3D Human Pose Estimation from Monocular Image Sequences

Project Report

Adela Barbulescu

Aalborg University
Department of Electronic Systems
Fredrik Bajers Vej 7B
DK-9220 Aalborg

Aalborg University
VGIS 10th Semester
Department of Electronic Systems
Fredrik Bajers Vej 7,
9220 Aalborg, Denmark,
Tel: +45 9940 8600,
E-mail: webinfo@es.aau.dk

Title:
3D Human Pose Estimation from Monocular Image Sequences

Theme:
Object Detection and Tracking

Project Period:
Master Semester 2012

Project Group:
VGIS10 group 1027

Participant(s):
Adela Barbulescu

Supervisor(s):
Thomas Moeslund
Jordi Gonzalez

Copies: 2

Page Numbers: 55

Date of Completion:
May 31, 2012

Synopsis:

The topic of the project is 3D human pose estimation from monocular image sequences. The problem addresses video frames in uncontrolled conditions, containing persons revealed in a high variety of poses.

The main goal is implementing a system that is able to automatically detect human poses in consecutive frames and map them to 3D configurations, without requiring additional background information.

To this end, the implemented system meets the initial requirements, being able to estimate 3D configurations on benchmark databases and outperforming previous works. However, the system performance is limited by the quality and size of the dataset of poses on which it is trained.

Adela Barbulescu
badela10@student.aau.dk

Abstract

Automatic 3D reconstruction of human poses from monocular images is a challenging and popular topic in the computer vision community, which provides a wide range of applications in multiple areas. Solutions for 3D pose estimation involve various learning approaches, such as Support Vector Machines and Gaussian processes, but many encounter difficulties in cluttered scenarios and require additional input data, such as silhouettes, or controlled camera settings.

The project outlined consists of a framework that is capable of estimating the 3D pose of a person from monocular image sequences without requiring background information and which is robust to camera variations. The framework models the inherent non-linearity found in human motion as it benefits from flexible learning approaches, including a highly customizable 2D detector and a Gaussian process regressor trained on specific action motions.

Results on the HumanEva benchmark show that the system outperforms previous works obtaining a 70% decrease in average estimation error on identical datasets. Detailed settings for experiments, test results and performance measures on 3D pose estimation are provided.

Contents

Abstract **v**

List of Figures **ix**

List of Tables **xiii**

Preface **xv**

1 Introduction **1**
1.1 Goal and Applications . . . 3
1.2 Issues and Challenges . . . 4
1.3 System requirements . . . 5
1.4 Problem formulation . . . 5
1.5 Deliminator . . . 5

2 State of the art **6**
2.1 Related surveys . . . 6
2.2 Related work . . . 8
2.3 Theoretical aspects . . . 10
2.3.1 3D Human body representation . . . 10
2.3.2 Low-level image descriptors . . . 10
2.3.3 Part-based models for 2D pose estimation . . . 14
2.3.4 Machine learning for pose estimation . . . 15
2.4 HumanEva benchmark . . . 18
2.5 Outline of proposed framework . . . 20

3 2D Human Detection and Smoothing **22**
3.1 2D Human Detector . . . 22
3.1.1 Model . . . 23
3.1.2 Inference . . . 25
3.1.3 Learning . . . 26
3.1.4 Experiments . . . 27
3.2 2D Motion Smoothing . . . 28
3.2.1 Algorithm description . . . 29

3.2.2 Experiments . . . 31
3.3 Results . . . 32
3.4 Conclusions . . . 34

4 3D Pose Regression **37**
4.1 Gaussian processes . . . 37
4.2 Training a Gaussian process . . . 38
4.3 Posterior Gaussian process . . . 40
4.4 Data representation . . . 41
4.4.1 Direction cosines . . . 42
4.4.2 3D body pose . . . 43
4.5 Experiments . . . 43
4.6 Results . . . 45
4.7 Conclusions . . . 47

5 Conclusion **50**

Bibliography **52**

A 3D Human Pose Estimation using 2D Body Part Detectors **55**

List of Figures

1.1 Examples of motion capture systems used in game and film industry. Pose information is captured using expensive equipment: multi-cameras, infrared cameras, invasive markers and suits. 2
1.2 Snaphots containing interfaces for 3D recontruction software using Organic Motion (a) and Kinect (b). 3
1.3 Example of 2D-3D ambiguities in human poses. The silhouette (a) maps to two possible poses (b). 4

2.1 Geometric reconstruction approach: limb lengths in images can reconstruct the displacement in direction z.(Figure from [13]). 9
2.2 (a) Stick figure model. (Figure from [1]) (b) Volumetric model consisting of super quadrics. (Figure from [18]) (c) Volumetric model consisting of elliptical cylinders. (Figure from [17]). 11
2.3 (a) Extracted silhouette (b) Edge points (c) Shape contexts. (Figure from [15]). 11
2.4 Model customized for finding lateral-walking poses. The template structure and part location bounds are initialized by hand. To prune false detections in textured backgrounds the score is reevaluated by adding person pixel constraints. (Figure from [7]). 12
2.5 (a) Average gradient image extracted from training sets. (b) Positive SVM weights centered on blocks of images. (c) Negative SVM weights. (d) Test image. (e) Computed HOG descriptor. (f) HOG descriptor weighted by positive SVM weights. (g) HOG descriptor weighted by negative SVM weights.(Figure from [28]). 13
2.6 On the left, definition of Binford's generalized cylinders [2] by two functions: the cross-section (a) and the sweeping rule (b), which modulates the width of the cylinder (c) through the transverse axis to create the final cylinder (d). On the right, Fischler's pictorial structure [24] which models object using local part templates and geometric constraints, visualized by strings. 14

2.7 Single-component model defined by a root filter (a), higher resolution part-filters specifying weights for HOG features (b) and spatial model representing weights associated with parts for placing their centers at different locations relative to the root (c).(Figure from [9]). 15
2.8 Dashed lines represent different possible hyperplanes for separating the two classes. The red line represents the hyperplane with the largest margin and best generalization. 16
2.9 Images of a subject performing walking action from synchronized video cameras with overlaid motion capture data. 19
2.10 Outline of the 3-stage framework. 20

3.1 Filters associated to a mixture of head part-types. Each filter favors a particular orientation of the head. 23
3.2 Tree visualization describing a full-body 14-part model(Figure from [41]). 24
3.3 Clusters used to generate mixture labels during training for T=4. Part-types represent different orientations of parts relative to their parents (Figure from [41]). 27
3.4 A visualization based on part templates for an 18-part upper-body model, 14-part and 26-part full-body model. Parts are represented by 5x5 HOG templates, placed at their highest scoring locations for a given root position and type. 28
3.5 Examples of successful detections on the Parse dataset. 28
3.6 HumanEva examples: The top row presents successful detection on different actions and cameras. The bottom row presents failing situations: double-counting, body or limb misdetection, self-occlusion. 29
3.7 Detections without smoothing (a) Detections with robust smoothing (b) Detections with weighted robust smoothing (c) Motions (red) and detected pose (green) bounding boxes (d). 32
3.8 Part error plots per joints over all frames and actions. 35
3.9 Error plots per actions over all frames and mean error for weighted robust smoothing. 36

4.1 Random functions drawn from a GP prior with dots indicating output values generated from a finite set of input points (a). Random functions drawn from the GP posterior, conditioned on 5 noise-free observations (b)(Figure from [33]). 38
4.2 Sample function drawn from a GP with hyperparameters (1,1,0.1)(a), (0.3,1.08,0.00005) (b) and (3,1.16,0.89)(c). '+' Symbols represent noise-free observations and the shaded area corresponds to a 95% confidence region. (Figure from [33]). 39

4.3 Steps of the prediction algorithm implementation, where L is the lower triangular matrix obtained in the Cholesky decomposition of a matrix $A = LL^T$. The equation $Ax = b$ is solved efficiently using two triangular systems: $x = L\backslash(L\backslash y)$, where the notation $A\backslash b$ represents the solution for $Ax = b$. (Figure from [33]). 41
4.4 3D human body model (a) Direction cosines for an orientated limb (b).(Figure from [34]). 44
4.5 The first row presents test images from all actions datasets. The second row presents the corresponding kinematic tree structures with estimated limbs, presented from a 45° view relative to the body. 45
4.6 The ground truth (left) and estimated (right) tree kinematic structures for the Box dataset from frame 199 to 201. The error peak of 87 mm is reached in frame 200 and the ground-truth missing data is deducted from the discontinuous motion of the limbs. 47
4.7 Error plots per actions over all testing frames and mean error. 48
4.8 Part error plots per joints over all testing frames and actions. 49

List of Tables

3.1 Mean pose errors and on specific joints are computed for S1, cam1, on all action types. Results are expressed in pixels and they are compared for the smoothing option: no smoothing used, robust or weighted smoothing. 33

4.1 Size of training and testing data used from HumanEva, subject S1, camera C1. 44

4.2 Average limb position and angle errors are computed for S1, Cam1, on all action types. Results are compared for the presented framework, and the ones used in [3] and [13]. 46

Preface

The thesis outlined is intended for the graduation of the Vision, Graphics and Interactive Systems master program at Aalborg University, under the guidence of Thomas B. Moeslund. The project was carried during a research stay at Centre de Visio per Computador, Universitad Autonoma de Barcelona, between November 2011 and June 2012. The work was co-supervised by Jordi Gonzalez, academic member of the Image Sequence Evaluation Lab. During this period, part of the work was directed towards submitting an article to the International Conference of Pattern Recognition 2012, entitled 3D Human Pose Estimation using 2D Body Part Detectors. The content of the article is found in the Appendix.

Chapter 1

Introduction

The surrounding environment is perceived uniquely by each individual through its sensory systems and at the same time, it is more and more digitally captured by technological means which relate to the human sensory systems. The most widespread technologies are related to capturing visual signals and enjoy a large audience of users, spreading from amateur photography and multimedia content, to medical imaging and complex photometric systems.

The large amount of visual data obtained has lead to the need of developing automatic systems that are able to interpret this data. For example, millions of security cameras have been installed in the past years for security purposes in public transport systems, border monitoring and private alarm systems. Also, the multimedia content is exponentially increasing along with the need of an automatic indexing procedure. Therefore, the focus on research in vision related problems has also increased in order to develop more cost-effective and time efficient systems.

The most researched topic is understanding and interpretation of visual recordings containing human activities. Human pose estimation represents the process of estimating the configuration of human body parts from data input such as static images, image sequences, multi-view imagery etc. When the sensor input also contains a temporal dimension the term of human motion analysis is used. In recent years, human pose estimation has received a significant amount of attention in the computer vision community and has become one of the main challenges due to its difficulties and widespread applications in various fields, ranging from advanced human computer interaction and smart video surveillance to the entertainment industry and arts.

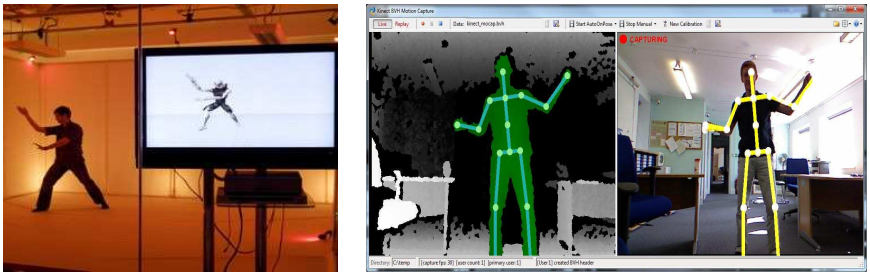
The vision topics that are addressed in this context are human detection and tracking, classification and action recognition. In order to carry a fine analysis of motion and action recognition, a 3D estimation of the articulated pose, shape and motion of the human body is needed. 3D estimation of poses implies generating a representation of certain keypoints belonging to the human skeleton in the 3D space.

Traditional technology applied in fields such as movie industry for motion capture uses expensive multi-camera and invasive marker systems, which require careful calibration and highly controlled laboratory conditions as pictured in Figure 1.1. Recently, Organic Motion has developed a new computer vision commercial system that is able to recreate a 3D model of a subject complete with 3D motion at milimeter accuracy in realtime. The system uses multiple 2D video cameras to track the subject by combining the triangulated locations of identical pixels in the scene. The technology eliminates requirements such as body suits or markers, thus improving flexibility and significantly reducing the costs implied by achieving such a system.

Figure 1.1: Examples of motion capture systems used in game and film industry. Pose information is captured using expensive equipment: multi-cameras, infrared cameras, invasive markers and suits.

Another vision commercial system that received a great amount of attention from the motion capture community is the XBOX Kinect camera, which uses two synchronized infrared and RGB cameras to capture depth and RGB scenario information. The low cost and availability of open source drivers and software has turned the Kinect into an easy configurable device for motion caption related applications placed at the disposal of a very large audience. However, the camera can only be used indoors with a range distance limit of around 10 meters and it presents less accuracy then other commercial motion capture systems. Figure 1.2 shows how these technologies can be used. As a large amount of the visual content which needs to be indexed cannot be captured by such commercial systems, the attention has been directed towards recovering the 3D human pose using only monocular image sequences.

The next sections introduce the open problem of human pose estimation, presenting the goal and applications, issues and challenges, and finally giving presenting the requirements of a system being able overview of the work carried in the thesis within the outlined context.



(a) (b)

Figure 1.2: Snaphots containing interfaces for 3D recontruction software using Organic Motion (a) and Kinect (b).

1.1 Goal and Applications

The goal of research carried in the topic of human pose estimation is to develop less invasive automatic systems that are able to generate 3D estimates of human poses in uncontrolled conditions, given only video sequences containing persons (such as outputs from surveillance or web cameras). A critical subject for automatic pose estimation is the use of monocular image sequences, which would enable a larger range of commercial applications and individual users to benefit from it: 3D animations, gaming, human computer interaction, abnormal behavior recognition etc.

The applications in human pose estimation can be organized in 3 main categories:

- *activity and gesture recognition*: given a motion sequence of a human performing activities or actions, the activity is recognized. Related topics are: smart video surveillance (recognizing actions or abnormal behavior with minimal user supervision), advanced human computer interfaces (using machine interfaces which are able to recognize gestures or interpret user behavior), automatic annotation (annotating huge amounts of digital data automatically by detecting activities without user supervision).

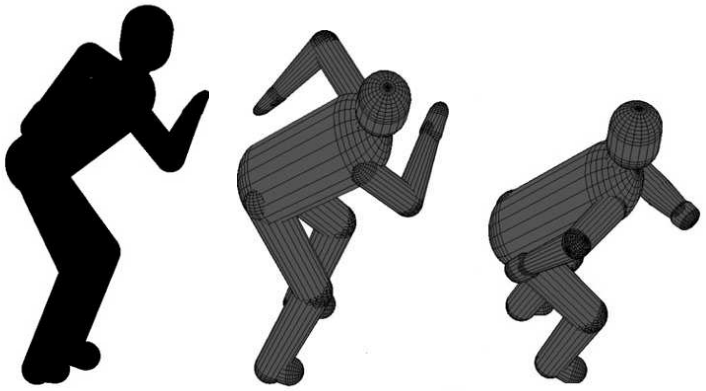
- *motion capture*: given a video sequence containing human motion, a set of keypoints are tracked in order to obtain a 3D representation of the analyzed body parts over time. Such applications are used in: sports biomechanics (enhancing sporting performances), arts and entertainment (improving 3D animations and visual effects in games and film industry, studying the motions of artists and dancers).

- *motion synthesis*: automatic creation of human pose data with applications in human computer interaction (interfaces using synthetic data), virtual reality, pose reanimation (recreating poses which can be observed from different viewpoints).

1.2 Issues and Challenges

3D estimation of human poses is still an open problem as the vision based systems encounter challenges that emerge from the following main issues:

- *3D space to 2D image plane projection ambiguities*: 2D image planes can be easily generated from 3D scenes using perspective projection from the pinhole camera model, also leading to loss of depth information. The inverse process maps one point of the 2D image to a line in 3D scene, revealing a one-to-many relation as any point from the 3D line may correspond to the 2D point. This results into an ill-conditioned problem which can be solved using learning or modeling approaches to map 2D to 3D data.



(a) (b)

Figure 1.3: Example of 2D-3D ambiguities in human poses. The silhouette (a) maps to two possible poses (b).

- *variability in shape and appearance of human poses*: humans may appear in a wide variety of poses, shapes and appearance, largely due to the highly articulated nature of the human body and complex distortions encountered in a single activity sequence, but also because of changes in clothes, illumination, noise, camera viewpoint.

- *image clutter*: human localization, which is a requirement in 3D pose estimation, is highly influenced by image clutter. Realistic scenarios, where background subtraction cannot be applied because of moving cameras, changes in illumination and variable background, require image descriptors and accurate predictors that are robust to background noise.

- *occlusions and self-occlusions*: changes in viewpoint and activities such as walking and running lead to situations in which different limbs are self occluded, increasing the complexity of possible poses. Also, other objects may occlude body-parts.

- *high dimensionality and non-linearity in human motion*: as the human body is composed of more than 30 main joints, an articulated body model presents around 60 degrees of freedom creating a high dimensional space of possible human poses with non-linear dynamics.

1.3 System requirements

The implementation of a system that is able to estimate 3D human poses from monocular image sequences is subject to a set of requirements:

- input is represented by a monocular video sequence containing one person performing actions while displaying a wide variety of poses
- the system is fully automatic, with the video input not containing any annotations
- the system does not require background information nor camera calibration
- the chosen 3D data representation allows visualization and error measure methods

1.4 Problem formulation

Considering the goals and challenges of research carried in the field of 3D human pose estimation and requirements of implementing such a system, the following problem formulation is chosen:

How should the 3D human configuration be estimated given only a monocular video sequence of a person performing various actions?

1.5 Deliminator

The implemented system has met the following limitations:

- the video must contain only one person
- the system is confused by horizontal poses
- the quality of 3D estimations and variability of detectable poses depends on the available benchmark datasets containing 3D ground truth data
- the absolute position and orientation of the body are not retrieved as 3D configurations are represented as relative 3D body part locations in a local coordinate system

Chapter 2

State of the art

This chapter covers theoretical notions and state of the art approaches related to 3D human motion analysis from monocular image sequences, based on which the proposed system is implemented. Therefore, the most important surveys and works in the related literature are outlined in the first two sections of the chapter and theoretical key aspects that emerge from these are described in more detail in Section 2.3. Next, a benchmark dataset framework which has the purpose of maintaining a ranking and a measurable comparison between all these approaches and which is also used in the project is presented in Section 2.4. Finally, Section 2.5 describes a general outline of the system.

2.1 Related surveys

A broad overview of the most common approaches used in vision-based human motion analysis is given in a few surveys: [38], [27], [30], [36], [6], [15]. These also present a general taxonomy of motion analysis techniques: detection, tracking, human pose estimation and recognition:

- T. B. Moeslund presents two surveys that review human motion capture related papers published until 2000 [38] and an extension [27] outlining related work from 2000 to 2006. The first survey presents a taxonomy of system functionalities composed of four main processes: initialization, tracking, pose estimation and recognition. Initialization represents the first stage of data processing as an appropriate model is established for the subject, ensuring a correct interpretation for the initial scene. A general description for tracking is analyzing the human motion in consecutive frames by segmenting the human body from the background and finding correspondences between the segments. Next, the human pose is estimated by determining the configuration of the body and limbs in a given frame. During recognition the resulted parameters are processed in order to classify the motion as belonging to a certain type of action. A general description of methods and

performance comparison are made in order to analyze the state of art. The second survey presents advances in the field of motion capture with emphasis on automatic methods for pose estimation and tracking in natural scenes rather than controlled laboratory conditions and the general advances obtained in each of the above mentioned processes.

- Poppe [30] presents a pose estimation taxonomy of two main classes: model-based and model-less approaches. Model-based methods imply a human body model consisting of a kinematic chain and body dimensions, while pose estimation consists of modeling and estimation. The modeling phase provides a likelihood function according to known parameters such as camera viewpoint, image descriptors, human body model. Pose estimation finds the most likely pose considering the likelihood function. On the other hand, model-free approaches do not assume a known human body model and implicitly model pose variations during a training phase. These approaches are divided into learning-based methods, when a function maps image descriptors to poses, and example-based, when mapping is done by similarity searching in a database of exemplars and corresponding poses.

- Sminchisescu [36] focuses on the 3D pose reconstruction problem and presents two main approaches: generative (top-down) and discriminative (bottom-up). Generative approaches use high-level descriptions of the complete human pose to explain low-level image descriptors. In this scope, synthetic 3d models are modeled and 2D poses are generated explicitly by rendering 3D pose hypotheses to the 2D image plane. An observation likelihood function is built and reaches a maxima when the 3d human model is mapped to the correct pose hypothesis. Discriminative approaches attempt to learn the inverse of perspective projection by directly mapping image descriptors to 3D poses. These methods use statistical learning models extensively and require training sets of corresponding images and poses.

- Forsyth [6] focuses on tracking and motion synthesis problems. Two main problems are depicted: lifting human poses from 2D to 3D space and determining which pixels are associated to the human body in the image space. Lifting ambiguities are shown to be easily solved when the temporal context is involved in the probabilistic framework.

- Hen [15] gives a review of techniques used for single camera 3D pose estimation and emphasizes on new research directions: bottom up approaches with the intermediary stage of local parts detection and mapping to 3D poses, the tendency of learning in low-dimensional pose space rather than high-dimensional appearance space and learning motion models from video sequences for smoother 3D poses.

2.2 Related work

As outlined in the previous chapter, the literature that covers the problem of 3D pose estimation is extremely vast and can be organized using different taxonomies depending on the focus of research. Considering the huge set of approaches used to tackle this problem, two main classes emerge: the first tries to map low-level image features directly to 3D human poses and the second uses an intermediary phase of finding 2D estimates of body parts and then mapping them to 3D poses.

One example of work included in the first class belongs to Agarwal and Triggs [1] who extract a dense grid of shape context histograms and map them directly to 3D poses, using various non-linear regression methods: ridge regression, relevance vector machine (RVM) regression, and support vector machine (SVM) regression. The method is also embedded in a regressive tracking framework. The second class can be divided into two sub-classes: learning and modeling approaches.

The first type of approaches involve 2D human detectors and learning the 2D-3D mapping from training examples. Recent work focuses on realistic environments with complex backgrounds and a more diverse set of poses. An approach would be joint human localization and 3D reconstruction as used in [23] or the use of more detailed 2D part-models [9], [41]. As some human detectors show a relatively high rate of false detections or difficulties with certain poses and viewpoints, the 2D pose estimates can be improved by incorporating the temporal dimension, using a tracking framework and learning dynamic human motion models.

A common method for tracking is the use of particle filters, which attempt to determine the distribution of a latent variable at a specific time, given all the observations until that particular time. Particles are propagated through the dynamic model and are continuously re-weighted by evaluating the likelihood. However, estimation of 3D poses implies high data dimensionality, particle filters being effective in more constrained scenarios such as: possibility of manual initialization, strong likelihood models or assumption of strong dynamic models. Otherwise more efficient search methods are employed. For example, Sidenbladh [14] uses a large training set on which efficient search is applied based on learned motion models. The method is called importance sampling and it is used to guide particle filtering search on the motion database. Motion priors enforce strong constraints and help in 3D pose tracking. However, the approach depends on the amount and variability of training data to learn accurate models of all possible motions. Another example is given by Andriluka et al [22] who first find 2D poses in single frames and then improve the estimates by extracting tracklets over short sequences of frames. The results are mapped to 3D poses using a latent Gaussian process model.

The second type of approaches try to model the 2D-3D mapping explicitly by using the inverse of the 3D to 2D mapping. The most used methods imply geometric recon-

structions of the 3D poses:

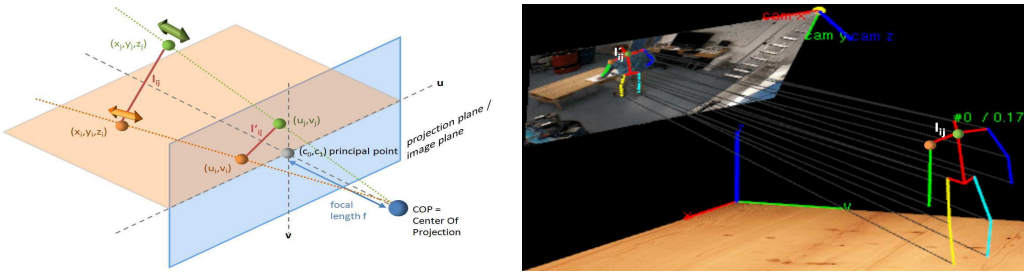


Figure 2.1: Geometric reconstruction approach: limb lengths in images can reconstruct the displacement in direction z.(Figure from [13]).

Taylor [37] uses a scaled orthographic projection model to reconstruct the displacements in depth of foreshortened limbs. Each limb endpoint presents two possibilities of projection depending on the chosen sense on the z direction. In the original work, user labels are required to point out which endpoint of a limb is closer to the camera. Other solutions involve matching shape context descriptors from a motion capture database. Also, the method is applicable only for poses that are far away from the camera. Another approach belongs to Brauer [16] who uses the perspective camera model which, for known settings of the camera, limb lengths and 2D parts coordinates, returns correct depth information irrespective of the distance between camera and detected person. Also, the tree of possible poses is checked for violations of anatomical joint positions which lead to pruning abnormal poses. The final 3D pose is selected by matching the set of remaining poses within a learning framework trained on a motion capture dataset.

The advantages and disadvantages of modeling and learning approaches are outlined by Gong et al [13] in an comparison of experiments employed on identical data inputs using a geometrical approach and a Gaussian process regressor from 2D estimates to 3D poses. The experiments imply scenarios which include various human actions and camera viewpoints, ground truth and noisy 2D data inputs. Results show that learning approaches perform better when the training data is more similar to testing data, precisely, when there are minor changes in viewpoint and action type, irrespective of the level of input data noise. They are outperformed by modeling approaches when the changes are major, as no similar poses are learned. On the other hand, the latter are outperformed in all scenarios when estimated or synthetic noisy 2D poses are used.

Another comparison between learning and modeling approaches in monocular 3D pose estimation is performed by Gong [39] on the effect of temporal information, by varying the number of consecutive frames used in the estimation process. Results show a general advantage of using consecutive frames against single frames as input data. Using ground truth temporal 2D data shows a slow increase in the precision, leading to

the preliminary conclusion that the window size of consecutive frames should be proportional to the quality of 2D estimates to obtain a better performance.

2.3 Theoretical aspects

The first two sections covered the most important approaches used in previous works on 3D pose estimation and motion analysis. From these, theoretical key aspects emerge that need to be considered when implementing a system in this scope. The next sections cover such theoretical notions.

2.3.1 3D Human body representation

Human pose estimation requires a general human body representation that is able to keep human specific features, considering issues such as pose and shape variability or changes in clothes and appearance. As a trade-off between low computational complexity and feature generality, the most common representation is the stick figure model or kinematic tree, (2.2 (a))composed of pivoting joints connected by rigid limbs or body parts. Depending on the degree of detail required, a variable number of joints and limbs may be used and a joint presents up to 3 degrees of freedom (DOF). For example, Sidenbladh [14], Sigal [20] and Gong [39] use models composed of 50, 47 and 30 DOF, respectively. Independent of the level of detail, a human body model should have at least 20 DOF: one for each knee and elbow, two for each hip, three for each shoulder and six for the root. Limbs are connected in a hierarchical manner, allowing different body parts to be represented and expressed relative to each other.

The body model can be represented in more detail using volumetric models which use 3D primitives such as elliptical cylinders (2.2 (c)), truncated cones, spheres, super quadrics (2.2 (b)). The super quadrics volumetric model is more accurate in comparison to less elliptic models as it supports a larger pose variability at the cost of more parameters required in the estimation process.

Although volumetric models present a more detailed structure which can lead to an improved matching between image and 3D space, the process of initializing the base primitive parameters implies higher computational complexity. Therefore, they are preferred in motion synthesis applications, while for tracking problems the kinematic models show overall better performance.

2.3.2 Low-level image descriptors

The first step in 3D human pose estimation is extracting the low-level image features which can later be mapped to high-level understanding tasks, such as 2D or 3D pose

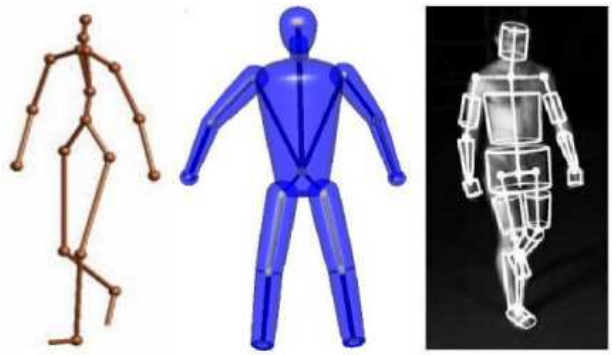


Figure 2.2: (a) Stick figure model. (Figure from [1]) (b) Volumetric model consisting of super quadrics. (Figure from [18]) (c) Volumetric model consisting of elliptical cylinders. (Figure from [17]).

estimates. The most common used image features in the problem of pose estimation are: silhouettes, shapes, edges, motions, colors, gradients and combinations of them. State of the art techniques for human detection use a combination of image features, shape contexts and learning approaches to robustly detect human poses.

Silhouettes and Contours

Silhouettes represent relevant image features to human detection as they are highly correlated with body contours, and therefore can be mapped to human pose. Silhouettes can accurately be extracted in the case of static backgrounds with stable illumination conditions using background subtraction. Most methods used for extracting silhouettes involve image differencing, single or mixture of Gaussian distribution on color statistics. To improve the segmentation process, Agarwal [1] uses shape context distributions: histograms of local regularly spaced edge pixels in log-polar bins which encode silhouette shape over scale ranges. Matching silhouettes reduces to matching shape context distributions as pictured in Figure 2.3.

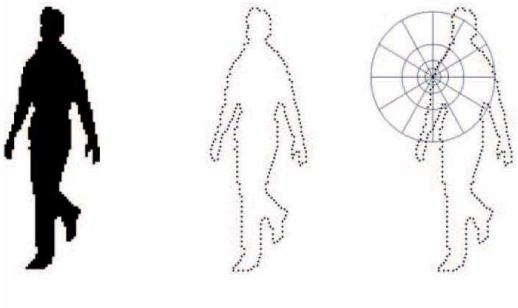


Figure 2.3: (a) Extracted silhouette (b) Edge points (c) Shape contexts. (Figure from [15]).

However, silhouette extraction becomes unreliable in the case of natural scenes with

cluttered background, illumination changes, camera motion and occlusions and may require additional background information for improved robustness.

Edges

Edges are important image features as they can be easily extracted, they can be used for body part delimitation and are insensitive to color, texture and illumination changes. Deutscher [17] uses a human detector in which the first stage is computing a pixel map-based weighting function. The map is produced by using an edge detection mask applied on the image and then thresholded to remove noisy edges. The second stage is improving the result with silhouette-based features, as edge information is not robust against clothes variability and cluttered background. Ramanan et al [7] integrate constant appearance information with extracted edges to find body segments by building person-specific templates. Reducing body-part contour edges to rectangles, the model built captures the full appearance of parts and tracks similar parts across consecutive frames as pictured in Figure 2.4.

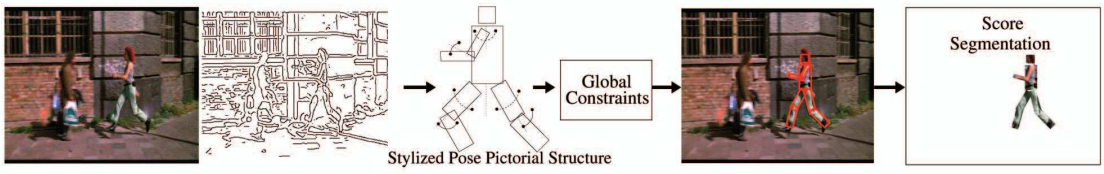


Figure 2.4: Model customized for finding lateral-walking poses. The template structure and part location bounds are initialized by hand. To prune false detections in textured backgrounds the score is reevaluated by adding person pixel constraints. (Figure from [7]).

Motions

Extracting motion information from image sequences is a common approach in human pose tracking and segmentation. Motion can be measured using optical flow approaches, by creating a 2D velocity map of pixel displacements between frames. Urtasun [31] uses silhouettes and optical flow to create mappings of 2D points between consecutive frames. An objective function which describes the mapping is minimized to obtain smoothness in 3D pose estimates. Andriluka et al [23] extends the pedestrian detector [22] and builds a multi-person tracking-by-detection framework by generating robust estimates of 2D body parts over short frame sequences called tracklets. Tracklets are extracted by matching pose hypotheses in different frames according to position, scale and appearance.

Colors

Color information can be used under stable illumination conditions for body part detection as it is invariant to scale and pose variability. Lee [25] improves body part detection

by integrating skin color histograms to find positions of arms, legs and faces. Ramanan [7] uses color features to track body parts with similar appearance. To ensure robust detection, normalization and post-processing are required, and also integration of other appearance features.

Oriented gradients

A very successful approach in object detection is working with gradient orientations rather than pixel values. Dalal et al [28] introduce histogram of oriented gradients (HOG) descriptors, which are invariant to changes in illumination, scale and viewpoint. State-of-the-art approaches for object detection [41], [9] use HOG descriptors, outperforming previous work on widely acknowledged benchmarks for object and human detection. The performance obtained is explained by the fact that object appearance can be very well described by the distribution of intensity gradients or edge directions.

The method used for extracting HOG descriptors is based on the evaluation of normalized histograms of oriented gradients obtained from a dense grid of image blocks. Practically, the image windows are contrast normalized and divided into small cells, each being associated with a local 1-dimensional histogram of edge directions over the cell pixels. Each cell votes for the direction weights and the combined vectors form the descriptor. The results are improved with a new normalization process over overlapping spatial blocks and a combined feature vector is formed by overlapping dense HOG descriptors. The final step consists of feeding the feature vectors to an SVM trained on images of the particular object. Figure 2.5 shows how HOG detectors cue the contrast of silhouette contours, against the presence of cluttered background, and not internal edges or foreground.

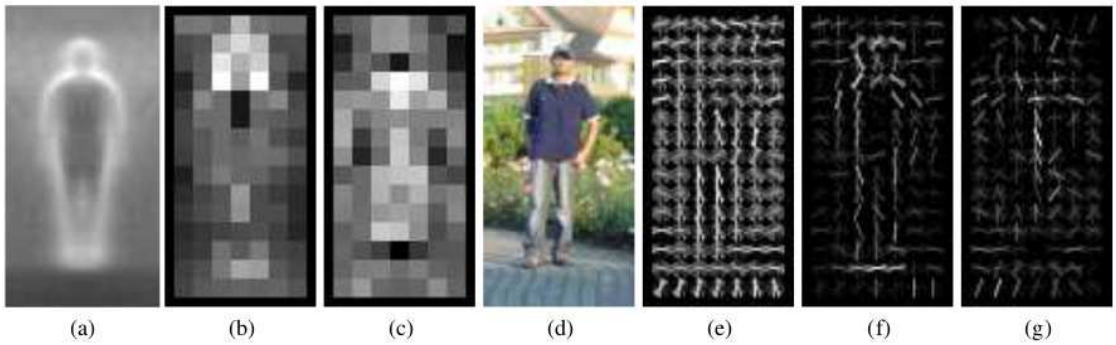


Figure 2.5: (a) Average gradient image extracted from training sets. (b) Positive SVM weights centered on blocks of images. (c) Negative SVM weights. (d) Test image. (e) Computed HOG descriptor. (f) HOG descriptor weighted by positive SVM weights. (g) HOG descriptor weighted by negative SVM weights.(Figure from [28]).

2.3.3 Part-based models for 2D pose estimation

Most 3D pose estimation frameworks require an intermediary stage for object detection and estimation in which the 2D estimates of body parts are obtained. The state-of-the-art approach towards 2D human pose estimation involves the use of part-based models.

The main idea behind part models dates from Binford's generalized cylinder models [2] and the pictorial structures of Fischler [24] and Felzenszwalb [29]: objects can be modeled as a set of part templates which can be arranged in deformable configurations. Part templates reflect local object appearance while the configuration captures geometrical, spring-like connections between pairs of parts.

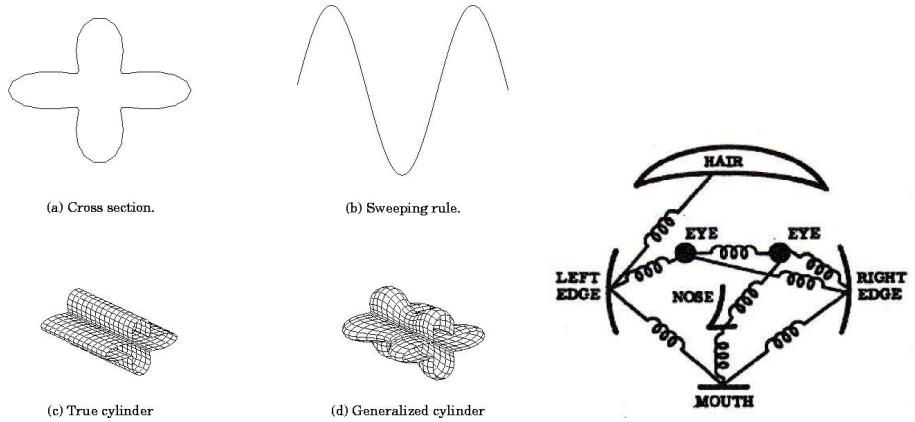


Figure 2.6: On the left, definition of Binford's generalized cylinders [2] by two functions: the cross-section (a) and the sweeping rule (b), which modulates the width of the cylinder (c) through the transverse axis to create the final cylinder (d). On the right, Fischler's pictorial structure [24] which models object using local part templates and geometric constraints, visualized by strings.

Depending on the connectivity representation, different types of part-based models have been proposed over the years. One of the first successful models is the *Constellation model* introduced by Fergus et al [10], which presents full-connectivity, between any two pairs of parts. The model was introduced for an unsupervised learning framework for object classification. Objects are represented by estimating a joint appearance and shape distribution of their parts based on all aspects of the object: shape, appearance, occlusion and relative scale. As a result, the model is very flexible, but it requires a high number of parameters and the evaluation is too computationally expensive: for a k-part model, the complexity is $O(N^k)$.

Another approach is the use of *Star models*, where each part is connected only to a root reference part and is independent of all other part locations, leading to a inference complexity of $O(N^2)$. The approach is called Implicit shape model [21] as it implicitly encodes a large vocabulary of parts, respective to the reference part. Felzenszwalb et al

[9] use a star-shaped model defined by a root filter and part-filters based on the HOG descriptor [28] and the associated deformation, modeling visual appearance at difference scales as shown in 2.7.

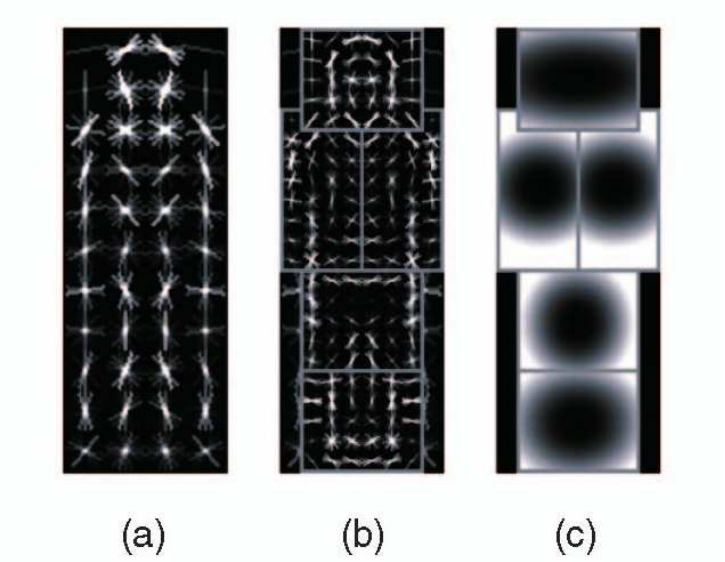


Figure 2.7: Single-component model defined by a root filter (a), higher resolution part-filters specifying weights for HOG features (b) and spatial model representing weights associated with parts for placing their centers at different locations relative to the root (c).(Figure from [9]).

Tree models are a generalization of star-shaped models which allow for efficient inference of $O(N^2)$, and relations between parts are of parent-child nature. The configuration cost is computed according to a coordinate system defined by each parent. One limitation of the model is the double-counting phenomena, where two child-parts are partially overlaid as the geometrical positions are estimated independently.

2.3.4 Machine learning for pose estimation

The ill-posed problem of 3D pose estimation can be solved by stating initial constraints such as possible camera viewpoints, pruning unnatural poses, and using various statistical learning frameworks to learn certain image features or improve detected poses by tracking: support vector machines (SVMs), relevance vector machines (RVMs), Gaussian processes, adaboost, nearest-neighbor, particle filters, hidden Markov models etc. The following subsections cover theoretical aspects related to SVMs and Gaussian processes, which have proven high performance as fine learning tools in human motion analysis.

Support vector machines

Over the past two decades the scientific community has shown a very high interest in kernel machines, most of it being focused on support vector machines (SVM) since they

were introduced by Boser et al [4] in 1992. SVMs represent a very popular method for binary classification, but they are also used for multi-class classification and regression analysis. In the classification context, an SVM can be seen as an extension of a single layer neural network which tries to find a way of separating data using any hyperplane, without measuring how data is separated. SVMs introduce a technical measure, called margin, which represents the distance from the hyperplane to the closest point in the dataset. The hyperplane that separates data most clearly is the one that is set as far away from either class, or the one that generates the highest margins as shown in Figure 2.8.

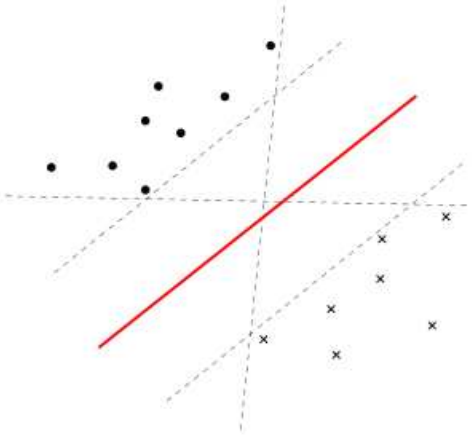


Figure 2.8: Dashed lines represent different possible hyperplanes for separating the two classes. The red line represents the hyperplane with the largest margin and best generalization.

In the linear definition, given a training set D of n observations:

$$D = \{(x_i, y_i)|x_i \in \Re^p, y_i \in \{-1, 1\}\}_{i=1}^{n} \tag{2.1}$$

, where x_i is a p-dimensional vector and y_i is a binary label, an SVM finds a maximum margin separating hyperplane. For each training dataset, SVMs learn a parameter consisting of a vector w, which represents the normal to the maximum margin hyperplane. From the optimization point of view, there are two formulations for SVMs.

- *Primal SVM formulation*

  The hyperplane that is characterized by the normal vector w which maximizes the margins for the training set D can be found by solving the following quadratic problem (QP):

$$minimize_w \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} max(0, 1 - y(w \cdot x)) \tag{2.2}$$

, where $\lambda \geq 0$ is the regularization parameter that scales $\|w\|^2$. Instead, a scaling parameter for the empirical loss term can be used: $C = \frac{1}{n\lambda}$.

- *Dual SVM formulation*

  The primal formulation implies a linear classifier in which the hyperplane is defined in the same space as the data. Nonlinear classifiers can be represented by an SVM using the kernel trick: mapping the data points x_i to a higher dimensional space F using a function $\phi : X \to F$ and finding a linear classifier in the high-dimensional space. The kernel trick allows defining a kernel matrix $K_{ij} = \phi(x_i) \cdot \phi(x_j)$ such that any data point dot products can be replaced by the associated element in the matrix. According to Mercer's theorem, any positive semi-definite matrix can be a kernel matrix. Also, the mapping σ is not required as long as a valid kernel matrix has been found. Using the Lagrange multipliers α_i, the formulation of the dual optimization problem is:

$$maximize_\alpha \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \; such \; that \; 0 \leq \alpha_i \leq \frac{1}{n\lambda} \tag{2.3}$$

  When a kernel is introduced the formulation becomes:

$$maximize_\alpha \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \; such \; that \; 0 \leq \alpha_i \leq \frac{1}{n\lambda} \tag{2.4}$$

  The support vectors represent the training points which are either misclassified or fall inside the margin region. According to the dual formulation, support vectors correspond to the α_i multipliers that have non-zero values and the optimal weight vector represents a linear combination of them. Therefore, training an SVM implies finding the support vectors.

Gaussian Processes

The basic theory for prediction with Gaussian processes (GP) dates back from 1940's, in the work of Wiener [40] and Kolgomorov [19]. The earliest applications were made in the field of geostatistics, meteorology, spacial statistics and computer experiments. In the last decade, the focus has been directed towards the general regression context and the connection to other learning methods such as support vector machines has been outlined. Moreover, GPs are computationally equivalent to many known models, and represent general cases of Bayesian linear models, spline models, or neural networks.

Formally, GPs are defined as collections of random variables, any finite number of which have a joint Gaussian distribution. They extend multivariate Gaussian distributions to infinite dimensionality. Using Gaussian processes for prediction problems can be regarded as defining a probability distribution over functions, such that inference takes place directly in the function space-view. The training data observations $y = \{y_1, ..., y_n\}$ are considered samples from the n-variate Gaussian distribution that is associated to a Gaussian process and which is specified by a mean and a covariance function. Usually, it is assumed that the mean of the associated GP is zero and that observations are related using the covariance function $k(x, x')$. The covariance function describes how function values $f(x_1)$ and $f(x_2)$ are correlated, given x_1 and x_2. As the GP regression requires continuous interpolation between known input data, a continuous covariance is also needed. A typical choice for the covariance function is the squared exponential:

$$k(x, x') = \sigma_f^2 \exp \frac{-(x - x')^2}{2l^2} \tag{2.5}$$

,where σ_f represents the amplitude or the maximum allowable covariance, reached when $x \approx x'$ and $f(x)$ is very close to $f(x')$, and l represents the length parameter which influences the separation effect between input values. If a new input data x is distant from x' then $k(x, x') \approx 0$ and the observation x' will have a negligible effect upon the interpolation.

Given the independent variable x with the set of known observations y, then the estimate y_* for the new value x_* is found knowing that data can be represented as a sample of the multivariate Gaussian distribution:

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K & K_*^T \\ K_* & K_{**} \end{bmatrix}\right) \tag{2.6}$$

where K represents the covariance matrix, K_* represents the row in the matrix that corresponds to x_* and $K_{**} = k(x_*, x_*)$. We are searching for the conditional probability $p(y_*|y)$ which follows a Gaussian distribution:

$$y_*|y \sim N(K_* K_*^{-1} y, K_{**} - K_* K_*^{-1} K_*^T) \tag{2.7}$$

which leads to the mean and variance values:

$$\overline{y}_* = K_* K_*^{-1} y \tag{2.8}$$

$$var(y_*) = K_{**} - K_* K_*^{-1} K_*^T. \tag{2.9}$$

2.4 HumanEva benchmark

Given the extensive amount of work on 3D pose estimation, a benchmark database is needed in order to create a ranking of the results obtained by different approaches. An

important contribution in this regard is the HumanEva dataset, introduced by Sigal et al [35] using a hardware system able to capture synchronized video and 3D ground truth motion.

The HumanEva datasets contain 4 subjects performing 3 trials of a set of 5 predefined actions: walking, boxing, jogging, gestures and throw-catch. All data is divided into subsets of training, validation and testing. The body model presents 15 joints and data is captured using 5 synchronized cameras used from different viewpoints as showed in Figure 2.9.

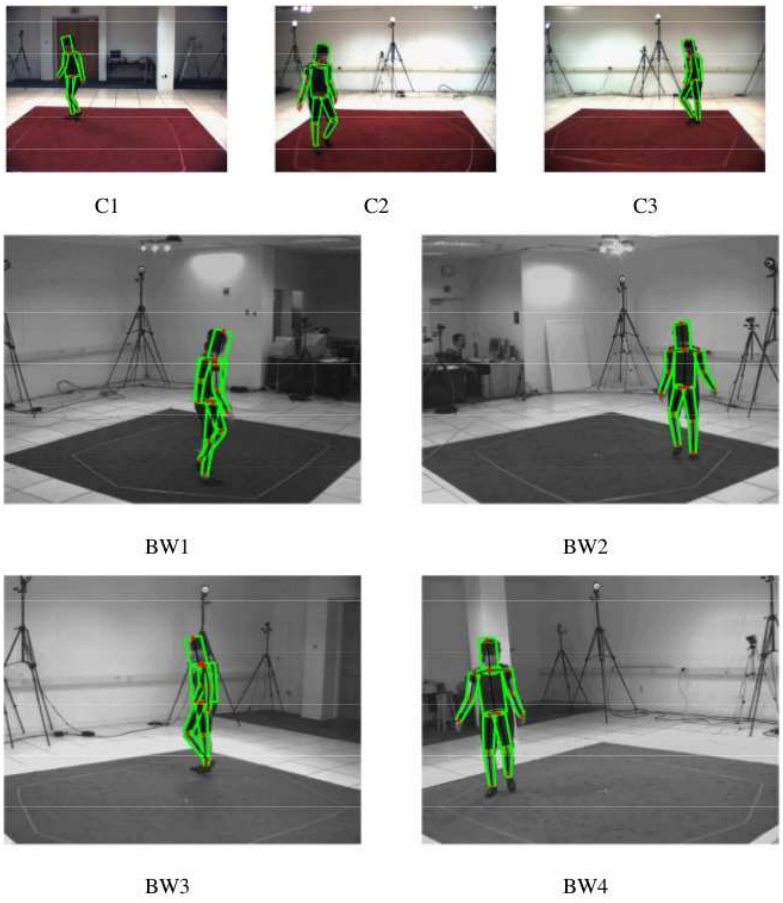


Figure 2.9: Images of a subject performing walking action from synchronized video cameras with overlaid motion capture data.

In addition to the dataset, a baseline Bayesian filtering algorithm is provided for motion tracking. The performance is analyzed according to a variety of parameters allowing the user to experiment with new settings and motion models. Also, a standard set of error measures based on Euclidean distances between points is defined for 2D and

3D pose estimation and tracking algorithms evaluation.

2.5 Outline of proposed framework

Considering the various methods and notions outlined in the previous sections, the thesis presents a learning-based approach towards 3D pose estimation using monocular image sequences. The system presented is fully automatic, marker-less and does not require camera calibration nor background information. In comparison to all presented related works which require silhouette extraction or 2D ground truth configurations, the system proposed is complete, in the sense that it takes raw image sequences as input and outputs the estimated 3D configurations.

Overall, it can be described as a three-stage framework composed of a 2D human detector, a motion smoother and a 3D regressor as pictured in Figure 2.10.

Figure 2.10: Outline of the 3-stage framework.

The 2D detector is based on Ramanan's articulated mixture model [41] as it obtains state of the art results on standard benchmarks and it also benefits of a very fast implementation in comparison to previous works. For each frame in the video sequence, image features are mapped to 2D poses using a flexible mixture model which captures co-occurrence relations between body parts. The 2D detector processes singular frames from the video sequence input and outputs vectors of 2D joint coordinates following a kinematic model representation. The results of the 2D detector are improved using temporal smoothing techniques, which works on an optimal window size of consecutive frames. On the other hand, as the 2D detector works on a single-frame basis, the framework can be used to recover 3D poses from singular monocular images, smoothing being unnecessary.

In the final stage of the framework, the new vectors of 2D coordinates are normalized and mapped to 3D poses using a Gaussian process regressor. As stated in the related work section, many 3D pose estimation frameworks use Gaussian process regression as GPs represent a flexible learning approach, capable of modeling the inherent non-linearity found in human motion. Comparative to SVMs and RVMs, they provide a better predicting accuracy at the cost of longer training times. Experiments are conducted systematically on the HumanEva benchmark, for each stage of the framework, on different types of activities and camera viewpoints. The final 3D estimates are compared with results obtained using different methods of mapping image features to

Gaussian process inputs.

The next chapters are organized according to the described framework. Chapter 3 describes the 2D human detector based on the articulated mixture model from [41] and presents experiments and results on different motion types and viewpoints from the HumanEva dataset. The chapter also includes a description of the motion smoother used to improve the results of the detector for image sequences.

In chapter 4, the Gaussian process regressor is described and comparisons are made between other approaches which use the same regressor to generate 3D estimates. The chosen 3D human body representation is described and results are interpreted in 3D space.

Lastly, Chapter 5 outlines conclusions based on the methods used and all experimental results obtained and discusses future work.

Chapter 3

2D Human Detection and Smoothing

This chapter describes the method used for human pose estimation and motion smoothing in static images. The 2D human detector is based on the solution presented by Ramanan et al. in [41], which uses a novel representation of part models, while outperforming past work and being faster by orders of magnitude. The smoothing technique is based on Garcia's work from [12] on robust DCT-based penalized least squares smoothing.

3.1 2D Human Detector

The dominant approach towards human pose estimation implies articulated models in which parts are described by pixel location and orientation. In the pictorial structure framework, objects are decomposable into local part templates described by geometric constraints. The approach used by Ramanan introduces a model based on a mixture of non-oriented pictorial structures, tuned to represent particular classes of parts. The variability of poses and appearances can be described using mixtures of templates for each body part used in a model.

Current approaches involving object recognition are build on mixtures of star structured [9] or implicitly-defined models [5]. Ramanan's model adds certain constraints to the classic spring model [24] favoring particular combinations of part types. Objects are represented using tree relational graphs which capture geometric or semantic constraints. The usage of tree models allows for efficient learning and inference, but also for the phenomena of double-counting. From the object detection point of view, the model is most similar to that of pictorial structures based on mixtures of parts [9], [29], [8].

3.1.1 Model

The mixture model implies mixtures of parts or part types, which may include orientations of parts (horizontally or vertically oriented limbs) or may extend semantic types (an open or closed hand). Similar to the star-structured part-based model in [9], this mixture model involves a set of filters that are applied to a feature map extracted from the analyzed image. A dense feature map is obtained by extracting the HOG features [28] from equally sized image patches, and by iterating this process at different scales of the image, a feature pyramid combines all the features maps.

Generally, a filter is a patch template defined by an array of weight vectors and it represents a certain part type in the model. The score of a filter is obtained by computing the dot product of that filter and a same-sized subwindow of a feature map (Figure 3.1).

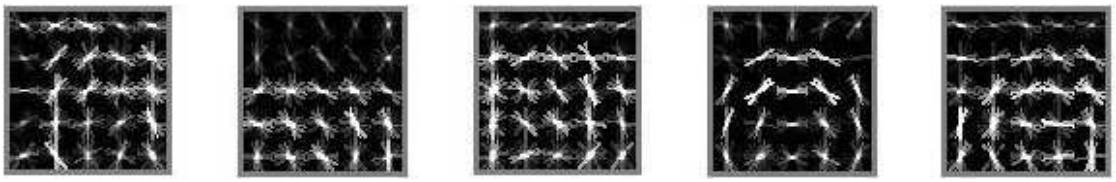


Figure 3.1: Filters associated to a mixture of head part-types. Each filter favors a particular orientation of the head.

An n-part model is defined in a $(b, d, f, C_1, ..., C_n)$ format, where b is the vector of bias values for each mixture component, d is the vector of deformation values, f is the vector of filter templates and C_i is the i-th mixture part. A mixture component C_i described by $(b_{id}, d_{id}, f_{id}, par)$, where the first three terms represent the vectors of indices in b, d and f at which the respective values of the part-types are found, and par represents the index of the part's parent.

A configuration of parts for an n-part model specifies which part type is used from each mixture and its location relative to the pixel grid. Considering an image I and $p_i(x, y)$ a pixel located in part i, we call t_i its part-type, where $i \in 1, ..., K, p_i \in 1, ..., L$ and $t_i \in 1, ..., T$, K being the number of parts, L the number of pixels and T the number of part-types used in the model.

Each possible configuration of parts found in an image receives a certain score and a person is considered detected when the highest scoring configuration is found (above a determined threshold). The score of a configuration of parts is computed according to three model components: co-occurrence (3.1), appearance and deformation (3.2). A general mixture model can be described as a K-node graph $G = (V, E)$, where nodes represent parts and edges represent strong relations between parts. The graph is built by manually defining the edge structure E (Figure 3.2).

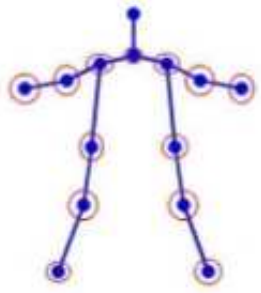


Figure 3.2: Tree visualization describing a full-body 14-part model(Figure from [41]).

The co-occurrence component measures the part-type compatibility by adding local and pairwise scores[41]:

$$S(t) = \sum_{i \in V} b_i^{t_i} + \sum_{i,j \in E} b_{ij}^{t_i,t_i} \tag{3.1}$$

The first term favors certain type assignments for each part, while the second favors part co-occurrences. For example, parts placed on rigid limbs will maintain similar orientations if the part types correspond to orientations.

The full score equation can be written as[41]:

$$S(I,p,t) = S(t) + \sum_{i \in V} w_i^{t_i} \cdot \Phi(I,p_i) + \sum_{i,j \in E} w_{ij}^{t_i,t_i} \cdot \Psi(p_i - p_j) \tag{3.2}$$

where the second term expresses the local appearance score by placing a weight template of part i, tuned for part-type t_i at p_i and the third term expresses the deformation score as relative location between connected parts i and j. The appearance model is based on dot product between the weight templates and the extracted feature vector at p_i, $\Phi(I,p_i)$ (in our case, the HOG descriptor). The deformation model is based on the dot product between the part-type pair assignment parameters $w_{ij}^{t_i,t_i}$ and the relative location between pair parts, computed as:

$$\Psi(p_i - p_j) = [dx\ dx^2\ dy\ dy^2]^T \tag{3.3}$$

,where $dx = x_i - x_j$ and $dy = y_i - y_j$.

Starting from the general mixture model, a few particular cases can be used to describe other known models. For example, if $T = 1$, the model describes the standard pictorial structure [29]. Also, semantic part models [8] are obtained if the part-types capture semantics instead of visual features by using the same part-type pair parameters:

$$w_{ij}^{t_i,t_j} = w_{ij} \tag{3.4}$$

The mixture model of deformable parts described in [9] restricts the co-occurrence model such that all parts share the same type and the configuration score is a sum of biases and local appearance and deformation scores. Our model is also a simplified version in which the deformation score of a part depends only on the relative location of parts and analyzed pair-type, but not on parent-type:

$$w_{ij}^{t_i,t_i} = w_{ij}^{t_i} \tag{3.5}$$

As the model described is highly customizable, a more efficient model structure can be found by varying T and K. For a full-body model, a 14 and a 26 part-model are used, the results showing increased performance in the latter, due to the capture of additional orientation. The model used in this project represents a full human body and is composed of 26 parts, including midpoints between limbs. Also using a variable number (5 or 6) of part-types results in better performance, as it covers an extended, variable set of poses.

3.1.2 Inference

Inference using the mixture model described is obtained by retrieving the highest-scoring configuration, precisely by maximizing $S(I, p, t)$ (3.2) over all parts and part-types. Building the associated relational graph G as a tree allows for efficient inference with dynamic programming. For this, the score of part i is computed depending on the part-type and pixel location:

$$score_i(t_i, p_i) = b_i^{t_i} + w_{t_i}^i \cdot \Phi(I, p_i) + \sum_{k \in kids(i)} m_k(t_i, p_i) \tag{3.6}$$

where $kids(i)$ is the set of children of part i and $m_k(t_i, p_i)$ is the message a child k sends to its parent i, which represents the maximum scoring location and type of child part k for a given location and type of parent part i:

$$m_i(t_j, p_j) = \max_{t_i} b_{ij}^{t_i,t_j} + \max_{p_i} score(t_i, p_i) + w_{ij}^{t_i,t_j} \cdot \Psi(p_i - p_j) \tag{3.7}$$

As all messages are received by the root part, $score_1(t_1, p_1)$ represents the maximum score of all possible configurations that can be obtained at that particular location and root type. The global highest score is found by thresholding and applying non-maximum suppression over overlapping configurations. Starting from a root, all the locations and part-types of configuration parts are found by backtracking, if the $argmax$ indices have

been stored.

The most computationally expensive portion of the dynamic programming algorithm is occupied by computing the child messages. A maximum score is computed among $L*T$ child locations and types for $L*T$ parent parts, reaching the complexity of $O(L^2*T^2)$ per part. Setting $\Psi(p_i - p_j)$ (3.3) as a quadratic function allows computing the inner maximization in (3.7) using a distance transform [29] for each combination of parent-child part types, reducing the complexity to $O(L*T^2)$ per part. In our case (3.5), only T springs exist per part, reducing (3.7) to $O(L*T)$. Also, local appearance scores $w^i_{t_i} \cdot \Phi(I, p_i)$ are linear and can be efficiently computed using convolutions.

3.1.3 Learning

The learning process of the articulated mixture model is done in a supervised context. Sets of positive images with manually annotated limbs and a set of negative images are provided for this purpose. The implemented model includes two components: detection and pose estimation, which means that on the one hand it generates high scores on ground-truth poses and low scores on negative images and on the other hand, it outputs a set of parameters containing limb locations. The solution used for training such a model with labeled data is a structural SVM, an extended version of SVMs which allows for structured output classification.

Considering the labeled positive dataset I_n, pn, t_n and the negative dataset I_n, where $z_n = (p_n, t_n)$ is a ground-truth configuration and $\beta = (\omega, b)$ represents the linear model parameters, then the scoring function (3.2) becomes:

$$S(I, z) = \beta \cdot \Psi(I, z) \tag{3.8}$$

Thus, the SVM objective function can be written as:

$$\min_{\omega, \xi \geq 0} \frac{1}{2}\beta \cdot \beta + C \sum_n \xi_n \tag{3.9}$$

$$\begin{aligned} \text{s.t. } \forall n \in pos \quad & \beta \cdot \Phi(I_n, z_n) \geq 1 - \xi_n \\ \forall n \in neg, \forall z \quad & \beta \cdot \Phi(I_n, z) \leq -1 + \xi_n \end{aligned}$$

where the slack variables ξ_n penalize the constraints of the objective function and the model parameters can be obtained from the arguments of the minimization. The form of the optimized function leads to a problem of quadratic programming, which in this case is solved using dual coordinate-descent.

As the positive images contain manually annotated joints and the part locations are obtained according to this data, the labels actually represent part locations but not part-types, which need to be generated. Because parts can be found at different locations relative to their parents in the relational graph G, defining articulation by capturing orientation implies associating part-types to the relative locations. Orientation of parts depends on position as, for example, a horizontally-oriented hand is found next to the elbow, while a vertically-oriented hand is found under the elbow. In our case, part-types are derived by clustering the values of each part's relative position to their parent using K-means with $K = T$. Each cluster obtained corresponds to a part-type based on orientation (Figure 3.3). Another solution would be introducing a latent SVM which takes the part-types as latent variables. This approach rests as possible future work.

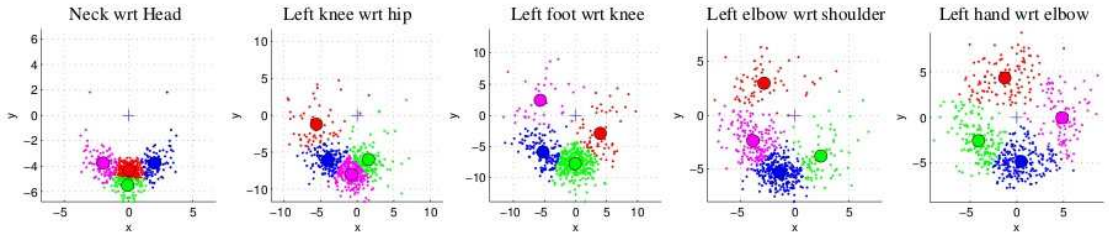


Figure 3.3: Clusters used to generate mixture labels during training for T=4. Part-types represent different orientations of parts relative to their parents (Figure from [41]).

3.1.4 Experiments

The proposed model is tested using the Image Parse [32] and Buffy datasets [11]. The Image Parse set contains 305 annotated images of highly-articulated full-body poses, and the Buffy set contains 748 annotated video frames of upper-body poses. For these experiments different models are built to compare performance. Examples of models are pictured in Figure 3.4. As opposed to previous approaches, the error on these datasets is reduced by up to 50%, requiring a processing time at the order of units of seconds on these datasets.

The most efficient models are the 18-part model and the 26-part model for upper and full-body detection, respectively. For our project, the 26-part full-body model is chosen and further experiments are carried on the Parse and HumanEva dataset to outline its performance (Figures 3.5 and 3.9). The following figures present successful detections, but also failing situations are encountered. Generally, horizontal poses and multiple persons are not detected so these will be considered limitations of the detector. To overcome problems such as double-counting phenomena and limb misdetection, and also to obtain a more continuous appearance of 2D joint positions over consecutive frames, motion smoothing is required. The solution used in our framework and results are covered in the next section.

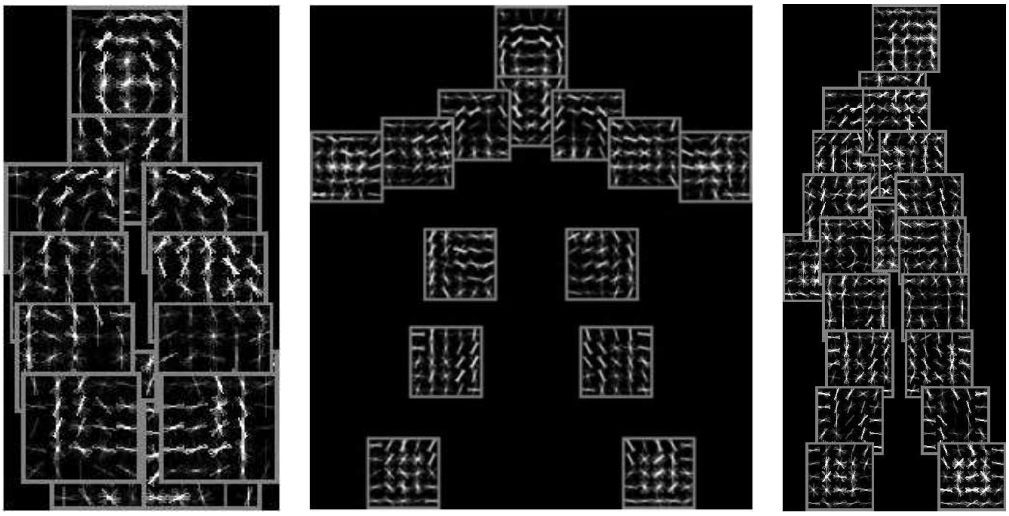


Figure 3.4: A visualization based on part templates for an 18-part upper-body model, 14-part and 26-part full-body model. Parts are represented by 5x5 HOG templates, placed at their highest scoring locations for a given root position and type.

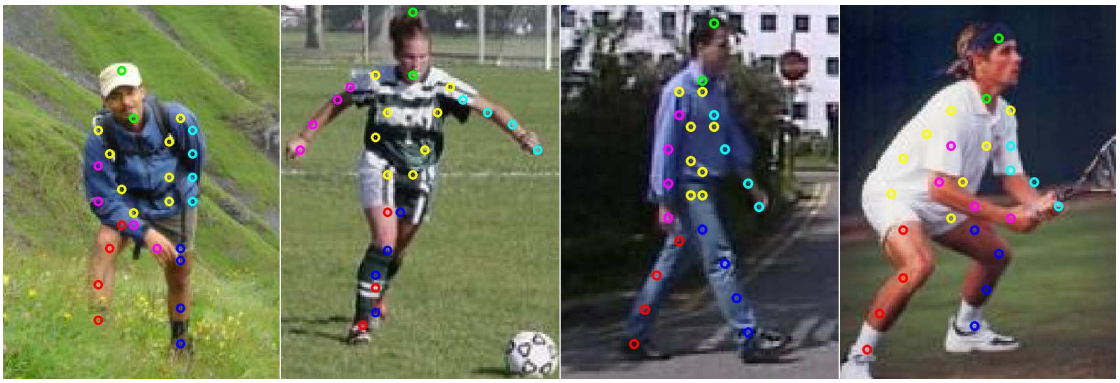


Figure 3.5: Examples of successful detections on the Parse dataset.

3.2 2D Motion Smoothing

One solution to obtain a motion model is incorporating the temporal information in the articulated mixture model, obtaining a spatio-temporal part-based model in which the deformation score receives also a temporal term. The spatio-temporal deformation adds a new constraint favoring co-occurrences between identical parts in consecutive frames. Specifically, the left upper leg part must lie next to the hip in the current frame and also next to the left upper leg part in the previous frame. However, the temporal looping

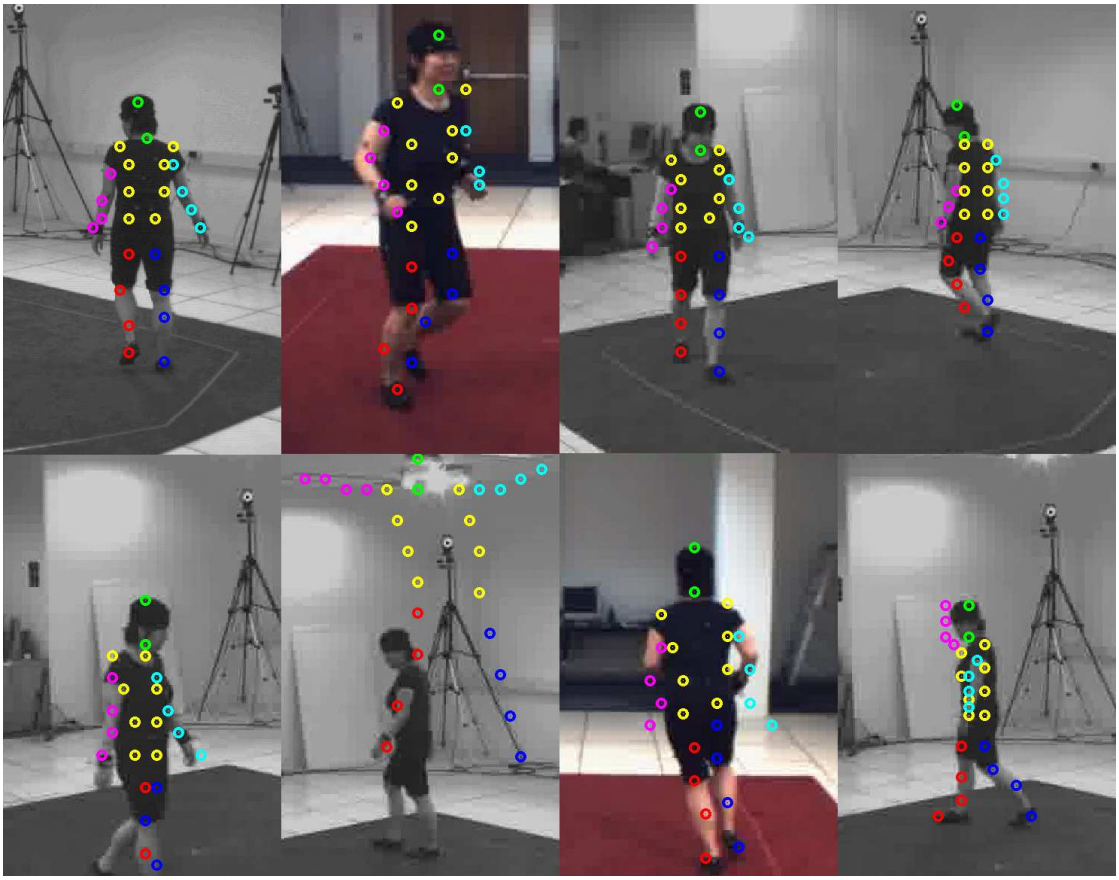


Figure 3.6: HumanEva examples: The top row presents successful detection on different actions and cameras. The bottom row presents failing situations: double-counting, body or limb misdetection, self-occlusion.

added in the model makes inference difficult and drifting problems may occur.

A more robust approach is similar to Andriluka's tracking-by-detection [23] solution: obtaining the 2D detections over all frames and then running a smoothing algorithm. By definition, smoothing tries to estimate the current pose using information from all the frames in the sequence. Therefore, it provides increased accuracy in comparison to filtering approaches which only use information propagated until the current frame. The smoothing technique used in the framework is based on the automatic algorithm described in [12].

3.2.1 Algorithm description

In statistics, smoothing a data set represents finding an estimating function that captures data patterns and is able to reduce experimental noise or fine-scale structures. Given a

one-dimensional noisy signal y:

$$y = \hat{y} + \epsilon \tag{3.10}$$

where ϵ is a zero mean Gaussian noise with unknown variance, then $\hat{y}$ represents the smoothed signal, which has a continuous derivates up to an order greater than 2 over the domain of the signal. The goal of smoothing is finding the best estimate for $\hat{y}$.

The algorithm is based on the classical approach of penalized least squares regression, which approximates $\hat{y}$ by trying to minimize:

$$F(\hat{y}) = RSS + sP(\hat{y}) = \|\hat{y} - y\|^2 + s\|D\hat{y}\|^2 \tag{3.11}$$

,where RSS is the residual sum of squares, s is the a parameter which controls the smoothing degree and D is a tridiagonal matrix given by the steps between $\hat{y}_i$ and $\hat{y}_{i+1}$.

The minimization retrieves:

$$\hat{y} = (I_n + sD^T D)^{-1} y = H(s)y \tag{3.12}$$

,where H(s) is called hat matrix. Smoothing becomes fully automated when the parameter s can be estimated using the generalized cross-validation method (GCV), minimizing:

$$GCV(s) \equiv \frac{RSS/n}{(1 - Tr(H)/n)^2} = n\sum_{i=1}^{n}(\hat{y}_i - y_i)^2/(n - \sum_{i=1}^{n}(1 + s\lambda_i)^{-1})^2 \tag{3.13}$$

,where λ_i represent the eigenvectors of $D^T D$, used to compute $Tr(H)$.

When data is evenly spaced D changes such that its eigenvectors become $\lambda_i = -2 + 2\cos((i-1)\pi/n)$ leading to a DCT-based formulation of the $GVC(s)$ term, which is very fast to compute:

$$GCV(s) = \frac{n\sum_{i=1}^{n}(\frac{1}{1 + s\lambda_i^2} - 1)^2 DCT_i^2(y)}{(n - \sum_{i=1}^{n}\frac{1}{1+s\lambda_i^2})^2} \tag{3.14}$$

Due to the presence of measurement errors, it is convenient for the algorithm to support weighed or missing data. This is obtained by associating specific inputs with low weights $w_i \in [0, 1]$, organized in a diagonal matrix $W = diag(w_i)$ such that RSS becomes:

$$wRSS = \|W^{1/2}(\hat{y} - y)\|^2 \tag{3.15}$$

When y_i is missing, then $w_i = 0$ and y_i is assigned with a value estimated by interpolation and using the entire dataset. Computation time increases as $\hat{y}$ is computed at each minimization of the GCV. For n inputs with n_{miss} missing data, the GCV score becomes:

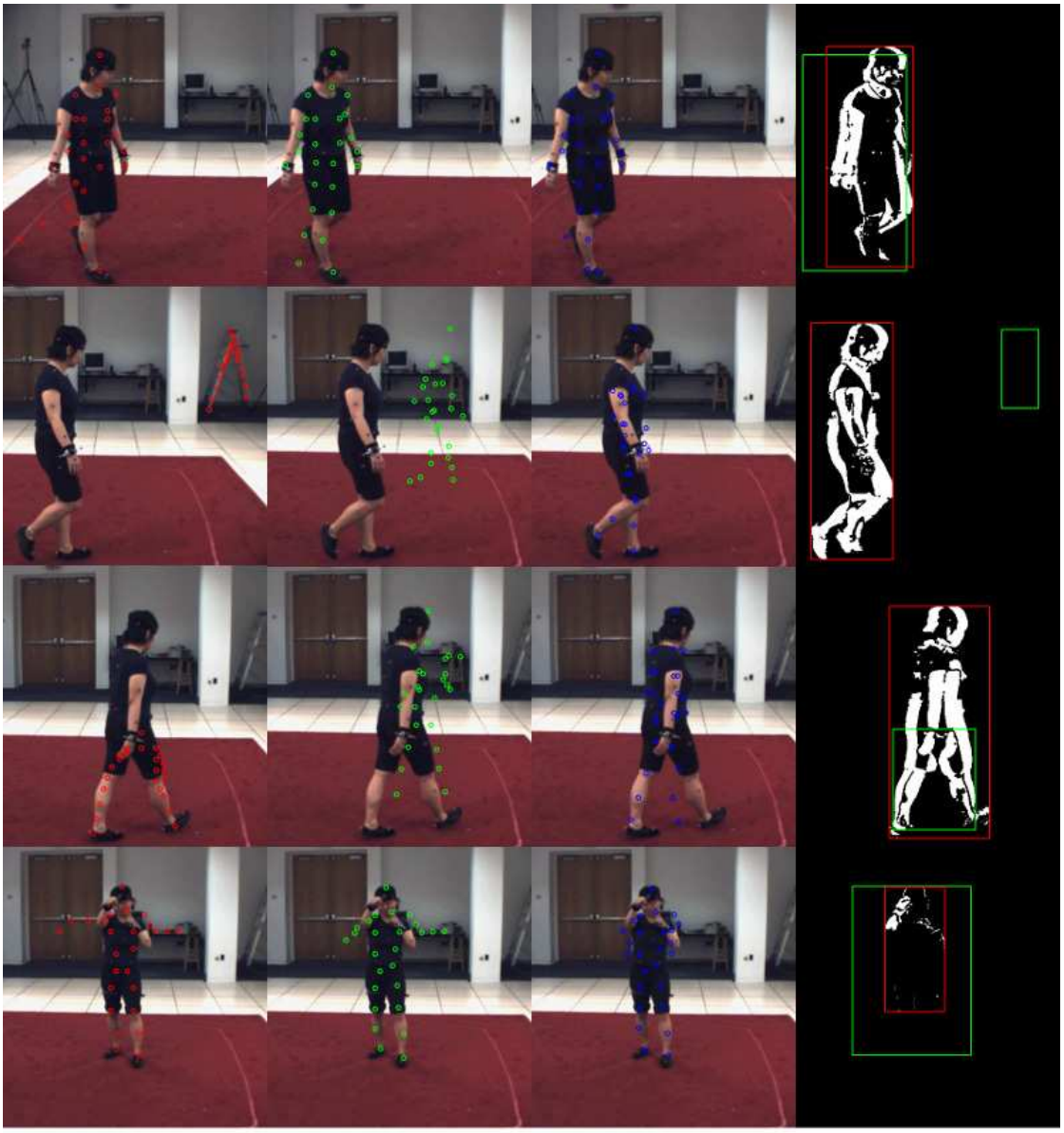
$$GCV(s) = \frac{wRSS/(n - n_{miss})}{(1 - Tr(H)/n)^2} = \frac{\|W^{1/2}(\hat{y} - y)\|^2/(n - n_{miss})}{(1 - Tr(H)/n)^2} \tag{3.16}$$

The algorithm also benefits of a robust version, being able of canceling the effects of outliers and high leverage points to which penalized least squares are usually sensitive. This can be done by iteratively reassigning low weights to such points. Current residuals are updated at each iteration until they remain unchanged.

3.2.2 Experiments

Smoothing is introduced to improve the results obtained by the 2D human detector, by creating a more continuous set of coordinates and eliminating outliers. The dataset contains inherent discontinuities as frame detections are independent. Moreover outliers exist in the data for various reasons: the full body is misdetected over a small number of frames, double-counting in the case of legs, background clutter or self-occlusion leads to limb misdetection.

As the input image sequences are composed of consecutive frames, the data may be considered uniformly spaced in the temporal dimension. After running detection over all the frames, robust smoothing is performed separately for the vectors of x an y joint coordinates. As the frames are consecutive, motions performed on a specific number of frames can be extracted by image differencing. If we assume that that the camera is static and the person is the only moving object in the scene, a correct detected pose should be localized in the area covered by pixel displacements. Otherwise, more complex optical flow methods should be considered. Therefore, to improve the accuracy of the detector, image differencing is performed on a small window size of frames to compare the relative position of the detected pose to the area covered by motions. The window size used should be changed automatically according to the framerate of the video sequence, for example the size is 5 for the framerate of 120 Hz used on the HumanEva dataset. The bounding boxes which cover the current frame pose and the motions are computed and joint data is considered missing if the respective joints fall outside the motions coverage or if there is a substantial difference between the detections and motions bounding boxes (Figure 3.7). We will call the approach weighted robust smoothing as the missing data will be associated with zero weights.



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

Figure 3.7: Detections without smoothing (a) Detections with robust smoothing (b) Detections with weighted robust smoothing (c) Motions (red) and detected pose (green) bounding boxes (d).

3.3 Results

Experiments are carried on the HumanEva dataset for subject 1, camera 1, all motion types. All detection data (unsmoothed detections, robust smoothing and weighted robust smoothing) is reprojected from the 26-part body model to the 15 markers-based

model used in the dataset. Using the 2D pixel error measure provided by the HumanEva framework, errors are computed for the full body and for specific joints (head, torso, pelvis, shoulders, wrists)(Table 3.1).

Table 3.1: Mean pose errors and on specific joints are computed for S1, cam1, on all action types. Results are expressed in pixels and they are compared for the smoothing option: no smoothing used, robust or weighted smoothing.

Motion	Sm.	Head	Torso	Pelvis	Shoulders	Wrists	Knees	Mean
W	-	22	24	24	38	62	35	39
	R	17	19	16	33	55	28	33
	WR	9	12	9	26	44	20	25
B	-	9	13	8	12	72	9	21
	R	8	12	5	10	67	8	18
	WR	9	12	**5**	**10**	59	**8**	**17**
G	-	6	12	13	17	52	12	21
	R	5	12	13	17	45	12	19
	WR	**5**	12	13	16	42	13	18
J	-	17	20	18	46	77	34	43
	R	15	17	13	41	69	30	38
	WR	9	11	7	38	54	24	31
TC	-	21	25	27	29	58	31	36
	R	18	20	18	19	40	23	25
	WR	6	**10**	11	13	**31**	15	18

Results show that the robust smoothing with missing data approach outperforms the other methods for all actions and joints. Also, joints such as head, torso or pelvis present a better error rate and are more stable than shoulders and knees. The least accurate detections are represented by the wrists as limbs are more prone to be misdetected due to background clutter, foreshortening and self-occlusion. A more detailed visualization of data is presented in Figure 3.8, where specific joint estimation error are plotted for all actions over all the frames.

In actions such as Walking and Jog a series of patterns occur in the shape of rising errors at regular intervals, which is due to the circular nature of motions performed by the subject. For example, higher regular error in estimating the position of the left wrist occurs because of left arm occlusion caused by the camera viewpoint. In motions such as Box, Gestures and ThrowCatch, the subject maintains a static position relative to the camera and only arms are moving. Therefore, joints such as head, hips, knees and ankles maintain a small and constant error rate, while shoulders, elbows and wrists present higher errors.

When joints are misdetected in too many consecutive frames, the error propagates and better detections can be considered outliers and removed in the process of robust smoothing. However, in the case of weighted robust smoothing, the smoother recovers by ignoring detected joints that are too far away from the area covered by motions, as can be seen in Figure 3.9 Gestures, Left Shoulder.

Figure 3.9 presents the mean error obtained for each action in every frame plus the mean error obtained per action for the weighted robust smoothing approach over all frames. As the figures and Table 3.1 show, smoothing improves the results obtained by the detector with an average of 35%. The biggest improvements per action are obtained in the case of Walking and ThrowCatch as the smoother recovers from many full-body misdetections. In the case of joints, head, torso and pelvis estimates present the smallest errors for the same reason.

3.4 Conclusions

The chapter presents a system proposed for 2D human pose estimation which receives a sequence of consecutive images containing one person performing different actions and outputs a vector of smooth pose estimates for 26 joints per frame.

The 2D human detector is based on the flexible mixture model proposed in [7]. The 26-part model version, which is used in the project, outperforms previous works on the Parse dataset by 50% while being faster by orders of magnitude. As the detector works on a per-frame basis, pose estimates are improved using the temporal information by performing weighted robust smoothing on the joint coordinates vectors. This leads to a more continuous estimated motion and to a smaller error rate by removing outliers and poses that are estimated outside the area of the detected motions.

Results obtained on the HumanEva dataset show that detection performance increases with 35% on average after smoothing is applied. The biggest part of the improvement is due to recovery from full-body misdetections. The most accurate estimates are obtained for joints such as head, pelvis, torso, while elbows and wrists are more unstable due to foreshortening effects and self-occlusion. Also, the smallest mean error is obtained on actions such as box or gestures, in which the subject maintains a constant position relative to the camera as opposed to walking and jogging where all body joints change their position drastically over frames, leading to more self-occlusions and limb misdetections.

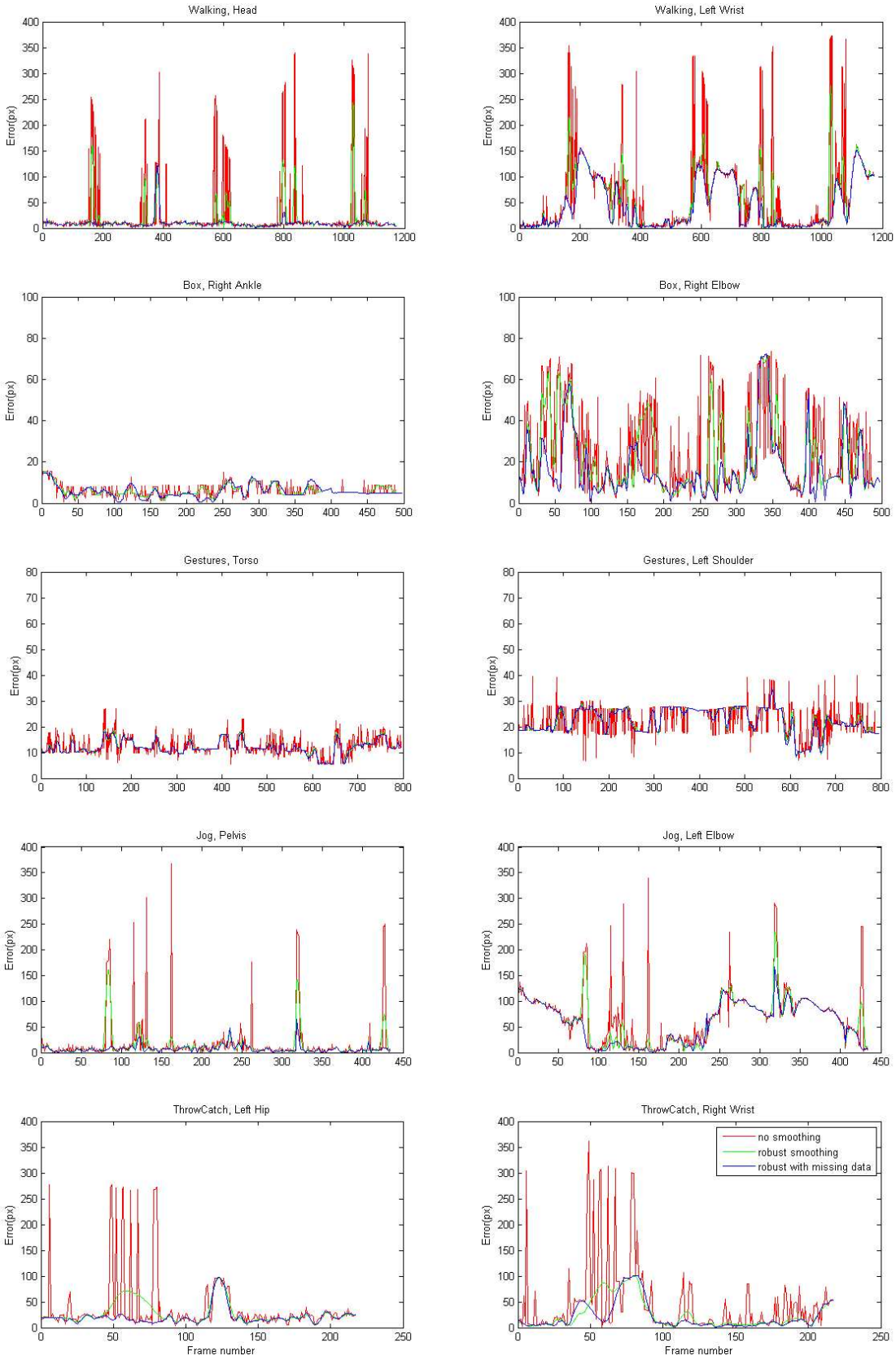


Figure 3.8: Part error plots per joints over all frames and actions.

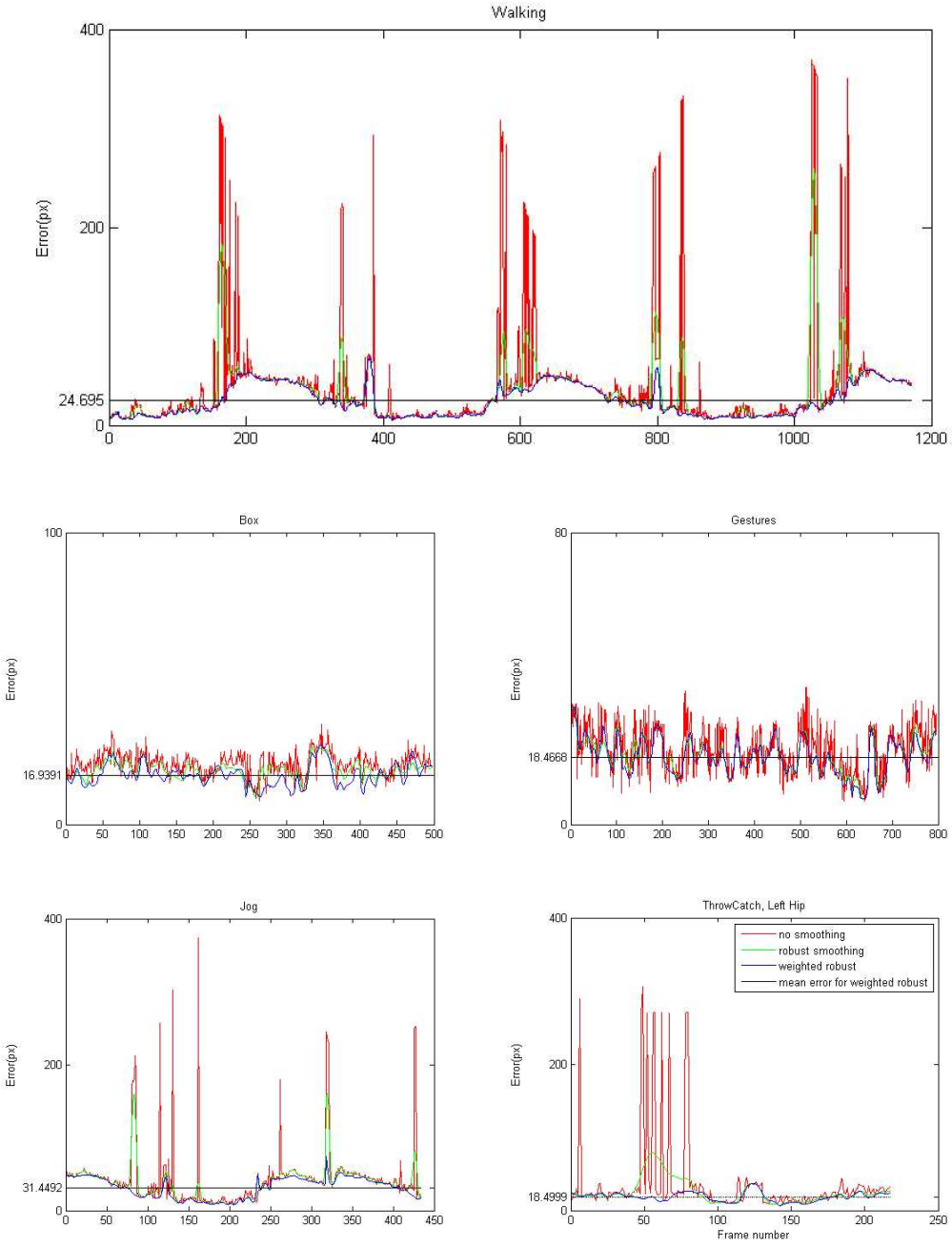


Figure 3.9: Error plots per actions over all frames and mean error for weighted robust smoothing.

Chapter 4

3D Pose Regression

The chapter presents the stage of the 3D human pose estimation framework that uses learning approaches to map 2D to 3D pose estimates. The solution presented uses Gaussian processes and is based on Rasmussen's reference implementation [33]. Given an extensive training set consisting of 2D pose data and the associated 3D poses, the GP regressor tries to find the mapping between inputs and outputs. The following sections discuss representations and settings chosen for the Gaussian process regression.

4.1 Gaussian processes

Using Gaussian processes in a regression context can be interpreted as direct inference in the function space. As a GP is completely specified by its mean $m(x)$, which is usually reduced to zero, and covariance functions $Cov[f(x), f(x')] = k(x, x')$, a process $f(x)$ can be written:

$$\begin{array}{lcl} m(x) & = & E[f(x)] \\ k(x,x') & = & E[(f(x) - m(x))(f(x') - m(x'))] \\ f(x) & \sim & GP(m(x), k(x,x')) \end{array} \tag{4.1}$$

The linear regression model $f(x) = \phi(x)^T w$ with the prior $w \sim N(0, \sum_p)$ provides a basic example of a GP:

$$\begin{array}{lcl} E[f(x)] & = & \phi(x)^T E[w] = 0 \\ E[f(x)f(x')] & = & \phi(x)^T E[ww^T]\phi(x') = \phi(x)^T \sum_p \phi(x') \end{array} \tag{4.2}$$

A crucial ingredient in GP regression is encoding the initial assumptions regarding the function distribution with an appropriate covariance function. It is basic to assume that inputs that are similar or close will have close outputs, so for training points (x, y) that are close to test inputs x', y will be informative about the prediction at x'. As the covariance function measures similarity between input points, choosing an accurate form will be able to determine a better prediction.

A commonly used function in this sense is the squared exponential (SE) which specifies the covariance between random variables:

$$cov(f(x), f(x')) = k(x, x') = exp(-\frac{1}{2} \mid x - x' \mid^2) \tag{4.3}$$

Samples can be drawn from the distribution of functions described by a GP, as seen in Figure 4.1. As the SE covariance function is infinitely differentiable, the corresponding GP is infinitely mean-square differentiable and the sampled functions have a smooth appearance. The shaded area represents the coverage of double standard deviation centered at each input point mean.

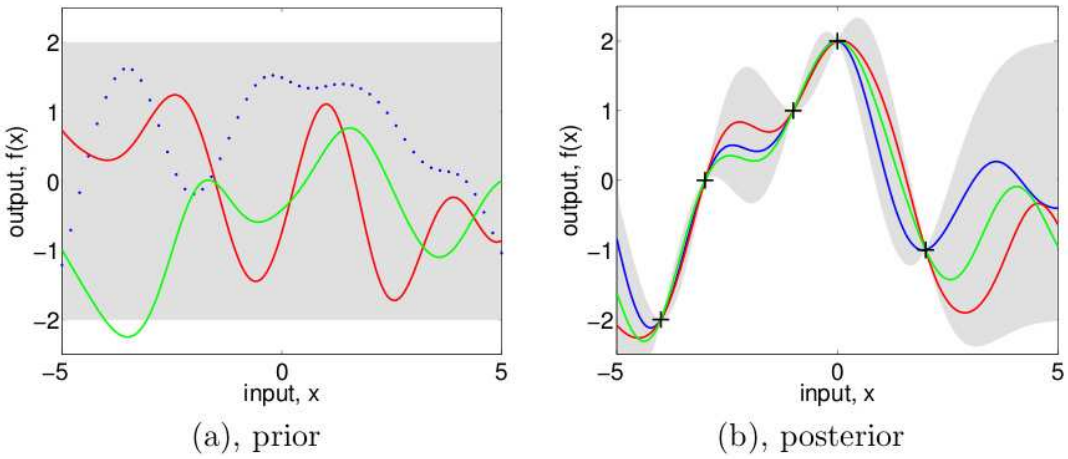


Figure 4.1: Random functions drawn from a GP prior with dots indicating output values generated from a finite set of input points (a). Random functions drawn from the GP posterior, conditioned on 5 noise-free observations (b)(Figure from [33]).

4.2 Training a Gaussian process

The prediction performance on each dataset depends on the chosen parameters for the mean and covariance functions. The process of choosing more accurate functions represents training a GP on the known observations.

For this, a more general formulation is given for the SE covariance function, involving free parameters: the characteristic length-scale l, the signal variance σ_f^2 and the noise variance σ_n^2. These are also called hyperparameters.

$$k(x, x') = \sigma_f^2 exp(-\frac{1}{2l^2}(x - x')^2) + \sigma_n^2 \delta' \tag{4.4}$$

,where δ' is the Kronecker delta which outputs 1 in the case of identical inputs and 0 otherwise. The length-scale l can roughly be interpreted as the amount of displacement needed in input space for a significant change in the function value space. Figure 4.2 shows the effects of varying the hyperparameters of a GP:

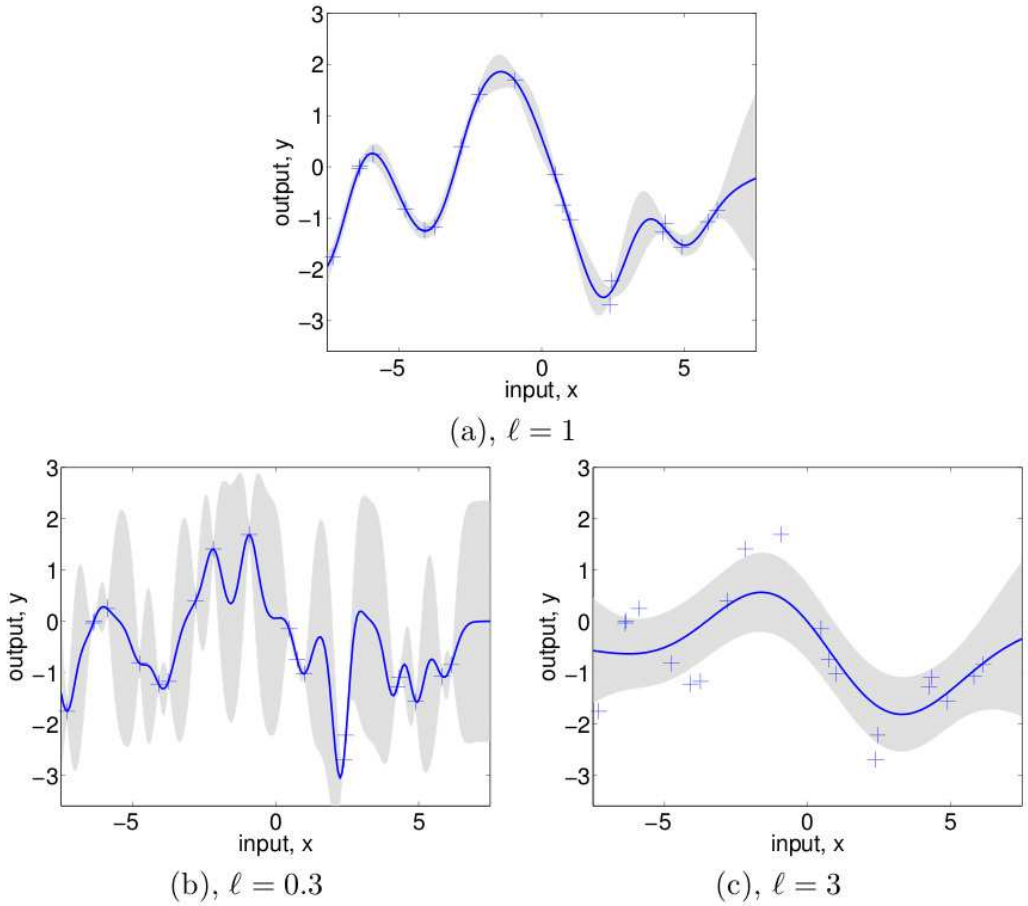


Figure 4.2: Sample function drawn from a GP with hyperparameters (1,1,0.1)(a), (0.3,1.08,0.00005) (b) and (3,1.16,0.89)(c). '+' Symbols represent noise-free observations and the shaded area corresponds to a 95% confidence region. (Figure from [33]).

The figures are obtained by optimizing the signal and noise variances for the same training points but different length-scales. Figure 4.2 (a) shows how the error bars are shorter for input points that are closer to training points. Reducing l in (b) means that the signal is more flexible and noise variance can be reduced also. A shorter length-scale also means that error bars grow more rapidly away from training points. When l is longer (c), the function is varying slowly and the noise level is greater.

In order to make inferences about the hyperparameters, we compute the likelihood

which represents the probability density of the observations given the parameters of the model

$$\log p(y|X) = -\frac{1}{2}y^T(K + \sigma_n^2 I)^{-1}y - \frac{1}{2}\log|K + \sigma_n^2 I|^2 - \frac{n}{2}\log 2\pi \tag{4.5}$$

The hyperparameters are obtained by optimizing the log marginal likelihood described in 4.5 based on the partial derivatives. The log marginal is composed of three terms: a negative quadratic term, which measures the data-fit, a log determinant term, which measures and penalizes the model complexity and a log normalization term. Training is simplified as the trade-off between complexity and data-fit is done automatically.

4.3 Posterior Gaussian process

The previous sections showed how GPs can be used as prior in Bayesian inference and how hyperparameters are computed for the covariance function. Next, the posterior is computed using training data in order to make predictions for testing data. Initially, we consider the case of noise-free observations, where f is the set of known function values for the input x and f_* are the values corresponding to testing data X_*. The joint prior distributions is:

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \tag{4.6}$$

,where K is the covariance matrix for all pairs of training and test points. The posterior distribution over functions is obtained conditioning the prior on the observations:

$$f_*|X_*, X, f \sim N(K(X_*,X)K(X,X)^{-1}f, K(X_*,X_*) - K(X_*,X)K(X,X)^{-1}K(X,X_*)) \tag{4.7}$$

The function values f_* are obtained from the posterior described in 4.6 by evaluating the mean and variance and generating the samples. For the general case of noisy observations, we consider additional Gaussian noise with variance σ_n^2 in the outputs, which is independent of the input points such that:

$$\begin{gathered} cov(f(x), f(x')) = k(x,x') + \sigma_n^2\delta' \; or \; cov(f) = K(X,X') + \sigma_n^2 I \\ \begin{bmatrix} f \\ f_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \end{gathered} \tag{4.8}$$

The conditional distribution leads to the key equations used in GP prediction:

$$\begin{array}{ll} f_*|X_*, X, f & \sim N(\overline{f}_*, cov(f_*)) \\ \overline{f}_* & = K(X_*,X)[K(X,X) + \sigma_n^2 I]^{-1}f \\ cov(f_*) & = K(X_*,X_*) - K(X_*,X)[K(X,X) + \sigma_n^2 I]^{-1}K(X,X_*) \end{array} \tag{4.9}$$

Using a more compact notation $k_* = K(X, X_*)$, for a single test point x_* the predictive equations 4.9 become:

$$\begin{array}{ll} \overline{f}_* & = k_*^T(K + \sigma_n^2 I)^{-1} f = \sum_{i=1}^n \alpha_i k(x_i, x_*) \\ cov(f_*) & = k(x_*, x_*) - k_*^T(K + \sigma_n^2 I)^{-1} k_* \end{array} \tag{4.10}$$

,where the mean is regarded as a linear combination of n covariance functions with $\alpha = (K + \sigma_n^2 I)^{-1} f$. Therefore, prediction for one testing point and n training points involves only the $(n+1)$-dimensional distribution defined by these points. Also, the predicted variance only depends on the inputs provided and not on the observed values, which is a property of Gaussian distributions.

The implementation of the prediction algorithm is described in Figure 4.3. The algorithm receives the training dataset described by input X and observations y, the covariance function k, noise variance σ_n^2 and the test input x_*, and outputs the mean $\overline{f}_*$, variance $V[f_*]$ and log marginal likelihood $\log p(y|X)$.

input: X (inputs), $\mathbf{y}$ (targets), k (covariance function), σ_n^2 (noise level), $\mathbf{x}_*$ (test input)

2: $L := \text{cholesky}(K + \sigma_n^2 I)$

$\boldsymbol{\alpha} := L^\top \backslash (L \backslash \mathbf{y})$ } predictive mean

4: $\bar{f}_* := \mathbf{k}_*^\top \boldsymbol{\alpha}$

$\mathbf{v} := L \backslash \mathbf{k}_*$ } predictive variance

6: $\mathbb{V}[f_*] := k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{v}^\top \mathbf{v}$

$\log p(\mathbf{y}|X) := -\frac{1}{2}\mathbf{y}^\top \boldsymbol{\alpha} - \sum_i \log L_{ii} - \frac{n}{2} \log 2\pi$

8: **return**: $\bar{f}_*$ (mean), $\mathbb{V}[f_*]$ (variance), $\log p(\mathbf{y}|X)$ (log marginal likelihood)

Figure 4.3: Steps of the prediction algorithm implementation, where L is the lower triangular matrix obtained in the Cholesky decomposition of a matrix $A = LL^T$. The equation $Ax = b$ is solved efficiently using two triangular systems: $x = L\backslash(L\backslash y)$, where the notation $A\backslash b$ represents the solution for $Ax = b$. (Figure from [33]).

The predictive equations 4.10 are implemented using the Cholesky decomposition instead of direct matrix inversion, as it represents a faster and more stable method. The complexity for the decomposition in line 2 is $O(n^3/6)$ and for solving the triangular systems in line 3 and 5 is $O(n^2/2)$. In the case of multiple inputs, steps 4 to 6 are repeated and as the observations used in the project are noisy, the noise variance σ_n^2 is added to the predictive variance obtained $V[f_*]$.

4.4 Data representation

The input of the regressor is represented by normalized body part positions. The flexible model uses 26 body parts as it produces better performance for introducing additional

orientation with a better body coverage. Regression using GPs implies considering a trade-off between the redundant information and computation efficiency and training time. Experiments show that best performance is obtained when inputs are represented by 16 body parts: head, neck, upper and lower torso, shoulders, elbows, wrists, hips, knees and ankles, as they provide relevant information for a lower complexity of the human body representation. These are obtained by remapping the 26 body part positions obtained in the previous step considering geometrical human body constraints.

The 2D poses require normalization such that the input poses are independent of body size and distance to the camera. As most poses represent upright standing persons, all coordinates are normalized using the y range of each frame according to the following equations:

$$\begin{aligned} BP &= \{x_1, y_1, ..., x_{16}, y_{16}\} \\ BP_{norm} &= \{BP + M_{off}\} * M_{scale} \\ M_{scale} &= \left\{\frac{1}{y_{range}}, ..., \frac{1}{y_{range}}\right\} \\ M_{off} &= \{x_{off}, y_{off}, ..., x_{off}, y_{off}\} \\ x_{off} &= -\min(X) + (y_{range} - x_{range})/2 \\ y_{off} &= -\min(Y) \end{aligned} \tag{4.11}$$

where BP and BP_{norm} represent the original and the normalized body part positions input respectively, and X and Y represent original vectors of x and y coordinates.

With regard to the output data representation, the most straightforward approach for representing the 3D human body is directly storing and manipulating the raw 3D coordinates. However, Cartesian coordinates are generally not a good option for pose modeling, since 3D positions suffer of much variability due to subject appearance, type of action, camera viewpoint. Also, topology relations between part coordinates cannot be established and their usage requires a lot of post-processing. Other approaches that are faster and widely used in motion analysis are Euler angles and quaternions. However, the latter suffer from discontinuities and singularities problems and the former require too many parameters which do not benefit of a direct geometrical interpretation. A robust and more efficient approach is representing the orientation of each limb using direction cosines.

4.4.1 Direction cosines

The direction cosines of a vector represent the cosines of the angles formed by the direction of the vector with all the coordinate axes of a chosen reference system. As shown in Figure 4.4 (b), direction cosines are computed:

$$\begin{aligned} \cos\theta_l^x &= \frac{l_x}{\sqrt{l_x^2+l_y^2+l_z^2}} \\ \cos\theta_l^y &= \frac{l_y}{\sqrt{l_y^2+l_y^2+l_z^2}} \\ \cos\theta_l^z &= \frac{l_z}{\sqrt{l_z^2+l_y^2+l_z^2}} \end{aligned} \tag{4.12}$$

which leads to:

$$\cos^2\theta_l^x + \cos^2\theta_l^y + \cos^2\theta_l^z = 1 \tag{4.13}$$

Therefore, direction cosines have an intuitive geometric interpretation and can be easily obtained. As 4.2 shows, they are dependent of each other and require 3 parameters to define 2 DOF. One important advantage of direction cosines is the fact that they are continuous and smooth, as they are defined in the unit sphere. Therefore, they are a very suitable representation for 3D motion related methods, being exempt from discontinuities and easily treatable.

4.4.2 3D body pose

The chosen representation for the 3D pose is identical to the one used in [34], which consists of 12 rigid body parts: mid-hip, torso, mid-shoulder, neck, two upper legs, two lower legs, two upper arms and two lower arms. The parts are connected by a total of 10 joints as shown in Figure 4.4(a). A local coordinate system is defined in the hip with the y axis pointing towards torso, z axis towards the left hip and the x axis given by the cross product between the two. The 3D pose is represented as a vector of 36 direction cosines corresponding to each body part:

$$\psi = \left\{\cos^2\theta_l^x, \cos^2\theta_l^y, \cos^2\theta_l^z, ..., \cos^2\theta_{l2}^x, \cos^2\theta_{l2}^y, \cos^2\theta_{l2}^z\right\} \tag{4.14}$$

,where $\theta_i^x, \theta_i^y, \theta_i^z$ are the angles formed by the limb with the axes of the local coordinate system as shown in Figure 4.4(b).

4.5 Experiments

The performance of the Gaussian process regressor is tested on the HumanEva dataset on all the actions, taking as input the vectors of 2D poses obtained with the 2D detector described in the previous chapter. The experiments imply dividing each action dataset into two equal sets for training and testing. Experiments are carried on identical training and testing data as used in [3] and [13], according to Table 4.1, such that results can be compared to the ones obtained in the referred papers.

Both papers use similar methods for detecting 2D poses from images, by extracting histograms of shape contexts from silhouettes. Mapping to 3D poses is done using

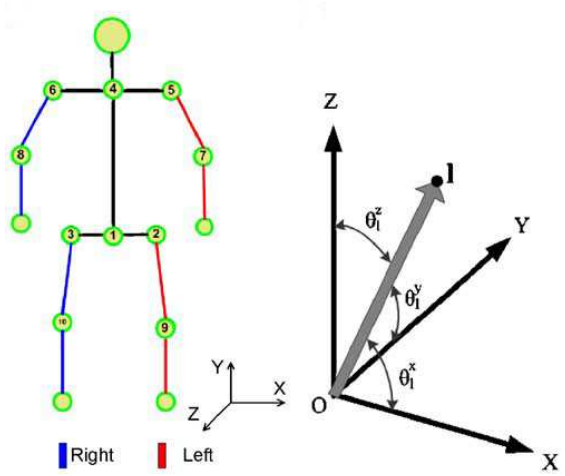


Figure 4.4: 3D human body model (a) Direction cosines for an orientated limb (b).(Figure from [34]).

Table 4.1: Size of training and testing data used from HumanEva, subject S1, camera C1.

Motion	Number of frames
Walking	1197
Jog	597
Gestures	795
Box	498
ThrowCatch	217

different learning methods: [36] uses Structured Output-Associative Regression(SOAR), which learns functional dependencies where outputs are both input-dependent and self-dependent, while [13] uses a Gaussian process regressor similar to the one described in this chapter.

The ground truth poses from HumanEva dataset are represented using 15 virtual markers placed at limb ends and midjoints. Therefore, all the predicted body poses are reprojected to match the HumanEva body configuration. To compute the estimated 3D marker location, the predicted limb angles and absolute limb lengths are used, the latter being obtained assuming an average U.S.-sized body model [26] and using pre-computed limb lengths ratios. 3D estimation performance is measured using the average angular error and average absolute marker position error:

$$\begin{aligned} Err_{ang} &= \frac{\sum_{i=1}^{J} |\Theta_i - \hat{\Theta}_i| mod 180^\circ}{J} \\ Err_{pos} &= \frac{\sum_{i=1}^{M} |P_i - \hat{P}_i|}{M} \end{aligned} \tag{4.15}$$

,where $\Theta_i = [\theta_l^x, \theta_l^y, \theta_l^z, ..., \theta_{l4}^x, \theta_{l4}^y, \theta_{l4}^z]$ and $\hat{\Theta}_i$ represent the J-dimensional vectors of ground truth and predicted limb angles respectively, and $J = 3 \cdot 14$, for 3 Euler angles per 14 limb joints (the hip is ignored as it represents the origin of the local coordinate system). Similarly, $P_i = [x_1, y_1, z_1, ..., x_{15}, y_{15}, z_{15}]$ and $\hat{P}_i$ represent the the M-dimensional vectors of ground truth and predicted limb marker positions respectively, and $M = 3 \cdot 15$, for 3 coordinates per 15 limb joints.

4.6 Results

Visual results of the tree kinematic structures obtained on the HumanEva dataset are shown in Figure 4.5. Since the body marker locations are estimated using a local coordinate system that is placed in the hip, all examples of estimated body pose are presented from a singular view relative to the body for a better visualization of the pose. Using camera calibration settings, the 3D pose can be reprojected in the image plane for a visualization of the tree structure overlaid on the person in the image.

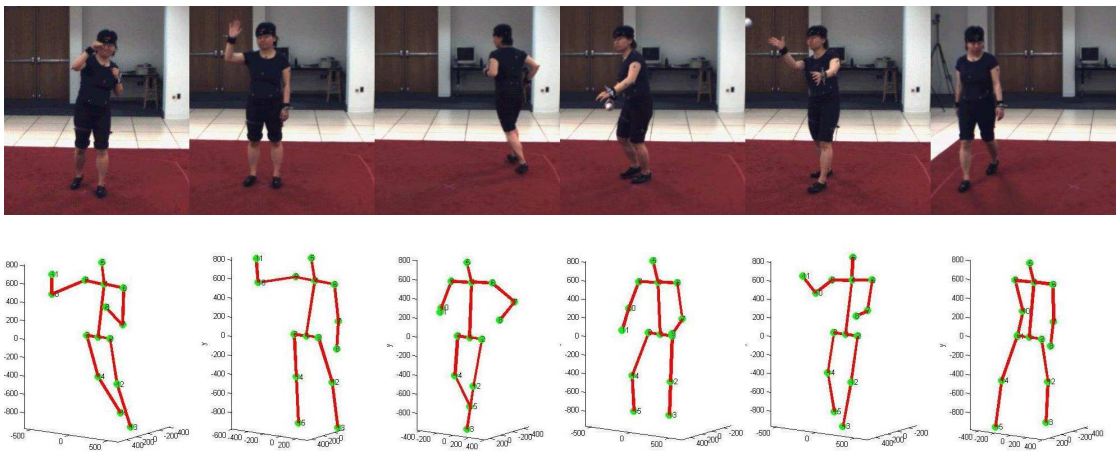


Figure 4.5: The first row presents test images from all actions datasets. The second row presents the corresponding kinematic tree structures with estimated limbs, presented from a 45° view relative to the body.

Comparative results on the HumanEva dataset are presented in Table 4.2. Overall, the system presented outperforms or performs similarly to previous works that use shape contexts. This can be explained by the quality of the inputs i.e. the general good estimates obtained with the 2D detector. Also the GP training is much faster than in [13] since our inputs are 16-dimensional, compared to the 400-dimensional inputs based on histograms of shape contexts. For example, training on the Gestures dataset which consists of 398 frames is done in 124 min and 48 s using shape contexts, while our method only takes 12 min and 50 s.

Table 4.2: Average limb position and angle errors are computed for S1, Cam1, on all action types. Results are compared for the presented framework, and the ones used in [3] and [13].

Method	**Motion**	Err_{pos} $[mm]$	Err_{ang} $[^\circ]$
SOAR [3]	Walking	59.8	-
	Jog	62.7	-
	Gestures	49.6	-
	Box	77.3	-
	ThrowCatch	110.3	-
GP [13]	Walking	21.75	0.96
	Jog	26.96	1.42
	Gestures	68.37	2.87
	Box	**16.97**	**1.04**
	ThrowCatch	19.19	1.08
our system	Walking	**3.50**	**0.17**
	Jog	**6.85**	**0.38**
	Gestures	**1.57**	**0.11**
	Box	18.72	1.30
	ThrowCatch	**8.33**	**0.71**

It is important to note that the errors obtained on identical training sets for 2D pose estimation are expressed in pixels while the ones obtained for 3D estimation are expressed in milimeters, as the errors are computed using world-space 3D coordinates. Therefore, there is a fine correlation between the quality of 2D and 3D estimates. Assuming that better 2D full-body detections lead to a better 3D pose estimation is straightforward, but other factors affect the results, such as the size of the training datasets, camera viewpoint, quality of detections per part etc.

One point of view for interpreting the results is the size of the datasets. The best mean error rates are obtained on the Walking and Gestures datasets which can be related to the fact that the training sets for these actions are larger, providing more possible poses for a better data-fit and obtaining a more accurate covariance function. The highest error rate is obtained on the Box and ThrowCatch datasets which have smaller training sets.

As shown in Table 3.1 from Section 3.3, the Box dataset obtained the lowest mean error. However, the final 3D results show increased error rate for the same dataset. This is partially due to higher errors in 2D upper limb detection which was not clearly shown in the 2D detection process because of the imprecision of the pixel error measure. This accounts for all the datasets, but the main factor that brings increased error rate is missing Mocap data in the shape of sets of almost 20 consecutive frames. The lack of 3D information deteriorates the training process and leads to high partial and mean errors as can be seen for frame 200 in Figures 4.7 and 4.8. The ground truth and

estimated tree kinematic structures are shown in Figure 4.6, with pose error information:

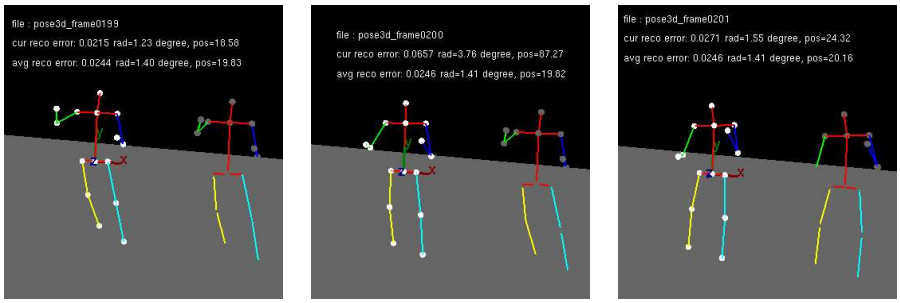


Figure 4.6: The ground truth (left) and estimated (right) tree kinematic structures for the Box dataset from frame 199 to 201. The error peak of 87 mm is reached in frame 200 and the ground-truth missing data is deducted from the discontinuous motion of the limbs.

The same behavior is shown as a peak in error rate in the Jog dataset around frame 100, corresponding to a missing information for a block of 30 frames. The lack of 3D information violates the assumption of input received as a sequence of consecutive frames so the experiment should be repeated for a continuous block of image sequence of the Box dataset.

The following figures present mean pose and joint errors per actions over all testing data. As in the case of 2D detections, the head and torso present the lowest errors and are the most stable joints. Also the shoulders estimates have a low error rate and they represent more stable joints than in the 2D detections. This is due to the fact that the overall 3D configuration is rebuilt using pre-computed limb lengths from a generic 3D model. The joints that are most prone to errors are elbows, wrists and ankles, as the corresponding high errors from 2D detections are propagated.

4.7 Conclusions

The chapter presents our solution for 2D to 3D pose mapping using Gaussian process regression. The 2D input space is represented by normalized 2D estimates of the human body joint and the outputs are represented by direct cosine angles from which the 3D configuration is rebuilt using pre-defined geometrical constraints. Training is done an action databases in order to compute the hyperparameters that describe the covariance function which represents the GP.

The results show that GPs can be used as a very flexible and fine tool for non-linear regression, outperforming on average previous works by over 70%. The results are explained generally by the quality of the inputs, which also present the advantage of

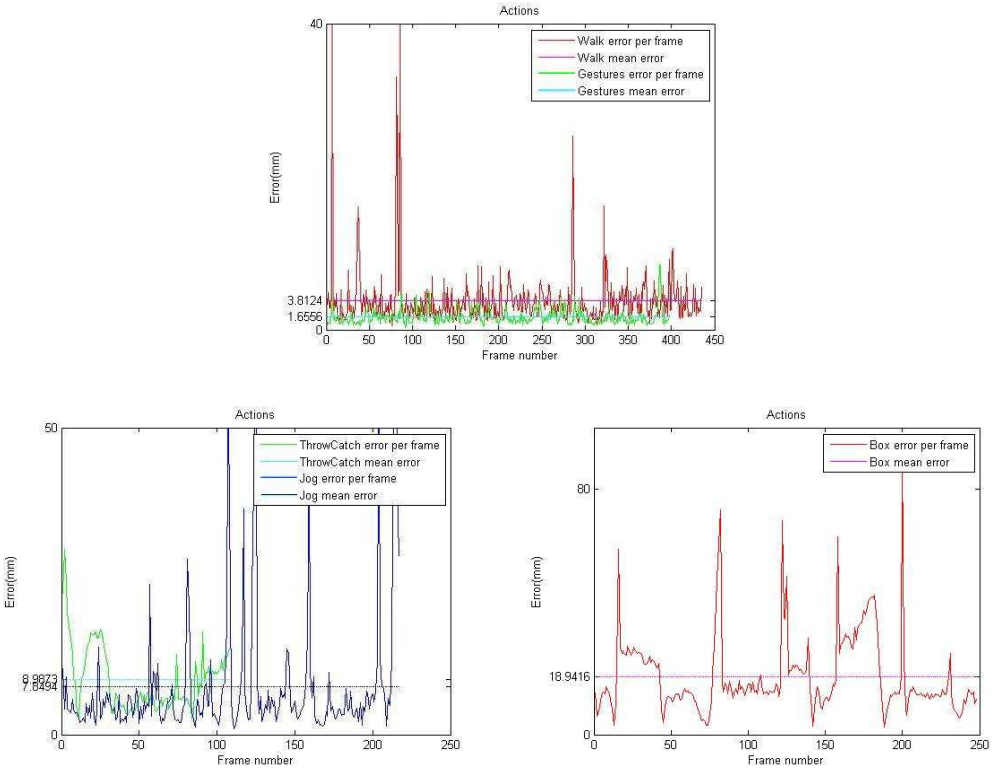


Figure 4.7: Error plots per actions over all testing frames and mean error.

low-dimensionality. Joints have a similar behavior as in the case of 2D detection, as the head and torso are more stable, retrieving a lower error rate than wrists, elbows or ankles.

However, the proposed system is able to generate good predictions on the learned action databases i.e. on video sequences that present similar poses to the ones annotated in HumanEva: Box, Gestures, Walking, Jog and ThrowCatch. For improved results on general videos, all actions, including various camera viewpoints, should be included in a large dataset for training a single Gaussian process.

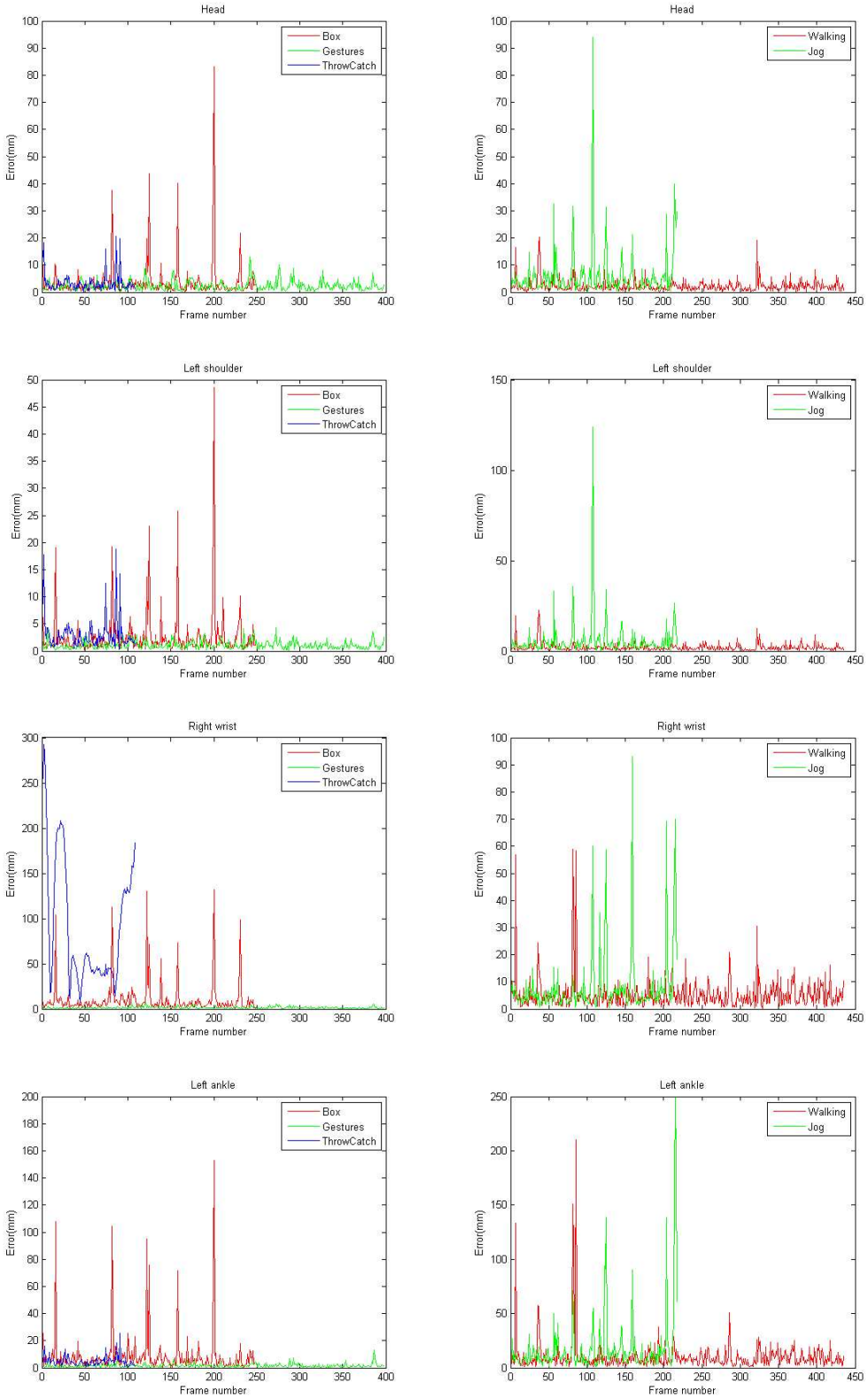


Figure 4.8: Part error plots per joints over all testing frames and actions.

Chapter 5

Conclusion

The work presented in this thesis is related to automated human motion analysis. Precisely, it is aimed at implementing an automated system for 3D human pose estimation in video sequences. Towards this end, the system is built as a 3-stage framework: first the 2D body parts are estimated from consecutive frames using a human detector which uses a flexible mixture model based on structural SVM, providing state of the art results at the order of seconds. Next, the overall estimated body part locations are improved using a weighted robust smoothing technique, leading to a lower error rate in 2D part estimation and a more continuous appearance of the human motion. Finally, the 3D configurations are estimated using a Gaussian process regressor trained on specific action datasets. The representation chosen for the 3D poses consists of direction cosines to express limb orientations, thus avoiding singularities and discontinuities. The final outputs of the system are represented by vectors of relative 3D locations in a local coordinate system.

The performance of the overall framework is measured by experimenting on the action datasets provided by the HumanEva benchmark. Results show that the system generally outperforms previous work that based on histograms of shape contexts and that it is robust against self-occlusions, foreshortening effects and highly articulated non-horizontal poses, meeting the initial requirements. As it is trained on specific action datasets, the regressor will generate good predictions for poses that are related to similar actions. In order to address general unconstrained motions, the overall approach should be trained on a wider dataset consisting of a wider range of actions captured from different camera viewpoints.

Another limitation is represented by the fact that retrieval of absolute position and orientation of the body is not addressed in the project. The 3D configurations are described as body part locations in a local coordinate system that is placed in the hip. However, if camera calibration settings are available, these parameters can be computed and the 3D configuration can be reprojected in the 2D image space.

Generally, the system can be regarded as a black box as it involves minimal user interaction and the only input required is represented by the raw image sequence, which can be obtained with a video converter such as ffmpeg. The 2D detection part takes about 7 seconds for a 644x488 resolution frame and the 3D prediction using a trained GP regressor takes roughly 2 seconds for a testing dataset of 200 frames.

Improvements can be brought to the system by training the GP regressor on a wider dataset that includes more actions, transitions between actions and camera viewpoints and also by integrating the temporal information while training the regressor. For the purpose of visualization and as an example of application in 3D animations, the obtained vectors of 3D coordinates could be mapped to a 3D body model in software such as Blender or Autodesk SoftImage.

Bibliography

[1] Agarwal, A. and Triggs, B. (2006). Recovering 3d human pose from monocular images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on.*

[2] Binford, T. O. (1971). Visual perception by computer. *Systems and Control, IEEE Conference on.*

[3] Bo, L. and Sminchisescu, C. (2009). Structured output-associative regression. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.*

[4] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory.*

[5] Bourdev, L. and Malik, J. (2009). Poselets: Body part detectors trained using 3d human pose annotations. In *Computer Vision, 2009 IEEE 12th International Conference on.*

[6] D. A. Forsyth, O. Arikan, L. I. J. O. D. R. (2006). Computational studies of human motion part 1: tracking and motion synthesis. *Foundations and Trends in Computer Graphics and Vision 1.*

[7] D. Ramanan, D.A. Forsyth, A. Z. (2007). Tracking people by learning their appearance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on.*

[8] Epshtein, B. and Ullman, S. (2007). Semantic hierarchies for recognizing objects and parts.

[9] Felzenszwalb, Ross B. Girshick, D. M. D. R. (2010). Object detection with discriminatively trained part based models. *PAMI.*

[10] Fergus, R., Perona, P., and Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on.*

[11] Ferrari, V., Marin-Jimenez, M., and Zisserman, A. (2008). Progressive search space reduction for human pose estimation.

[12] Garcia, D. (2010). Robust smoothing of gridded data in one and higher dimensions with missing values. *Computational Statistics and Data Analysis*.

[13] Gong, W., Brauer, J., Arens, M., and Gonzalez, J. (2011). Modeling vs. learning approaches for monocular 3d human pose estimation. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*.

[14] H. Sidenbladh, M. Black, L. S. (2002). Implicit probabilistic models of human motion for synthesis and tracking.

[15] Hen Y.W., P. R. (2009). Single camera 3d human pose estimation: A review of current techniques. *International Conference for Technical Postgraduates*.

[16] J. Brauer, M. A. (2011). Reconstructing the missing dimension: From 2d to 3d human pose estimation. In *Computer Analysis of Images and Patterns*.

[17] J. Deutscher, R. I. (2005). Articulated body motion capture by stochastic search. *International Journal of Computer Vision*.

[18] Kehl, R. and Gool, L. V. (2006). Markerless tracking of complex human motions from multiple views. *Computer Vision and Image Understanding*.

[19] Kolgomorov, A. N. (1941). Interpolation und extrapolation. In *Izv. Akad. Nauk SSSR*.

[20] L. Sigal, M. B. (2006). Predicting 3d people from 2d pictures. In *Articulated Motion and Deformable Objects*.

[21] Leibe, B., Leonardis, A., and Schiele, B. (2006). An implicit shape model for combined object categorization and segmentation. In *Toward Category-Level Object Recognition*.

[22] M. Andriluka, S. Roth, B. S. (2008). People-tracking-by-detection and people-detection-by-tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*.

[23] M. Andriluka, S. Roth, B. S. (2010). Monocular 3d pose estimation and tracking by detection. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*.

[24] M. Fischler, R. E. (1973). The representation and matching of pictorial structures. *IEEE*.

[25] M. W. Lee, R. N. (2009). Human pose tracking in monocular sequence using multilevel structured models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*.

[26] M.A. McDowell, C. D. Fryar, C. L. O. (2008). Anthropometric reference data for children and adults: United states, 2003-2006. In *National Health Statistics Reports*.

[27] Moeslund, T. B., Hilton, A., and Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*.

[28] N. Dalal, B. T. (2005). Histograms of oriented gradients for human detection.

[29] P. Felzenszwalb, D. H. (2005). Pictorial structures for object recognition. *IJCV*.

[30] Poppe, R. (2007). Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*.

[31] R. Urtasun, D. J. Fleet, P. F. (2006). Temporal motion models for monocular and multiview 3d human body tracking. *Computer Vision and Image Understanding*.

[32] Ramanan, D. (2007). Learning to parse images of articulated bodies. *NIPS*.

[33] Rasmussen, C. (2004). Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning*.

[34] Rius, I., Gonzàlez, J., Varona, J., and Roca, F. X. (2009). Action-specific motion prior for efficient bayesian 3d human body tracking. *Pattern Recognition*.

[35] Sigal, L., Balan, A., and Black, M. (2010). Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*.

[36] Sminchisescu, C. (2008). 3d human motion analysis in monocular video: techniques and challenges. *Human Motion - Understanding, Modelling, Capture and Animation*.

[37] Taylor, C. (2000). Reconstruction of articulated objects from point correspondences in a single uncalibrated image. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*.

[38] Thomas B. Moeslund, E. G. (2001). A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*.

[39] W. Gong, J. Brauer, M. A. J. G. (2011). On the effect of temporal information on monocular 3d human pose estimation. *ICCV*.

[40] Wiener, N. (1949). Extrapolation, interpolation and smoothing of stationary time series. In *MIT Press, Cambridge, Mass.*

[41] Yang, Y. and Ramanan, D. (2011). Articulated pose estimation with flexible mixtures-of-parts. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Appendix A

3D Human Pose Estimation using 2D Body Part Detectors

The following pages represent the contents of the article submitted to ICPR 2012

3D Human Pose Estimation using 2D Body Part Detectors

Adela Bărbulescu [1,2] Wenjuan Gong [1] Jordi Gonzàlez [1] Thomas B. Moeslund [2]

[1] *Centre de Visió per Computador*
Universitat Autònoma de Barcelona

[2] *Aalborg University, Denmark*

Abstract

Automatic 3D reconstruction of human poses from monocular images is a challenging and popular topic in the computer vision community, which provides a wide range of applications in multiple areas. Solutions for 3D pose estimation involve various learning approaches, such as support vector machines and Gaussian processes, but many encounter difficulties in cluttered scenarios and require additional input data, such as silhouettes, or controlled camera settings.

We present a framework that is capable of estimating the 3D pose of a person from single images or monocular image sequences without requiring background information and which is robust to camera variations. The framework models the non-linearity present in human pose estimation as it benefits from flexible learning approaches, including a highly customizable 2D detector. Results on the HumanEva benchmark show how they perform and influence the quality of the 3D pose estimates.

1. Introduction

3D human pose estimation from monocular images represents an important and top researched subject in the computer vision community due to its challenging nature and widespread applications, ranging from advanced human computer interaction, smart video surveillance to arts and entertainment industry. The difficulty of the topic resides in loss of depth information that occurs when projecting from 3D space to the 2D image plane. Thus, a wide set of approaches have been proposed to tackle the problem of 3D pose recovery from monocular images.

Due to the 2D-3D ambiguity, many approaches rely on well-defined laboratory conditions and are based on additional information such as silhouettes or edge-maps obtained for example from background subtraction methods [1, 2, 3]. However, realistic scenarios present highly articulated human poses affected by self-occlusion, background clutter and camera motion, requiring more complex learning approaches.

A particular class of learning approaches use direct mapping methods from image features such as grids of local gradient orientation histograms, interest points, image segmentations to 3D poses [4, 5, 6, 7]. Another class of approaches maps the image features to 2D parts and then uses modeling or learning approaches to map these to 3D poses [8, 9]. Among these learning approaches, the most used ones are support vector machines, relevance vector machines and Gaussian processes. In [9] a comparison is presented between modeling and learning approaches in estimating 3D poses from available 2D data, using geometrical reconstruction and Gaussian processes.

This paper describes a two-stage framework which recovers 3D poses without requiring background information or static cameras. Image features are mapped to 2D poses using a flexible mixture model which captures co-occurrence relations between body parts, while 3D poses are estimated using a Gaussian process regressor. Experiments are conducted systematically on the HumanEva benchmark, comparing the 3D estimates based on different methods of mapping the image features to Gaussian process inputs.

2. Detector of 2D Poses

The dominant approach towards 2D human pose estimation implies articulated models in which parts

are parameterized by pixel location and orientation. The approach used by Ramanan [10] introduces a model based on a mixture of non-oriented pictorial structures. The main advantages of using the articulated mixture model consist in the fact that it is highly customizable, using a variable number of body parts, and that it reflects a large variability of poses and appearances without requiring background or temporal information. Also, it outperforms state-of-the-art 2D detectors while requiring less processing time. The next sections describe the model proposed in [10]:

2.1. Part-based Model for Human Detection

The mixture model implies mixtures of parts or part types for each body part, in our case spanning different orientations and modeling the implied correlations. The body model can be associated with a graph $G = (V, E)$ in which nodes are represented by body parts and edges connect parts with strong relations.

Similar to the star-structured part-based model in [3], this mixture model involves a set of filters that are applied to a HOG feature map [11] extracted from the analyzed image. A configuration of parts for an n-part model specifies which part type is used from each mixture and its relative location. The score of a configuration of parts is computed according to three model components: co-occurrence, appearance and deformation [10]:

$$S(I,p,t) = \sum_{i\in V} {b_i}^{t_i} + \sum_{i\in V} {w_i}^{t_i} \cdot \Phi(I,p_i) + \sum_{i,j\in E} {w_{ij}}^{t_i} \cdot \Psi(p_i - p_j) \quad (1)$$

where the first term favors certain part type associations, the second term expresses the local appearance score by assigning weight templates associated to part i and part-type t_i to certain locations p_i, described by the extracted HOG descriptor, and the third term expresses the deformation score by assessing the part-type pair assignment parameters and the relative location between connected parts i and j.

As the model described is highly customizable, experiments have been deployed as to find a more efficient model structure by varying the number of part-types and mixtures. A full-body 26-part model (Figure 1) is chosen, as it shows increased performance due to the capture of additional orientation.

2.2. Inference and Learning

Inference using the mixture model described is obtained by retrieving the highest-scoring configuration, precisely by maximizing $S(I, p, t)$ (1) over all parts and part-types. Building the associated relational graph G as a tree allows for efficient inference with dynamic programming.

The solution used for training a model which generates high scores and outputs a set of parameters containing limb locations is a structural SVM, leading to a problem of quadratic programming (QP), which in this case is solved using dual coordinate-descent.

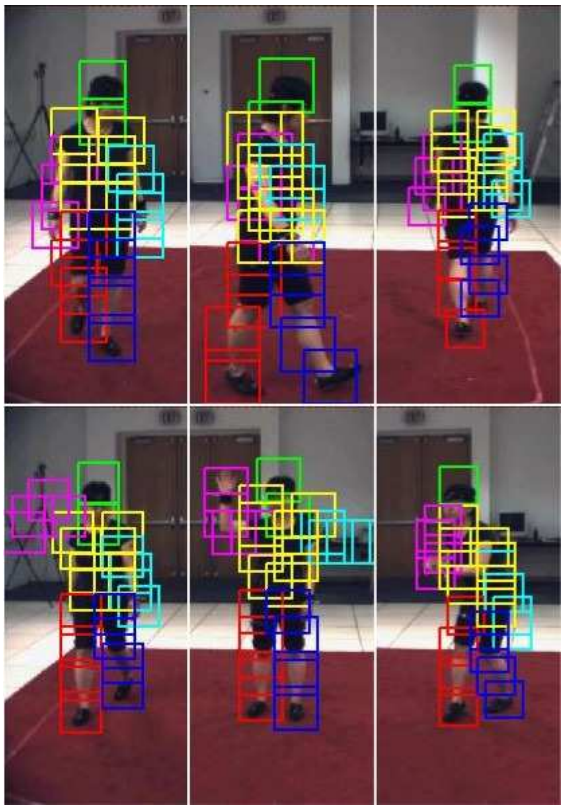


Figure 1. Person detected using a 26-part model, highlighting body parts with bounding boxes. The first row presents successful detections and the second presents limb misdetections.

Although the detector covers a wide variability of articulated poses, there are situations of limb misdetection, generated by self-occlusion, double-counting phenomena or background clutter.

3. Estimation of 3D Poses

As proven to be an effective approach for tackling the 2D to 3D mapping problem [4], Gaussian processes regression is currently the most widespread learning method used in pose estimation. Given a prediction problem, Gaussian processes can be considered as a fine tool that extends a multivariate Gaussian distribution of the training data and which, using a correlation between observations and test data, maps the test data to new estimates. In our case, the input data is represented by the 2D body-part coordinates given by the previously described detector and the output is represented by 3D pose estimates as direction cosines of limb orientations.

3.1. 3D pose representation

Considering the fact that the regressor outputs 3D poses, a robust representation is needed for the human pose. As training time is also an important factor, a smaller dimension representation is desirable. The human body is represented by a stick figure model composed of 13 body parts. As described in [5], a robust and efficient manner of representing 3D body limbs is the use of direction cosines. The angles of the limbs are considered with respect to a local coordinate system, fixed in the hip, with the *y* axis given by the torso, the *z* axis given by the hip line pointing from the left to right hip and the *x* axis given by the direction of their cross product.

The output is represented as a 36-dimensional vector:

$$\varphi = [cos\theta_1^x, cos\theta_1^y, cos\theta_1^z, \dots, cos\theta_{12}^x, cos\theta_{12}^y, cos\theta_{12}^z] \quad (2)$$

where θ_l^x, θ_l^y, θ_l^z are the angles formed by the limb l with the axes. The use of direction cosines is robust and easily treatable as it prevents singular positions and discontinuities of angle values.

3.2. Gaussian process regression

Using Gaussian processes for prediction problems can be regarded as defining a probability distribution over functions, such that inference takes place directly in the function space-view. The training data observations $y = \{y_1, \dots, y_n\}$ are considered samples from the *n*-variate Gaussian distribution that is associated to a Gaussian process and which is specified by a mean and a covariance function. Usually, it is assumed that the mean of the associated Gaussian process is zero and that observations are related using the covariance function $k(x, x')$. The covariance function describes how function values $f(x_1)$ and $f(x_2)$ are correlated, given x_1 and x_2. As the Gaussian process regression requires continuous interpolation between known input data, a continuous covariance is also needed. A typical choice for the covariance function is the squared exponential:

$$k(x, x') = \sigma_f^2 \exp \frac{-(x - x')^2}{2l^2} \quad (3)$$

where σ_f represents the amplitude or the maximum allowable covariance, reached when $x \approx x'$ and $f(x)$is very close to $f(x')$, and l represents the length parameter which influences the separation effect between input values. If a new input data x is distant from x' then $k(x, x') \approx 0$ and the observation x' will have a negligible effect upon the interpolation.

Therefore, Gaussian processes represent a flexible learning approach, capable of modeling the inherent non-linearity found in human pose estimation.

3.3. Testing and results

All experiments are carried on the HumanEva dataset as it provides ground-truth 2D and 3D information on subjects performing different actions. For every action, the image frames are equally divided in training and testing data, the input received being vectors of 2D coordinates. 3D estimation performance is measured using the average angular error and average absolute marker position error:

$$Err_{ang} = \frac{\sum_{i=1}^{J} |\theta_i - \hat{\theta}_i| mod\ 180°}{J} \quad (4)$$

$$Err_{pos} = \frac{\sum_{i=1}^{M} |P_i - \hat{P}_i|}{M} \quad (5)$$

where $J = 3 \cdot 14$, for 3 Euler angles and 14 limbs, $\theta_i, \hat{\theta}_i$ represent ground truth and predicted limb angles, $M = 3 \cdot 15$, for 3 coordinates per marker and 15 markers and $P_i, \hat{P}_i$ represent ground truth and predicted marker positions.

Results are compared in the case of 26-dimensional input vectors containing 2D coordinates obtained directly from the 2D detector, 16-dimensional vectors with re-projected coordinates matching the HumanEva markers and a silhouette-based method that maps image features directly to 3D estimates using histograms of shape contexts [6]. As the silhouette-based experiments are carried in controlled conditions, requiring fixed cameras and background information, we will consider the method as ground truth experiment.

Table 1. Results obtained on the HumanEva dataset

Input	Motion (CAM1, S1)	Err_{ang} [°]	Err_{pos} [mm]
26-dim	Walking	2.8750	68.3740
	Box	3.7580	66.1650
	ThrowCatch	4.0690	66.9660
16-dim	Walking	2.6260	53.1960
	Jog	3.2800	63.6090
	Box	3.4270	56.9350
GT	Walking	0.9630	21.7530
	Jog	1.4270	26.9640
	Box	1.0400	16.9770
	ThrowCatch	1.0860	19.1960

The results show that using a simpler body representation for regression input performs better while training and prediction are less time consuming. The shape context-based solution [6] outperforms the two-stage framework because of the increased reliability of the features extracted from silhouettes. The biggest error rate is obtained for the "Walking" and "Jog" databases, where some frames present self-occlusions and generate double-counting and limb misdetections. Figure 2 presents visualizations of results on similar poses for the three cases:

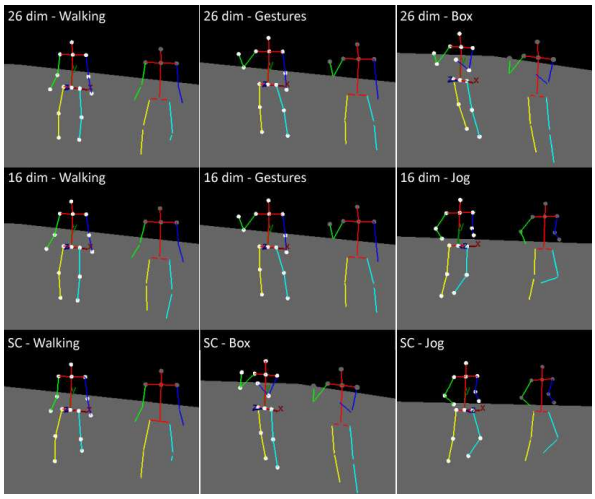


Figure 2. Estimates (left skeleton) and associated ground truth coordinates (right skeleton) for 26-dimensional inputs (first row), 16-dimensional inputs (second row) and shape context (third row).

4. Conclusion and future work

The paper presents learning approaches for the problem of 3D pose estimation from monocular images. The framework is composed of an articulated 2D detector with a varying number of body parts based on a structural SVM and a 2D to 3D Gaussian process regressor. Experiments carried on the HumanEva benchmark show that a simpler 2D body part model performs better, while the 3D estimates depend on the reliability of the 2D inputs.

For future work, the 2D detector will be improved within the temporal context, using a "tracklets" approach [8] for different frame window sizes [9], followed by motion smoothing.

References

[1] A. Balan, L. Sigal, M. Black, J. Davis, H. Haussecker. Detailed human shape and pose from images, *CVPR*, 2007

[2] J. Deutscher, I. Reid. Articulated body motion capture by stochastic search, *IJCV,* 2005

[3] L. Sigal, M. J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation, *CVPR*, 2007

[4] A. Agarwal, B. Triggs. Recovering 3D human pose from monocular images. *PAMI*, 2006

[5] C. Ionescu, L. Bo, C. Sminchisescu. Structural SVM for visual localization and continuous state estimation. *ICCV*, 2009

[6] L. Bo, C. Sminchisescu. Structured output – associative regression. *CVPR,* 2009

[7] C. Ionescu, F. Li, C. Sminchisescu. Latent Structured Models for Human Pose Estimation. *ICCV*, 2011

[8] M. Andriluka, S. Roth, B. Schiele. Monocular 3d pose estimation and tracking by detection. *CVPR,* 2010.

[9] W. Gong, J. Brauer, M. Arens, J. Gonzàlez. On the Effect of Temporal Information on Monocular 3D Human Pose Estimation. *ICCV,* 2011

[10] D. Ramanan, Y. Yang. Articulated pose estimation using flexible mixtures of parts. *CVPR*, 2011

[11] Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection.1:886 –893 vol. 1, 2005

[12] Felzenszwalb, Ross B. Girshick, D. M. D. R. Object detection with discriminatively trained part based models. *PAMI*, 2010

[13] J.M.Wang, D. J. Fleet, A. Hertzmann. Gaussian process dynamical models for human motion. *PAMI*, 2008

[14] I. Rius, J. Gonzàlez, J. Varona, and F. X. Roca. Actionspecific motion prior for efficient bayesian 3d human body tracking. *Pattern Recognition*, 42(11):2907–2921, 2009