain better statistical significance, we perform the same experiments 20 times and from now on we compute numerical result as the average over these 20 runs. We repeat the same operation using a similarity transformation instead of our homographic transformation. First we propose to evaluate the performance of the tracker, independently of the state estimator. We thus consider that a target has been lost and the localization is not valid if the minimum distance (in the particle set) to ground truth location exceeds a certain value. We believe that a pose estimation does not make sense if the nearest particle is 1 meter away from the target true location,

[5] Most of these techniques are based on a scheme where the images are scanned with a sliding window at different scales.

[6] The similarity transformation is computed using 2 reference points in image and model view: head center and ground floor location recovered from the real world ground plane coordinate (X, Y) using $\mathbf{H}_{\mathrm{h}}$ and $\mathbf{H}_{\mathrm{g}}$ from Sect. 4.2.2.

i.e. $\min_{n\in\{1,N\}}\left(||\widehat{\chi}_t^{Gt} - \chi_t^{(n)}||_{gd}\right) \geq 100$ cm. A track is then considered lost when then the target has been lost during 20 frames or more and has not been recovered in the last frame of the sequence.

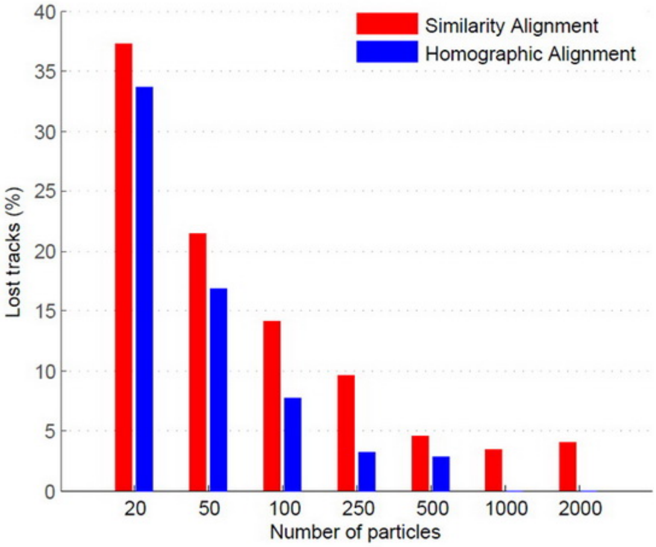


Figure 5.7: Percentage of lost tracks vs number of particles for similarity and homographic alignment. we present the average performance over 20 runs of the tracking algorithm on the 11 sequences: a track is considered lost when the tracking has failed during 20 frames or more (the distance between the nearest particle and ground truth location is over 1 meter) and it has not recovered by the end of the sequence, i.e. in the last frame the subject is still one meter away from ground truth for the nearest particle.

Results show that the proposed homographic alignment reduces the average percentage of lost tracks as can be observed in Fig. 5.7. The percentage of lost tracks decreases with the number of particles employed in the filter for both methods, but we reach 0% of lost tracks with 1000 particles and over while 5% of the tracks are still lost when considering 2000 particles and a similarity transformation. The perspective correction allows for better shape matching and consequently a more efficient shape-based tracking. If we look at the detailed results given in Tab. 5.2 and Tab. 5.3, we can observe how, on average, the homographic alignment outperforms the similarity alignment for 7 of the 11 tested tracks, i.e. it reaches 0% tracking failure with a smaller particle set (tracks 1, 2, 6, 7, 8, 9 and 10), while performing similarly for 3 other tracks (4, 5 and 11). Good tracking performance requires larger particle sets for the sequences presenting occlusions and multiple interacting people. We can also see that much fewer particles are required when a good foreground detection is available: a perfect result is obtained with our projective alignment method and only 20 particles in sequences where people wear dark clothes.

If we compute the average number of valid localizations, i.e. the cases where the distance between the nearest particle and ground truth location is below 1 meter, the tracking usually looses less targets when a homographic alignment is used rather than a similarity alignment, (see Fig. 5.8a). We even reach an average of 99% of valid localizations above 1000 particles.

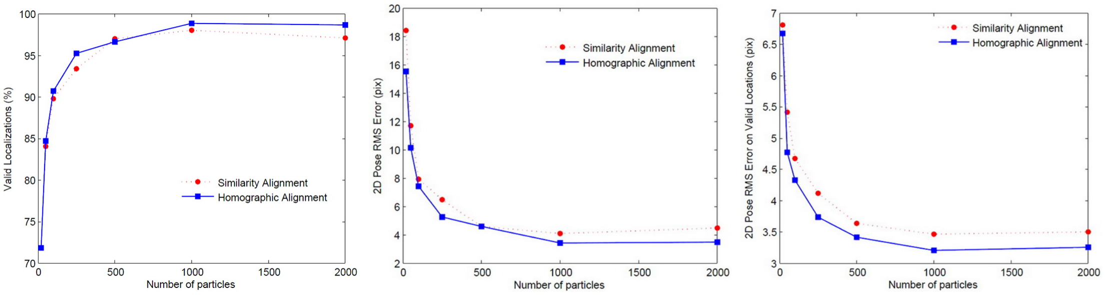
We now evaluate the pose estimation performance and compute a RMS error between the 2D pose of the best particle and the ground truth 2D pose, thus evaluating the best pose that could be estimated from the posterior independently of the employed state estimator. In Fig. 5.8b, we see how the 2D pose error (RMSE between nearest particle and ground truth) decreases

Table 5.2: Percentage of tracking failure using a similarity transformation for shape alignment. For each of the 11 selected tracks (see details in Tab. 5.1), we present the average performance over 20 runs of the tracking algorithm for different number of samples: a track is considered lost when tracking has failed during 20 frames or more (the distance between the nearest particle and ground truth location is over 1 meter) and it has not recovered by the end of the sequence.

Alignment		**Similarity**						
No. Particles		20	50	100	250	500	1000	2000
Track (no occlusion)	1	5	0	0	0	0	0	0
	2	95	85	60	5	0	0	0
	3	45	0	0	0	0	0	0
	4	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0
	6	10	0	0	0	0	0	0
Track (with occlusions)	7	65	40	25	40	30	25	20
	8	35	5	10	5	0	0	0
	9	100	70	45	30	10	5	10
	10	35	30	15	30	10	5	10
	11	20	0	0	0	0	0	0
Average		**37.27**	**21.36**	**14.09**	**9.55**	**4.55**	**3.18**	**3.64**

Table 5.3: Percentage of tracking failure using the proposed homographic projection for shape alignment. For each of the 11 selected tracks (see details in Tab. 5.1), we present the average performance over 20 runs of the tracking algorithm for different number of samples: a track is considered lost when tracking has failed during 20 frames or more (the distance between the nearest particle and ground truth location is over 1 meter) and it has not recovered by the end of the sequence.

Alignment		**Homography**						
No. Particles		20	50	100	250	500	1000	2000
Track (no occlusion)	1	0	0	0	0	0	0	0
	2	95	60	20	0	0	0	0
	3	40	5	0	0	0	0	0
	4	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0
Track (with occlusions)	7	50	45	10	0	0	0	0
	8	40	0	0	0	0	0	0
	9	95	65	45	20	15	0	0
	10	10	15	10	15	15	0	0
	11	40	0	0	0	0	0	0
Average		**33.64**	**16.82**	**7.73**	**3.18**	**2.73**	**0**	**0**



(a) (b) (c)

Figure 5.8: Tracking Results. Average Performance over 20 runs of the tracking algorithm on the 11 tracks for similarity and homographic alignment vs number of particles. (a) Percentage of valid localizations (valid if the distance between nearest particle and ground truth location is below 1 meter), (b) 2D pose error over all the poses and (c) 2D pose error computed using only valid poses from (a). The pose error in (b) and (c) is computed as the MRSE between nearest particle and ground truth using the 13 2D-joints in pixels.

with the number of particles and how the framework, again, performs better when a projective transformation is used and allows for a more accurate pose estimation. If we compute the same error using only the valid localizations (from Fig. 5.8a,) we reach lower 2D pose errors, especially for small particle sets and for the similarity based approach (see Fig. 5.8c). This makes sense because of the larger amount of failed localizations which return a bad pose estimation and influence the average pose error.

To aid the comparison of pose estimation performance and focus on the pose estimation when localisation is satisfactory, from now on, we exclude the non-valid poses and the different errors are computed over poses from valid localizations only. More frames are then considered for our homography based alignment because of its lower failure rate. We can see in Fig. 5.8c that the average 2D pose error obtained using our projective transformation is not too far from the result returned with a similarity transformation despite a qualitative improvement observed when watching the estimated poses and silhouettes.

This last observation inspired us to carry out a deeper analysis of the different results, in particular visualize the different rates in function of the distance between the subject and the camera. In Fig. 5.9, we present the average percentage of valid localizations, the average 2D pose error and the average ground plane location error varying the maximum distance to the camera. Results are presented for different sized particle sets. In the middle row, we can observe that the average pose error globally decreases as we augment the maximum distance to the camera and add new poses further away. The opposite happens with the ground plane location error. This is expected because when people move away from the camera their size in the image gets smaller. Thus, the 2D pose gets smaller when moving away from the camera leading to a consecutive lower 2D pose error while an accurate localization on the ground plane becomes more difficult with the distance. We should also point out that the different errors are computed using a ground truth data obtained from manual labelling and the accuracy and reliability of this labelling also decrease with the distance to the camera.

From Fig. 5.9, we can clearly observe that the improvement in terms of pose and ground plane localization is globally obtained when the subjects are close to the camera. This makes perfect sense since the viewpoint changes when a subject goes far away from the camera and tends to a tilt angle $\varphi = 0$ which is similar to the training viewpoint employed in this thesis. It seems that when the subject moves far away from the camera, a projective transformation is not required and a similarity transformation could be enough.

In Fig. 5.10, we present the average 2D pose error obtained when estimating the state at

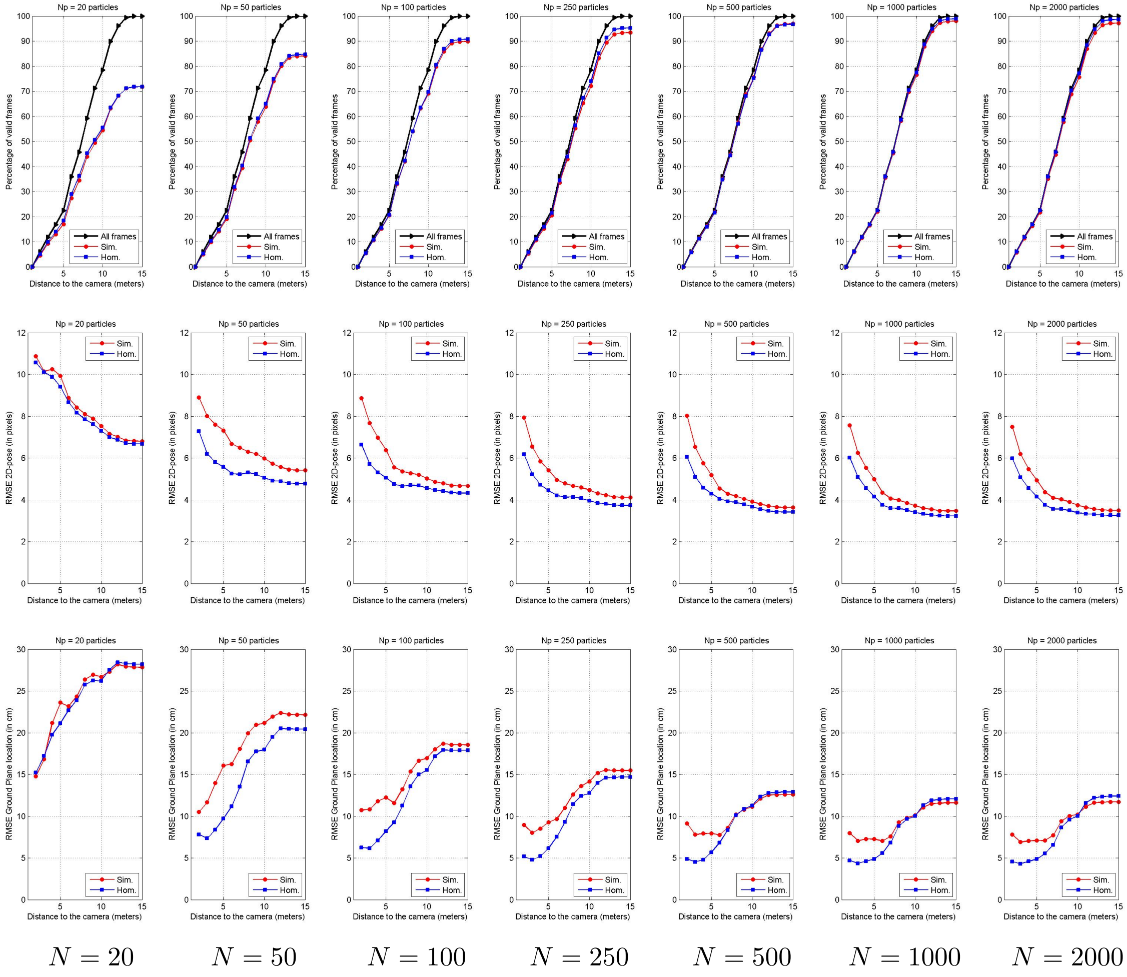


Figure 5.9: Detailed performances w.r.t. the distance to the camera: percentage of valid localizations (top row), 2D pose error (middle row) and (X, Y) ground plane localization error (bottom row) are represented (from left to right) for 20, 50, 100, 250, 500, 1000 and 2000 particles. Note that the different values are computed using the poses from 0 meter up to the given distance to the camera. Only valid localizations from the top row have been used to compute the performances in middle and bottom rows. Again 2D pose error and (X, Y) ground plane localization error are computed as the MRSE between nearest particle and ground truth using the 13 2D-joints locations in pixels and the 2D location in cm respectively.

each time step using the Monte Carlo approximation (MC), the Maximum A Posteriori (MAP) criteria, the Viterbi path finding algorithm (Viterbi) and the proposed weighted sum around the Viterbi estimate (Viterbi WS)[7]. We present the results for the two different types of alignment. Corresponding numerical evaluation is given in Tab. 5.4. The first observation is that our homographic alignment clearly outperforms the similarity transformation independently of the employed state estimation technique. We can also observe that our proposed approach for state estimation outperforms all the other techniques for $N \geq 500$ particles while MC estimate is better for smaller particle sets.

[7] Note that the state estimate used to model each subject's 3D occupancy on the ground floor in Eq. 5.22 is always computed using the Monte Carlo approximation as we want to compare the different state estimators from the same clouds of samples.

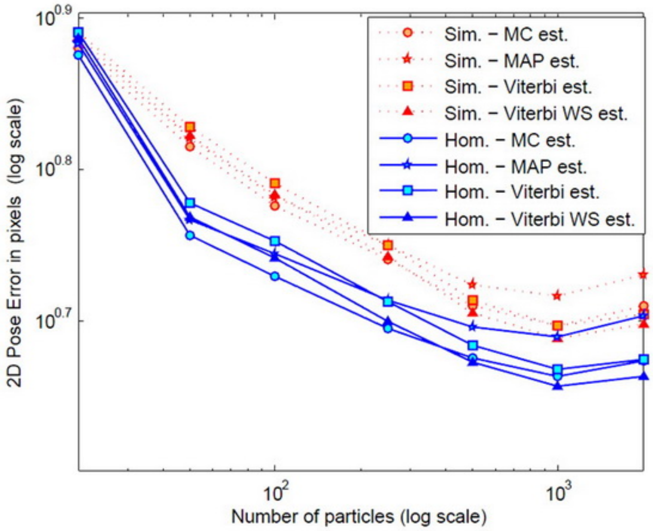


Figure 5.10: Average 2D pose error varying the state estimator: performance over 20 runs of the tracking algorithm on the 11 tracks for homographic and similarity alignments. We present the RMS 2D pose error when estimating the state at each time step using the Monte Carlo approximation (MC), the MAP criteria, Viterbi path finding algorithm (Viterbi) and the weighted sum around the Viterbi estimate (Viterbi WS).

Table 5.4: Average 2D pose error: performance over 20 runs of the tracking algorithm on the 11 tracks for similarity and homographic alignments. We present the RMS 2D pose error (in pixels) when estimating the state at each time step using the Monte Carlo approximation $\widehat{\chi_t}^{MC}$, the MAP criteria $\widehat{\chi_t}^{MAP}$, Viterbi path finding algorithm $\widehat{\chi_t}^{Vit}$ and the weighted sum around the Viterbi estimate $\widehat{\chi_t}^{Vit+WS}$.

Alignment		**Similarity**						
No. Particles		20	50	100	250	500	1000	2000
State	$\widehat{\chi_t}^{MC}$	**7.59**	**6.53**	**5.97**	**5.50**	5.13	4.98	5.14
Estimator	$\widehat{\chi_t}^{MAP}$	7.66	6.62	6.05	5.63	5.30	5.21	5.38
	$\widehat{\chi_t}^{Vit}$	7.77	6.74	6.18	5.63	5.17	4.98	5.07
	$\widehat{\chi_t}^{Vit+WS}$	7.72	6.65	6.08	5.53	**5.08**	**4.89**	**5.00**
Alignment		**Homography**						
No. Particles		20	50	100	250	500	1000	2000
State	$\widehat{\chi_t}^{MC}$	**7.52**	**5.71**	**5.37**	**4.96**	4.75	4.61	4.72
Estimator	$\widehat{\chi_t}^{MAP}$	7.65	5.85	5.55	5.17	4.97	4.90	5.05
	$\widehat{\chi_t}^{Vit}$	7.78	6.00	5.66	5.16	4.84	4.67	4.73
	$\widehat{\chi_t}^{Vit+WS}$	7.70	5.88	5.52	5.02	**4.71**	**4.54**	**4.61**

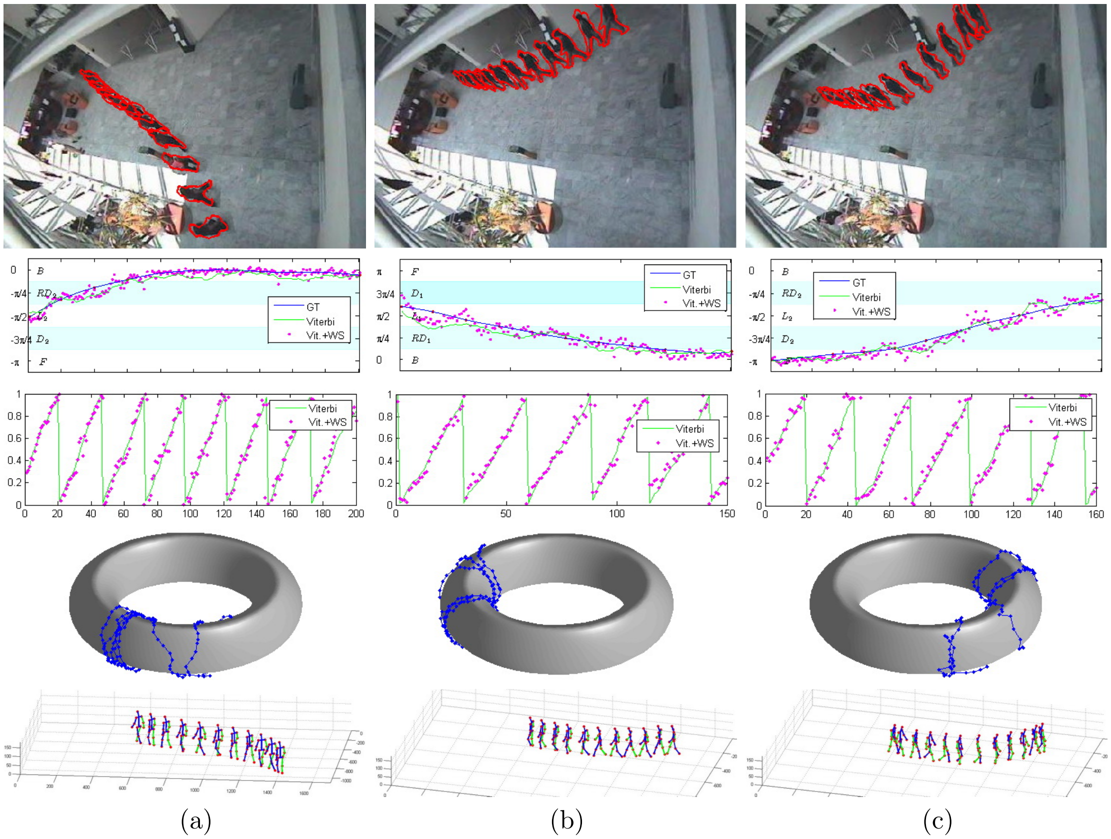
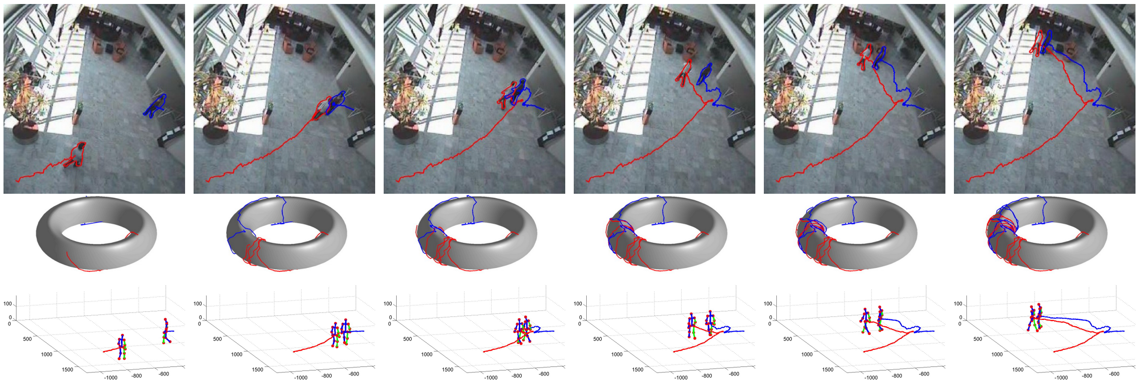


Figure 5.11: Qualitative 3D pose tracking results for tracks 1 (a), 4 (b) and 5 (c) in Tab. 5.1 using our projective method for view-invariant pose tracking and 500 particles. For each sequence, we show from top to bottom: the tracked silhouettes for a few selected frames (same frames considered in Fig4.8), the estimated viewpoint θ (vs ground truth), the estimated gait parameter μ, the trajectory of the subject on the torus manifold (μ and θ together) and the estimated 3D poses corresponding to the silhouette in the first row.

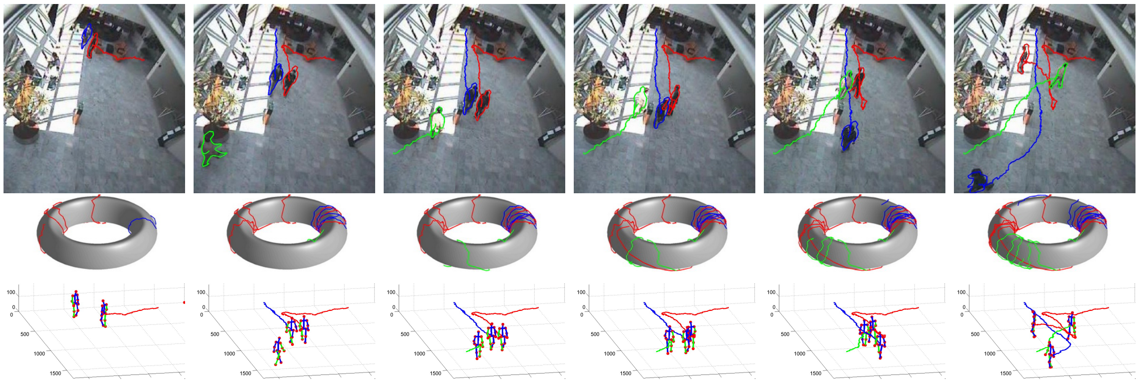
In Fig. 5.11, we present qualitative results for tracks 1, 4 and 5 from Tab. 5.1 using our projective method for view-invariant pose tracking and 500 particles. Note that the same sequences were considered in Fig. 4.8. For each sequence, we can observe the tracked silhouettes for a few frames and the trajectory of the subject in the image as well as the trajectory on the torus manifold and the estimated 3D poses which have been successfully tracked. If we look at the temporal evolution of the viewpoint, we can see that using the Viterbi algorithm and our approach, we achieve a smooth continuous estimation of the viewpoint angle θ while using a model constructed from a discrete set of training views. As in [Elgammal and Lee, 2009], we recover the typical sawtooth curve of the walking cycle but in our case with challenging perspective videos.

We present more qualitative results for 2 sequences with multiple interacting subjects in Fig. 5.12 (tracks 7 and 8) and Fig. 5.13 (tracks 9, 10 and 11) using our proposed method and 1000 particles. For each sequence, we show the result for a few frames: the tracked silhouettes and the trajectories in the image, the trajectories on the torus manifold and the estimated 3D poses which have been successfully tracked despite the occlusions and the perspective effect.



(Frames 40) (120) (190) (230) (270) (310)

Figure 5.12: Qualitative 3D pose tracking results for the *Meet_WalkTogether2* sequence (tracks 7 and 8 in Tab. 5.1) with 2 interacting subjects using our projective method for view-invariant pose tracking and 1000 particles. from top to bottom we show: the tracked silhouettes and the trajectories in the image, the trajectories on the torus manifold and the estimated 3D poses.



(Frames 150) (260) (300) (330) (365) (415)

Figure 5.13: Qualitative 3D pose tracking results for the *Meet_Split_3rdGuy* sequence (tracks 9, 10 and 11 in Tab. 5.1) with 3 interacting subjects using our projective method for view-invariant pose tracking and 1000 particles. from top to bottom we show: the tracked silhouettes and the trajectories in the image, the trajectories on the torus manifold and the estimated 3D poses.

5.5 Conclusions

In this chapter, we have presented a complete framework for view invariant shape based 3D body pose tracking in man-made environments from monocular surveillance videos with high perspective effect. We have assumed that the camera is calibrated w.r.t. the scene and that observed people move on a known ground plane, which are realistic assumptions in surveillance scenarios. We have demonstrated that exploiting projective geometry alleviates the problems caused by roof-top cameras with high tilt angles, and have shown that using a mapping from a low dimensional pose manifold to 8 training views was enough to produce acceptable results when using a projective alignment for silhouette matching: our framework is able to track 3D human walking poses in a 3D environment exploring only a 4 dimensional state space with a particle filter.

We have conducted a series of experiments to quantitatively and qualitatively evaluate our tracking framework for a wide variety of viewing angles and a variety of sequences, some with multiple interacting subjects and occlusions. In our experimental evaluation, we have demonstrated the significant improvements of the proposed projective alignment over a commonly used similarity alignment and have provided numerical pose tracking results for the monocular sequences with perspective effect from the CAVIAR dataset. Our results demonstrate that the incorporation of this perspective correction in the pose tracking framework results in a higher tracking rate and allows for a better estimation of body poses under wide viewpoint variations.

Part III

Pose Estimation with a Moving Camera or in Static Images

6

Multi-class Pose Classifier

6.1 Introduction

The third part of this thesis considers the problem of human detection and pose estimation in the most difficult scenario: an isolated static image with no prior information on the structure of the scene or the number of subjects. By extension, the solutions to that problem are also valid for moving camera sequences, as each frame can be treated as an isolated static image.

Given an input image, an ideal system would be able to localize any humans present in the scene and recover their poses. The two stages, known as *human detection* and *human pose estimation*, are usually considered separately. There is an extensive literature on both *detection* [Viola et al., 2005, Wu et al., 2005, Dalal and Triggs, 2005, Zhu et al., 2006, Gavrila, 2007, Sabzmeydani and Mori, 2007] and *pose estimation* [Shakhnarovich et al., 2003, Agarwal and Triggs, 2006, Mori and Malik, 2006, Thayananthan et al., 2006, Bissacco et al., 2007, Jaeggli et al., 2009, Elgammal and Lee, 2009, Lee and Elgammal, 2010] but relatively few papers consider the two stages together [Dimitrijevic et al., 2006, Bissacco et al., 2006, Sminchisescu et al., 2006, Okada and Soatto, 2008, Bourdev and Malik, 2009]. Most algorithms for pose estimation assume that the human has been localized and the silhouette has been recovered, making the problem substantially easier.

We tackle the problem of simultaneous human detection and pose estimation. We follow a sliding window approach to jointly localize and classify human pose using a multi-class classifier. Such classifier needs to be very fast as it will have to classify thousands of windows for each processed image (considering multiple locations and scales in the image). In this chapter, we propose a fast multi-class classifier that combines the best components of state-of-the-art classifiers including hierarchical trees, cascades of rejectors and randomized forests.

6.1.1 Related Previous Work

Much work focuses on human detection specifically without considering pose [Dalal and Triggs, 2005, Gavrila, 2007, Zhu et al., 2006, Viola et al., 2005, Wu et al., 2005, Sabzmeydani and Mori, 2007]. Dalal and Triggs [Dalal and Triggs, 2005] use a dense grid of Histograms of Orientated Gradients (HOG) and learn a Support Vector Machine (SVM) classifier to separate human from background examples. Later Zhu et al. [2006] extend this work by applying integral histograms to efficiently calculate HOG features and use a cascade of rejectors classifier to achieve near real time detection performance.

Several works attempt to combine localization and pose estimation. Dimitrijevic et al. [2006] present a template-based pose detector and solve the problem of huge datasets by detecting only human silhouettes in a characteristic postures (sideways opened-leg walking postures in this case). They extend this work in [Fossati et al., 2007] by inferring 3D poses between consecutive detections using motion models. This method can be used to track walking people even with moving cameras, however, it seems somehow difficult to generalize to any actions that do not exhibit characteristic posture. Hofmann and Gavrila [2012] propose to perform 3D pose detection for several cameras independently and fuse information at the pose parameter level by means of an efficient multi-stage recovery process. Sminchisescu et al. [2006] jointly learn coupled generative-discriminative models in alternation and integrate detection and pose estimation in a common sliding window framework. Okada and Soatto [2008] learn k kernel SVMs to discriminate between k predefined pose clusters, and then learn linear regressors from feature to pose space. They extend this method to localization by adding an additional cluster that contains only images of background.

We introduce a novel algorithm that jointly tackles human detection and pose estimation in a similar way to template tree approaches [Gavrila, 2007, Stenger, 2004], while exploiting some advantages of AdaBoost style cascade classifiers [Viola and Jones, 2004, Zhu et al., 2006] and Random Forests [Breiman, 2001]. Random Forests (RF) have seen a great deal of success in many varied applications such as object recognition [Bosch et al., 2007] or clustering [Moosmann et al., 2008, Shotton et al., 2008]. RF have shown to be fast and robust classification techniques that can handle multi-class problems [Lepetit and Fua, 2006], so makes them ideal for use in human pose estimation. Recently, Shotton et al. [2011] trained a decision forest to estimate body parts from depth images with excellent results on pose estimation.

6.1.2 Motivation and Overview of the Approach

Many different types of features have been considered for human detection and pose estimation: silhouette [Agarwal and Triggs, 2006], shape [Gavrila, 2007], edges [Dimitrijevic et al., 2006], HOG descriptors [Dalal and Triggs, 2005, Zhu et al., 2006, Felzenszwalb et al., 2010a], Haar filters [Viola et al., 2005], motion and appearance patches [Bissacco et al., 2007], edgelet feature [Wu et al., 2005], shapelet features [Sabzmeydani and Mori, 2007] or SIFT [Lowe, 2004]. Driven by the recent success of HOG descriptors for both human detection [Dalal and Triggs, 2005, Zhu et al., 2006] and pose estimation [Shakhnarovich et al., 2003], and that they can be implemented efficiently to achieve near real time speeds [Zhu et al., 2006], we chose to use HOG descriptors as a feature in our algorithm. For pose estimation, an ideal dataset should contain variation in subject pose, camera viewpoint, appearance and physical attributes. Combining the dataset with a very dense image feature set such as HOG captures discriminative details between very similar poses [Okada and Soatto, 2008] but also considerably increases the dimension of the training set.

Random Forests [Breiman, 2001] are inherently good for multi-class problems, so makes them ideal for use in pose estimation. They allow for a better handling of large datasets as they can be faster to train and are less prone to over-fitting than selecting features from an exhaustive search over all features[1]. We performed an initial test of pose classification (see Fig. 6.1 and Sect. 6.5) and identified two main drawbacks with the algorithm and the existing implementation: as illustrated in Fig. 6.1 using denser HOG feature grids improves pose classification accuracy. Neighboring classes can be very close to one another in image space, and in practice are only separable by some sparse subset of features. This means that,

[1] RF are grown by randomly selecting a subset of features at each node of the tree to help avoid a single tree over fitting the training data. The best split is found for each dimension m_i by evaluating all possible splits along that dimension using a measure such as information gain [Bosch et al., 2007]. The dimension m^* that best splits the data according to that score is used to partition the data at that node. This process continues recursively until all the data has been split and each node contains a single class of data.

having randomly picked an arbitrary feature to project on from a high dimensional feature space, it is highly unlikely that an informative split in this projection (i.e. one that improves the information measure) exists. While we do not need perfect trees, informative trees are still rare and finding them naively requires us to generate an infeasible number of trees.

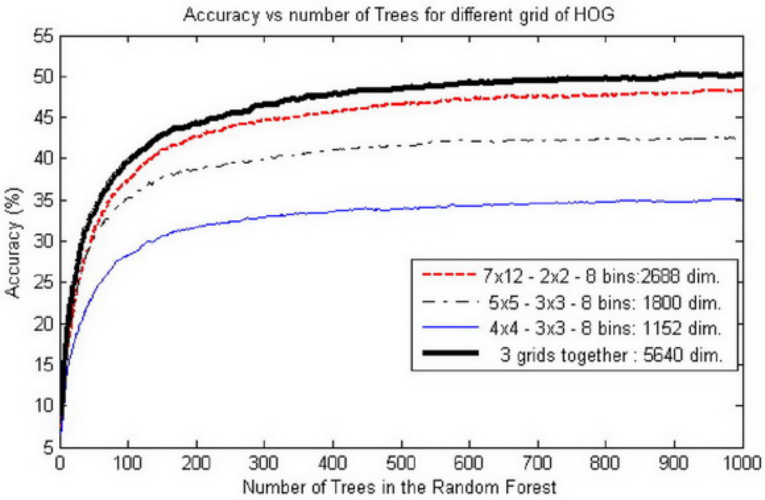


Figure 6.1: Random Forest preliminary results. An initial test was performed on the MoBo walking dataset [Gross and Shi, 2001]: dense grid of HOG features are extracted for 15 different subjects from around 50,000 images which are grouped in 64 pose classes. We build the training subset by randomly sampling 10 subjects and keep the remaining 5 subjects for testing. We run the same test for 3 different grids of HOG and show the classification results varying the number of trees used in the forest. Using denser HOG grids improves pose classification accuracy but we are quickly facing memory issues that prevent us from working with denser grids.

Another drawback of the Random Forests algorithm is that it is not very well adapted for sliding window approaches. Even if on-demand feature extraction can be considered as in [Deselaers et al., 2007], for each scanned sub-image, the trees still have to be completely traversed to produce a vote/classification. This means that a non-negligible amount of features have to be extracted for each processed window, making the algorithm less efficient than existing approaches like cascades-of-rejectors that quickly reject most of the negative candidates using a very small subset of features. Works such as [Viola and Jones, 2004, Zhu et al., 2006] use AdaBoost to learn a cascade structure using very few features at the first level of the cascade, and increasing the number of features used for later stages. Other approaches such as those described in [Ma and Ding, 2005, Zhang et al., 2002] for multi-view face detection, organize the cascade in to a hierarchy structure consisting of two types of classifier; face/non-face, and face view detection. Zhang et al. [2007a] present a probabilistic boosting network for joint real-time object detection and pose estimation. Their graph structured network also alternates binary foreground/background and multi-class pose classifiers.

Inspired by these ideas, we train multi-class hierarchical cascades for human pose detection. First a class hierarchy is built by recursively clustering and merging the predefined pose classes. Hierarchical template trees have been shown to be very effective for real time systems [Gavrila, 2007, Stenger, 2004], and we extend this approach to non-segmented images. For each branch of this tree-like structure, we use a novel algorithm to build a list of potentially discriminative HOG descriptor blocks. We then train a weak classifier on each one of these blocks and select the ones that show the best performances. We finally grow an ensemble of cascades by randomly sampling one of these HOG-based rejectors at each branch of the hierarchy. By randomly sampling the features, each cascade uses different sets of features to vote and adds some robustness to noise and prevents over fitting as with RF. Each cascade can vote for one or more class so the final classification is a distribution over classes.

In the next section, we present a new method for data driven discriminative feature selection that enables our proposal to deal with large datasets and high dimensional feature spaces. Next, we extend hierarchical template tree approaches [Gavrila, 2007, Stenger, 2004] to unsegmented images. Finally, we explain how we use random feature selection inspired by Random Forests to build an ensemble of multi-class cascade classifiers.

6.2 Sampling of Discriminative HOGs

Feature selection is probably the key point in most recognition problems. It is very important to select the relevant and most informative features in order to alleviate the effects of the *curse of dimensionality*. Many different types of features are used in general recognition problems. However, only a few of them are useful for exemplar-based pose estimation. For example, features like color and texture are very informative in general recognition problems, but because of their variation due to clothing and lighting conditions, they are seldom useful in exemplar-based pose estimation. On the other hand, gradients and edges are more robust cues with respect to clothing and lighting variations[2]. Guided by their success for both human detection [Dalal and Triggs, 2005, Zhu et al., 2006] and pose estimation [Shakhnarovich et al., 2003] problems, we chose to use HOG descriptors as a feature in our algorithm.

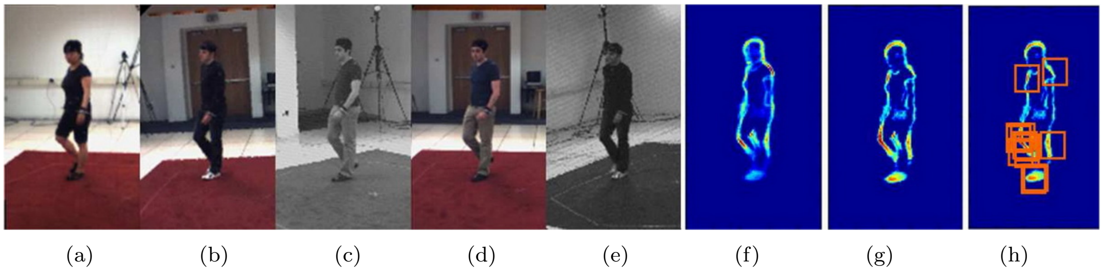


Figure 6.2: Log-likelihood ratio for human pose. (**a** to **e**): examples of aligned images belonging to the same class (for different subjects and cameras) as defined in Sect. 6.4.1.1 using HumanEVA dataset [Sigal et al., 2010]. Resulting gradient probability map $p(E|B)$ (**f**) and log-likelihood ratio $L(B, C)$ (**g**) for this same class vs all the other classes. Hot colors in (**g**) indicate the discriminative areas. The sampled HOG blocks are represented on top of the likelihood map in (**h**).

Each HOG block represents the probability distribution of gradient orientation (quantized into a predefined number of histogram bins) over a specific rectangular neighborhood. The usage of HOGs over the entire training image, usually in a grid, leads to a very large feature vector where all the individual HOG blocks are concatenated. So an important question is how to select the most informative blocks in the feature vector. Some works have addressed this question for human detection and pose estimation problems using SVMs or RVMs [Dalal and Triggs, 2005, Zhu et al., 2006, Bissacco et al., 2006, Okada and Soatto, 2008]. However, such learning methods are computationally inefficient for very large datasets.

AdaBoost is often used for discriminative feature selection such as in [Zehnder et al., 2005, Villamizar et al., 2009]. Instead of an exhaustive search over all possible features, or uniformly sampling a random subset from these features, we introduce a guided sampling scheme based

[2] Clothing could still be a problem if there are very few subjects in the training set: some edges due to clothing could be considered as discriminative edges when they should not.

on log-likelihood gradient distribution. Collins and Liu [2003] also use a log-likelihood ratio based approach in the context of adaptive on-line tracking, and select the most discriminative features from a set of 49 RGB features that separate a foreground object from the background. In our case, we have a much higher set of possible features (histogram bins) if we consider all the possible configurations of HOG blocks at all locations exhaustively. So to make the problem more tractable, rather than selecting individual features in a dense HOG vector, entire HOG blocks are randomly selected using a log-likelihood ratio derived from the edge gradients of the training data. Dimitrijevic et al. [2006] use statistical learning techniques during the training phase to estimate and store the relevance of the different silhouette parts to the recognition task. We use a similar idea to learn relevant gradient features, although slightly different because of the absence of silhouette information. In what follows, we present our method to select the most discriminative and informative HOG blocks for human pose classification. The basic idea is to take advantage of accurate image alignment and study gradient distribution over the entire training set to favor locations that we expect to be more discriminative between different classes. Intra-class and inter-class probability density maps of gradient/edge distribution are used to select the best location for the HOG blocks.

6.2.1 Formulation

Here we describe a simple Bayesian formulation to compute the log-likelihood ratios which can be used to determine the importance of different regions in the image when discriminating between different classes. Given a set of classes C, the probability that the classes represented by C could be explained by the observed edges E can be defined using a simple Bayes rule:

$$p(C|E) = \frac{p(E|C)p(C)}{p(E)}. \tag{6.1}$$

The likelihood term $p(E|C)$ of the edges being observed given classes C, can be estimated using the training data edges for the respective classes. Let $T = \{(I_i, c_i)\}$ be a set of aligned training images, each with a corresponding class label. Let $T_C = \{(I, c) \in T \mid c \in C\}$ be the set of training instances for the set of classes $C = \{c_i\}$. Then the likelihood of observing an edge given a set of classes C can be estimated as follows:

$$p(E|C) = \frac{1}{|T_C|} \sum_{(I,c)\in T_C} \nabla(I), \tag{6.2}$$

where $\nabla(\cdot)$ calculates a normalized oriented gradient edge map for a given image I, with the value at any point being in the range $[0, 1]$. Note that an accurate alignment of the positive samples is required to compute $p(E|C)$. We refer the reader to the automatic alignment method proposed for human pose in Sect. 6.4.1.1 for a solution to this problem. Class specific information is represented by high values of $p(E|C)$ from locations where edge gradients occur most frequently across the training instances. Edge gradients at locations that occur in only a few training instances (e.g. due to background or appearance) will tend to average out to low values. To increase robustness toward background noise the likelihood can be thresholded by a lower bound:

$$p(E|C) = \begin{cases} p(E|C) & \text{if } p(E|C) > \tau \ , \\ 0 & \text{otherwise.} \end{cases} \tag{6.3}$$

Suppose we have a subset of classes $B \subset C$. Discriminative edge gradients will be those that are strong across the instances within B but are not common across the instances within C.

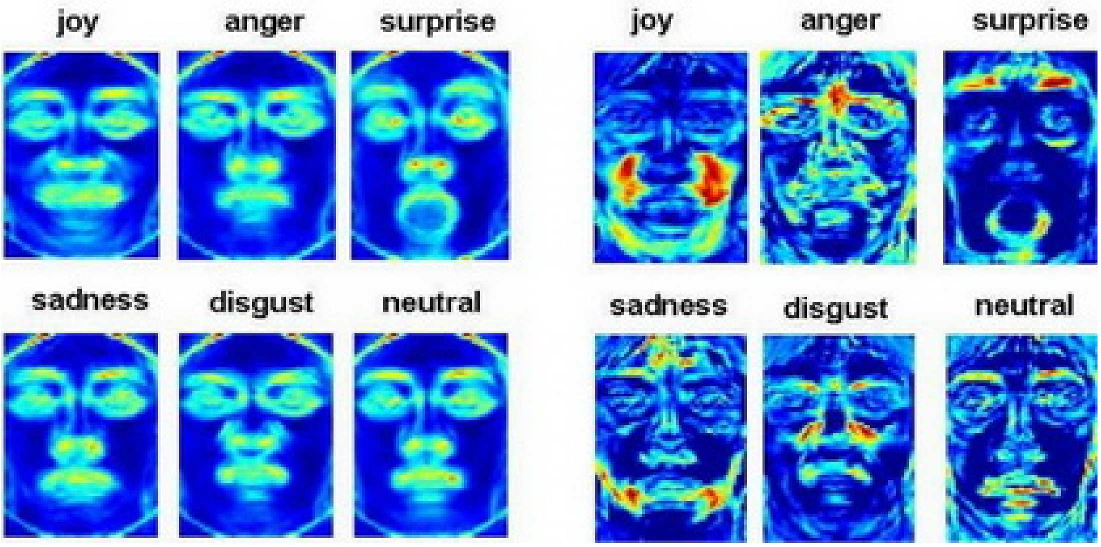


Figure 6.3: Log-likelihood ratio for face expressions. We used a subset of [Kanade et al., 2000] composed by 20 different individuals acting 5 basic emotions besides the neutral face: joy, anger, surprise, sadness and disgust. All images were normalized, i.e. cropped and manually rectified. Gradient probability map (*left*) and log-likelihood ratio (*right*) are represented for each one of the 6 classes. Hot colors indicate the discriminative areas for a given facial expression.

Using the log-likelihood ratio between the two likelihoods $p(E|B)$ and $p(E|C)$ gives:

$$L(B,C) = \log\left(\frac{p(E|B)}{p(E|C)}\right). \tag{6.4}$$

The log-likelihood distribution defines a gradient prior for the subset B. High values in this function give an indication of where informative gradient features may be located to discriminate between instances belonging to subset B and the rest of classes in C. For the example given in Fig. 6.2, we can see how the right knee is a very discriminative region for this particular class (see Fig. 6.2g). In Fig. 6.3, we present the log-likelihood distributions for 5 different facial expressions.

Gradient orientation can be included by decomposing the gradient map into n_θ separate orientation channels according to gradient orientation. The log-likelihood $L_\theta(B,C)$ is then computed separately for each channel, thereby increasing the discriminatory power of the likelihood function, especially in cases when there are many noisy edge points present in the images. Maximizing over the n_θ orientation channels, the log-likelihood gradient distribution for class B then becomes[3]:

$$\mathcal{L}(B,C) = \max_\theta \left(L_\theta(B,C)\right). \tag{6.5}$$

We also obtain the corresponding orientation map:

$$\Theta(B,C) = \operatorname*{argmax}_\theta \left(L_\theta(B,C)\right). \tag{6.6}$$

Uninformative edges from a varied dataset will generally be present for only a few instances and not be common across instances from the same class, whereas common informative edges

[3] $\mathcal{L}(B,C) = L(B,C)$ if no separated orientation channels are considered.

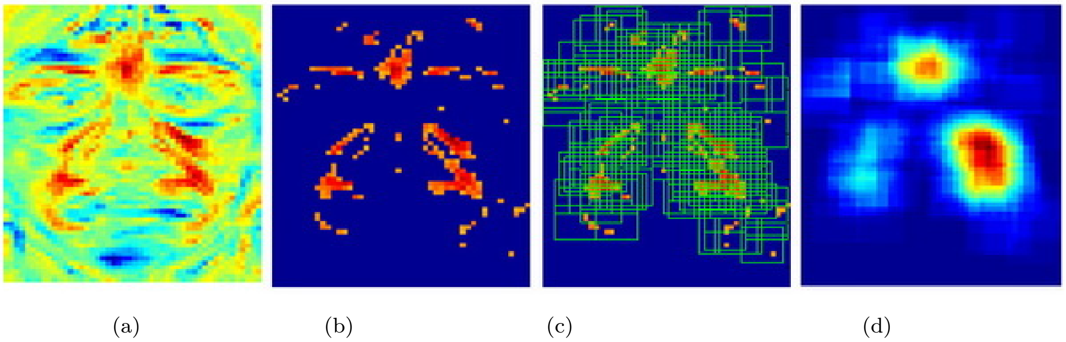


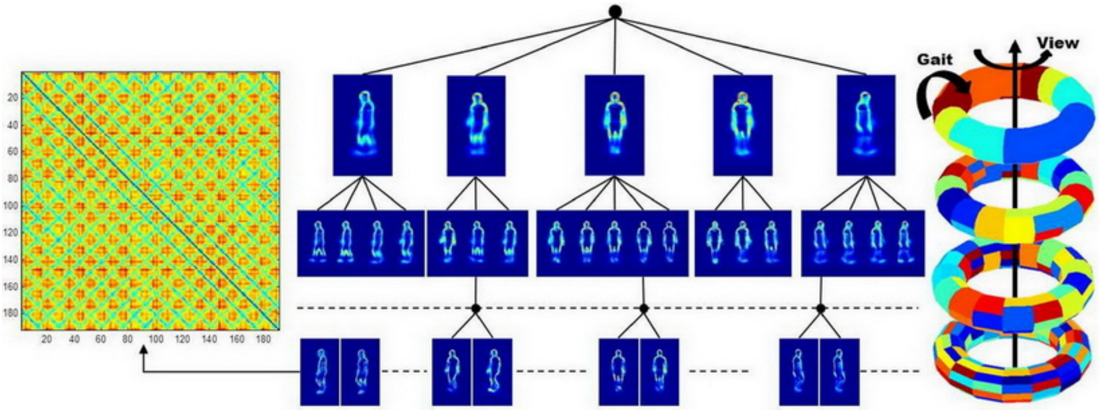
Figure 6.4: Selection of discriminative HOG for facial expressions classification: details of the database are given in Fig. 6.3. In (a), we show the log-likelihood map for 2 facial expressions together: joy and disgust. After thresholding (b), we obtain the most discriminative areas where HOG blocks will be extracted (c) to differentiate joy and disgust expressions from the other classes. In (d), we can see the areas covered by the sampled HOG blocks and how the resulting density follows the distribution from (b). Results using this feature selection scheme on facial expression recognition have been reported in [Orrite et al., 2009].

for pose will be reinforced across instances belonging to a subset of classes B, and be easier to discriminate from edges that are common between B and all classes in parent set C. Even if background edges were shared by many images (e.g. if positive training samples consist of images of standing actions shot by stationary cameras such as the gesture and the box actions in the HumanEVA dataset [Sigal et al., 2010]), then these edges become uninteresting as $p(E|B) \approx p(E|C)$ and a low log-likelihood value would be returned. However, when uninformative edges (from background or clothing) are not common across all training instances but occur in the images of the same class all the time by coincidence, edges can be falsely considered as potentially discriminative. We observed that particular case when trying to work with the Buffy dataset from [Ferrari et al., 2008]: all the images for the same pose (standing with arms folded) correspond to one unique character with the same clothes and same background scene.

To address this issue, we create a varied dataset of instances which is discussed more detail in Sec.6.5. Given this log-likelihood gradient distribution $\mathcal{L}(B,C)$, we can randomly sample HOG blocks from positions (x,y) where they are expected to be informative, thus reducing the dimension of the feature space. We then use $\mathcal{L}(B,C)$ as distribution proposal to drive blocks sampling $(x^{(i)},y^{(i)})$:

$$(x^{(i)},y^{(i)}) \sim \mathcal{L}(B,C). \tag{6.7}$$

Features are then extracted from areas of high gradient probability across our training set more than areas with low probability (see Fig. 6.2h). By using this information to sample features, the amount of useful information available to learn efficient classifiers is increased. In Fig. 6.4a we represent the log-likelihood for 2 facial expressions together: joy and disgust. After thresholding (Fig. 6.4b), we obtain the most significant areas where HOG blocks will be extracted (Fig. 6.4c) to differentiate joy and disgust expressions from the other classes. In Fig. 6.4d, we can see the areas covered by the selected HOG blocks and how the resulting density follows the distribution from Fig. 6.4b. Results using this feature selection scheme for facial expression recognition have been reported in [Orrite et al., 2009].



(a) (b) (c)

Figure 6.5: Bottom-up hierarchical tree construction: the structure $\mathcal{S}$ is built using a bottom-up approach by recursively clustering and merging the classes at each level. We present an example of tree construction from 192 classes (see class definition in Sect. 6.4.1.2) on a torus manifold where the dimensions represent gait cycle and camera viewpoint. The matrix presented here (a) is built from the initial 192 classes and used to merge the classes at the very lowest level of the tree. The similarity matrix is then recomputed at each level of the tree with the resulting new classes. The resulting hierarchical tree-like structure $\mathcal{S}$ is shown in (b) while the merging process on the torus manifold is depicted in (c). We can observe (b) how the first initial node acts as a viewpoint classifier.

6.3 Randomized Cascades of Rejectors

The classifier is an ensemble of hierarchical cascade classifiers. The method takes inspiration from cascade approaches such as [Viola and Jones, 2004, Zhu et al., 2006], hierarchical template trees such as [Gavrila, 2007, Stenger, 2004] and Random Forests [Breiman, 2001, Lepetit and Fua, 2006, Bosch et al., 2007].

6.3.1 Bottom-up Hierarchical Tree Construction

Tree structures are a very effective way to deal with large exemplar sets. Gavrila [2007] constructs hierarchical template trees using human shape exemplars and the chamfer distance between them. He recursively clusters together similar shape templates selecting at each node a single cluster prototype along with a chamfer similarity threshold calculated from all the templates that the cluster contains. Multiple branches can be explored if edges from a query image are considered to be similar to cluster exemplars for more than one branch in the tree. Stenger [2004] follows a similar approach for hierarchical template tree construction applied to articulated hand tracking, the main difference being that the tree is constructed by partitioning the state space. This state space includes pose parameters and viewpoint. Inspired by these two papers, Okada and Stenger [2008] present a method for human motion capture based on tree-based filtering using a hierarchy of body poses found by clustering the silhouette shapes. Although these existing template tree techniques are shown to have interesting qualities in terms of speed, they present some important drawbacks for the task we want to achieve. First, templates need to be stored for each node of the tree leading to memory issues when dealing with large sets of templates. The second limitation is that they require that a clean silhouette or template data is available from manual segmentation [Gavrila, 2007] or generated synthetically from a 3D model [Stenger, 2004, Okada and Stenger, 2008]. Their methodology can not be directly applied to unsegmented image frames because of the presence of too many noisy edges from background and clothing of the individuals which dominate informative pose-related edges. By using only the silhouette outlines as image features, the approaches in [Gavrila, 2007, Okada and Stenger, 2008] ignore the non-negligible amount of information contained in the *internal edges* which are very informative for pose estimation applications. We thus propose a solution to adapt the construction of such a hierarchical tree structure for images.

Algorithm 6: Class Hierarchy Construction.

```
input  : Labeled training images.
output: Hierarchical structure $\mathcal{S}$.
while num. of classes of the level $n_l > 1$ do
    for each class $n$ do
        Compute new $\mathcal{L}(C_n, C)$ (cf. § 6.2);
    Compute the $n_l \times n_l$ similarity matrix $M$ (cf. Eq. 6.8);
    Set the number of merged classes $n_m = 0$ ;
    Set the number of clusters $n_{cl} = 0$ ;
    while $n_m < n_l$ do
        Take the next 2 closest classes $(C_1, C_2)$ in $M$;
            • case 1: $C_1$ and $C_2$ have not been merged yet.
              Create a new cluster $\mathcal{C}_{n_{cl}+1}$ with $C_1$ and $C_2$: $\mathcal{C}_{n_{cl}+1} = C_1 \cup C_2$;
              Update $n_{cl} = n_{cl} + 1$ and $n_m = n_m + 2$;
            • case 2: $C_1$ and $C_2$ already merged together.
              Do nothing;
            • case 3: $C_1$ has already been merged in $\mathcal{C}_r$.
              Merge $C_2$ in $\mathcal{C}_r$: $\mathcal{C}'_r = \mathcal{C}_r \cup C_2$;
              $n_m = n_m + 1$;
            • case 4: $C_2$ has already been merged in $\mathcal{C}_s$.
              Merge $C_1$ in $\mathcal{C}_s$: $\mathcal{C}'_s = \mathcal{C}_s \cup C_1$;
              $n_m = n_m + 1$;
            • case 5: $C_1 \in \mathcal{C}_r$ and $C_2 \in \mathcal{C}_s$.
              Merge the 2 clusters $\mathcal{C}_r$ and $\mathcal{C}_s$: $\mathcal{C}'_r = \mathcal{C}_r \cup \mathcal{C}_s$ ;
              $n_{cl} = n_{cl} - 1$;
    Create a new level $l$ with new hyper-classes $\{C'_n\}_{n=1}^{n_{cl}}$ ;
    Update the number of classes for that level $n_l = n_{cl}$;
    Update the structure $\mathcal{S}' = \mathcal{S} \cup \{C'_n\}_{n=1}^{n_l}$ ;
```

Instead of successively partitioning the state space at each level of the tree [Stenger, 2004] or clustering together similar shape templates from bottom-up [Gavrila, 2007] or top-down [Okada and Stenger, 2008], we propose a hybrid algorithm. Given that a parametric model of the human pose is available (3D or 2D joint locations), we first partition the state space into a series of classes[4]. Then, we construct the hierarchical tree structure by merging similar classes in a bottom-up manner as in [Gavrila, 2007] but using for each class its gradient map. This process only requires that the cropped training images have been aligned without the need for clean silhouettes or templates.

We thus recursively cluster and merge similar classes based on a similarity matrix that is recomputed at each level of the tree (Fig. 6.5a). The similarity matrix $M = \{M_{i,j}\}$ with $i, j \in \{1, \cdots, n_l\}$ (being n_l the number of classes at each level) is computed using the L2 distance between the log-likelihood ratios of the set of classes C_n that represent the classes that fall below each node n of the current level and the global edge map C constructed from all the classes together:

$$M_{i,j} = ||\mathcal{L}(C_i, C) - \mathcal{L}(C_j, C)||. \tag{6.8}$$

Using the log-likelihood ratio to merge classes reduces the effect of uninformative edges on the hierarchy construction while at the same time increasing the influence of discriminative edges. At each level, classes are clustered by taking the values from the similarity matrix in ascending order and successively merge corresponding classes until they all get merged, then go to next level. The class hierarchy construction is depicted in Algorithm 6. This algorithm is fully automatic as it does not require any threshold to be tuned, and works well with continuous and symmetrical pose spaces like the ones considered in this thesis. However, we have observed that it fails with non-homogeneous training data or in presence of outliers. In that case, the use of a threshold on the similarity value in the second while loop should be considered to stop the clustering process before merging together classes which are too different. This process leads to a *hierarchical* structure $\mathcal{S}$ (Fig. 6.5b). The leaves of this tree-like structure $\mathcal{S}$ define the partition of the state space while $\mathcal{S}$ is constructed in the feature space: the similarity in term of image features between compared classes increases and the classification gets more difficult when going down the tree and reaching lower levels as in [Gavrila, 2007]. But in our case each leaf represents a cluster in the state space as in [Stenger, 2004] while in [Gavrila, 2007], templates corresponding to completely different poses can end-up being merged in the same class making the regression to a pose difficult or even impossible. In [Gavrila, 2007] the number of branches are selected before growing the tree, potentially forcing dissimilar templates to merge too early. while in our case, each node in the final hierarchy can have 2 or more branches as in [Stenger, 2004, Okada and Stenger, 2008].

Instead of storing and matching an entire template prototype at each node as in [Gavrila, 2007, Stenger, 2004, Okada and Stenger, 2008], we now propose a method to build a reduced list of discriminative HOG features, thus making the approach more scalable to the challenging size and complexity of human pose datasets.

6.3.2 Discriminative HOG Blocks Selection

While other algorithms (PSH, RVMs, SVMs, etc) must extract the entire feature space for all training instances during learning, making them less practical when dealing with very large datasets, our method for learning hierarchical cascades only selects a small set of discriminative features extracted from a small subset of the training instances at a time. This makes it much more scalable for very large training sets.

For each branch of our structure $\mathcal{S}$, we use our algorithm for feature selection to build a list of potentially discriminative features, in our case a vector of HOG descriptors. HOG blocks need only to be placed near areas of high edge probability for a particular class. Feature sampling will then be concentrated in locations that are considered discriminative following the

[4] Two methods have been implemented to obtain the discrete set of classes: the torus manifold discretization (see Sect. 6.4.1.2) and 2D pose space clustering (see Sect. 6.4.2.1).

discriminative log-likelihood maps (discussed in Sect. 6.2). For each node, let the set of classes that fall under this node be C_n and the subset of classes belonging to one of its branches b be C_b. Then for each branch b, n_H locations are sampled from the distribution $\mathcal{L}(C_b, C_n)$ (as described in Sect. 6.2.1) to give a set of potentially discriminative locations for HOG descriptors:

$$H_p = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n_H} \sim \mathcal{L}(C_b, C_n). \tag{6.9}$$

For each of these positions we sample a corresponding HOG descriptor parameter:

$$H_\Psi = \{\psi^{(i)}\}_{i=1}^{n_H} \in \Psi \tag{6.10}$$

from a parameter space $\Psi = (W \times B \times A)$ where $W = \{16, 24, 32\}$ is the width of the block in pixels, $B = \{(2,2),(3,3)\}$ are the cell configurations considered and $A = \{(1,1),(1,2),(2,1)\}$ are aspect ratios of the block. An example is proposed in Fig. 6.6a for a 16 × 16 HOG block with 2 × 2 cells.

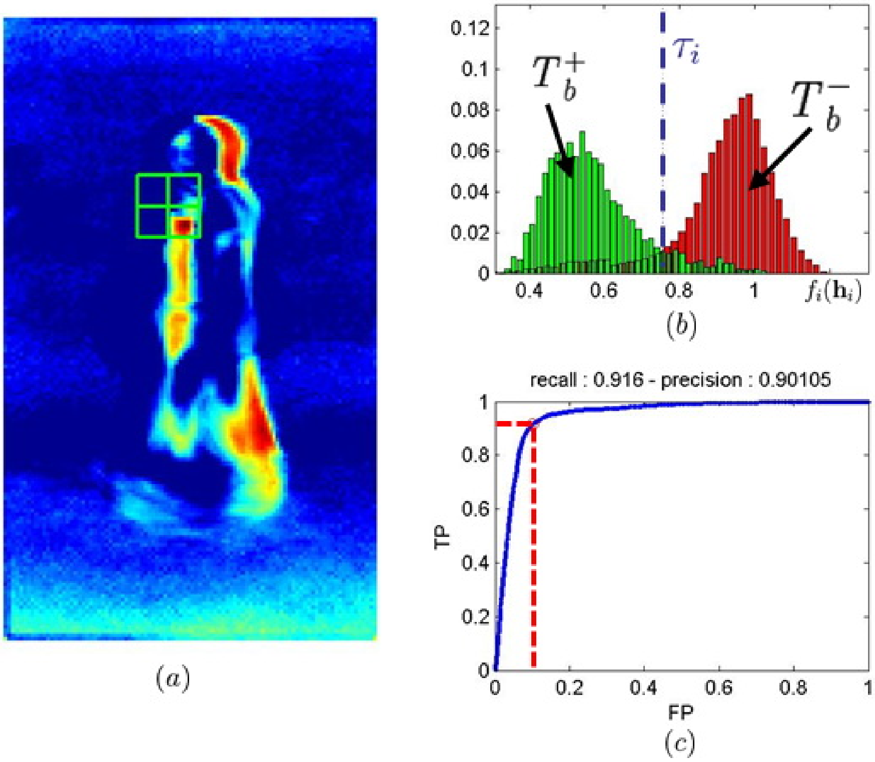


Figure 6.6: Example of a selected HOG block from $\mathcal{B}$: we show the location of the 16×16 block (2×2 cells) on top of the log-likelihood map for the corresponding branch of the tree (**a**). We represent in (**b**) the 2 distributions obtained after training a binary classifier: the green distribution corresponds to the set of training images T_b^+ that should pass through that branch while the red one corresponds to the set T_b^- that should not pass. In this example, $f_i(\mathbf{h}_i)$ is the projection of the block $\mathbf{h}_i$ on the hyperplane found by Support Vector Machines (SVM). Finally, we represent in (**c**) the ROC curve with True Positive (TP) vs False Positive (FP) rates varying the decision threshold. We give the precision and recall values for the selected threshold (red dotted line).

Next, at each location $(x^{(i)}, y^{(i)}) \in H_p$ HOG features are extracted from all positive and negative training examples using corresponding parameters $\psi^{(i)} \in H_\Psi$. For each location a positive set T_b^+ is created by sampling from instances belonging to C_b and a negative set T_b^-

Algorithm 7: HOG Blocks Selection

```
input  : Hierarchical structure $\mathcal{S}$, and training images. Discriminative classifier $g(\cdot)$.
output: List of discriminative HOG blocks $\mathcal{B}$.
for each level $l$ do
    for each node $n$ do
        Let $C_n$ = set of classes under $n$;
        for each branch $b$ do
            Let $C_b$ = set of classes under branch $b$;
            Compute $\mathcal{L}(C_b, C_n)$ (cf. § 6.2);
            $H_p = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n_H} \sim \mathcal{L}(C_b, C_n)$;
            $H_\Psi = \{\psi^{(i)}\}_{i=1}^{n_H} \in \Psi$;
            for $i = 1$ to $n_H$ do
                $(x^{(i)}, y^{(i)}) \in H_p$;
                $\psi_i \in H_\Psi$;
                Let $T_b^+ \in C_b$;
                Let $T_b^- \in C_n - C_b$;
                for all images under $n$ do
                    Extract HOG at $(x^{(i)}, y^{(i)})$ using $\psi^{(i)}$;
                Let $h_i^+$ = HOG from $T_b^+$;
                Let $h_i^-$ = HOG from $T_b^-$;
                Train classifier $g_i$ on $\frac{2}{3}$ of: $h_i^+$ and $h_i^-$;
                Test $g_i$ on OOB set $\frac{1}{3}$ of: $h_i^+$ and $h_i^-$;
                Rank block $(x^{(i)}, y^{(i)}, \psi^{(i)}, g_i)$
            Select $n_h$ best blocks, $B_b = \{(x_j, y_j, \psi_j, g_j)\}_{j=1}^{n_h}$ ;
            Update the list $\mathcal{B}' = \mathcal{B} \cup B_b$ ;
```

is created by sampling from $C_n - C_b$. An out-of-bag (OOB) testing set is created by removing 1/3 of the instances from the positive and negative sets. A discriminative binary classifier g_i (e.g. SVM) is trained using these examples. We then test this weak classifier on the OOB test instances to select a threshold τ_i and rank the block according to the actual True Positive (TP) and False Positive (FP) rates achieved: the overall performance of the weak classifier g_i is determined by selecting the point on its ROC curve lying closest to the upper-left hand corner that represents the best possible performance (i.e. 100% TP and 0% FP). We then rank the rejector using the Euclidean distance to that point. Each weak classifier $g_i(h_i)$ thus consists of a function f_i, a threshold τ_i and a parity term p_i indicating the direction of the inequality sign:

$$g_i(\mathbf{h_i}) = \begin{cases} 1 & \text{if } p_i f_i(\mathbf{h}_i) < p_i \tau_i \ , \\ 0 & \text{otherwise.} \end{cases} \tag{6.11}$$

Here $\mathbf{h}_i$ is the HOG extracted at location $(x^{(i)}, y^{(i)})$ using parameters $\psi^{(i)}$. In the example proposed in Fig. 6.6b, $f_i(\mathbf{h}_i)$ is the projection of the block $\mathbf{h}_i$ on the hyperplane found by SVM. The corresponding ROC curve is shown in Fig. 6.6c. For each branch, the n_h best blocks are kept in the list $\mathcal{B}$ which is a bag/pool of HOG blocks and associated weak classifiers. If n_B is the total number of branches in the tree over all levels, the final list $\mathcal{B}$ has $n_B \times n_h$ block elements. The selection of discriminative HOG blocks is depicted in Alg. 7.

By this process, features are extracted from areas of high edge probability across our training set more than areas with low probability. By using this information to sample features, the proportion of useful information available for random selection is increased.

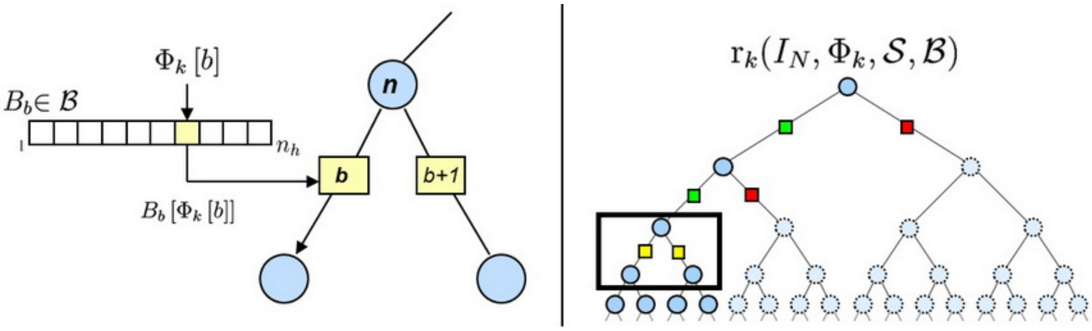


Figure 6.7: Rejector branch decision: shown here is a diagram of a rejector classifier. Note that the structure allows for any number of branches at each node and is not fixed at 2. The tree-like structure of the classifier is defined by $\mathcal{S}$ and the blocks (and associated weak classifiers) stored during training for each branch for this structure are held in $\mathcal{B}$. The random vector Φ_k defines a cascade r_k by selecting one of the rejector blocks in B_b at each branch b, being $B_b \in \mathcal{B}$ the bag of HOG blocks for branch b.

6.3.3 Randomization

Random Forests, as described in [Breiman, 2001] are constructed as follows. Given a set of training examples $T = \{(y_i, \mathbf{x_i})\}$, where y_i are class labels and $\mathbf{x_i}$ the corresponding D-dimensional feature vectors, a set of random trees F is created such that for the k-th tree in the forest, a random vector Φ_k is used to grow the tree resulting in a classifier $t_k(\mathbf{x}, \Phi_k)$. Each element in Φ_k is a randomly selected feature index for a node. The resulting forest classifier F is then used to classify a given feature vector $\mathbf{x}$ by taking the mode of all the classifications made by the tree classifiers $t \in F$ in the forest.

Each vector Φ_k is generated independently of the past vectors $\Phi_1..\Phi_{k-1}$, but with the same distribution:

$$\phi \sim \mathcal{U}(1, D), \forall \phi \in \Phi_k, \tag{6.12}$$

where $\mathcal{U}(1, D)$ is a discrete uniform distribution on the index space $\mathcal{I}_D = \{1, \cdots, D\}$. The dimensionality of Φ_k depends on its use in the tree construction (i.e. the number of branches in the tree). For each tree, a decision function $f(\cdot)$ splits the training data that reaches a node at a given level in the tree by selecting the best feature m* from a subset of $m << D$ randomly sampled features (typically $m = \sqrt{D}$ dimensions). An advantage of this method over other tree based methods (e.g. single decision trees) is that since each tree is trained on a randomly sampled $\frac{2}{3}$ of the training examples [Breiman, 1996] and that only a small random subset of the available dimensions are used to split the data, each tree makes a decision using a different view of the data. Each tree in the forest F learns quite different decision boundaries, but when averaged together the boundaries end up reasonably fitting the training data.

In Random Forests (RF), the use of randomization over feature dimensions makes the forest less prone to over-fitting, more robust to noisy data and better at handling outliers than single decision trees [Breiman, 2001]. We exploit this random selection of features in our algorithm so that our classifier will also be less susceptible to over-fitting and more robust to noise compared to a single hierarchical decision tree that would use all the selected features together: a hierarchical tree-structured classifier $r_k(I_N, \Phi_k, \mathcal{S}, \mathcal{B})$, with I_N being a normalized input image, is thus built by randomly sampling one of the HOG blocks in the list $B_b \in \mathcal{B}$ at each branch b of the hierarchical structure $\mathcal{S}$. This gives a random n_B-dimensional vector Φ_k where each element corresponds to a branch in the structure $\mathcal{S}$:

$$\Phi_k \in \mathcal{I}^{n_B} \text{ where } \phi \sim \mathcal{U}(1, n_h), \forall \phi \in \Phi_k. \tag{6.13}$$

Here $\mathcal{U}(1, n_h)$ is a discrete uniform distribution on the index space $\mathcal{I} = \{1, \cdots, n_h\}$. The value of each element in Φ_k is the index of the randomly selected HOG block from B_b for its corresponding branch. An ensemble R of n_c hierarchical tree-structured classifiers is grown by repeating this process n_c times:

$$R(I_N) = \{r_k(I_N, \Phi_k)\}_{k=1}^{n_c}, \tag{6.14}$$

where $\mathcal{S}$ and $\mathcal{B}$ are left out for brevity. R thus has 2 design parameters n_c and n_h whose main effects on performance will be evaluated in Sect. 6.4.2. Let $\mathcal{P}_\omega$ be the path through the tree-structure $\mathcal{S}$ from the top node to the class leaf ω. $\mathcal{P}_\omega$ is in fact an ordered sequence of ${n_b}^\omega$ branches[5]: $\mathcal{P}_\omega = \left(b_1^\omega, b_2^\omega, \cdots, b_{{n_b}^\omega}^\omega\right)$. For each classifier $r_k \in R$ and for each class $\omega \in \Omega$, where $\Omega = \{1, \cdots, n_\omega\}$, we have the corresponding ordered sequence of HOG blocks and associated weak classifiers:

$$\mathcal{H}_k^\omega = \{(x_j, y_j, \psi_j, g_j)\}_{j=1}^{{n_b}^\omega}, \tag{6.15}$$

$$\text{with } (x_j, y_j, \psi_j, g_j) = B_b\left[\Phi_k(b)\right], \tag{6.16}$$

where the branch index b is the j^{th} element in $\mathcal{P}_\omega$ (i.e $b = \mathcal{P}_\omega[j]$) and $B_b \in \mathcal{B}$. See Fig. 6.7a.

For each tree-structured classifier r_k, the decision to explore any branch in the hierarchy is based on a accept/reject decision of a simple binary classifier g_j that works in a similar way to a cascade decision (see Fig. 6.7b). The ensemble classifier R is then, in fact, a series of *Randomized Hierarchical Cascades of Rejectors.*

The decision at each node of a hierarchical cascade classifier r_k is made in a one-vs-all manner, between the branch in question and its sibling branches. In this way multiple paths can be explored in the cascade that can potentially vote for more than one class. This is a useful attribute for classifying potentially ambiguous classes and allows the randomized cascades classifier to produce a distribution over multiple likely poses. Each cascade classifier r_k therefore returns a vector of binary outputs (yes/no) $\mathbf{o}_k = \left(o_k^1, o_k^2, \cdots, o_k^{n_\omega}\right)$ where a given o_k^ω from $\mathbf{o}_k$ for class index ω, takes a binary value:

$$o_k^\omega = \begin{cases} 1 & \text{if } \forall (x_j, y_j, \psi_j, g_j) \in \mathcal{H}_k^\omega,\ g_j(\mathbf{h}_j) = 1\ , \\ 0 & \text{otherwise.} \end{cases} \tag{6.17}$$

Here $\mathbf{h}_j$ is the HOG extracted at location (x_j, y_j) using parameters ψ_j. Each output o_k^ω is initialized to zero, which means that if no leaf is reached during classification, r_k will return a vector of zeros and will not contribute to the final classification. The uncertainty associated with each rejector during learning could be considered to provide a crisp output (i.e. each cascade voting for only one class) or to compute a soft confidence values of the cascade classifiers. Although other classifier combination techniques could be considered, here we choose to use a simple sum rule to combine the binary votes from the different cascade classifiers in the ensemble R that outputs the vector $\mathbf{O}$:

$$\mathbf{O} = \left(O^1, O^2, \cdots, O^{n_\omega}\right) = R(I), \tag{6.18}$$

where each $O^\omega \in [0, n_c]$ represents a number of *votes*:

$$O^\omega = \sum_{k=1}^{n_c} o_k^\omega, \tag{6.19}$$

[5] Note that ${n_b}^\omega$ could be different for each class $\omega \in \Omega$

and can be used to assign a confidence value[6] (or score) to class ω. $\mathbf{O}$ can ultimately be an input in to another algorithm, e.g. tracking, or a class can be estimated by majority voting, i.e. taking for instance the mode of all the classifications as in the example presented in Fig. 6.8.

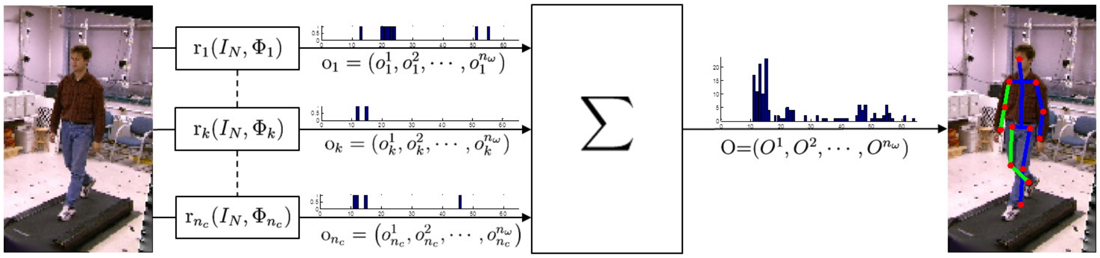

$\mathrm{r}_1(I_N, \Phi_1)$
$\mathrm{o}_1 = (o_1^1, o_1^2, \cdots, o_1^{n_\omega})$
$\mathrm{r}_k(I_N, \Phi_k)$
$\mathrm{o}_k = (o_k^1, o_k^2, \cdots, o_k^{n_\omega})$
$\mathrm{r}_{n_c}(I_N, \Phi_{n_c})$
$\mathrm{o}_{n_c} = (o_{n_c}^1, o_{n_c}^2, \cdots, o_{n_c}^{n_\omega})$
$\sum$
$\mathrm{O}=(O^1, O^2, \cdots, O^{n_\omega})$

Figure 6.8: Diagram of Image classification using an ensemble of n_c randomized cascades (*from left to right*): When applied to a normalized input image I_N, each cascade classifier $r_k(I_N, \Phi_k, \mathcal{S}, \mathcal{B})$ returns a vector of binary outputs $\mathbf{o}_k = \left(o_k^1, o_k^2, \cdots, o_k^{n_\omega}\right)$. A sum-rule is used to combine the binary votes from the different cascade classifiers in the ensemble R that outputs the distribution over classes $\mathbf{O} = \left(O^1, O^2, \cdots, O^{n_\omega}\right)$, with $O^\omega = \sum_{k=1}^{n_c} o_k^\omega$ and $O^\omega \in [0, n_c]$, being n_ω the number of classes and n_c the number of cascades. The image I_N can be classified by taking, for instance, the class ω^* that received more *votes*, i.e. the peak in the final distribution over classes. The average 2D pose corresponding to class ω^* is shown on the *right*.

6.4 Class Definition

6.4.1 HumanEVA Dataset

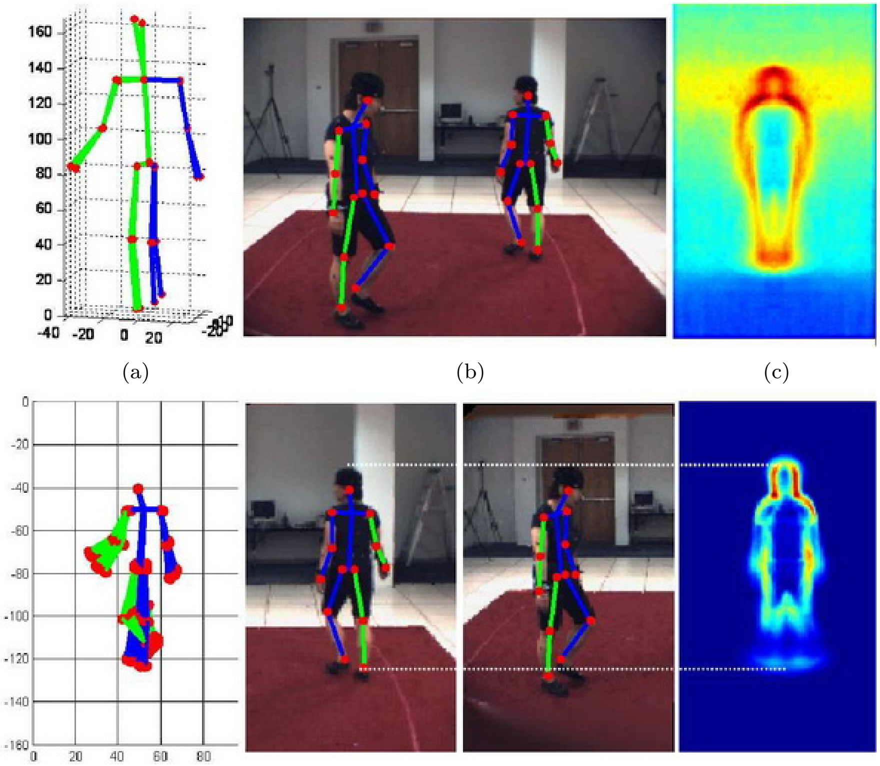
The first dataset we consider is the HumanEva dataset [Sigal et al., 2010]: HumanEVA I for training and HumanEVA II for testing. This dataset consists of 4 subjects performing a number of actions (e.g. walking, running, gesture) all recorded in a motion capture environment so that accurate ground truth data is available.

6.4.1.1 Alignment of Training Data

Before training the algorithm, strong correspondences are established between all the training images. While other approaches require a manual process [Dalal and Triggs, 2005] or a clean silhouette shape (either synthetic [Shakhnarovich et al., 2003, Agarwal and Triggs, 2006, Dimitrijevic et al., 2006] or from manual labeling [Gavrila, 2007]) for feature alignment, we propose a fully automatic process for accurately aligning real training images. It is assumed that the 3D mocap data (3D poses) or other pose labeling corresponding to the training images are available. Then by applying a simple but effective way of using their 2D pose projections in the image plane, all of the training images are aligned.

The complete process is depicted in Fig. 6.9: the 3D joints of every training pose are first projected onto the corresponding image plane (Fig. 6.9b). The resulting 2D joints are then aligned using rigid-body Procrustes alignment [Bookstein, 1991], uniformly scaled and centered in a reference bounding-box (Fig. 6.9d). For each training pose, we then estimate the four parameters corresponding to a similarity transformation (one for rotation, two degrees of freedom for translation and one scale parameter) between the original 2D joints locations in the original input image and the corresponding aligned and resized 2D joints. This transformation

[6] Although it can be expressed as a *percentage of votes* by dividing by the number of cascades n_c, $\widetilde{O}^\omega = \frac{1}{n_c}\sum_{k=1}^{n_c} o_k^\omega$ and $\widetilde{O}^\omega \in [0, 1]$, it does not express a probability ($\sum_{k=1}^{n_\omega} \widetilde{O}^\omega \neq 1$).



(a) (b) (c)

(d) (e) (f) (g)

Figure 6.9: Alignment of training images. (**a**) Training 3D poses from Mocap data. (**b**) Training images with projected 2D Poses. (**c**) Average gradient image over INRIA training examples (96 × 160). (d) Aligned and re-sized 2D poses (96 × 160). (**e**) and (**f**) cropped images (96 × 160) with normalized 2D poses. (**g**) Average gradient image over aligned HumanEva [Sigal et al., 2010] training examples.

is finally applied to the original image leading to the cropped and aligned image (Fig. 6.9e and f). In this work, we normalize all the training images to 96×160 as in [Dalal and Triggs, 2005]. The dataset (images and poses) is then flipped along the vertical axis to double the training data size. Applying the process described above to this data (3 Subjects and 7 camera views) for training, we generate a very large dataset of more than 40,000 aligned and normalized 96×160 images of walking people[7], with corresponding 2D and 3D Poses. After that is done, classes need to be defined to train our pose classifier.

6.4.1.2 Class Definition - Torus Manifold Discretization

Since similar 3D poses can have very different appearance depending on the camera viewpoint (see Fig. 6.9a and Fig. 6.9b), the viewpoint information has to be included into the class definition. To define the set of classes, we thus propose to utilize the same torus manifold for viewpoint and action as we did in the previous chapters. The walking sequences are then mapped on the surface of this torus manifold and classes are defined by applying a regular grid

[7] The average gradient image obtained with our training dataset (Fig. 6.9g) shows more variability in the lower region compared to the one obtained from INRIA dataset (Fig. 6.9c). This is due to the fact that most of the INRIA images present standing people.

on the surface of the torus thus discretizing gait and camera viewpoint[8]. By this process, we create a set of 192 homogeneous classes (12 for gait and 16 for camera viewpoint) with about the same number of instances (see Fig. 6.10).

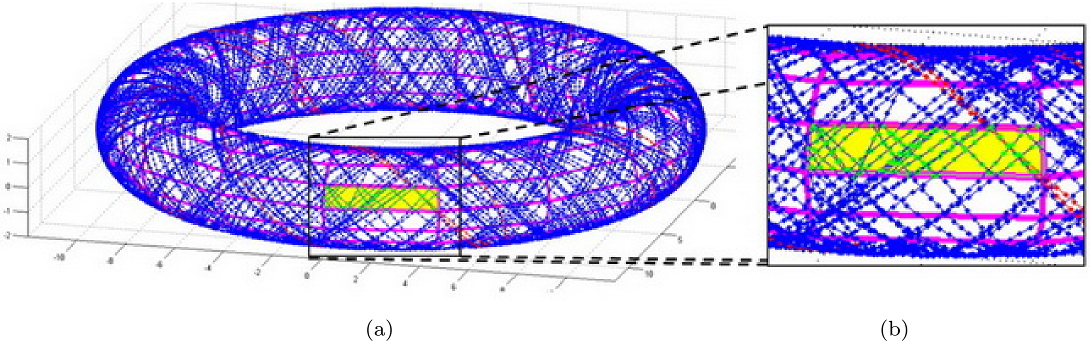


Figure 6.10: Class definition - HumanEVA: (a) Torus manifold with discrete set of classes (12 for gait and 16 for camera viewpoint) and training sequences (each blue dot represents a training image). (b) Zoom on a particular class (yellow) with corresponding training images (green dots). Please refer to Fig. 6.2 where we present 5 examples of aligned images belonging to the class highlighted in (b).

We choose to define non-overlapping class quantization, due to the property of our hierarchical cascade classifier that each cascade compares feature similarity rather than make a greedy decision, so can traverse more than one branch in the hierarchical cascade. When an image reaches a node with a decision between very similar classes (and subsequently close in pose space), then it is possible that a query image that lies close to a quantization border between those two classes can arrive at both class leaves.

6.4.1.3 Classification

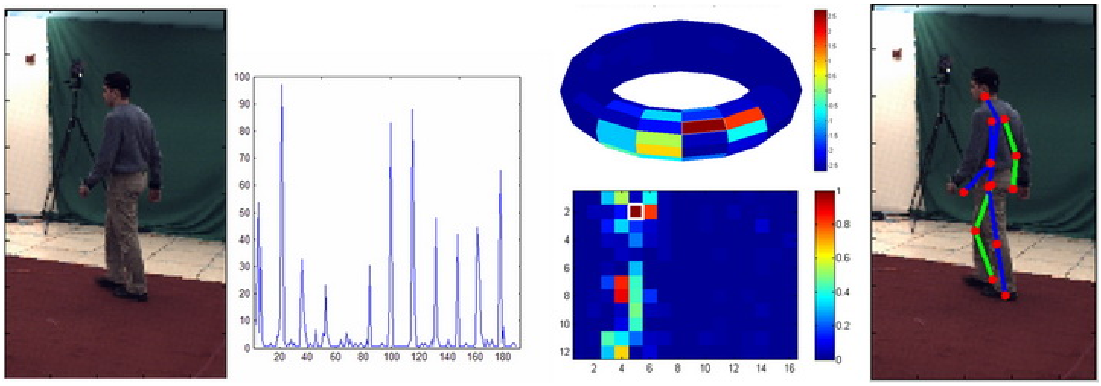
We first performed a qualitative experimentation to validate the classifier in similar conditions and extract normalized 96 × 160 images from the HumanEva II dataset. We ran a 200-cascade classifier ($n_h = 50$) on this set of testing images. Generally the classification was very satisfactory as shown in the example given in Fig. 6.11 where the resulting distribution over classes is visualized on the torus manifold.

When testing the same classifier in different conditions we observed that the classification results were not so good. This is due to the low variability in pose present in HumanEVA walking database: even if 40,000 images are available, they are not representative of the variability in terms of gait style since only 3 subjects walking at the same speed have been considered. Additionally, HumanEva has very little background (one unique capture room) and clothing (capture suit) variation, which make the classifier unrobust against cluttered background. We thus propose to consider a second data set to fully validate our cascade classifier.

6.4.2 MoBo Dataset

Our second set of experiments is performed using the CMU Mobo dataset [Gross and Shi, 2001]. Again, we consider the walking action but this time add more variability in the dataset by including 15 subjects, two walking speeds (high and low) and a discretized set of 8 camera viewpoints, uniformly distributed between 0 and 2π. Between 30 and 50 frames of each sequence

[8] Since we only learnt our pose detector from walking human sequences, we do not attempt to detect people performing other actions than walking.



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

Figure 6.11: HumanEva II image classification of a normalized 96 × 160 input image (a) using an ensemble of 100 randomized cascades. The resulting distribution over the 192 classes after classification (b) is also represented on the 3D and 2D representations of the torus manifold in (c). The average 2D pose corresponding to the class with the highest score is represented on top of the input image in (d).

were selected in order to capture exactly one gait cycle per subject. By this process we generate a training database encompassing around 8000 images and the corresponding 2D and 3D pose parameters. The same alignment procedure is applied to this dataset as with the HumanEva dataset, but in addition we use hand labeled silhouette information to superimpose each of the instances on random backgrounds to increase the variability of the nonhuman area of the image (see Fig. 6.12).

We observed that the classification in section 6.4.1.3 was very strict so that very fine alignment was necessary to obtain an acceptable result. Looser alignment in the training should allow for a more tolerant classification. Following [Laptev, 2009] and [Ferrari et al., 2008], the training set is thus augmented by perturbing the original examples with small rotations and shears, and by mirroring them horizontally. This improves the generalization ability of the classifier. The augmented training set is 6 times larger and contains more than 48,000 examples with different background. The same dataset is generated for the 3 following configuration: original background, no background, and random background (see Fig. 6.12). This dataset will enable to measure the effect of cluttered background on classification. The richer background variation compared to HumanEva dataset allows a more robust cascade to be learnt, as demonstrated later by our experiments.

6.4.2.1 Class Definition - 2D Pose Space Clustering

Class definition is not a trivial problem because changes in the human motion are continuous and not discrete. In other words, it is not trivial to decide where a class ends and where the next one starts. Previously no one has attempted to efficiently define classes for human pose classification and detection. In [Okada and Soatto, 2008] they clustered 3D pose space and also considered a discrete set of cameras. Gavrila [2007] clustered the silhouette space using binary edge maps. Ferrari et al. [2008] defined classes as 3D pose but only considered frontal views and a limited set of poses. In PSH [Shakhnarovich et al., 2003], they automatically build the neighborhood in 3D pose space but only consider frontal views. This definition produced good results in the absence of viewpoint changes. However, two poses which are exactly the same in the pose space could still have completely different appearances in the images due to changes in viewpoint (see example in Fig. 6.9a and b). Thus it is critical to consider viewpoint information

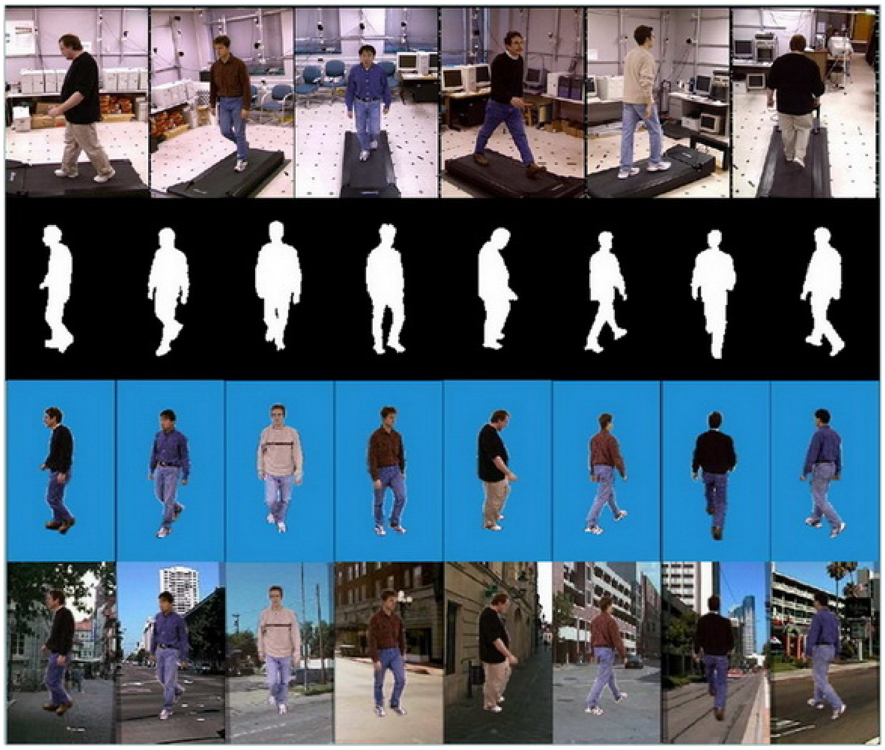


Figure 6.12: Modified MoBo dataset (*from top to bottom*): we show an original image for each one of the 6 available camera views (*1st row*). We then present a normalized 96 × 160 binary segmentation per training view (*2nd row*), the corresponding segmented images (*3rd row*) and images with a random background (*4th row*).

in the class definition.

Ideally, classes should be defined in the feature space or, at least, in the 2D pose space (2D projection of the pose in the image) that takes into account the information of the camera viewpoint. When we considered the HumanEVA dataset (Sect. 6.4.1 and later Sect. 7.4.2.1), classes were defined applying a regular grid on the surface of the 2D manifold of pose plus viewpoint (torus manifold). Mapping training poses on a torus worked nicely with HumanEva because only 3 subjects with similar aspect and walking style were considered. When the dataset is richer (more subjects, more walking speeds and styles, more variability in morphology), it is more difficult to map the poses on a torus and the discretization of the manifold produces non-homogeneous classes. Additionally, the discretization process assigns the same number of classes for quite different viewpoints. This means that, for instance, frontal and lateral viewpoints of a walking cycle were quantized with equal number of classes, despite the difference in appearance variability. However, since there is much less visual change over the gait cycle when viewed from front or back views than for lateral views, differences between classes do not reflect the same amount of change in visual information over each of these views. This over-quantization of visually quite similar views can make the class data unrealistically difficult to separate for certain viewpoints, and can introduce some confusion when a cascade must classify those examples. Therefore it is important to define homogeneous classes. Too many class clusters become too specific to a particular subject or pose, and do not generalize well enough. Too few clusters can merge poses that are too different and no feature can be found to represent all the images of the class.

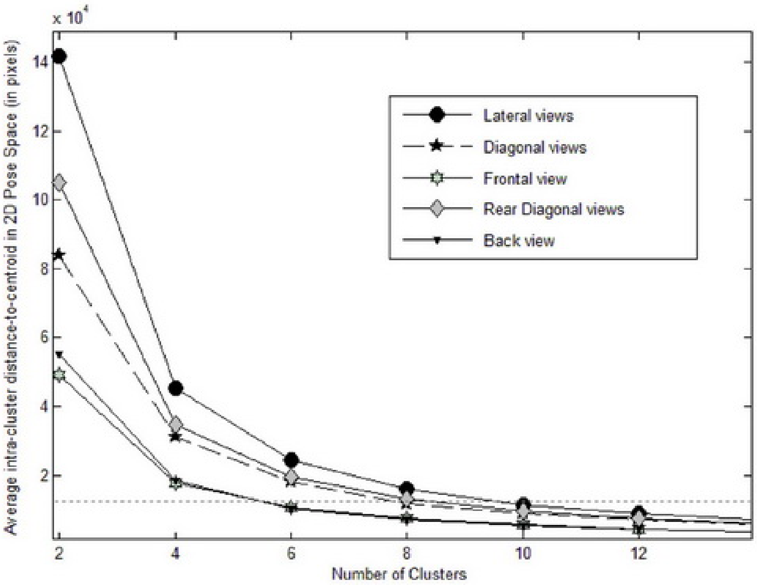


Figure 6.13: Class Definition for MoBo dataset: For each training view, the 2D pose space is clustered and for each number of clusters, the intra-cluster distances to centroid are computed. By selecting a thresholding value of this average intra-cluster distance, the views are quantized into a number of classes that reflect the variability in appearance for that view. For the selected threshold (dashed line), frontal views of walking have a coarser quantization (6 classes) compared to the diagonal (8 classes) and lateral views (10 classes).

For each training view, the 2D pose space is clustered several times with K-means and for each number of clusters K, the intra-cluster distances to centroid are computed in the entire space. This distance indicates the tightness of the clusters. By selecting a thresholding value of the average intra-cluster distance (See Fig.6.13), the views are quantized into a number of classes that reflect the variability in appearance for that view. For the selected threshold (dashed line) frontal views of walking have a coarser quantization (6 classes) compared to the diagonal (8 classes) and lateral views (10 classes)[9]. We can see on Fig. 6.13 that if we choose the same number of clusters for all the views as we did in section 6.4.1 or in the first part of this thesis, the resulting average intra-cluster distances are very different from a view to another: for example, if we select 6 clusters per view (as in chapter 2), the resulting clusters from the frontal view are about two times tighter than the ones from lateral views.

In the proposed automatic class definition described here, views are quantized into a number of classes that reflect the variability in appearance for that view, and frontal views of walking would have a coarser quantization (i.e. less classes) compared to the lateral views. Using this method it is possible to create class definitions that better reflect the differences in variation over the gait cycle between different views. The resulting 64 classes are presented in Fig. 6.14.

6.5 Experiments using MoBo Dataset

A training subset is first built by randomly selecting 10 of the 15 available subjects from the database and a testing subset with the remaining 5 subjects, thus considering 30,000 images

[9] We select this threshold in order to obtain a minimum of 6 clusters in the frontal and back views as in the first part of this thesis.

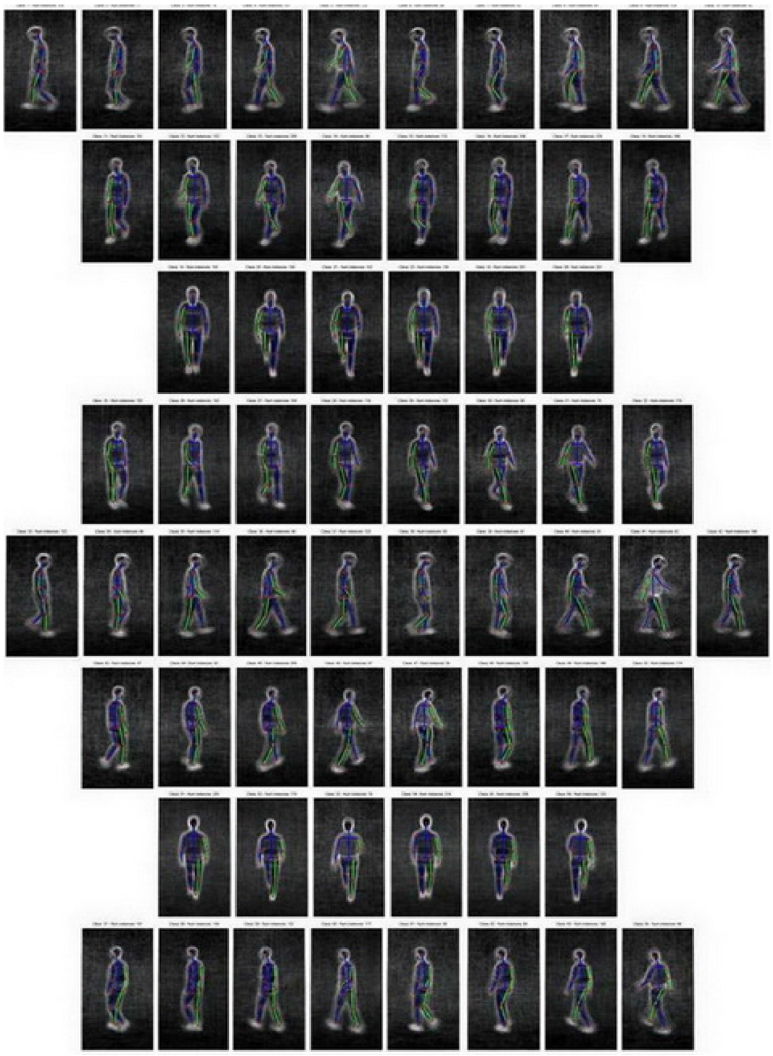


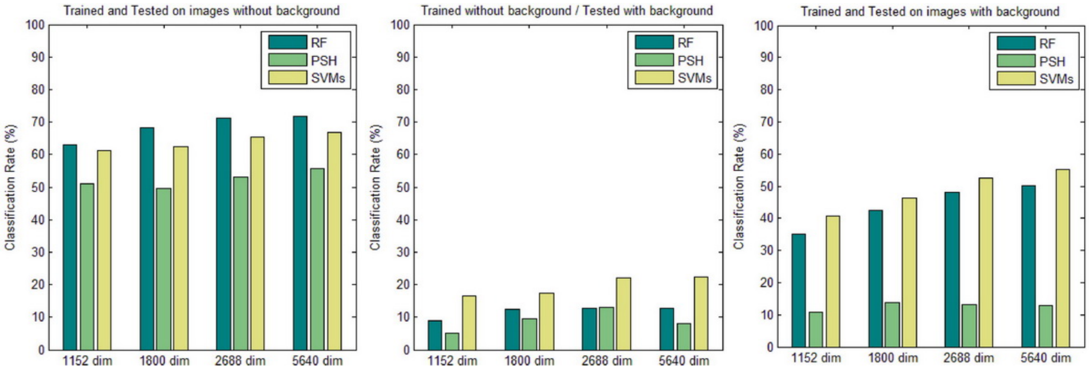
Figure 6.14: Classes (64) obtained with the class definition method presented in Fig.6.13. The 8 rows correspond to the 8 training views available in the dataset. For each class, we show the average gradient map and average pose.

for training and 18,000 for testing. Benchmark experiments were performed on this dataset to get initial baseline results with three state-of-the-art multi-class (pose) classifiers:

- Random Forests [Breiman, 2001], inherently good for multi-class problems, that share similarities with our approach.
- PSH [Shakhnarovich et al., 2003] which is a fast and effective way of finding the neighboring poses of a query image. We will take the class of the nearest pose as a classification.
- A multi-SVMs classifier, similar in spirit to that of [Okada and Soatto, 2008], which learns k one-vs-all linear SVMs to discriminate between k predefined pose clusters[10].

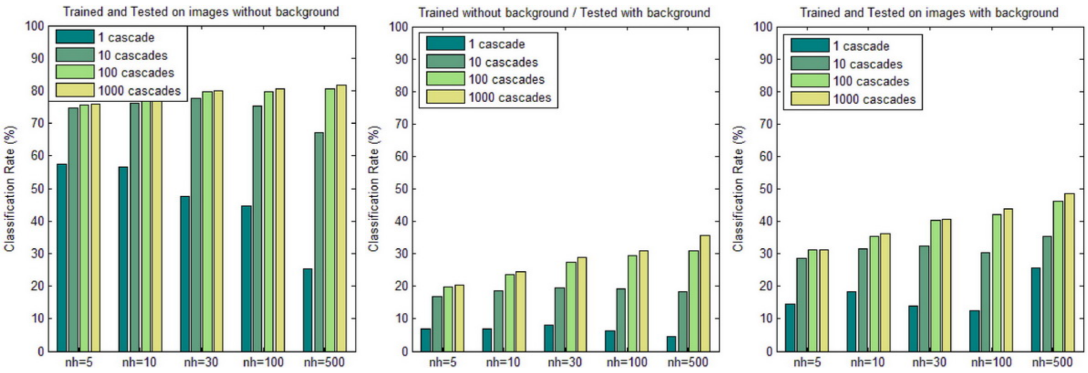
[10] As we do not perform pose-dependent feature selection and use linear SVMs instead of the ARD-Gaussian

During classification the class is determined by choosing the cluster that has the highest probability $p(C_k|\mathbf{x})$ from the SVMs. We will refer to this work as multi-SVMs or SVMs in the rest of the chapter.



(a) (b) (c)

Figure 6.15: Pose classification for PSH, Random Forest (1000 trees) and SVMs classifiers trained on a subset of the MoBo database containing 10 subjects and tested on the remaining 5 subjects. We compare the results using segmented images without background and images with a random background for 4 different grids of HOG descriptors.



(a) (b) (c)

Figure 6.16: Pose classification rates for 4 different ensembles (1, 10, 100 and 1000 cascades) trained on the subset of the MoBo database containing 10 subjects and tested on the remaining 5 subjects. We compare the results using segmented images without background and images with a random background for different number n_h of sampled HOG blocks.

We tuned the parameters of PSH and RF to get optimal performances on our 64 class MoBo dataset: we tailored PSH parameters to the number of images and used 200 18-bit hash functions and empirically validated that the optimum number m of randomly selected features for RF was near $\sqrt{D}$ (as indicated in [Breiman, 2001]). We trained 64 one-vs-all linear SVMs. Two groups of classifiers are trained using segmented images without background and images with a random background respectively. We run the same test for 3 different grids of HOG descriptors. Table 6.1 gives the corresponding training time and Fig. 6.15 shows the

kernel SVMs, it is expected to perform a little lower than that of [Okada and Soatto, 2008] but the training will be much faster. Since we use linear SVMs to train our cascade rejectors, we find it is a reasonable comparison.

performances[11] when classifying images with or without background. RF results are given for a 1000-tree forest since convergence is reached for that number as observed in Fig. 6.1.

Table 6.1: Classifiers training time on an Intel Core Quad Processor at 2.00GHz with 8Gb of RAM (see experiments in Fig. 6.15).

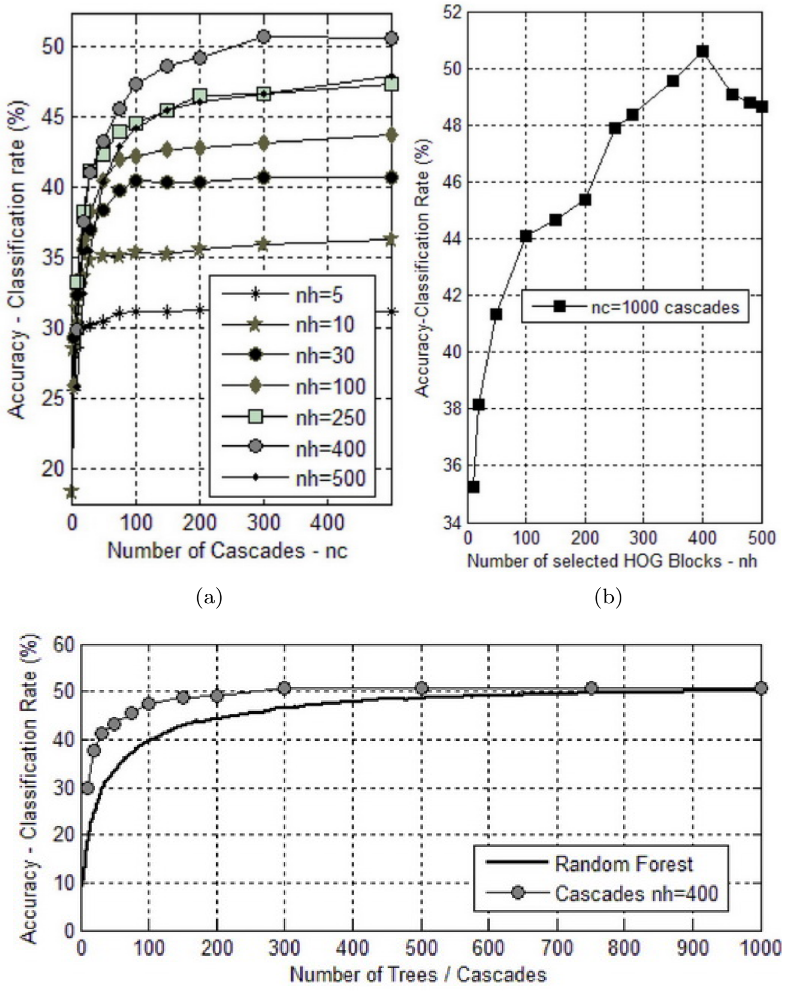
HOG Grid		4x4	5x5	7x12	3 together
Dimension		1152	1800	2688	5640
Backgrd	RF	5h30	6h45	13h00	21h45
	SVMs	2h00	4h00	8h00	17h30
	PSH	1h50	2h30	3h00	5h30
No Backgrd	RF	4h40	5h30	11h00	18h30
	SVMs	45min	52min	1h00	2h20
	PSH	1h20	2h20	3h08	5h30

The first observation we made is that all of the classifiers trained on segmented images perform poorly when classifying images with background (see Fig. 6.15b). This demonstrates the importance of considering a training dataset that includes a wider background appearance. The second observation is that using denser HOG feature grids improves pose classification accuracy but we are quickly facing memory issues that prevent us from working with denser grids. The third observation we can make is that PSH does not perform well in presence of cluttered background (Fig. 6.15c) while performing decently on segmented images (Fig. 6.15a). PSH tries to take a decision based on 1-bin splits. That could well be a reason as the histogram will be altered by the presence of background and affect that bin. In presence of cluttered background (Fig. 6.15c), SVM is the best classifier in terms of accuracy while Random Forest achieves the highest classification rate when working with segmented images (Fig. 6.15a).

After constructing the tree structure $\mathcal{S}$ (112 branches using $n_\theta = 1$), two lists of HOG block rejectors $\mathcal{B}$ are built using the two same training subsets of the MoBo database (with and without random background). The training took about 2 hours (testing 2500 HOG blocks at each branch) which is much faster than both SVMs and RF classifier training. Then, different ensembles are created varying the number n_c of cascades and the number n_h of HOG blocks that are considered for sampling. Corresponding classification rates on MoBo dataset are given in Fig. 6.16.

We can observe that the accuracy increases with both n_c and n_h for the 3 different tests and the cascades outperform the other classifiers when trained on images without background (Fig. 6.16 a and b). In particular, the cascades show better generalization performances when the testing data is significantly different from the training data (Fig. 6.16b). Performances seems to be lower than SVMs and RF when the classifiers are trained and tested on images with random background (see Fig. 6.16c vs Fig. 6.15c). Detailed results are reported in Fig. 6.17 for that concrete case. Fig. 6.17a shows that convergence is reached sooner when n_h is low and the accuracy improves when increasing n_h until $n_h = 400$ for $n_c = 1000$ (See Fig. 6.17b). The figure 6.17c compares the performances of a cascades classifier grown using the 400 best HOG blocks with the best RF classifier from Fig 6.1. The classification rate of the cascades classifier converges earlier (300 cascades) because the first cascades are already very informative compared to the first trees of the Random Forest: 30% of accuracy for the first cascade and 10% for the first tree of the RF. Actually, the accuracy of RF does not seem to converge to

[11] If neighboring classes are not considered as misclassification, some issues with images on the "boundaries between classes" are solved and the classification rates increase by about $15 - 20\%$



(a)

(b)

(c)

Figure 6.17: Pose classification experiments on MoBo using randomized cascades: (**a**) we run the same experiment with our randomized cascades classifier (trained on the subset containing 10 subjects and tested the remaining 5 subjects from MoBo dataset) and compare the results obtained sampling from different numbers n_h of selected HOG blocks at each branch of the tree. (**b**) We show the results varying n_h with $nc = 1000$. (**c**) We compare the performance of the best Random Forest vs our randomized cascades classifier (with $n_h = 400$).

a specific value as every new tree is informative. Our approach performs better than RF for $n_c \leq 1000$ and is slightly less efficient than SVMs.

6.6 Conclusions

In this chapter, we have proposed a new multi-class classifier that combines the best components of state-of-the-art classifiers including hierarchical trees, cascades of rejectors and randomized forests. Other algorithms require that the full feature space be available from an image during training. This can lead to very high dimensional feature vectors being extracted from an image for large configurations of features and potentially leads to memory problems for training sets with a large number of examples. Our algorithm selects the features it considers to be the most informative during training, and can build a smaller more useful feature space by sampling a small set of features from a much larger configuration of feature descriptors. This approach is computationally efficient as cascades learning takes around 2 hours while SVM and RF approaches took respectively 17h30 and 21h45 with the same training set.

Cascade approaches are efficient at quickly rejecting negative examples, and we have exploited this property by learning fast multi-class hierarchical cascades. By randomly sampling the feature, each cascade uses different sets of features to vote, it adds some robustness to noise that helps to prevent over fitting. Moreover, each cascade can vote for one or more class so the ensemble of random cascade classifiers outputs a distribution over possible poses that can be useful when combined with tracking algorithms for resolving ambiguities in pose.

In this chapter, we have validated our approach for human pose classification with a numerical evaluation. In the next chapter, we will evaluate the performance of our cascade classifiers in a detection framework.

7

Human Localization and Pose Estimation

7.1 Introduction

In this chapter, we continue the study of the proposed multi-class pose classifier and explore its use for joint human localization and pose estimation in a detection framework.

Each hierarchical cascade in the ensemble can make a decision and efficiently reject negative candidates by only sampling a few features of the available feature space. This makes our classifier more suitable for sliding window detectors than Random Forests (RF) or Support Vector Machine (SVM) classifiers. In RF, all the trees have to be completely traversed to produce a vote while SVM classifiers require an entire feature vector to be extracted for each tested window. We thus expect a faster detection with our ensemble of cascades. Additionally, we will also analyze two of our classifier properties that allow a considerable speed-up of the pose detection.

We have carried out an exhaustive experimentation to validate our approach with a numerical evaluation (using different publicly available training and testing datasets) and present a comparison with state-of-the-art methods for 2 different levels of analysis: human detection and human pose estimation (with body joints localization) performances with both fixed and moving cameras.

This chapter is organized as follows. Section 7.2 shows how our pose classifier can be used for joint detection and pose estimation. In Section 7.3, we discuss the two particular properties of our algorithm while a numerical evaluation of our pose detector is presented in Section 7.4 for both localization and pose estimation.

7.2 Human Pose Detection

Since we learn a classifier that is able to discriminate between very similar classes, we can also tackle localization. Given an image I, a sliding-window mechanism then localizes the individual within that image and, at each visited location (x, y) and scale s, a window $I_{\mathbf{p}}$ can be extracted and classified by our multi-class pose classifier R obtaining $\mathbf{O}_{\mathbf{p}} = \left(O^{\omega}_{\mathbf{p}}\right)^{n_\omega}_{\omega=1}$ where $\mathbf{p} = (x, y, s)$, is a location vector that defines the classified window.

A multi-scale saliency map $\mathcal{M}$ of the classifier response can be generated by taking the maximum value of the distribution for each classified window $I_{\mathbf{p}}$:

$$\mathcal{M}(x, y, s) = \max_{\omega\in\Omega}(\mathbf{O}_{\mathbf{p}}) = \max_{\omega\in\Omega}\left((O^{\omega}_{\mathbf{p}})^{n_\omega}_{\omega=1}\right). \tag{7.1}$$

A dense scan (trained on pose only) is shown in Fig. 7.1a where many isolated false positives appear where the classifier responds incorrectly.

Zhang et al. [2007a] tackle joint object detection and pose estimation by using a graph-structured network that alternates the two tasks of foreground/background discrimination and pose estimation for rejecting negatives as quickly as possible. Instead of combining binary foreground/background and multi-class pose classifiers, we propose to perform the two tasks simultaneously by including hard background examples in the negative set T_b^- (see Fig. 6.6 and Sect. 6.3.2) during the training of our rejectors. To create the hard negatives dataset, the classifiers trained on pose images only can be run on negative examples (e.g. from the INRIA dataset [Dalal and Triggs, 2005]) and strong positive classifications are incorporated as hard negative examples (see details in Sect. 7.4.1). Repeating the process of hard negative retraining several times helps to refine the classifiers as can be appreciated in Fig. 7.1 b and c.

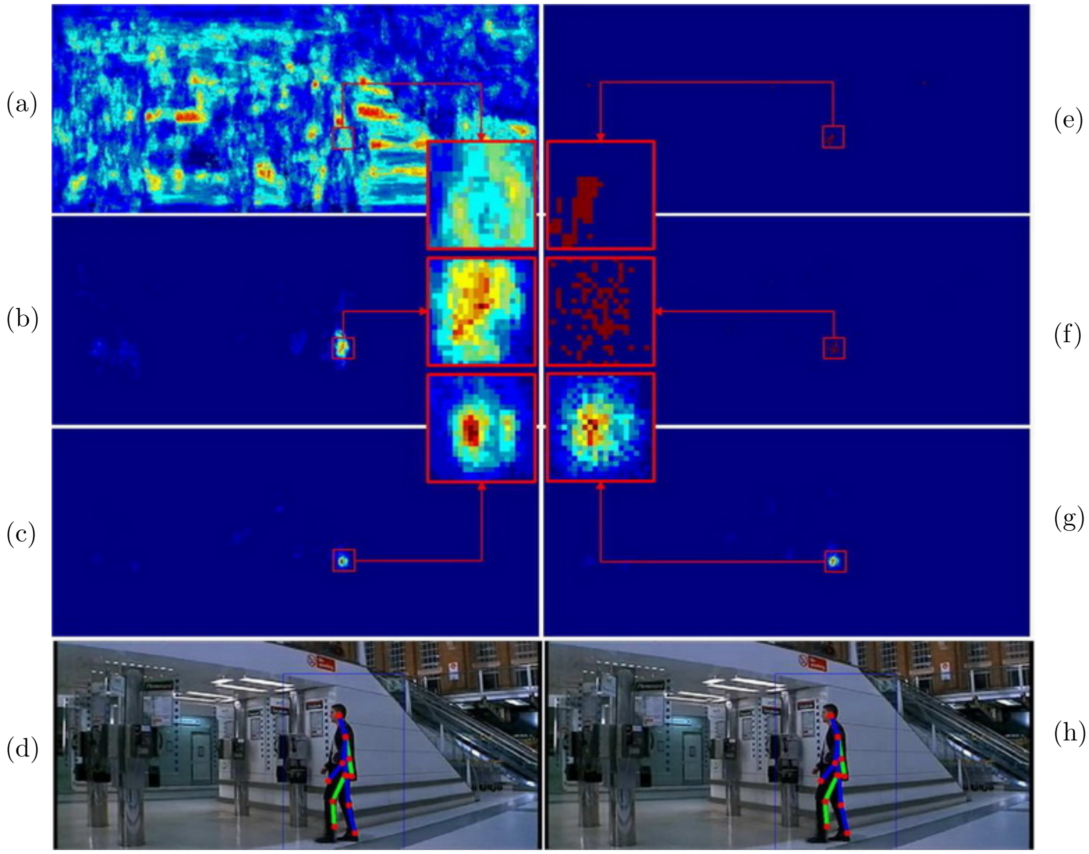


Figure 7.1: Cascade classifier response for a dense scan. *Left side, from top to down*: saliency map $\mathcal{M}$ after 0 (**a**), 1 (**b**) and 3 (**c**) passes of hard negatives retraining (i.e. adding hard negative examples in the cascade learning process). For visualization purpose we represent $\mathcal{M}$ for ground truth scale (i.e. $s = 0.6$) and show a zoom around the ground truth location. (**d**) input image with the pose corresponding to the peak in (**c**). *Right side, from top to down*: saliency map $\mathcal{M}$ obtained with a 1-cascade (**e**), 1 cascade randomized on-line (**f**) and a 100-cascade ensemble randomized on-line (**g**), all after 3 passes of hard negatives retraining. In (**h**) we show the pose corresponding to the peak in (**g**).

By generating dense scan saliency maps for many images, we have found that humans tend to have large "cores" of high confidence value as in Fig. 7.1c. This means that a coarse localization can be obtained with a sparse scan and a local search (e.g. gradient ascend method) can then be used to find the individual accurately. In the example proposed in Fig. 7.1, taking the maximum classification value over the image, (after exploring all the possible positions and scales) results in reasonably good localization of a walking pedestrian. In that case, we have:

$$\mathbf{p}^* = (x^*, y^*, s^*) = \underset{(x,y,s)}{\operatorname{argmax}}(\mathcal{M}(x, y, s)), \tag{7.2}$$

and the corresponding distribution over poses $\mathbf{O}^* = \mathbf{O}_{\mathbf{p}^*} = \left(O^{\omega}_{\mathbf{p}^*}\right)^{n_\omega}_{\omega=1}$.

Once a human has been detected and classified, 3D joints for this image can be estimated by weighting the mean poses of the classes resulting from the distribution using the distribution values as weights, or regressors can be learnt as in [Okada and Soatto, 2008]. The normalized 2D pose is computed in the same way and re-transformed from the normalized bounding box to the input image coordinate system obtaining the 2D joints location (see Fig. 7.1d).

7.3 Properties of the Random Cascades Classifiers for Pose Detection

In addition to the advantages stated so far, our classifier exhibits some additional interesting properties for pose detection. Since the tree structure $\mathcal{S}$ and HOG block rejectors list $\mathcal{B}$ remain fixed after training, a new cascade classifier $\widetilde{r_k}(I_N, \widetilde{\Phi}_k)$ can be constructed *on-line* by simply creating a new random vector $\widetilde{\Phi}_k$. This means that for each classified window $I_\mathbf{p}$ in an image, a different ensemble $R_\mathbf{p}$ can be regenerated instantly at no extra-cost:

$$R_\mathbf{p}(I_N) = \{\widetilde{r_k}(I_N, \widetilde{\Phi_k})\}^{n_c}_{k=1}, \tag{7.3}$$

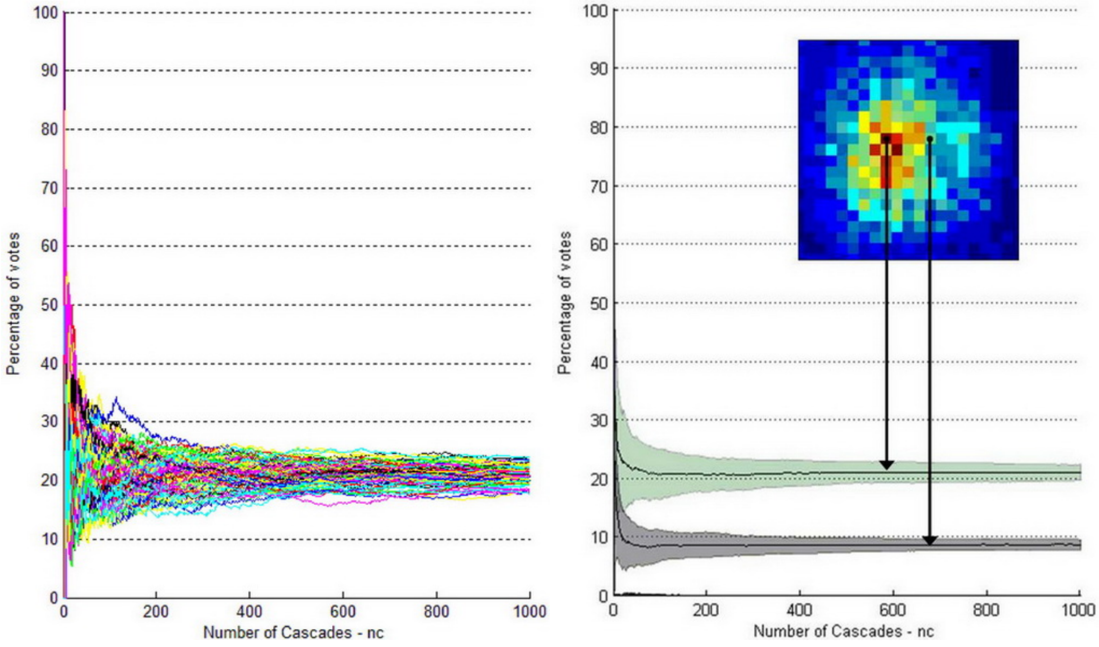
where $\mathbf{p} = (x, y, s)$, is the location vector that defines $I_\mathbf{p}$ and $\widetilde{\Phi_k}$ is sampled using Eq. 6.13.

The qualitative effects of this *on-line randomization* can be appreciated in Fig. 7.1: a dense scan (1-pixel stride) with a 1-cascade classifier (Fig. 7.1e) produces responses which are grouped while randomizing Φ on-line (using a different random cascade at each location) produces a cloud of responses around the ground truth (see Fig. 7.1f). This property will favor an efficient localization at a higher search stride, as verified later by the numerical evaluation presented in Sect. 6.4.2 [1]. When randomizing a 100-cascade classifier (Fig. 7.1g), the saliency map becomes similar to the one obtained with a regular 100-cascade ensemble without on-line randomization (Fig. 7.1d).

Performing construction on-line, each new random vector $\widetilde{\Phi_k}$ contributes to the final distribution. The probability that any given randomly generated cascade classifier $\widetilde{r_k}$ will vote close to an object location increases as the position gets closer to the true location of the object. Even if the classification by the initial cascade for the pose class is wrong, it is generally close to an area where the object is. Subsequent classifications from other randomized rejectors push the distribution toward a stable result. As more classifications are made, the distribution converges and becomes stable as shown in Fig 7.2a and Fig 7.2b where we can observe the higher variability in classification with few cascades compared to the one obtained using bigger ensembles. This explains the similarity of the saliency map obtained with and without on-line randomization (Fig. 7.1) when considering a large number of cascades in the ensemble R.

This convergence property can also be exploited for fast localization using *cascade thresholding*: a cascade r_k is drawn at random (without replacement) from R until the

[1] This property may also be exploited to spread the computation of image classification and localization over time. They may also be readily parallelized due to the structure $\mathcal{S}$ and $\mathcal{B}$ being fixed once the classifier has been trained.



(a) (b)

Figure 7.2: On-line Randomization: (**a**) Detection score (in percentage of votes) $\max(\mathbf{O}) = \max(\frac{100}{n_c}\sum_{k=1}^{n_c} o_k^{\omega})$, i.e. the peak of the distribution, obtained when classifying the same location (x, y, s) with 100 different randomized ensembles. When increasing n_c, the number of cascades in the ensemble R, all the curves converge to 20%. In (**b**), we present the average and std across all these curves for 2 different locations in the image: ground truth location and 5 pixels away from ground truth.

aggregated score reaches a sufficient confidence level. The confidence level can be selected based on a desired trade-off between speed and accuracy. Another possibility is to consider an *adaptive cascade filtering* by classifying with the entire ensemble R ($R_{\mathbf{p}}$ if on-line randomization is considered) only the locations $\mathbf{p} = (x, y, s)$ that yield a detection using a single or few cascades from the ensemble, i.e. filtering with the first n_f cascades of the ensemble:

$$\mathcal{M}(x, y, s) = \begin{cases} \max_{\omega \in \Omega} \left(O_{\mathbf{p}}^{\omega} \in \mathbf{O}_{\mathbf{p}}\right) & \text{if } S_{n_f} > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{7.4}$$

where the detection score S_{n_f} is basically:

$$S_{n_f} = \max_{\omega \in \Omega} \left(\sum_{k=1}^{n_f} o_k^{\omega} \right). \tag{7.5}$$

In other words, if a strong vote for background has been made with the first n_f cascades (i.e. no vote for any human pose classes and $S_{n_f} = 0$), then the classifier stops classifying with the rest of the cascades in the ensemble. Localizing using this approximate approach means that a classifier with a few cascades can be used as a region of interest detector for a more dense classification. The *adaptive cascade filtering* combined with *on-line randomization* will produce an efficient and fast detection, as verified later in Sect. 7.4.1.1.

7.4 Experiments

7.4.1 Localization Results using Mobo Dataset for Training

To construct our localization dataset, we took images from several different datasets to compare each of the algorithms in different environments and at different subject scales. These images were taken from HumanEva [Sigal et al., 2010], CamVid [Brostow et al., 2008], INRIA [Dalal and Triggs, 2005], some images of pedestrians collected from movie sequences, and some images captured from a web camera in a simple lab environment. See Fig. 7.3 for some selected images from this dataset. We then manually annotated each of these 120 images so we could determine localization accuracy.

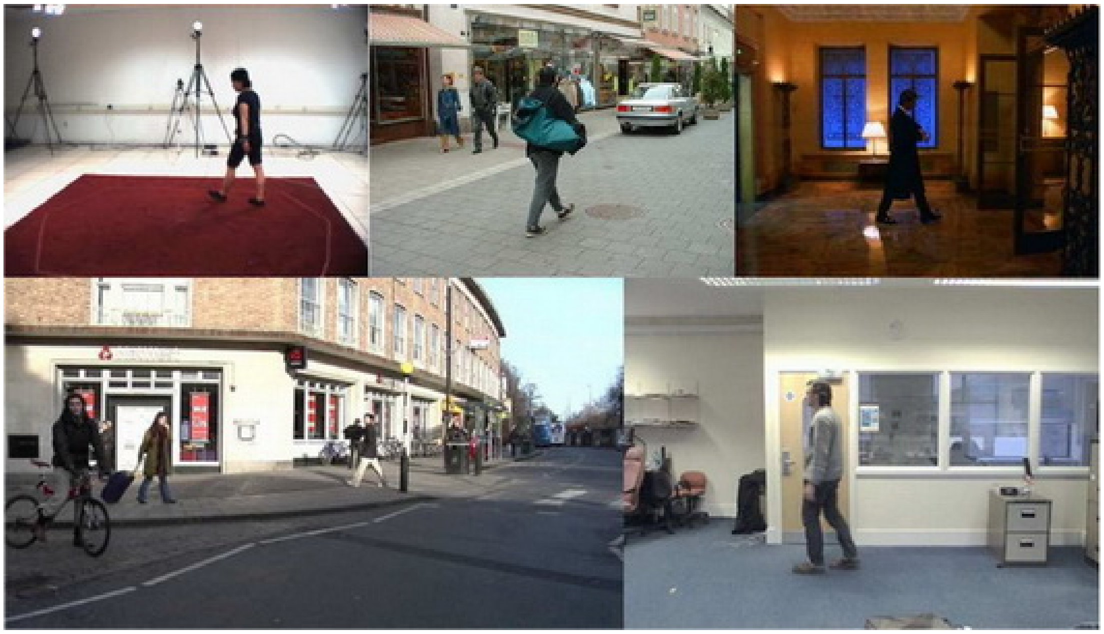


Figure 7.3: Localization dataset: Clockwise from top left: HumanEva, INRIA, movie sequences, lab sequence and CamVid.

For a fair comparison, we have implemented our cascades, the multi-SVMs and RF classifiers in the same C++ framework with an efficient on-demand feature extraction system. In the original HOG implementation [Dalal and Triggs, 2005], multi-scale searches required the input image to be re-scaled at every searched scale and scanning window resolution before extracting HOG features for that scale. We believe that different scales can be explored by keeping the source image size fixed and reusing the integral histograms. The classifier window is rescaled to the equivalent scale (1/s) and the Integral HOG features are resized with respect to the new window size. Since the integral histogram sampling consists of only 4 coordinate samples for each cell considered, independent of the size of the HOG feature, the features can be scaled with respect to the new window size at no additional computation time, thus allowing different scales to be explored at approximately the same computational cost (See Fig. 7.4).

SVMs, RF and cascade classifiers trained on MoBo images in Sect. 6.5 were then run on the localization dataset over 6 discrete scale setting ($[0.3, 0.45, 0.75, 1.0, \quad 1.5, 2.0]$) with a 4 pixels stride, thus exploring and classifying a total of $10,599,042$ sub-images. We used a state-of-the-art laptop with an Intel Core @ 1.73GHz. A non-maximum suppression step is used to merge

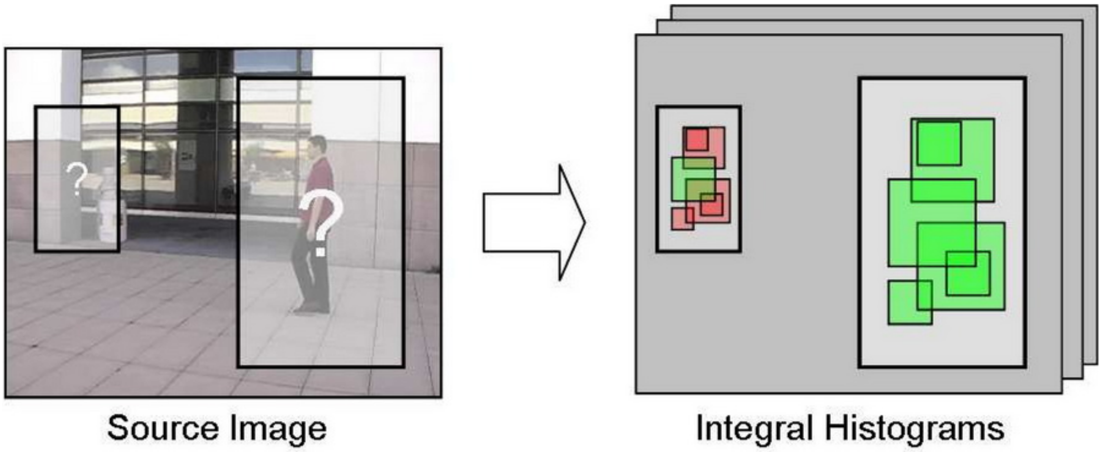


Figure 7.4: HOG feature scaling: different scales are queried by rescaling the classifier window, and HOG feature boxes are rescaled in proportion to the new classifier window scale. The sampling coordinates for the HOG boxes can be rescaled at no extra cost in computation time.

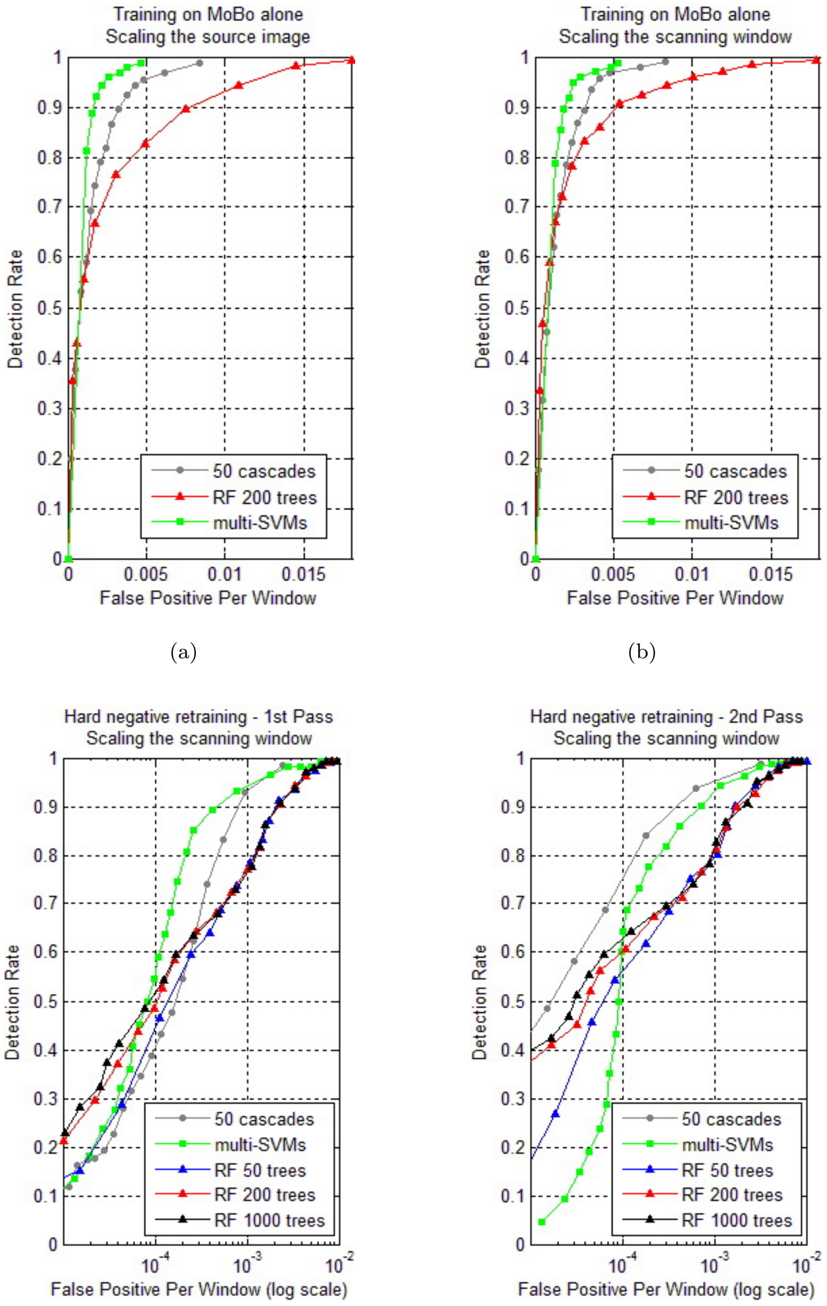
nearby detections to one final hypothesis. In the first experiment, we compare accuracy and processing time using the 2 different scanning schemes, i.e. scaling the input image or scaling the sliding window. As can be seen in Fig. 7.5a and Fig. 7.5b, the results are very similar with the 2 different scanning methods that do not seem to affect the detection rate much. The target architecture for the experiments is an AMD/Intel x86 processor, and although the results show that scaling the window is much faster (see Table 7.1) on a single thread implementation on a system with local caches, this comparison is architecture dependent[2].

Table 7.1: Detection times (after hard negative retraining) for the experiments reported in Fig. 7.5.

Hard negatives	scaling	SVMs	RF (trees)			Cascades
			50	200	1000	$n_c = 50$
Pose only	image	15h03	-	8h37	-	19h38
	window	11h45	-	2h03	-	18h36
1^{st} Pass	image	20h30	7h19	7h42	8h12	12h48
	window	10h55	1h11	1h33	3h07	5h07
2^{nd} Pass	window	11h08	1h12	1h32	3h56	5h09

Each of the classifiers were re-trained on human subjects from the aligned Mobo dataset, and combined with hard background examples from the INRIA dataset [Dalal and Triggs, 2005]. To create the hard examples dataset, the classifiers trained on pose images only from Sect. 6.5 were run on negative examples from the INRIA dataset [Dalal and Triggs, 2005] (with a 4 pixels stride and the same 6 scales), and strong positive classifications were incorporated as hard examples. Hard background examples were included in the negative set T_b^- (see Fig 6.6) during the training of each rejectors of our cascades. They were added to SVMs and RF classifiers as a background class so that there are a total of 65 classes; 1 for background and

[2] As noted by Sugano et al [Sugano and Miyamoto, 2007] on a parallel SIMD (Single Instruction Multiple Data) implementations scaling the image can be more suitable for parallel processing and actually negate the additional impact of resizing the image to different scales.



(a)

(b)

(c)

(d)

Figure 7.5: Detection results on the combined localization dataset using different classifiers trained on Mobo images only (**a** and **b**), trained with a first pass of hard negatives from INRIA (**c**) and with a second pass (**d**). For the 1^{st} case, we present results for 2 different types of multi-scales scanning scheme: scaling the input image and keeping the size of the scanning windows fixed (**a**) and scaling the scanning window and keeping the size of the input image fixed (**b**), while the other 2 graphs (**c** and **d**) are obtained with the second scanning scheme. Note that a log scale is used for **c** and **d** to aid visualization.

64 for pose. We repeat the process of hard negatives retraining several times to refine the classifiers. The detection rates for the three types of classifiers after hard negatives retraining are given in Fig. 7.5c and Fig. 7.5d while the corresponding classification times are shown in table 7.1. We can observe that the 1^{st} pass of hard negatives retraining helps to improve the accuracy as the FPPW rate (False Positive Per Window) is reduced by a factor of 3. Adding more trees to the RF classifier does not improve the performances (see Fig. 7.5c and Fig. 7.5d). Both SVMs and RF classifiers reach their best performances after one hard negative retraining while our cascade classifier outperforms them after the 2^{nd} pass (see Table 7.2). The reference point of 1×10^{-4} FPPW is arbitrary but is a reasonable comparison point given that it has been used in other detection works such as [Dalal and Triggs, 2005]. At this rate we achieve a better detection rate (73.5%) than both multi-SVMs and RF classifiers (62.79% and 62.53% respectively) at the same FPPW rate, but if a higher rate of 5×10^{-4} is acceptable, then we achieve over 90% accuracy as illustrated in Fig. 7.5d. However the Random Forests are still faster than this configuration of cascade classifier (see Table 7.1).

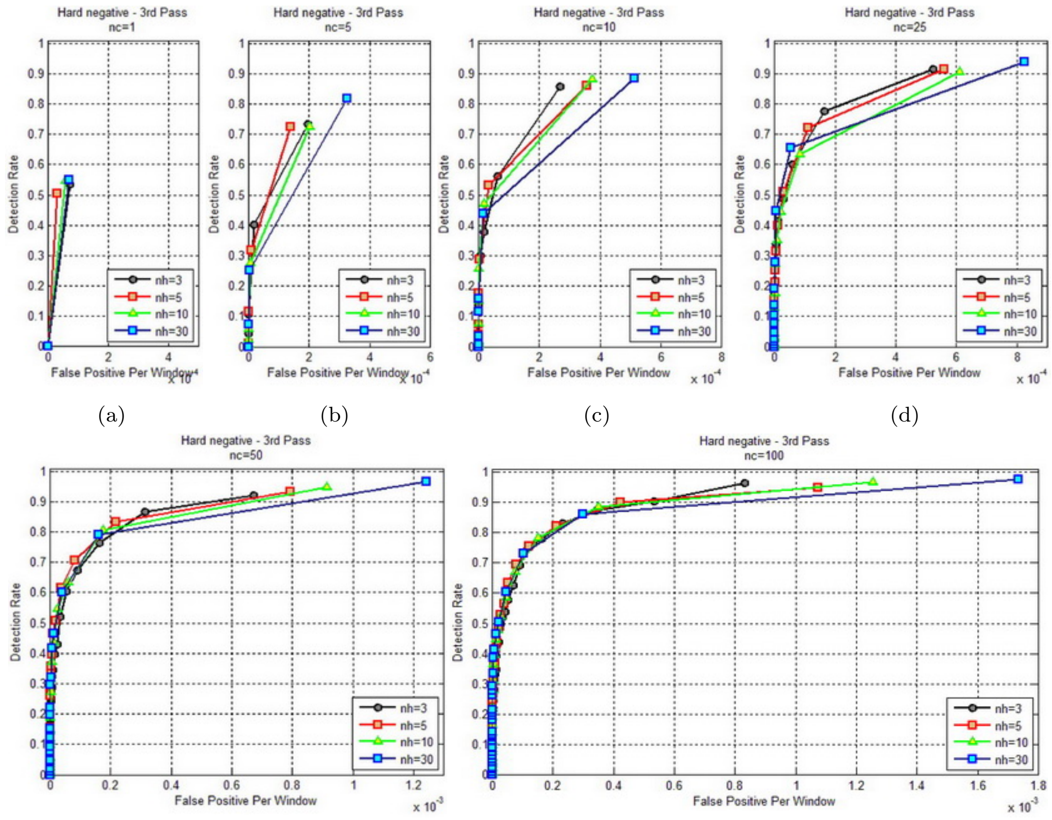
Table 7.2: Detection rates at 1×10^{-4} FPPW and 1×10^{-3} FPPW for the experiments reported in Fig. 7.5.

Hard negatives	FPPW	SVMs	RF (trees)			Cascades
			50	200	1000	$n_c = 50$
Pose only	1.10^{-4}	6.21	-	**12.11**	-	7.81
	1.10^{-3}	**62.04**	-	59.86	-	54.46
1^{st} Pass	1.10^{-4}	**54.89**	43.29	49.03	51.93	40.37
	1.10^{-3}	**94.12**	76.82	76.64	76.04	92.97
2^{nd} Pass	1.10^{-4}	62.79	56.13	60.31	62.53	**73.5**
	1.10^{-3}	92.35	78.21	79.67	80.80	**94.53**

The previous experiments have been carried out using an ensemble made of 50 cascades built from the best $n_h = 30$ HOG rejectors at each branch. In Fig. 7.6, we show how the number of cascades n_c and the number of HOG blocks n_h we sample from affect the detection and FFPW rates, while the corresponding classification times are shown in Fig. 7.7. If we consider a classifier with 10 or less cascades we can obtain a better and faster classification than with the best RF (less than an hour). Adding more cascades allows the hierarchical cascade classifier to reach higher detection rates but at a higher cost in terms of FPPW, while sampling from a smaller subset of HOGs (i.e. using a smaller n_h) makes the classifier more strict. Constraining the features that our cascades can randomly draw from impairs generalization performances and considerably affects pose classification rates as demonstrated in Fig. 6.16 and Fig. 6.17. A trade-off thus has to be made between the computational speed, the efficiency in detection and the pose classification accuracy. One approach to this is to localize using an ensemble built from a small pool of HOG blocks and then refine the pose classification of a few selected locations with additional cascades grown by sampling from a larger pool of rejectors.

7.4.1.1 On-line Randomization and Filtering

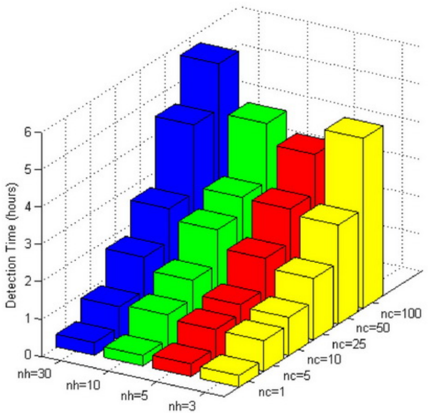
In addition to the advantages stated in previous sections, our classifier has two very interesting properties that have not been evaluated yet. First, the on-line randomization of Φ_k which allows the generation of a different classifier $r_k(I_N, \Phi_k, \mathcal{S}, \mathcal{B})$ at each location considered (i.e. for each sub-image to be classified) at no extra-cost. The qualitative effects of the on-line randomization on the saliency map can be appreciated in Fig. 7.8 where we show the result of a dense scan (1-pixel stride) with several different cascades classifier. A 1-cascade classifier



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

(e) (f)

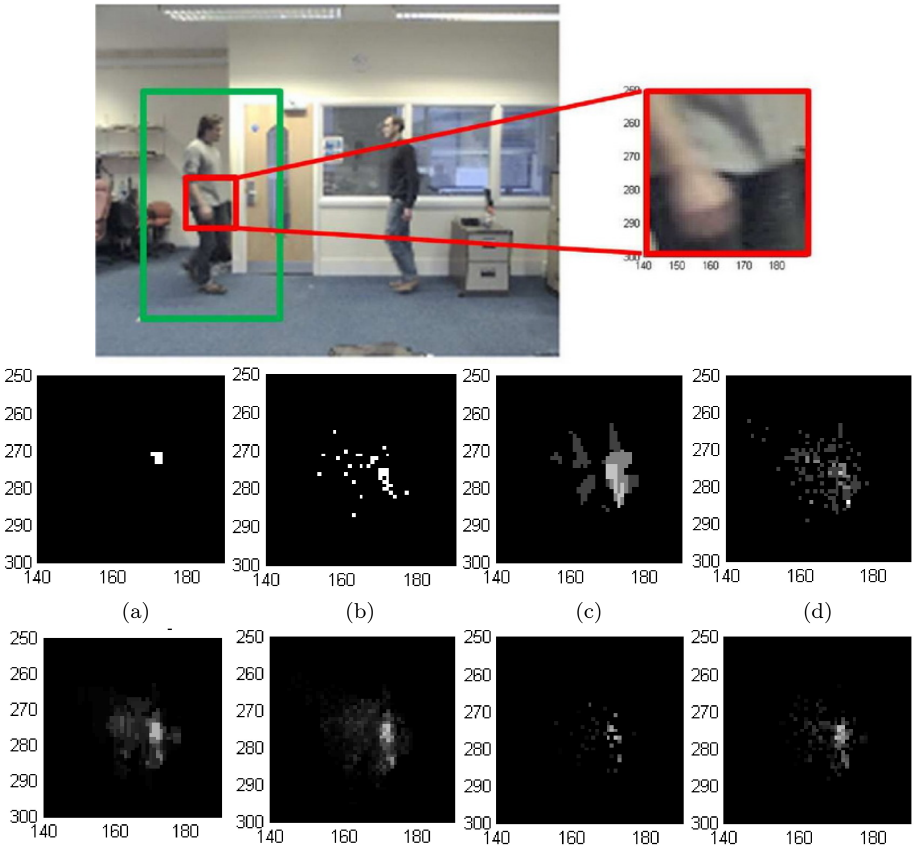
Figure 7.6: Detection results for different configurations of the cascades classifiers with a 3^{rd} pass of hard negative retraining: we present the results with respectively 1 (**a**), 5 (**b**), 10 (**c**), 25 (**d**), 50 cascades (**e**) and 100 cascades (**f**). For each case we vary the number n_h of HOG blocks we sample from at each branch of the tree. The corresponding classification times are shown in Fig. 7.7.



n_c	n_h			
	3	5	10	30
1	0h19	0h20	0h17	0h20
5	0h49	0h50	0h57	0h53
10	1h01	1h05	1h27	1h47
25	1h39	1h54	2h23	2h40
50	2h39	2h48	2h50	4h29
100	4h33	3h50	4h23	5h42

Figure 7.7: Detection time when scanning the entire localization dataset for different configurations of the cascades classifiers after a 3^{rd} pass of hard negative retraining (experiments reported in Fig. 7.6).

(Fig. 7.8a) produces few responses which are grouped and more likely to be missed at a higher search stride while using a different cascade at each location (Fig. 7.8b) produces a cloud of responses that favors a good detection at a higher search stride. With a 5-cascade classifier, the benefit of the randomization is less immediately obvious (Fig. 7.8c and Fig. 7.8d). Though this property is less useful with a 100-cascade classifier (Fig. 7.8e and Fig. 7.8f).



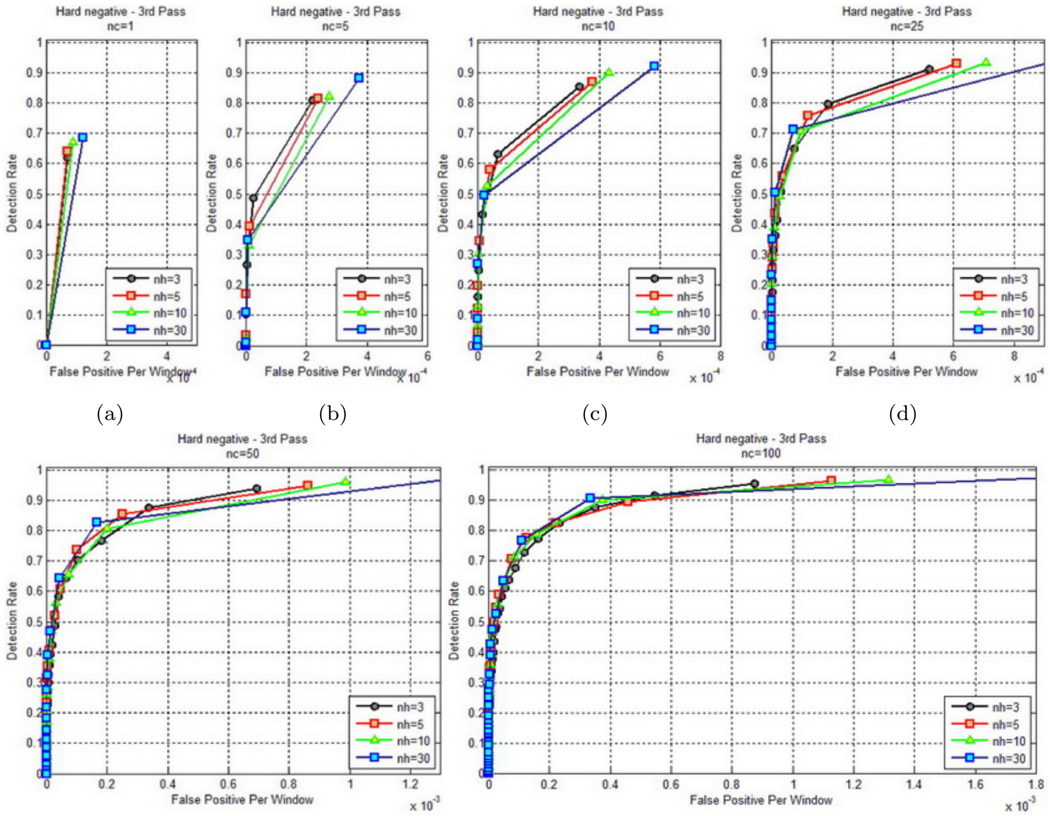
(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7.8: On-line randomization and filtering: the picture shown in the upper part with a ground truth box (green) is scanned at a 1-pixel stride. We present the saliency maps for a region of interest around the ground truth location (red box) obtained with different cascade classifiers: 1-cascade (**a**), 1-cascade randomized(**b**), 5-cascade (**c**), 5-cascade randomized (**d**), 100-cascade (**e**), 100-cascade randomized (**f**), 100-cascade randomized and filtered with 1st cascade (**g**) and 100-cascade randomized and filtered with the first 5 cascades (**h**). Processing times of the entire image are respectively: 13.27s (**a**), 13.25s(**b**), 31.66s(**c**), 28.53s(**d**), 114.52s(**e**), 103s(**f**), 9.34s(**g**) and 23s(**h**).

To evaluate quantitatively the benefit of this property, we run the same test on the localization dataset, varying the number n_c of cascades and number n_h of HOGs, but this time randomizing the vectors Φ_k at each location. The results are presented in Fig. 7.9. As expected, if we compare with Fig. 7.6, we can see that the on-line randomization has a great influence for classifiers with few cascades: the maximum detection rate increases by about 10% for a 5-cascade classifier while it increases by over 15% for a single cascade classifier.

Single hierarchical cascades have good approximate localization performance (see Fig. 7.6), but the disadvantage is that they may misclassify locations that would have otherwise been correctly classified using more cascades. This leads to the second property of our algorithm:



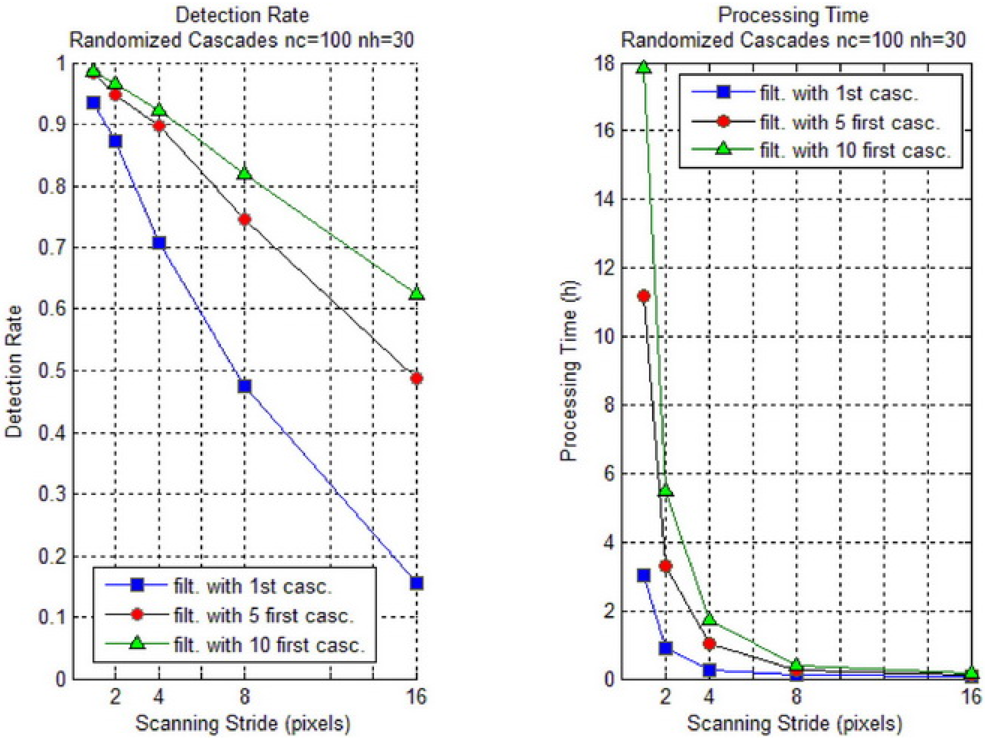
(a) (b) (c) (d)

(e) (f)

Figure 7.9: Detection results for different configurations of the cascades classifiers with a 3^{rd} pass of hard negative retraining and on-line randomization: we present the results with respectively 1 (**a**), 5 (**b**), 10 (**c**), 25 (**d**), 50 cascades (**e**) and 100 cascades (**f**). For each case we vary the number n_h of HOG blocks we sample from at each branch of the tree.

we can adapt the number of cascades in the classifier on-line, and thus, each sub-image can be classified using a different number of cascades. Locations that yield a detection using a few cascades can be classified with more trees until the detection result converges. We will leave for future work the study and optimization of the convergence criteria for the *cascade thresholding* approach as the detection of this convergence requires a minimum number of cascades to be applied. However, since the goal is to create a classifier that can make a very fast detection using as few cascades as possible, we instead exploit the *adaptive cascade filtering* property for efficient localization: we will use the first n_f cascades of the classifier as a filter and classify with the rest of the cascades only if a positive vote has been made. A qualitative analysis can be made from Fig. 7.8: in the presented example, we can see that filtering with the first (g) or the first 5 cascades (h) of a 100-cascade classifier leads to a decent distribution, almost as good as the ones obtained without any filtering (e and f) but it is much faster (between 5 and 10 times faster for that example).

We have evaluated the combination of these 2 properties by running a series of experiments to select the best cascade classifier: we evaluated detection rates and processing times for different search strides (1, 2, 4, 8 and 16 pixels) using a 100-cascade classifier filtered with the first, the first 5 and the first 10 cascades. Quantitative results are reported in Fig. 7.10a

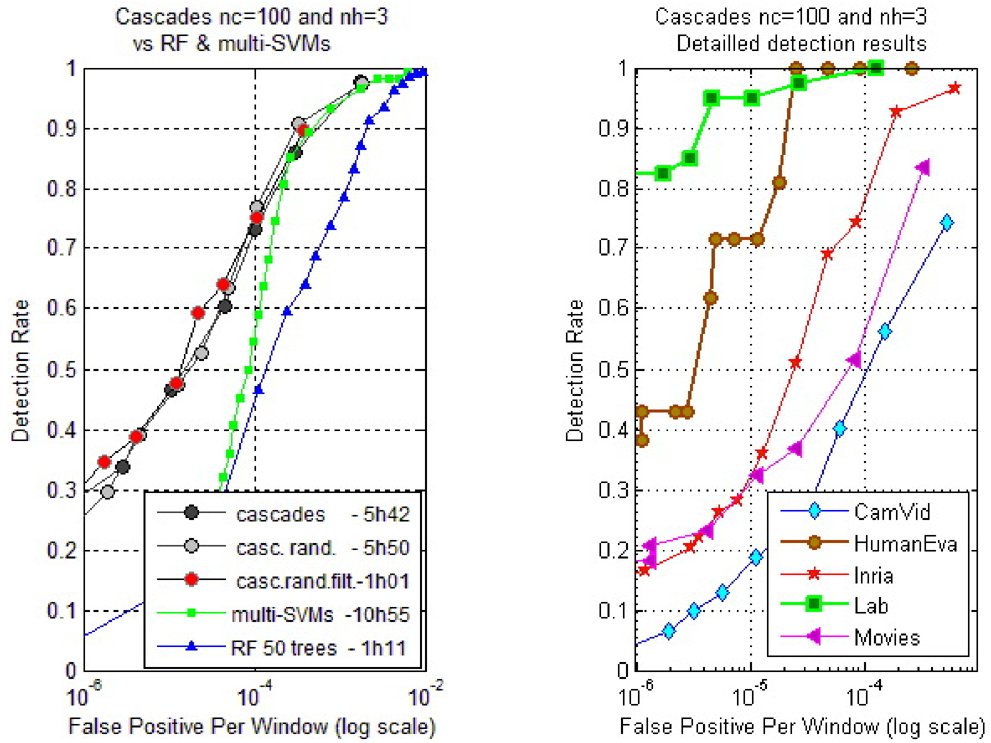


(a) (b)

Figure 7.10: Selection of the best cascade classifier: maximum detection rate varying search stride and number of cascades used for filtering (**a**), corresponding processing times are given in (**b**). Classifying every 4 pixels and filtering with the first 5 cascades offers a reasonable compromise between accuracy (maximum detection rate of 90%) and computational cost (1h01).

and Fig. 7.10b. As expected, using a denser search (i.e. using a smaller stride) considerably improves the detection rate but it also increases the processing time up to 3 or more hours. The same observation can be made concerning the number of cascades used for filtering: more cascades also increase the detection rates for any search strides but at a higher computation cost. Classifying every 4 pixels and filtering with the first 5 cascades offers a reasonable compromise between accuracy (maximum detection rate of 90%) and computational cost (1h01 i.e 35.5 seconds per image). Our selected cascade classifier performs slightly better than the best multi-SVMs classifier but is 10 times faster. It is also slightly faster than the best RF classifier (1h11) but considerably outperforms it in classification (see Fig. 7.11a). Two examples of multiple detections are presented in Fig. 7.12.

A detailed analysis of the performances of our classifier (see Fig. 7.11b) shows that the best detection rates are obtained for the HumanEVA and the Lab subsets: for a FPPW rate of 1×10^{-4} or higher there is no misdetection. This makes perfect sense since our MoBo training data has been captured in a similar lab environment and the background is less complex for this 2 subsets. The drop in the classifier performances with INRIA images can be explained by their richer backgrounds, and are more difficult to correctly classify. The lower performances on the remaining 2 subsets, the Camvid and Movies, can be explained by the fact that the images also present richer backgrounds and have been captured with a moving camera. More importantly, the humans are much more difficult to detect than in the other subsets. Indeed, the Camvid



(a) (b)

Figure 7.11: Detection rate of the selected cascade classifier (100 cascades filtered with the first 5 cascades) compared to other classifiers for a 4-pixels search stride (**a**) and corresponding detailed detection rates over the 5 different types of images in our localization dataset (**b**).

and Movies subsets present images with street views where people wear coats, dresses, hats and all sorts of clothing completely absent in our lab-type training dataset. Some severe occlusions also occur in these uncontrolled environments and the resolution is often not fine enough to give accurate results. All these observations are visually confirmed with the examples of pose detection errors, see the examples of False Positives (Fig. 7.13 and False Negatives (Fig. 7.14).

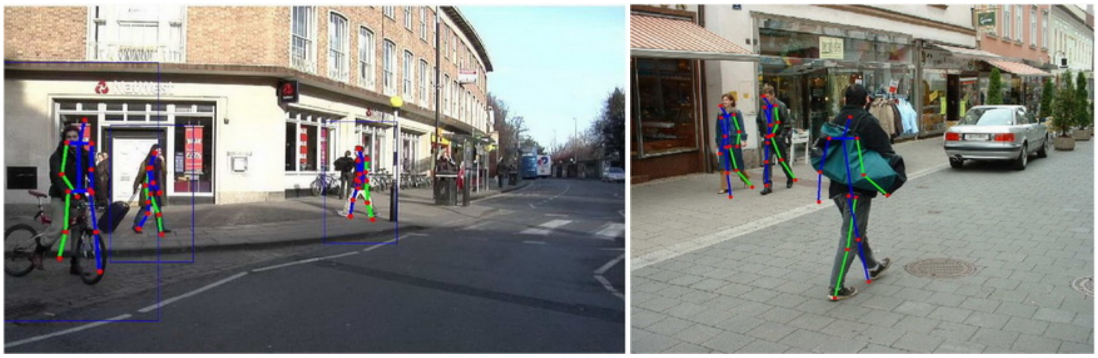


Figure 7.12: Detection results: we present 2 examples of multiple multi-scale detections in the same image.

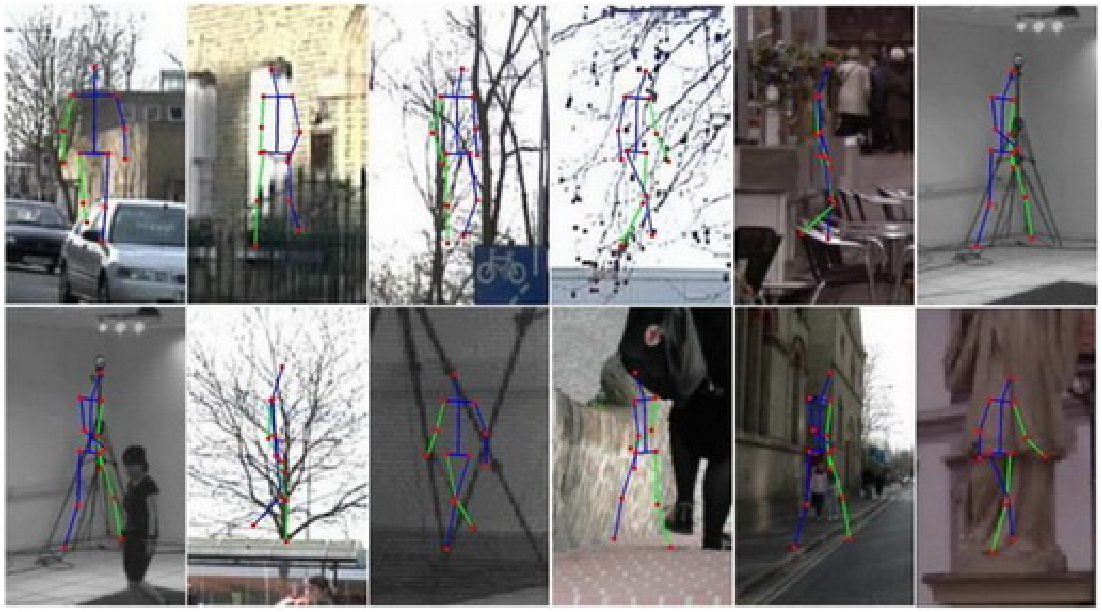


Figure 7.13: Cascades detection errors - False Positives: we can observe that the presence of many edges in textured areas and trees make these regions difficult to classify as background.

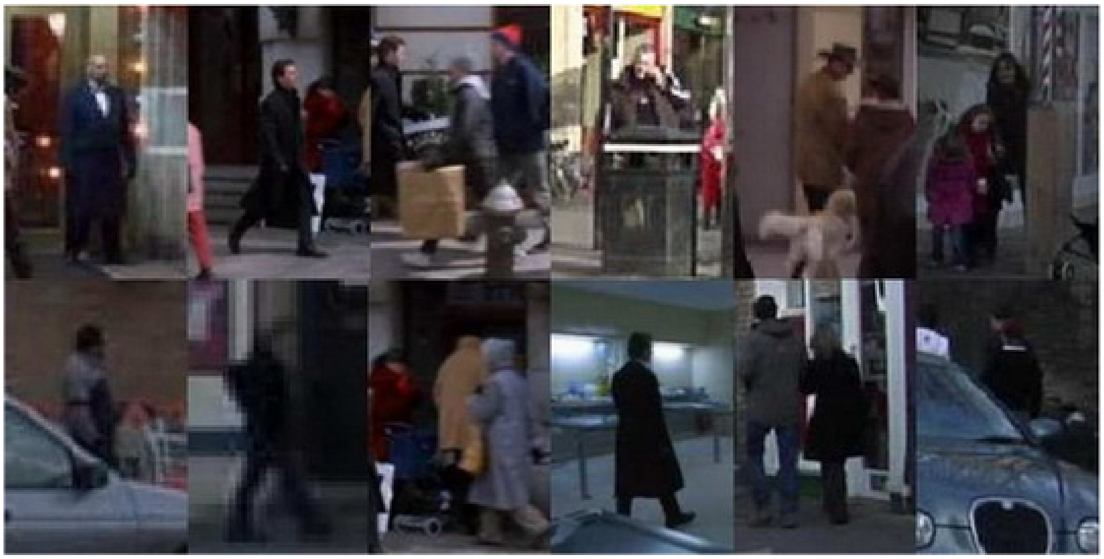


Figure 7.14: Cascades detection errors - False Negatives: we can observe that the clothing, occlusions and low resolution make the detection harder for some subjects in Camvid and Movie subsets.

Finally, we present some examples of good detections for the 5 different subsets in Fig. 7.15. For each detection, the represented pose corresponds to the "winning class" i.e. the highest peak in the pose distribution. As we can see, a correct detection does not always returns a correct pose classification. In some cases, the pose is not well-recognized because it does not belong to our training dataset and does not correspond to any of our classes (e.g. walking with hands in pockets). In other cases, we believe that a proper analysis of the pose distribution would probably improves the accuracy in terms of pose but we will leave it for future work.

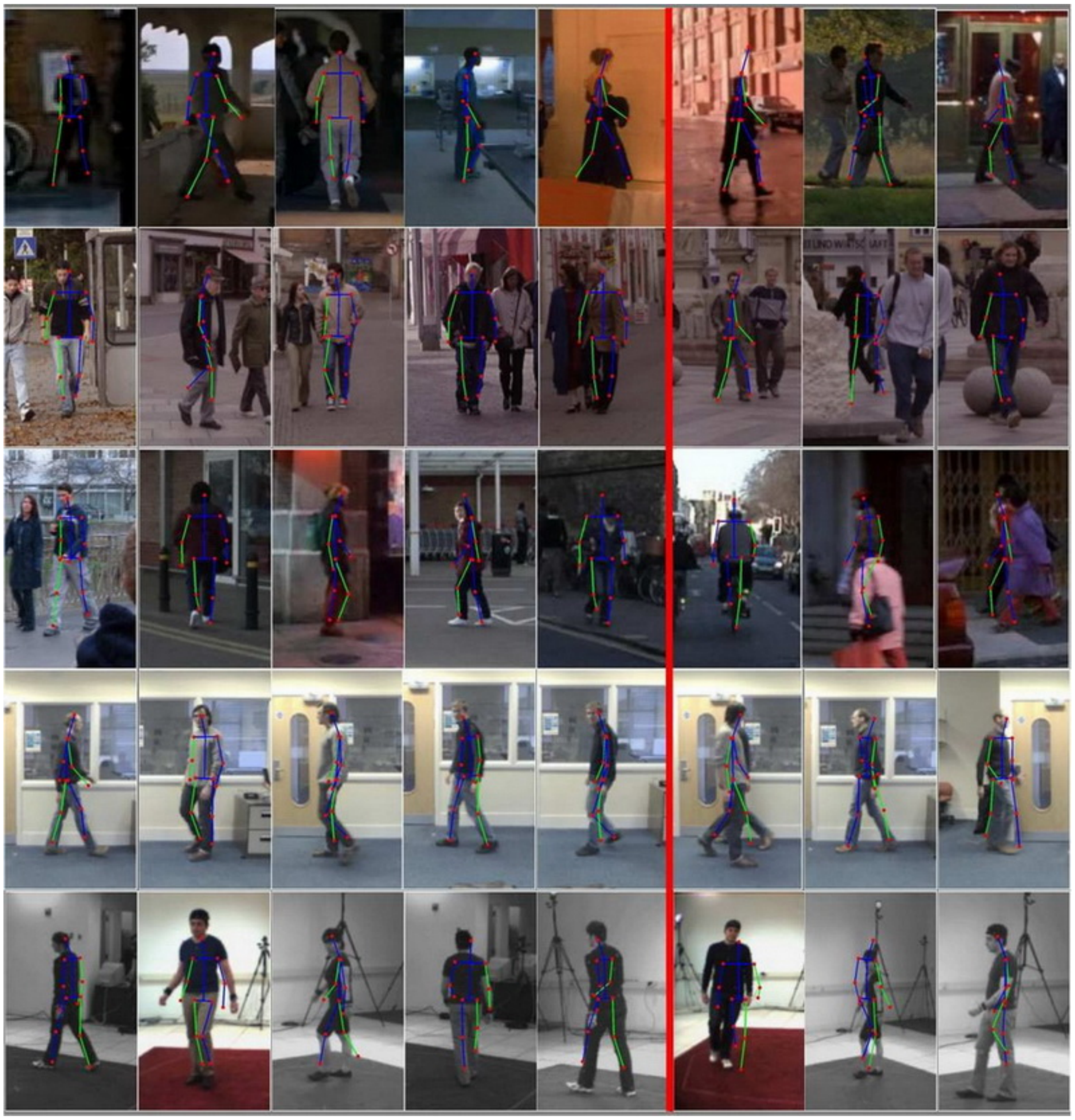


Figure 7.15: Cascades pose detections - True Positives: Each row corresponds to one of the testing sequences (*from top to down*): Movies, INRIA, Camvid, Lab and HumanEVA. For each sequence, we show 5 examples of correct pose classification (*left*) and 3 examples of wrong pose classification (*right*). For each presented frame, we present the normalized 96 × 160 image corresponding to the highest values of the saliency map obtained when applying the pose detector. The pose corresponding to the "winning class" i.e. the highest peak in the distribution is represented on top of the cropped image.

7.4.2 Pose Estimation in Video Sequences

Our pose detector performs well with isolated static images. By extension, it should perform similarly with video sequences, as each frame can be treated as an isolated static image. We will evaluate the pose estimation performances in videos captured from a moving or fixed camera. For the evaluation of the pose estimation performances, we will consider the multi-class cascade classifier that has been trained on HumanEVA I dataset (in Sect. 6.4.1). The torus manifold offers better properties for tracking and visualization of the distributions over poses.

7.4.2.1 Numerical Evaluation using HumanEva Dataset

The pose estimation performances of our classifier are estimated in similar conditions (i.e. indoor with a fixed camera) using HumanEVA II dataset. We apply a simple Kalman filter on the position, scale and rotation parameters along the sequence and locally look for the maxima. We select only the probable classes based on spatio-temporal constraints i.e. transitions between neighboring classes on the torus manifold (see computation of the Probabilistic Transition Matrix in chapter 3 for details). By this process, we do not guarantee to reach the best result but a reasonably good one in relatively few iterations. In Fig. 7.16, we show the pose detection results we obtain for one of the four sequences: for each depicted frame, the estimated 2D pose reprojected on the input image (the pose which is later considered for numerical evaluation), the resulting distributions over the 192 classes on the 3D and 2D representations of the torus manifoldand the distribution after selection (in green) of the most probable classes based on spatiotemporal constraints.

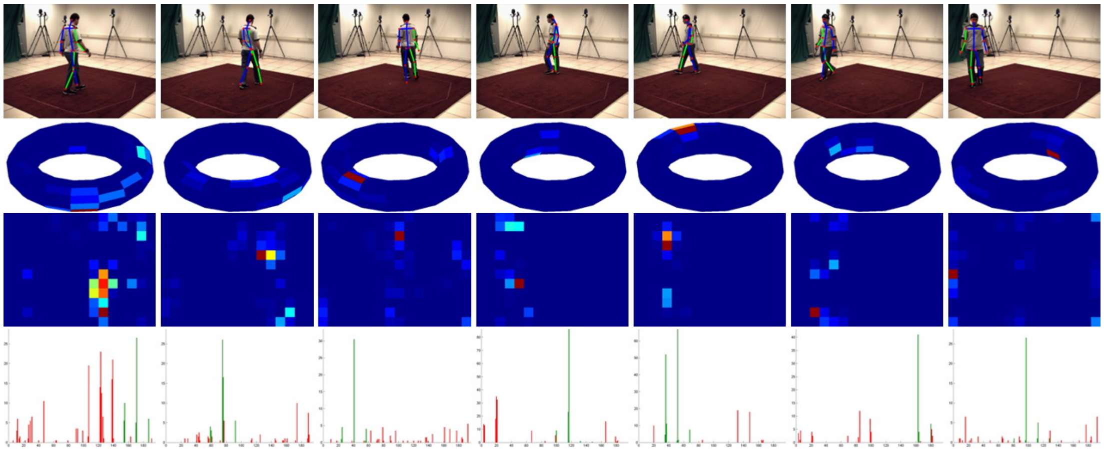


Figure 7.16: Pose detection results on HumanEva II Subject S4-Camera C2. *From top to down*: for each presented frame (2, 50, 100, 150, 200, 250 and 300) the reprojected 2D pose is represented on top of the input image (*top row*). The resulting distributions over the 192 classes after classification using our random cascades classifier is represented on the 3D and 2D representations of the torus manifold. In the bottom row, we show the distribution after selection (in green) of the most probable classes based on spatiotemporal constraints (i.e. transitions between neighboring classes on the torus).

In Fig. 7.17 are depicted the pose detection results obtained for another sequence. This time, for each frame, we show the cropped 96 × 160 image corresponding to the peak obtained when applying the pose detection in the original input image and the resulting 2D pose corresponding to the “winning class”.

Quantitative evaluation is provided in Tab.7.3 together with the results reported by [Gall et al., 2010, Bergtholdt et al., 2010, Andriluka et al., 2010]. Gall et al. [2010] present the best

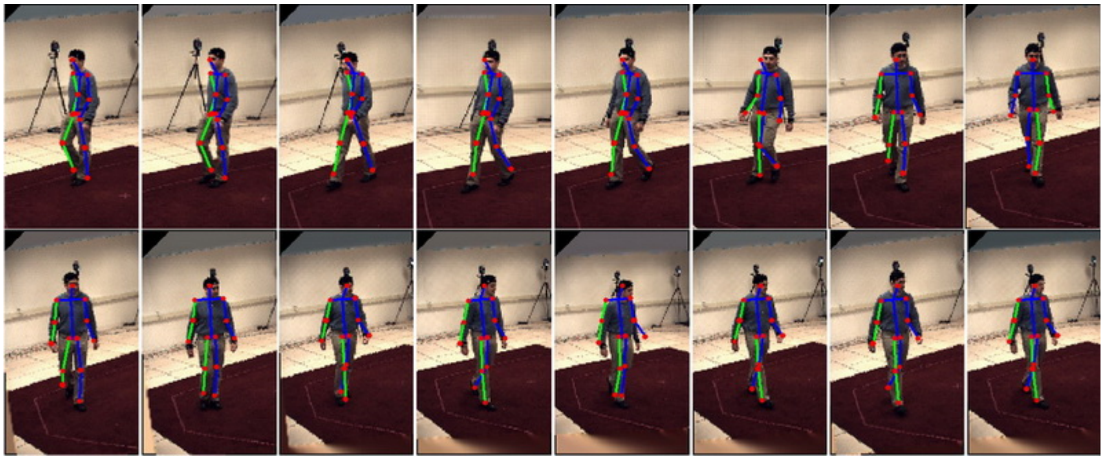


Figure 7.17: Pose detection results on HumanEva II dataset Subject S2-Camera C1: normalized 96 × 160 images corresponding to the peak obtained when applying the pose detection. For each presented frame (1, 50, 100, 150, 200, 250, 300 and 350) the resulting pose is represented on top of the cropped image.

numerical results using a multi-layer framework based on background subtraction with local optimization while Andriluka et al. [2010] use extra training data and optimize over the entire sequence. Our results are obtained training on HumanEVA I only and estimating the pose frame by frame without background subtraction. Note that our pose classifier outperforms the silhouette-based approach we have presented in the chapter 3.

Table 7.3: 2D Pose Estimation Error on HumanEva II dataset: mean (and standard deviation) of the 2D joints location error (in pixels) obtained by other state-of-the-art approaches and our Randomized Cascades.

Subject	S2		S4	
Camera	C1	C2	C1	C2
Frames	$1 - 350$	$1 - 350$	$1 - 290$	$1 - 290$
Chapter 3	16.96 ± 4.83	18.53 ± 5.97	16.36 ± 4.99	14.88 ± 3.44
Gall et al. [2010]	4.10 ± 1.11	4.38 ± 1.36	3.58 ± 0.74	3.35 ± 0.51
Bergtholdt et al. [2010]	25.48 ± 13.18	25.48 ± 13.18	38.82 ± 16.32	38.82 ± 16.32
Andriluka et al. [2010]	10.49 ± 2.70	10.72 ± 2.44	-	-
our approach	12.98 ± 3.5	14.18 ± 4.38	16.67 ± 5.66	13.03 ± 3.49

7.4.2.2 Moving Camera Sequence

The cascades classifier is now applied to a moving camera sequence (from [Fossati et al., 2007]). In Fig. 7.18 we show an example for a particular frame: the cropped 96 × 160 image corresponding to the highest value in the saliency map after a dense scan is represented together with the 2D pose of the "peak" in the resulting pose distribution. This distribution is also illustrated on the 3D and 2D representations of the torus manifold.

The results over the whole sequence are depicted in Fig. 7.19. For each presented frame, we present the normalized 96 × 160 image corresponding to the maximum in the saliency map. The 2D pose corresponding to the "winning class" i.e. the highest peak in the distribution

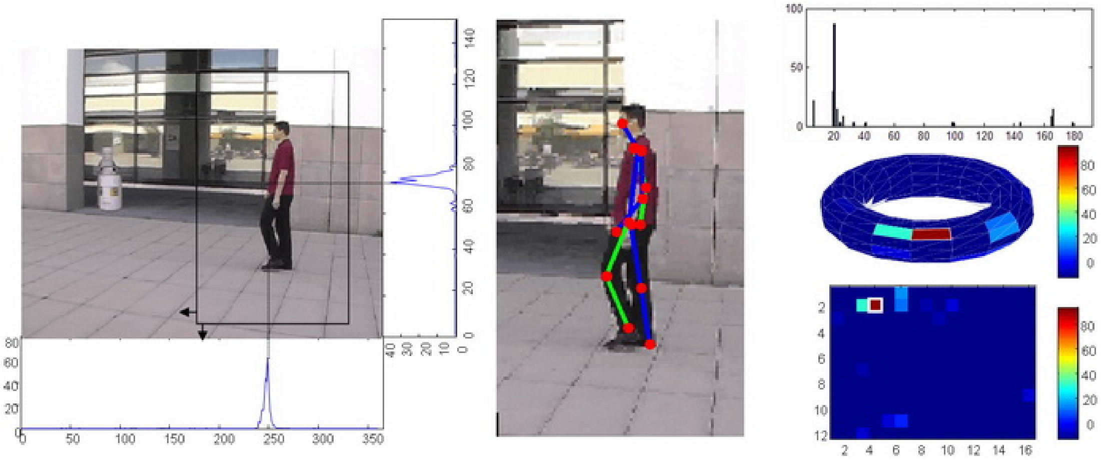


Figure 7.18: Scanning a frame from a moving camera: (left) input image from the moving camera sequence from [Fossati et al., 2007]. Scanning in X and Y directions of the image. (center) Resulting cropped image and pose corresponding to the "pick" resulting from the classification using Random Forest. (right) Resulting distribution over the 192 classes after classification using Random Forest. We also represent this distribution on the 3D and 2D representations of the torus manifold.

is represented on top of the cropped image while the associated 3D pose is show below. The cascade classifier shows over-all satisfactory performances.

7.5 Conclusions

In this chapter, our pose classifier has proven to be an efficient method to jointly localize and recognize the pose of humans in isolated static images with no prior information on the structure of the scene or the number of subjects. It has also shown to be very effective with moving camera sequences.

We have validated our approach with an exhaustive numerical evaluation for 2 different levels of analysis: human detection and pose tracking (with body joints localization). We have compared the localization performances of our classifier with Random Forest (RF) and multi-SVM (Support Vector Machine) classifiers. Our cascade classifier performs slightly better than the multi-SVMs classifier but is 10 times faster. It is also slightly faster than the RF classifier but considerably outperforms it in classification.

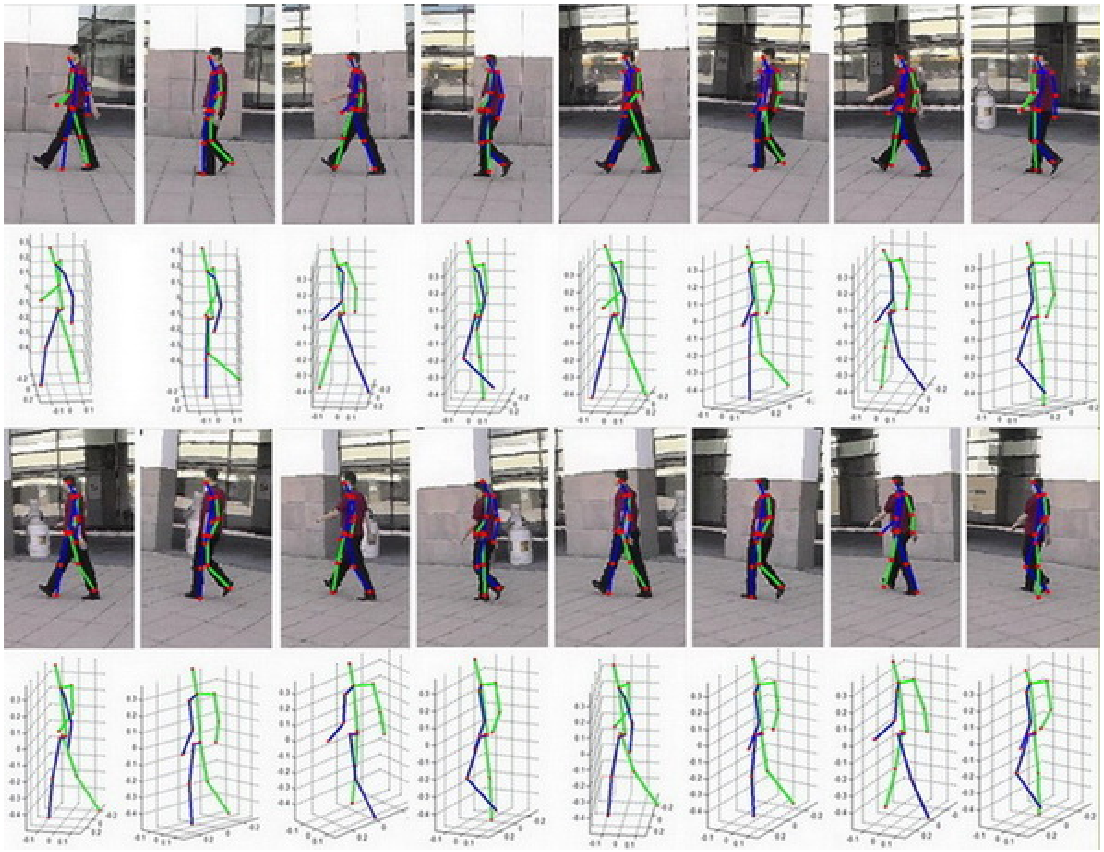


Figure 7.19: Pose detection result with a moving camera. Normalized 96 × 160 images corresponding to the peak obtained applying the pose detection to a moving camera sequence (from [Fossati et al., 2007]). For each presented frame, the mean 2D pose corresponding to the "winning" class is represented on top of the cropped image while the corresponding 3D pose is presented just below.

Part IV

Discussions and Conclusions

8

Conclusions

8.1 Introduction

In the previous chapters, we have proposed and presented several techniques and frameworks which have proved to be effective for pose analysis in different scenarios. For each one of them, we have followed a common methodological approach consisting of a review of the previous work, justification and description of the proposed methodology and an experimental evaluation. In the next sections, we will conclude presenting the contributions and the work done in this thesis, and we will discuss future lines of research.

In this thesis, we have proposed to consider a limited training set captured from a small number of fixed cameras parallel to the ground and distributed around the subject. Then, three types of testing environments with increasing level of difficulty have been identified and studied: 1) a static camera with a similar viewing angle observing only one individual, 2) a fixed surveillance camera with a considerably different viewing angle and multiple targets and 3) a moving camera sequence or just a single static image of an unknown scene. Each testing environment raising different problems, we have considered them separately and have structured the thesis in three main parts corresponding to these three testing conditions.

In the next section, we will present the conclusions of the thesis and report on how we have fulfilled the goals we set out in the introduction. Then, in Section 8.3 we will discuss the main contributions of each part of the thesis and some of the possible future lines of research will be presented.

8.2 Conclusions

The main goals of the thesis were to analyse and find solutions to the problems of 1) the modeling of the human pose and appearance, 2) the detection/localization of the individuals present in the scene and 3) the tracking in pose and image spaces.

8.2.1 Modeling

In this thesis, we have followed a method consisting in discretizing the camera viewpoint around the subject and have considered a set of training views with a camera axis parallel to the ground. The MoBo database has been employed, thus considering only 8 training views, three of which were obtained by mirroring the data from symmetrically opposite views. To jointly model camera viewpoint and pose, in the first part we have introduced the torus manifold which has been employed all along the thesis. This toroidal representation has proven to have non-negligible properties for visualization purpose (to visualize spatio-temporal models in PartI, trajectories in Part II or pose distribution in Part III). In Part II, we have shown that projective geometry can be exploited when the camera axis is not parallel to the ground: the usual 2D similarity transformation relating image and model planes can be replaced by a homography-based alignment. Our results have demonstrated that the incorporation of this perspective correction in a pose tracking framework results in a higher tracking rate and allows for a better estimation of body poses under wide viewpoint variations.

We have considered both 2D and 3D representations for the human pose which has been associated to appearance descriptor. First, the 2D pose and 2D shape of the silhouette have been encapsulated in a point distribution model (PDM), allowing the joint segmentation and pose recovery of the subject observed in the scene. Non-linearities have been dealt with by fitting a multi-view Gaussian mixture model (GMM) to our training data set. The resulting spatio-temporal 2D-models have been concatenated in a global framework and sorted on the surface of the torus manifold.

For 3D pose modeling, we have considered Principal Component Analysis (PCA) for dimensionality reduction in chapter 2. Later, in chapter 5, we have employed a more sophisticated supervised learning method to map the training 3D poses and view-based silhouettes to the torus manifold using kernel-based regressors, which have been learnt using a Relevance Vector Machine (RVM). Given a point on the surface of the torus, the resulting generative model can regress the corresponding pose and view-based silhouette.

In the third part of the thesis, the torus manifold has been used to define a set of classes by discretizing the surface of the torus. Each class is comprised of training images and associated 2D and 3D poses. We have proposed to take advantage of the alignment of the training images to construct a class hierarchy. At each branch of the hierarchy, our training algorithm then selects a small subset of informative class-specific features from a much larger feature space, making this approach computationally efficient and scalable.

8.2.2 Detection

In the first two parts of the thesis, we have taken advantage of the static background to perform foreground detection using a background subtraction algorithm. While in the first part (chapters 2 and 3), we only processed videos with one subject, in the second part (chapters 4 and 5), we have considered multiple targets and have used the heads to detect individuals and solve the problem of occlusions and people moving in group.

In the third part of the thesis, we have studied the cases where the computation of a background image and consequently the segmentation of the subjects are not trivial. We have considered the problem of simultaneous human detection and pose estimation. We have followed a sliding window approach to jointly localize and classify human pose using a fast multi-class classifier which uses edge-based features to classify each tested window: HOG (Histograms of Oriented Gradients) descriptors have been chosen to represent the edge information of images. Our classifier combines the best components of state-of-the-art classifiers including hierarchical trees, cascades of rejectors and randomized forests. Cascade classifiers are designed to quickly reject a large majority of negative candidates to focus on more promising regions, we have exploited this property by learning an ensemble of multi-class hierarchical cascades. Additionally, by randomly sampling from these candidate features, each cascade uses different sets of features to vote which adds some robustness to noise and helps to prevent over-fitting. Both randomization and selection of the number of cascades can be performed on-line at no extra-cost, therefore classifying each window with a different ensemble of cascades. This adaptive classification scheme allows a considerable speed-up and an even more efficient pose detection than simply using the same fixed size ensemble over the image. Each random cascade can vote for one or more class so the ensemble outputs a distribution over poses that can be useful for resolving pose ambiguities.

8.2.3 Tracking

The two problems of tracking in the image and tracking in the pose space have been dealt with in the thesis.

The discretized version of the torus manifold has been employed to limit the space of plausible models in chapter 3 or plausible classes in chapter 7 using simple spatio-temporal constraints. The continuous version of the torus has been used in chapter 5 to sample pose-viewpoint candidates on the surface of the manifold in a particle filter framework. This last method has proved to be more robust to solve pose ambiguities as it can maintain multiple hypothesis throught time.

The tracking in the image has been performed using a simple Kalman filter on image location, scale and angle in chapter 3 and chapter 7 with easy cases where only one subject was tracked. The problem has been facilitated in the second part of this thesis by introducing the calibration of the camera w.r.t. the scene and the tracking has been applied on the ground plane, thus exploring a 2D space instead of the original 4-dimensional space (location, scale and angle). In the chapter 5, we have proposed an efficient particle filtering framework for 3D poses tracking in calibrated surveillance scenes: the tracking was jointly performed on the ground plane and on the surface of the pose-viewpoint torus manifold. Thus, only 4 dimensions need to be explored to track walking human poses in 3D world. In Tab. 8.1 we compare our proposed algorithm w.r.t. the most similar state-of-the-art approaches [Elgammal and Lee, 2009, Jaeggli et al., 2009].

Table 8.1: Comparison of the settings and performances of our algorithm w.r.t. state-of-the-art methods for monocular 3D pose tracking. These approaches share other similarities with our work in that they use silhouette features and low dimensional pose manifolds with particle filtering. The main differences are listed below. Note that we give a range of values for the required number of particles per individual tracker as this number varies depending on single or multiple targets.

Settings/Performances	Jaeggli et al. [2009]	Elgammal and Lee [2009]	**Our work**
Scene Calibration	No	No	**Yes**
State Dimension	10	2	**4**
Training Views	36	12	**8**
Pose Evaluation	Qualitative	Numerical	**Numerical**
Localization	Yes	No	**Yes**
Multiple Targets	No	No	**Yes**
Perspective Videos	No	No	**Yes**
No. Particles	500	900	**250-1000**

8.3 Main Contributions and Future Lines of Work

We will now summarize the main contributions of each part of the thesis and list some of the possible future lines of research.

8.3.1 Part I

We have presented a probabilistic spatio-temporal 2D-models framework for human motion analysis. To cope with the restriction to the viewpoint, local spatio-temporal 2D-models corresponding to several views of the same sequences were trained, concatenated and sorted in a global framework. When processing a sequence, temporal and spatial constraints have been considered to build the Probabilistic Transition Matrix (PTM) that gives the frame to frame prediction of the most probable models from the framework. The experiments carried out on both indoor and outdoor sequences have demonstrated the ability of this approach to adequately segment the pedestrians and estimate their postures independently of the direction of motion during the sequence. They have also demonstrated that the method responds quite robustly to any change of direction during the sequence.

Even though it has been tested with the specific gait motion, the presented approach is generic and could be applied to any other action. A large human motion capture data-base and a 3D computer graphics human model could be used for synthesizing automatically training pairs of 2D and 3D representations. In this thesis, a way has been provided to transition between view-based manifolds of a same action. Transitions between different activities sub-manifolds embedded in a global one could be considered.

8.3.2 Part II

In the second part of the thesis, we have combined the best components of state-of-the-art human pose trackers and have exploited projective geometry in an efficient particle filtering framework for 3D poses tracking in calibrated surveillance scenes. By means of projective geometry, we have replaced the usual 2D similarity transformation relating image and model planes by a homography-based alignment. We have proposed an efficient likelihood computation whose only clues are edges and background subtraction resulting in a fast top-down shape matching. We have also introduced a new state estimator. The efficiency of our algorithm has been demonstrated by processing a set of challenging surveillance videos and present a numerical evaluation for 2784 poses which have been manually labelled and will be made available to the scientific community for further research. Experiments shows that our system successfully tracks walking pedestrians and estimate their 3D poses in cases where a small number of people move together, have occlusion, and cast shadow.

The following lines could be considered in future work:

1. Once we have calibrated the camera in the scene, the camera cannot be moved, which is a limitation of the proposal. An automatic method to detect the vanishing points and compute the homographies could be considered to make the system completely automatic.

2. To deal with the tracking of multiple interacting subjects, we have downweighted the samples using a simple 3D occupancy approach which has shown to be effective with the videos processed in this thesis. The problem of multiple target tracking in more complex situations was out-of-scope for this thesis and the use of a multiple-object filter or a more adequate modeling of the interactions will be explored in future work.

3. Even if all the experiments are specific to the walking activity (due to the higher availability of training and evaluation datasets), our framework is general enough to extend to other activities. The low dimensionality of the searched space combined with the limited number of required training views makes our work easily extendable to more activities and makes more feasible the development of future action recognition software in real surveillance applications. A low dimensional model could be learnt for different actions and a mapping could be used to model activity switching.

4. In future work, multi-stage particle filters, gradient ascent techniques or a more complex study of the posterior could be employed for searching for the optimal solution of the estimate at each time step. Adapting our projective view-invariant method for uncalibrated scenes and unconstrained environments offers another intriguing line for future research.

8.3.3 Part III

In the third part of this thesis, we have considered the problem of simultaneous human detection and pose estimation. We have followed a sliding window approach to jointly localize and classify human pose using a fast multi-class classifier that combines the best components of state-of-the-art classifiers including hierarchical trees, cascades of rejectors and randomized forests.

We have validated our approach with a numerical evaluation for three different levels of analysis: human detection/localization, pose classification and pose estimation (with body joints localization). If the search space (locations in the image and scales) can be reduced, e.g. using a tracking algorithm or limiting the distance to the camera in a Human-Computer Interface, then our method can reach real-time performances.

Several lines of work could be followed to improve the algorithm:

1. The feature selection scheme requires a large amount of aligned training images. Future work should focus on a learning algorithm that could handle smaller training sets or weakly labelled images.

2. A computationally efficient implementation of the HOG descriptor, for example, using a GPU implementation could speed up the detection even further.

3. Our method finds a distribution over exemplars, an intriguing direction for future work would be to combine this with a kernel regression to see if this could produce a method that is both computationally efficient and more accurate in terms of estimated pose.

4. Finally, this work opens several other interesting lines for future work: for instance, we could try to efficiently combine different types of features (color, depth, etc) inside our cascade classifier and extend the algorithm to wider range of motions and actions, or apply the algorithm to general machine learning problems.

Part V

Appendix

A

Projection to a Vertical Plane

Following the classical notation of 3D projective geometry [Hartley and Zisserman, 2004], a 3D point $[X, Y, Z, 1]^T$ is related to its 2D image projection $[u, v, 1]^T$ via a 3×4 projection matrix M:

$$[u, v, 1]^T = \mathbf{M} \cdot [X, Y, Z, 1]^T, \tag{A.1}$$

where points in the projective space $\mathbb{P}^2$ are expressed in homogeneous coordinates and " $=''$ means equality up to scale. The projective transformation matrix $\mathbf{M}$ can be determined with a series of intrinsic and extrinsic parameters or, as shown in [Criminisi et al., 2000], it can be defined as a function of the vanishing points of the dominant 3D directions.

Suppose we want to relate the image I with a vertical plane Π ($\Pi \perp \Pi_{\mathrm{gd}}$), whose intersection with the ground plane Π_{gd} is $\mathbf{G}$. The plane Π is thus spanned by the vertical Z-axis and horizontal G-axis. In that sense, (A.1) becomes:

$$[u, v, 1]^T = \mathbf{H}_{I \leftarrow \Pi} \cdot [G, Z, 1]^T, \tag{A.2}$$

with G a coordinate on the G-axis and $\mathbf{H}_{I \leftarrow \Pi}$ a homography matrix that can be computed from the vanishing points of the dominant 3D directions of Π :

$$\mathbf{H}_{I \leftarrow \Pi} = [\mathbf{v}_G \;\; \alpha \mathbf{v}_Z \;\; \mathbf{o}]. \tag{A.3}$$

where $\mathbf{v}_Z$ is the vertical vanishing point, $\mathbf{o}$ is the origin of the world coordinate system and α is a scale factor. $\mathbf{v}_G$ is the horizontal vanishing point of plane Π in I i.e. the vanishing point along the horizontal direction $\mathbf{G}$ in image I. This vanishing point $\mathbf{v}_G$ can be localized as the intersection of line $\mathbf{g}$, the projection of $\mathbf{G}$ in the image I and $\mathbf{l}$, the horizontal vanishing line in I:

$$\mathbf{v}_G = \mathbf{l} \times \mathbf{g}, \tag{A.4}$$

where × represents the vector product, and $\mathbf{l}$ is the vanishing line of the ground plane (see [Hartley and Zisserman, 2004] for details). Two examples of horizontal vanishing point localizations are given in Fig. A.1.

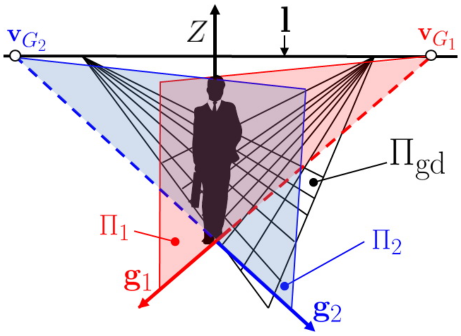


Figure A.1: Horizontal vanishing point localization for homography to vertical plane centered on the human body: 2 examples are given for 2 different directions $\mathbf{g}_1$ and $\mathbf{g}_2$ on the ground plane Π_{gd}. Π_1 is the vertical plane parallel to the real-world direction G_1 and Π_2 the one parallel to G_2. The vanishing points $\mathbf{v}_{G_1}$ and $\mathbf{v}_{G_2}$ are the intersection points of $\mathbf{g}_1$ and $\mathbf{g}_2$ with the horizon line $\mathbf{l}$, i.e. the vanishing line of the ground plane.

Bibliography

[Abdelkader et al., 2011] Abdelkader, M. F., Abd-Almageed, W., Srivastava, A., and Chellappa, R. (2011). Silhouette-based gesture and action recognition via modeling trajectories on riemannian shape manifolds. *Computer Vision and Image Understanding*, 115(3):439–455.

[Agarwal and Triggs, 2004] Agarwal, A. and Triggs, B. (2004). Tracking articulated motion using a mixture of autoregressive models. In *ECCV (3)*, pages 54–65.

[Agarwal and Triggs, 2006] Agarwal, A. and Triggs, B. (2006). Recovering 3d human pose from monocular images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(1):44–58.

[Andriluka et al., 2009] Andriluka, M., Roth, S., and Schiele, B. (2009). Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, pages 1014–1021.

[Andriluka et al., 2010] Andriluka, M., Roth, S., and Schiele, B. (2010). Monocular 3d pose estimation and tracking by detection. In *CVPR*, pages 623–630.

[Athitsos and Sclaroff, 2003] Athitsos, V. and Sclaroff, S. (2003). Estimating 3d hand pose from a cluttered image. In *CVPR*, pages II: 432–439.

[Balan et al., 2007] Balan, A., Sigal, L., Black, M., Davis, J., and Haussecker, H. (2007). Detailed human shape and pose from images. In *CVPR*, pages 1–8.

[Baumberg, 1995] Baumberg, A. (1995). *Learning Deformable Models for Tracking Human Motion.* PhD Tesis, Univ. of Leeds.

[Baumberg and Hogg, 1994] Baumberg, A. and Hogg, D. (1994). Learning flexible models from image sequences. In *ECCV*, pages 299–308.

[Belongie et al., 2002] Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522.

[Bergtholdt et al., 2010] Bergtholdt, M., Kappes, J. H., Schmidt, S., and Schnörr, C. (2010). A study of parts-based object class detection using complete graphs. *International Journal of Computer Vision*, 87(1-2):93–117.

[Bissacco et al., 2007] Bissacco, A., Yang, M., and Soatto, S. (2007). Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. In *CVPR*, pages 1–8.

[Bissacco et al., 2006] Bissacco, A., Yang, M.-H., and Soatto, S. (2006). Detecting humans via their pose. In *NIPS*, pages 169–176.

[Bookstein, 1991] Bookstein, F. (1991). *Morphometric Tools for Landmark Data: Geometry and Biology.* Cambridge Univ. Press.

[Bosch et al., 2007] Bosch, A., Zisserman, A., and Munoz, X. (2007). Image classification using random forests and ferns. In *Proc. of the 11th Intern. Conf. on Computer Vision, Rio de Janeiro, Brazil.*

[Bouchrika et al., 2009] Bouchrika, I., Goffredo, M., Carter, J. N., and Nixon, M. S. (2009). Covariate analysis for view-point independent gait recognition. In *ICB*, pages 990–999.

[Bourdev and Malik, 2009] Bourdev, L. and Malik, J. (2009). Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*.

[Bourdev et al., 2010] Bourdev, L. D., Maji, S., Brox, T., and Malik, J. (2010). Detecting people using mutually consistent poselet activations. In *ECCV (6)*, pages 168–181.

[Bowden, 1999] Bowden, R. (1999). *Learning non-linear Models of Shape and Motion.* PhD thesis, Dept Systems Engineering, Brunel University.

[Bowden et al., 2000] Bowden, R., Mitchell, T. A., and Sarhadi, M. (2000). Non-linear statistical models for the 3d reconstruction of human pose and motion from monocular image sequences. *Image Vision Comput.*, 18(9):729–737.

[Brand, 1999] Brand, M. (1999). Shadow puppetry. In *International Conference on Computer Vision*, volume 2, page 1237.

[Bregler et al., 2004] Bregler, C., Malik, J., and Pullen, K. (2004). Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*, 56(3):179–194.

[Breiman, 1996] Breiman, L. (1996). Out-of-bag estimation.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Mach. Learn.*, 45(1):5–32.

[Breitenstein et al., 2011] Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E., and Gool, L. J. V. (2011). Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(9):1820–1833.

[Brostow et al., 2008] Brostow, G. J., Shotton, J., Fauqueur, J., and Cipolla, R. (2008). Segmentation and recognition using structure from motion point clouds. In *ECCV*, pages 44–57.

[Canton-Ferrer et al., 2011] Canton-Ferrer, C., Casas, J. R., and Pardás, M. (2011). Human motion capture using scalable body models. *Computer Vision and Image Understanding*, 115(10):1363–1374.

[Caviar, 2004] Caviar (2004). Ec funded caviar project ist 2001 37540.

[Chang and Lin, 2010] Chang, I.-C. and Lin, S.-Y. (2010). 3d human motion tracking based on a progressive particle filter. *Pattern Recognition*, 43(10):3621–3635.

[Collins and Liu, 2003] Collins, R. and Liu, Y. (2003). On-line selection of discriminative tracking features. In *ICCV*. Published by the IEEE Computer Society.

[Cootes and Taylor, 1997] Cootes, T. and Taylor, C. (1997). A mixture model for representing shape variation. In *BMVC*.

[Corazza et al., 2006] Corazza, S., Muendermann, L., Chaudhari, A., Demattio, T., Cobelli, C., and Andriacchi, T. (2006). A markerless motion capture system to study musculoskeletal biomechanics: Visual hull and simulated annealing approach. *Annals of Biomedical Engineering*, 34(6):1019–29.

[Cremers, 2006] Cremers, D. (2006). Dynamical statistical shape priors for level set-based tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(8):1262–1273.

[Criminisi et al., 2000] Criminisi, A., Reid, I. D., and Zisserman, A. (2000). Single view metrology. *International Journal of Computer Vision*, 40(2):123–148.

[Cucchiara et al., 2005] Cucchiara, R., Grana, C., Prati, A., and Vezzani, R. (2005). Probabilistic posture classification for human-behavior analysis. *IEEE Trans. Systems, Man, and Cybernetics - part A*, 35(1):42–54.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR*, volume 2, pages 886–893.

[Datar et al., 2004] Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. (2004). Locality-sensitive hashing scheme based on p-stable distributions. In *Proc. of the 20th annual symposium on Computational geometry*, pages 253–262.

[Datta et al., 2011] Datta, A., Sheikh, Y., and Kanade, T. (2011). Linearized motion estimation for articulated planes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):780–793.

[Davies et al., 2003] Davies, R. H., Twining, C. J., Allen, P. D., Cootes, T. F., and Taylor, C. J. (2003). Building optimal 2d statistical shape models. *Image Vision Comput.*, 21(13-14):1171–1182.

[Deselaers et al., 2007] Deselaers, T., Criminisi, A., Winn, J. M., and Agarwal, A. (2007). Incorporating on-demand stereo for real time recognition. In *CVPR*.

[Deutscher et al., 2000] Deutscher, J., Blake, A., and Reid, I. (2000). Articulated body motion capture by annealed particle filtering. In *CVPR*, volume 2, pages 126 – 133.

[Deutscher and Reid, 2005] Deutscher, J. and Reid, I. (2005). Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61(2):185–205.

[Dimitrijevic et al., 2006] Dimitrijevic, M., Lepetit, V., and Fua, P. (2006). Human body pose detection using bayesian spatio-temporal templates. *Comput. Vis. Image Underst.*, 104(2):127–139.

[Ek et al., 2008] Ek, C. H., Rihan, J., Torr, P. H. S., Rogez, G., and Lawrence, N. D. (2008). Ambiguity modeling in latent spaces. In *MLMI*, pages 62–73.

[Elgammal et al., 2002] Elgammal, A., Duraiswami, R., Harwood, D., and Davis, L. S. (2002). Background and foreground modeling using nonparametric kernel density for visual surveillance. In *Proceedings of the IEEE*, volume 7, pages 1151–1163.

[Elgammal and Lee, 2004] Elgammal, A. M. and Lee, C. (2004). Inferring 3d body pose from silhouettes using activity manifold learning. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2:681–688.

[Elgammal and Lee, 2009] Elgammal, A. M. and Lee, C.-S. (2009). Tracking people on a torus. *IEEE Trans. on PAMI*, 31(3):520–538.

[Enzweiler and Gavrila, 2008] Enzweiler, M. and Gavrila, D. M. (2008). A mixed generative-discriminative framework for pedestrian classification. In *CVPR*.

[Enzweiler and Gavrila, 2009] Enzweiler, M. and Gavrila, D. M. (2009). Monocular pedestrian detection: Survey and experiments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(12):2179–2195.

[Fan et al., 2003] Fan, L., Sung, K., and Ng, T. (2003). Pedestrian registration in static images with unconstrained background. *Pattern Recognition*, 36:1019–1029.

[Farhadi and Tabrizi, 2008] Farhadi, A. and Tabrizi, M. K. (2008). Learning to recognize activities from the wrong view point. In *ECCV (1)*, pages 154–166.

[Felzenszwalb et al., 2010a] Felzenszwalb, P. F., Girshick, R. B., and McAllester, D. A. (2010a). Cascade object detection with deformable part models. In *CVPR*, pages 2241–2248.

[Felzenszwalb et al., 2010b] Felzenszwalb, P. F., Girshick, R. B., McAllester, D. A., and Ramanan, D. (2010b). Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645.

[Felzenszwalb and Huttenlocher, 2005] Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79.

[Ferrari et al., 2008] Ferrari, V., Marin-Jimenez, M. J., and Zisserman, A. (2008). Progressive search space reduction for human pose estimation. In *CVPR*.

[Fossati et al., 2007] Fossati, A., Dimitrijevic, M., Lepetit, V., and Fua, P. (2007). Bridging the gap between detection and tracking for 3d monocular video-based motion capture. In *CVPR*.

[Gall et al., 2010] Gall, J., Rosenhahn, B., Brox, T., and Seidel, H.-P. (2010). Optimization and filtering for human motion capture. *Int. Journal of Computer Vision*, 87(1-2):75–92.

[Gall et al., 2011] Gall, J., Yao, A., Razavi, N., Gool, L. J. V., and Lempitsky, V. S. (2011). Hough forests for object detection, tracking, and action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(11):2188–2202.

[Gavrila and Davis, 1996] Gavrila, D. and Davis, L. (1996). 3d model-based tracking of humans in action: A multi-view approach. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 73–80.

[Gavrila, 1999] Gavrila, D. M. (1999). The visual analysis of human movement: A survey. *Computer Vision and Image Understanding: CVIU*, 73(1):82–98.

[Gavrila, 2007] Gavrila, D. M. (2007). A bayesian, exemplar-based approach to hierarchical shape matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(8):1408–1421.

[Giebel et al., 2004] Giebel, J., Gavrila, D., and Schnörr, C. (2004). A bayesian framework for multi-cue 3d object tracking. In *ECCV (4)*, pages 241–252.

[Goffredo et al., 2008] Goffredo, M., Seely, R. D., Carter, J. N., and Nixon, M. S. (2008). Markerless view independent gait analysis with self-camera calibration. In *FG*, pages 1–6.

[Gong and Medioni, 2011] Gong, D. and Medioni, G. G. (2011). Dynamic manifold warping for view invariant action recognition. In *ICCV*.

[Grauman et al., 2003] Grauman, K., Shakhnarovich, G., and Darrell, T. (2003). Inferring 3d structure with a statistical image-based shape model. In *ICCV*, pages 641–648.

[Grauman et al., 2004] Grauman, K., Shakhnarovich, G., and Darrell, T. (2004). Example-based 3d shape inference from a single silhouettes. In *Proc. ECCV Workshop SMVP*.

[Gross and Shi, 2001] Gross, R. and Shi, J. (2001). The cmu motion of body (mobo) database.

[Guerrero and Sagüés, 2003] Guerrero, J. and Sagüés, C. (2003). Robust line matching and estimate of homographies simultaneously. In *Proc. Ib. Conf. on Pattern Recognition and Image Analysis (IbPria)*, pages 297–307.

[Haritaoglu et al., 2000] Haritaoglu, I., Harwood, D., and Davis, L. (2000). W4: Real-time surveillance of people and their activities. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):809–830.

[Hartley and Zisserman, 2004] Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.

[Heap and Hogg, 1998] Heap, T. and Hogg, D. (1998). Wormholes in shape space: Tracking through discontinuous changes in shape. In *ICCV*, pages 344–349.

[Hofmann and Gavrila, 2012] Hofmann, M. and Gavrila, D. (2012). Multi-view 3d human pose estimation in complex environment. *International Journal of Computer Vision*, 96(1):103–124.

[Hyvaerinen et al., 2001] Hyvaerinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. Wiley Interscience.

[Inman et al., 1981] Inman, V. T., Ralston, H. J., and Todd, F. (1981). *Human Walking*. Williams and Wilkins, Baltimore, USA.

[Isard and Blake, 1998] Isard, M. and Blake, A. (1998). Condensation – conditional density propagation for visual tracking. *IJCV*, 29(1):5–28.

[Isard and MacCormick, 2001] Isard, M. and MacCormick, J. (2001). Bramble: A bayesian multiple-blob tracker. In *Proc. IEEE Int. Conf. Computer Vision*, pages 34–41.

[Jaeggli et al., 2009] Jaeggli, T., Koller-Meier, E., and Gool, L. J. V. (2009). Learning generative models for multi-activity body pose estimation. *International Journal of Computer Vision*, 83(2):121–134.

[Ji and Liu, 2010] Ji, X. and Liu, H. (2010). Advances in view-invariant human motion analysis: A review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 40(1):13–24.

[Kakadiaris and Metaxas, 2000] Kakadiaris, I. A. and Metaxas, D. N. (2000). Model-based estimation of 3d human motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1453–1459.

[Kale et al., 2003] Kale, A., Chowdhury, A. K. R., and Chellappa, R. (2003). Towards a view invariant gait recognition algorithm. In *IEEE Int. Conf on Advanced Video and Signal based Surveillance*, pages 143–150.

[Kalman, 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(D):35–45.

[Kanade et al., 2000] Kanade, T., li Tian, Y., and Cohn, J. F. (2000). Comprehensive database for facial expression analysis. In *FG*, pages 46–53.

[Koschan et al., 2003] Koschan, A., Kang, S., Paik, J., Abidi, B., and Abidi, M. (2003). Color active shape models for tracking non-rigid objects. *Pattern Recogn. Lett.*, 24(11):1751–1765.

[Lan and Huttenlocher, 2004] Lan, X. and Huttenlocher, D. P. (2004). A unified spatio-temporal articulated model for tracking. In *CVPR (1)*, pages 722–729.

[Laptev, 2009] Laptev, I. (2009). Improving object detection with boosted histograms. *Image Vision Comput.*, 27(5):535–544.

[Lee and Elgammal, 2006] Lee, C.-S. and Elgammal, A. M. (2006). Simultaneous inference of view and body pose using torus manifolds. In *ICPR (3)*, pages 489–494.

[Lee and Elgammal, 2010] Lee, C.-S. and Elgammal, A. M. (2010). Coupled visual and kinematic manifold models for tracking. *International Journal of Computer Vision*, 87(1-2):118–139.

[Lepetit and Fua, 2006] Lepetit, V. and Fua, P. (2006). Keypoint recognition using randomized trees. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(9):1465–1479.

[Li et al., 2010] Li, R., Tian, T.-P., Sclaroff, S., and Yang, M.-H. (2010). 3d human motion tracking with a coordinated mixture of factor analyzers. *International Journal of Computer Vision*, 87(1-2):170–190.

[Li et al., 2008] Li, Y., Wu, B., and Nevatia, R. (2008). Human detection by searching in 3d space using camera and scene knowledge. In *ICPR*, pages 1–5.

[Lin and Davis, 2010] Lin, Z. and Davis, L. S. (2010). Shape-based human detection and segmentation via hierarchical part-template matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(4):604–618.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[Lutton et al., 1994] Lutton, E., Maitre, H., and Lopez-K., J. (1994). Contribution to the determination of vanishing points using hough transform. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(4):430–438.

[Lv and Nevatia, 2007] Lv, F. and Nevatia, R. (2007). Single view human action recognition using key pose matching and viterbi path searching. In *CVPR*.

[Ma and Ding, 2005] Ma, Y. and Ding, X. (2005). Real-time multi-view face detection and pose estimation based on cost-sensitive adaboost. *Tsinghua Science & Technology*, 10(2):152–157.

[MacCormick and Blake, 2000] MacCormick, J. and Blake, A. (2000). A probabilistic exclusion principle for tracking multiple objects. *International Journal of Computer Vision*, 39(1):57–71.

[MacCormick and Isard, 2000] MacCormick, J. and Isard, M. (2000). Partitioned sampling, articulated objects and interface-quality hand tracking. In *ECCV*, Dublin.

[Martínez et al., 2007] Martínez, J., Orrite-Uruñuela, C., and Rogez, G. (2007). Rao-blackwellized particle filter for human appearance and position tracking. In *Proc. of the Third Iberian Conference on Pattern Recognition and Image Analysis (IbPria)*, pages 201–208.

[Moeslund and Granum, 2001] Moeslund, T. B. and Granum, E. (2001). A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding: CVIU*, 81(3):231–268.

[Moeslund et al., 2006] Moeslund, T. B., Hilton, A., and Kruger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104:90–126.

[Moeslund et al., 2011] Moeslund, T. B., Hilton, A., Krüger, V., and Sigal, L., editors (2011). *Visual Analysis of Humans - Looking at People*. Springer.

[Moosmann et al., 2008] Moosmann, F., Nowak, E., and Jurie, F. (2008). Randomized clustering forests for image classification. *IEEE Trans. on PAMI*, 30(9):1632–1646.

[Mori and Malik, 2006] Mori, G. and Malik, J. (2006). Recovering 3d human body configurations using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(7):1052–1062.

[Munder et al., 2008] Munder, S., Schnörr, C., and Gavrila, D. M. (2008). Pedestrian detection and tracking using a mixture of view-based shape-texture models. *IEEE Transactions on Intelligent Transportation Systems*, 9(2):333–343.

[Navaratnam et al., 2005] Navaratnam, R., Thayananthan, A., Torr, P., and Cipolla, R. (2005). Hierarchical part-based human body pose estimation. In *BMVC*.

[Ning et al., 2004a] Ning, H., Tan, T., Wang, L., and Hu, W. (2004a). Kinematics-based tracking of human walking in monocular video sequences. *IVC*, 22:429–441.

[Ning et al., 2004b] Ning, H., Tan, T., Wang, L., and W, H. (2004b). People tracking based on motion model and motion constraints with automatic initialization. *Pattern Recognition*, 37(7):1423–1440.

[Okada and Soatto, 2008] Okada, R. and Soatto, S. (2008). Relevant feature selection for human pose estimation and localization in cluttered images. In *ECCV*, pages 434–445.

[Okada and Stenger, 2008] Okada, R. and Stenger, B. (2008). A single camera motion capture system for human-computer interaction. *IEICE TRANSACTIONS on Information and Systems*, 91(7):1855–1862.

[Ong et al., 2006] Ong, E., Micilotta, A. S., Bowden, R., and Hilton, A. (2006). Viewpoint invariant exemplar-based 3d human tracking. *Computer Vision and Image Understanding*, 104:178–189.

[Orrite et al., 2009] Orrite, C., Gañán, A., and Rogez, G. (2009). Hog-based decision tree for facial expression classification. In *IbPRIA*, pages 176–183.

[Orrite-Uruñuela et al., 2004] Orrite-Uruñuela, C., del Rincón, J. M., Jaraba, J. E. H., and Rogez, G. (2004). 2d silhouette and 3d skeletal models for human detection and tracking. In *ICPR*, pages 244–247.

[Parameswaran and Chellappa, 2004] Parameswaran, V. and Chellappa, R. (2004). View independent human body pose estimation from a single perspective image. In *CVPR (2)*, pages 16–22.

[Parameswaran and Chellappa, 2006] Parameswaran, V. and Chellappa, R. (2006). View invariance for human action recognition. *International Journal of Computer Vision*, 66(1):83–101.

[Plankers and Fua, 2001] Plankers, R. and Fua, P. (2001). Articulated soft objects for video-based body modeling. In *Proceedings of the Ninth International Conference on Computer Vision*, Vancouver, Canada.

[Poppe, 2007] Poppe, R. (2007). Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108:4–18.

[Ramanan, 2006] Ramanan, D. (2006). Learning to parse images of articulated bodies. In *NIPS*, pages 1129–1136.

[Ramanan and Forsyth, 2003] Ramanan, D. and Forsyth, D. A. (2003). Finding and tracking people from the bottom up. In *CVPR (2)*, pages 467–474.

[Ramanan et al., 2007] Ramanan, D., Forsyth, D. A., and Zisserman, A. (2007). Tracking people by learning their appearance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):65–81.

[Riklin-Raviv et al., 2007] Riklin-Raviv, T., Kiryati, N., and Sochen, N. A. (2007). Prior-based segmentation and shape registration in the presence of perspective distortion. *International Journal of Computer Vision*, 72(3):309–328.

[Roberts et al., 2004] Roberts, T., McKenna, S., and Ricketts, I. (2004). Human pose estimation using learnt probabilistic region similarities and partial configurations. In *ECCV*, pages 291–303. Springer.

[Rogez et al., 2006a] Rogez, G., Guerrero, J., Martínez, J., and Orrite, C. (2006a). Viewpoint independent human motion analysis in man-made environments. In *Proc. of the 17th British Machine Vision Conference (BMVC)*, volume 2, pages 659–668, Edinburgh, UK.

[Rogez et al., 2007a] Rogez, G., Guerrero, J. J., and Orrite, C. (2007a). View-invariant human feature extraction for video-surveillance applications. In *Proc. of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 324–329.

[Rogez et al., 2007b] Rogez, G., Martínez, J., and Orrite, C. (2007b). Dealing with non-linearity in shape modelling of articulated objects. In *Proc. of the Third Iberian Conference on Pattern Recognition and Image Analysis (IbPria)*, pages 63–71.

[Rogez et al., 2008a] Rogez, G., Orrite, C., and Martínez, J. (2008a). A spatio-temporal 2d-models framework for human pose recovery in monocular sequences. *Pattern Recognition*, 41(9):2926–2944.

[Rogez et al., 2006b] Rogez, G., Orrite, C., Martínez, J., and Jaraba, J. E. H. (2006b). Probabilistic spatio-temporal 2d-model for pedestrian motion analysis in monocular sequences. In *Proc. of the 4th International Conference on Articulated Motion and Deformable Objects (AMDO)*, pages 175–184.

[Rogez et al., view] Rogez, G., Orrite, C., Rihan, J., Guerrero, J. J., and Torr, P. H. (under review). View-invariant shape-based 3d human pose tracking in monocular surveillance videos. *submitted to the International Journal of Computer Vision.*

[Rogez et al., 2005] Rogez, G., Orrite-Uruñuela, C., and Martínez, J. (2005). Human figure segmentation using independent component analysis. In *IbPRIA (1)*, pages 300–307.

[Rogez et al., 2012] Rogez, G., Rihan, J., Orrite, C., and Torr, P. H. (2012). Fast human pose detection using randomized hierarchical cascades of rejectors. *International Journal of Computer Vision*, 99(1):25–52.

[Rogez et al., tion] Rogez, G., Rihan, J., Orrite, C., and Torr, P. H. (invited chapter in preparation). Randomized trees for human pose classification. In Criminisi, A., Shotton, J., and Konukoglu, E., editors, *Decision Forests - for Computer Vision and Medical Image Analysis*, Advances in Computer Vision and Pattern Recognition. Springer.

[Rogez et al., 2008b] Rogez, G., Rihan, J., Ramalingam, S., Orrite, C., and Torr, P. H. (2008b). Randomized trees for human pose detection. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR).*

[Rogez et al., 2007c] Rogez, G., Rius, I., Martínez, J., and Orrite, C. (2007c). Exploiting spatio-temporal constraints for robust 2d pose tracking. In *Proc. of the Second Workshop of Human Motion - Understanding, Modeling, Capture and Animation*, pages 58–73.

[Rosales and Sclaroff, 2006] Rosales, R. and Sclaroff, S. (2006). Combining generative and discriminative models in a framework for articulated pose estimation. *International Journal of Computer Vision*, 67(3):251–276.

[Rosales et al., 2001] Rosales, R., Siddiqui, M., Alon, J., and Sclaroff, S. (2001). Estimating 3d body pose using uncalibrated cameras. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1:821–827.

[Rosenhahn et al., 2006] Rosenhahn, B., Kersting, U. G., Powell, K., and Seidel, H.-P. (2006). Cloth x-ray: Mocap of people wearing textiles. In *DAGM-Symposium*, pages 495–504.

[Sabzmeydani and Mori, 2007] Sabzmeydani, P. and Mori, G. (2007). Detecting pedestrians by learning shapelet features. In *CVPR07*.

[Shakhnarovich et al., 2003] Shakhnarovich, G., Viola, P., and Darrell, R. (2003). Fast pose estimation with parameter-sensitive hashing. In *ICCV*.

[Shotton et al., 2011] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *CVPR*.

[Shotton et al., 2008] Shotton, J., Johnson, M., Cipolla, R., Center, T., and Kawasaki, J. (2008). Semantic texton forests for image categorization and segmentation. In *CVPR*.

[Sidenbladh et al., 2000] Sidenbladh, H., Black, M., and Fleet, D. (2000). Stochastic tracking of 3d human figures using 2d image motion. In *ECCV*, volume 2, pages 702–718, Dublin.

[Sidenbladh et al., 2002] Sidenbladh, H., Black, M. J., and Sigal, L. (2002). Implicit probabilistic models of human motion for synthesis and tracking. In *ECCV (1)*, pages 784–800.

[Siebel and Maybank, 2002a] Siebel, N. T. and Maybank, S. J. (2002a). Fusion of multiple tracking algorithms for robust people tracking. In *Europ. Conf. Computer Vision*, pages 373–387.

[Siebel and Maybank, 2002b] Siebel, N. T. and Maybank, S. J. (2002b). Fusion of multiple tracking algorithms for robust people tracking. In *ECCV (4)*, pages 373–387.

[Sigal et al., 2010] Sigal, L., Balan, A. O., and Black, M. J. (2010). Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1-2):4–27.

[Sigal et al., 2004] Sigal, L., Bhatia, S., Roth, S., Black, M. J., and Isard, M. (2004). Tracking loose-limbed people. In *CVPR*, pages 421–428.

[Sigal and Black, 2010] Sigal, L. and Black, M. J. (2010). Guest editorial: State of the art in image- and video-based human pose and motion estimation. *International Journal of Computer Vision*, 87(1-2):1–3.

[Sminchisescu et al., 2005] Sminchisescu, C., Kanaujia, A., Li, Z., and Metaxas, D. N. (2005). Discriminative density propagation for 3d human motion estimation. In *CVPR*, pages 390–397.

[Sminchisescu et al., 2006] Sminchisescu, C., Kanaujia, A., and Metaxas, D. N. (2006). Learning joint top-down and bottom-up processes for 3d visual inference. In *CVPR*, pages 1743–1752.

[Sminchisescu et al., 2007] Sminchisescu, C., Kanaujia, A., and Metaxas, D. N. (2007). Bm^{3}e : Discriminative density propagation for visual tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(11):2030–2044.

[Sminchisescu and Triggs, 2003] Sminchisescu, C. and Triggs, B. (2003). Estimating articulated human motion with covariance scaled sampling. *Int. Journal of Robotic Research*, 22(6):371–392.

[Smith et al., 2005] Smith, K., Gatica-Perez, D., and Odobez, J.-M. (2005). Using particles to track varying numbers of interacting people. In *CVPR (1)*, pages 962–969.

[Sola et al., 2012] Sola, J., Vidal-Calleja, T., Civera, J., and Montiel, J. (2012). Impact of landmark parametrization on monocular ekf-slam with points and lines. *International Journal of Computer Vision*, 97:339–368.

[Stenger, 2004] Stenger, B. (2004). *Model-Based Hand Tracking Using A Hierarchical Bayesian Filter*. PhD thesis, Department of Engineering, University of Cambridge.

[Stenger et al., 2006] Stenger, B., Thayananthan, A., Torr, P. H. S., and Cipolla, R. (2006). Model-based hand tracking using a hierarchical bayesian filter. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(9):1372–1384.

[Sugano and Miyamoto, 2007] Sugano, H. and Miyamoto, R. (2007). A real-time object recognition system on cell broadband engine. In *Proc. of the 2nd Pacific Rim conference on Advances in image and video technology*, pages 932–943.

[Taylor, 2000] Taylor, C. J. (2000). Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Comput. Vis. Image Underst.*, 80(3):349–363.

[Thayananthan et al., 2006] Thayananthan, A., Navaratnam, R., Stenger, B., Torr, P. H. S., and Cipolla, R. (2006). Multivariate relevance vector machines for tracking. In *ECCV (3)*, pages 124–138.

[Toyama and Blake, 2002] Toyama, K. and Blake, A. (2002). Probabilistic tracking with exemplars in a metric space. *Int. J. Comput. Vision*, 48(1):9–19.

[Tsai, 1986] Tsai, R. Y. (1986). An efficient and accurate camera calibration technique for 3D machine vision. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 364–374.

[Urtasun et al., 2006a] Urtasun, R., Fleet, D., and Fua, P. (2006a). Temporal motion models for monocular and multiview 3d human body tracking. *Computer Vision and Image Understanding*, 103:157–177.

[Urtasun et al., 2005] Urtasun, R., Fleet, D., Hertzmann, A., and Fua, P. (2005). Priors for people tracking from small training sets. In *ICCV*, volume 1, pages 403–410.

[Urtasun et al., 2006b] Urtasun, R., Fleet, D. J., and Fua, P. (2006b). 3d people tracking with gaussian process dynamical models. In *CVPR (1)*, pages 238–245.

[Veeraraghavan et al., 2005] Veeraraghavan, A., Roy-Chowdhury, A. K., and Chellappa, R. (2005). Matching shape sequences in video with applications in human movement analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(12):1896–1909.

[Villamizar et al., 2009] Villamizar, M., Sanfeliu, A., and Andrade-Cetto, J. (2009). Local boosted features for pedestrian detection. In *IbPRIA*, pages 128–135.

[Viola et al., 2005] Viola, P., Jones, M. J., and Snow, D. (2005). Detecting pedestrians using patterns of motion and appearance. *Int. J. Comput. Vision*, 63(2):153–161.

[Viola and Jones, 2004] Viola, P. A. and Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154.

[Wang et al., 2003] Wang, L., Hu, W., and Tan, T. (2003). Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601.

[Weinland et al., 2007] Weinland, D., Boyer, E., and Ronfard, R. (2007). Action recognition from arbitrary views using 3d exemplars. In *ICCV*, pages 1–7.

[Wu and Nevatia, 2009] Wu, B. and Nevatia, R. (2009). Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. *International Journal of Computer Vision*, 82(2):185–204.

[Wu et al., 2005] Wu, Y., Lin, J., and Huang, T. (2005). Analyzing and capturing articulated hand motion in image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1910–1922.

[Wu and Yu, 2006] Wu, Y. and Yu, T. (2006). A field model for human detection and tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(5):753–765.

[Yacoob and Black, 1999] Yacoob, Y. and Black, M. (1999). Parameterized modeling and recognition of activities in temporal surfaces. *Computer Vision and Image Understanding*, 73(2):232–247.

[Yacoob and Davis, 2000] Yacoob, Y. and Davis, L. (2000). Learned models for estimation of rigid and articulated human motion from stationnary or moving cameras. *International Journal of Computer Vision*, 36(1):5–30.

[Zehnder et al., 2005] Zehnder, P., Koller-Meier, E., and Van Gool, L. (2005). A hierarchical system for recognition, tracking and pose estimation. In *MLMI*, pages 329–340.

[Zhang, 2006] Zhang, J. (2006). *Statistical Modeling and Localization of Nonrigid and Articulated Shapes*. doctoral dissertation, tech. report CMU-RI-TR-06-18, Robotics Institute, Carnegie Mellon University.

[Zhang et al., 2004] Zhang, J., Collins, R., and Liu, Y. (2004). Representation and matching of articulated shapes. In *CVPR*, pages 342–349.

[Zhang et al., 2005a] Zhang, J., Collins, R. T., and Liu, Y. (2005a). Bayesian body localization using mixture of nonlinear shape models. In *ICCV*, pages 725–732.

[Zhang et al., 2005b] Zhang, J., Collins, R. T., and Liu, Y. (2005b). Bayesian body localization using mixture of nonlinear shape models. In *ICCV*, pages 725–732.

[Zhang et al., 2007a] Zhang, J., Zhou, S., McMillan, L., and Comaniciu, D. (2007a). Joint real-time object detection and pose estimation using probabilistic boosting network. In *CVPR*, pages 1–8.

[Zhang et al., 2007b] Zhang, L., Wu, B., and Nevatia, R. (2007b). Detection and tracking of multiple humans with extensive pose articulation. In *ICCV*.

[Zhang et al., 2002] Zhang, Z., Zhu, L., Li, S., and Zhang, H. (2002). Real-time multi-view face detection. In *Proc. Int. Conf. Automatic Face and Gesture Recognition*, pages 149–154.

[Zhao and Nevatia, 2002] Zhao, T. and Nevatia, R. (2002). Stochastic human segmentation from static camera. In *IEEE Workshop on Motion and Video Computing*.

[Zhao and Nevatia, 2004] Zhao, T. and Nevatia, R. (2004). Tracking multiple humans in complex situations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1208–1221.

[Zhao et al., 2008] Zhao, T., Nevatia, R., and Wu, B. (2008). Segmentation and tracking of multiple humans in crowded environments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(7):1198–1211.

[Zhu et al., 2006] Zhu, Q., Yeh, M., Cheng, K., and Avidan, S. (2006). Fast human detection using a cascade of histograms of oriented gradients. In *CVPR06*, pages II: 1491–1498.