From pixels to gestures: learning visual representations for human analysis in color and depth data sequences

Antonio Hernández-Vela

From pixels to gestures: learning visual representations for human analysis in color and depth data sequences

A dissertation submitted by **Antonio Hernández-Vela** at Universitat de Barcelona to fulfil the degree of **Doctor en Matemàtiques**.

Barcelona, January 20, 2015

Director	**Dr. Sergio Escalera** Dept. de Matemàtica Aplicada i Anàlisi, Universitat de Barcelona & Centre de Visió per Computador
Co-director	**Prof. Stan Sclaroff** Dept. of Computer Science, Boston University Boston, USA
Thesis committee	**Prof. Thomas Baltzer Moeslund** Dept. of Architecture, Design and Media Technology, Aalborg University Aalborg, Denmark **Prof. Jordi Vitrià Marca** Dept. de Matemàtica Aplicada i Anàlisi, Universitat de Barcelona & Computer Vision Center Barcelona, Spain **Dr. Jordi Gonzalez Sabaté** Dept. Ciències de la Computació, Universitat Autònoma de Barcelona & Computer Vision Center Barcelona, Spain
International evaluators	**Dr. Leonid Sigal** Disney Research & Dept. of Computer Science, Carnegie Mellon University Pittsbutgh, USA **Dr. Marco Pedersoli** KU Leuven Leuven, Belgium **Prof. Thomas Baltzer Moeslund** Dept. of Architecture, Design and Media Technology, Aalborg University Aalborg, Denmark

This document was typeset by the author using LaTeX 2$_\varepsilon$.

The research described in this book was carried out at Universitat de Barcelona and the Computer Vision Center.

ISBN: 978-84-940902-0-2

Printed by Ediciones Gráficas Rey, S.L.

A mis padres ...

Acknowledgements

The work presented in this dissertation would not have been possible without the guidance and support of my supervisors. I am extremely thankful to my advisor Sergio Escalera for his continuous efforts, dedication and encouragement during all these years. I am also grateful to Petia Radeva, for her guidance during the early stages of my PhD. Finally, I am especially thankful to my co-advisor Stan Sclaroff for giving me the chance to visit his research group at Boston University and having numerous conversations providing me with valuable feedback and brilliant ideas.

During my short stay in Boston I had the great chance to meet Stan and a lot of nice and brilliant people from the Image and Video Computing research group, from whom I learnt a lot during the five months I spent at Boston University. Among them all, I must give special thanks to Shugao and Kun for his productive conversations and brainstormings. I am also deeply thankful to Ramazan Gokberk and his willingness to help and share useful insights through the numerous e-mail conversations we had during my days in Boston. I would also like to thank Tarique for making my stay in Boston a great time, I felt at home from the very first moment.

I am also really thankful to all my colleagues from the Computer Vision Center, in special to the people I had the great chance to meet during the Master's academic training: Ekain, Lluís-Pere, David, Jon and Anjan. Special thanks also to Carles, Ivet, Fran, Camp, Joan, Alejandro, Yainubis and all the people I shared a lot of precious moments with; some of you have become really good friends.

I feel very lucky to have seen the birth of the Human Pose and Behavior Analysis (HuPBA) research group at the University of Barcelona. I am really thankful to all my colleagues from HuPBA, in special to Miguel, Miguel Ángel, Víctor, Xavi and Albert. Not only I learnt a lot working with you, but also shared unforgettable moments and laughs.

Muchas gracias a mis amigos de Sabadell, con ellos he pasado gran parte de mi vida y momentos que nunca se olvidan. Gracias también a las nuevas amistades que he hecho durante estos años desde que me mudé a Barcelona; en especial a Rubén, Oroitz, Xabi, Maria y Aina.

Quiero agradecer a mis padres el apoyo y afecto que siempre me han brindado, así como la educación que me han dado y la paciencia que siempre han tenido conmigo durante todos estos años. Os quiero mama y papa.

Per últim però no per això menys important, vull agraïr a la Gisela el seu afecte i la seva paciència. Em sento enormement afortunat d'haver-te conegut durant els anys d'aquest viatge. T'estimo moltíssim.

Abstract

The visual analysis of humans from images is an important topic of interest due to its relevance to many computer vision applications like pedestrian detection, monitoring and surveillance, human-computer interaction, e-health or content-based image retrieval, among others.

In this dissertation we are interested in learning different visual representations of the human body that are helpful for the visual analysis of humans in images and video sequences. To that end, we analyze both RGB and depth image modalities and address the problem from three different research lines, at different levels of abstraction; from pixels to gestures: human segmentation, human pose estimation and gesture recognition.

First, we show how binary segmentation (object vs. background) of the human body in image sequences is helpful to remove all the background clutter present in the scene. The presented method, based on Graph cuts optimization, enforces spatio-temporal consistency of the produced segmentation masks among consecutive frames. Secondly, we present a framework for multi-label segmentation for obtaining much more detailed segmentation masks: instead of just obtaining a binary representation separating the human body from the background, finer segmentation masks can be obtained separating the different body parts.

At a higher level of abstraction, we aim for a simpler yet descriptive representation of the human body. Human pose estimation methods usually rely on skeletal models of the human body, formed by segments (or rectangles) that represent the body limbs, appropriately connected following the kinematic constraints of the human body. In practice, such skeletal models must fulfill some constraints in order to allow for efficient inference, while actually limiting the expressiveness of the model. In order to cope with this, we introduce a top-down approach for predicting the position of the body parts in the model, using a mid-level part representation based on Poselets.

Finally, we propose a framework for gesture recognition based on the bag of visual words framework. We leverage the benefits of RGB and depth image modalities by combining modality-specific visual vocabularies in a late fusion fashion. A new rotation-variant depth descriptor is presented, yielding better results than other state-of-the-art descriptors. Moreover, spatio-temporal pyramids are used to encode rough spatial and temporal structure. In addition, we present a probabilistic reformulation of Dynamic Time Warping for gesture segmentation in video sequences. A Gaussian-based probabilistic model of a gesture is learnt, implicitly encoding possible deformations in both spatial and time domains.

Resum

La anàlisi visual de persones a partir d'imatges és un tema de recerca molt important, degut a la rellevància que té a una gran quantitat d'aplicacions dins la visió per computador, com per exemple: detecció de vianants, monitorització i vigilància, interacció persona-màquina, *e-salut*, o sistemes de recuperació d'imatges a partir de contingut, entre d'altres.

En aquesta tesi volem aprendre diferents representacions visuals del cos humà, que siguin útils per a la anàlisi visual de persones en imatges i vídeos. Per a tal efecte, analitzem diferents modalitats d'imatge com son les imatges de color RGB i les imatges de profunditat, i adrecem el problema a diferents nivells d'abstracció, des dels píxels fins als gestos: segmentació de persones, estimació de la pose humana i reconeixement de gestos.

Primer, mostrem com la segmentació binària (objecte vs. fons) del cos humà en seqüències d'imatges ajuda a eliminar soroll pertanyent al fons de l'escena en qüestió. El mètode presentat, basat en optimització *Graph cuts*, imposa consistència espai-temporal a les màscares de segmentació obtingudes en *frames* consecutius. En segon lloc, presentem un marc metodològic per a la segmentació multi-classe, amb la qual podem obtenir una descripció més detallada del cos humà: en comptes d'obtenir una simple representació binària separant el cos humà del fons, podem obtenir màscares de segmentació més detallades, separant i categoritzant les diferents parts del cos.

A un nivell d'abstracció més alt, tenim com a objectiu obtenir representacions del cos humà més simples, tot i ésser suficientment descriptives. Els mètodes d'estimació de la pose humana sovint es basen en models esqueletals del cos humà, formats per segments (o rectangles) que representen les extremitats del cos, connectades unes amb altres seguint les restriccions cinemàtiques del cos humà. A la pràctica, aquests models esqueletals han de complir certes restriccions per tal de poder aplicar mètodes d'inferència que permeten trobar la solució òptima de forma eficient, però a la vegada aquestes restriccions suposen una gran limitació en l'expressivitat que aquests models son capaços de capturar. Per tal de fer front a aquest problema, proposem un enfoc *top-down* per a predir la posició de les parts del cos del model esqueletal, introduint una representació de parts de mig nivell basada en *Poselets*.

Finalment, proposem un marc metodològic per al reconeixement de gestos, basat en els *bag of visual words*. Aprofitem els avantatges de les imatges RGB i les imatges de profunditat combinant vocabularis visuals específics per a cada modalitat, emprant *late fusion*. Proposem un nou descriptor per a imatges de profunditat invariant a rotació, que millora l'estat de l'art, i fem servir piràmides espai-temporals per capturar certa estructura espaial i temporal dels gestos. Addicionalment, presentem una reformulació probabilística del mètode *Dynamic Time Warping* per al reconeixement de gestos en seqüències d'imatges. Més específicament, modelem els gestos amb un model probabilistic Gaussià que implícitament codifica possibles deformacions tant en el domini espaial com en el temporal.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

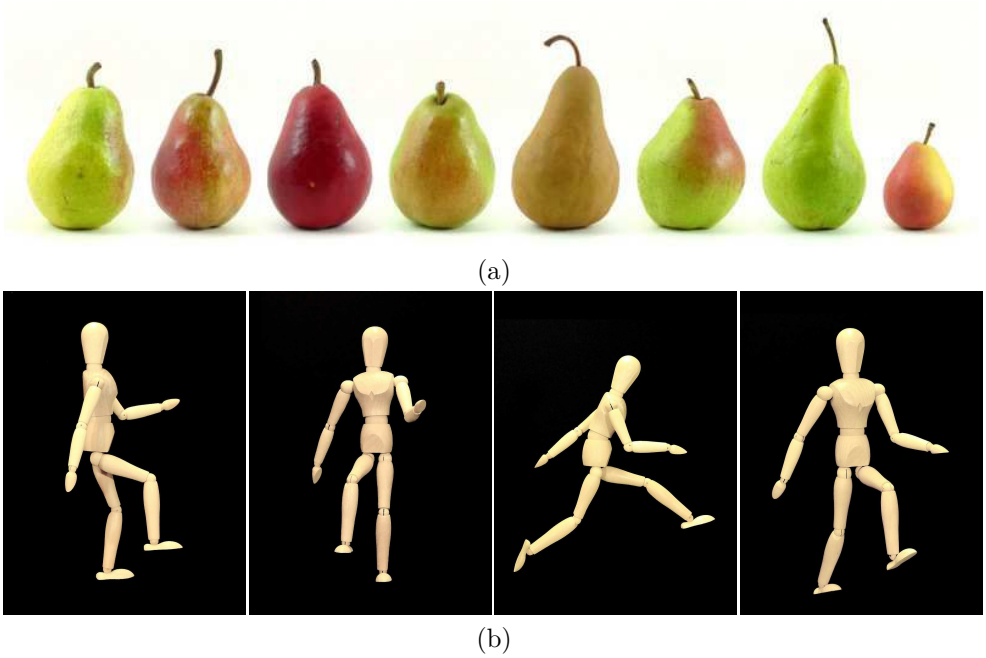
1.1 Motivation

In the quest for artificial intelligence, the ultimate goal could be envisioned as the creation of "intelligent" robots, i.e. robots which have an "intelligent" behavior, following the guidelines of our human intelligence. This intelligence can be defined in many different ways, depending on the different processes that take place in our brains in order to perform a certain task. Among them all, it has been proven that visual intelligence –the reasoning processes related to our visual system– plays an important role in our global intelligent behavior, and this fact can be easily illustrated by imagining ourselves performing daily tasks like commuting or even preparing breakfast without perceiving and processing visual information.

Visual intelligence can be defined as understanding the 3-D spatial world that surrounds us from 2-D projections of it, *i.e.* images, captured by our retinae. While we are not aware of it, our brain is constantly processing visual information in order to understand our environment, and it is pretty good at it, though it is a really challenging task. In our daily lives we are generally able to recognize objects, people and faces without much trouble, even though our 3-D world is full of objects that occlude each other and we are able to infer it from just 2-D projections. Moreover, we are able to recognize objects under different viewpoints or projections, usually implying a change in their appearance. However, when trying to accomplish these tasks artificially by means of computers, we realize the challenging nature of the problem and the power of our brains.

Computer vision is the field of Artificial Intelligence dedicated to the acquisition and processing of images, trying to replicate the human visual intelligence using computer software and hardware. Visual tasks like object recognition have been vastly researched during many years and they are still a challenge for researchers in the field. The main problems to face when designing algorithms for object recognition are: changes in illumination, changes in viewpoint, presence of occlusions or object class variability, among others. While some objects may have little appearance variations in size, color or shape things get complicated in the case of articulated objects composed by movable parts (see Fig. 1.1). Such deformable objects can eventually change their shape appearance considerably, thus complicating the process of learning patterns of their appearance.

Besides object detection and recognition, people detection has caught the attention of many researchers, because of its multiple applications, *e.g.* pedestrian detection, monitoring and surveillance, human-computer interaction e-health, or content-based image retrieval. Detecting people in images is challenging, in the first place, due to the the articulated



(a)

(b)

Figure 1.1: (a) Pears are an example of objects which are simple to detect, since little variations can be found among different samples. (b) In contrast, articulated objects can suffer significant changes in their shape given their high deformability, hence are harder to detect.

nature of the human body: people can adopt a wide range of poses and consequently, the body shape has a large variability. Not only that, but certain poses may also incur in self-occlusions of some body parts, thus making it more difficult to detect. In addition, different clothing can also result in slight changes in the shape of the body, plus a significant change in color appearance. Finally, humans are animated entities that are able to perform different actions with different purposes in comparison with static objects. Therefore, understanding images of people is much more complicated, since human behavior or social signal come into play (see Fig. 1.2).

In contrast to common RGB images used in Computer Vision, range images (*a.k.a.* depth images) provide additional information about the 3-D world, allowing to capture the depth information of each pixel in the image, *i.e.* we know the distance from the camera sensor to points in the scene. Therefore, range images provide a 2.5-D projection of the real world, in contrast to the 2-D projections of common RGB images, allowing for more robust object detection methodologies due to the richer description of the scenes and other additional properties like invariance to changes in illumination, for example. The main issue with range imaging is sensor device technologies like Time-of-flight are very expensive, so access to this kind of cameras is budget-limited. However, the release of low-cost consumer depth cameras like the Kinect from Microsoft in late 2010, supposed an affordable range imaging solution, and many researchers in computer vision made contributions based on multi-modal RGBD (RGB plus Depth) data. As a consequence, a lot of progress has been done in the computer vision community during the past few years, especially in the fields of human

(a) (b)

Figure 1.2: Understanding still life scenes (a) just requires to detect the objects it is composed of. In contrast, understanding scenes of people (b) entail human pose detection, facial expression recognition, or gesture recognition (in the case of video sequences).

pose estimation and gesture recognition for human-computer interaction. Nevertheless, such low-cost range imaging solutions still present some issues in outdoor applications, rendering them almost useless in those cases, in front of RGB cameras.

1.2 Objective of this thesis

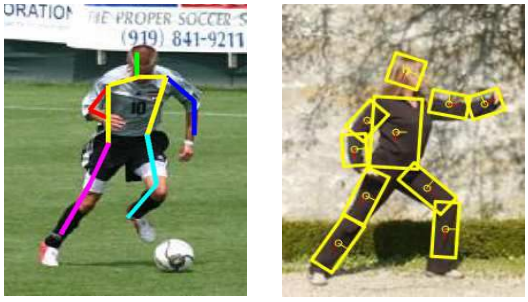
In this thesis, we are interested in learning different visual representations of the human body that are helpful for the visual analysis of humans in images and video sequences. To that end, we analyze both RGB and depth image modalities and address the problem from three different research lines, at different levels of abstraction; from pixels to gestures:

1. **Human body segmentation:** At the lowest level of abstraction, we consider segmentation in order to obtain a pixel-wise representation of the human body. Segmentation methods aim to partition an image in different segments, usually containing different objects or classes of interest. On one hand, we consider binary segmentation (object vs. background) as a pre-processing step in order to remove all the background clutter. After that, further techniques for a deeper analysis of the human body (*e.g.* human pose estimation) can be applied in much smaller image regions of interest, where the actual body is located. On the other hand, multi-label segmentation methods can be also applied for a much more detailed pixel categorization; instead of just obtianing a binary representation separating the human body from the background, finer segmentation masks can be obtained separating the different body parts (see Fig 1.3).

2. **Human pose estimation:** At a higher level of abstraction, we aim for a simpler yet descriptive representation of the human body. Human pose estimation methods usually rely on skeletal models of the human body, formed by segments (or rectangles) that represent the body limbs, appropriately connected following the kinematic constraints of the human body (see Fig. 1.4). Estimating the pose of a person is then formulated as inferring the 2-D or 3-D position and orientation of the body limbs. This information can be then used as an intermediate image descriptor for higher-level semantic reasoning about the actions or activities being performed by the subjects in the image.

(a) (b)

Figure 1.3: (a) Binary human body segmentation. (b) Multi-label human body segmentation.



(a) (b)

Figure 1.4: Skeleton-based representations of the human body formed by (a) segments, and (b) rectangles.

3. **Gesture recognition:** A deeper analysis and understanding of human behavior from visual information, usually requires to take into account the temporal dimension, *i.e.* to process video sequences instead of just still images. Topics like gesture recognition aim to detect specific motion patterns outlined by different body parts along time. Usually, these motion patterns have to be put in correspondence with finer-grained spatial configurations of the body parts, *e.g.* the hands, in order to detect complex gestures like in sign language (see Fig. 1.5), for example. Furthermore, gesture recognition can be used for recognizing higher-level semantic units related to human behavior, *e.g.* human activity recognition.

1.3 Contributions

We summarize the contributions of this thesis in the following list, classified by the corresponding research line:

Human body segmentation

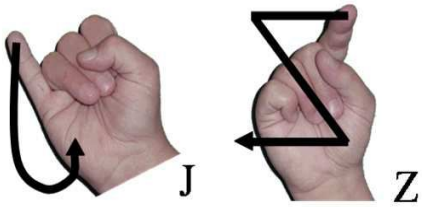


Figure 1.5: Gestures for letters "J" and "Z" in the American Sign Language.

- We propose a fully-automatic method for binary segmentation of people (*i.e.* segmenting the human body from the background) appearing in video sequences. Our proposed method extends the formulation of GrabCut to video sequences, incorporating spatio-temporal consistency by means of Mean Shift clustering (spatial consistency) and a mask initialization algorithm that enforces smooth changes between consecutive frames (temporal consistency).
- We present a generic framework for spatio-temporally consistent multi-label object segmentation based on Random Forest classification and Graph-cuts theory. The presented methodology is applied to human limb segmentation in depth data, yielding a more detailed segmentation of the human body compared to a simple foreground/background mask.

Human pose estimation

- We propose a contextual rescoring methodology for predicting the position of body parts in a human pose estimation framework based on pictorial structures. This rescoring approach encodes high-order body part dependencies by means of a mid-level part representation, yielding more confident position predictions of the body parts while keeping a tree-structured CRF topology in the pictorial structure framework.
- We propose an algorithm for the fully-automatic discovery of a compact contextual mid-level part representation based on Poselets. This contextual representation is able to capture pose-related information that is exploited by our proposed contextual rescoring methodology for human pose estimation.

Gesture recognition

- We propose a framework for gesture recognition based on the bag of visual words framework. We leverage the benefits of RGB and depth image modalities by combining modality-specific visual vocabularies in a late fusion fashion. A new rotation-variant depth descriptor is presented, yielding better results than other state-of-the-art descriptors. Moreover, spatio-temporal pyramids are used to encode rough spatial and temporal structure.
- We present a probabilistic reformulation of Dynamic Time Warping for gesture segmentation in video sequences. A Gaussian-based probabilistic model of an idle gesture is learnt, implicitly encoding possible deformations in both spatial and time domains.

1.4 Thesis outline

This thesis is divided in three self-contained main parts, one for each of the three lines of research we followed: Human body segmentation, Human pose estimation, and Gesture

recognition. For the reader's convenience, the symbol notation of each part is summarized in a table at the beginning of each part. Given the multidisciplinary nature of this thesis, most of the chapters are structured in a similar way, including an introduction, then presenting related work, method formulation, experimental section and a final discussion section summarizing the contributions.

In Part I, we start in chapter 2 by introducing the basis of Graph cuts optimization for both the binary case and its multi-label generalization, used in the following chapters. In chapter 3 we present a method for binary segmentation of subjects in video sequences using graph cuts theory. Finally, in chapter 4 we take advantage of the multi-label generalization of the graph cuts framework to present a methodology for multi-limb segmentation of upper bodies in multi-modal video sequences including RGB and depth data.

Part II is dedicated to Human Pose Estimation. Chapter 5 introduces two widely used frameworks for articulated object detection and consequently, for the problem of human pose estimation: deformable part models and pictorial structures. In chapter 6 we present a contextual rescoring method for obtaining more robust part detections in part-based object detection frameworks like those introduced in chapter 5.

Part III contains our contributions in Gesture Recognition. In chapter 7 we present a Bag-of-Visual-and-Depth-Words representation for gesture recognition in multi-modal RGBD data sequences. In Chapter 8 we propose an extension of Dynamic Time Warping by defining a distance metric based on a probabilistic formulation.

Finally, conclusions and contributions of the thesis are summarized in chapter 9, as well as future lines of research.

In the appendices we present the list of contributions resulting from the work presented in this thesis. We first present the codes and datasets made publicly available to the community. Finally, we list the publications regarding the content of this thesis.

Part I

Human body segmentation

Symbol notation in Part I

Table 1.1: Symbols and conventions for chapters 2- 4

$\mathcal{G} =< \mathcal{V}, \mathcal{E} >$	Graph formed by a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$
$\mathcal{P}$	Set of non-terminal nodes
$\{s, t\}$	Set of terminal nodes: source s and sink t
$\mathcal{N}$	Set of non-terminal edges
$\mathcal{T} = \{t_p^s, t_p^t \mid p \in \mathcal{P}\}$	Set of terminal links
$\mathcal{C} = \{\mathcal{P}_s, \mathcal{P}_t\}$	A cut on $\mathcal{G}$: a binary partition of nodes $\mathcal{P}$ into $\mathcal{P}_s, \mathcal{P}_t$
$\mathcal{L}$	Set of labels
$\mathbf{f} = \{f_p \mid p \in \mathcal{P}\}$	Labeling of non-terminal nodes $\mathcal{P}$
E	Energy function
D	Unary potential in E
V	Pair-wise potential in E
α, β, γ	Random labels
I	Image
$\mathbf{z}$	Array of pixels in I
$z_i = (x_i, y_i, R_i, G_i, B_i)$	Pixel information: spatial coordinates (x_i, y_i) and color components R_i, G_i, B_i
N	Number of pixels in I
$T = \{T_F, T_B, T_U\}$	GrabCut trimap
$\boldsymbol{\theta} = \{\pi, \mu, \sigma\}$	GMM parameters: mixing weights, mean and covariance matrix
$\mathbf{k}$	Array of pixel GMM component assignments
β	Statistics of boundaries in I
Γ	Pair-wise potential weight.
$\mathcal{I} = \{I^t \mid t = 1, ..., M\}$	Video sequence
M	Number of frames in the video sequence $\mathcal{I}$
B	Bounding box around the detected person
R	Central subregion of B
h_{skin}	Skin color model
$\delta(z_i, h_{skin})$	Function that returns a subset of pixels with high likelihood $P(h_{skin} \mid z_i)$
$\mathbf{m}_h$	Mean-shift modes
F	Image segmentation mask
ST_e	Structuring element
O	Overlapping factor
$\mathbf{g}$	Texture vector
μ_g, σ_s^2	Mean and variance of elements of $\mathbf{g}$

$\bar{x}$	Mean shape
$\bar{\mathbf{g}}$	Mean texture
$\mathbf{Q}_s, \mathbf{Q}_g$	Matrices designing modes of variation
$\mathbf{X}$	A shape in the image
$S_t(x)$	Similarity transformation
$\mathbb{E}$	Fitting error
$\Im_F, \Im_R, \Im_L$	Face meshes for frontal, right lateral and left lateral views
$L = \{l_i \mid i = 1, ..., P\}$	Configuration of body parts l_i
$l_i = (x, y, o)$	Parametrization of part l_i: postion (x, y) and orientation o
Υ	Unary potential in energy function for human pose estimation
Ψ	Pair-wise potential in energy function for human pose estimation
$\lambda \in \Lambda$	Random tree in random forest Λ
$\psi_\theta(z_i)$	Depth comparison feature for pixel z_i
$\theta = (u, v)$	A pair of offsets
Φ	Set of node splitting criteria $\phi = (\theta, \tau)$
τ	Threshold on ψ_θ
Z	Set of random training pixels to train λ
Z_L, Z_R	Subsets of pixels resulting from a splitting node in λ
$G(\cdot)$	Information gain function
$H(\cdot)$	Entropy function
$P_\lambda(c \mid z_i)$	Probability density function stored at the leafs of λ, $c \in \mathcal{L}$
d_{max}	Averaged maximum depth level of trees in Λ
O_{max}	Upper limit for the module of offset θ
$\Omega_1(f_i, f_j), \Omega_2(f_i, f_j)$	Label cost functions between labels f_i, f_j
r_c^s	Friedman relative ranking for strategy s and label $c \in \mathcal{L}$
R_s	Friedman mean ranking for strategy s
k	Number of strategies for Friedman test
N_F	Number of experiments for Friedman test

Chapter 2

Graph cuts optimization

2.1 Introduction

Graph cuts optimization has been extensively used in different Computer Vision applications like image segmentation, image restoration, stereo matching, or any other problem that can be formulated as an energy minimization problem [17, 20]. Graph cuts are able to find the optimal solution (the one with minimum energy) in the case of binary problems, and near-optimal approximate solutions in the multi-label case, as long as the defined energy function satisfies some conditions.

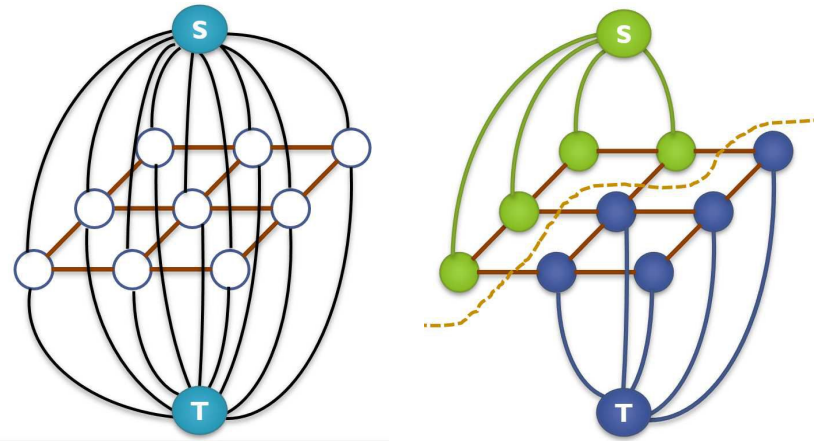
In this chapter we review the theory behind Graph cuts optimization and its properties, which will be later applied in chapters 3- 4 for segmenting the human body.

2.2 Basic concepts

Let $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$ be a graph formed by a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$ connecting them. The set of nodes $\mathcal{V} = \{s, t\} \cup \mathcal{P}$ can be decomposed in two subsets. On one hand, we note the terminal nodes s (*source*) and t (*sink*) (light-blue-filled nodes in Fig. 2.1(a)). On the other, we denote by $\mathcal{P}$ the set of non-terminal nodes (dark-blue-lined nodes in Fig. 2.1(a)).

The edges $\mathcal{E} = \mathcal{N} \cup \mathcal{T}$ can be also divided in two classes. We first denote by $\mathcal{N} = \{(p, q) \in \mathcal{E} \mid p, q \in \mathcal{P}\}$ the set of edges connecting two non-terminal nodes, also referred to as $n-links$ (see brown-colored edges in Fig. 2.1(a)). Secondly, we consider the set of $t-links$, *i.e.* the links connecting a non-terminal node with the terminal nodes s and t: $\mathcal{T} = \{t_p^s, t_p^t \mid p \in \mathcal{P}\}$, where $t_p^s = (s, p), t_p^t = (p, t)$ (black-colored edges in Fig 2.1(a)). Finally, every other edge in the graph is assigned a cost, defined by the energy function that we want to minimize.

A "cut" $\mathcal{C} = \{\mathcal{P}_s, \mathcal{P}_t\}$ of the graph $\mathcal{G}$ is a partitioning of the nodes $\mathcal{P}$ into two disjoint subsets $\mathcal{P}_s$ and $\mathcal{P}_t$, named after the terminal nodes s (source) and t (sink), respectively. Therefore, a cut unequivocally assigns each node $p \in \mathcal{P}$ to one of the $t-links$, producing a labelling $\mathbf{f} = \{f_p \mid p \in \mathcal{P}\}$, where $f_p \in \mathcal{L}$ ($\mathcal{L} = \{0, 1\}$ for the binary case). The cost of a cut $\mathcal{C}$ is then defined as the sum of the costs of edges $\mathcal{E}$ in the graph $\mathcal{G}$. An example of a cut is shown as an orange dotted line in Fig. 2.1(b), and the corresponding partitioning of the nodes in green and purple colors.



(a) (b)

Figure 2.1: (a) Example topology of $\mathcal{G}$ for a typical computer vision application for 2-D images. (b) Example of a cut and the resulting labeling of the nodes.

2.3 Graph topology

In computer vision applications, typical topologies for $\mathcal{G}$ are in the shape of N-D grids, being $N = 2$ the most common case, arranging the nodes in the graph following the 2-D lattice of an image (see Fig. 2.1). Given two contiguous pixels i and j in an image I, nodes $p, q \in \mathcal{P}$ represent them in the graph, and an edge $(p, q) \in \mathcal{N}$ represents their neighborhood property. Similarly, we can imagine a 3-D grid topology in order to process spatio-temporal volumes *i.e.* video sequences.

Different neighboring patterns can be considering to give shape to the N-D grid topologies. The most popular neighborhood patterns in the case of 2-D lattices, are the 4-connectivity and the 8-connectivity. While the former just considers two nodes to be contiguous if they share either the x or the y coordinates (see Fig. 2.1(a-b)), the latter also considers nodes placed in the diagonals. Similarly, typical neighborhood systems in a 3-D grid are 6-connectivity or 26-connectivity (see Fig. 2.1(c-d)).

2.4 Energy minimization in binary problems

The $min - cut/max - flow$ algorithm is able to efficiently find the exact solution $\mathbf{f}$ with minimum energy, as long as the problem is binary ($\mathcal{L} = \{0, 1\}$), and the potentials in the defined energy function are of order 2 at most. In other words, only functions which can be expressed as a sum of functions that take into account at most 2 variables at a time, are allowed. The standard form of such energy functions E is:

$$E(\mathbf{f}) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} V_{p,q}(f_p, f_q), \tag{2.1}$$

where terms $D_p(f_p)$ and $V_{p,q}$ are "unary potential"and the "pair-wise potential", respectively.

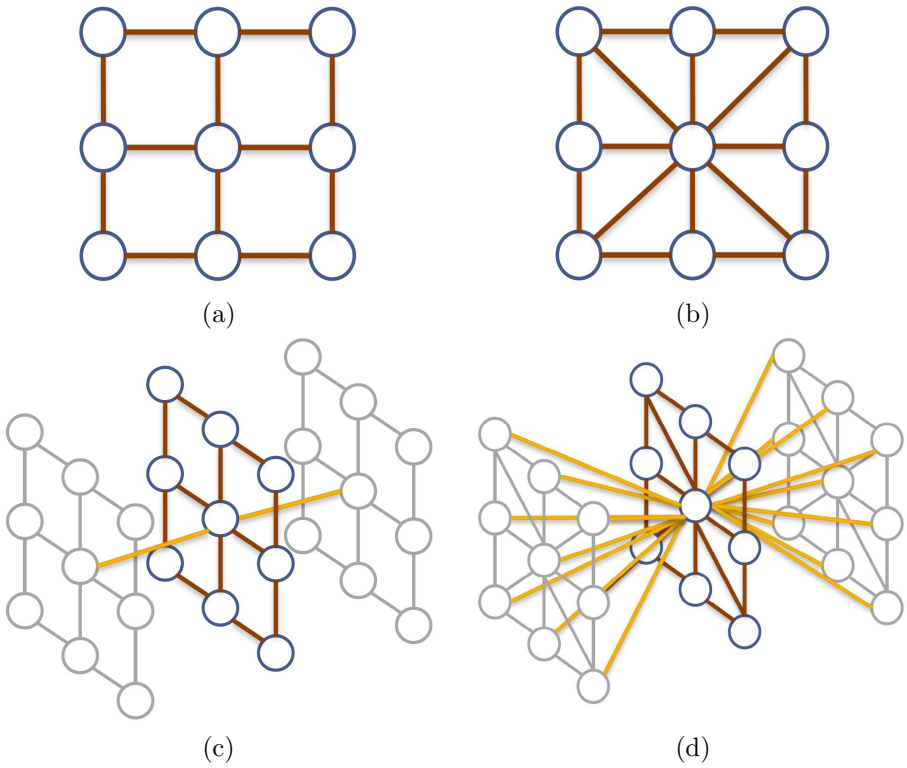


Figure 2.2: Common graph topologies in computer vision applications. 2-D grids (images): (a) 4-connectivity, and (b) 8-connectivity. 3-D grids (video sequences): (c) 6-connectivity, and (d) 26-connectivity (brown edges show intra-frame connections, yellow edges show inter-frame connections).

2.4.1 Unary potential

The unary potential D_p in Eq. 2.1, computes the cost of assigning the label f_p to node p, based on observed data. This cost is assigned to edges t_p^s and t_p^t in case $f_p = 1$ and $f_p = 0$, respectively.

A common definition of the unary potential in many computer vision applications is the negative log likelihood:

$$D_p(f_p) = -\log P(p \mid f_p). \tag{2.2}$$

This way, the likelihood probability $P(p \mid f_p)$ (which we are interested in maximizing in our solution) is converted to a cost, so finding the minimum value (remember we are interested in minimizing E) corresponds to finding the maximum likelihood.

2.4.2 Pair-wise potential

The pair-wise potential ($V_{p,q}$ in Eq. 2.1) is the responsible for assigning costs to edges $(p, q) \in \mathcal{N}$, *i.e.* the $n-links$ connecting two non-terminal nodes, and measures the cost of assigning

labels f_p, f_q to contiguous nodes p, q, based on observed data like in the case of the unary potential. This pair-wise potential is meant to enforce "smoothness" in the solution, fostering similar pixels to have the same label. Therefore, $V_{p,q}$ is meant to be a non-convex function of $|f_p - f_q|$, *i.e.* a discontinuity-preserving function.

An important and widely used example for such discontinuity-preserving function is the Potts model:

$$V_{p,q}(f_p, f_q) = [f_p \neq f_q], \tag{2.3}$$

where $[\chi]$ is an indicator function that takes the value 1 if condition χ is satisfied, or 0 otherwise. Therefore, this model encourages solutions consisting on different regions such that pixels in the same region are assigned the same label.

2.5 Multi-label generalization

Although graph cuts optimization is inherently binary, the same framework has been also extended to multi-label problems [1], *i.e.* when $|\mathcal{L}| > 2$. The generalization of a binary cut $\mathcal{C} = \{\mathcal{S}, \mathcal{T}\}$ to a multi-label case can be then formulated as a partitioning $\mathbf{P} = \{\mathcal{P}_l \mid l \in \mathcal{L}\}$, where $\mathcal{P}_l = \{p \in \mathcal{P} \mid f_p = l\}$.

Given that $|\mathcal{L}| > 2$, the number of label combinations in the boundaries of a possible cut $\mathcal{C}$ can be much higher (it grows quadratically on $|\mathcal{L}|$). Hence, more interesting versions of the Potts model can be taken into consideration where different values can be assigned to different pairs of labels $\alpha, \beta \in \mathcal{L}$.

Exact multi-label optimization is only possible when labels $\mathcal{L}$ can be linearly ordered, and the pair-wise potential is defined as a specific convex function $V_{p,q} = |f_p - f_q|$, but in practice for computer vision applications, the obtained result is oversmoothed. However, there are more interesting algorithms that get approximate solutions: α-β swap and α-expansion. We review them in the following subsections.

2.5.1 α-β swap

This algorithm is able to find an approximate solution (without any guarantee of closeness to the optimal solution), as long as the pair-wise potential satisfies the following conditions for any pair of labels $\alpha, \beta \in \mathcal{L}$:

$$V(\beta, \alpha) = V(\alpha, \beta) \geq 0 \tag{2.4}$$

$$V(\alpha, \beta) = 0 \iff \alpha = \beta. \tag{2.5}$$

In case the above mentioned conditions are satisfied, we call V a semi-metric on the space of labels $\mathcal{L}$.

The α-β swap algorithm (see Algorithm 1) is based on the concept of a "swap" move, as its name indicates. A "swap" move between labels $\alpha, \beta \in \mathcal{L}$ is summarized as generating a new labeling $\mathbf{f}'$ (partitioning $\mathbf{P}'$) from an arbitrary labeling $\mathbf{f}$, such that $\mathcal{P}_l = \mathcal{P}'_l$ for any label $l \neq \alpha, \beta$. In other words, an α-β swap move from $\mathbf{f}$ to $\mathbf{f}'$ can just swap labels from α to β, or viceversa.

We denote by $G_{\alpha\beta} =< \mathcal{V}_{\alpha\beta}, \mathcal{E}_{\alpha\beta} >$ the graph construction for multi-label optimization with α-β swap, which is very similar to that presented for the binary case, $\mathcal{G}$. In this case, the set of nodes $\mathcal{V}_{\alpha\beta} = \{\alpha, \beta\} \cup \mathcal{P}_{\alpha\beta}$ is formed by terminal nodes α and β, plus non-terminal

[1] While we use here the term "multi-label" following the bibliography, we think it would be more appropriate to use "multi-class". Please note that in this case, a variable can only take one possible value, in contrast to "multi-label" classification problems.

Algorithm 1 α-β swap algorithm

1: Start with labeling $\mathbf{f}$
2: $success \leftarrow 0$
3: **for** each pair of labels $\{\alpha, \beta\} \subset \mathcal{L}$ **do**
4: Find $\hat{\mathbf{f}} = \arg\min E(\mathbf{f}')$ among $\mathbf{f}'$ within one α-β swap of $\mathbf{f}$
5: **if** $E(\hat{\mathbf{f}}) < E(\mathbf{f})$ **then**
6: $\mathbf{f} \leftarrow \hat{\mathbf{f}}$
7: $success \leftarrow 1$
8: **end if**
9: **end for**
10: **if** $success = 1$ **then**
11: go to 2
12: **end if**
13: **return** $\mathbf{f}$

Table 2.1: Weights of edges $\mathcal{E}_{\alpha\beta}$ in $\mathcal{G}_{\alpha\beta}$.

edge	weight (cost)	for
t_p^α	$D_p(\alpha) + \sum_{q \in \mathcal{N}_p \mid q \notin \mathcal{P}_{\alpha\beta}} V_{p,q}(\alpha, f_q)$	$p \in \mathcal{P}_{\alpha\beta}$
t_p^β	$D_p(\beta) + \sum_{q \in \mathcal{N}_p \mid q \notin \mathcal{P}_{\alpha\beta}} V_{p,q}(\beta, f_q)$	$p \in \mathcal{P}_{\alpha\beta}$
(p,q)	$V_{p,q}(\alpha, \beta)$	$(p,q) \in \mathcal{N}, p, q \in \mathcal{P}_{\alpha\beta}$

nodes $\mathcal{P}_{\alpha\beta} = \mathcal{P}_\alpha \cup \mathcal{P}_\beta$. Similarly, the set of edges $\mathcal{E}_{\alpha\beta} = \mathcal{N} \cup \mathcal{T}_{\alpha\beta}$, where $\mathcal{N}$ is the set of edges connecting non-terminal nodes, and $\mathcal{T}_{\alpha\beta} = \{t_p^\alpha, t_p^\beta \mid p \in \mathcal{P}_{\alpha\beta}\}$. The assignment of costs to edges $\mathcal{E}_{\alpha\beta}$ is summarized in Table 2.1.

Finally, energy minimization is performed by a $min - cut$ algorithm on $\mathcal{G}_{\alpha\beta}$, like in the binary case. It has been proven by Boykov *et al.* [19] that finding the solution $\hat{\mathbf{f}}$ with minimum energy (step 4 in Algorithm 1) is equivalent to finding the minimum cut $\mathcal{C}$ on $\mathcal{G}_{\alpha\beta}$.

2.5.2 α-expansion

In contrast to the previous algorithm, α-expansion ensures finding a solution within a known factor (as small as 2) from the optimal one. However, in addition to conditions in Eq. 2.4 and Eq. 2.5, the following condition must be also satisfied:

$$V(\alpha, \beta) \leq V(\alpha, \gamma) + V(\gamma, \beta), \tag{2.6}$$

for any $\alpha, \beta, \gamma \in \mathcal{L}$. If all these conditions are satisfied, then we say that V is a metric on the space of labels $\mathcal{L}$, and α-expansion can be successfully applied.

An α-expansion move from $\mathbf{f}$ to $\mathbf{f}'$ allows any set of nodes to change their labels to α. Therefore, $\mathcal{P}_\alpha \subset \mathcal{P}'_\alpha$, and $\mathcal{P}'_l \subset \mathcal{P}_l$ for any $l \neq \alpha$. All the steps of α-expansion are shown in Algorithm 2.

Algorithm 2 α-expansion algorithm

1: Start with labeling $\mathbf{f}$
2: $success \leftarrow 0$
3: **for** each label $\alpha \subset \mathcal{L}$ **do**
4: Find $\hat{\mathbf{f}} = \arg\min E(\mathbf{f}')$ among $\mathbf{f}'$ within one α-expansion of $\mathbf{f}$
5: **if** $E(\hat{\mathbf{f}}) < E(\mathbf{f})$ **then**
6: $\mathbf{f} \leftarrow \hat{\mathbf{f}}$
7: $success \leftarrow 1$
8: **end if**
9: **end for**
10: **if** $success = 1$ **then**
11: go to 2
12: **end if**
13: **return** $\mathbf{f}$

The graph construction for α-expansion is quite different from the previous case. In this case, a graph $\mathcal{G}_\alpha =< \mathcal{V}_\alpha, \mathcal{E}_\alpha >$ is built, where the set of nodes $\mathcal{V}_\alpha$ contains not only the terminal nodes α and $\bar{\alpha}$ and non-terminal nodes $\mathcal{P}$, but an additional set of arbitrary non-terminal nodes $a_{p,q}$:

$$\mathcal{V}_\alpha = \{\alpha, \bar{\alpha}\} \cup \mathcal{P} \cup \bigcup_{(p,q)\in\mathcal{N}, f_p \neq f_q} a_{p,q}. \tag{2.7}$$

Arbitrary nodes $a_{p,q}$ are introduced in the graph connecting neighboring nodes $p, q \in \mathcal{N}$ that have different assigned labels, *i.e.* $f_p \neq f_q$ (see Fig. 2.3). The set of edges is then defined as:

$$\mathcal{E}_\alpha = \{\bigcup_{p\in\mathcal{P}} \{t_p^\alpha, t_p^{\bar{\alpha}}\}, \bigcup_{(p,q)\in\mathcal{N}, f_p\neq f_q} \mathcal{E}_{p,q}, \bigcup_{(p,q)\in\mathcal{N}, f_p = f_q} (p,q)\}, \tag{2.8}$$

where $t_p^\alpha, t_p^{\bar{\alpha}}$ are the usual $t-links$, and $\mathcal{E}_{p,q} = \{(p,a),(a,q),t_p^{\bar{\alpha}a}\}$ is a triplet containing non-terminal links connecting nodes p and q (which are assigned different labels) through an arbitrary node $a = a_{p,q}$, and a $t-link$ $t_a^{\bar{\alpha}}$ connecting a to the $\bar{\alpha}$ terminal node. Table 2.2 summarizes the assignment of costs to edges $\mathcal{E}_\alpha$.

With such graph construction (see Fig. 2.3) and the cost assignment summarized in Table 2.2, energy minimization is performed via $min-cut$, like in the α-β swap case. Again, finding the minimum cut $\mathcal{C}$ on $\mathcal{G}_\alpha$ is equivalent to find the solution $\hat{\mathbf{f}}$ with minimum energy (step 4 in Algorithm 2).

Table 2.2: Weights of edges $\mathcal{E}_\alpha$ in $\mathcal{G}_\alpha$.

edge	weight (cost)	for
$t_p^{\bar{\alpha}}$	∞	$p \in \mathcal{P}_\alpha$
$t_p^{\bar{\alpha}}$	$D_p(f_p)$	$p \notin \mathcal{P}_\alpha$
t_p^{α}	$D_p(\alpha)$	$p \in \mathcal{P}_\alpha$
(p, a)	$V_{p,q}(f_p, \alpha)$	$(p,q) \in \mathcal{N}, f_p \neq f_q$
(a, q)	$V_{p,q}(\alpha, f_q)$	
$t_a^{\bar{\alpha}}$	$V_{p,q}(f_p, f_q)$	
(p, q)	$V_{p,q}(f_p, \alpha)$	$(p,q) \in \mathcal{N}, f_p = f_q$

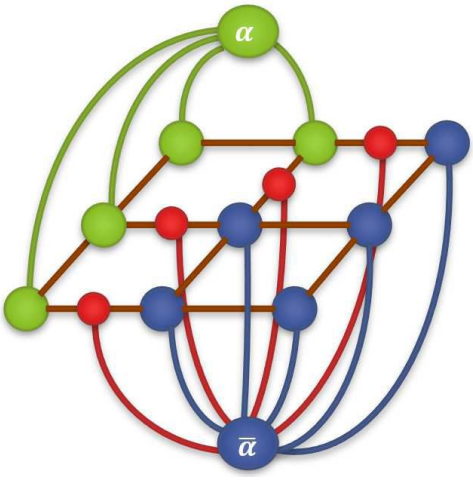


Figure 2.3: Graph topology $\mathcal{G}_\alpha$ for α-expansion energy minimization. Additional aribtrary nodes $a_{p,q}$ and respective $t-links$ $t_p^{\bar{\alpha}}$ are depicted in red color.

Chapter 3

Binary human segmentation

3.1 Introduction

One of the main problems that computer vision methodologies have to face when analyzing real-world scenes is the background clutter. Many objects can appear in a given scene, while we might be interested just in some of them, depending on the application. In our case, we are interested in understanding humans, so we would like to separate human shapes from the background.

In this chapter, we present a fully-automatic Spatio-Temporal GrabCut human segmentation methodology for video sequences that combines tracking and segmentation. GrabCut initialization is performed by a HOG-based person detection, face detection, and skin color model. Spatial information is included by Mean Shift clustering whereas temporal coherence is considered by the historical of foreground and background color models computed from previous frames, as well as segmentation initialization for upcoming frames. Finally, we show how segmentation can help higher-level human understanding processes like face alignment, or human pose estimation. Results over public datasets including a new Human Limb dataset show a robust segmentation and recovery of both face and body pose using the presented methodology.

3.2 Related work

Human segmentation in uncontrolled environments is a hard task because of the constant changes produced in natural scenes: illumination changes, moving objects, changes in the point of view or occlusions, just to mention a few. Because of the nature of the problem, a common way to proceed is to discard most parts of the image so that the analysis can be performed on a reduced set of small candidate regions. In [30], the authors propose a full-body detector based on a cascade of classifiers [88] using HOG features. This methodology is currently being used in several works related to the pedestrian detection problem [42]. GrabCut [74] has also shown high robustness in Computer Vision segmentation problems, defining the pixels of the image as nodes of a graph and extracting foreground pixels via iterated Graph cuts optimization. This methodology has been applied to the problem of human body segmentation with high success [39, 40]. In the case of working with sequences of images, this optimization problem can also be considered to have temporal coherence. In the work of [28], the authors extended the Gaussian Mixture Model (GMM) of Grab-

Cut algorithm so that the color space is complemented with the derivative in time of pixel intensities in order to include temporal information in the segmentation optimization process. However, the main problem of that method is that moving pixels correspond to the boundaries between foreground and background regions, yielding an unclear discrimination between the two classes.

Once a region of interest is determined, pose is often recovered by the determination of the body limbs together with their spatial coherence (also with temporal coherence in case of image sequences). Most of these approaches are probabilistic, and features are usually based on edges or appearance. In [71], the author propose a probabilistic approach for limb detection based on edge learning complemented with color information. The body is modeled as a Conditional Random Field (CRF) where each node represents a different body limb, and they are connected following a tree structure, so optimization can be performed via belief propagation. This method has obtained robust results and has been extended by other authors including local GrabCut segmentation and temporal refinement of the CRF model [39, 40].

3.3 GrabCut segmentation

In this section we review the GrabCut segmentation method, given its relevance to the proposed methodology for automatic human segmentation in video sequences.

In [74], the authors proposed an approach to find a binary segmentation (background and foreground) of an image by formulating an energy minimization scheme, and solving it using graph cuts [17, 20, 55], extended to color images (instead of gray-scale ones).

Given a color image I, let us consider the array $\mathbf{z} = (z_1, ..., z_i, ..., z_N)$ of N pixels where $z_i = (x_i, y_i, R_i, G_i, B_i)$, contains the spatial coordinates (x_i, y_i) and the RGB values R_i,G_i,B_i. The segmentation is defined as array $\mathbf{f} = (f_1, ...f_N)$, $f_i \in \{0, 1\}$, assigning a label to each pixel of the image indicating if it belongs to background or foreground. A trimap T is defined by the user (in a semi-automatic way) consisting of three regions: T_B, T_F and T_U, each one containing initial background, foreground, and uncertain pixels, respectively. Pixels belonging to T_B and T_F are clamped as background and foreground respectively—which means GrabCut will not be able to modify these labels, whereas those belonging to T_U are actually the ones the algorithm will be able to label. Color information is introduced by GMMs over the RGB components of pixels in $\mathbf{z}$. A full covariance GMM of K components is defined for background pixels ($f_i = 0$), and another one for foreground pixels ($f_i = 1$), parametrized as follows

$$\boldsymbol{\theta} = \{\pi(f, k), \mu(f, k), \Sigma(f, k), f \in \{0, 1\}, k = 1..K\}, \tag{3.1}$$

being π the mixing weights, μ the means and Σ the covariance matrices of the model. We also consider the array $\mathbf{k} = \{k_1, ..., k_i, ...k_N\}$, $k_i \in \{1, ...K\}$, $i \in \{1, ..., N\}$ indicating the component of the background or foreground GMM (according to f_i) the pixel z_i belongs to. The energy function for segmentation is then formulated as

$$E(\mathbf{f}, k, \boldsymbol{\theta}, z) = \sum_{i=1}^{N} D_i(f_i, k_i, \boldsymbol{\theta}, z_i) + \sum_{\{i,j\} \in \mathcal{N}} V_{i,j}(f_i, f_j, z_i, z_j), \tag{3.2}$$

following the standard form of energy functions suitable for minimizing via graph cuts, as reviewed in chapter 2. Hence, D_i is the unary or likelihood potential for pixel i, based on the probability distributions $p(\cdot)$ of the GMM:

$$D_i(f_i, k_i, \boldsymbol{\theta}, z_i) = -\log p(z_i | f_i, k_i, \boldsymbol{\theta}) - \log \pi(f_i, k_i) \tag{3.3}$$

and V is a the pair-wise potential or regularizing prior assuming that segmented regions should be coherent in terms of color, taking into account a neighborhood $\mathcal{N}$ around each pixel

$$V_{i,j}(f_i, f_j, z_i, z_j) = \Gamma[f_i \neq f_j] \exp(-\beta \|z_i - z_j\|^2), \tag{3.4}$$

where $\Gamma \in \mathbb{R}^+$ is a weight that specifies the relative importance of the pair-wise potential w.r.t. the unary potential, and $\beta = \left(2\langle (z_i - z_j)^2 \rangle\right)^{-1}$ is the expected value of the pixel differences among the whole image. With this energy minimization scheme and given the initial trimap T, the final segmentation is performed using a $min - cut/max - flow$ [17, 18, 20]. The classical semi-automatic GrabCut algorithm is summarized in Algorithm 3.

Algorithm 3 Original GrabCut algorithm

1: $T \leftarrow$ Trimap initialization with manual annotation.
2: $f_i \leftarrow 0, \forall i \in T_B$
3: $f_i \leftarrow 1 \forall i \in T_U \cup T_F$.
4: $\theta \leftarrow$ Initialize Background and Foreground GMMs with k-means, using $\mathbf{f}$.
5: $\mathbf{k} \leftarrow$ Assign GMM components to pixels.
6: $\theta \leftarrow$ Learn GMM parameters from data $\mathbf{z}$.
7: $\mathbf{f} \leftarrow$ Estimate segmentation: Graph cuts ($min - cut$ algorithm).
8: Repeat from step 5, until convergence.

3.4 Spatio-temporal GrabCut segmentation

We propose a fully-automatic Spatio-Temporal GrabCut human segmentation methodology, which benefits from the combination of tracking and segmentation. First, subjects are detected by means of a HOG-based classifier. Face detection and a skin color model are used to define a set of seeds used to initialize GrabCut algorithm. Spatial information is taken into account by means of Mean Shift clustering, whereas temporal information is considered taking into account the pixel probability membership to the color-based Gaussian Mixture Models of the previous frames.

Our proposal is based on the previous GrabCut framework, focusing on human body segmentation, being fully automatic, and extending it to take into account temporal coherence. We define a video sequence $\mathcal{I} = \{I^1, ..., I^t, ..., I^M\}$, composed by M frames. Given a frame I^t, we first apply a person detector based on a classifier over HOG features [30]. Then, we initialize the trimap T from the bounding box B retuned by the detector: $T_U = \{i \mid (x_i, y_i) \in B\}$, $T_B = \{i \mid (x_i, y_i) \notin B\}$. Furthermore, in order to increase the accuracy of the segmentation algorithm, we include Foreground seeds exploiting spatial and appearance prior information. On one hand, we define a small central rectangular region R inside B, proportional to B in such a way that we are sure it corresponds to the torso of the person. Thus, pixels inside R are set to foreground. On the other, we apply a face detector based on a cascade of classifiers using Haar-like features [88] over B, and learn a skin color model h_{skin} consisting of a histogram over the *Hue* channel of the *HSV* image representation. All pixels inside B fitting in h_{skin} are also set to foreground. Therefore, we initialize $T_F = \{i \mid (x_i, y_i) \in R\} \cup \{i \in \delta(z_i, h_{skin})\}$, where δ returns the indices of pixels with high likelihood $P(h_{skin} \mid z_i)$. An example of seed initialization is shown in Figure 4.1(b). Steps 1-5 in Algorithm 4 summarize the trimap initialization.

3.4.1 Spatial Extension

Once we have initialized the trimap, we can initialize the background and foreground GMM models, as shown in step 4 of original GrabCut (Algorithm 3). However, instead of applying k-means for the initialization of the GMMs we propose to use Mean-Shift clustering (step 6 in Algorithm 4), which also takes into account spatial coherence. Given an initial estimation of the distribution modes $m_h(x^0)$ and a kernel function g, Mean-shift iteratively updates the mean-shift vector with the following formula:

$$\mathbf{m}_h(x) = \frac{\sum_{i=1}^{n} z_i g(\| \frac{x - z_i}{h} \|^2)}{\sum_{i=1}^{n} g(\| \frac{x - z_i}{h} \|^2)}, \tag{3.5}$$

until it converges, where n determines the neighborhood of a pixel z_i (CIELuv space is used instead of RGB), and returns the centers of the clusters (distribution modes) found.

3.4.2 Temporal extension

After initializing the color model θ^1, at the first frame of the sequence $\mathcal{I}$, we apply the iterative minimization shown in steps 5-8 in Algorithm 3, obtaining in our case a segmentation $\hat{\mathbf{f}}^t$ of frame I^t (step 10 in Algorithm 4) and the updated foreground and background GMMs $\boldsymbol{\theta}^t$, which are used for further initialization for frame I^{t+1}. The result of this step is shown in Figure 4.1(c). Finally, we refine the segmentation of frame I^t eliminating false positive foreground pixels. By definition of the energy minimization scheme, GrabCut tends to find convex segmentation masks having a lower perimeter, given that each pixel on the boundary of the segmentation mask contributes on the global cost. Therefore, in order to eliminate these background pixels (commonly in concave regions) from the foreground segmentation, we re-initialize the trimap T as follows:

$$\begin{aligned}
T_B &= \{i \mid \hat{f}_i^t = 0\} \cup \\
&\left\{ i \mid \frac{\sum_{u=t-m}^{t} p(z_i^t | \hat{f}_i^t = 0, k_i^t, \boldsymbol{\theta}^u)}{m} > \frac{\sum_{u=t-m}^{t} p(z_i^t | \hat{f}_i^t = 1, k_i^t, \boldsymbol{\theta}^u)}{m} \right\}, \\
T_F &= \{i \in \delta(z_i^t, h_{skin})\}, \\
T_U &= \{i \mid \hat{f}_i^t = 1\} \setminus T_B \setminus T_F,
\end{aligned} \tag{3.6}$$

where the pixel background probability membership is computed using the GMM models of previous m frames (step 12 in Algorithm 4). This formulation can also be extended to detect false negatives. However, in our case we focus on false positives since they appear frequently in the case of human segmentation. The result of this step is shown in Figure 4.1(d). Once the trimap has been redefined, false positive foreground pixels still remain, so the new set of seeds is used to iterate again GrabCut algorithm (steps 13-16 in Algorithm 4), resulting in a more accurate segmentation $\mathbf{f}^t$, as we can see in Figure 4.1(e).

Considering $\hat{F}^t$ as the binary image representing $\hat{\mathbf{f}}^t$ (the one obtained before the refinement), we initialize the trimap for I^{t+1} (step 17 in Algorithm 4) as follows:

$$\begin{aligned}
T_F &= \{i \mid (x_i^t, y_i^t) \in (\hat{F}^t \ominus ST_e)\}, \\
T_U &= \{i \mid (x_i^t, y_i^t) \in (\hat{F}^t \oplus ST_d)\} \setminus T_F, \\
T_B &= \{1, ..., N\} \setminus (T_F \cup T_U),
\end{aligned} \tag{3.7}$$

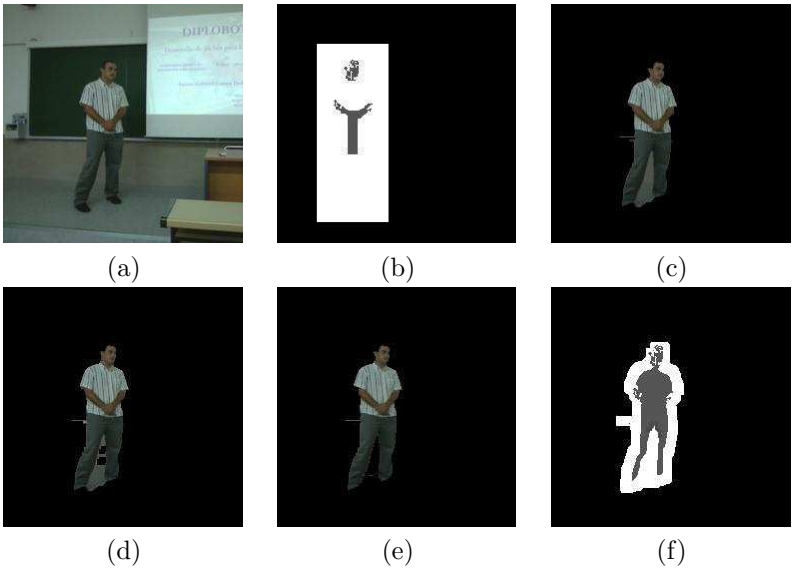


Figure 3.1: STGrabcut pipeline example: (a) Original frame, (b) Seed initialization, (c) GrabCut, (d) Probabilistic re-assignment, (e) Refinement and (f) Initialization mask for I^{t+1}.

where $\ominus$ and $\oplus$ are erosion and dilation operations with their respective structuring elements ST_e and ST_d, and $\setminus$ represents the set difference operation. The structuring elements are simple squares of a given size depending on the size of the person and the degree of movement we allow from I^t to I^{t+1}, assuming smoothness in the movement of the person. An example of a morphological mask is shown in Figure 4.1(f). The whole segmentation methodology is detailed in the ST-GrabCut Algorithm 4.

3.5 Experiments

In this section we provide an experimental evaluation of the proposed ST-GrabCut methodology for human segmentation in video sequences. Moreover, we also show experimentally how the proposed segmentation method can be substantially helpful for other applications related to human understanding, like face alignment and human pose estimation. We first present the data, methods and parameters of the comparative, and the validation measurements. Next, we present quantitative and qualitative results for all the experiments.

3.5.1 Data

We use the public image sequences of the Chroma Video Segmentation Ground Truth (cVSG) [87], a corpus of video sequences and segmentation masks of people. Chroma-based techniques have been used to record Foregrounds and Backgrounds separately, being later combined to achieve final video sequences and accurate segmentation masks almost automatically. Some samples of the sequence we have used for testing are shown in Figure 3.2(a). The sequence has a total of 307 frames. This image sequence includes several critical factors that

Algorithm 4 Spatio-Temporal GrabCut algorithm.

$B \leftarrow$ Person detection on I^1
$h_{skin} \leftarrow$ Face detection and skin color model learning
$T \leftarrow$ Trimap initialization with B and h_{skin}
$f_i^1 \leftarrow 0, \forall i \in T_B$
$f_i^1 \leftarrow 1, \forall i \in T_U \cup T_F$
$\boldsymbol{\theta}^1 \leftarrow$ Initialize GMMs with Mean-shift, using $\mathbf{f}$
for $t = 1$ ... M **do**
  $\mathbf{k}^t \leftarrow$ Assign GMM components to pixels, using T
  $\boldsymbol{\theta}^t \leftarrow$ Learn GMM parameters from data $\mathbf{z}^t$
  $\hat{\mathbf{f}}^t \leftarrow$ Estimate segmentation: Graph cuts ($min - cut$ algorithm)
  Repeat from step 8, until convergence
  $T \leftarrow$ Re-initialize trimap (Equation (3.6))
  $\mathbf{k}^t \leftarrow$ Assign GMM components to pixels, using T
  $\boldsymbol{\theta}^t \leftarrow$ Learn GMM parameters from data $\mathbf{z}^t$
  $\mathbf{f}^t \leftarrow$ Estimate segmentation: Graph cuts ($min - cut$ algorithm)
  Repeat from step 13, until convergence
  $T \leftarrow$ Initialize trimap using $\hat{\mathbf{f}}^t$ (equation 3.7) for I^{t+1}
end for

make segmentation difficult: object textural complexity, object structure, uncovered extent, object size, Foreground and Background velocity, shadows, background textural complexity, Background multimodality, and small camera motion.

As a second database, we have also used a set of 30 videos corresponding to the defense of undergraduate thesis at the University of Barcelona to test the methodology in a different environment (UBDataset). Some samples of this dataset are shown in Figure 3.2(b).

Moreover, we present the Human Limb dataset, a new dataset composed by 227 images from 25 different people. At each image, 14 different limbs are labeled (see Figure 3.2(c)), including the "do not care" label between adjacent limbs, as described in Appendix 3.

3.5.2 Methods

We test the classical semi-automatic GrabCut algorithm for human segmentation comparing it with the proposed ST-GrabCut algorithm. In the case of GrabCut, we set the number of GMM components $k = 5$ for both foreground and background models. Furthermore, the already trained models used for person and face detectors have been taken from the OpenCV 2.1.

We also present a baseline methodology for face alignment, and test it on segmented and unsegmented images in order to show how removing the background can benefit further visual processing of humans. Furthermore, we also show that our segmentation method yields better results when applying a human pose estimation baseline method on the segmented images. The body model used for the pose recovery was taken directly from the work of Ramanan [71].

(a)

(b)

(c)

Figure 3.2: (a) Samples of the cVSG corpus and (b) UBDataset image sequences, and (c) HumanLimb dataset.

3.5.3 Validation measurements

In order to evaluate the robustness of the methodology for human body segmentation, face and body pose estimation, we use the ground truth masks of the images to compute the

overlapping factor O as follows

$$O = \frac{\sum F \cap F_{GT}}{\sum F \cup F_{GT}} \tag{3.8}$$

where F and F_{GT} are the binary masks obtained for spatio-temporal GrabCut segmentation and the ground truth mask, respectively.

3.5.4 Spatio-Tempral GrabCut Segmentation

First, we test the proposed ST-GrabCut segmentation on the sequence from the public cVSG corpus. The results for the different experiments are shown in Table 3.1, in an incremental fashion. In order to avoid the manual initialization of classical GrabCut algorithm, for all the experiments, seed initialization is performed applying the commented person HOG detection, face detection, and skin color model. First row of Table 3.1 shows the overlapping performance of Equation (3.8) applying GrabCut segmentation with k-means clustering to design the GMM models. Second row shows the overlapping performance considering the spatial extension of the algorithm introduced by using Mean Shift clustering (Equation (3.5)) to design the GMM models. One can see a slight improvement when using the second strategy. This is mainly because Mean Shift clustering takes into account spatial information of pixels in clustering time, which better defines contiguous pixels of image to belong to GMM models of foreground and background. Third row in Table 3.1 shows the overlapping results adding the temporal extension to the spatial one, considering the morphology refinement based on previous segmentation (Equation (3.7)). In this case, we obtain near 10% of performance improvement respect the previous result. Finally, last result of Table 3.1 shows the full-automatic ST-GrabCut segmentation overlapping performance introducing full temporal coherence via the segmentation refinement introduced in Equation (3.6). One can see that it achieves about 25% of performance improvement in relation with the previous best performance. Some segmentation results obtained by the GrabCut algorithm for the cVSG corpus are shown in Figure 3.4. Note that the ST-GrabCut segmentation is able to robustly segment convex regions. We have also applied the ST-GrabCut segmentation methodology on the image sequences of UBDataset. Some segmentations are shown in Figure 3.4.

Table 3.1: GrabCut and ST-GrabCut Segmentation results on cVSG corpus.

Approach	Mean overlapping
GrabCut	0.5356
Spatial extension	0.5424
Temporal extension	0.6229
ST-GrabCut	0.8747

3.5.5 Face alignment

We combine the proposed segmentation methodology with Shape and Active Appearance Models (AAM) to define three different meshes of the face, one near frontal view, and the other ones near lateral views. Temporal coherence and fitting cost are considered in conjunction with GrabCut segmentation to allow a smooth and robust face alignment in video sequences.

Once we have properly segmented the body region, the next step consists of fitting the face and the body limbs. For the case of face recovery, we base our procedure on mesh fitting using AAM, combining Active Shape Models and color and texture information [26].

AAM is generated by combining a model of shape and texture variation. First, a set of points are marked on the face of the training images that are aligned, and a statistical shape model is built [27]. Each training image is warped so the points match those of the mean shape. This is raster scanned into a texture vector, $\mathbf{g}$, which is normalized by applying a linear transformation, $\mathbf{g} \mapsto (\mathbf{g} - \mu_g \mathbf{1})/\sigma_g$, where $\mathbf{1}$ is a vector of ones, and μ_g and σ_g^2 are the mean and variance of elements of $\mathbf{g}$. After normalization, $\mathbf{g}^T\mathbf{1} = 0$ and $|\mathbf{g}| = 1$. Then, principal component analysis is applied to build a texture model. Finally, the correlations between shape and texture are learnt to generate a combined appearance model. The appearance model has a parameter $\mathbf{c}$ controlling the shape and texture according to:

$$x = \overline{x} + \mathbf{Q}_s \mathbf{c}, \tag{3.9}$$

$$g = \overline{g} + \mathbf{Q}_g \mathbf{c}, \tag{3.10}$$

where $\overline{x}$ is the mean shape, $\overline{g}$ the mean texture in a mean shaped patch, and $\mathbf{Q}_s$, $\mathbf{Q}_g$ are matrices designing the modes of variation derived from the training set. A shape $\mathbf{X}$ in the image frame can be generated by applying a suitable transformation to the points, $\mathbf{x} : \mathbf{X} = S_t(\mathbf{x})$. Typically, S_t will be a similarity transformation described by a scaling s, an in-plane rotation, θ, and a translation (t_x, t_y).

Once constructed the AAM, it is deformed on the image to detect and segment the face appearance as follows. During matching, we sample the pixels in the region of interest $\mathbf{g}_{im} = T_u(\mathbf{g}) = (u_1 + 1)\mathbf{g}_{im} + u_2\mathbf{1}$, where $\mathbf{u}$ is the vector of transformation parameters, and project into the texture model frame, $\mathbf{g}_s = T_u^{-1}(\mathbf{g}_{im})$. The current model texture is given by $\mathbf{g}_m = \overline{g} + \mathbf{Q}_g \mathbf{c}$, and the difference between model and image (measured in the normalized texture frame) is as follows:

$$\mathbf{r}(\mathbf{p}) = \mathbf{g}_s - \mathbf{g}_m. \tag{3.11}$$

Given the error $\mathbb{E} = |\mathbf{r}|^2$, we compute the predicted displacements $\delta\mathbf{p} = -\mathbf{R}\mathbf{r}(\mathbf{p})$, where $\mathbf{R} = \left(\frac{\partial \mathbf{r}^T}{\partial \mathbf{p}} \frac{\partial \mathbf{r}}{\partial \mathbf{p}}\right)^{-1} \frac{\partial \mathbf{r}^T}{\partial \mathbf{p}}$. The model parameters are updated $\mathbf{p} \mapsto \mathbf{p} + k\delta\mathbf{p}$, where initially $k = 1$. The new points $\mathbf{X}'$ and model frame texture $\mathbf{g}'_m$ are estimated, and the image is sampled at the new points to obtain $\mathbf{g}'_{mi}$ and the new error vector $\mathbf{r}' = T_{u'}^{-1}(g'_{im}) - g'_m$. A final condition guides the end of each iteration: if $|\mathbf{r}'|^2 < \mathbb{E}$, then we accept the new estimate, otherwise, we set $k = 0.5$, $k = 0.25$, and so on. The procedure is repeated until no improvement is made to the error.

With the purpose of discretizing the head pose between frontal face and profile face, we create three AAM models corresponding to the frontal, right, and left view. Aligning every mesh of the model, we obtain the mean of the model. Finally, the class of a given aligned face is determined by its proximity to the closest mean model.

Taking into account the discontinuity that appears when a face moves from frontal to profile view, we use three different AAM corresponding to three meshes of 21 points: frontal view $\Im_F$, right lateral view $\Im_R$, and left lateral view $\Im_L$. In order to include temporal and spatial coherence, meshes at frame I^{t+1} are initialized by the fitted mesh points at frame I^t. Additionally, we include a temporal change-mesh control procedure, as follows:

$$\Im^{t+1} = \min_{\Im^{t+1}}\{\mathbb{E}_{\Im_F}, \mathbb{E}_{\Im_R}, \mathbb{E}_{\Im_L}\}, \Im^{t+1} \in \nu(\Im^t), \tag{3.12}$$

where $\nu(\Im^t)$ corresponds to the meshes contiguous to the mesh $\Im^t$ fitted at time t (including the same mesh), and $\mathbb{E}_{\Im_i}$ is the fitting error cost of mesh $\Im_i$. This constraint avoids false jumps and imposes smoothness in the temporal face behavior (e.g., a jump from right to left profile view is not allowed).

Finally, in order to obtain more accurate pose estimation, after fitting the mesh, we take advantage of its variability to differentiate among a set of head poses. Analyzing the

spatial configuration of the 21 landmarks that composes a mesh, we create a new training set divided in five classes. We define five different head poses as follows: right, middle-right, frontal, middle-left, and left. In the training process, every mesh has been aligned, and PCA is applied to save the 20 most representative eigenvectors. Then, a new image is projected to that new space and classified to one of the five different head poses according to a 3-Nearest Neighbor rule.

Figure 3.3 shows examples of the AAM model fitting and pose estimation in images (obtained from [48]) for the five different head poses.

Figure 3.3: From left to right: left, middle-left, frontal, middle-right and right mesh fitting.

We performed the overlapping analysis of meshes in both un-segmented and segmented image sequence of the public cVSG corpus, in order to measure the robustness of the spatio-temporal AAM mesh fitting methodology. Overlapping results are shown in Table 3.2. One can see that the mesh fitting works fine in unsegmented images, obtaining a final mean overlapping of 89.60%. In this test, we apply HaarCascade face detection implemented and trained by the Open-source Computer Vision library (OpenCV). The face detection method implemented in OpenCV by Rainer Lienhart is very similar to the one published and patented by Paul Viola and Michael Jones, namely called Viola–Jones face detection method [88]. The classifier is trained with a few hundreds of sample views of a frontal face, that are scaled to the same size (20×20), and negative examples of the same size. However, note that by combining the temporal information of previous fitting and the ST-GrabCut segmentation, the face mesh fitting considerably improves, obtaining a final of 96.36% of overlapping performance. Some example of face alignment using the AAM meshes for different face poses of the cVSG corpus are shown in Figure 3.5.

To create three AAM models that represent frontal, right and left views, we have created a training set composed by $1,000$ images for each view. The images have been extracted from the public database [48]. To build three models we manually put 21 landmarks over 500 images for each view. The landmarks of the remaining 500 images which cover one view, have been placed by a semi-automatic process, applying AAM with the learnt set and manually correcting them. Finally, we align every resulting mesh and we obtain the mean for each model. As the head pose classifier, to classify the spatial mesh configuration in 5 head poses, we have manually labeled the class of the obtained meshes applying the closest AAM model. Every spatial mesh configuration is represented by the 20 most representative eigenvectors. The training set is formed by $5,000$ images from the public database [48]. Finally, we have tested the classification of the five face poses on the cVSG corpus, obtaining the percentage of frames of the subject at each pose. The obtained percentages are shown in Table 3.3.

Figure 3.4: Segmentation examples of (**a**) UBDataset sequence 1, (**b**) UBDataset sequence 2 and (**c**) cVSG sequence.

Figure 3.5: Samples of the segmented cVSG corpus image sequences fitting the different AAM meshes.

Table 3.2: AAM mesh fitting on original images and segmented images of the cVSG corpus.

Approach	Mean overlapping
Mesh fitting without segmentation	0.8960
ST-Grabcut & Temporal mesh fitting	0.9636

Table 3.3: Face pose percentages on the cVSG corpus.

Face view	System classification	Real classification
Left view	0.1300	0.1211
Near Left view	0.1470	0.1347
Frontal view	0.2940	0.3037
Near Right view	0.1650	0.1813
Right view	0.2340	0.2590

3.5.6 Human pose estimation

We rely on the limb detection and CRF-based methodologies from [71] for recovering the body pose of the segmented people. Considering the refined segmented body region obtained using the proposed ST-GrabCut algorithm, we construct a pictorial structure model [38]. We use the method of Ramanan [71], which captures the appearance and spatial configuration of body parts. The human body is modeled by means of a tree-structured CRF connecting the different body parts. Thus, parts $l_i = (x, y, o)$, are oriented patches of fixed size, and their position is parameterized by location (x, y) and orientation o. The posterior of a configuration of parts $L = \{l_i \mid i \in \{1, ..., P\}\}$ given a frame I^t is:

$$P(L|I^t) \propto \exp\left(\sum_{i=1}^{P} \Upsilon(l_i|I^t) + \sum_{(i,j)\in\mathcal{E}} \Psi(l_i, l_j)\right), \tag{3.13}$$

The pair-wise potential $\Psi(l_i, l_j)$ corresponds to a spatial prior on the relative position of parts and embeds the kinematic constraints. The unary potential $\Upsilon(l_i|I)$ corresponds to the local image evidence for a part in a particular position. Inference is performed over tree-structured conditional random field.

Since the appearance of the parts is initially unknown, a first inference uses only edge features in Υ. This delivers soft estimates of body part positions, which are used to build appearance models of the parts and background (color histograms). Inference is then repeated with Υ using both edges and appearance. This parsing technique simultaneously estimates pose and appearance of parts. For each body part, parsing delivers a posterior marginal distribution over location and orientation (x, y, o) [71].

In order to show the benefit of applying previous ST-GrabCut segmentation, we compare the overlapping performance of full pose recovery with and without human segmentation, always within the bounding box obtained from HOG person detection. Results are shown in Table 3.4. One can see that pose recovery considerably increases its performance when reducing the region of search based on ST-GrabCut segmentation. Some examples of pose estimation within the human segmentation regions for cVSG corpus and UBdataset are shown in Figure 3.6. One can see that in most of the cases body limbs are correctly detected. Only in some situations, occlusions or changes in body appearance can produce a wrong limb fitting.

Table 3.4: Pose estimation results: overlapping of body limbs based on ground truth masks.

Approach	Mean overlapping
No segmentation	0.7919
ST-Grabcut	0.8760

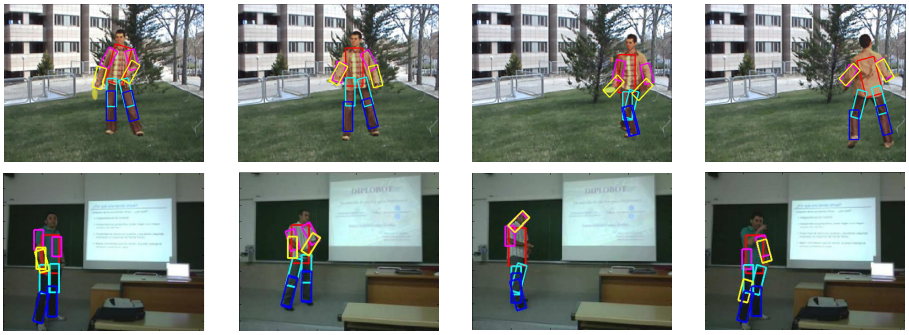


Figure 3.6: Pose recovery results in cVSG sequence.

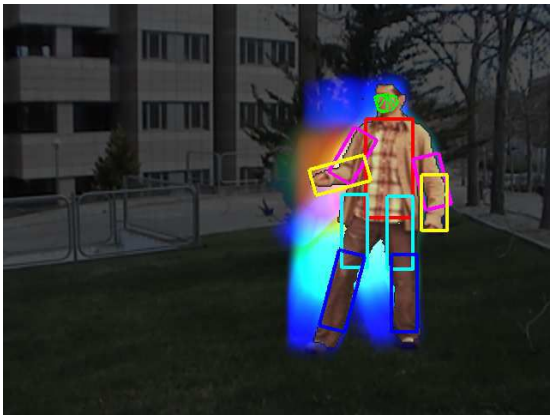


Figure 3.7: Application of the whole framework (pose and face recovery) on an image sequence.

In Figure 3.7 we show the application of the whole framework to perform temporal tracking, segmentation and full face and pose recovery. The colors correspond to the body limbs. The colors increase in intensity based on the instant of time of its detection. One can see the robust detection and temporal coherence based on the smooth displacement of face and limb detections.

Finally, we test our methodology on the presented Human Limb dataset. From the 14 total limb annotations, we grouped them into six categories: torso, upper-arms, lower-arms, upper-legs, lower-legs and head, and we tested the full pose recovery framework. In this case, we tested the body limb recovery with and without applying the ST-GrabCut segmentation, and computed three different overlapping measures: (1) %, which corresponds

Table 3.5: Overlapping percentages between body parts (intersection over union), wins (comparing the highest overlapping with and without segmentation), and matching (considering only overlapping greater than 0.6). * STGrabCut was used without taking into account temporal information.

		Torso	U.arm	U.leg	L.arm	L.leg	Head	Mean
%	**No seg.**	0.58	0.53	0.59	0.50	0.48	0.67	0.56
	STGrabCut*	0.58	0.53	0.58	0.50	0.56	0.67	**0.57**
Wins	**No seg.**	106	104	108	109	68	120	102.5
	STGrabCut*	121	123	119	118	159	107	**124.5**
Match	**No seg.**	133	127	130	121	108	155	129
	STGrabCut*	125	125	128	117	126	157	**129.66**

to the overlapping percentage defined in Equation (3.8); (2) wins, which corresponds to the number of Limb regions with higher overlapping comparing both strategies; (3) match, which corresponds to the number of limb detections with overlapping superior to 0.6. The results are shown in Table 3.5. One can see that because of the reduced region where the subjects appear, in most cases there is no significant difference applying the limb recovery procedure with or without previous segmentation. Moreover, the segmentation algorithm is not working at maximum performance due to the same reason, since very small background regions are present in the images, and thus the background color model is quite poor. Furthermore, in this dataset we are working with images, not videos, and for this reason we cannot include the temporal extension in our ST-GrabCut algotithm for this experiment. On the other hand, looking at the mean average overlapping in the last column of the table, one can see that ST-GrabCut improves for all overlapping measures the final limb overlapping. In particular, in the case of the lower-legs recovery is when a more clear improvement appears using ST-GrabCut segmentation. The part of the image corresponding to Low-legs is where more background influence exists, and thus the limb recovery has the highest confusion. However, as ST-GrabCut is able to properly segment the concave regions of the Low-legs regions, a significant improvement is obtained when applying the limb recovery methodology. Some results are illustrated on the images of Figure 3.8, where the images on the bottom correspond to the improvements obtained using the ST-GrabCut algorithm. Finally, Figure 3.9 show examples of the face fitting methodology applied on the human body limb dataset.

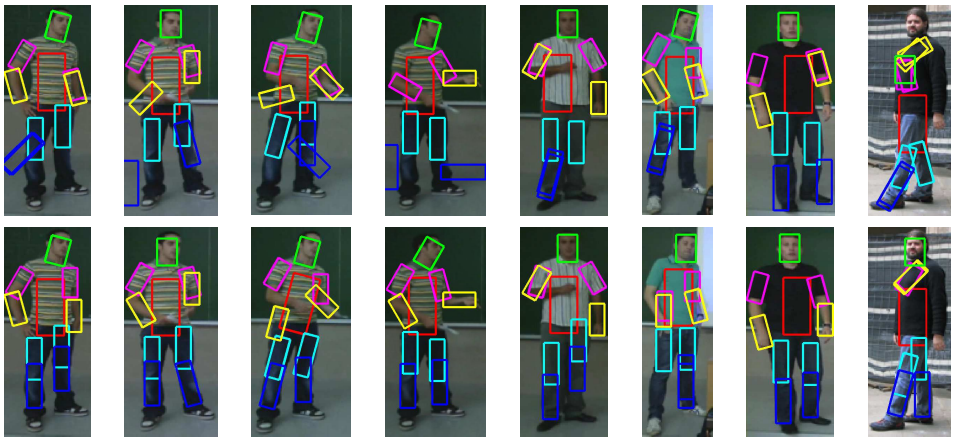


Figure 3.8: Human Limb dataset results. Upper row: body pose estimation without ST-GrabCut segmentation. Lower row: body pose estimation with ST-GrabCut segmentation.

Figure 3.9: Application of face alignment on human body limb dataset.

3.6 Discussion

We presented an evolution of the semi-automatic GrabCut algorithm for dealing with the problem of human segmentation in image sequences. The new fully-automatic ST-GrabCut algorithm uses a HOG-based person detector, face detection, and skin color model to initialize GrabCut seeds. Spatial coherence is introduced via Mean Shift clustering, and temporal coherence is considered based on the historical of color-related Gaussian Mixture Models and the trimap initialization for the next frame. The segmentation procedure is combined with Shape and Active Appearance models to perform face alignment, and a CRF model is used for human pose estimation.

This general and fully-automatic human segmentation methodology showed higher performance than simple GrabCut, in terms of segmentation accuracy. Moreover, we showed how the proposed segmentation method yields better results in face and body pose recovery, when using it as a pre-processing step in public image sequences and a novel Human Limb dataset from uncontrolled environments.

One of the limitations of the proposed ST-GrabCut method is that it depends on the initialization step algorithm, which basically depends on the person and face detectors. Initially, we wait until at least one bounding box is returned by the person detector. This is a critical point, since we will trust the first detection and start segmenting with this hypothesis. In contrast, there is no problem if a further detection is missed, since we initialize the mask with the previous detection (temporal extension). Moreover, due to its sequential application, false seed labeling can accumulate segmentation errors along the video sequence.

As future work, it would be interesting to combine human segmentation and human pose estimation methods in a same optimization framework. We have seen how a prior segmentation of people appearing in a scene can yield much better results when estimating their pose, but maybe the reverse approach can be also interesting, *i.e.* human pose estimation could help segmentation to obtain better results.

Chapter 4

Multi-label human segmentation

4.1 Introduction

In the previous chapter we presented a methodology for segmenting people from the background in video sequences, based on graph cuts optimization. In this chapter, we present a methodology for obtaining finer-detailed segmentations of the human body, classifying pixels belonging to the different body limbs, instead of a simple binary classification (foreground, background).

We present a framework for multi-label object segmentation using depth maps based on Random Forest and Graph Cuts theory, and apply it to the segmentation of human limbs. First, from a set of random depth features, Random Forest is used to infer a set of label probabilities for each data sample. This vector of probabilities is used as unary term in α-β swap Graph cuts algorithm. Moreover, depth values of spatio-temporal neighboring data points are used in the pair-wise potential. Results on a new multi-label human depth dataset show high performance in terms of segmentation overlapping of the novel methodology compared to classical approaches.

4.2 Related work

Human motion capture is an essential acquisition technology with many applications in computer vision. However, detecting humans in images or videos is a challenging problem due to the high variety of possible configurations of the scenario, such as changes in the point of view, illumination conditions, and background complexity. An extensive research on this topic reveals that there are many recent methodologies addressing this problem [24, 30, 31, 78]. Most of these works focus on the extraction and analysis of visual features. For example, the work of Bray *et al.* [21] presents a method that integrates segmentation and 3D pose estimation of a human body from multiple views. These methods have made a breakthrough in the treatment of human motion capture, achieving high performance despite the occasional similarities between the foreground and the background in the case of changes in light or viewpoint. In order to treat human pose recovery in uncontrolled scenarios, an early work used range images for object recognition or modeling [76]. This approach achieved a straightforward solution to the problem of intensity and view changes in RGB images through the representation of 3D structures. The progress and spread of this method came slowly since data acquisition devices were expensive and bulky, with cumbersome

communication interfaces when conducting experiments. In late 2010, Microsoft launched the Kinect, a cheap multisensor device based on structured light technology, capable of capturing visual depth information (RGBD technology, from Red, Green, Blue, and Depth, respectively). The device is so compact and portable that it can easily be installed in any environment to analyze scenarios where humans are present. Before Kinect, in the last decade, researchers have also used different methodologies and techniques for constructing 3D structures, such as stereoscopic images [32, 95]. However, in this case the problems of different lighting conditions and calibration still exist. Some of the research has also focused on the use of time-of-flight range cameras (TOF) for human parts detection and pose estimation [41, 73, 94], combining depth and RGB data [51].

Following the high popularity of Kinect and its depth capturing abilities, there exists a strong research interest for improving the current methods for human pose and hand gesture recognition. While this could be achieved by inter-frame feature tracking and matching against predefined gesture models, there are scenarios where a robust segmentation of the hand and arm regions are needed, e.g. for observing upper limb anomalies or distinguishing between finger configurations while performing a gesture. In this sense, depth information appears quite handy by reducing ambiguities due to illumination, colour, and texture diversity. Many researchers have obtained their first results in the field of human motion capture using this technology. In particular, Shotton et al. [79] presented one of the greatest advances in the extraction of the human body pose from depth images, an approach that also forms the core of the Kinect human recognition framework. The method is based on inferring pixel label probabilities through Random Forest (RF), using mean shift to estimate human joints, and representing the body in skeletal form. Other recent work uses the skeletal model in conjunction with computer vision techniques to detect complex poses in situations where there are many interacting actors [62].

Currently, there exists a steady stream of updates and tools that provide robustness and applicability to the device. In December 2010, OpenNI [6] and PrimeSense [7] released their own Kinect open source drivers and motion tracking middleware (called NITE [4]) for PCs running Windows (7, Vista, and XP), Ubuntu and MacOSX. FAAST (Flexible Action and Articulated Skeleton Toolkit) is a middleware developed at the University of Southern California (USC) Institute for Creative Technologies that aims to facilitate the integration of full-body control within virtual reality applications and video games when using OpenNI-compliant depth sensors and drivers [2, 82]. In June 2011, Microsoft released a non-commercial Kinect Software Development Kit (SDK) for Windows that includes Windows 7-compatible PC drivers for the Kinect device [3]. Microsoft's SDK allows developers to build Kinect enabled applications in Microsoft Visual Studio 2010 using C++, C# or Visual Basic. There is also a third set of Kinect drivers for Windows, Mac and Linux PCs by the OpenKinect (libFree- Nect) open source project [5]. Code Laboratories CL NUI Platform offers a signed driver and SDK for multiple Kinect devices on Windows XP, Vista, and 7 [1]. Finally, last advances in the field have shown outperforming results in 3D human pose estimation from RGB images [50].

4.3 Method overview

We present a framework for multi-label object segmentation using depth maps based on RF and Graph cuts theory (GC) and apply it to the segmentation of human limbs. As we have seen in previous chapters, the use of GC theory has been applied to the problem of image segmentation, obtaining successful results [17, 47, 49]. RF is used to infer a set of probabilities for each data sample, each one indicating the probability of a pixel to belong

to a particular label. Then, this vector of probabilities is used as unary term in the $\alpha - \beta$ swap GC algorithm. Moreover, depth of neighbor data points in space and time are used as pair-wise potentials. As a result, we obtain an accurate segmentation of depth images based on the defined energy terms. Moreover, as long as we have a priori likelihoods representing target appearance, the presented method is generic enough to be applicable in any other object segmentation scenario. Our method is evaluated on a 3-D data set designed in our lab, obtaining higher segmentation accuracy compared to standard segmentation approaches.

The depth-image based approach suggested in [79] interprets the complex pose estimation task as a classification problem by evaluating the likelihood of each depth pixel to be assigned a body part label, using the corresponding Probability Distribution Functions (PDF). The pose recognition phase is addressed by re-projecting the pixel classification results and inferring the 3D positions of several skeletal joints using the RF and mean-shift algorithms. The work of [79] shows a number of achievements and improvements over previous work, most notably the growing of a randomized decision forest classifier based on decision trees applied on simple and computationally efficient depth features.

Our goal is to extend the work of [79] and combine it with a general segmentation method based on GC optimization optimization to define a globally optimum segmentation of objects in depth images. As a case study, we segment pixels belonging to the following seven body parts[1]: LU/LW/RU/RW for arms, (from Left, Right, Upper and loWer, respectively), LH/RH for hands, and the torso. The pipeline of the segmentation framework is illustrated in Fig. 4.1.

4.4 Random Forests for body part recognition

Considering a priori segmented human body from the background in a training set of depth images I, the procedure for growing a randomized decision tree λ is formulated over the same definition of a depth comparison feature as defined in [79]:

$$\psi_\theta(z_i) = z_{i+\frac{u}{z_{i+u}}} - z_{i+\frac{v}{z_{i+v}}}, \tag{4.1}$$

where z_i is the depth at pixel i in image I, and $\theta = (u, v)$, is a pair of offsets u and v, the normalization of which ensures depth invariance. Thus, each θ determines two new pixels relative to z_i, the depth difference of which accounts for the value of $\psi_\theta(z_i)$. Each tree consists of split and leaf nodes (the root is also a split node), as depicted in the upper part of Fig. 4.1. The training procedure of a given tree λ over a unique set of ground truth images (avoiding to share images among different trees), runs through the following steps:

1. Define a set Φ of node splitting criteria $\phi = (\theta, \tau)$, through the random selection of $\theta = (u, v)$, and $\tau \in \mathbb{R}$ (a set of splitting thresholds for each θ), with both θ and τ lying within some predefined range limits. After training, each split node will be assigned with its optimal ϕ value from Φ.

2. Define a set Z of randomly selected pixels z_i over the entire set of training images for the tree, where the number of pixels per image is fixed. Estimate the PDF of Z over the whole set of labels $\mathcal{L}$ (in our case $|\mathcal{L}| = 7$):

$$P_Z(c) = \frac{h_Z(c)}{|Z|}, c \in \mathcal{L}, \tag{4.2}$$

[1] Note that the method can be applied to segment any number of labels of any object contained in a depth image.

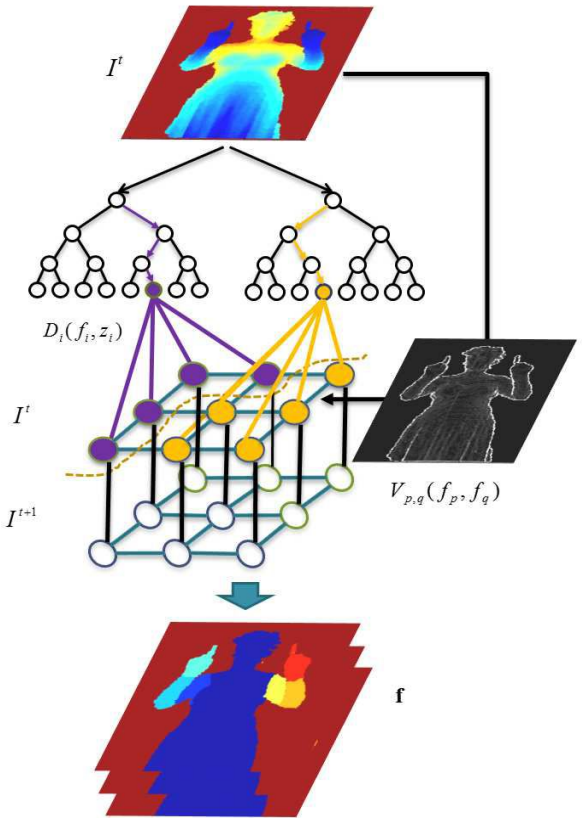


Figure 4.1: Pipeline of the presented method, including the input depth information, Random Forest, Graph-cuts, and the final segmentation result.

where $h_Z(c)$ is the histogram of the examples from Z associated with the label $c \in \mathcal{L}$. Each example from Z enters the root node, thus ensuring optimal training of the tree λ.

3. At the currently being processed node (starting from the root), split the (sub)set Z, entering this node into two subsets Z_L and Z_R obeying Eq. (4.1):

$$\begin{array}{rcl} Z_L(\phi) & = & \{i \mid \psi_\theta(z_i) < \tau\}, \phi = (\theta, \tau), \\ Z_R(\phi) & = & Z \setminus Z_L, \end{array} \tag{4.3}$$

and estimate the PDF of Z_L, $P_{Z_L}(c)$, as defined in Eq. (4.2). Compute the PDF of Z_R, which may be speeded up by the following formulae:

$$P_{Z_R}(c) = \frac{|Z|}{|Z_R(\phi)|} P_Z(c) - \frac{|Z_L(\phi)|}{|Z_R(\phi)|} P_{Z_L}(c), \; c \in \mathcal{L}. \tag{4.4}$$

4. Estimate the best splitting criterion ϕ^* for the current node, so that the information gain $G_Z(\phi^*)$ of partitioning set Z entering the node into left and right subsets to be

maximum:

$$G_Z(\phi) = H(Z) - \frac{|Z_L(\phi)|}{|Z|} H(Z_L(\phi)) - \frac{|Z_R(\phi)|}{|Z|} H(Z_R(\phi)),\ \phi = (\theta, \tau) \in \Phi, \quad (4.5)$$

where $H(Z) = -\sum_{c \in \mathcal{L}} P_Z(c) \ln(P_Z(c))$ represents Shannon's entropy for the input (sub)set Z and its splits (Z_L and Z_R) over the set of labels $\mathcal{L}$. It is more or less obvious that $G_Z(\phi) > 0$, $\phi \in \Phi$, but it is difficult to make a more analytical statement for the behaviour of $G_Z(\phi)$. That is why we also use the full search approach to evaluate ϕ^*:

$$\phi^* = \arg\max_{\phi \in \Phi} G_Z(\phi). \quad (4.6)$$

5. Recursively repeat step 3 and 4 over $Z_L(\phi^*)$ and $Z_R(\phi^*)$ for the left and right node children respectively until some preset stop conditions are met: the tree reaches maximum depth; the information gain or the number of pixels in the node falls below a minimum. The node where the stop condition occurred is treated as a leaf node, where, instead of ϕ^*, the respective PDF for the subset Z reaching the node is stored (see Eq. (4.2)).

Once trained, such a randomized tree serves as a pixel-wise classifier for a test depth image. Each image pixel z_i is tested through the tree, starting from the root and ending at a leaf node, taking a path that only depends on the inequality $\psi_\theta(z_i) < \tau$, using the splitting criterion $\phi = (\theta, \tau)$ stored at the tree nodes. Finally, the pixel is tested against the PDF kept at the reached leaf node. Because of the random factor when growing the tree, different trees have different predictions for the pixels of the same image. It cannot be stated that one tree is a better single classifier than another one since each tree is fitted to its training set. But an ensemble of trees, which form a random forest Λ, is expected to increase the predictive power of the classifier. Therefore, the inferred pixel probability distribution within the forest is estimated by averaging the PDFs over all trees in the forest as follows:

$$P(c \mid z_i) = \frac{1}{|\Lambda|} \sum_{\lambda \in \Lambda} P_\lambda(c \mid z_i), c \in \mathcal{L}, \quad (4.7)$$

where $P_\lambda(c \mid z_i)$ is the PDF stored at the leaf, reached by the pixel z_i and traced through the tree $\lambda \in \Lambda$. Assuming that trees in the forest Λ are fairly balanced, the time complexity of classifying an image is $\mathcal{O}(|\Lambda| \cdot N \cdot \bar{d}_{max})$, where N is the total number of pixels from the image I and $\bar{d}_{max}$ is the averaged maximum depth level over the trees of Λ.

The randomized tree growing process suggested by Shotton et al. [79] involves two levels of randomness: in choosing the training images and in the random definition of the node splitting criteria. This ensures minimum correlation among the trees in the forest. Unlike Breiman's classic Random Forest algorithm [81], which chooses the best split candidate (criterion) among a small subset of all possible candidates, the presented split candidate selection procedure greedily explores all possible choices in order to guarantee the most efficient split at the current node. The after-effects are two: the most informative features are filtered down and pushed onto the tree; similar pixels have better chances of falling within the same descendant nodes. The estimated time complexity of building a randomized decision tree under the above conditions is $\mathcal{O}(|\Phi| \cdot |Z| \cdot d_{max})$.

We apply the RF methodology of [79], as described above, in the following two use cases: for rough detection of the main body parts, and for detailed segmentation of the fingers of the hands (to eventually be applied for sign-cued language recognition problems).

4.5 Multi-limb human segmentation

In this section, we formulate an energy minimization problem for multi-limb segmentation of the human body, based on the GC optimization framework. More specifically, we rely on the output from the previous RF approach applied on depth images, and optimize it introducing spatio-temporal consistency via the GC graph topology.

Given the set of frames $\mathcal{I} = \{I^1, ..., I^t, ...; I^M\}$, of a video sequence, and $\mathbf{z} = (z_1, ..., z_i, ..., z_{N \cdot M})$ the array of pixels of $\mathcal{I}$, $\mathcal{N}$ as the set of unordered pairs $\{i, j\}$ of neighboring pixels in space and time, under a defined neighborhood system –typically 6- or 26-connectivity–, and $\mathbf{f} = (f_1, ..., f_i, ..., f_{N \cdot M})$ a vector whose components f_i specify the labels assigned to pixels in video sequence $\mathcal{I}$. We define the following energy function E whose minimum value corresponds to the optimal solution to our problem:

$$E(\mathbf{f}, \mathbf{z}) = \sum_{i=1}^{N \cdot M} D_i(f_i, z_i) + \sum_{\{i,j\} \in \mathcal{N}} V_{i,j}(f_i, f_j, z_i, z_j). \tag{4.8}$$

The unary potential is defined as the negative logarithm of the likelihood of each pixel to be classified as a certain human limb, computed by the RF approach (Equation 4.7):

$$D_i(f_i, z_i) = -\log(P(f_i|z_i)). \tag{4.9}$$

The pair-wise potential encodes contextual information by introducing penalties to each pair of neighboring pixels as follows:

$$V_{i,j}(f_i, f_j, z_i, z_j) = \Gamma \cdot \Omega(f_i, f_j) \cdot \frac{1}{\mathsf{dist}(i,j)} e^{-\beta \|z_i - z_j\|^2}, \tag{4.10}$$

where $\beta = \left(2\langle (z_i - z_j)^2 \rangle\right)^{-1}$, $\mathsf{dist}(i, j)$ computes the Euclidean distance between the cartesian coordinates of pixels z_i and z_j and $\Gamma \in \mathbb{R}^+$ is a weight that specifies the relative importance of the boundary term against the unary term. Finally, the function $\Omega(f_i, f_j)$ is a generalization of the Potts model to the multi-label case, introducing prior costs between each possible pair of neighboring labels.

We defined two different $\Omega(f_i, f_j)$ functions in order to introduce some prior costs between different labels. On one hand, we considered the trivial case where all different labels have the same cost, generalizing the Potts model to the multi-label case:

$$\Omega_1(f_i, f_j) = \begin{cases} 0 & \text{for} \quad f_i = f_j \\ 1 & \text{for} \quad f_i \neq f_j. \end{cases} \tag{4.11}$$

On the other hand, we introduced some spatial coherence between the different labels, taking into account the kinematic constraints of the human body limbs:

$$\Omega_2(f_i, f_j) = \begin{cases} 0 & \text{for} & f_i = f_j \\ 10 & \text{for} & f_i = \text{LU}, f_j = \text{RU} \\ & & f_i = \text{LH}, f_j = \text{RH} \\ 5 & \text{for} & f_i = \text{LW}, f_j = \text{RH} \\ & & f_i = \text{RW}, f_j = \text{LH} \\ 1 & \text{otherwise} & . \end{cases} \tag{4.12}$$

With this definition of the inter-label costs, we complicate the optimization algorithm to find a segmentation in which there exists a frontier between the right and left upper-arms, right and left hands, or in the lower measure, between left hand and right lower-arm, and

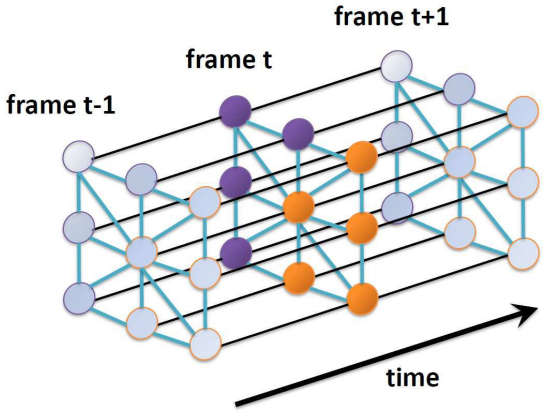


Figure 4.2: Graph topology introducing temporal coherence.

vice-versa. Therefore, we are assuming that poses in which the two hands are touching are not probable[2].

In order to build the graph for the energy function optimization, we adopt a 3-D grid topology that introduces spatial and temporal connections between pixels of the video sequence. We define a 10-connectivity pattern for the pixel neighborhood system (8 spatial and 2 temporal neighbors), shown in Fig. 4.2. From a practical point of view, and considering that computer memory resources are limited, we adopt a sliding-window approach. More specifically, we define a fixed size volume window of M frames *i.e.* M frames will be simultaneously segmented in a single execution of the optimization algorithm.

The sliding-window approach starts segmenting the first M frames, and covers all the video sequence volume, with a one-frame stride. This means that all the frames except the first and the last one are segmented at least twice and M times at most. In order to select the final hypothesis for each frame, we use the energy value resulting from the minimization algorithm at each execution. Therefore, the execution with the lowest energy value is the one we trust as the best hypothesis.

As we reviewed in Chapter 2, multi-label graph cuts optimization can be performed via two different algorithms; depending on the choice of the pair-wise potential V, we could use α-expansion (in case V is a metric), or α-β swap (if V is a semi-metric). In our case, we have that $\Omega_1(f_i, f_j)$ satisfies the conditions to be a metric, so α-expansion can be used in order to find the solution. In contrast, $\Omega_2(f_i, f_j)$ does not meet all the necessary conditions to become a metric (it is a semi-metric instead), so α-β swap must be used in this case.

4.6 Experiments

We first describe the considered data for the experiments, the different methods, parameters and validation protocol of the evaluation. Next we present results for pixel classification

[2]This label coherence cost should be estimated for each particular problem domain. In our particular dataset of poses, the values of 1, 5, and 10 were experimentally computed.

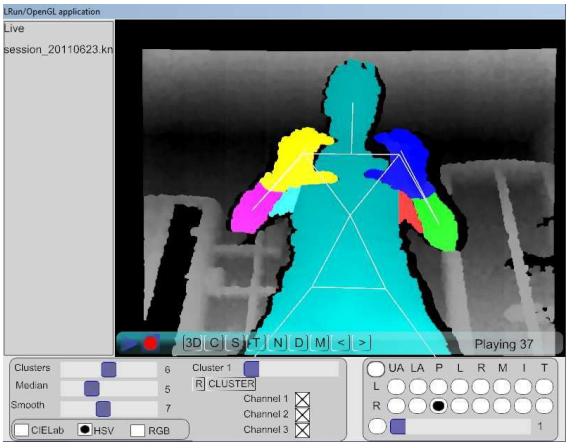


Figure 4.3: Interface for semi-automatic ground-truth generation.

using Random Forest, and the results obtained when applying multi-label segmentation.

4.6.1 Data

For the purposes of gathering ground truth data, we defined a new dataset of several video sequences where the actors are performing different gestures with their hands in front of a Kinect camera – only the upper body is considered. See Fig. 4.4 for some pose samples. Each frame is composed by one 24 bit RGB image of size 640x480 pixels, one 12 bit depth buffer of the same dimension, and a skeletal graph describing relevant joints of the upper human body. In order to label every pixel we created a special editing tool to facilitate labelling in a semi-supervised manner. Each frame is accompanied with label buffer of the same dimension as the captured images. The label buffer is automatically initialized through a rough label estimation algorithm. The pixels bounded by the cylinders between the enclosing joints of the shoulder to elbow are labelled as upper arm (LU/RU). By analogy the pixels inside the cylinder between the elbow and the joint of the hand are labelled as lower arm (LW,RW). The palm is labelled by the pixels bounded by a sphere centered in the joint of the hand (LH,RH). The RGB, depth, and skeletal data are directly obtained via the OpenNI library [6]. Finally each frame is manually edited to correct the roughly estimated labels by the initialization algorithm. The whole ground truth used in our experiments is created from capturing 2 actors in 3 sessions gathering 500 frames in total (15 fps). It should be noted that after the manual editing there still exist around 1% of false positive labels due to editor mistakes. An example of the developed interface for semi-automatic ground-truth generation is shown in Figure 4.3. See Appendix A for more information on how to obtain the dataset.

We also made an extra experiment for finger segmentation defining 6 labels per hand - one label for each finger and one for the palm. For gathering ground truth data from the fingers, we applied another initialization algorithm using coloured gloves, with each finger being painted with a different colour. Finally manual editing is still necessary due to the high level of false positive errors. 63 frames are generated and used in the experiment.

4.6.2 Methods and validation

In the first place, we analyze the results obtained directly using the probabilities returned by the RF approach. The RF algorithm used for the experiments computation has been implemented following the description of Shotton et al. [79]. In the same way, inspired by the reported test parameters and accuracy results in [79], our experiments rest on the following setup: we perform a 5-fold cross-validation over the available 500 frames by training random forest of $\Lambda = 3$ trees, therefore 130 unique training images per tree, with $1,000$ uniformly distributed pixels per image. We limit the maximum depth level d_{max} for all trees to 20, and use 100 candidate offset pairs θ, and 20 thresholds τ per θ to build the splitting criteria Φ. The remaining 100 images form the test set. Carrying a randomized test trial, we analyze the effect of the choice of test parameters on the classification accuracy and compare the results with another set of features: a mixture of the original depth features $\psi_\theta(z_i)$ from Eq. (4.1) and new features $\hat{\psi}_\theta(z_i)$ based on the depth gradient:

$$\hat{\psi}_\theta\left(z_i\right) = \measuredangle\left(\nabla\left(z_{i+\frac{u}{z_{i+u}}}\right), \nabla\left(z_{i+\frac{u}{z_{i+v}}}\right)\right), \tag{4.13}$$

where $\nabla\left(z_i\right)$ is the gradient of depth at pixel z_i. In fact, the new feature $\hat{\psi}_\theta(z_i)$ represents the angle between the two gradient vectors at offsets u and v from i.

In second place, we compare our proposed segmentation approach with the state-of-the-art Random Walks (RWalks) image segmentation algorithm [44], both applied to the probability maps returned by the execution of the RF method. Since RWalks is designed to segment still images, no temporal coherence is taken into account in this approach. Furthermore, besides the probability maps, RWalks also receives some user-designed seeds, since it is semi-automatic. However, in order to perform a fair comparison between this method and our proposal, which is fully-automatic, we need to automatize the seed-selection process. For this task, we select the seeds for each label as the pixels with greatest probability value. When applying GC, the Γ parameter was set to 50 for all the performed experiments, and the size of the sliding window is set to $M = 5$.

In order to achieve a more appropriate comparison of the results, we perform an additional GC experiment consisting on removing the temporal coherence, *i.e.* segmenting each frame of the sequence independently, using a 2-D lattice graph topology with 8-connectivity. Furthermore, in this second experiment we also compare the performance of prior cost functions Ω_1 and Ω_2, as well as the use of different pixel information for the computation of the pair-wise potential. In addition to depth information, we also test including RGB information to the pair-wise potential by normalizing depth information in the range [0...255] and concatenating it with RGB values, resulting in a 4-dimensional RGBD vector for each pixel.

Finally, we also apply the Friedman test [33] in order to look for statistical significance of the performed experiments.

4.6.3 Random forest pixel-wise classification results

Table 4.1 shows the estimated average classification accuracy for each of the considered labels. The most likely label predicted for a pixel is chosen to be the one that corresponds to the maximum of the inferred RF probabilities for that pixel. Without claiming exhaustiveness of our experiments, the results from Table 4.1 allow us to make the following analysis: The upper limit O_{max} for the module of u and v offsets has the greatest impact on the accuracy results at the hands regions, which have the smallest area in our body part definition. Doubling O_{max} leads to an increase in the accuracy of about 20% for the hands and about 6% for the other body parts. In other words, higher O_{max} values increase the feature diversity and the global ability to represent spatial detail. The number of candidate offset

Table 4.1: Average per class accuracy in % calculated over the test samples in a 5-fold cross validation. ψ_θ represents features of the depth comparison type from Eq. (4.1), while $\hat{\psi}_\theta$ - the gradient comparison feature from Eq. (4.13). O_{max} is the upper limit of the u and v offsets, and d_{max} stands for the maximal depth level of the decision trees.

Setup				Performance							
ψ_θ	$\hat{\psi}_\theta$	O_{max}	d_{max}	Torso	LU	LW	LH	RU	RW	RH	Avg.
100	0	30	20	92.90	73.29	71.42	57.75	74.25	76.26	59.38	72.18
80	0	60	20	**94.22**	79.08	76.46	74.19	81.24	83.26	79.05	81.07
60	0	60	20	94.09	78.86	75.86	73.49	79.43	82.60	78.08	80.34
100	0	60	20	94.17	79.83	77.69	**77.10**	81.04	82.65	80.17	81.81
100	0	60	15	94.06	79.81	78.69	76.59	81.18	83.10	**80.23**	**81.95**
100	0	60	10	91.83	**81.47**	**78.98**	72.30	**83.00**	**83.74**	76.85	81.17
60	20	60	20	94.04	77.73	74.93	71.97	77.62	81.22	76.64	79.17

pairs θ would not have such a tremendous impact on the accuracy as the O_{max} parameter, although a higher number of θ candidates would help in identifying the most discriminative features. A decrease of the number of candidates from 100 to 80 features drops the hands accuracy by $1-3\%$.

We also tested the impact of d_{max}, the depth level limit of the decision trees. Trimming the trees to level 15 has a very little impact, showing an improvement of 0.1% on the average accuracy. The latter may be weakly attributed to better classification at the lower arm regions. Trimming to depth level 10 shows a 4% decrease in the accuracy at the hands. Our analysis indicates that we may be witnessing slight overfitting at tree depth level of 20 due to the small amount of training images. Our final test includes comparison over combination of both features ψ_θ and $\hat{\psi}_\theta$ of Eq. (4.1) and Eq. (4.13). Since the depth data provided by Kinect is noisy, we apply a Gaussian smoothing filter before calculating the image gradients and the $\hat{\psi}_\theta$ features. We chose the gradient feature since it complements the relations of depth features with information about the orientation of local surfaces. However, in our test we did not find significant improvement in the performance results of the RF approach.

In order to show the generalization capability of the proposed approach, we carried out an extra case study, stressing on the segmentation of the finger regions. For this test we only considered a manual annotated set of 63 depth image frames without including temporal coherence. The results applying the same validation as in the previous case show the best performance for the following setup: 1 tree of depth 15, 500 pixels per image, 100 candidate offset pairs θ, 20 candidate thresholds τ, and $O_{max} = 45$. The estimated average per class accuracy was 58.5%, mostly due to the small number of training images. Fig. 4.5 displays a couple of test images comparing the ground truth and the inferred labels for the fingers and hands. Reviewing the classification results from both use cases, the body parts and finger regions, we observe that some of the errors appear due to left/right uncertainty. Nevertheless, the results are promising, showing the generalization ability of the presented approach for general multi-class labelling in depth images.

4.6.4 Multi-label segmentation results

The results we obtained when applying GC over the probabilities returned by the RF are detailed in Table 4.2. We can see how these results improved the labelling obtained by the RF approach, and also the one obtained in the frame-by-frame approach. Moreover, all the GC approaches shown in Table 4.2 outperform the results obtained using the RWalks

Table 4.2: Average per class accuracy in % obtained when applying the different GC approaches –TC: Temporally coherent, Fbf: Frame-by-Frame– , and the best results from the RF probabilities [79] and the RWalks segmentation algorithm [44], in the first and second rows, respectively.

Method	Torso	LU	LW	LH	RU	RW	RH	Avg.
RF results	94.06	**79.81**	78.69	76.59	81.18	83.10	80.23	81.95
RWalks results	**99.05**	72.17	81.04	86.98	73.27	88.48	91.68	84.67
Fbf, Depth, Ω_1	98.86	75.05	82.87	91.45	77.57	87.35	93.96	86.73
Fbf, Depth, Ω_2	98.86	75.03	83.36	**92.41**	77.54	87.67	**94.20**	87.01
Fbf, RGBD, Ω_1	99.02	72.02	81.86	90.29	76.56	86.84	92.14	85.53
Fbf, RGBD, Ω_2	99.02	72.03	81.95	91.19	76.53	87.12	92.12	85.71
TC, Depth, Ω_2	98.44	78.93	**84.38**	88.32	**82.57**	**88.85**	93.86	**87.91**

segmentation algorithm in most of the body parts. If we take a closer look at the measurements, we can see that we obtain the best results when using only depth information for the computation of the boundary potential. In our case study, adding RGB to the depth information reduces the generalization of the boundary potential. In Fig. 4.4 we can see some qualitative results of the segmentations.

Another interesting result is the influence of the prior costs given by the different $\Omega(f_i, f_j)$ functions. Clearly, when introducing spatial coherence with $\Omega_2(f_i, f_j)$, we obtain better results, especially in the segmentation of the hands, which are the parts with more confusion among all. Fig. 4.6 shows a qualitative example of both approaches.

A more detailed analysis of the results from the temporally-coherent approach reveals that the highest improvement is obtained in the case of the upper part of the limbs. In contrast, the results related to both the left and right hands are slightly worse than the frame-by-frame approach. However, hands are the most moving body parts in the video sequences, and the time lapse between one frame and the next one can be too large, inducing the introduction of some noise.

Taking a look at the qualitative results in Fig. 4.4, one can first see how the spatial coherence introduced by the basic frame-by-frame GC approach –Fig. 4.4 (d)– allows to recover more consistent regions than the ones obtained with just the RF probabilities, in such a way that each limb is represented by just one blob. Moreover, when introducing temporal coherence –Fig. 4.4 (e)–, the classification of certain labels like the ones corresponding to the arms is more accurate compared to the results obtained without temporal coherence. The RWalks algorithm –Fig. 4.4 (c)– obtains accurate segmentations when the RF probabilities have low noise, but it fails in the opposite case, though in the shown cases it seems to perform better than the frame-by-frame GC approach. Furthermore, RWalks is prone to confuse the labels between the right and left body limbs, since no label consistency is enforced.

Finally, we use the Friedman test [33] to show that the results are not affected by randomness. For this purpose, we compute the ranks of each segmentation strategy in Table 4.2 independently for each segmentation label –and also for the average. We define the computation of the ranks for a certain label as one "experiment". More specifically, the rankings are obtained estimating each relative rank r_c^s for each label $c \in \mathcal{L}$ and each segmentation strategy s, and computing the mean ranking R for each strategy as $R_s = \frac{1}{N_F} \sum_{i=1}^{N_F} r_i^s$ with $N_F = |\mathcal{L}| + 1$, being $|\mathcal{L}|$ the total number of possible labels. The Friedman statistic value is

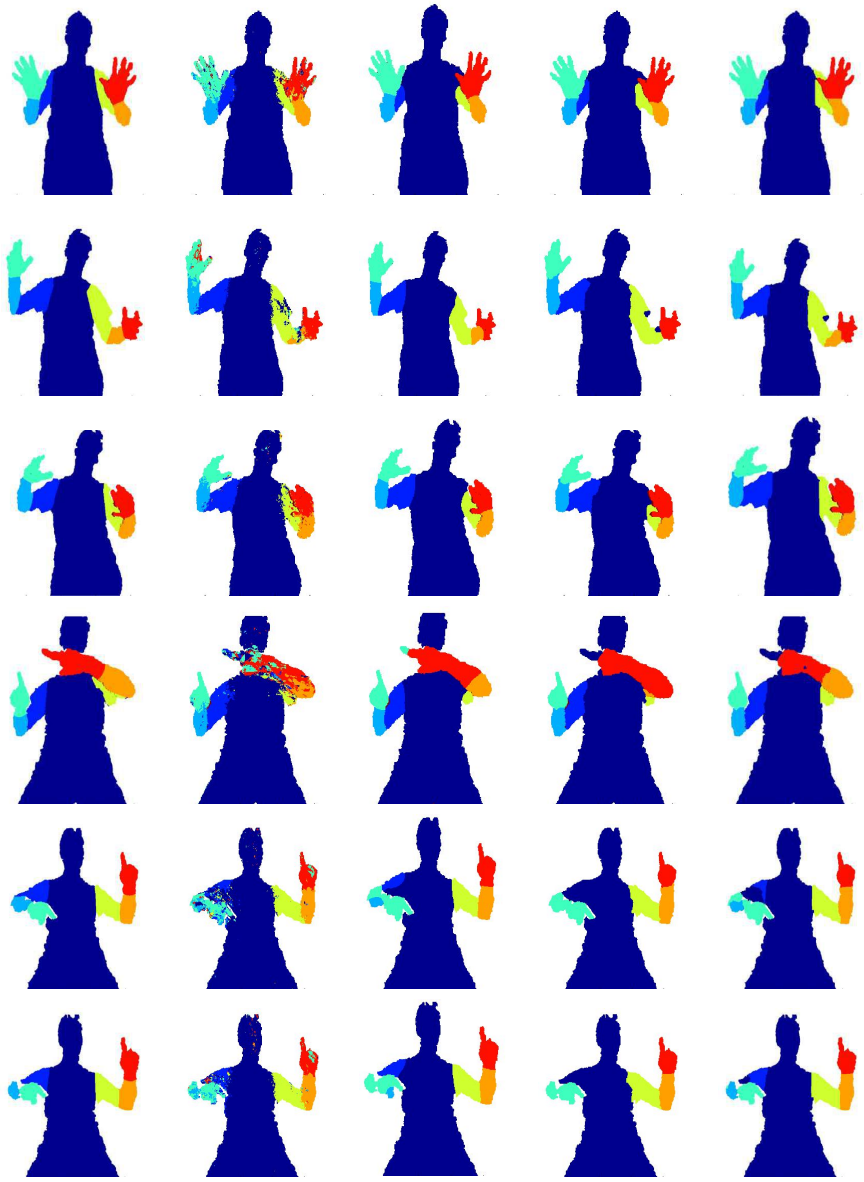


Figure 4.4: Qualitative results; Ground Truth (a), RF inferred results (b), RWalks results (c), frame-by-frame GC results (d), and Temporally-coherent GC results (e).

then computed as follows:

$$X_F^2 = \frac{12N_F}{k(k+1)}\left[\sum_s R_s^2 - \frac{k(k+1)^2}{4}\right]. \tag{4.14}$$

In our case, with $k = 7$ segmentation strategies to compare, $N_F = 8$ different experiments, and ranks $R = [5.63, 4.88, 2.5, 3, 2.75, 4.75, 4.5]$ in the row order of Table 4.2, $X_F^2 = 15.48$. Since this value is undesirable conservative, Iman and Davenport proposed a corrected statistic:

$$F_F = \frac{(N_F - 1)X_F^2}{N_F(k+1) - X_F^2}. \tag{4.15}$$

Applying this correction we obtain $F_F = 3.38$. With seven strategies and eight experiments, F_F is distributed according to the F distribution with six and 42 degrees of freedom. The critical value of $F(6, 42)$ for 0.05 is 2.23. As the value of F_F is higher than 2.23 we can reject the null hypothesis, and thus, looking at the best mean performance in Table 4.2, we can conclude that the spatio-temporal GC proposal is the best choice from the presented experiments.

In the second experiment, labelling pixels from hands –in a frame-by-frame fashion, we achieve an average per class accuracy of 70.9%, which supposes even a greater improvement than in the case of human limbs. Fig. 4.5 shows some qualitative results of the GC approach, where we can appreciate that regions are more consistent and better defined than in the case of just using RF probabilities. It is worth mentioning that for this experiment we used $\Omega_1(f_i, f_j)$ as the cost function between labels, and yet we obtained consistent results.

4.7 Discussion

We proposed a generic framework for object segmentation using depth maps based on Random Forest and Graph cuts theory in order to benefit from the use of spatial and temporal coherence, and applied it to the segmentation of human limbs. Random Forest estimated the probability of each depth sample point to belong to a set of possible object labels, while Graph-cuts was used to optimize, both spatially and temporally the RF probabilities. Results on two novel datasets showed higher performance segmenting several body parts in depth images, compared to state-of-the-art approaches.

Among the different parameters of the Random Forest method for pixel-wise classification, we saw that the upper limit O_{max} on the module of offsets u and v when computing depth comparison features ψ_θ plays an important role in the final performance of the system. When choosing the appropriate value for O_{max}, the pixel-wise classification mean accuracy increases in a 10%, yielding a especially higher 20% performance boost in the accuracy of hand region classification.

When enforcing spatio-temporal consistency on the classification by applying multi-label Graph cuts optimization on the probabilities computed by Random Forest, we showed that adding asymmetric costs between different labels via the defined $\Omega_2(f_i, f_j)$ function (Eq. 4.12) yields better results, especially in the case of the upper part of the body limbs. While not ensuring any approximation factor w.r.t the optimal solution, α-β swap yields better results when using the asymmetric label cost assignments defined by $\Omega_2(f_i, f_j)$ than the solution found by α-expansion and the multi-label generalization of the Potts model $\Omega_1(f_i, f_j)$.

As interesting future research lines, such fine-grained segmentation masks of the human body could be used to extract body pose features (especially in the case of the hands), and

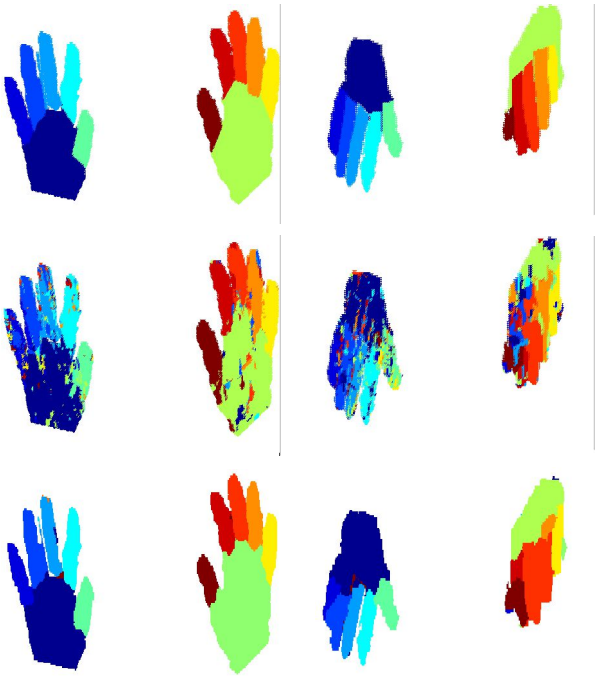


Figure 4.5: Results from RF classification in the case of hands. First row shows the ground-truth for two examples. Second row shows the RF classification results. Third row shows the final α-expansion GC segmentation results.

(a) (b)

Figure 4.6: Comparison of results without (a) and with (b) spatially-consistent labels.

apply them to recognize gestures, for human-computer interaction and smart environment applications.

Part II

Human Pose Estimation

Symbol notation in Part II

Table 4.3: Symbols and conventions for chapters 5- 6

$L = \{l_1, ...l_M\}$	Set of M parts of an articulated object, l_i
(x_i, y_i)	Part position
θ_i	Part rotation
s_i	Part scale
$\mathcal{G} =< \mathcal{V}, \mathcal{E} >$	CRF graph composed of nodes $\mathcal{V}$ and edges $\mathcal{E}$
v_i	node in $\mathcal{V}$, representing part l_i.
e_ij	edge in $\mathcal{E}$ connecting nodes i and j
E	Energy function for the PS model
E_i^u	Unary potential for node i
$E_{i,j}^u$	Pair-wise potential for edge between nodes i and j
I	Image
$\{I^n, L^n\}$	Training set, $n = 1, ..., N$, composed by N pairs of images and part configurations
$\boldsymbol{\beta} = (\beta^u, \beta^p)$	Parameters of the PS model
t_i	Type index of part l_i
T	Number of types for each part in the DPM model
$S(I, L, \boldsymbol{\beta})$	Score function of the DPM model
S_i^u	Score unary term for part l_i
$S_{i,j}^p$	Score pair-wise term for the connection between parts l_i and l_j
S^b	Score bias term
$\boldsymbol{\beta} = (\beta^u, \beta^p, \beta^b)$	Parameters of the DPM model
$\phi(I, x_i, y_i)$	Feature extraction function
$\psi(l_i, l_j)$	Vector of position differences between parts l_i and l_j
$m_i(l_j)$	Message from part i to its parent j
$score_i(l_i)$	Local score at part i
ξ_n	Slack variable
P	Number of poselets in the mid-level part representation
$\mathbb{P}$	Set of P poselet detectors
P'	Initial random number of poselet candidates
$B_{p'}$	Bounding box with position $(x_{p'}, y_{p'})$, width $wi_{p'}$, height $he_{p'}$ and scale $s_{p'}$
τ_{proc}	Threshold on the Procrustes alignment cost
$\omega_{p'}$	Detector for poselet proposal p'
(x_k^{gt}, y_k^{gt})	Ground-truth annotations for the body joint k

$(\mu^k_{p'}, \Sigma^k_{p'})$	Gaussian distribution on the spatial coordinates of keypoint k among the training samples for poselet detector proposal p'
A	Binary matrix of poselet activations in the validation set
$\mathbf{x}_{p'}$	Binary variable of the Integer Program for poselet selection
$\mathcal{P}$	Set of poselet detections
$C_i^{\mathcal{P}}$	Set of contextual feature vectors between body part l_i and $\mathcal{P}$
G	Number of detections taken from each poselet $p \in \mathbb{P}$
$R(C)$	Rescoring function given the set of contextual feature vectors C
Q_θ	Weak set classifier
Θ	Number of weak set classifiers
q_θ	Weak item classifiers (decision trees)
α_θ	Weight of weak set classifier Q_θ
k_c	Relevance of item $c \in C$
F	Number of leaves in decision trees q_θ
$U_1, ..., U_F$	F Partitions of the feature space given by leaves in q_θ
α^f	Weights of leaf nodes in q_θ
λ	Regularization parameter
$\mathcal{L}(\cdot)$	Loss function
H^f	Total sum of relevance of items falling in partition U^f
y_{B_i}	Label of bounding box B_i from a detection of part l_i
$O(B_i, B_j)$	Overlapping between bounding boxes B_i and B_j
$\hat{\beta}_i^{t_i}$	Weight of the rescoring function in the DPM score function
$E^{u,boost}$	Local appearance unary potential
$E^{u,poselets}$	Poselet evidence unary potential
β_w^u	Weight of the rescoring function in the PS energy function

Chapter 5

Detecting people: Part-based object detection

5.1 Introduction

Detecting rigid objects in images has been commonly formulated as a binary classification problem, where input subregions of a given image are classified as positive or negative, whether a specific object is present in that subregion or not, respectively. Some well-known examples of such methodologies are face detection [88], or pedestrian detection [30] (see Fig. 5.1). Given that the appearance variability of these objects is relatively low, the classifier learns subregions of a specific aspect ratio. In contrast, articulated objects have a much higher variability of their shape and appearance, hence more sophisticated models are needed in order to capture these object deformations.

In particular, part-based models have proven high performance rates for object detection. The main idea behind part-based models is the decomposition of an object into a set of parts

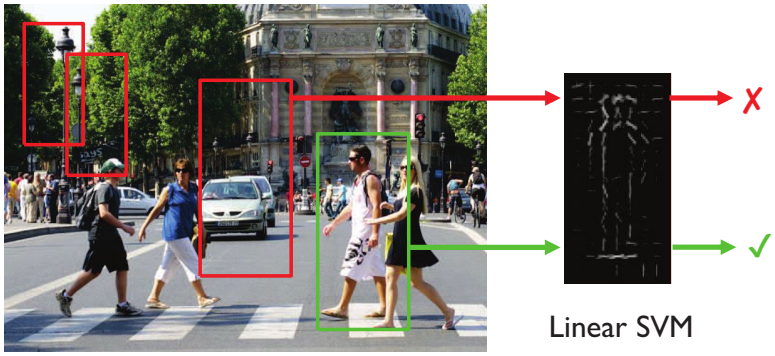


Figure 5.1: Pedestrian detection as a classic sliding-window approach for object detection. HOG features extracted from candidate bounding boxes in the image are tested against a Linear SVM trained on images of people, which predicts a positive (green) or negative (red) answer for each candidate window.

connected to each other following a class-specific topology. The motivation for this is that, while the global appearance of an articulated object may vary considerably when suffering different deformations, the appearance of some of its parts may not be significantly altered. Therefore, detecting an articulated object is reduced to detecting its parts (considered as new object classes) and applying a known relational model that fosters spatially-consistent part configurations.

In this chapter we briefly introduce two methodologies for part-based object detection, successfully used to tackle the problem of human pose estimation: pictorial structures (PS) and deformable part models (DPM).

5.2 Pictorial structures

As mentioned in the introduction, articulated object detection is usually tackled by learning a model formed by different connected parts. Pictorial structures proposed by Felzenswalb and Huttenlocher [38] follow this philosophy by modeling an object as a collection of parts arranged in a deformable configuration. More specifically, parts are defined as local appearance templates, while spatial connections between parts can be considered as flexible "springs" (see Fig, 5.2(a)).

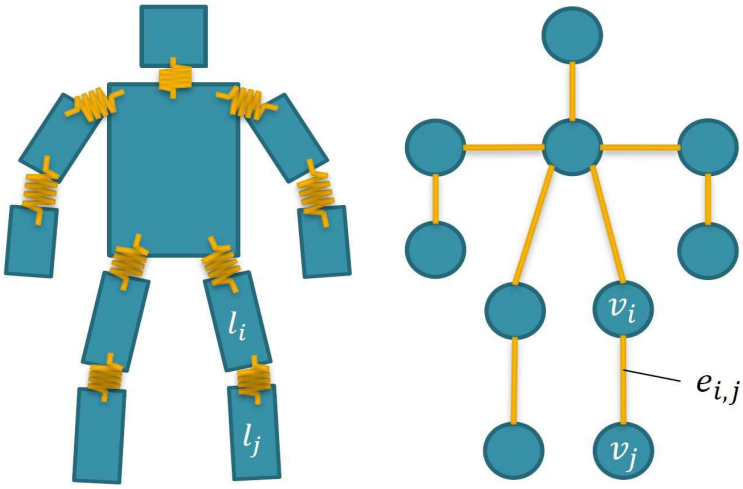


Figure 5.2: (a) Sample pictorial structure for human pose estimation; blue rectangles depict the different parts of the model (corresponding to parts of the human body) and yellow springs show the flexible connections between parts. (b) The corresponding CRF for the pictorial structure model in (a); blue nodes represent the parts of the model and yellow edges codify the spring-like connections.

Let $L = \{l_1, ..., l_i, ...l_M\}$ be the set of parts of an object, where each part $l_i = (x_i, y_i, r_i, s_i)$ is parametrized by its position x_i, y_i, absolute rotation r_i and scale s_i. Given a graph $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$, a pictorial structure model is then formulated as a Conditional Random Field (CRF) in which parts l_i are indexed by nodes $v_i \in \mathcal{V}$ in the graph $\mathcal{G}$, and the set of edges $e_{i,j} \in \mathcal{E}$ represent the "spring"-like connectivity pattern between nodes (see Fig. 5.2(b)).

Similarly to graph cuts optimization reviewed in chapter 2, an energy function associated to such CRF can then be defined as follows:

$$E(I, L, \boldsymbol{\beta}) = \sum_{i=1}^{M} E_i^u(I, l_i, \beta_i^u) + \sum_{(i,j)\in\mathcal{E}} E_{i,j}^p(l_i, l_j, \beta_{i,j}^p), \tag{5.1}$$

where I is an image, E^u and E^p are the unary and pair-wise potentials and β^u, β^p their respective sets of model parameters.

In this case, the unary potential E_i^u can be considered as a mismatch function for placing part i in the image under the configuration specified by l_i, based on appearance cues. Hence, β_i^u parameters could be thought of as a classifier for detecting part i and recover its configuration l_i, *e.g.* a Linear SVM classifier on HOG features, like in the pedestrian detection example. On the other hand, the pair-wise potential $E_{i,j}^p$ can be defined as a function measuring the degree of deformation of the "spring" connecting parts i and j under configurations l_i and l_j, respectively. In this case, parameters $\beta_{i,j}^p$ are the responsible for learning the flexibility of the "springs" connecting different parts. Then, given an image at test time, the goal is to find the part configutation L that minimizes the defined energy function:

$$L^* = \arg\min_L E(I, L, \boldsymbol{\beta}). \tag{5.2}$$

5.2.1 Inference

In order to efficiently find the best configuration L^* of the pictorial structure model, the graph $\mathcal{G}$ defining the topology of the corresponding CRF must not contain any cycles, hence the topology must be a tree. Then, belief propagation algorithms like sum-product message-passing can be used for efficient exact inference and find L^*.

Given this restriction on the topology for $\mathcal{G}$, the majority of works that use pictorial structures, define their models following a tree structure. More specifically, works on human pose estimation like [10, 71] use models like the one shown in Fig. 5.2.

Nevertheless, some other works based on pictorial structures introduce cycles in $\mathcal{G}$, and use loopy belief propagation for inference. However, this algorithm only allows for approximate inference, so the optimum solution L^* can not be found efficiently in this case.

5.2.2 Learning

Learning a pictorial structure can be then understood as estimating the parameters $\boldsymbol{\beta} = (\beta^u, \beta^p)$ of the model. While there are no strict guidelines for parameter estimation, Felzenswalb and Huttenlocher [38] proposed Maximum Likelihood Estimation (MLE) for that matter:

$$\boldsymbol{\beta}^* = \arg\max_{\boldsymbol{\beta}} \prod_{k=1}^{N} P(I^k \mid L^k, \boldsymbol{\beta}) \prod_{k=1}^{N} P(L^k \mid \boldsymbol{\beta}), \tag{5.3}$$

given a training set $\{I^n, L^n\}, n = 1, ..., N$ composed of N images and their corresponding labels.

An important fact is that this learning can be done separately for parameters of the unary and pair-wise potentials, β^u and β^p, respectively. As an example of this, Andriluka *et al.* [10] learn the unary potential parameters β_i^u discriminatively, using Adaboost over shape context features extracted from images, and pair-wise parameters β^p are learnt as Gaussian distributions via MLE.

5.3 Deformable Part Models

Following the same idea of pictorial structures, deformable part models [37, 96] also decompose an object into a set of different parts, connected to each other following the topology specified by means of a graph $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$. However, the hypothesis that parts share the appearance among different instances of the object under different deformations, is discarded.

One of the main problems when detecting an articulated object is that some of its parts can be eventually self-occluded by other ones, under certain deformations. Not only that, but slight changes in the viewpoint can result in dramatically different appearances. We illustrate this in Fig. 5.3, considering the human body as an articulated object, as in the model shown in Fig 5.2. While some viewpoints and poses of the human body may yield a clear view of most of its parts (see the arms in Fig. 5.3(a)), some other poses may produce self-occlusions of some parts (the arms in Fig. 5.3(b) are occluded by the legs), or even some parts can appear considerably foreshortened, due to the viewpoint (see upper legs in Fig. 5.3(c)).

(a) (b) (c)

Figure 5.3: Different poses of the human body.

In order to deal with these changes in appearance of the parts in the articulated model, the concept of mixtures of parts is introduced. Not only an object is decomposed into a set of different parts, but each part is defined as a mixture composed of different types of parts. We follow the same notation from the previous section and denote by L the configuration of parts in the model. However, in this case each part in the model $l_i = (x_i, y_i, t_i)$ is parametrized just by a position in the image and a type index $t_i \in \{1, ..., T\}$, where each part has T possible types of parts.

A score function equivalent to the energy funcion in Eq. 5.1 is defined then as:

$$S(I, L, \boldsymbol{\beta}) = S^b(L, \beta^b) + \sum_{i=1}^{M} S_i^u(I, l_i, \beta_i^u) + \sum_{(i,j)\in\mathcal{E}} S_{i,j}^p(l_i, l_j, \beta_{i,j}^p), \tag{5.4}$$

$$S^b(L, \beta^b) = \sum_{i=1}^{M} \beta_i^{b,t_i} + \sum_{(i,j)\in\mathcal{E}} \beta_{i,j}^{b,t_i,t_j}, \tag{5.5}$$

where $S^b(L, \beta^b)$ computes a bias that favors particular assignments for specific parts and types (parameters β_i^{b,t_i}), and their co-occurrences (parameters $\beta_{i,j}^{b,t_i,t_j}$).

The unary term S_i^u computes the score of considering configuration l_i, *i.e.* for placing part type t_i at position (x_i, y_i) in the image, for part i in the model. More specifically, the unary term is defined as

$$S_i^u(I, l_i, \beta_i^u) = \beta_i^{t_i} \cdot \phi(I, x_i, y_i), \tag{5.6}$$

where $\phi(I, x_i, y_i)$ represents a feature vector extracted from position (x_i, y_i) in image I, and $\beta_i^{t_i}$ are learnt weights on the features (as in the Linear SVM example shown in Fig. 5.1).

The pair-wise term $S_{i,j}^p$ computes the agreement between configurations of contiguous parts i, j in the model:

$$S_{i,j}^p(l_i, l_j, \beta_{i,j}^p) = \beta_{i,j}^{t_i,t_j} \psi(l_i, l_j), \tag{5.7}$$

where $\psi(l_i, l_j) = [dx, dx^2, dy, dy^2]$, computes first and second order differences $dx = x_i - x_j$ and $d_y = y_i - y_j$ among locations (x_i, y_i) and (x_j, y_j), and $\beta_{i,j}^{t_i,t_j}$ encode the flexibility of "springs" connecting parts i, j, tailored to their corresponding types t_i, t_j.

Given this score function, then the problem of detecting an articulated object is defined as finding the best configuration L^* with maximum score:

$$L^* = \arg\max_L S(I, L, \boldsymbol{\beta}). \tag{5.8}$$

5.3.1 Inference

As we just explained above, inference corresponds to maximizing the score function $S(I, L, \boldsymbol{\beta})$ over L. Again, similarly to pictorial structures, exact inference can be efficiently performed when the structure of $\mathcal{G}$ is acyclic, *i.e.* $\mathcal{G}$ is a tree, via message passing. The following messages from part i to its parent j are computed:

$$m_i(l_j) = \max_{t_i} \beta_{i,j}^{b,t_i,t_j} + \max_{(x_i,y_i)} score(l_i) + \beta_{i,j}^{t_i,t_j} \cdot \psi(l_i, l_j), \tag{5.9}$$

$$score_i(l_i) = \beta_i^{b,t_i} + \beta_i^{t_i} \cdot \phi(I, x_i, y_i) + \sum_{k \in kids(i)} m_k(l_i). \tag{5.10}$$

Starting from the leaf nodes to the root, Eq. 5.9 computes for all possible configurations l_j for part j the best scoring location and type of its child part i, and Eq. 5.10 computes the local score of part i, collecting messages from its children.

5.3.2 Learning

Since Eq. 5.10 is linear in $\boldsymbol{\beta}$, we could write $S(I, L, \boldsymbol{\beta}) = \boldsymbol{\beta} \cdot \Phi(I, L)$. In order to learn the parameters $\boldsymbol{\beta} = (\beta^b, \beta^u, \beta^p)$ of the model, a discriminative approach is considered, similar to a structural SVM. Given a set of labeled positive examples $\{I^n, L^n\}$, the following optimization problem is formulated:

$$\arg\min_{\boldsymbol{\beta}, \xi_n \geq 0} \quad \tfrac{1}{2}\boldsymbol{\beta} \cdot \boldsymbol{\beta} + C \sum_{n=1}^{N} \xi_n \tag{5.11}$$

$$s.t. \forall n \in pos \quad \boldsymbol{\beta} \cdot \Phi(I^n, L^n) \geq 1 - \xi_n \tag{5.12}$$

$$\forall n \in neg \quad \boldsymbol{\beta} \cdot \Phi(I^n, L^n) \leq -1 + \xi_n.$$

Basically, the constraints of this optimization problem state that positive examples should score more than 1, and negative examples should score less than -1. Violations of these constraints are penalized by slack variables ξ_n. In order to solve this quadratic program, stochastic gradient descent can be used, like in [37].

Chapter 6

Contextual Rescoring for Human Pose Estimation

6.1 Introduction

In this chapter we propose a contextual rescoring method for predicting the position of body parts in a human pose estimation framework. A set of poselets is incorporated in the model, and their detections are used to extract spatial and score-related features relative to other body part hypotheses. A method is proposed for the automatic discovery of a compact subset of poselets that covers the different poses in a set of validation images while maximizing precision. A rescoring mechanism is defined as a set-based boosting classifier that computes a new score for each body joint detection, given its relationship to detections of other body joints and mid-level parts in the image. In order to show its adaptability to other methods, we incorporate the proposed rescoring method in two state-of-the-art human pose estimation frameworks, one of them based in deformable part models [37], and another one based on pictorial structures [69]. Experiments on two benchmarks show comparable results to Pishchulin et al. [69] while reducing the size of the mid-level representation by an order of magnitude, reducing the execution time by 68% accordingly.

6.2 Related work

In the context of human pose estimation, pictorial structure models have been widely used. Since the search space of such highly-articulated models can be very large, some works proposed methodologies for reducing it. Ramanan [71] proposed an iterative inference process for the pictorial structure model. An edge-based appearance model of the body parts is firstly applied, giving a rough estimate of their location. Using this rough estimate, a color model of the foreground and the background is learnt, which helps in reducing the search space of the inference. A further extension of this idea was proposed by Ferrari *et al.* [40]; they directly segment the human body from the background of the scene, eliminating the need of a two-step inference. In contrast, Andriluka *et al.* [10] proposed to use stronger body part detectors based on dense shape context features, outperforming previous works.

Since the number of possible poses a person can adopt is huge, usually a simple pictorial structure model is not able to model all this variability in the data. In order to overcome this, some works increase the flexibility of pictorial structure models by adding pose prior

information in the model. For example, Sapp *et al*. [77] adjust the parameters of the pictorial structure model (particularly the ones regarding the pairwise potential) depending on image evidence. Similarly, Johnson and Everingham [52] partition the space of human poses into different clusters and learn specific pose priors and appearance models. In a further extension of their work [53], they replace a single pictorial structure model by a mixture model, such that each component of the model is specialized to a particular region of the pose space. Additionally, they present a new dataset of 10,000 images that they use as additional training data. Yang and Ramanan [96] followed the same trend and proposed a "flexible mixture of parts" model. Moreover, they proposed a new formulation of the problem; they model the body joints instead of the body parts, thus simplifying the formulation and reducing the complexity of inference. Specifically, the body joints are modeled as a mixture of small HOG filters [30] capturing a small neighborhood around them. While simplifying the inference and attaining better results than previous works, the performance of their method is still compromised by the use of a tree-structured model.

In contrast, some works have proposed to work with loopy graphs. Tian and Sclaroff [85] augmented the tree model by adding a small set of edges, and presented an efficient inference algorithm for their model. In the experimental section, they show how their method overcomes the so-called "double-counting" phenomena that tree-structures tipically suffer from. [83] proposed a branch-and-bound inference algorithm, allowing to compute exact inference on loopy graph models efficiently. [35] proposed a multi-layer pictorial structure model, incorporating body part evidence at different scales; from full body to local parts. In their proposed model, each layer is tree-structured by itself, but edges between adjacent layers make the whole model a loopy graph. However, this graph can be easily decomposed into tree-structured subproblems (which are amenable to exact inference), and they show how inference can be performed via dual-decomposition. Finally, [92] also proposed a hierarchical approach. They presented a manually-defined hierarchical decomposition of the human body; they introduced poselets ([15, 16]) as new parts in the model, adding also the corresponding set of higher-order edges with the existing basic parts. Since this extension results in a loopy-graph model, the method can only perform approximate inference.

Some recent works propose hierarchical tree models that are able to model high-order part dependencies by recursively dividing the human body into different mid-level parts, in addition to the set of basic body parts. Tian *et al*. [86] present a hierarchical model where leaf nodes correspond to body parts, and intermediate latent nodes cover different subsets of nearby body parts. Similarly, [92] and [90] also included mid-level poselet-based body parts in their pictorial structure model, but they propose an algorithm for discovering the best possible tree topology that connects all the parts. In contrast, [68] introduced higher-order part dependencies in their pictorial structure model by conditioning the unary and pairwise terms on poselet evidence. At training time, they cluster the relative distance between a particular part and the torso, and model this offset as a Gaussian distribution for each cluster. At test time, they solve a classification problem in which poselet responses are used to predict the correct cluster and recover the corresponding Gaussian distribution. This additional information about the location of that particular part is then incorporated in the pictorial structure model as an extra unary potential. They proceed analogously to predict part orientations and to learn the pairwise terms of the model. In a further extension of their work, [69] achieve a higher performance by incorporating additional parts that model body joints in their pictorial structure, following the idea of flexible spatial models from [96].

A bridge between human pose estimation and object detection was proposed by Yao and Fei-Fei [97]. They model mutual contextual information between poses and objects for a certain set of human-object interaction activities, like "tennis serve" or "croquet shot." The results indiciate that pose estimation can help object detection and vice versa. An-

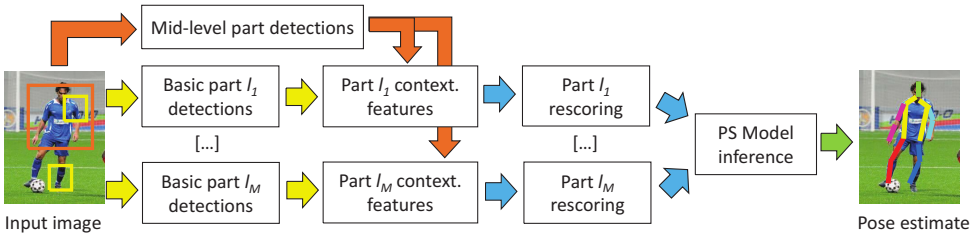


Figure 6.1: Proposed pipeline for human pose estimation. Given an input image, a set of basic and mid-level part detections is obtained. For each basic part detection l_i, a contextual representation is computed based on relations with respect to the set of mid-level part detections. Using these contextual representations, basic part detections are rescored using a classifier for that particular basic part class. The original and rescored detections for all basic parts are then used in inference on a pictorial structure (PS) model to obtain the final pose estimate.

other work on exploiting contextual information between objects was proposed by Cinbis and Sclaroff [25]; they present an approach for rescoring detections of different objects, introducing the notion of sets of contextual relations between object detections in an image. Each detection from a certain object class is represented by its context, defined as a set of detections from every other object detector. Then, a feature vector is extracted from each contextual detection, encoding spatial relations, relative scores and class-related relations. A generalization of the well-known Adaboost algorithm, called SetBoost, is used for rescoring an object detection given its set-based context representation.

6.3 Method overview

An overview of the proposed pipeline for human pose estimation is presented in Fig. 6.1. Following the notation used in Chapter 5, we denote by $L = \{l_1, ..., l_M\}$ the set of M parts in the model. We propose a method for obtaining robust part detections l_i in pictorial-structure-based human pose estimation frameworks, motivated by the aforementioned limitations of basic, low-level part detectors that are commonly used in pictorial structure models, *e.g.* HOG patches centered at body joints ([96]). We define and learn an additional set of mid-level body part detectors that improve the localization of the basic ones. Mid-level and basic part detectors are applied in order to extract a set of pairwise contextual features between each pair of basic and mid-level part hypotheses (see Fig. 6.3). A classifier for a certain basic part class will compute a new score for its detections, based on the set of contextual features computed between the basic and mid-level parts. The original and rescored detections for all basic parts are then used in inference on a pictorial structure model to obtain the final pose estimate.

In Section 6.4 we present the proposed mid-level part representation, and we introduce the rescoring methodology in Section 6.5. For the purpose of illustrating and evaluating the benefits of our contextual rescoring framework, we incorporate it within two different existing methods for human pose estimation: the discriminatively-trained deformable part model from Yang and Ramanan [96] in Section 6.6, and the pictorial structure (PS) model formulation of [69], in Section 6.7.

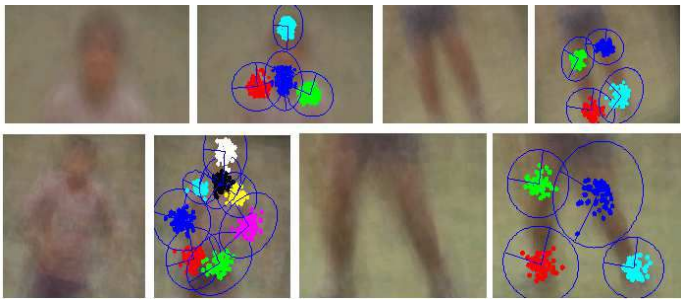


Figure 6.2: Sample Poselet templates. Body joints are shown with colored dots, and their corresponding estimated Gaussian distributions as blue ellipses.

6.4 Mid-level part representation

Since higher-level body parts model a larger image portion than just a small local patch as in the case of basic parts, it is expected they will perform better in terms of object detection performance. Moreover, poselets [16] model a specific human region extent under a certain pose configuration (see Fig. 6.2), so they are a suitable choice for the definition of the mid-level representation if one wants to predict the position of the body parts using their evidence. However, different strategies may be followed for the definition of a set of poselet detectors. While the manually-defined hierarchical decomposition presented by Pishchulin *et al.* [68] is sound and follows the kinematic constraints of the human body, the choice of its parameters may seem arbitrary, and not optimized for our position prediction task. In contrast, we propose a method for the automatic discovery of a compact subset of poselets to define a compact mid-level representation. We first review the manual hierarchical decomposition and then we introduce our automatic poselet discovery algorithm in the next lines.

6.4.1 Hierarchical decomposition

In their paper, Pishchulin *et al.* [69] define the set of poselets in their mid-level representation with guidance from a human annotator, similarly to [92]. In addition to the 10 basic parts in their original PS model, they also define a set of 11 parts, namely full body, upper body with arms, torso and head, right arm and torso, left arm and torso, right arm alone, left arm alone, torso with legs, legs, right leg alone, and left leg alone. Then, for each one of these parts they cluster relative positions of a subset of related body parts using K-means with a predefined number of clusters, discarding clusters with fewer than 10 examples. In practice, in their experiments they select $K = 200$, obtaining a final total number of $P = 1,013$ poselet detectors. While this hierarchical decomposition of the human body is directly motivated by the human body structure and kinematics, it might not be the best one for the task of predicting body parts locations since the choice of this mid-level representation is apparently arbitrary, without evaluating the prediction accuracy of the different body parts. Additionally, the number of poselets generated with this methodology can be very large, depending on the number of clusters defined for each mid-level part.

6.4.2 Poselet discovery

We propose a poselet selection methodology, inspired by the one proposed by Bourdev *et al.* [15], to define our mid-level parts $\mathbb{P} = \{1, ..., p, ..., P\}$. We first generate a large number P' (thousands) of random seed windows $B_{p'} = [x_{p'}, y_{p'}, wi_{p'}, he_{p'}, s_{p'}]$ from the training set images $\{I^n\}_{n=1}^N$, being $(x_{p'}, y_{p'})$ its position, $wi_{p'} \times he_{p'}$ its size, $s_{p'}$ its scale. Then, for each one of these seed windows we collect similar patches from other training images by Procrustes alignment of the body joint annotations (x_k^{gt}, y_k^{gt}) from the ground-truth. Next, for each seed window and its associated set of similar examples, we train a mid-level part detector. Additionally, we model the spatial distribution of the keypoints k contained within the spatial extent of each seed window as Gaussian distributions $(\mu_{p'}^k, \Sigma_{p'}^k)$ (see Fig. 6.2), that we use to evaluate the precision of predicting body joint locations. To that end, we test these poselet detectors in a validation set $\{I^n\}_{n=1}^V$, and look for True Positive (TP) and False Positive (FP) detections. In order to do that, we use the same criterion as the Percentage of Correctly-placed Parts (PCP) metric [40], widely used for evaluating human pose estimation methods. More specifically, we consider a detection as a TP if:

$$\text{dist}(\mu_{p'}^k, (x_k^{gt}, y_k^{gt})) \leq \kappa, \forall k \in B_{p'}, \tag{6.1}$$

where κ is a threshold value. That is, we classify a detection as a TP if the distance between the body joint estimations $\mu_{p'}^k$ and their corresponding ground-truth annotations (x_k^{gt}, y_k^{gt}) is below a threshold κ, for all the joints k contained in the poselet. Conversely, we consider a detection as a FP if none of the keypoints k fulfill the condition above. Since the seed windows are generated randomly, some of them will be redundant, or some others might have poor performance, so we could select a subset of relevant poselets and reduce the size of the mid-level part representation. This selection is treated as a "set cover" problem by Bourdev *et al.* [15]; poselets are selected in a greedy manner so as to "cover" more examples, i.e. the poselets that found TP detections in a larger number of training images. However, this methodology does not prioritize poselets with good performance if they only fire in a little subset of training images. In order to overcome this problem, we propose using a weighted version of the "set cover" problem, in which the precision of the selected poselets is maximized, while ensuring coverage of the images in a validation dataset. We define a binary matrix $A_{N \times P'}$ to keep track of which poselet proposal p' fires in which n-th validation image. Finally, we formulate this weighted "set cover" problem with the following integer programming:

$$\begin{aligned} \text{minimize} \quad & \sum_{p'} (1 - \text{Prec}(p')) \mathbf{x}_{p'} \\ \text{subject to} \quad & \sum_{p': A_{np'}=1} \mathbf{x}_{p'} \geq 1 \ \forall n \\ & \mathbf{x}_{p'} \in \{0, 1\}, \end{aligned} \tag{6.2}$$

where Prec(.) computes the precision of a poselet. The solution $\mathbf{x}$ will find the subset of poselets $\mathbb{P} = \{p' \mid \mathbf{x}_{p'} = 1\}$, i.e. a set of poselets ensuring that in every validation image there is at least one poselet that fires. The constraints of the integer program will enforce each validation image n to be covered by at least one poselet, but also the best-performing ones will be prioritized, since we are minimizing $(1 - \text{Prec}(p'))$. Given the NP-complete nature of the problem, we find the solution via a Linear Programming relaxation ($\mathbf{y}_{p'} \in \mathbb{R}_{\geq 0}, \mathbf{y}_{p'} \leq 1$) and round the solution $\mathbf{y}$ to obtain $\mathbf{x}$. The proposed method is summarized in Algorithm 5.

Algorithm 5 Poselet training and selection algorithm

Poselet training:
Generate random seed windows $B_{p'}, p' \in \{1, ..., P'\}$.
for $p' \in 1\{1, ..., P'\}$ **do**
  for $n \in \{1, ..., N\}$ **do**
    $Proc(n) \leftarrow$ Procrustes alignment cost between I^n with $B_{p'}$
  end for
  Train $\omega_{p'}$ with $I^n, \forall n \; s.t. Proc(n) \leq \tau_{proc}$
  Estimate Gaussian distribution $(\mu^k_{p'}, \Sigma^k_{p'})$
end for

Poselet selection:
$A_{np'} = 0, \forall n \in \{1, ..., V\}, \forall p' \in \{1, ..., P'\}$
for $p' \in 1\{1, ..., P'\}$ **do**
  $TP \leftarrow 0$
  $FP \leftarrow 0$
  for $n \in \{1, ..., V\}$ **do**
    $S_n \leftarrow \sum_{k=1}^{K_{p'}} [\text{dist}(\mu^k_{p'}, (x^{gt}_k, y^{gt}_k)) \leq \kappa]$
    if $S_n = K_{p'}$ **then**
      $TP \leftarrow TP + 1$
      $A_{np'} = 1$
    else if $S_n = 0$ **then**
      $FP \leftarrow FP + 1$
    end if
  end for
  $Prec(p') \leftarrow \frac{TP}{TP+FP}$
end for
$\mathbf{x}_{p'} \leftarrow arg\min \sum_{p'} (1 - Prec(p'))\mathbf{x}_{p'}$ (Eq. 6.2)
$\mathbb{P} \leftarrow \{p' \mid \mathbf{x}_{p'} = 1\}$

6.5 Contextual rescoring

We build our contextual model on top of the mid-level part representation presented in Section 6.4. More specifically, we want to model underlying spatial and score-related relationships between basic and mid-level part detections (see Fig. 6.3). By doing this, a certain mid-level part detection would be able to determine a hypothesis for the location of a certain basic part. For this task, we define the context of a given basic part detection l_i as a set composed by contextual feature vectors $c \in C$:

$$C_i^{\mathcal{P}} = \left\{ c_{B_i, B_p} \mid \forall B_p \in \mathcal{P} \right\}, \tag{6.3}$$

(6.4)

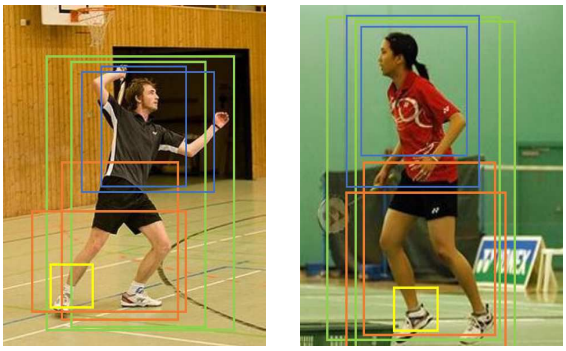


Figure 6.3: Two sample images depicting a reference detection bounding box B_i in yellow (for the right ankle), and the set of contextual mid-level detections $\mathcal{P}$ in blue, orange and green for the upper body, lower body and full body, respectively.

where $\mathcal{P}$ is a set of mid-level part detections (bounding boxes) B_p^i, generated by taking G detections for each part $p \in \mathbb{P}$:

$$\mathcal{P} = \left\{ B_p^i \right\}_{i=1,p=1}^{G,P}. \tag{6.5}$$

These contextual feature vectors c encode spatial, score-related and class-related relationships between a reference basic part detection B_i and a contextual mid-level detection B_p (we use the same set of features as [25]). The specific set of features we use is summarized in Table 6.1.

Finally, the rescoring function given a set of contextual feature vectors C is then defined as:

$$R(C) = \sum_{\theta=1}^{\Theta} Q_\theta(C), \tag{6.6}$$

$$Q_\theta(C) = \alpha_\theta \sum_{c \in C} k_c \cdot q_\theta(c), \tag{6.7}$$

where Q_θ is a weak set classifier, and q_θ is a weak item classifier, weighted by α_θ. The term k_c introduces an additional weight related to the relevance of the item. In practice, k_c is set to its corresponding detection score s_p, and q_θ functions are defined as decision trees with F leaves, which generate $U_1, ..., U_F$ partitions of the feature space. The weights α^f for each leaf f are computed as:

$$\alpha = \underset{\alpha^1,\ldots,\alpha^F}{\arg\min}\, \lambda \alpha^\top \alpha + \sum_{n=1}^{N} \mathcal{L}\left(y_n R(C_n) + y_n \sum_{f=1}^{F} \alpha^f H^f(C_n) \right), \tag{6.8}$$

where λ is a regularization parameter, $y_n \in \{-1, 1\}$ is the label of training sample n, $\mathcal{L}(z) = e^{-z}$ is the exponential loss function, and $H^f(C) = \sum_{c \in C, c \in U_f} k_c$ is the total sum of relevance weights for all items in the set C that fall into a given partition U_f. The whole training procedure is summarized in Algorithm 6. In order to train the rescoring function R_i for basic part l_i, we run its corresponding basic part detector on a validation set, as well as the

Table 6.1: List of contextual features included in c_{B_i,B_p}. For clarification, the detection score is encoded classwise in a sparse vector, i.e. a vector of size P set to zeros except the position corresponding to the class of the detection, which contains the detection score.

Feature	Value
detection score	$[0, ..., 0, s_p, 0, ..., 0]$
relative position	$(x_i - x_p)/he_i,\ (y_i - y_p)/he_i$
relative height	he_i/he_p
relative width	wi_i/wi_p
distance	$\|((x_i, y_i) - (x_p, y_p))\|$
overlap	$(B_i \cap B_p)/(B_i \cup B_p)$
score ratio	s_i/s_p
score difference	$s_i - s_p$

whole set of mid-level part detectors p. Then, for each basic part detection B_i, we compute the corresponding mid-level contextual feature set $C^{\mathcal{P}}_{B_i}$, and assign a label y_{B_i}:

$$y_{B_i} = \begin{cases} 1 & O(B_i, B_i^{gt}) \geq \tau \\ -1 & otherwise \end{cases}, \tag{6.9}$$

where B_i^{gt} is the bounding box for part l_i computed from the ground-truth annotation data, $O(B_i, B_j) = B_i \cap B_j / B_i \cup B_j$ is the overlapping Jaccard index between two bounding boxes, and τ is a threshold value. The complexity of rescoring a basic part detection is $\mathcal{O}(|\mathcal{P}| \cdot \Theta \cdot \bar{d})$, being $\bar{d}$ the average depth of the decision trees q_θ.

6.6 Deformable part model formulation

In this section we extend the original formulation of the deformable part model proposed by Yang and Ramanan [96], in order to include the proposed rescoring functions. Following the notation in Section 5.3, we denote by $L = \{l_1, ..., l_M\}$ a full body pose, consisting of M parts. Each part $l_i = (x_i, y_i, t_i)$ is parametrized by its center position (x_i, y_i) and its mixture component index $t_i \in \{1, ..., K\}$, being K the number of possible types for each part. As we comment in the related work, an important aspect of the work of Yang and Ramanan [96] is that, differently from most of the works in the field, the parts in their model represent the body joints instead of the limbs. Thanks to this, eventual foreshortening of the parts does not suppose a major problem, since the flexible springs between joints already model this kind of deformations. We define the score of a given pose L as follows:

$$S(I, L, \boldsymbol{\beta}, \mathcal{P}) = S^b(L, \beta^b) + \sum_{i=1}^{M} S_i^u(I, l_i, \beta_i^u, \mathcal{P}) + \sum_{(i,j)\in\mathcal{E}} S_{i,j}^p(l_i, l_j, \beta_{i,j}^p), \tag{6.10}$$

$$S^b(L, \beta^b) = \sum_{i=1}^{M} \beta_i^{b,t_i} + \sum_{(i,j)\in\mathcal{E}} \beta_{i,j}^{b,t_i,t_j}, \tag{6.11}$$

where $\mathcal{P}$ is the set of mid-level contextual detections, and $R_i^{t_i}$ is the rescoring function for part i and type t_i. Compared to the original score function (Eq. 5.4), the unary potential

Algorithm 6 SetBoost training algorithm ([25]). $D(n)$ contains the sample weight for set n in the training data

1. Initialize $D(n) = 1/N, n = 1...N$ and $R(C) = 0$.
2. For θ=1 to Θ
   (a) Sample subset of indices J with respect to D.
   (b) Train $q_\theta(c)$ to min. $\sum_{j \in J} \sum_{c \in C_e} D(j)[q_\theta(c) \neq y_j]$.
   (c) Solve for $[\alpha_\theta^1, ..., \alpha_\theta^F]$ via Eq. 6.8.
   (d) Let $Q_\theta(C) = \sum_{f=1}^{F} \alpha_\theta^f H_\theta^f(C)$.
   (e) $D(n) \leftarrow D(n) \exp(-y_n Q_\theta(C_n))$, $n = 1...N$.
   (f) Normalize D to a probability distribution.
3. Obtain $R(C) = \sum_{\theta=1}^{\Theta} Q_\theta(C)$.

now depends on $\mathcal{P}$, defined as:

$$S_i^u(I, l_i, \beta_i^u, \mathcal{P}) = \beta_i^{t_i} \cdot \phi(I, x_i, y_i) + \hat{\beta}_i^{t_i} \cdot R_i^{t_i}(C_i^{\mathcal{P}}), \tag{6.12}$$

where the feature extraction function $\phi(I, x_i, y_i)$ extracts HOG features from image I. We introduce the rescoring functions $R_i^{t_i}$ as a new term in the unary potential, weighted by a new set of parameters $\hat{\beta}_i^{t_i}$, which are learnt jointly at training time.

In order to efficiently train the model, we simplify the optimization procedure. We first need to train the basic part detectors, since we need to run them on the validation set in order to train the rescoring functions $R_i^{t_i}$. However, the weights $\beta_i^{t_i}$, related to such part detectors are jointly optimized with the rest of parameters, so we need to initialize them in some way. In fact, the optimization procedure proposed by Yang and Ramanan [96] initializes these weights $\beta_i^{t_i}$ by solving Linear SVM optimization problems independently for each part i and type t_i. Then, using these initialized weights we can apply them on the validation set, as well as the mid-level part detectors, and compute the sets of contextual features $C_i^{\mathcal{P}}$ that allow us to learn the rescoring functions $R_i^{t_i}$. Therefore, the rescoring functions are learnt just once at the beginning using the initialization of $\beta_i^{t_i}$, and used in the further steps of the optimization of the objective function (Eq. 6.10) in order to learn the whole set of parameters $\boldsymbol{\beta}$. Algorithm 7 summarizes all the steps in both training and test stages.

6.7 Pictorial structure formulation

In this section we propose a modification of the PS model proposed by Pishchulin *et al.* [69], so we start by introducing the original model formulation. As in the previous section, we define $L = \{l_1, ..., l_M\}$ a full body pose, consisting of M parts. However, each part $l_i = (x_i, y_i, r_i, s_i)$ is parametrized by its center position (x_i, y_i), rotation $r_i \in [0, 360)$ and scale $s_i \in \mathbb{R}_+$. In practice, images are rescaled so as to normalize the body size across the whole dataset, so s_i can be omitted. Pishchulin *et al.* [69] define the CRF energy function

Algorithm 7 Execution pipeline for learning and testing the extended deformable part model

Training:
Initialize basic part detectors $\beta_i^{t_i}$ independently via Linear SVM
Run basic part detectors on validation images
Train poselet detectors
Collect poselet responses on validation images
Train SetBoost classifiers for position prediction
Solve the corresponding Quadratic Program (see Eq. 5.11)

Test:
Run basic part detectors on test images
Collect poselet responses on test images
Run SetBoost classifiers for contextual position prediction
Run DPM inference

of a given pose L as:

$$E(I, L, \boldsymbol{\beta}, \mathcal{P}) = \sum_{i=1}^{M} E^u(I, l_i, \beta^u, \mathcal{P}) + \sum_{i \sim j} E^p(l_i, l_j, \beta^p), \tag{6.13}$$

where $\mathcal{P}$ is a set of mid-level part detections. The unary potential is defined as a weighted combination of two terms, like in the original formulation [69]:

$$E^u(I, l_i, \beta^u, \mathcal{P}) = E^{u,boost}(I, l_i, \beta_a^u) + \beta_p^u \cdot E^{u,poselets}(I, l_i, \mathcal{P}). \tag{6.14}$$

The first term $E^{u,boost}$ is the local appearance-based unary potential, defined as the log-likelihood obtained with pre-trained AdaBoost classifiers (parameters β_a^u). The second term $E^{u,poselets}$ is the one that incorporates evidence from poselet detections into the PS model, and β_p^u is a weight that balances both terms in the unary potential. More specifically, the authors use poselet detections to predict the position and rotation of the body parts. Their poselet-based feature representation is defined as a feature vector $f \in \mathbb{R}^P$, built by aggregating the maximum scores from the detections of poselets $p = 1, ..., P$ in an image. In addition, they apply a max-pooling step in order to obtain only relevant poselet detections around a certain area w.r.t the torso. During training, the relative position between a certain poselet p and the torso location is clustered into K clusters. Then, for each one of these clusters they model the relative offset from the torso as an isotropic Gaussian distribution. Finally, at test time the position prediction is formulated as a classification problem where one of these K clusters is predicted from a feature vector f, and the corresponding Gaussian distribution is recovered. They use the same approach in order to predict the rotation of a certain part l_i (see [69] for more details).

One of the main limitations of this approach is that the position (or rotation) prediction is restricted to one of the K Gaussian components (pre-defined at training time), so the choice of the value for K is decisive for the performance of the system. Moreover, another weakness of this approach is that it strongly relies on a torso location estimation, so a robust torso detector would be required to correctly predict the position of a body part l_i. The training and testing execution pipelines from [69] are summarized in Algorithm 8.

As an alternative to the position prediction approach from [69], we propose to discriminatively train a classifier to learn contextual relationships between detections from part l_i

Algorithm 8 Execution pipeline for [69]

Training:
Train basic part detectors & spatial model
Compute torso position prior
Train poselet detectors
Run torso detector on training images
Collect poselet responses on training images
Train LDA classifiers for position/rotation/pwise prediction

Test:
Run basic part detectors on test images
Run PS model inference (obtain torso position & rotation)
Collect poselet responses on test images
Run LDA classifiers for position/rotation/pwise prediction
Run PS model inference

and from a set of mid-level parts (e.g. poselets), and produce a poselet-conditioned postion prediction for part l_i, without the need to discretize the space of possible predictions into different clusters. In order to do that, we learn a different rescoring function R_i for each body part l_i, such that it assigns a new score to evidence from this part l_i, given a set of poselet evidence. We define:

$$E^{u,poselets}(l_i, \mathcal{P}) = R_i(C_i^{\mathcal{P}}), \tag{6.15}$$

where R_i is the rescoring function for part l_i, defined in Eq. 6.6, which receives as input a set of contextual feature vectors $C_i^{\mathcal{P}}$ and returns a new score for part l_i.

In contrast to Pishchulin *et al.* [69], our method does not strongly rely on any specific torso detection hypotheses, but on the ones from the whole set of basic parts in the PS model. Additionally, the proposed approach implicitly takes into account the detection scores from both poselet and basic parts' hypotheses as a confidence value to measure their trustfulness, and weight them accordingly to leverage high-scoring detections or weaken the low-scoring ones. Algorithm 9 shows the execution pipeline for training and testing of our proposed approach.

6.8 Experiments

We conducted experiments on two different benchmarks, comparing the human poses estimated by our proposed method with the results of state-of-the-art methods; especially with the results obtained by [69]. We first detail the data, methods and validation, and the evaluation measurements we fixed in our experimental setup. Next, we present the different experiments we performed when incorporating the proposed rescoring methodology in the deformable part model of Yang and Ramanan [96], and the pictorial structure model of Pishchulin *et al.* [69].

6.8.1 Data

We conducted experiments on two publicly available challenging datasets: Leeds Sports (LSP) [52], which comprises by 2,000 images of people playing 8 different sports, and PARSE

Algorithm 9 Execution pipeline for our proposed method

Training:
Train basic part detectors & spatial model
Run basic part detectors on validation images
Train poselet detectors
Collect poselet responses on validation images
Train SetBoost classifiers for position prediction

Test:
Run basic part detectors on test images
Collect poselet responses on test images
Run SetBoost classifiers for contextual position prediction
Run PS model inference

dataset [71], which contains 305 images of people playing different sports and activities. The annotations for both datasets consist of 14 position labels, one for each body joint: left/right ankle, knee, hip, wrist, elbow and shoulder, neck and head top. In the case of LSP, the annotations are observer-centric, i.e. left/right labels on the limbs are defined as the left-most/right-most limb in the image respectively. In contrast, the labels in PARSE are person-centric, i.e. left/right labels are related to the actual left/right limbs of the person in the image. We divided the training set of LSP into 2 subsets: training and validation. The final training set (contains 75% of the images in the whole original training set) is used for learning the PS model. The validation set contains the remaining 25% of the images and is used for learning the rescoring functions R_i and the final subset of poselets. In the case of the PARSE dataset, we just use the test split in order to test our rescoring approach, pre-trained on LSP.

6.8.2 Methods and validation

Our poselet selection method automatically selects 47 poselets in the LSP dataset, from an initial set of $2,000$ poselet proposals (see Fig. 6.4). In order to define the set $\mathcal{P}$ of contextual detections, we take the $G = 2$ best detections from each mid-level part detector. Each rescoring function $R(C)$ is defined as a forest of $\Theta = 20$ decision tree weak classifiers, each one of them having a maximum of 150 leaf nodes. In addition, we use $\lambda = 0.01$ and $\tau = 0.6$. For the experiments involving the deformable part model from Yang and Ramanan [96], we downloaded the last version of their implementation and re-trained their model using $M = 14$ parts, and each part being defined as a mixture of $T = 6$ components. We compare the results obtained with our re-formulation of the score objective function (Eq. 6.10) with other state-of-the-art methods for human pose estimation, paying especial attention to the results from Yang and Ramanan [96]. The Percentage of Correctly-placed Pars (PCP) [40] is used as the evaluation measure to numerically quantify the performance of the methods in the comparison. Moreover, we also provide some image examples to show qualitative results.

For the experiments with the pictorial structure model from Pishchulin *et al.* [69], we downloaded their implementation and re-trained their model for our experimental setup. The PS model we use for our experiments has $M = 22$ parts: while the original PS model presented by [68] was composed by $M = 10$ parts (left/right lower legs, left/right upper legs, torso, head, left/right forearms, left/right upper arms), a further extension proposed in [69] introduced an additional set of parts modeling the body joints of the limbs, thus resulting in

(a) (b)

Figure 6.4: Sample poselets from the LSP dataset. (a) Poselets with highest precision. (b) Poselets discovered by our selection method, maximizing precision and enforcing covering of the validation set.

a total of $M = 22$ parts. For the manually-defined mid-level representation, we first use the parameters reported in their paper: we fix $K = 200$ clusters for each one of the 11 mid-level parts, resulting in a total number of $P = 1,013$ poselets. Furthermore, we also fix K so as to obtain the same number of poselets as our poselet selection method does. We compare our proposed formulation against the original PS model of [69] on both benchmark databases, as well as with some relevant methods from the state of the art. Again, we use the PCP for the quantitative analysis of the results. In addition, we perform further analyses of the results, w.r.t. the work of Pishchulin *et al.* [69]. More specifically, we analyze: the effect of β_p^u on the results, the contextual features learnt by R_i^{ti}, and the execution times of both pipelines.

6.8.3 Experiments with deformable part models

In Table 6.2 we show the performance of the proposed pipeline summarized in Algorithm 7, and other related methods from the state of the art in the LSP dataset. We divide the table into three sections depending on the feature representation that each method uses, in order to do a fairer comparison with the state of the art. We are specially interested in comparing the proposed reformulation of Yang and Ramanan [96] against their original formulation. While the average improvement is very small (+0.54% PCP), it is worth noting that our proposal yields especially better results when estimating the pose in the lower body: we get +2.5% PCP in the case of the upper legs, and +2.8% for the lower legs. Fig. 6.5 clearly shows this improvement in the localization of the lower body limbs. While the DPM proposed by Yang and Ramanan [96] suffers from the "double-counting" phenomena in many images, our proposal is able to correct this errors in most of the cases.

The "double-counting" phenomena is due to the tree-structure of the model: since the appearance of left and right parts in the model is very similar (usually the part detectors are trained using left and right versions of the same part), left and right detectors from a same part, *e.g.* left hand and right hand detectors, may produce high scoring locations at the same region images. Then, the pair-wise term in the score function may eventually accept such result as a plausible part configuration, confusing for example a frontal view with a profile view (see examples in Fig. 6.5). In contrast, the presented rescoring approach together with the proposed compact mid-level part representation based on poselets is able to encode higher-order part dependencies, in addition to those encoded by the tree-structured

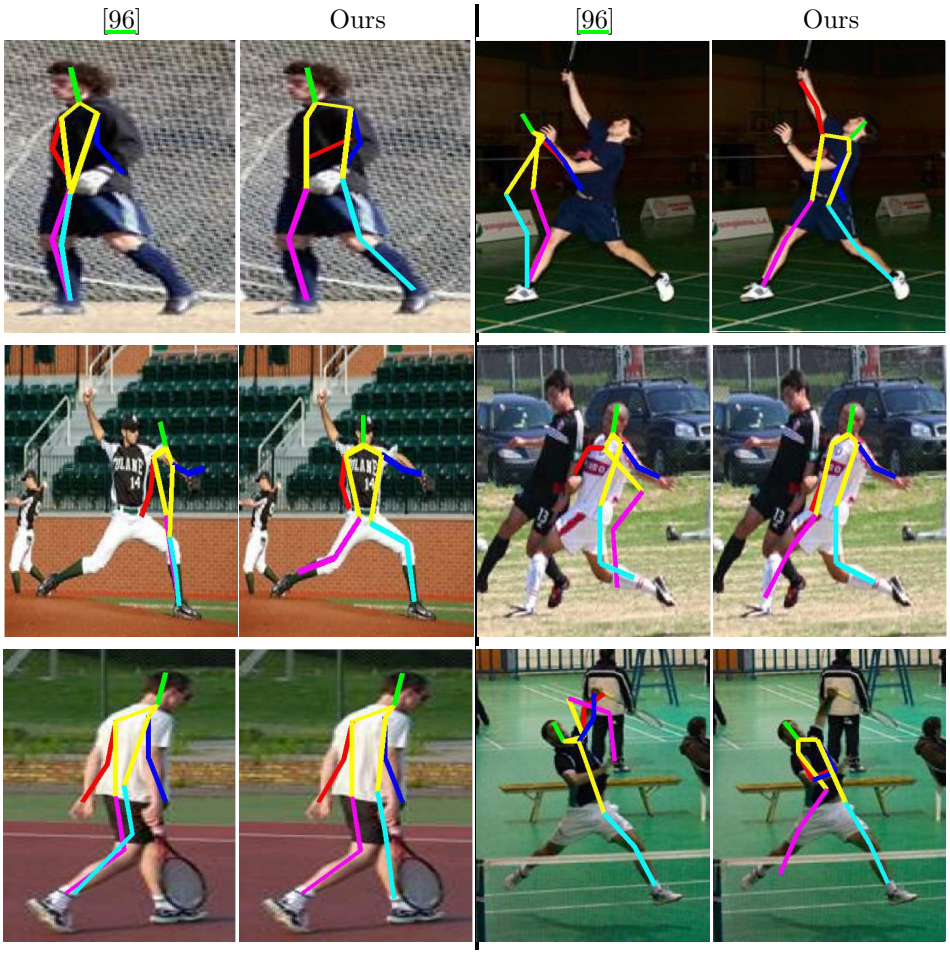


Figure 6.5: Qualitative results for the proposed rescoring approach incorporated in the DPM model from Yang and Ramanan [96], in the LSP dataset

deformable part model. More specifically, the rescoring of the part detections helps to disambiguate between left and right body parts as long as the poselets in the mid-level representation are able to discriminate between left and right sides of the human body.

Results on the PARSE dataset are presented in Table 6.3. We use the mid-level representation learn in the LSP dataset, using our proposed poselet selection method, and learn the rescoring functions on the validation set from the PARSE dataset. In this case, our proposed method yields a lower improvement w.r.t Yang and Ramanan [96], compared to the results on the LSP dataset. More specifically, we obtain +1.46% PCP in the lower legs, and +1.72% PCP in the forearms. Qualitative results in Fig. 6.6 show some example images where our proposed method yields better localization of the forearms, mainly.

All in all, the performance improvement of our reformulation, w.r.t Yang and Ramanan [96] seem not no be very significant. We attribute this to the fact that we are not re-training the rescoring functions at each step of the DPM optimization, but we are

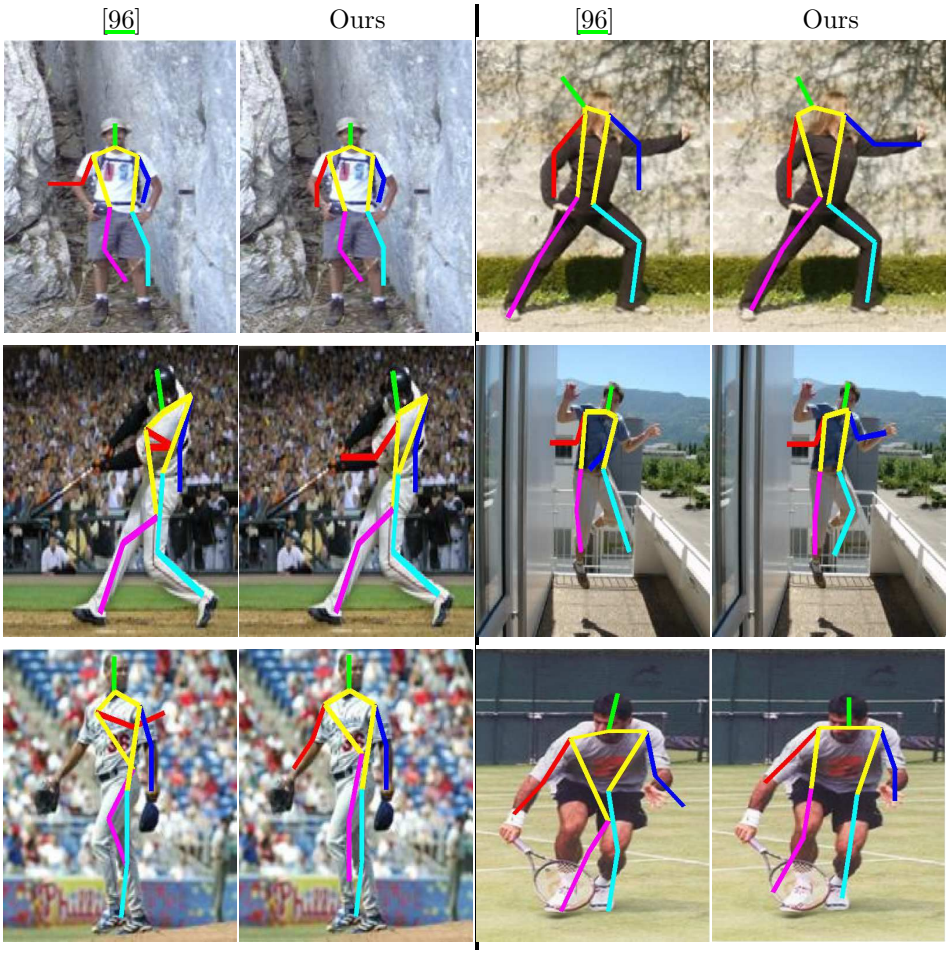


Figure 6.6: Qualitative results for the proposed rescoring approach incorporated in the DPM model from Yang and Ramanan [96], in the PARSE dataset

learning the rescoring functions just once using the initialization of weights $\beta_i^{t_i}$, performed for each part detector independently at the beginning of the optimization. In further experiments with the pictorial structure model from Pishchulin *et al.* [69] (which allows to learn the unary and pair-wise potentials independently, in contrast to the joint discriminative learning of DPMs), we better illustrate the benefits of the proposed rescoring mechanism.

6.8.4 Experiments with pictorial structures

In the first part of the experimental results we compare the position prediction from [69] against the one we propose, based on contextual rescoring. In this first comparison, rotation prediction is disabled, and the pair-wise potential is not conditioned on any poselet evidence. Furthermore, specialized head and torso detectors are disabled, as well as the torso position prior, in order to better illustrate the influence of the position prediction in the final pose

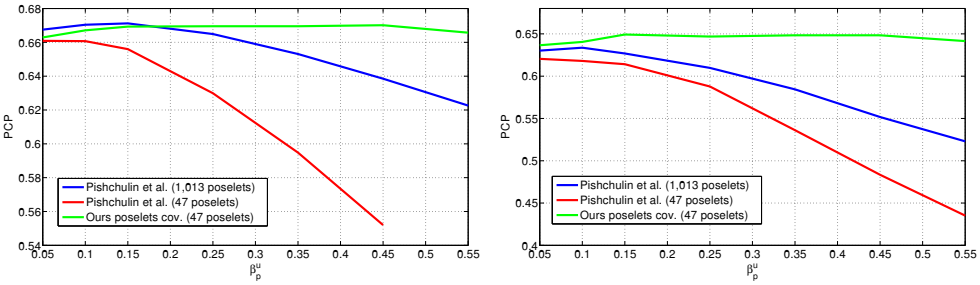


Figure 6.7: Position prediction comparison in (a) LSP and (b) PARSE datasets. In each plot, PCP performance is shown as a function of β_p^u. We compare our proposed rescoring approach when using $P = 47$ poselets, automatically selected by our proposed poselet discovery method, w.r.t. the position prediction from [69] with $P = 1,013$ and $P = 47$ poselets.

estimation results. Moreover, we compare different mid-level representations including our proposed poselet selection method maximizing precision and enforcing covering, selecting the poselets with maximum precision, the greedy algorithm from [15], and the manually-defined hierarchy from [68]. Next, we compare our proposal with respect to relevant works from the state of the art. Finally, we analyze the contextual features selected by the SetBoost classifiers and the execution time of our proposal and the full model from [69].

Position prediction evaluation

We first evaluate the proposed part position prediction based on contextual rescoring in conjunction with our proposed poselet selection algorithm for the automatic discovery of the mid-level representation. We compare it against the position prediction method proposed by [69] and their manually-defined hierarchical decomposition as mid-level representation. One can see in Fig. 6.7 how our proposed position prediction yields maximum performance that is similar to [69] for the LSP dataset, while reducing the number of poselets P by more than an order of magnitude. We can also observe how the performance of [69] drops for $\beta_p^u \geq 0.15$, while our approach gets the maximum performance for $\beta_p^u = 0.45$ and yields a much smoother performance response when varying β_p^u. Additionally, when using [69] and reducing P so as to match the size of our proposed mid-level representation, their performance drops more sharply.

When testing on the PARSE dataset using the same models we learnt on LSP, the behavior is similar, even obtaining a slightly better performance w.r.t. [69]. We noted that the optimal values for β_p^u in terms of PCP performance are higher when using our contextual rescoring approach, w.r.t. [69] ($\beta_p^u = 0.45$ vs. $\beta_p^u = 0.15$ in the LSP dataset, and $\beta_p^u = 0.2$ vs. $\beta_p^u = 0.1$ in the PARSE dataset), indicating that our extra unary potentials incorporate more relevant information to the PS model in comparison to [69]. This could be due to the unimodality nature of the Gaussian-shaped extra unary potential $E^{u,poselets}$ defined by [69]. In contrast, our rescoring approach does not limit $E^{u,poselets}$ to be unimodal, and thus can keep several possible position prediction hypotheses that can eventually be selected (see column (d) in Fig. 6.11 and Fig. 6.12).

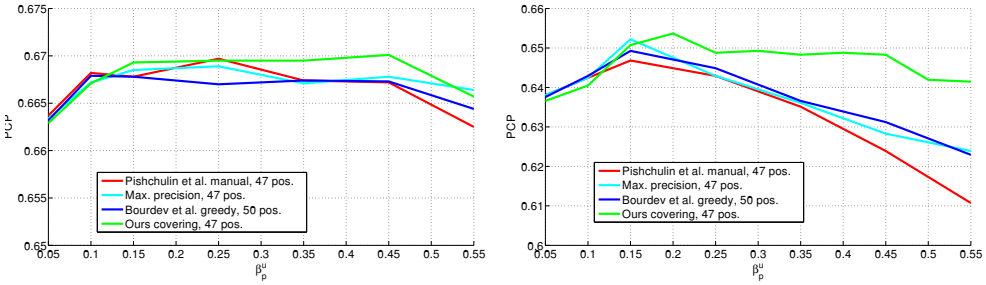


Figure 6.8: Comparison of different mid-level representations in (a) LSP and (b) PARSE datasets. In each plot, PCP performance is shown as a function of β_p^u. We compare our poselet selection maximizing precision and enforcing covering against (1) the manual hierarchical decomposition from [69], (2) selecting the poselets with maximum precision, and (3) the poselet selection greedy method from [15].

Figures 6.11 and 6.12 show some qualitative examples (as well as the corresponding extra unary potentials $E^{u,poselets}$) in the LSP and PARSE datasets, where our proposed position prediction approach obtains better results than the one from [69]. We observed that the method of [69] is prone to fail in cases where the torso detection is unreliable due to the viewpoint or the pose itself, since their approach strongly relies on the torso location hypotheses in order to introduce information from the mid-level representation into the model. In contrast, our proposed rescoring mechanism is able to correctly recover most of the body parts when the torso is hard to detect.

In order to validate our proposed poselet selection method, we run the pipeline in Algorithm 9 choosing different alternatives for the definition of the mid-level representation, and compare the obtained respective performance in terms of PCP. Fig. 6.8 shows that our proposed poselet selection method maximizing precision and enforcing covering yields the best results. While there is no significant difference in performance between the different strategies in the LSP dataset, the performance of our method clearly stands out for the PARSE dataset. It is worth noting that the proposed method automatically finds the appropriate number of poselets P, similarly to the method proposed by [15]. Taking this into account, our method obtains better performance than [15] while still selecting a smaller number of poselets ($P = 47$ vs. $P = 50$).

Comparison with state of the art

In Table 6.2 we compare our proposed method with other recent methods in the state of the art, for the LSP dataset. Our proposed formulation based on contextual rescoring and automatic poselet selection obtains slightly worse results (-1.77%) than the best performing method: [69] (in this case we show results using their full model). However, our approach just uses $P = 47$ poselets in contrast to their set of $P = 1,013$ (a reduction of 95%), hence our method is much faster as we will show in Section 6.8.4. Moreover, our method reaches similar performance to [69] in only predicting position, while the latter uses heavier artillery (position, rotation and pair-wise prediction, specialized head & torso detectors, and torso position prior).

Table 6.2: Pose estimation results for LSP dataset. The table shows the PCP for each part separately, as well as the mean PCP. Columns with two values indicate the PCP for the left and right parts, respectively. The methods in the table are grouped according to the features they use, namely: HOG (H), HOG + RGB (HC) and Shape context (SC). We compare our rescoring proposal and mid-level image representation computed with our proposed poselet selection method, against the state of the art. $*$ They use extra 11,000 images for training. $\diamond$ 14 parts. † ($P = 1,013$). ‡ ($P = 47$, pred-pos only).

Method		Torso	Upper Leg		Lower Leg		Upper Arm		Forearm		Head	Mean
H	[52]	78.10	64.80	66.70	60.30	57.30	48.30	46.50	34.50	31.20	62.90	55.21
	[53]*	88.10	74.50		66.50		53.70		37.50		74.60	62.70
	[96]$^\diamond$	82.90	65.30	66.70	61.80	61.50	50.60	49.30	27.90	27.40	78.10	57.15
	[96]R	83.70	67.70	70.00	64.50	64.40	49.00	48.90	25.90	25.30	77.50	57.69
	[86]	**95.80**	69.90		60.00		51.90		32.90		**87.80**	61.30
	[90]	91.90	74.00		69.80		48.90		32.20		86.00	62.80
HC	[36]	86.20	74.30		69.30		56.50		37.40		74.00	64.14
SC	[10]	80.90	67.10		60.70		46.50		26.40		74.90	55.70
	[69]†	88.30	**79.00**	**77.80**	**74.10**	**72.80**	**60.80**	**60.70**	**45.60**	**43.30**	85.40	**68.78**
	[69]R‡	87.70	78.20	75.80	73.20	71.80	58.70	55.90	43.80	40.20	84.80	67.01

In the PARSE dataset, we test the model from [69] (and our proposed reformulation) trained on the LSP dataset; the other numbers in the table are gathered from the corresponding papers. In this case, [86] reports outstanding results (74.4%). However, they also report a mean PCP of 67% for the PARSE dataset when they train their model on the LSP dataset, which yields a fairer comparison against [69] and our proposal. [53] still gets higher performance (67.40%), but they use an additional 11,000 images for training, so the results are not directly comparable. Our method scores 65.37% (-1.36% w.r.t. [69]) in this case, following the same trend as in the LSP dataset.

Qualitative results in the LSP and PARSE datasets using the full model from [69] can be seen in column (c) from Fig. 6.11 and Fig. 6.12, respectively. As we discussed in the previous subsection, our proposed rescoring approach can cope with images where the torso is hard to detect, while [69] fails. In addition, our proposed approach is more resilient to changes in viewpoint and scale, as we can see in Fig. 6.11. These images present some cases where some body parts suffer from foreshortening (rows 2-3 in Fig. 6.11) or lengthening (row 4 in Fig. 6.11) due to the viewpoint. In these cases, fitting a PS model in the image may become an ill-posed problem, since the expected size ratio among the different body limbs is not kept the same and the PS model assumes the scale is fixed for all the parts. However, our proposed rescoring approach is inherently invariant to scale, since the computed contextual features regarding spatial coordinates are normalized by the reference body part detection height. As a result, the proposed method tends to recover from these aberrations by placing the body parts in the image in a more visually-coherent way (row 4 in Fig. 6.11).

As shown in Fig. 6.9, there are cases where our method does not produce a pose estimate or localization of body parts that agrees with the ground-truth. These challenging images include: (a) multiple people and (b) strongly rotated poses, e.g., people doing handstands. When there is more than one person in the image, the poselet detectors are likely to fire on all the people appearing in the scene, while we are only interested in a specific subject (the one for whom the ground-truth annotations were made available). In future work, this could be addressed by first running a person detector, and then running the proposed pipeline in Alg. 9 separately for each detected subject. When a person appears strongly rotated, then

Table 6.3: Pose estimation results for PARSE dataset. See Table 6.2 for table legend. *They use extra 11,000 images for training. ⋄ 14 parts. † ($P = 1,013$). ‡ ($P = 47$, pred-pos only).

Method		Torso	Upper Leg		Lower Leg		Upper Arm		Forearm		Head	Mean
H	[52]	85.40	76.10	70.70	69.80	61.00	64.90	64.40	49.30	44.40	76.10	66.20
	[53]*	87.60	76.10	73.20	68.80	65.40	69.80	64.90	48.30	43.40	76.80	67.40
	[96]⋄	81.46	68.78	61.95	62.44	54.63	56.59	48.78	30.73	26.34	81.46	57.32
	[96]R	81.95	68.78	60.98	65.85	54.15	55.61	45.85	32.22	28.29	80.49	57.41
	[35]	85.60	71.70		65.60		57.10		36.60		80.40	62.80
	[86]	**97.10**	**85.10**		**76.10**		**71.00**		45.10		**92.20**	**74.40**
SC	[10]	81.40	67.30	59.00	63.90	46.30	47.30	47.80	31.20	32.10	75.60	55.20
	[69]†	89.76	74.15	71.71	66.83	60.00	65.37	57.07	**49.76**	**47.80**	84.88	66.73
	[69]R‡	89.27	71.71	70.24	69.76	60.98	60.98	56.59	47.80	47.80	78.54	65.37

the localization of body parts can be incorrect. Since images that depict rotated poses are relatively rare in the training dataset, our mid-level representation is unlikely to represent them well. Nevertheless, it is likely that there will be a rotated version among the selected poselets in our model. Thus, in future work, it may be possible to address such cases by extending our contextual features to include relative rotation features, and run the poselet detectors at a wider range of rotations. When a person appears strongly rotated

Contextual features

In Fig. 6.10 we analyze the most selected contextual features that the decision trees q_θ choose for splitting each node, in order to see which features are more important in each case. More specifically, we show the feature selection histograms for different subsets of the human body parts: upper-body limbs, lower-body limbs, head & torso and full body. We see that in a general trend, the signed relative position (rel.x and rel.y) is the most important contextual feature, since it is in the top three most selected features in all cases. This tells us that our rescoring functions are able to exploit the pattern about the relative locations of the body parts and the locations of poselet detections. Furthermore, it is interesting to note that in the case of the lower-body limbs, the relative position in the x-axis is selected more times than in the y-axis (in fact, the relative distance (dist) is selected more than rel.y). One possible explanation for this is that the pattern between left and right lower body limbs and the poselet detections tends to be more consistent, in contrast with the upper-body. Interestingly, the opposite case takes place with the head and the torso: relative position in the y-axis is selected more often than in the x-axis. In this case, the head and the torso are usually centered w.r.t the body parts, so vertical relationships tend to be more discriminative than horizontal ones.

Finally, the relative score is the second most selected contextual feature, in general terms. While the score difference (score.diff) is more frequent in the case of head & torso and the lower-body limbs, the score ratio stands out in the case of upper-body limbs, and is slightly more frequent in the full body case.

In conclusion, the proposed rescoring approach not only learns the patterns of the spatial layout between the basic parts in the PS model and the poselets in the mid-level representation, but also learns and exploits the patterns of the outputs from the corresponding detectors, as in other machine learning techniques like stacked sequential learning ([70]).

(a) (b)

Figure 6.9: Failure cases in (a) LSP and (b) PARSE datasets. Our proposed method cannot recover the human pose correctly, mainly due to upside-down poses and cases with extra people close to the actual subject.

Execution time

In this subsection, we compare the running time of the proposed test execution pipeline (Algorithm 9) with respect the one proposed by [69] (Algorithm 8). Running times of both pipelines are summarized in Table 6.4. Considering that basic part detectors have been already run (it is a common step in both pipelines), we achieve a running time reduction of 68.23%: our method takes 209.36 sec. in total, in contrast to the 659.15 sec of [69]. This reduction comes mainly from the mid-level representation size reduction: the running time for the poselet detectors is reduced by 95.24%. Furthermore, we just need to perform PS model inference once at the end of the execution pipeline ([69] needs a first inference to obtain a good torso position hypothesis), resulting in an additional time saving.

It is worth noting that we could not evaluate the running time of the specialized torso detector used by [69] since the code for that part of the pipeline is not available, so the real time reduction we can achieve may be slightly greater.

6.9 Discussion

We have proposed a contextual rescoring methodology for predicting the position of body parts in a human pose estimation framework based on pictorial structures. This contextual rescoring approach benefits from a mid-level body part representation to obtain more confident hypothesis for locating basic body parts. In order to define this mid-level representation, we propose an algorithm for the automatic discovery of a set of poselets that maximizes precision while enforcing covering of the different poses that appear in a validation set. Using spatial and score-related features extracted from the set of basic and mid-level part detections, we rescore the body joints hypotheses and combine them with the original

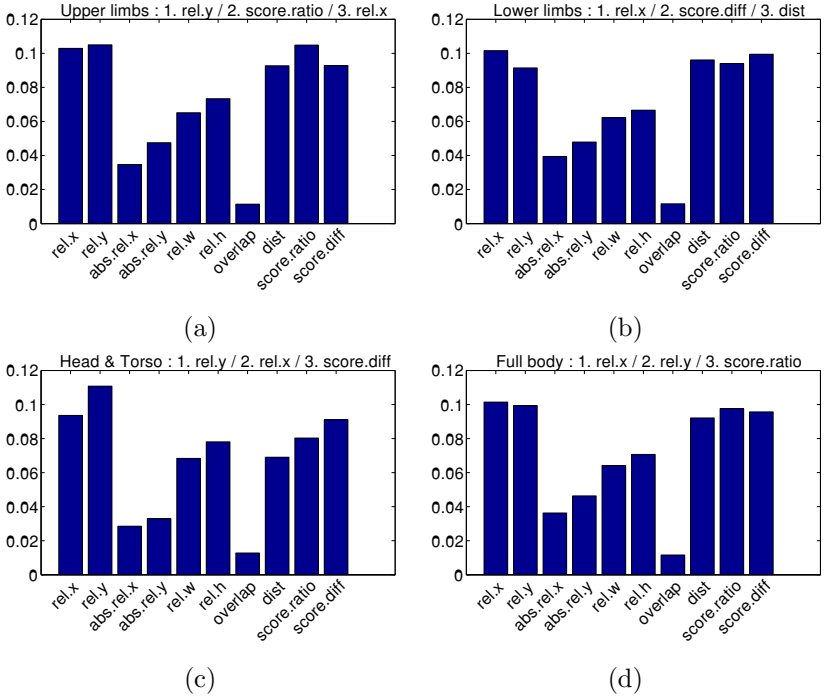


Figure 6.10: Contextual feature selection histograms computed from the learnt decision trees q_θ, grouped by subsets of joints: (a) upper-body limbs, (b) lower-body limbs, (c) head & torso, and (d) full body.

Table 6.4: Running time (in seconds) of the test pipelines from [69] (Algorithm 8) and our proposal (Algorithm 9).

Steps	**Alg. 8**	**Alg. 9**
1. Run PS model inference (torso hyp.)	148.70	-
2. Run poselets	344.77	16.39
3. Run LDA/SetBoost classifiers	3.06	23.37
4. Run PS model inference	162.62	169.60
TOTAL	659.15	209.36

scores in the unary potential of a PS model.

Experiments with two standard benchmarks demonstrated that our body part position prediction formulation can yield a performance similar to that of [69] in the LSP dataset and even better in the PARSE dataset, while reducing the number of poselets in the mid-level representation by 95.36%. Compared to other automatic poselet discovery strategies, the one proposed in this chapter yields the best results in combination with the proposed rescoring mechanism, in terms of PCP performance in the task of human pose estimation.

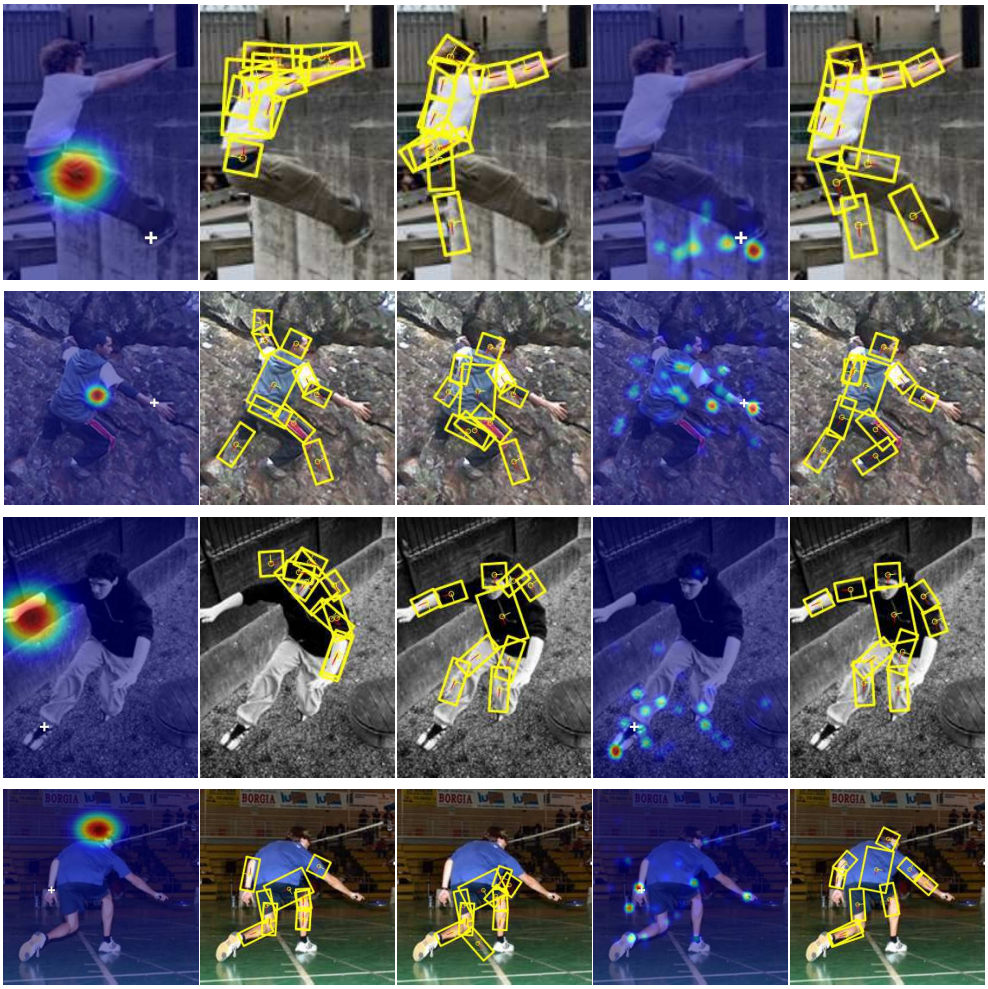


Figure 6.11: Qualitative results on LSP dataset. (a) Gaussian-shaped position prediction from [69], (b) Estimated pose from [69] (just predicting position), (c) Estimated pose from [69] (full model), (d) Position prediction with our proposed rescoring, and (e) Our results. White crosses in columns (a) and (d) show the part being rescored in each case; from the first row to the last one: rightmost ankle, rightmost wrist, leftmost ankle, leftmost wrist.

In addition, the proposed poselet discovery method is fully automatic, while the hierarchical decomposition from [69] is manually-defined.

In experiments that examined the influence of β_p^u, the weight for the extra unary potential that encodes position prediction based on poselet evidence, we noted that the results (PCP) are more stable with respect to changes β_p^u, vs. [69]. Perhaps more crucially, the optimal

Figure 6.12: Qualitative results on PARSE dataset. (a) Gaussian-shaped position prediction from [69], (b) Estimated pose from [69] (just predicting position), (c) Estimated pose from [69] (full model), (d) Position prediction with our proposed rescoring, and (e) Our results. White crosses in columns (a) and (d) show the part being rescored in each case; from the first row to the last one: right ankle, right wrist, right ankle, left ankle.

value for β_p^u is higher in our formulation vs. [69]: 0.45 vs. 0.15 for LSP, and 0.2 vs. 0.1 for PARSE. This suggests that our proposed rescoring approach can produce a more reliable prediction of the part location in the image. A further analysis on the contextual features selected by the weak classifiers in the rescoring approach reveals that the most important features are the relative position and relative detection scores. This affirms that our method can discover patterns of the spatial layout of body parts and thus improve the body part localization obtained from the outputs of standard body part detectors.

Finally, we note that the reduction in the mid-level part representation results in a

significant reduction in the execution time. We reduce the execution time by 68.23% with less than 2% reduction in accuracy w.r.t. the full model from [69]. Moreover, we just predict the position of the body parts, while [69] also predicts their rotation and the pair-wise parameters, and makes use of additional specialized head and torso detectors and a torso position prior.

As future work, we could adapt the proposed rescoring approach for predicting not only the position, but the rotation of the body parts. Additionally, we could extend the set of contextual features in order to directly model relative orientations between basic and mid-level parts through the contextual rescoring functions.

Part III

Gesture Recognition

Symbol notation in Part III

Table 6.5: Symbols and conventions for chapters 7- 8

S_{RGB}	Set of interest points in RGB modality
S_D	Set of interest points in depth modality
$\mathcal{P}$	Number of points in the cloud of points
$\rho^{(i)}$	i-th point in the cloud of points
$\tau^{(i)}$	Normal of $\rho^{(i)}$
P_{xy}	Plane ortogonal to viewing axis z
$\tau_{xy}^{(i)}$	Projection of $\tau^{(i)}$ in P_{xy}
ϕ	Angle between $\tau^{(i)}$ and the viewing axis z
ψ	Projected angle between $\tau_{xy}^{(i)}$ and the y-axis
V	Number of visual words in the vocabulary
b_u, b_v, b_p	Number of bins in the u,v and p dimensions, respectively, of the pyramid decomposition of the spatio-temporal volume of the video sequence
h^{RGB}	Histogram of the visual words in the RGB vocabulary
h^D	Histogram of the visual words in the depth vocabulary
d^F	Distance function for k-NN classification, $F \in \{RGB, D\}$
β	Weight parameter associated to d^D
$Q = \{q_1, ..., q_n\}$	Input sequence
$C = \{c_1, ..., c_m\}$	Model sequence
M	DTW matching cost matrix
$W = \{w_1, ..., w_\tau\}$	Warping path of length τ
θ	Threshold over the cost matrix M
$d(i, j$	Euclidean distance between feature vectors c_i and q_j
$S = \{S_1, ..., S_N\}$	Training set of N sequences for a gesture class
$S_g = \{s_1^g, ..., s_{L_g}^g\}$	Sample gesture sequence of length L_g
$\bar{S}$	Median length sequence among S
$\hat{S} = \{\hat{S}_1, ..., \hat{S}_N\}$	Warped training set
$\hat{F}_t = \{f_t^1, ..., f_t^N\}$	Set of feature vectors in the training set, at time t
$\lambda_t = \{\alpha_k^t, \mu_k^t, \Sigma_k^t\}$	Parameters of the GMM, $k = 1, ..., G$
G	Number of components of the GMM
$\mathcal{N}(x)$	Three upper-left neighbor locations of x in M

Chapter 7

BoVDW for gesture recognition

7.1 Introduction

Gestures play an important role in human communication. We use them naturally in our daily lives, from linguistic to artistic points of view, including for example referee signals in sports, gestures to accompany speech or sign language for the deaf. More specifically, a gesture is a form of non-verbal communication involving the movement of some parts of the body (being usually the arms and the hands the most used ones), with the purpose of expressing a sign or an action with a specific meaning.

Gestures take place in the spatio-temporal domain, given that they imply a movement of the body parts among the 3-D spatial coordinates, during a specific amount of time. Therefore, a gesture is delimited by a "beginning" and an "end" time instants. In the Computer Vision field, gesture recognition methodologies typically distinguish two steps: gesture segmentation, and gesture recognition *per se*. Usually, gesture segmentation is firstly performed to find the "beginning" and "end" of the gestures performed in a video sequence. Then, gesture recognition is defined as a classification step in which a gesture label is assigned to a given time segment, previously found in the gesture segmentation step.

Human gesture recognition is one of the most challenging tasks in computer vision. Current methodologies have shown preliminary results on very simple scenarios, but they are still far from human performance. Due to the large number of potential applications involving human gesture recognition in fields like surveillance [45], sign language recognition [80], or clinical assistance [67] among others, there is a large and active research community devoted to deal with this problem. Independently of the application field, the usual human gesture recognition pipeline is mainly formed by two steps: *gesture representation* and *gesture classification*.

In this chapter we introduce a Bag-of-Visual-and-Depth-Words (BoVDW) representation for gestures, as an extension of the Bag-of-Visual-Words (BoVW) that fuses information from multi-modal RGBD data streams. Regarding gesture classification, we address the problem of one-shot human gesture recognition, *i.e.* at training time just one gesture sample is available, for each gesture class we want to recognize.

In order to evaluate the presented approach, we compare the performances achieved with state-of-the-art RGB and depth feature descriptors separately, and combine them in a late fusion fashion. All the experiments are performed in the proposed framework using

the public data set provided by the ChaLearn Gesture Challenge[1]. Results of the proposed BoVDW method show better performance using late fusion in comparison to early fusion and standard BoVW model. Moreover, our BoVDW approach outperforms the baseline methodology provided by the ChaLearn Gesture Challenge 2012.

7.2 Related work

When it comes to representing gestures, literature shows a variety of methods that have obtained successful results. Commonly applied in image retrieval or image classification scenarios, *Bag-of-Visual-Words* (BoVW) is one of the most used approaches. This methodology is an evolution of *Bag-of-Words* (BoW) [60] representation, used in document analysis, where each document is represented using the frequency of appearance of each word in a dictionary. In the image domain, these words become visual elements of a defined visual vocabulary. First, each image is decomposed into a large set of patches, either using some type of spatial sampling (grids, sliding window, etc.) or detecting points with relevant properties (corners, salient regions, etc.). Each patch is then described obtaining a numeric descriptor. A set of V representative visual words are selected by means of a clustering process over the descriptors. Once the visual vocabulary is defined, each new image can be represented by a global histogram containing the frequencies of visual words. Finally, this histogram can be used as input for any classification technique (i.e. $k-$Nearest Neighbor or SVM) [29, 64]. In addition, extensions of BoVW from still images to image sequences have been recently proposed in the context of human action recognition, defining Spatio-Temporal-Visual-Words (STVW) [65].

As commented in chapters 1 and 4, the release of the Microsoft Kinect^{TM} sensor in late 2010 has allowed an easy and inexpensive access to almost synchronized range imaging with standard video data. Those data combine both sources into what is commonly named RGBD images (RGB plus Depth). This multi-modal data has reduced the burden of the first steps in many pipelines devoted to image or object segmentation, and opened new questions such as how these data can be effectively described and fused. Motivated by the information provided by depth maps, several 3-D descriptors have been recently developed [14, 75] (most of them based on codifying the distribution of normal vectors among regions in the 3D space), as well as their fusion with RGB data [56] and learning approaches for object recognition [13]. This depth information has been particularly exploited for gesture recognition and human body segmentation and tracking. While some works focus on just the hand regions for performing gesture recognition [12, 34, 54, 61, 66, 89], in [79] Shotton *et al.* introduced one of the greatest advances in the extraction of the human body pose using RGBD, which is provided as part of the Kinect^{TM} human recognition framework. The method is based on inferring pixel label probabilities through Random Forest from learned offsets of depth features. Then, mean shift is applied to estimate human joints and representing the body in skeletal form. In chapter 4 we extended Shotton *et al.*'s work applying Graph-cuts to the pixel label probabilities obtained through Random Forest, in order to compute consistent segmentations in the spatio-temporal domain. Girshick *et al.* [43] proposed later a different approach in which they directly regress the positions of the body joints, without the need of an intermediate pixel-wise body limb classification as in [79]. The extraction of body pose information opens the door to one of the most challenging problems nowadays, i.e. human gesture recognition.

Recently, many works have been proposed based on the skeletal models provided by RGB-D devices like Kinect^{TM}.Wu *et al.* [93] presents a method that combines the output

[1] http://gesture.chalearn.org/

of two different classifiers to perform gesture recognition. The first classifier is based on skeletal features whole the second receives audio features as input. Finally, in order to provide a global score for a video sequence to be labeled as a given gesture, the outputs of both classifiers are weighted. A similar approach is presented by Bayer *et al.* [11]: the authors perform model averaging by assigning different weights to classifiers over audio and skeleton-based features.

More similarly to the methodology we propose, other works in the state of the art combine different visual modalities like RGB images and depth maps (a.k.a. RGBD images) provided by Kinect sensors to perform gesture recognition. In [84] the authors propose a combination of Histograms of Oriented Gradients (HOG) features extracted from RGB and depth images, using as keypoints the joints from the retrieved skeleton model of the person. In [23] the authors concatenate HOG features extracted from the image regions around each hand with features extracted from the skeletal models retrieved by Kinect. In total, they fuse three different features: HOG for the hand regions, normalized 3-D coordinates of the skeleton keypoints, and pairwise skeleton distances among keypoints. In addition, the authors also propose a late fusion approach to combine the classification outputs of when concatenating different subsets of features, as well as the concatenation of all three features.

7.3 Bag of Visual Words

Bag of visual words is the extension of the bag of words representation, used to capture the semantics of the words in a text, to the image domain. The original bag of words representation is depicted in Fig. 7.1(a). Basically, it consists on a histogram that captures the appearance frequency of a set of words in a given *vocabulary*, over a whole document. While the ordering of the words in the text is lost in such histogram-based representation, it is expected to capture higher-level semantics: problems like spam filtering (*i.e.* classifying e-mails as spam or not), or newspaper article classification, have been successfully addressed using bag of words representation.

When extending the bag of words representation to the image domain, the concepts of visual words and visual vocabulary are introduced. In the text domain, we naturally detect words and identify them given a vocabulary, *e.g.* an English dictionary. Following the same idea, a visual word is defined as a small patch in an image, which can be identified in a given visual vocabulary or codebook, *i.e.* a set of visual words. In practice, defining the visual vocabulary is performed in an unsupervised fashion, following the next pipeline:

1. **Keypoint detection:** Given a set of images, a set of interest points are detected, *e.g.* Harris [46], SIFT [63], or a dense grid detector. These interest points are commonly parametrized as oriented circular patches of a given radius.
2. **Keypoint description:** Given a detected interest point and its corresponding image patch, a feature vector (*e.g.* SIFT [63] ,HOG [30]) is computed, and stored for further processing.
3. **Codebook generation:** In order to define the visual vocabulary or codebook, a quantization step is performed over the feature vectors, previously extracted from a set of images. One simple yet commonly used method for codebook generation is k-means clustering: given a number of visual words k and a set of feature vectors, k-means partitions the feature space into k clusters or visual words.

Once the visual codebook is generated, an image can be represented as a histogram of visual words by first detecting a set of interest points and extracting the same feature vectors used for the generation of the codebook. Then, each interest point can be identified

as a specific visual word in the vocabulary by finding the closest centroid among those previosly returned by k-means at the time of codebook generation. Finally, a histogram of visual words is computed by accumulating the quantized feature vectors extracted from the detected interest points (see Fig. 7.1(b)).

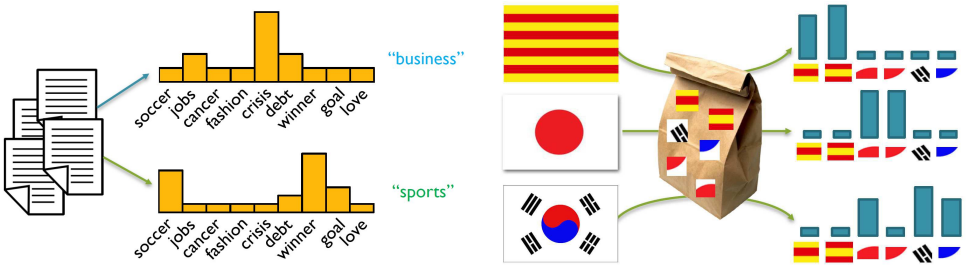


Figure 7.1: (a) An example of the bag of words representation for text classification. (b) Bag of visual words representation for image categorization.

7.4 Bag of Visual and Depth Words

In this section, the BoVDW approach for human gesture representation is introduced. Figure 7.2 contains a conceptual scheme of the approach. In this figure, we can see the idea of fusing RGB and Depth image modalities, and circles representing the spatio-temporal interest points described by means of the proposed novel VFHCRH descriptor.

7.4.1 Keypoint detection

The first step of BoW-based models consists of selecting a set of points in the image/video with relevant properties. In order to reduce the amount of points in a dense spatio-temporal sampling, the Spatio-Temporal Interest Point (STIP) detector [57] is used, which is an extension of the well-known Harris detector [46] in the temporal dimension. The STIP detector first computes the second-moment 3×3 matrix η of first order spatial and temporal derivatives. Finally, the detector searches regions in the image with significant eigenvalues λ_1,λ_2,λ_3 of η, combining the determinant and the trace of η,

$$H = |\eta| - K \cdot T_r(\eta)^3, \tag{7.1}$$

where $|.|$ corresponds to the determinant, $T_r(.)$ computes the trace, and K stands for a relative importance constant factor. As multi-modal RGBD data is employed, the STIP detector is applied separately on the RGB and Depth volumes, so two sets of interest points S_{RGB} and S_D are obtained.

7.4.2 Keypoint description

In this step, the interest points detected in the previous step should be described. On one hand, state-of-the-art RGB descriptors are computed for S_{RGB}, including Histogram of Gradients (HOG) [30], Histogram of Optical Flow (HOF), and their concatenation HOG/HOF [58]. On the other hand, a new descriptor VFHCRH (Viewpoint Feature Histogram Camera Roll Histogram) is introduced for S_D, as detailed below.

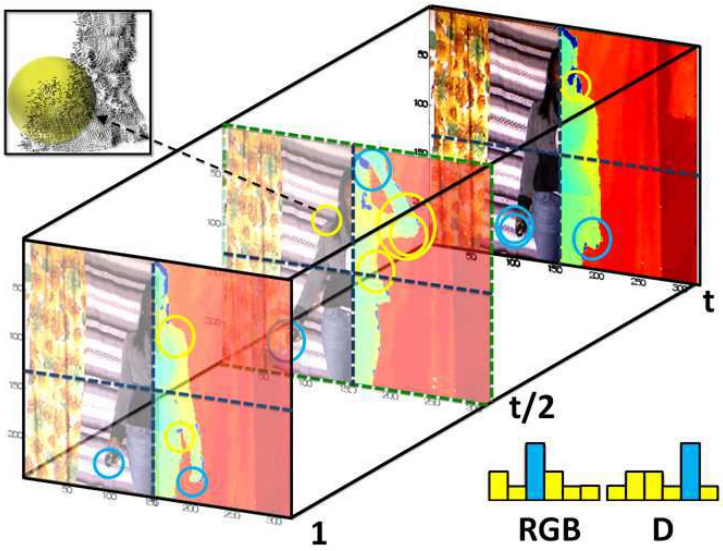


Figure 7.2: BoVDW approach in a Human Gesture Recognition scenario. Interest points in RGB and depth images are depicted as circles. Circles indicate the assignment to a visual word in the shown histogram – computed over one spatio-temporal bin. Limits of the bins from the spatio-temporal pyramids decomposition are represented by dashed lines in blue and green, respectively. A detailed view of the normals of the depth image is shown in the upper-left corner.

VFHCRH

The recently proposed Point Feature Histogram (PFH) and Fast Point Feature Histogram (FPFH) descriptors [14] represent each instance in the 3-D cloud of points with a histogram encoding the distribution of the mean curvature around it. Both PFH and FPFH provide $\mathcal{P}6$ DOF (Degrees of Freedom) pose invariant histograms, being $\mathcal{P}$ the number of points in the cloud. Following their principles, Viewpoint Feature Histogram (VFH)[75] describes each cloud of points with one descriptor of 308 bins, variant to object rotation around pitch and yaw axis. However, VFH is invariant to rotation about the roll axis of the camera. In contrast, Clustered Viewpoint Feature Histogram (CVFH) [8] describes each cloud of points using a different number of descriptors r, where r is the number of stable regions found on the cloud. Each stable region is described using a non-normalized VFH histogram and a Camera's Roll Histogram (CRH), and the final object description includes all region descriptors. CRH is computed by projecting the normal of the point cloud $\tau^{(i)}$ of the i-th point $\rho^{(i)}$ onto a plane P_{xy} that is orthogonal to the viewing axis z, the vector between the camera center and the centroid of the cloud, under orthographic projection,

$$\tau_{xy}^{(i)} = ||\tau^{(i)}|| \cdot \sin(\phi), \tag{7.2}$$

where ϕ is the angle between the normal $\tau^{(i)}$ and the viewing axis. Finally, the histogram encodes the frequencies of the projected angle ψ between $\tau_{xy}^{(i)}$ and y-axis, the vertical vector of the camera plane (see Fig. 7.3(a)).

In order to avoid descriptors of arbitrary lengths for different point clouds, the whole

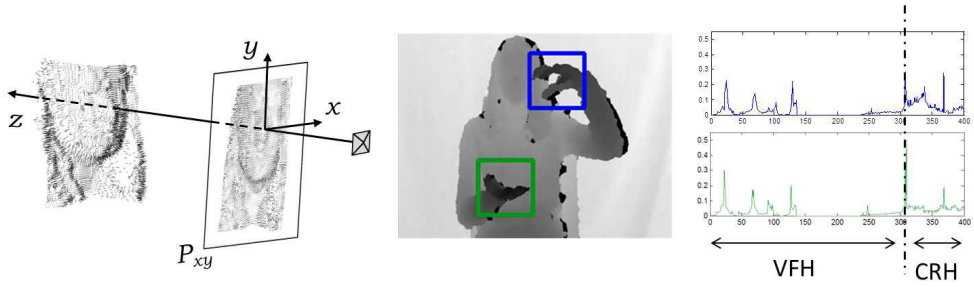


Figure 7.3: (a) Point cloud of a face and the projection of its normal vectors onto the plane P_{xy}, orthogonal to the viewing axis z. (b) VFHCRH descriptor: Concatenation of VFH and CRH histograms resulting in 400 total bins

cloud is described using VFH. In addition, a 92 bins CRH is computed for encoding 6 DOF information. The concatenation of both histograms results in the proposed VFHCRH descriptor of 400 bins shown in Figure 7.3(b). Note how the first 308 bins of the concatenated feature vector correspond to the VFH, that encode the normals of the point cloud. Finally, the remaining bins corresponding to the CRH descriptor, encode the information of the relative orientation of the point cloud to the camera.

7.4.3 BoVDW histogram

Once all the detected points have been described, the vocabulary of V visual/depth words is designed by applying a clustering method over all the descriptors. Hence, the clustering method –k-means in our case– defines the words from which a query video sequence will be represented, shaped like a histogram h that counts the appearance frequency of each word. Additionally, in order to introduce geometrical and temporal information, spatio-temporal pyramids are applied, following the work of Lazebnik *et al.* [59]. Basically, spatio-temporal pyramids consist of dividing the video volume in b_x, b_y, and b_t bins along the spatial x and v dimensions, and the temporal dimension t of the volume. Then, $b_x \times b_y \times b_t$ separate histograms are computed with the points lying in each one of these bins, and they are concatenated jointly with the general histogram computed using all points.

These histograms define the model for a certain class of the problem –in our case, a certain gesture. Since multi-modal data is considered, different vocabularies are defined for the RGB-based descriptors and the depth-based ones, and the corresponding histograms, h^{RGB} and h^{D}, are obtained. Finally, the information given by the different modalities is merged in the next and final classification step, hence using *late fusion*.

7.4.4 BoVDW-based classification

The final step of the BoVDW approach consists of predicting the class of a query video. For that, any kind of multi-class supervised learning technique could be used. In our case, a simple k-Nearest Neighbour classification is used, computing the complementary of the

histogram intersection as a distance,

$$d^F = 1 - \sum_i \min(h^F_{model}(i), h^F_{query}(i)), \tag{7.3}$$

where $F \in \{RGB, D\}$. Finally, in order to merge the histograms h^{RGB} and h^D, the distances d^{RGB} and d^D are computed separately, as well as the weighted sum,

$$d_{hist} = (1 - \beta)d^{RGB} + \beta d^D, \tag{7.4}$$

to perform late fusion, where β is a weighting factor.

7.5 Experiments

In this section we evaluate the proposed BoVDW framework for one-shot gesture recognition, comparing different state-of-the-art descriptors in both RGB and depth image modalities. We first present the dataset used in the performed experiments, provided by the ChaLearn Gesture Challenge organization [22]. Next, parameters and evaluation measures are discussed, and finally we show quantitative results of the proposed method.

7.5.1 Data

For the experimental evaluation, we used the ChaLearn [22] data set, provided by the CVPR2012 Workshop's challenge on Human Gesture Recognition. The data set consists of 50,000 gestures each one portraying a single user in front of a fixed camera. The images are captured by a Kinect device providing both RGB and depth images. A subset of the whole data set has been considered, formed by 20 development batches with a manually tagged gesture segmentation. Each batch includes 100 recorded gestures grouped in sequences of 1 to 5 gestures performed by the same user. The gestures from each batch are drawn from a different lexicon of 8 to 15 unique gestures and just one training sample per gesture is provided. These lexicons are categorized in nine classes, including: (1) body language gestures (scratching your head, crossing your arms, etc.), (2) gesticulations performed to accompany speech, (3) illustrators (like Italian gestures), (4) emblems (like Indian Mudras), (5) signs (from sign languages for the deaf), (6) signals (diving signals, mashalling signals to guide machinery or vehicle, etc.), (7) actions (like drinking or writing), (8) pantomimes (gestures made to mimic actions), and (9) dance postures.

7.5.2 Methods and validation

In all the experiments shown in this section, the vocabulary size was set to $V = 200$ words for both RGB and depth cases. For the spatio-temporal pyramids, the volume was divided in $2 \times 2 \times 2$ bins (resulting in a final histogram of 1800 bins). Since the nature of our application problem is one-shot learning (only one training sample is available for each class), a simple Nearest Neighbor classification is employed. Finally, for the late fusion, the weight $\beta = 0.8$ was empirically set, by testing the performance of our method in a small subset of development batches from the dataset. We observed that when increasing β, starting from $\beta = 0$, the performance keeps increasing in a linear fashion, until the value $\beta = 0.45$. From $\beta = 0.45$ to $\beta = 0.8$ the performance keeps improving more slowly, and finally, from $\beta = 0.8$ to $\beta = 1$ the performance drops again.

For the evaluation of the methods, in the context of Human Gesture Recognition, the Levenshtein distance or edit distance was considered. This edit distance between two strings

Table 7.1: Mean Levenshtein distance for RGB and depth descriptors.

RGB desc.	MLD	Depth desc.	MLD
HOG	0.3452	VFH	0.4021
HOF	0.4144	VFHCRH	**0.3064**
HOGHOF	**0.3314**		

is defined as the minimum number of operations (insertions, substitutions or deletions) needed to transform one string into the other. In our case, strings contain the gesture class labels detected in a video sequence. For all the comparison, the Mean Levenshtein distance (MLD) was computed over all sequences and batches.

7.5.3 Results

Table 7.1 shows a comparison between different state-of-the-art RGB and depth descriptors (including our proposed VFHCRH), using our BoVDW approach. Moreover, we compare our BoVDW framework with the baseline methodology provided by the ChaLearn 2012 Gesture Recognition challenge. This baseline first computes an average movement image for each gesture class by computing the mean value for each pixel in the image, along the time dimension. Then, the classification of a new gesture is basically performed by means of a Nearest Neighbor classifier computing the Euclidean distance between the average movement image from the query gesture and the templates in the training database. This baseline obtains a MLD of 0.5096. Table 7.2 shows the results in all the 20 development batches separately.

When using our BoVDW approach, in the case of RGB descriptors, HOF alone performs the worst. In contrast, the early concatenation of HOF to HOG descriptor outperforms the simple HOG. Thus, HOF contributes adding discriminative information to HOG. In a similar way, looking at the depth descriptors, it can be seen how the concatenation of the CRH to the VFH descriptor clearly improves the performance compared to the simpler VFH. When using late fusion in order to merge information from the best RGB and depth descriptors (HOGHOF and VFHCRH, respectively), a value of 0.2714 for MLD is achieved. Figure 7.4 shows the confusion matrices of the gesture recognition results with this late fusion configuration. In general, the confusion matrices follow an almost diagonal shape, indicating that the majority of the gestures are well classified. However, the results of batches 3, 16, 18, 19 are significantly worse, possibly due to the static characteristics of the gestures in these batches. Furthermore, late fusion was also applied in a 3-fold way, merging HOG, HOF, and VFHCRH descriptors separately. In this case the weight β was assigned to HOG and VFHCRH descriptors (and $1-\beta$ to HOF), improving the MLD to 0.2662. From this result it can be concluded that HOGHOF late fusion performs better than HOGHOF early fusion.

7.6 Discussion

We presented the BoVDW approach for Human Gesture Recognition, using multi-modal RGBD video sequences. A new depth descriptor VFHCRH has been proposed, which outperforms simpler VFH, in the task of gesture recognition. Moreover, the effect of the late fusion has been analyzed for the combination of RGB and depth descriptors in the BoVDW, obtaining better performance in comparison to early fusion.

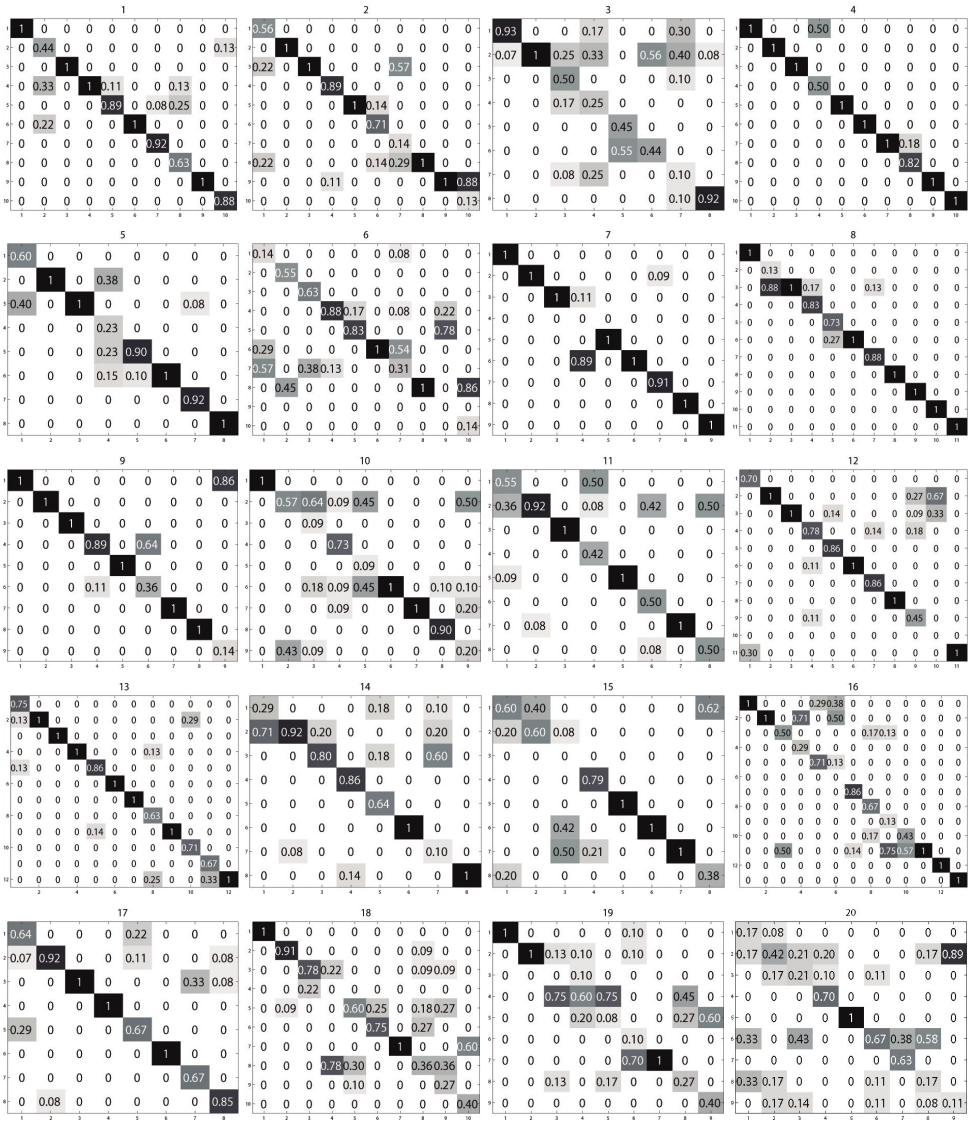


Figure 7.4: Confusion matrices for gesture recognition in each one of the 20 development batches.

Among the different RGB descriptors included in the comparison, we showed that the concatenation of HOG and HOF is the one yielding the best results. Moreover, we show how the concatenation of VFH and CRH descriptors outperform VFH in the case of depth descriptors. Generally speaking, the best scoring depth descriptor (VFHCRH) by itself obtains better results than the best one from the RGB stream (HOGHOF). Hence, when using late fusion to merge information from both RGB and Depth video streams, we not

Table 7.2: Mean Levenshtein Distance of the best RGB and depth descriptors separately, as well as the 2-fold and 3-fold late fusion of them. Results obtained by the baseline from the ChaLearn challenge are also shown. Rows 1 to 20 represent the different batches.

	HOGHOF	**VFHCRH**	**2-fold L.F.**	**3-fold L.F.**	**Baseline**
1	0.19	0.17	**0.12**	0.20	0.42
2	**0.24**	0.30	**0.24**	0.26	0.57
3	0.76	**0.39**	0.40	0.49	0.78
4	0.14	**0.08**	**0.08**	0.11	0.32
5	**0.08**	0.33	0.17	0.17	0.25
6	0.41	0.47	0.44	**0.34**	0.54
7	**0.10**	0.18	0.11	0.13	0.64
8	0.12	0.26	0.14	**0.08**	0.40
9	**0.11**	0.18	0.15	0.13	0.30
10	0.57	0.40	**0.39**	0.46	0.79
11	0.47	0.36	**0.27**	0.34	0.54
12	0.37	0.20	0.21	**0.17**	0.42
13	0.16	0.14	0.10	**0.09**	0.34
14	0.41	0.34	**0.30**	**0.30**	0.69
15	0.38	**0.28**	0.34	**0.28**	0.54
16	**0.22**	0.41	0.34	0.29	0.42
17	0.38	0.16	**0.15**	0.17	0.55
18	**0.38**	0.43	0.40	**0.38**	0.53
19	0.67	0.50	0.50	**0.44**	0.61
20	**0.46**	0.57	0.56	0.48	0.52
Mean	0.3314	0.3064	0.2714	0.2662	0.5096

surprisingly discover that the best value for β favors depth words in front of those from the RGB vocabulary. Not only that, but we also show that building different vocabularies for HOG and HOF separately and merging their information at the histogram level (late fusion) yields better results than building just one vocabulary over the concatenation of both descriptors (early fusion).

An additional advantage of using late fusion w.r.t early fusion is the flexibility of the model upon possible extensions. Adding an additional set of features to an existing model in an early fusion fashion, implies to generate a new codebook or visual vocabulary, and therefore, re-train the whole system from blank. In contrast, by keeping different vocabularies for each feature we want to include in the system, *i.e.* when using late fusion, we are able to add new features to the by just computing a new vocabulary while the vocabularies for the features already computed remain the same. In addition, we can define different cardinalities for each visual vocabulary, allowing for a more flexible formulation of the model.

As further research on the topic, it would be interesting to study the behavior of late fusion when adding more descriptors to the framework, and studying ways to optimize the corresponding weights of each visual vocabulary.

Chapter 8

PDTW for continuous gesture recognition

8.1 Introduction

In the previous chapter we presented a method for representing and classifying gestures, by computing a bag-of-visual-and-depth-words descriptor over a spatio-temporal volume corresponding to an RGBD multimodal video sequence, where a person standing in front of the camera is performing one gesture. While we show successful results in gesture recognition with this methodology, we assume that we know the beginning and the end time frames that delimit the gestures performed by the actor. However, in many computer vision applications related to gesture recognition, we do not have such beginning/end labels available, so prior temporal segmentation of the gestures should be performed in order to obtain the spatio-temporal volumes containing isolated gestures. Then, the corresponding bag-of-visual-and-depth-words feature vectors can be computed for each volume, and final classification is performed independently for each isolated gesture.

On the other hand, other methods try to directly recognize gestures in a continuous input stream, *i.e.* continuous gesture recognition. The problem of continuous gesture recognition is addressed in this chapter: we introduce an extension of the DTW method to a probability-based framework able to capture the intra-class variability of each gesture class. In the experimental section, we show quantitative and qualitative results on the ChaLearn Gesture Challenge dataset [22]. We apply the proposed method to recognize the "idle" or reference gesture, performed by the actors between the end of a gesture and the beginning of the next one, *i.e.* we perform temporal gesture segmentation.

8.2 Related work

Methods based on dynamic programming algorithms for both alignment and clustering of temporal series [98] are commonly used for continuous gesture recognition. In addition, other probabilistic methods such as Hidden Markov Models (HMM) or Conditional Random Fields (CRF) have been also commonly used in the literature [80]. Nevertheless, one of the most common methods for continuous human gesture recognition is Dynamic Time Warping (DTW) [9, 72], since it offers a simple yet effective temporal alignment between sequences of different lengths. However, the application of such methods to gesture detection

in complex scenarios becomes a hard task due to the high variability of the environmental conditions among different domains. Some common problems are: wide range of human pose configurations, influence of background, continuity of human movements, spontaneity of human actions, speed, appearance of unexpected objects, illumination changes, partial occlusions, or different points of view, just to mention a few. These effects can cause dramatic changes in the description of a certain gesture, generating a great intra-class variability. In this sense, since usual DTW is applied between a sequence and a single pattern, it fails when taking into account such variability.

8.3 Dynamic Time Warping

The original DTW is introduced in this section, as well as its common extension to detect a certain sequence given an indefinite data stream.

The original DTW algorithm was defined to match temporal distortions between two models, finding an alignment/warping path between two time series: an input model $Q = \{q_1, .., q_n\}$ and a certain sequence $C = \{c_1, .., c_m\}$. In our particular case, the time series Q and C are video sequences, where each q_j and c_i will be feature vectors describing the $j-$th and $i-$th frame respectively. In this sense, Q will be an input video sequence and C will be the model gesture we are aiming to detect. Generally, in order to align these two sequences, a $M_{m \times n}$ matrix is designed, where position (i, j) of the matrix contains the alignment cost between c_i and q_j. Then, a warping path of length τ is defined as a set of contiguous matrix elements, defining a mapping between C and Q: $W = \{w_1, .., w_\tau\}$, where w_i indexes a position in the cost matrix M. This warping path is typically subject to several constraints,

- **Boundary conditions:** $w_1 = (1, 1)$ and $w_\tau = (m, n)$.
- **Continuity and monotonicity:** Given $w_{\tau'-1} = (a', b')$, $w_{\tau'} = (a, b)$, then $a - a' \leq 1$ and $b - b' \leq 1$. This condition forces the points in the cost matrix with the warping path W to be monotonically spaced in time.

Interest is focused on the final warping path that, satisfying these conditions, minimizes the warping cost,

$$DTW(M) = \min_W \left\{ \frac{M(w_\tau)}{\tau} \right\}, \tag{8.1}$$

where τ compensates the different lengths of the warping paths at each time t. This path can be found very efficiently using dynamic programming. The cost at a certain position $M(i, j)$ can be found as the composition of the Euclidean distance $d(i, j)$ between the feature vectors c_i and q_j of the two time series, and the minimum cost of the adjacent elements of the cost matrix up to that position, as,

$$M(i, j) = d(i, j) + \min\{M(i - 1, j - 1), M(i - 1, j), M(i, j - 1)\}. \tag{8.2}$$

However, given the streaming nature of our problem, the input video sequence Q has no definite length (it may be an infinite video sequence) and may contain several occurrences of the gesture sequence C. In this sense, the system considers that there is correspondence between the current block k in Q and the gesture when the following condition is satisfied, $M(m, k) < \theta, k \in [1, .., \infty]$ for a given cost threshold θ. At this point, if $M(m, k) < \theta$ k is consider a possible end of a gesture sequence C.

Once detected a possible end of the gesture sequence, the warping path W can be found through backtracking the minimum cost path from $M(m, k)$ to $M(0, g)$, being g the instant of time in Q where the detected gesture begins. Note that $d(i, j)$ is the cost function which

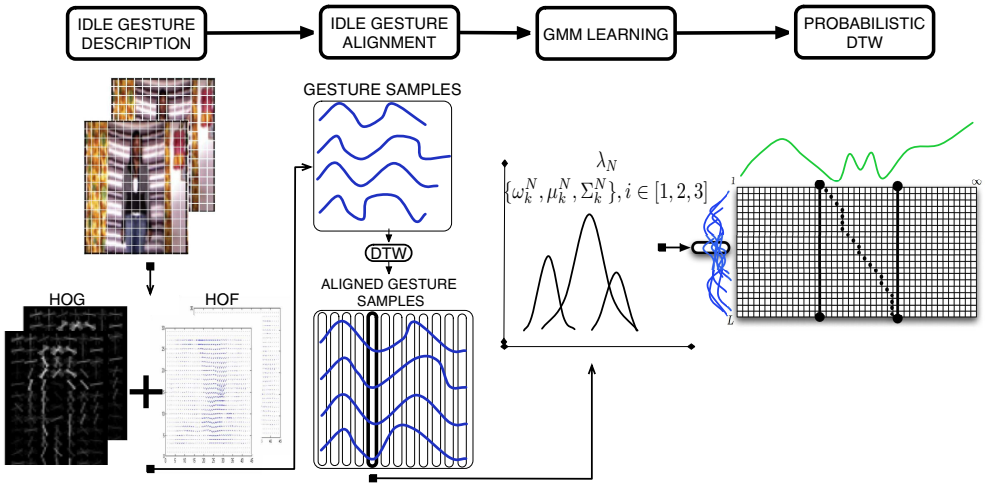


Figure 8.1: Flowchart of the Probabilistic DTW gesture segmentation methodology.

measures the difference among descriptors c_i and q_j, which in standard DTW is defined as the euclidean distance between c_i and q_j. An example of a begin-end gesture recognition together with the warping path estimation is shown in Figure 8.1 (last 2 steps: GMM learning and Probabilistic DTW).

8.4 Handling variance with Probability-based DTW

In this section, DTW is extended in order to align patterns taking into account the probability density function (PDF) of each element of the sequence by means of a Gaussian Mixture Model (GMM). A flowchart of the whole methodology is shown in Figure 8.1.

Consider a training set of N sequences, $S = \{S_1, \ldots, S_g, \ldots, S_N\}$, that is, N gesture samples belonging to the same gesture category. Then, each sequence $S_g = \{s_1^g, \ldots, s_t^g, \ldots, s_{L_g}^g\}$, (each gesture sample) is composed by a feature vector [1] for each frame t, denoted as s_t^g, where L_g is the length in frames of sequence S_g. In order to avoid temporal deformations of the gesture samples in S, all sequences are aligned with the median length sequence using the classical DTW with Euclidean distance. Let us assume that sequences are ordered according to their length, so that $L_{g-1} \leq L_g \leq L_{g+1}, \forall g \in [2, .., N-1]$, then, the median length sequence is $\bar{S} = S_{\lceil \frac{N}{2} \rceil}$.

It is worth noting that this alignment step by using DTW has no relation to the actual gesture recognition, as it is considered a pre-processing step to obtain a set of gesture samples with few temporal deformations and a matching length.

Finally, after this alignment process, all sequences have length $L_{\lceil \frac{N}{2} \rceil}$. The set of warped sequences is defined as $\tilde{S} = \{\tilde{S}_1, \tilde{S}_2, \ldots, \tilde{S}_N\}$ (See Figure 8.2(b)). Once all samples are aligned, the N feature vectors corresponding to each sequence element at a certain frame t, denoted as $\widetilde{F}_t = \{f_t^1, f_t^2, ..., f_t^N\}$, are modelled by means of a $G-$component Gaussian Mixture Model (GMM) $\lambda_t = \{\alpha_k^t, \mu_k^t, \Sigma_k^t\}, \quad k = 1, \ldots, G$, where α_k^t is the mixing value,

[1] HOG/HOF descriptors in our particular case, see Sec. 8.5 for further details.

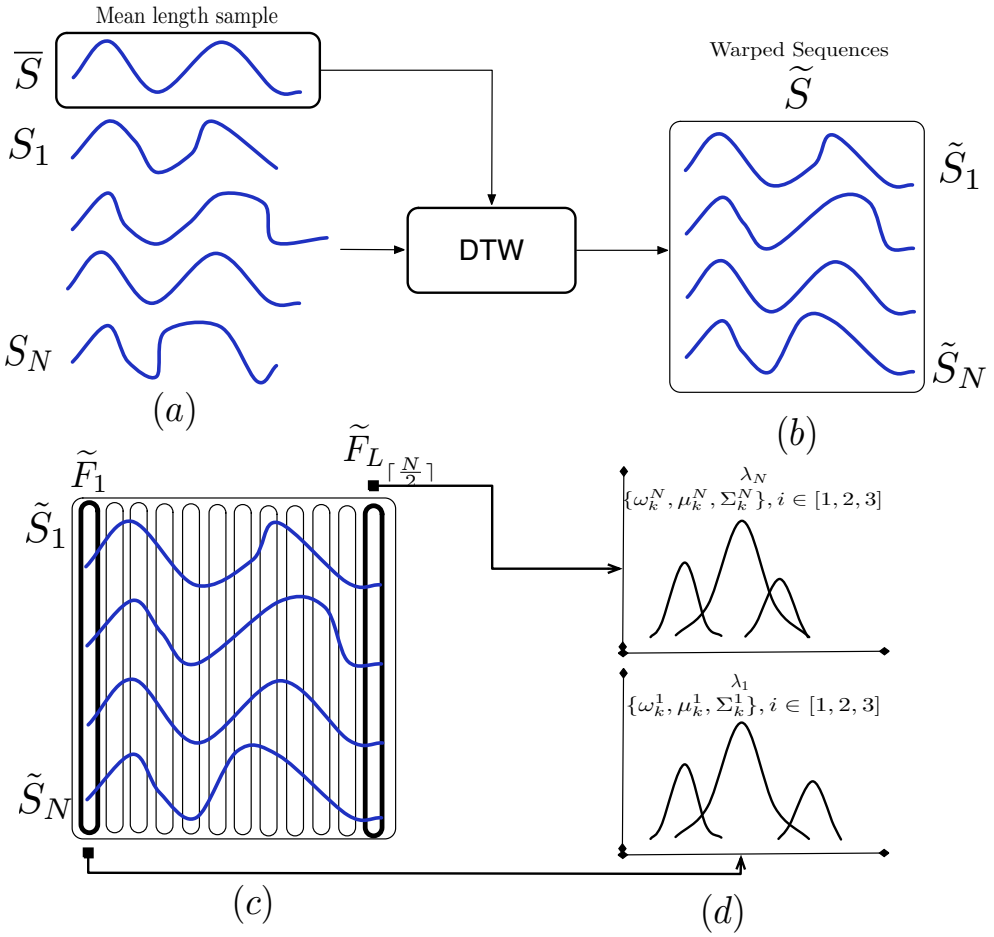


Figure 8.2: (a) Different sequences of a certain gesture category and the median length sequence. (b) Alignment of all sequences with the median length sequence by means of Euclidean DTW. (c) Warped sequences set $\tilde{S}$ from which each set of t-th elements among all sequences are modelled. (d) Gaussian Mixture Model learning with 3 components.

and μ_k^t and Σ_k^t are the parameters of each of the G Gaussian models in the mixture. As a result, each one of the GMMs that model each $\widetilde{F}_t$ is defined as follows,

$$p(\widetilde{F}_t) = \sum_{k=1}^{G} \alpha_k^t \cdot e^{-\frac{1}{2}(x-\mu_k^t)^T \cdot (\Sigma_k^t)^{-1} \cdot (x-\mu_k^t)}. \tag{8.3}$$

The resulting model is composed by the set of GMMs that model each set $\widetilde{F}_t$ among all warped sequences of a certain gesture class. An example of the process is shown in Figure 8.2.

8.4.1 Distance measures

In the classical DTW, a pattern and a sequence are aligned using a distance metric, such as the Euclidean distance. However, since our gesture samples are modelled by means of probabilistic models, in order to use the principles of DTW, the distance must be redefined. In this sense, a soft-distance based on the probability of a point x belonging to each one of the G components in the GMM is considered, i.e. the posterior probability of x is obtained according to Eq. (8.3). Therefore, since $\sum_{k=1}^{G} \alpha_k^t = 1$, the probability of a element $q_j \in Q$ belonging to the whole GMM λ_t can be computed as,

$$P(q_j, \lambda_t) = \sum_{k=1}^{G} \alpha_k^t \cdot P(q_j)_k, \tag{8.4}$$

$$P(q_j)_k = e^{-\frac{1}{2}(q_j - \mu_k^t)^T \cdot (\Sigma_k^t)^{-1} \cdot (q_j - \mu_k^t)}, \tag{8.5}$$

which is the sum of the weighted probability of each component. Nevertheless, an additional step is required since the standard DTW algorithm is conceived for distances instead of similarity measures. In this sense, a soft-distance based measure of the probability is used, which is defined as,

$$D(q_j, \lambda_t) = \exp^{-P(q_j, \lambda_t)}. \tag{8.6}$$

In conclusion, possible temporal deformations of different samples of the same gesture category are taken into account by aligning the set of N gesture samples with the median length sequence. In addition, by modelling each set of feature vectors which compose the resulting warped sequences with a GMM, we obtain a methodology for gesture detection that is able to deal with multiple deformations in gestures both temporal (which are modelled by the DTW alignment), or descriptive (which are learned by the GMM modelling). The algorithm that summarizes the use of the probability-based DTW to detect start-end of gesture categories is shown in Table 10. Figure 8.3 illustrates the application of the algorithm in a toy problem.

8.5 Experiments

In this section we provide with quantitative and qualitative evaluation of the proposed probabilistic dynamic time warping methodology, in the context of continuous gesture recognition. More specifically, we aim to detect an "idle" or resting gesture performed by the actors in the ChaLearn Gesture Challenge dataset, between labeled gestures in the video sequences. Hence, detecting the "idle" gesture between other gestures of interest can be understood as gesture segmentation. We start by presenting the data we used for the experiments, then we detail the experimental settings, and finally present quantitative and qualitative results for the proposed gesture segmentation application.

8.5.1 Data

For the temporal gesture segmentation experiments we used the 20 development batches provided by the organization of the ChaLearn Gesture Challenge 2012 [22], as in the previous chapter of this thesis. These batches contain a manual labelling of gesture start and end points. Each batch includes 100 recorded gestures, grouped in sequences of 1 to 5 gestures performed by the same user. For each sequence the actor performs an idle gesture between each gesture of the gestures drawn from lexicons. Finally, this means that we have a set of approximately $1,800$ idle gesture samples.

Algorithm 10 Probability-based DTW algorithm.

Input: A set of GMM models $\lambda = \{\lambda_1, .., \lambda_m\}$ corresponding to a gesture category, a threshold value θ, and the streaming sequence $Q = \{q_1, .., q_\infty\}$. Cost matrix $M_{m \times \infty}$ is defined, where $\mathcal{N}(x), x = (i, t)$ is the set of three upper-left neighbor locations of x in M.

Output: Warping path W of the detected gesture, if any.

for $i = 1 : m$ **do**
  for $j = 1 : \infty$ **do**
    $M(i, j) = \infty$
  end for
end for
for $j = 1 : \infty$ **do**
  $M(0, j) = 0$

end for
for $j = 0 : \infty$ **do**
  for $i = 1 : m$ **do**
    $x = (i, j)$
    $M(x) = D(q_j, \lambda_i) + \min_{x' \in \mathcal{N}(x)} M(x')$
  end for
  if $M(m, j) < \theta$ **then**
    $W = \{\operatorname{argmin}_{x' \in \mathcal{N}(x)} M(x')\}$
    return
  end if
end for

8.5.2 Methods and validation

Each video sequence of each batch was described using a 20×20 grid approach. For each patch in the grid we obtain a 208 feature vector consisting of HOG (128 dimensions) and HOF (80 dimensions) descriptors which are finally concatenated in a full image (posture descriptor). Due to the huge dimensionality of the descriptor of a single frame ($83,200$ dimensions), we used a Random Projection to reduce dimensionality to 150 dimensions.

For both of the DTW approaches the cost-threshold value θ is estimated in advance using ten-fold cross-validation strategy on the set of 1800 idle gesture samples. This involves using 180 idle gestures as the validation data, and the remaining observations as the training data. This is repeated such that each observation in the sample is used once as the validation data. Finally, the threshold value θ chosen is the one associated with the largest overlapping performance. For the probabilistic DTW approach, each GMM was fit with $G = 4$ components. The value of G was obtained using a ten-fold cross-validation procedure on the set of 1800 idle gestures as well. In this sense, the cross-validation procedure for the probability-based DTW is a double loop (optimizing on the number of GMM components G, and then, on the cost-threshold θ). In the HMM case, we used the Baum-Welch algorithm for training, and 3 states were experimentally set for the idle gesture, using a vocabulary of 60 symbols computed using K-means over the training data features. Final recognition is performed with temporal sliding windows of different wide sizes, based on the idle gesture

Table 8.1: *Overlapping* and *accuracy* results.

Method	Overlapping	Accuracy
Probability-based DTW	**0.3908± 0.0211**	**0.6781 ± 0.0239**
Euclidean DTW	0.3003 ± 0.0302	0.6043 ± 0.0321
HMM	0.2851 ± 0.0432	0.5328 ± 0.0519

samples length variability.

8.5.3 Results

Our probability-based DTW approach using the proposed distance D shown in Eq. (8.6) is compared to the usual DTW algorithm and the Hidden Markov Model approach. The evaluation measurements presented are *overlapping* and *accuracy* of the recognition for the idle gesture, considering that a gesture is correctly detected if overlapping in the idle gesture sub-sequence is greater than 60% (the standard overlapping value, computed as the intersection over the union between the temporal bounds in the ground truth, and the ones computed by our method). The accuracy is computed frame-wise as

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}, \tag{8.7}$$

being TP, TN, FP and FN the number of True Positives, True Negatives, False Positives and False Negatives, respectively.

The results of our proposal, HMM and the classical DTW algorithm are shown in Table 8.1. It can be seen how the proposed probability-based DTW outperforms the usual DTW and HMM algorithms in both experiments. Moreover, confidence intervals of DTW and HMM do not intersect with the probability-based DTW in any case. From this results it can be concluded that performing dynamic programming increases the generalization capability of the HMM approach, as well as a model defined by a set of GMMs outperforms the classical DTW on RGBD data without increasing the computational complexity of the method. Figure 8.3 shows qualitative results from two sample video sequences.

8.6 Discussion

We proposed an extension of the well-known Dynamic Time Warping (DTW) method for string matching, by defining a probability-based distance metric D between an element of a given input string and a probabilistic model learnt from training data. More precisely, we first align the training samples to the median length sequence, yielding a fixed length set of training sequences. Then, a different Gaussian Mixture Model is learnt for each element position in the aligned sequences, able to capture variability among the different training samples.

We applied the proposed Probabilistic-based DTW method to the problem of gesture temporal segmentation: given a video sequence where a person standing in front of the camera is performing several different gestures of interest, we aim for finding the beginning and end time instants of each of the performed gestures. We assume that the actor adopts an “idle” or resting position between each of the performed gestures, and we treat this “idle” position as a gesture we want to recognize.

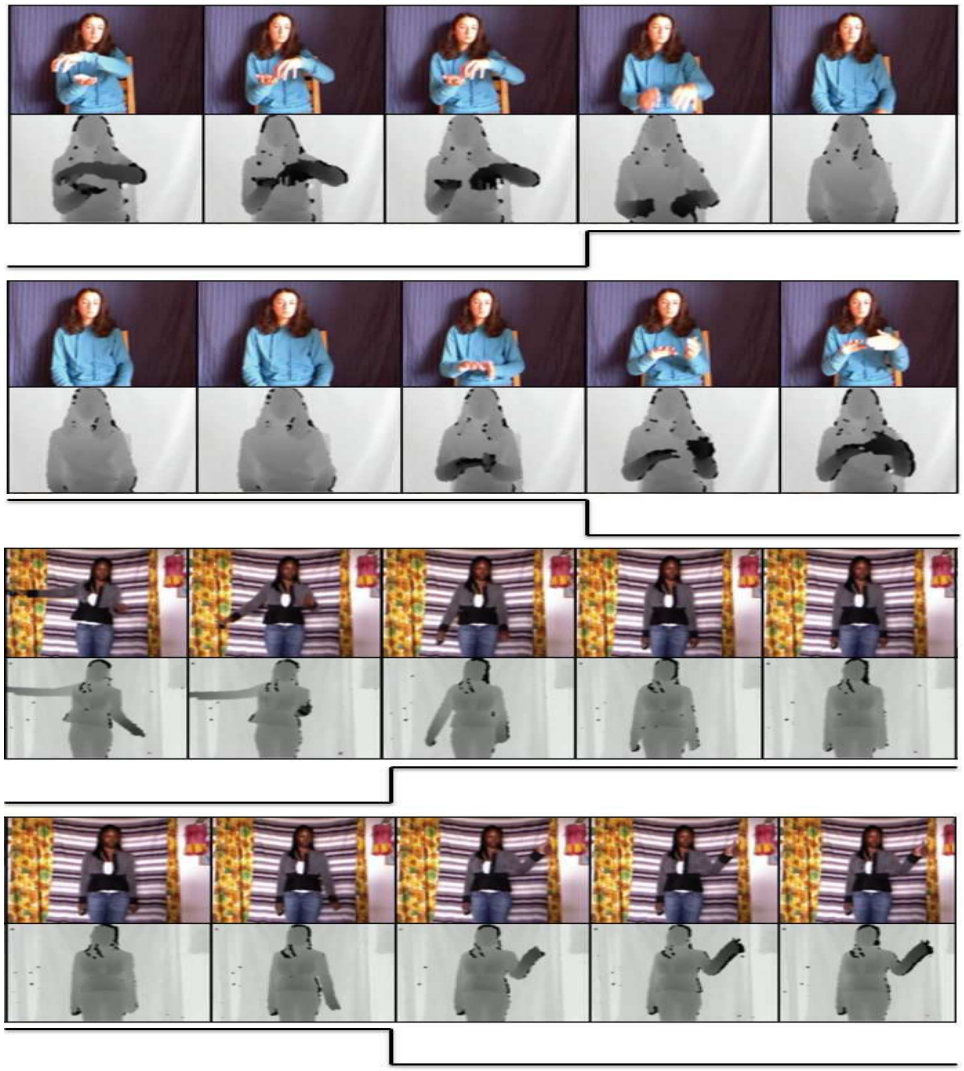


Figure 8.3: Examples of idle gesture detection on the Chalearn data set using the probability-based DTW approach. The line below each pair of depth and RGB images represents the detection of a idle gesture (step up: beginning of idle gesture, step down: end)

.

Experimental evaluation on the ChaLearn Gesture Challenge 2012 [22] showed that our proposed PDTW method outperforms basic Euclidean DTW and Hidden Markov Model (HMM) approaches for the task of gesture segmentation. Not only we obtain a better performance in terms of overlapping (intersection over the union of the found temporal segments), but also in terms of accuracy, at frame level.

As further experiments, it would be interesting to apply the proposed method for continuous recognition of more complex gestures, and test other feature extraction methods, like the body limb segmentation masks obtained in chapter 4, the skeletal representations of the human body from chapter 6, or the Bag-of-Visual-and-Depth-Words presented in chapter 7.

Part IV

Epilogue

Chapter 9

Conclusions

The visual analysis of humans from images is an important and interesting topic of interest due to its relevance to many computer vision applications like pedestrian detection, monitoring and surveillance, human-computer interaction, e-health or content-based image retrieval, among others. In this dissertation we have made contributions in three different research lines related to the visual analysis of humans: human segmentation, human pose estimation and gesture recognition.

9.1 Summary of contributions

Human segmentation

We have shown the power and adaptability of Graph cuts optimization, applied to binary and multi-label segmentation of humans in video sequences.

In the first case, an evolution of the semi-automatic GrabCut algorithm has been proposed for dealing with the problem of binary human segmentation in image sequences. Besides fully automatic initialization of the proposed ST-GrabCut system via person and face detectors, we have introduced spatial coherence via Mean Shift clustering, and temporal coherence has been considered based on the historical of Gaussian Mixture Models. The experiments carried out for validating the proposed system have shown higher performance in favor to ST-GrabCut w.r.t simple GrabCut, in terms of segmentation accuracy. More interestingly, we have shown how removing the background from scenes of people with the proposed segmentation methodology, contributes to obtain better results in higher-level human analysis techniques like face alignment or human pose estimation.

In the second case regarding multi-label segmentation of the human body, we have proposed a generic framework for object segmentation using Random Forest pixel classification on depth images, and using Graph cuts optimization to introduce spatial and temporal coherence and obtain much smoother segmentation results. We have applied the proposed methodology to obtain robust segmented regions corresponding to the different human body limbs. We have shown that including problem-specific asymmetric label boundary costs in the energy function to optimize can yield better results than the simple Potts model, while being forced to use α-β swap instead of α-expansion. The qualitative results have shown robust and spatially-coherent segmented regions of the human limbs, and can be useful for higher-level gesture recognition applications, such as sign language recognition.

Human pose estimation

We have presented a contextual rescoring methodology for predicting the position of body parts from a mid-level part representation based on Poselets, following the hypothesis that higher-level body parts are more discriminative than smaller ones, and provide more confident evidence.

We have proposed a method for the automatic discovery of high-precision poselets that cover the range of poses in a validation set of images, yielding a compact mid-level part representation. When using this mid-level part representation along with the presented contextual rescoring methodology, we achieve nearly comparable performance to that of Pishchulin *et al*. [69] (also using Poselets to predict the position of body parts), while reducing the number of poselets by 95%, resulting in a reduction of the execution time of 68.23%.

In experiments with benchmark datasets, the proposed contextual rescoring methodology powered by the mid-level part representation is especially beneficial for distinguishing between left and right body limbs, being able to correct from the "double-counting" phenomena that tree-structured graphical models suffer from.

Gesture recognition

We have presented a Bag-of-Visual-and-Depth-Words (BoVDW) approach for gesture representation and classification. The proposed bag-of-words-based representation benefits from RGB and depth image modalities available in multi-model RGBD data streams, by building separate visual vocabularies for each modality and combining them in a late fusion fashion. The problem of one-shot gesture recognition has been addressed in experiments that were designed to validate the proposed gesture representation with a simple classification methodology.

We have proposed a new depth descriptor VFHCRH, resulting from the concatenation of Viewpoint Feature Histogram (VFH) and Camera Roll Histogram (CRH) descriptors. We have shown that appending the CRH descriptor to VFH yields better results than simpler VFH for the task of gesture recognition, using the proposed BoVDW framework. More specifically, we have shown that VFHCRH introduces variance to rotations along the camera roll axs, w.r.t the simpler VFH descriptor. In the gesture recognition context, this variance is especially useful to distinguish between different orientations of body parts with the same shape.

We have analyzed the performance of different state-of-the-art descriptors in both RGB and depth modalities, as well as the effect of the late fusion for the combination of RGB and depth descriptors in the BoVDW framework, obtaining better performance in comparison to early fusion.

Finally, we have proposed an extension of the well-known Dynamic Time Warping (DTW) method for string matching, by defining a probability-based distance metric D between an element of a given input string and a probabilistic model learnt from training data. By learning a probabilistic model to compute the distances between the query string and the model string, we are able to capture variability of the data that simple DTW variants are not able to model.

9.2 Final conclusions

Among the different image modalities seen through the contributions proposed in this dissertation, we have seen that range imaging introduces some advantages w.r.t classical RGB images. First of all, range imaging provides a 2.5D projection of the real world, in contrast

to the 2D projections obtained with color images. Basically, a range image contains a depth value for each pixel, measuring the distance of that point in the scene from the capturing device sensor. This information has been proven to be very suitable for learning the shape of objects, and also provides useful cues for reasoning about occlusions. Secondly, range images are invariant to illumination given that no light reflections are captured, but just the geometrical cues of the scene. Depending on the application, the absence of color information may suppose an advantage or a disadvantage. For example, in order to detect pedestrians in a street, the absence of color in the feature representation is desirable, since pedestrians' color cues can be very different due to clothing. In contrast, detecting people wearing blue jeans in range images would not be possible. On the other hand, not all low-cost range image sensors work correctly outdoors due to infrared light interferences, although newer technologies (*e.g.* Microsoft's Kinect 2) are being developed so these limitations can be overcome.

In contrast, RGB images provide powerful cues for distinguishing objects of different colors and textures. However, changes in illumination and background clutter usually become important problems in almost all applications. On one hand, changes in illumination may produce undesired shadows that tend to confuse computer vision algorithms: brightness discontinuities in the image could arise due to real object boundaries or due to shadows in the scene, and it is challenging to resolve this ambiguity in general. On the other hand, background clutter can lead a computer vision method to confuse irrelevant background regions in the image with an object of interest.

In chapter 3, we presented a fully-automatic segmentation method based on Graph cuts optimization, producing a binary silhouette of a person appearing in a video sequence. While providing little information about the pose or the activities performed by the subject, this segmentation mask allows removing the distracting background clutter in the scene. We have shown how removing the background yields better results in baseline human pose estimation and face alignment methods, w.r.t to unsegmented images.

In addition, in chapter 4 we used the same Graph cuts optimization framework to produce robust finer-grained multi-label segmentation masks for segmenting the different limbs of the human body in range image sequences. In contrast to binary segmentation masks, these multi-label segmentation masks give a better idea of the body pose. Given the pixel-wise level of abstraction of such representation, it is possible to capture details like the finger configuration (as long as the image resolution allows for it); this could be especially advantageous in problems like sign language recognition, for example.

At a higher level of abstraction, skeletal models like pictorial structures introduced in chapter 5 are able to capture a coarse pose of subjects appearing in the scene. In practice, given the low resolution of the images in human pose estimation benchmarks, skeletal models are used to capture the position and orientation of the arms, legs, torso and head, but do not reach finer levels of detail like the fingers. Hence, this kind of representation could be useful for recognizing actions like runnning or jumping but not for sign language recognition, unless the skeletal model included the hands and fingers.

One of the main limitations such skeletal models suffer from is that their topology must follow a tree structure in order to allow for efficient exact inference, using dynamic programming. Consequently, such a loosely connected model is prone to confuse left and right body limbs ("double-counting" phenomena). For this reason, in chapter 6 we proposed a mid-level part representation based on Poselets, together with a contextual rescoring method providing with a top-down approach for predicting the position of the body parts in the model. In particular, this top-down approach could be potentially useful for recovering the position of self-occluded body parts (especially interesting for the hands and feet, being the leaf nodes of the underlying CRF), which could not be located via bottom-up detection. In addition, such a Poselet-based representation would be also very useful for providing a pixel-level likelihood

of regions belonging to body parts, and regions belonging to the background, being suitable for the trimap initialization step of the segmentation system proposed in chapter 3.

Finally, at the highest level of abstraction addressed in this dissertation, we consider the problem of gesture recognition. While in segmentation and human pose estimation methods the use of temporal information is optional, in the case of gesture recognition it is mandatory by definition, since gestures take place in the spatio-temporal domain. However, as we have shown in the presented contributions on human segmentation, considering the temporal domain enables inference algorithms to enforce some temporal consistency on the results, yielding a significant boost in performance.

However, considering the temporal domain introduces other problems like possible time deformities. In gesture recognition, this is one of the most important problems to take into account, since the same gesture can be performed at different speeds, yielding video sequences of different lengths. In chapter 7 we proposed a bag-of-visual-and-depth-words visual representation of gestures, introducing two main advantages. On one hand, the proposed representation has a fixed length independently from the length of the considered video sequence, so time deformities among different gesture samples are directly addressed via the proposed feature representation. On the other hand, it provides a simple yet effective way to combine RGB and Depth image modalities, hence taking advantage of RGB descriptors relying on textured images (like HOF) and shape descriptors computed from depth images (like VFHCRH).

Other well-known methods like Dynamic Time Warping (DTW) are able to match sequences of different lengths, thus allowing for time deformations. Therefore, gesture recognition can be formulated as matching an input video sequence with a model sequence of the gesture of interest. Note that a gesture performed by a human may not only suffer temporal variations, but also spatial ones: gestures can be produced in different regions of the space in front of the subject, and also the gestures can be produced with different range/scale of motion, but still be recognizable by a human. Although DTW is able to tackle temporal warping of the sequences, it is not well-suited to capture possible spatial variations in the feature space, given that the gesture model is composed by just one sequence. In order to overcome this problem, we proposed a probabilistic dynamic time warping, where the model is defined as a probability density function. This way, both temporal and feature-related variabilities among gesture sampels can be successfully modeled.

9.3 Future work

We now briefly describe possible future lines of reserch derived from the work presented in this dissertation.

Human segmentation

In chapter 3 we showed how segmentation can help to human pose estimation methods by removing the background clutter in the scenes. In this sense, it would be also interesting to see if human pose estimation methods can also contributes to obtain more accurate segmentation masks, resolving a chicken & egg problem. An iterative algorithm alternating between segmentation and pose estimation could be formulated, similarly to expectation-maximization algorithms.

Moreover, it would be interesting to use the Poselet representation presented in chapter 6 for the trimap initialization step of the segmentation system proposed in chapter 3. Similarly to the person and face detectors used to initialize the trimaps, Poselets provide with strong

prior pixel-wise information about the location of body and background regions among the corresponding image patch.

Human pose estimation

As commented in the previous section, the proposed Poselet-based mid-level part representation and contextual rescoring methodology proposed in chapter 6 may provide potentially useful cues to recover the position of occluded body parts. An interesting future work would be to automatically discover a set of Poselets that are especially useful for dealing with occluded body parts, while non-occluded parts could be detected by their corresponding filters in the pictorial structure model.

Gesture recognition

Finally, it would be interesting to develop a gesture recognition system taking advantage of the contributions presented in this dissertation regarding segmentation and pose estimation. A first segmentation of the subject in the scene could be performed in order to get rid of the background clutter. Then the body pose could be estimated by a pictorial structure incorporating top-down position prediction via contextual rescoring and Poselet detections. In addition, Poselets could also be extended to the temporal domain, similarly to the Dynamic-Poselets presented in [91]. Finally, estimated poses at different frames could be aggregated over time, and the proposed probabilistic dynamic time warping methodology could be applied.

Appendix A

Code and Data

The following code and data contributions have been made publicly available in the course of the years dedicated to this thesis.

Code

- **Contextual rescoring for Human Pose Estimation**.
  Available at http://www.cvc.uab.es/~ahernandez/contextual.html.

Data

- **Human Limb dataset**: This dataset is composed of 227 images from 25 different people. At each image, 14 different limbs are labeled, including the "do not care" label between adjacent limbs, as described in Figure A.1. Backgrounds are from different real environments with different visual complexity. This dataset is useful for human segmentation, limb detection, and pose recovery purposes.
  Available at http://www.cvc.uab.es/~ahernandez/data.html.
- **Human Limbs from RGBD Data**: This dataset contains labeled body parts in videos recorded with Kinect camera (RGB+Depth). Each frame is composed by one 24 bit RGB image of size 640x480 pixels, one 12 bit depth buffer of the same dimension, and a skeletal graph describing relevant joints of the upper human body. The whole ground truth used in our experiments is created from capturing 2 actors in 3 sessions gathering 500 frames in total (15 fps).
  Available at http://www.cvc.uab.es/~ahernandez/data.html.

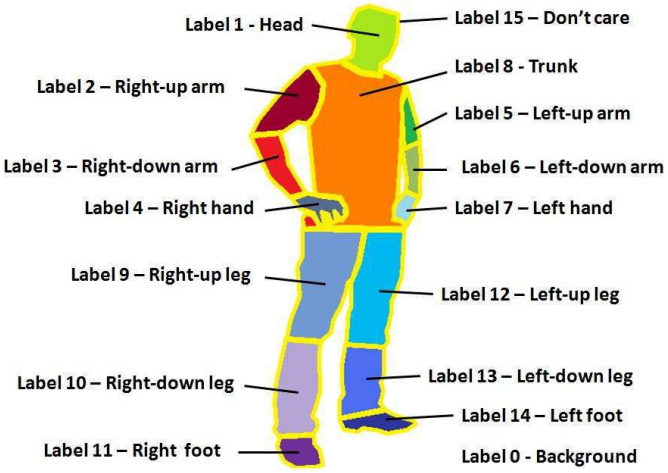


Figure A.1: Human Limb dataset labels description.

Appendix B

Publications

The following publications are a consequence of the research carried out during the elaboration of this thesis and give an idea of the progression that has been achieved.

Journal papers

- A. Hernández-Vela, S. Sclaroff and S. Escalera. Poselet-based Contextual Rescoring for Human Pose Estimation via Pictorial Structures. *International Journal of Computer Vision*, 2014, Under review.
- A. Hernández-Vela, M.A. Bautista, X. Pérez-Sala, V. Ponce-López, S. Escalera, X. Baró, O. Pujol and C. Angulo. Probability-based Dynamic Time Warping and Bag-of-Visual-and-Depth-Words for Human Gesture Recognition in RGB-D. *Pattern Recognition Letters*, 2013.
- A. Hernández-Vela, M. Reyes, V.Ponce and S. Escalera. GrabCut-Based Human Segmentation in Video Sequences. *Sensors*, 2012.
- A. Hernández-Vela, C. Gatta, S. Escalera, L. Igual, V. Martín-Yuste, M. Sabaté and P. Radeva. Accurate coronary centerline extraction, caliber estimation and catheter detection in angiographies, *IEEE Transactions on Information Technology in Biomedicine*, 2012.
- A. Hernández-Vela, N. Zlateva, A. Marinov, M. Reyes, P. Radeva, D. Dimov and S. Escalera. Human Limb Segmentation in Depth Maps based on Spatio-Temporal Graph Cuts Optimization. *Journal of Ambient Intelligence and Smart Environments (JAISE)*, 2012.
- L. Igual, J.C. Soliva, A. Hernández-Vela, S. Escalera, X. Jimenez, O. Vilarroya and P. Radeva. A Fully-Automatic Caudate Nucleus Segmentation of Brain MRI: Application in Volumetric Analysis of Pediatric Attention-Deficit/Hyperactivity Disorder. *BioMedical Engineering OnLine*, 2011.

International Conferences and Workshops

- A. Hernández-Vela, S. Sclaroff and S. Escalera. Contextual Rescoring for Human Pose Estimation. In *British Machine Vision Conference*, 2014.

- M.A. Bautista, A. Hernández-Vela, V. Ponce, X. Pérez-Sala, X. Baró, O. Pujol, C. Angulo and S. Escalera. Probability-based Dynamic Time Warping for Gesture Recognition on RGB-D data. In *International Workshop on Depth Image Analysis*, 2012.
- A. Hernández-Vela, M.A. Bautista, V. Ponce, X. Pérez-Sala, X. Baró, O. Pujol, C. Angulo and S. Escalera. BoVDW: Bag-of-Visual-and-Depth-Words for Gesture Recognition. In *International Conference on Pattern Recognition*, 2012.
- L. Igual, J.C. Soliva, A. Hernández-Vela, S. Escalera, O. Vilarroya and P. Radeva. Supervised Brain Segmentation and Classification in Diagnostic of Attention-Deficit/Hyperactivity Disorder. In *High Performance Computing & Simulation*, 2012.
- A. Hernández-Vela, N. Zlateva, A. Marinov, M. Reyes, P. Radeva, D. Dimov and S. Escalera. Graph Cuts Optimization for Multi-Limb Human Segmentation in Depth Maps. In *IEEE Computer Vision and Patern Recognition*, 2012.
- A. Hernández-Vela, C. Primo and S. Escalera. Automatic User Interaction Correction via Multi-label Graph Cuts. In *Workshop on Human Interaction in Computer Vision*, 2011.
- A. Hernández-Vela, C. Gatta, S. Escalera, L. Igual and P. Radeva. Accurate and robust fully-automatic QCA: Method and numerical validation. In *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2011.
- A. Hernández, M. Reyes, S. Escalera and P. Radeva. Spatio-Temporal GrabCut Human Segmentation for Face and Pose Recovery. In *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, 2010.

Bibliography

[1] Code laboratories cl nui platform - kinect driver/sdk. http://codelaboratories.com/nui/.

[2] Flexible action and articulated skeleton toolkit (faast). http://projects.ict.usc.edu/mxr/faast/.

[3] Kinect for windows sdk from microsoft research. http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/.

[4] Nite middleware. http://www.primesense.com/?p=515.

[5] Openkinect (libfreenect). http://openkinect.org/.

[6] Openni. http://www.openni.org.

[7] Primesensor^{TM}. http://www.primesense.com/?p=514.

[8] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. Rusu, and G. Bradski. Cad-model recognition and 6dof pose estimation using 3d cues. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 585 –592, nov. 2011.

[9] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff. A unified framework for gesture recognition and spatiotemporal gesture segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(9):1685–1699, 2009.

[10] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1014–1021, June 2009.

[11] I. Bayer and T. Silbermann. A multi modal approach to gesture recognition from audio and video data. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 461–466, New York, NY, USA, 2013. ACM.

[12] K. K. Biswas and S. Basu. Gesture recognition using microsoft kinect. In *Automation, Robotics and Applications (ICARA), 2011 5th International Conference on*, pages 100–103, 2011.

[13] L. Bo, X. Ren, and D. Fox. Depth kernel descriptors for object recognition. In *IROS*, pages 821–826, sept. 2011.

[14] R. Bogdan, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *ICRA*, pages 3212–3217, 2009.

[15] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *ECCV 2010*, volume 6316 of *Lecture Notes in Computer Science*, pages 168–181. Springer Berlin Heidelberg, 2010.

[16] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1365–1372, Sept 2009.

[17] Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. *International Journal on Computer Vision*, 70:109–131, 2006.

[18] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:359–374, 2001.

[19] Y. Boykov, O.Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23:1222–1239, November 2001.

[20] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *International Conference on Computer Vision*, 2001.

[21] M. Bray, P. Kohli, and P. H. Torr. Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In *Computer Vision–ECCV 2006*, pages 642–655. Springer, 2006.

[22] Chalearn gesture dataset, california, 2011.

[23] X. Chen and M. Koskela. Online rgb-d gesture recognition with extreme learning machines. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 467–474, New York, NY, USA, 2013. ACM.

[24] G. Cheung, T. Kanade, J.-Y. Bouguet, and M. H. A. Real time system for robust 3d voxel reconstruction of human motions. 2:714–720, 2000. IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island,(USA),.

[25] R. Cinbis and S. Sclaroff. Contextual object detection using set-based classification. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *ECCV 2012*, volume 7577 of *Lecture Notes in Computer Science*, pages 43–57. Springer Berlin Heidelberg, 2012.

[26] T. Cootes, J. Edwards, and C. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[27] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[28] D. Corrigan, S. Robinson, and A. Kokaram. Video matting using motion extended grabcut. *IET Conference Publications*, pages 1–9, 2008.

[29] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV*, pages 1–22, 2004.

[30] N. Dalal and B. Triggs. Histogram of oriented gradients for human detection. volume 2, pages 886–893, 2005.

[31] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. *European Conference on Computer Vision*, pages 7–13, 2006.

[32] T. Darrell, G. Gordon, J. Woodfill, and M. Harville. Integrated person tracking using stereo, color, and pattern detection. *IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara*, pages 601 –608, 1998.

[33] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, December 2006.

[34] P. Doliotis, A. Stefan, C. McMurrough, D. Eckhard, and V. Athitsos. Comparing gesture recognition accuracy using color and depth information. In *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '11, pages 20:1–20:7, 2011.

[35] K. Duan, D. Batra, and D. Crandall. A multi-layer composite model for human pose estimation. In *Proceedings of the British Machine Vision Conference*, pages 116.1–116.11. BMVA Press, 2012.

[36] M. Eichner and V. Ferrari. Appearance sharing for collective human pose estimation. In K. Lee, Y. Matsushita, J. Rehg, and Z. Hu, editors, *ACCV 2012*, volume 7724 of *Lecture Notes in Computer Science*, pages 138–151. Springer Berlin Heidelberg, 2013.

[37] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, Sept 2010.

[38] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

[39] V. Ferrari, M. Marin, and A. Zisserman. Pose search: retrieving people using their pose. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2009.

[40] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK (USA), 2008.

[41] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real time motion capture using a single time-of-flight camera. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 755–762, 2010.

[42] D. Geronimo, A. Lopez, and A. Sappa. Survey of pedestrian detection for advanced driver assistance systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1239–1258, 2010.

[43] R. Girshick, J. Shotton, A. Criminisi, and A. Fitzigibbon. Efficient regression of general-activity human poses from depth images. *ICCV*, 2011.

[44] L. Grady. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11):1768–1783, Nov. 2006.

[45] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, and S. Pankanti. Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking. *Signal Processing Magazine, IEEE*, 22(2):38–51, 2005.

[46] C. Harris and M. Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[47] A. Hernandez-Vela, C. Gatta, S. Escalera, L. Igual, V. Martin-Yuste, M. Sabate, and P. Radeva. Accurate coronary centerline extraction, caliber estimation, and catheter detection in angiographies. *Information Technology in Biomedicine, IEEE Transactions on*, 16(6):1332–1340, Nov 2012.

[48] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. (07-49), October 2007.

[49] L. Igual, J. Soliva, A. Hernandez-Vela, S. Escalera, X. Jimenez, O. Vilarroya, and P. Radeva. A fully-automatic caudate nucleus segmentation of brain mri: Application in volumetric analysis of pediatric attention-deficit/hyperactivity disorder. 10(105), december 2011.

[50] C. Ionescu, J. Carreira, and C. Sminchisescu. Iterated second-order label sensitive pooling for 3d human pose estimation. June 2014.

[51] H. Jain and A. Subramanian. Real-time upper-body human pose estimation using a depth camera. *HP Technical Reports*, 1(190), 2010.

[52] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, pages 12.1–12.11. BMVA Press, 2010.

[53] S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1465–1472, June 2011.

[54] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun. Real time hand pose estimation using depth sensors. In *Consumer Depth Cameras for Computer Vision*, pages 119–137. Springer, 2013.

[55] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:65–81, 2004.

[56] K. Lai, L. Bo, X. Ren, and D. Fox. Sparse distance learning for object recognition combining rgb and depth information. In *ICRA*, pages 4007 –4013, may 2011.

[57] I. Laptev. On space-time interest points. *Int. J. Comput. Vision*, 64(2-3):107–123, Sept. 2005.

[58] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1–8, 2008.

[59] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178, 2006.

[60] D. D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *ECML*, pages 4–15. Springer Verlag, 1998.

[61] Y. Li. Hand gesture recognition using kinect. In *Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on*, pages 196–199, 2012.

[62] Y. Liu, C. Stoll, J. Gall, H.-P. Seidel, and T. C. Markerless motion capture of interacting characters using multi-view image segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 14(1):1249–1256, 2011.

[63] D. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2, 1999.

[64] M. Mirza-Mohammadi, S. Escalera, and P. Radeva. Contextual-guided bag-of-visual-words model for multi-class object categorization. In *CAIP*, pages 748–756, 2009.

[65] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 79(3):299–318, 2008.

[66] F. Pedersoli, N. Adami, S. Benini, and R. Leonardi. Xkin -: extendable hand pose and gesture recognition library for kinect. In *ACM Multimedia*, pages 1465–1468, 2012.

[67] A. Pentland. Socially aware computation and communication. *Computer*, 38:33–40, 2005.

[68] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 588–595, June 2013.

[69] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Strong appearance and expressive spatial models for human pose estimation. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3487–3494, Dec 2013.

[70] E. Puertas, M. A. Bautista, D. Sanchez, S. Escalera, and O. Pujol. Learning to segment humans by stacking their body parts. In *ECCV 2014 Workshops (In Press)*, 2014.

[71] D. Ramanan. Learning to parse images of articulated bodies. In *NIPS*, 2006.

[72] M. Reyes, G. Dominguez, and S. Escalera. Feature weighting in dynamic time warping for gesture recognition in depth data. In *ICCV*, 2011.

[73] J. Rodgers, D. Anguelov, H.-C. Pang, and D. Koller. Object pose detection in range scan data. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2445 – 2452, 2006.

[74] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (SIGGRAPH)*.

[75] R. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2155 –2162, oct. 2010.

[76] B. Sabata, F. Arman, and J. K. Aggarwal. Segmentation of 3d range images using pyramidal data structures. *CVGIP: Image Understanding*, 57(3):373–387, 1993.

[77] B. Sapp, C. Jordan, and B. Taskar. Adaptive pose priors for pictorial structures. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 422–429, June 2010.

[78] W. Schwartz, A. Kembhavi, D. Harwood, and L. Davis. Human detection using partial least squares. pages 24–31, 2009. International Conference on Computer Vision.

[79] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1304, 2011.

[80] T. Starner and A. Pentland. Real-time american sign language recognition from video using hidden markov models. In *Computer Vision, 1995. Proceedings., International Symposium on*, pages 265 –270, nov 1995.

[81] L. B. Statistics and L. Breiman. Random forests. In *Machine Learning*, pages 5–32, 2001.

[82] E. Suma, B. Lange, A. Rizzo, D. M. Krum, and M. Bolas. FAAST: the flexible action and articulated skeleton toolkit. In *IEEE Virtual Reality*, pages 245–246, Singapore, Mar. 2011.

[83] M. Sun, M. Telaprolu, H. Lee, and S. Savarese. An efficient branch-and-bound algorithm for optimal human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1616–1623, June 2012.

[84] J. Sung, C. Ponce, B. Selman, and A. Saxena. Unstructured human activity detection from rgbd images. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 842–849, May 2012.

[85] T.-P. Tian and S. Sclaroff. Fast globally optimal 2d human detection with loopy graph models. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 81–88, June 2010.

[86] Y. Tian, C. Zitnick, and S. Narasimhan. Exploring the spatial hierarchy of mixture models for human pose estimation. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *ECCV 2012*, volume 7576 of *Lecture Notes in Computer Science*, pages 256–269. Springer Berlin Heidelberg, 2012.

[87] F. Tiburzi, M. Escudero, J. Bescos, and J. Martinez. A ground-truth for motion-based video-object segmentation: http://www-gti.ii.uam.es/cvsg. In *IEEE International Conference on Image Processing (Workshop on Multimedia Information Retrieval*, San Diego (USA), 2008.

[88] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57:137–154, 2004.

[89] T. Wan, Y. Wang, and J. Li. Hand gesture recognition system using depth data. In *Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on*, pages 1063–1066, 2012.

[90] F. Wang and Y. Li. Beyond physical connections: Tree models in human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 596–603, June 2013.

[91] L. Wang, Y. Qiao, and X. Tang. Video action detection with relational dynamic-poselets. In *Computer Vision - ECCV 2014*, volume 8693 of *Lecture Notes in Computer Science*, pages 565–580. Springer International Publishing, 2014.

[92] Y. Wang, D. Tran, and Z. Liao. Learning hierarchical poselets for human parsing. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1705–1712, June 2011.

[93] J. Wu, J. Cheng, C. Zhao, and H. Lu. Fusing multi-modal features for gesture recognition. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 453–460, New York, NY, USA, 2013. ACM.

[94] B. D. Y. Zhu and K. Fujimura. Controlled human pose estimation from depth image streams. *Computer Vision and Pattern Recognition Workshop on TOF Computer Vision*, pages 1–8, 2008.

[95] H. Yang and S. Lee. Reconstruction of 3d human body pose from stereo image sequences based on top-down learning. *Pattern Recognition*, 40(11):3120–3131, 2007.

[96] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(12):2878–2890, Dec 2013.

[97] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 17–24, June 2010.

[98] F. Zhou, F. De la Torre, and J. Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE TPAMI*, page 1, 2012.