Human Body Pose Tracking Based on Spatio-Temporal Joints Dependency Learning

by

Rodrigo Barrita Zebadúa

Thesis submitted in partial fulfillment of the requirements for the degree of:

MSc. in Computer Science

at

Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE)

August, 2018

Santa María de Tonantzintla, Puebla

Advisors:

PhD. Luis E. Sucar Succar

Computer Science Coordination at INAOE

Abstract

Human pose estimation consists on the localization of the body joints from images or videos and how they are connected between each other. An improvement in this issue will enhance the development of other areas such as video surveillance, action detection, human computer interaction, etc. The complexity of the problem is due to the flexibility of the human body structure that makes the space of movements high dimensional, in addition to other external factors such as clothes, illumination, occlusion, moving backgrounds, crowded scenes, etc. During the past couple of decades, great achievements have been made, although, there is still room for improvements and the problem remains open.

We propose an architecture for tracking the human pose in video sequences. The proposed model is composed by a Convolutional Neural Network (CNN) based part-detector and a Coupled Hidden Markov Model (CoHMM). The combination of both models allows learning spatial and temporal dependencies. The part-detector, in addition to the advantages of a CNN, exploits the spatial correlations between neighboring regions through a Conditional Random Field (CRF). On the other hand, the CoHMMs generate the best movement sequence between interacting processes. We evaluate our model on the PoseTrack benchmark dataset. The obtained results show that in such cases in which the part detector fails to properly keep the body structure between frames our model helps to fill in these gaps.

Resumen

La estimación de la pose humana consiste en la localización de las articulaciones del cuerpo a partir de imágenes o videos y como éstos se conectan entre sí. Los avances en las soluciones a este problema tienen impacto en otras áreas tales como video vigilancia, detección de acciones, interacción humano-computadora, etc. La complejidad de la estimación es debido a la flexibilidad de la estructura del cuerpo humano, lo que resulta en una espacio de movimientos de alta dimensionalidad. Además de otros factores como ropa, iluminación, oclusión, fondos en movimiento, escenas saturadas, etc. A pesar de los logros alcanzados en las últimas décadas aún hay espacio para mejoras y el problema permanece abierto.

Este trabajo propone una arquitectura para el seguimiento de la pose humana en secuencias de video. El modelo propuesto está compuesto de una combinación de un detector de articulaciones basado en Redes Neuronales Convolucionales (CNN) y Modelos Ocultos de Markov Acoplados (CoHMMs). La combinación de ambos modelos permite el aprendizaje de dependencias espacio-temporales. El detector de articulaciones, además de las ventajas de una CNN, aprovecha las correlaciones espaciales entre regiones vecinas a través de Campos Aleatorios Condicionales (CRF). Por otro lado, los CoHMMs generan las mejores secuencias de movimiento entre procesos que interactúan entre sí. Los resultados obtenidos indican que nuestro modelo llena los vacíos generados por los casos en donde el detector de articulaciones no mantiene la estructura del cuerpo humano a lo largos de toda la secuencia.

Acknowledgements

Foremost, I thank PhD. Luis E. Sucar Succar for his guidance on the research work presented here. I'm also grateful to Consejo Nacional de Ciencia y Tecnología (CONACYT) and Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) for providing me with a scholarship to support my studies.

Contents

Chapter 1

Introduction

1.1 Motivation

Human understanding has achieved the attention of researchers and the computer vision community from the past decades. The interpretation of human behavior is a complex task that had been broken into different challenges such as presence detection, gesture recognition, postures estimation, appearance modeling, etc. To give abstract semantic meaning to the data related to humans an improvement on each of these pieces plays an important role. The division of efforts on different sub-problems has raised great results in each area (Singh, 2017). In this work, we pay attention to the human postures through the commonly known problem Human Pose Estimation (HPE) to contribute in the enhancement of the estimations obtained from video sequences.

Human Pose Estimation is not an isolated problem but a key feature that could enhance the results of other ones such as video surveillance, human action recognition, gesture recognition, markerless motion capture, etc.

On the other hand, the increment of the amount of camera-equipped devices and the decrease in their cost have made computer vision implementations more accessible to the general public. This accessibility of devices leads to a broad variety of real-life applications. The most common real-life applications that have been attractive for research are the following: healthcare, human-computer interaction, augmented reality, virtual reality, gaming, etc.

In summary, the computer vision community has found attractive to give a solution to the HPE problem due to the potential contribution in other areas, the wide range of real-life application, and the ambitious goal of understanding the human behavior.

1.2 Problem description

Within the computer vision community the human understanding has been tackled from different perspectives related to the human body, their surroundings and the interaction between both.

Some questionings have raised the most common tasks related to research activities. In the case of isolated human bodies the related questions to this configuration are i) Is there a human in the scene? ii) Where is the person located? iii) Is the detected person a male or female? Is the detected person young or old? What is the detected person wearing? iv) Where are their limbs located? v) Is the detected person performing a specific gesture? Etc. If we consider the addition of the background environment to the analysis, the following questions are related to this configuration: i) Where the scene takes place? Is it inside a house? Is it an office? Ii) Is it a crowded scene? iii) What are the relevant objects In the scene? Etc. The interaction between humans and their surroundings lead to some questions, for example, i) Is the person performing a particular activity? ii) Is the individual holding, grasping or touching some objects? iii) How is the body posture related to the environment?

Is the person standing up or sitting? Etc.

The problem related to the question *Where are their limbs located?* is most commonly known as Human Pose Estimation (HPE). HPE consists of the localization of the body joints and how they are connected between each other. This task requires to take into consideration some challenges related to the inherent human body properties, some external factors related to the environment and other ones related to the interaction between them (Singh, 2017). The main challenges are the following:

- Occlusion: several factors make the occlusion a problem, for example, the flexibility of the human body in combination with the number of degrees of freedom. External cases of occlusion are most common in complex scenes such as crowded scenes or those in interaction with the environment.
- Background clutter: in most common real-life scenes there are several objects present within an image which make difficult the segmentation of the background from the object of interest.
- Body part foreshortening: images provide a 2D projection of the captured scenes. Therefore, the complex human body configuration leads their limbs appearance look out of proportion when they are not parallel to the image plane.
- Illumination variations: within indoor environments, the source images or videos could have controlled sources of illumination, but in outdoor scenarios, these conditions are more variable, and they could add difficulties to focus the target body.
- Clothing variations: the clothes that a target body is wearing could change its shape making difficult the limbs detection and the connections between them.

Figure 1.1 illustrates some examples of the different challenges we can face while

trying to estimate the human skeleton. The identification of particular individuals or body parts from still images must take into consideration cases when the person is occluded, the body is cluttered because of crowded scenes, or the body is out of the visible image area. These cases are present with the labels A and B.

The ideal case from Figure 1.1 C shows a relatively straightforward case for the estimator where the whole person is fully visible. Another case where a person takes a complex position is shown in D, the positions lead to self-occlusion and make hard to identify if it is a real body skeleton, in addition to the pose, the person is placed in front of another one. Although the target person is on top, this scenario makes it harder for the estimator to set a link between the articulations detected and the right person.

Within certain constraints, some methodologies have achieved great results on the HPE problem such as those based on depth images, infrared images, or other techniques like motion capture. However, their complex setup makes them unsuitable for outdoor environments. For instance, a solution based on depth information and color images, such as Kinect, has become very popular among the general public; however, these devices have an essential constraint with the distance in which they can acquire the data, which is about eight meters (Gong et al., 2016). The focus on low cost, highly accessible, and suitable for different environment solutions has led the research to rely just on RGB images as a data source since there is a vast amount of content in this format. For this reasons, the monocular and markerless are the attractive approaches to follow.

The monocular approach considers only one camera and the markerless approach considers to get rid of attachments to the body that could guide the estimations. These approaches take into consideration less controllable and more natural environments. However, this configuration increases the difficulty of the estimation since they are computed just with the information available within the images.

Although the pose estimation problem with markerless and monocular constraints isn't yet entirely solved, in recent years the research has moved to use convolutional neural networks (CNN) and graphical models. Due to this, significant improvements have been accomplished. The implementation of neural networks helps in the feature extraction phase, and it has also been employed in temporal analysis (Linna et al., 2016). Meanwhile, graphical models have been used for spatial correlations and also to maintain temporal consistency.

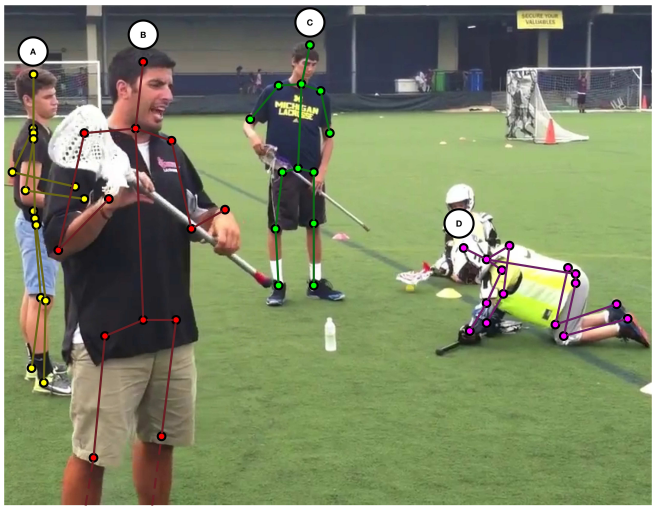

Figure 1.1: Pose estimation on a single image with multiple people. The image shows several cases involved in the human body pose estimation problem. A) Occlusion by external objects or people. B) Incomplete bodies in the visible area. C) Ideal complete body visibility. D) Complex body position.

1.3 Proposed solution

In this work, we propose a spatio-temporal architecture to track the human body from video sequences. This model follows a single person approach. The main contribution consists of performing a correction to an initial set of estimated sequences by taking into consideration the most probable sequence of movements for each joint.

Within video sequences, the human tracking process generates persistent paths

or trajectories of the target individuals through the available features. The video data source provides additional temporal information which could be exploited to maintain the structural consistency (Zou et al., 2009). This information offers strong clues for finding dependencies between temporarily adjacent frames and preserving the body structure through limb's trajectories.

Common tracking methods usually begin finding human body features that model the body parts in the current frame and then through a different model computes a prediction about the configuration in the next frame. This method is called *tracking by detection*. It reduces the tracking task to the association of the detections across all frames (Tian et al., 2015).

The proposed model is composed of the combination of a Convolutional Neural Networks and Probabilistic Graphical Models to improve the current human body pose tracking results. The spatial information is processed by a CNN, which also combines Conditional Random Fields (CRF) to maintain spatial consistency in the skeletal structure. This part of our approach is based on the model developed by (Chu et al., 2017) called Multi-Context Attention for Human Pose Estimation. They worked on different resolution variations on the target image to retrieve information from different semantics. Besides, they take into consideration neighboring regions through a CRF.

The temporal approach consists of the implementation of an HMM extension to couple two main processes from a body joint movement, the X and Y axis signals extracted from the spatial trajectory. We assume independence between the behavior of body joints to reduce the computational overhead. Moreover, we compute the best sequence of movement for each body part, coupling its X and Y signals.

The extension we implemented uses Coupled Hidden Markov Models (CoHMMs). The coupling algorithm was introduced by (Brand et al., 1997) and tested on a ges-

ture recognition classification problem. This coupling technique model outperforms the implementation of single HMMs or parallel HMMs. The proposed model is an approximation that reduces the state space for interacting processes from a naive approach $O(TN^{2C})$ to $O(T(CN)^2)$ where T is the number of observations, N the number of states and C the number of coupled chains. Our goal is to maintain a proper joint movement by computing the most probable sequence given the outputs of the part-detector. The most probable sequence is calculated by the N-Heads dynamic programming algorithm (Brand et al., 1997). Once we have the most probable sequence independently for each joint, a merging stage is performed to reconstruct the skeleton. Figure 1.2 shows the input images for the CNN and B the output of the model. It also illustrates the first skeletons from which our implementation is fed.

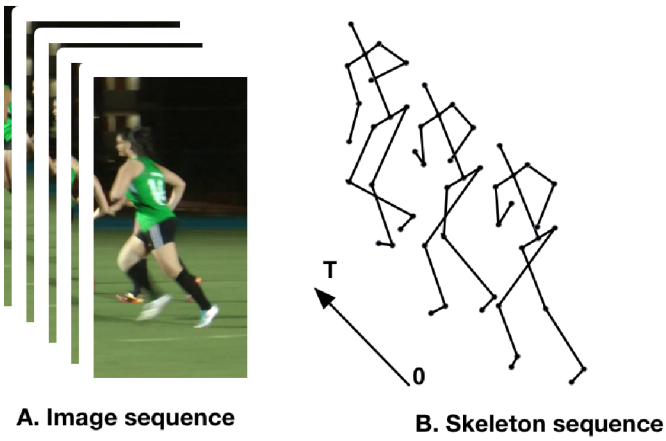

Figure 1.2: Human body pose estimation from an image sequence. A) The image is cropped with the target person centered. B) The estimation is a set of body kinematic models.

1.4 Objectives

In this work we explore the human body pose estimation through an architecture that combines the power of a part detection through a CNN which strongly con-

siders contextual information by working with different scales and neighboring regions and the computation of the most probable sequence through Coupled Hidden Markov Models (CoHMM). We build our CoHMM implementation on top of this part-detector.

The main objective of this work is to track the human body skeleton on video sequences through the computation of the most probable sequence of movements which is related to the following specific objectives.

- To estimate the body joints positions from still images.
- To implement a HMM to represent the temporal information of each body joint.
- To track each body joint by computing the most probable sequence for each joint, combining the spatial and temporal information.

1.5 Limitations

The proposed architecture relies on RGB images without any further information such as depth or body marks. Therefore the whole process is computed through a monocular and markerless approach. As this configuration arises some advantages by the increment of areas of application and the reduction of the setup requirement, it also carries some challenges due to the simple input data and the huge space of possible poses. This architecture is intended to work with single people detections, and the most probable sequence is computed independently on each body joint by the association of one CoHMM per body joint.

1.6 Document organization

The organization of this document is as follows. Chapter 2 describes the central theoretical concepts about the methods we use based on graphical models and neural networks. Chapter 3 provides an overview and a taxonomy related to the most relevant approaches around the body pose estimation problem; we also discuss some methods that have been worked on still images and video sequences. Chapter 4 describes the methodology. It explains the base human body representation and the base body joint trajectory representation. It also provides the details of the CoHMM implementation to perform the tracking process. The whole process is divided into the following stages: i) trajectories representation, ii) most probable sequence, and iii) body reconstruction. Chapter 5 presents the experiments configuration and discusses the results. Finally, chapter 6 gives the conclusions of this work, and it provides some future research directions.

Chapter 2

Theoretical framework

In this chapter, we review the literature closely related to our proposed solution, covering the relevant topics of graphical models and neural networks. The chapter is organized as follows. Section 2.1 describes the basic concepts of Hidden Markov Models. Section 2.2 discusses the Viterbi algorithm. Section 2.3 defines the primary model of this thesis, the Coupled Hidden Markov Models. In section 2.3.1 it is explained the deterministic approximation algorithm to couple HMMs. In Section 2.3, we describe the base definition of Convolutional Neural Networks (CNN). In section 2.4, the Conditional Random Fields (CRFs) are described. In Section 2.4.1, we describe the adaptation of the mean-field algorithm used in Multi-Context work. In Section 2.5 we describe the Multi-Context attention model for human pose estimation. In Section 2.6 we summarize this chapter.

2.1 HMM: Hidden Markov Model

As it is mentioned in (Sucar, 2015), a Hidden Markov Model (HMM) is a Markov chain where the states are not directly observable. We can view an HMM as a double

stochastic process, a hidden stochastic process that we cannot directly observe and a second stochastic process that produces the sequence of observations given the first process. HMMs are useful to model time series data and could be viewed as a quantization of a system's configuration space into a small number of discrete states, together with probabilities for transitions between states (Brand et al., 1997). It has been applied in signal processing, natural language processing, gesture recognition, etc. Figure 2.1 shows the graphical model representation of an HMM.

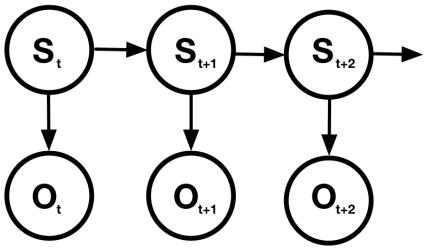

Figure 2.1: Hidden Markov Model. This graphical model shows two series of random variables, state $S_t = q_i$ and observation $O_t = o_j$, both at time t where $i = [1, ..., n]$ and $j = [1, ..., m]$.

Formally, a hidden Markov model is represented by $\lambda = \{A, B, \Pi\}$ where the parameters are the following:

- Set of states: $Q = \{q_1, q_2, ..., q_n\}$
- Set of observations: $O = \{o_1, o_2, ..., o_m\}$
- Vector of initial probabilities: $\Pi = \{\pi_1, \pi_2, ..., \pi_n\}$ where $\pi_i = P(S_0|q_i)$
- Matrix of transition probabilities: $A = \{a_{ij}\}, i = [1, ..., n], j = [1, ..., n]$ where $a_{ij} = P(S_t = q_i|S_{t-1} = q_j)$
- Matrix of observation probabilities: $B = \{b_{ij}\}, i = [1, ..., n], j = [1, ..., m]$ where $b_{ij} = P(O_t = o_j|S_t = q_i)$

where n is the number of states and m the number of observations; S_o is the initial state. This model involves the following assumptions: i) the probability of the

current state only depends on the previous state (Markov property). ii) the transition and observation probabilities do not change over time (it is a stationary process). iii) the observations only depend on the current state.

Given an HMM, there are three base tasks of interest in most domains: i) estimation of the probability of a sequence of observations, (ii) estimation of the most probable state sequence that produces the given sequence of observations, (iii) adjustment of the model parameters given a sequence of observations.

Our model is based on this concept, we follow up the HMMs with an extension of this approach. Since this model represents the states of a process within a single variable when the problem that is being tackled requires the interaction of more random variables this state interaction represented within a single variable explodes since at a glance it is required to consider the Cartesian product of the state space. However, as it is described in later sections, it is possible to make a relaxation of the cartesian product and therefore split the states on one variable per interacting process only taking into consideration a subset of all the possible states combination. In this case, we work with two random variables X and Y which represents spatial coordinates. The task of the model is focused in the estimation of the most probable sequence of states considering the interaction of X and Y given the part-detector observations, this task translates into the most probable movement of a body joint within a time interval.

2.1.1 Viterbi algorithm for HMMs

The Viterbi Algorithm in its more general form may be viewed as a solution to the problem of maximum *a posteriori* probability (MAP). Therefore, it is an efficient algorithm to estimate the state sequence of discrete time and finite state Markov process. It was firstly proposed in 1967 as a method for decoding convolutional codes

(Viterbi, 1967).

As it is mentioned in (Sucar, 2015), the Viterbi algorithm helps to obtain the most probable sequence of states given a sequence of observations. The formal definition of this statement is as follows:

$$\delta_t(i) = MAX_i[P(s_1, s_2, ..., s_t = q_i, o_1, o_2, ..., o_t|\lambda)] \tag{2.1}$$

We can obtain the most probable sequence through the following expression:

$$\delta_{t+1}(i) = MAX_i[\delta_t(i)A_{ij}]B_j(o_{t+1}) \tag{2.2}$$

The Viterbi algorithm is composed of four phases: initialization, recursion, termination, and backtracking. It requires an additional variable, $\psi_t(i)$, that stores for each state i at each time step t the previous state that gave the maximum probability. It is defined in Algorithm 1.

The initialization phase takes the prior probabilities and the emission values for the observation at time $t = 1$ for all the states as the following expression indicates $\pi_i B_i(0_1)$, where π_i is the prior probability and $B_i(0_1)$ is the emission function for the states $i = 1...N$ given the observation 0_1.

The recursion phase, implemented as dynamic programming, takes the maximum probability at each states as follows:

$$\delta_t(j) = MAX_i[\delta_{t-1}(i)A_{ij}]B_j(0_t) \tag{2.3}$$

$$\psi_t(j) = ARGMAX_i[\delta_{t-1}(i)A_{ij}] \tag{2.4}$$

for each state index $j = 1...N$ at each time $t = 2...T$. It also stores index i that corresponds to the maximum probability to keep track of the paths.

Since the dynamic programming implementation keeps track of the maximum probabilities at each time t, in the termination phase the path that maximized the whole sequence ends with the maximum probability therefore it is selected the maximum node as follows:

$$P^* = MAX_i[\delta_T(i)] \tag{2.5}$$

$$q_T^* = ARGMAX_i[\delta_T(i)] \tag{2.6}$$

where P^* is the probability of the most probable sequence and q_T^* is the index of the last state of this sequence. Finally the backtracking phase follows the stored references from this last node to the beginning of the sequences with the following expression:

$$q_{t-1}^* = \psi_t(q_t^*) \tag{2.7}$$

for time $t = T...2$. q^* stores the most probable sequence.

As this algorithm is described, it works only through the maximization of one process at a time. Our model tries to compute the most probable behavior of a body joint. Since the movement is represented in a Cartesian space, representing the required states within a single variable leads to an explosion of the state space, and therefore the Viterbi algorithm would take more computational time and also it could lack representation of some states. Nevertheless, the Viterbi algorithm keeps computational efficiency through the dynamic programming approach. The CoHMM extension allows the model to represent the interaction of these two interacting processes while keeping efficiency.

Algorithm 1 The Viterbi algorithm

Input: HMM,λ; Observations sequence,O; Number of states$N$; Number of observations,T

for $i = 1$ to N **do**
    (Initialization)
    $\delta_1(i) = \pi_i B_i(0_1)$
    $\psi_1(i) = 0$
end for
for $t = 2$ to T **do**
    for $j = 1$ to N **do**
        (Recursion)
        $\delta_t(j) = MAX_i[\delta_{t-1}(i)A_{ij}]B_j(0_t)$
        $\psi_t(j) = ARGMAX_i[\delta_{t-1}(i)A_{ij}]$
    end for
end for
(Termination)
$P^* = MAX_i[\delta_T(i)]$
$q_T^* = ARGMAX_i[\delta_T(i)]$
for $t = T$ to 2 **do**
    (Backtracking)
    $q_{t-1}^* = \psi_t(q_t^*)$
end for

2.2 CoHMM: Coupled Hidden Markov Model

The CoHMMs add conditional probabilities between the hidden state variables of different HMMs as it is illustrated in Figure 2.2. This model has multiple state variables that are temporally coupled via matrices of conditional probabilities. Within a standard HMM the current state of the system is represented by a single discrete variable, any information about the history of the process needed for future inferences must be reflected in the current value of this state variable. However, many interesting systems are composed of multiple interacting processes (Brand et al., 1997). This is a common case for systems that try to model both space and time.

The implementation developed by (Brand et al., 1997) is a solution to tackle the

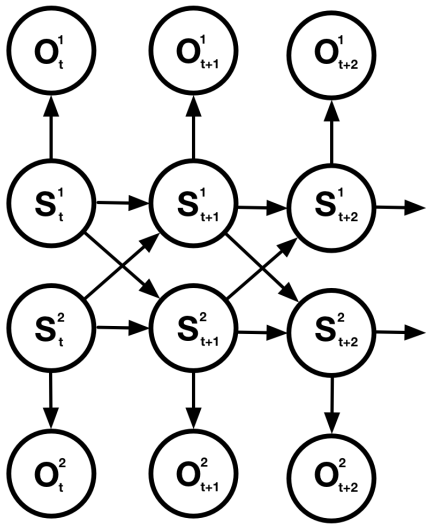

Figure 2.2: Coupled HMM. Graphical model with two coupled chains for processes S_t^1 and S_t^2 at time t. Each process has its observations O_t^1 and O_t^2 at time t, respectively.

gesture recognition problem based on two-handed actions. Their contribution relies on an extension of the restrictive Markovian assumptions about the process that generates the signal by adding more than one signal source. This approach consists of a projection between the models obtained from the involved processes, which are the HMMs and the joint HMM generated with the Cartesian product of all the states. The implementation uses the Viterbi algorithm to find the maximum likelihood model. Moreover, the introduced algorithm N-Heads to couple both processes.

They perform experiments with data related to T'ai Chi Ch'uan (Chinese martial art and meditative exercise). They split the signals extracted from each arm. The behind reasoning relies on the assumption that both arms are not entirely independent, but also their interaction is not solely tied. Therefore an interaction modeling approximation may be sufficient. The dataset is composed of 52 gestures. In comparison to other models such as standard HMMs and Linked HMMs (a simplification of CHMMs with symmetric noncausal joint probabilities between chains), this coordination could be interpreted as noise while within the CoHMMs the variations are useful to model the interaction through the coupling probabilities. The results were

the increment of accuracy in the coupled model in comparison to standard HMMs and Linked HMMs.

We model the human skeleton through a set of CoHMMs, one per body joints. Therefore this model is focused on the representation of a single body joint at a time. The skeletal structure of the body joint with the addition of temporal dependencies lead to a dense graph in which the inference process is expensive. Therefore, we break up the spatial dependencies to focus only on the temporal ones per joint which lead us to a set of temporal chains. Then if we work with the problem by splitting the state space in more variables to avoid the cartesian combination through a CoHMM, we can optimize each variable. However, it is required to perform another relaxation within the Viterbi algorithm since we are trying to optimize the interaction between more variable, we have again a cartesian product state space. An approximation to this problem could be relaxing the assumption that it is required to visit all possible paths to obtain the most probable sequence.

2.2.1 Viterbi algorithm for CoHMMs

The adaptation of the Viterbi algorithm was performed by (Brand et al., 1997) through the N-heads dynamic programming algorithm. This algorithm improves the naive cartesian product approach resulting from the chains coupling. The dynamic programming technique lets to collect statistics on an exponential number of possible paths through an HMM trellis in polynomial time. It requires $O(TN^2)$ time to collect statistics through dynamic programming in a trellis of length T and width N. Therefore a coupled HMM of C chains has a joint trellis that is in principle N^C states wide. This makes the dynamic programming problem bounded by $O(TN^{2C})$. (Brand et al., 1997) shows that it is possible to relax the assumption that every transition must be visited, thereby obtaining an $O(T(CN)^2)$ algorithm that closely approximates the full combinatoric results. To guarantee this complexity two con-

ditions must be followed, i) no higher than O(N) paths heads can be tracked and ii) every component state must be visited.

To satisfy the visiting criterion, let the tuple $\{head, sidekick\}$ be called "path", then each component state at time t must be head of some path. Therefore coupling two HMMs of N, M states takes $N + M$ heads each with a sidekick. Since every component state must be a head, we only need to maximize the MAP sidekicks through equation 2.8. Therefore, the equation $Q = argmax^N \sum_{\mathcal{X}} \Big\langle logP(\mathcal{M}|\mathcal{X}) \Big\rangle$, where $\mathcal{M}$ is the model and $\mathcal{X}$ the training data, could be approximated accordingly to (Brand et al., 1997) with the following expression:

$$Q_1 = argmax^N_{s \in S'^{\{\}}} \sum_{\mathcal{X}} \Big\langle \Big(logP_{s'_1} + logp_{s'_1}(o'_1) \Big) + \sum_{t=2}^{T} \Big(logP_{s'_t|s'_{t-1}} + logP_{s'_t|s_{t-1}} + logp_{s'_t}(o'_t) \Big) \Big\rangle \tag{2.8}$$

where s and s' are head and sidekick states within a subset $S'^{\{\}}$ of all possible paths. o and o' are observation from the head's and sidekick's chain. So $P_{s'_t|s'_{t-1}}$ is the transition probability between sidekick states and $P_{s'_t|s_{t-1}}$ is the conditional probability of the interaction between chains. $p_{s'_t}(o'_t)$ is the sidekick's output probability.

Algorithm 2 shows the N-Heads algorithm, where the additional functions maxH and maxS return the variables for the backtracking process in the correct order to preserve the order of the tuple (X, Y), which represents the coupled random variables of the two target processes in this work. It is important to maintain the order of the tuple within the operations since the random variables denote spatial coordinates.

In each step of the algorithm, it seeks the MAP density { head, sidekick } pairs given all antecedent paths. Unlike ordinary Viterbi algorithm, in which for each head at time t it chooses an antecedent path in $t - 1$, N-Heads must also choose a sidekick in t. This selection can be done in two steps: i) for each antecedent path in

$t-1$, select MAP sidekicks in t with Equation 2.8; ii) for each head in t, select the antecedent path and associated sidekick that maximizes the new head's posterior.

In other words, in the first step it maximizes the sidekick's probability through the Equation 2.8. Which requires the transition probability, the conditional probability, and the emission function. Since all states are already tracked as heads, according to the assumptions of this algorithm, the remaining task is for each state iterate over all sidekicks and choose the one that maximizes the previous equation. In the second operation there is already a set of tuples $\{head, sidekick\}$. Therefore the remaining task is to associate these tuples to the tuples at time $t-1$. This is done through the maximization of the head's posterior probability. This operation could be done in a time bounded by $O(N^2)$.

Figure 2.3 shows the comparison between an HMM with three states and a CoHMM with two chains with three and four states. The standard HMM trellis shows a single path through the optimization process, and the reduced CoHMM trellis shows $N+M$ states. The latter trellis illustrates with solid lines the transition probabilities while with dashed lines it illustrates the coupling probabilities.The relaxation of this algorithms is made by maintaining every state as a head on each time slice. Therefore the dashed lines are the ones the model maximize.

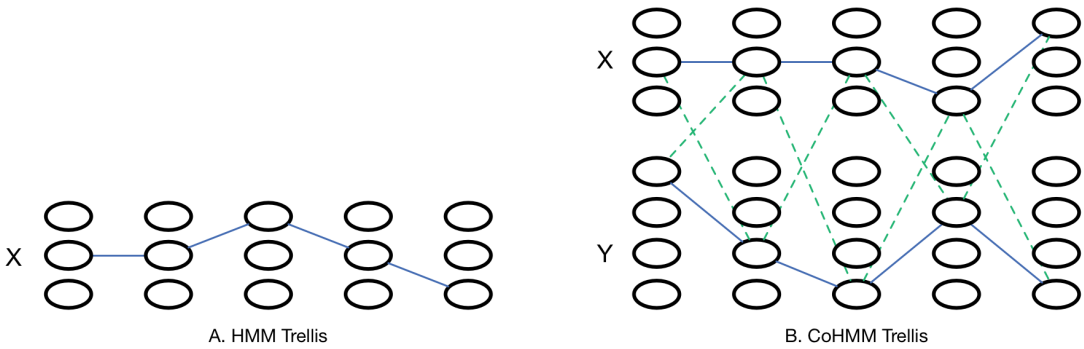

Figure 2.3: Trellis for HMM and CoHMM. A single path through a standard HMM trellis and a trellis reduction for CoHMM. The reduction involves $N+M$ states at each time step, where N and M are the number of states of the two processes. Solid lines illustrates the transition probabilities while the dashed lines are the coupling probabilities.

The algorithm also requires the transition matrix split by transitions within the same process and coupling probabilities from one process to another. Let $Cij = x_i, y_j$ be the coupled transition matrix, where $i = 1...N$ and $j = 1...M$ for N and M states for X and Y processes respectively. The following equations denote the transition matrix factorization used within the N-Heads algorithm.

$$P_{x_i|x_j} = \sum_l P_{y_l} \sum_k P_{c_{ik}|c_{jl}} \tag{2.9}$$

$$P_{y_k|y_l} = \sum_j P_{x_j} \sum_i P_{c_{ik}|c_{jl}} \tag{2.10}$$

$$P_{x_i|y_l} = \sum_j P_{x_j} \sum_k P_{c_{ik}|c_{jl}} \tag{2.11}$$

$$P_{y_k|x_j} = \sum_l P_{y_l} \sum_i P_{c_{ik}|c_{jl}} \tag{2.12}$$

To make a clear distinction of transition probabilities and coupled probabilities we use the notation of the factorizations as follows:

$$a_{x_i|x_j} = P_{x_i|x_j} \tag{2.13}$$

$$b_{y_k|y_l} = P_{y_k|y_l} \tag{2.14}$$

$$c_{x_i|y_l} = P_{x_i|y_l} \tag{2.15}$$

$$d_{y_k|x_j} = P_{y_k|x_j} \tag{2.16}$$

where a and b are transition probabilities within X and Y processes respectively

while c and d are coupling probabilities for X and Y.

This algorithm allows the model to couple the interaction of the x and y coordinates of a given body joint. Each body joint is split from the body skeletal structure to reduce dependencies and therefore the computational overhead. So the process is focused on the temporal information. The random variables X and Y are associated directly with the x and y axes. Then the interaction between these variables across time is intended to model a proper behavior of a body joint.

Algorithm 2 N-Heads algorithm

Input: CoHMM, λ; Observations sequence, O; Number of states, N; Number of observations, T

(INITILIZATION)
for $i = 1$ to N **do**
    $\delta_1(x_i) = \pi_{x_i} B_{x_i}(0_{x,1})$
    $\delta_1(y_i) = \pi_{y_i} B_{y_i}(0_{y,1})$
    $\psi_1(x_i) = 0$
    $\psi_1(y_i) = 0$
end for
for $i = 1$ to N **do**
    $\delta'_1(x_i) = MAX_i[\delta_1(y_i))]$
    $\delta'_1(y_i) = MAX_i[\delta_1(x_i))]$
    $\psi'_1(x_i) = ARGMAX_i[\delta_1(y_i))]$
    $\psi'_1(y_i) = ARGMAX_i[\delta_1(x_i))]$
end for
(RECURSION)
for $j = 2$ to N **do**
    $\phi(x_j) = ARGMAX_i[b_{y_i,\psi'_t(x_j)} + d_{y_i,x_j} + B_{y_i}(O_{y,t})]$
    $\phi(y_j) = ARGMAX_i[a_{x_i,\psi'_t(y_j)} + c_{x_i,y_j} + B_{x_i}(O_{x,t})]$
end for
for $t = 2$ to T **do**
    for $j = 1$ to N **do**
        $\delta_t(x_j) = MAX_i[\delta_{t-1}(x_i)a_{i,j}]B_j(0_{x,t})$
        $\psi_t(x_j) = ARGMAX_i[\delta_{t-1}(x_i)a_{i,j}]$
        $\psi'_t(x_j) = \phi(\psi_t(x_j))$
        $\delta_t(y_j) = MAX_i[\delta_{t-1}(y_i)b_{i,j}]B_j(0_{y,t})$
        $\psi_t(y_j) = ARGMAX_i[\delta_{t-1}(y_i)b_{i,j}]$
        $\psi'_t(y_j) = \phi(\psi_t(y_j))]$
    end for
end for
(TERMINATION)
$P^* = MAX_i[\delta'_T(x_i), \delta'_T(y_i)]$
$\Psi_t, q^*_{h,T} = \text{maxH}_i((\psi_t(x_i), \delta_T(x_i)), (\psi_t(y_i), \delta_T(y_j)))$
$\Psi'_t, q^*_{s,T} = \text{maxS}_i((\psi'_t(x_i), \delta'_T(x_i)), (\psi'_t(y_i), \delta'_T(y_j)))$
(BACKTRACKING)
for $t = T$ to 2 **do**
    $q^*_{h,t-1} = \Psi_t(q^*_{h,t})$
    $q^*_{s,t-1} = \Psi'_t(q^*_{s,t})$
end for

2.3 CNN: Convolutional Neural Networks

As it is mentioned in (LeCun et al., 1998), a convolutional network is a multilayer perceptron explicitly designed to recognize two-dimensional shapes with a high degree of invariance to translation, scaling, skewing, and other forms of distortion. This difficult task is learned in a supervised manner using a network whose structure includes the following types of constraints: feature extraction, feature mapping, and subsampling.

Each type of constraints consists on the following statements: i) on the feature extraction constraint process each neuron takes its synaptic inputs from a local receptive field in the previous layer, thereby forcing it to extract local features. ii) on the feature mapping constraint process each computational layer of the network is composed of multiple feature maps, with each feature map taking the form of a plane within which the individual neurons are constrained to share the same set of synaptic weights. iii) on the subsampling constraint process each convolutional layer is followed by a computational layer that performs local averaging and subsampling, whereby the resolution of the feature map is reduced. This operation has the effect of reducing the sensitivity of the feature map's output to shifts and other forms of distortion.

According to (LeCun et al., 1998), Figure 2.4 shows a LeNET-5 network as an example of an architectural layout. It is composed of 7 layers, without the input, which contains trainable parameters or weights. The input in this example is 32x32 pixel image. The notation of the figure denotes the convolutional layers, subsampling layers and fully-connected layers as Cx, Sx, Fx, where x is the layer index. The convolutional layer C1 is composed of 6 planes as feature maps of size 14x14. The weights are passed to S2 through a 2x2 neighborhood in C1. This four input values of S2 are added, then multiplied by a trainable coefficient, and added to a trainable

bias. The result is passed through a sigmoidal function. The subsampling operation reduces the dimensionality of the feature maps through the size of the receptive fields since these are non-overlapping. The layer C3 uses a receptive field of size 5x5 leading to feature maps of size 10x10.

In this case, each plane of the feature map is connected to a subset of S's feature maps in the same location with the receptive field. The latter statement allows to reduce the connections and to break symmetry which could retrieve complementary features. Layer S4, similarly to S2, uses a neighborhood of 2x2 to reduce the feature maps to 5x5. In layer C5, the convolution is performed through a receptive field of size 1x1.

The operations up to layer F6 are performed through the dot product. It is computed with both the input vector and the weight vector, then a bias is added. The result is represented as a_i for a unit i within a feature map. To compute the state of the unit i, denoted as x_i, a squashing sigmoid function is fed with a_i. So x_i is computed with the following expressions:

$$x_i = f(a_i) \tag{2.17}$$

$$f(a) = Atanh(Sa) \tag{2.18}$$

where A is the amplitud of the function and S determines its slope at the origin.

Finally, the output layer is composed of Euclidean Radial Basis Function (RBF), one for each target class, in this example with 84 inputs each. The outputs of each RBF unit y_i is computed through the distance between its input vector and its parameters vector with Equation 2.19.

$$y_i = \sum_j (x_j - w_{ij})^2 \tag{2.19}$$

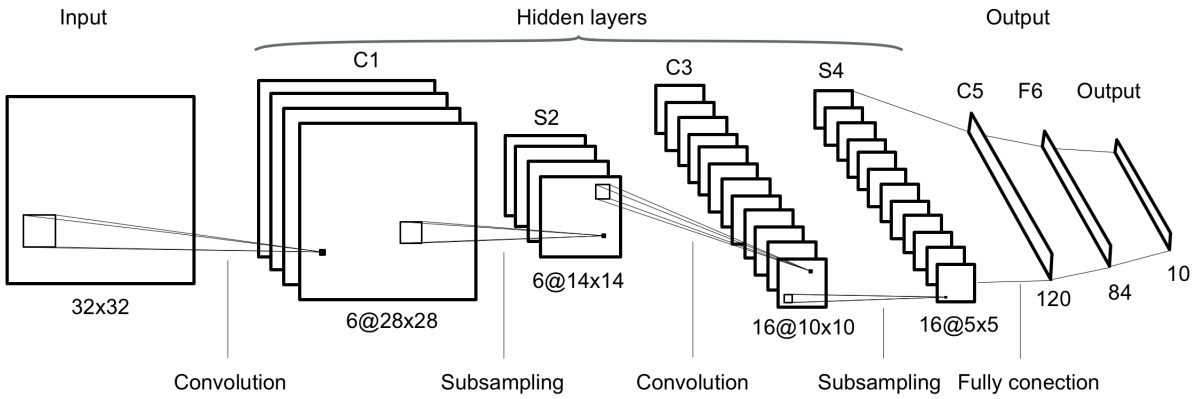

Figure 2.4: Convolutional network for image processing.

The architecture that follows the base estimation in this work follows the CNN concepts to learn features at different scales. The hourglass network (Newell et al., 2016) adapt a CNN architecture to tackle the pose estimation problem by retrieving features from different resolution through downsampling and upsampling operations. Then when it merges the features, it is intended to represent contextual information. Due to the complexity of the human body, this architecture works with local features to detect the position of the limb while the global features help to maintain the body structure.

2.3.1 Stacked hourglass network

The end-to-end architecture stacked hourglass networks (Newell et al., 2016) was introduced to tackle the human pose estimation problem. It takes into consideration different scale along the networks to retrieve spatial relationship within the image. Its goal is to determine human body keypoints from RGB images. This network is composed of a series of symmetrical hourglass modules. These modules

are formed by a set of symmetrical pooling and upsampling operations to retrieve useful information from low and high resolutions. The advantages of the resolutions diversity are the detection of local features such as faces or hands, while the global features help with the understanding of the full body. Then with different scales, this architecture is intended to recognize a person's orientation, the arrangement of the limbs, and the relationships of adjacent joints.

To go from high to low resolution an hourglass unit includes convolutional and max-pooling layers as it is shown in Figure 2.5. At each max-pooling step the network branches off, in which more convolutions are applied to the pre-pooled feature maps, also called as residual. The lowest resolution is 4x4. Then, the upsampling process uses a nearest neighbors approach in the lowest resolution of two adjacent ones, followed by an element-wise addition of the two sets of features, the pre-pooled and convolved features with the upsampled ones. These branches help to combine features of the different scales and to preserve spatial information; this step is also known as a residual module. At the end of the network, after several stacked hourglass modules, there are two 1x1 layers to obtain the final result which is a set of heatmaps. Each heatmap contains the probability of the presence of a body joint on every pixel.

The described architecture is the CNN implementation used in the Multi-Context work (Chu et al., 2017) which is focused on retrieving information with different semantics. Thereby, it represents the body skeleton from different contexts.

2.4 CRF: Conditional Random Fields

As it is mentioned in (Sucar, 2015), a limitation of HMMs is that they usually assumed that the observations are independent given each state variable. There are applications in which these independence assumptions are not appropriate. HMMs

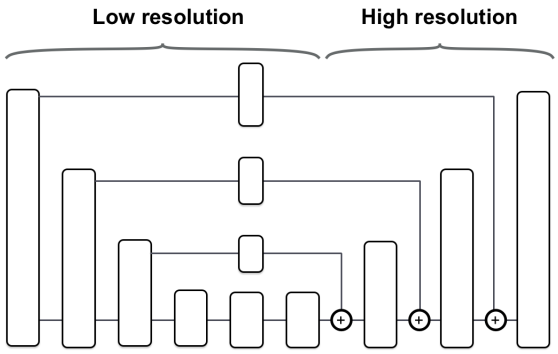

Figure 2.5: Hourglass module. This module is composed of symmetrical pooling and upsampling layers where for each max-pooled layer the is a upsampling one. For each max-pooling layer the network branches off to be element-wise added in its upsampling layers. The lowest resolution of two adjacent ones is upsampled using a nearest neighbor approach.

are generative models, which represent the joint probability distribution as the product of local functions based on the independence assumptions. If these conditional independence assumptions are removed, the models become intractable. One alternative that does not require these assumptions is Conditional Random Fields (CRF).

A conditional random field is an undirected graphical model globally conditioned on X, the random variable representing observations. Conditional models are used to label an observation sequence X by selecting the label sequence Y that maximizes the conditional probability $P(Y|X)$. The conditional nature of such models means that no effort is wasted on modeling the observations, and one is free from having to make unnecessary independence assumptions. A CRF models pixel labels as random variables that form a Markov Random Field (MRF) when conditioned upon a global observation when it is used in pixel label predictions.

2.4.1 CRF as a Recurrent Neural Network

In (Zheng et al., 2015) was proposed a convolutional neural network in combination with Conditional Random Fields (CRF) to tackle the pixel-level labeling tasks. The combination of the properties of both models

They formulated the mean-field approximate inference for the CRF with Gaussian pairwise potentials as Recurrent Neural Networks called CRF-CNN, the Algorithm 3 shows this adaptation. Therefore the network CRF-CNN is plugged in as a part of the CNN. Then the work by (Chu et al., 2017) used this mean-field approximate inference to model the spatial correlations among neighborhood body joints in their multi-context work.

The Initialization phase of the Algorithm 3 is modeled as a softmax function which is common operation used within deep learning.

The Message Passing phase is implemented in dense CRFs by applying M Gaussian filters to Q values. The Gaussian coefficients are computed based on image features that represent how strongly a pixel is related to others, to avoid the complexity of span every filter's receptive field in the image an approximation is used. Then during back-propagation, the error derivatives of the filter inputs are calculated by sending the error derivatives of the filter outputs through the same M Gaussian filters in reverse direction.

The Weighting Filter Outputs phase is performed by taking a weighted sum of the M filters outputs from the message passing phase, for each class label l. This can be implemented as 1 x 1 convolution with M input channels when each class label is considered individually.

The Compatibility Transform phase can be viewed as a 1 x 1 convolution, where in this case it assigns the same penalty for all different pairs of labels.

In the Adding Unary Potentials phase the output from the compatibility transform stage is subtracted element-wise from the unary inputs U, therefore, no parameters are required and the error differential can be copied from the differentials at the output of this step to both inputs with the appropriate sign.

Finally, the Normalization step of the iteration can be considered as another softmax operation with no parameters. Differentials at the output of this step can be passed on to the input using the softmax operation's backward pass.

Algorithm 3 Mean-field in dense CRFs, broken down to common CNN operations.

1: $Q_i(l) \leftarrow \frac{1}{z} exp(U_i(l))$ for all i ▷ Initialization
2: **while** not converged **do**
3: $\tilde{Q}_i^m(l) \leftarrow \sum_{j \neq i} k^{(m)}(f_i, f_j) Q_j(l)$ for all m ▷ Message Passing
4: $\breve{Q}_i(l) \leftarrow \sum_m w^{(m)} \tilde{Q}_i^m(l)$ ▷ Weighting Filter Outputs
5: $\hat{Q}_i(l) \leftarrow \sum_{l' \in \iota} \mu(l, l') \breve{Q}_i(l)$ ▷ Compatibility Transform
6: $\breve{Q}_i(l) \leftarrow U_i(l) - \hat{Q}_i(l)$ ▷ Adding Unary Potentials
7: $Q_i \leftarrow \frac{1}{z_i} exp(\breve{Q}_i(l))$ ▷ Normalizing
8: **end while**

This algorithm is included in the Multi-Context implementation (Chu et al., 2017) to learn spatial correlations instead of a softmax function. This work is focused on a strong analysis of contextual information. Therefore the inclusion of this spatial correlation in combination to the analysis at different resolutions help in a semantic interpretation of the features and then a better estimation of the complete skeleton.

2.5 Part-Detector on Still Images

Multi-Context attention for human pose estimation (Chu et al., 2017) is a CNN architecture to extract multi-context features to improve the results on the pose estimation problem. The end-to-end framework is composed of an adaptation of the stacked hourglass network (Newell et al., 2016) in which they replace the residual

units with nested micro hourglass units called Hourglass Residual Units (HRUs) as it is shown in Figure 2.6; and a CRF to model the correlations among neighboring regions. The design that follows this work allows them to focus on different granularity from local regions to global semantics.

This work is divided into three modules: i) Multi-Resolution Attention, it is implemented through an hourglass architecture; ii) Multi-Semantics Attention, it is implemented by merging the features of different scales; iii) and Hierarchical Attention Mechanism, it is implemented by branching the CNN architecture to focus on each body joint. The CRF model takes place globally within features of Multi-Semantics Attention process and within each branch of the Hierarchical Attention Mechanism.

The architecture is an 8-stack hourglass network. The input images are fed with dimensions 256 x 256, and the output heatmap dimensions are P x 64 x 64, where P is the number of body parts.

Each stack extracts different semantics from a local appearance in the lower stacks to global representations in higher stacks. Then the combination of the attention maps generated at each stack is intended to encode various semantic meanings. This process help to tackle occlusion since body joints that are not visible could be inferred from global information.

Within the whole process, the attention maps are generated over the hourglass stacks to retrieve different semantics information since the first stacks are shallower than the last ones. Figure 2.6 shows the composition of an hourglass stack. A set of attention maps from different resolutions are extracted and merged to refine the feature map before this result is fed to the next stack. As it is described in previous sections, the hourglass network is composed of a series of max-pooling layers followed by a series of upsampling layers. In this architecture of hourglass

network, the residual unit is composed of HRUs.

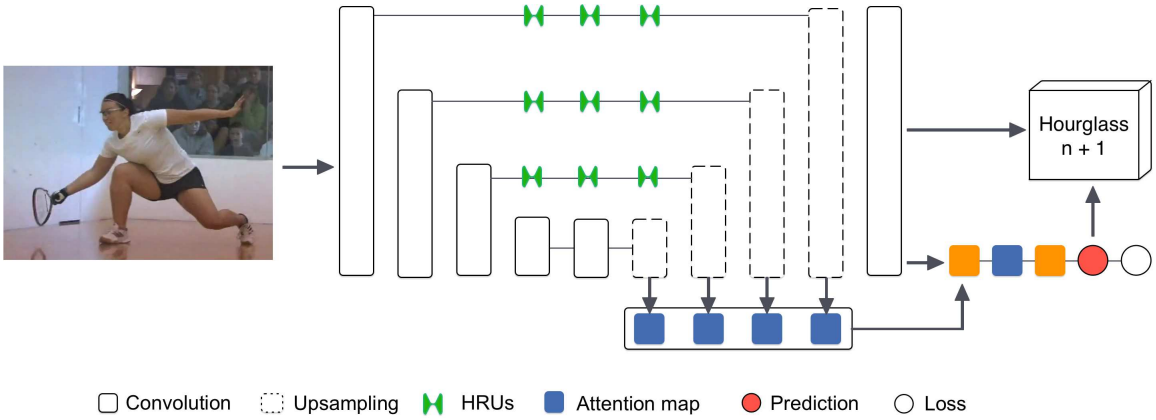

Figure 2.6: Multi-Resolution Attention module. Feature maps are generated with different resolutions at each hourglass stack. This maps are merged and feed into the next hourglass stack (Chu et al., 2017).

At the end of the stacks, the refinement of the features is done by first working with the complete body skeleton, then the network branches to focus on each joint. These new branches guarantee that the attention maps in the entire process are guided to each body part at the end. To capture the features related to each body part p the model uses the spatial correlations through CRFs instead of the softmax function within the CNN. The CRF helps model the correlations of the attention maps already extracted. This process is shown in Figure 2.7.

The branching starts by computing the feature map h_1^{att} with the following equation:

$$h_1^{att} = f \star \sum_r \phi_{r \to 64}$$

where $\star$ indicates the channel-wise Hadamard matrix product operation. f comes from the last layer of the stacked network. $\sum_r \phi_{r \to 64}$ denotes the summation of all the attention maps of the stacked network passed to the CRF process of Equation 2.22 and upsampled to resolution $r = 64$.

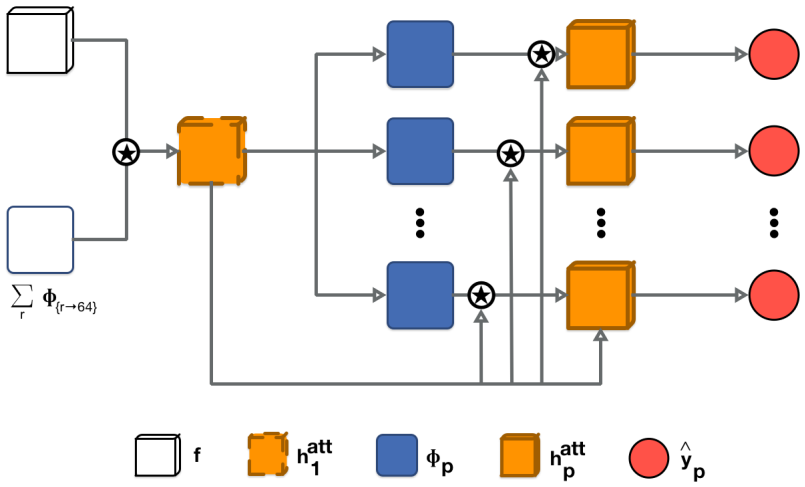

Figure 2.7: Hierarchical Holistic-Part Attention. Visualization of the feature map being refined to focus on each body part (Chu et al., 2017).

The features h_1^{att} are then refined to focus on each body part. They are fed on a separate branch for each part p to the CRF process of Equation 2.22. The CRF process gives the attention map ϕ_p, and further refinement is represented by h_p^{att} for each body part p.

To learn a spatial correlation kernel within the CRF implementation, the work by (Chu et al., 2017) uses a mean-field approximation (Zheng et al., 2015). The attention map is modeled as a two class problem. Denote $y_l = 0, 1$ as the attention label at the $i-th$ location. In the CRF model, the energy of a label assignment $y = \{y_l | l \in \mathbb{L}\}$ is as follows:

$$E(z) = \sum_l y_l \psi_u(l) + \sum_{l,k} y_l w_{l,k} y_k \tag{2.20}$$

where $\psi(y_l) = g(h, l)$ is the unary term that measures the inverse likelihood of the position l taking the attention label $y_l = 1$. $w_{l,k}$ is the weight for compatibility between y_l and y_k. Given the image I, the probability of the label assignment y is $P(y|I) = \frac{1}{Z} exp(-E(y|I))$, where Z is the partition function. The probability

for $y_l = 1$ is obtained iteratively using the mean-field approximation described in Section 2.4.1 as follows:

$$\Phi(y_l = 1)_t = \sigma(\psi_u(l) + \sum_k w_{l,k}\Phi(y_k = 1)_{t-1}) \tag{2.21}$$

where $\sigma(a) = 1/(1 + (exp(-a))$ is the sigmoid function. $\psi_u(l)$ is obtained by convolution from the features h_1^{att} obtained during the CNN workflow. $\sum_k w_{l,k}\Phi(y_k = 1)$ is implemented by convolving the estimated attention map Φ_{t-1} at the stage $t-1$ with the filters. Initially, $\Phi(y_i = 1)_1 = \sigma(\psi_u(i))$. In summary the attention map Φ at the stage t can be formulated as follows:

$$\Phi_t = \mathcal{M}(s, \mathrm{W}^k) \begin{cases} \sigma(\mathrm{W}^k * s) & t = 0 \\ \sigma(\mathrm{W}^k * \Phi_{t-1}) & t = 1, 2, 3 \end{cases} \tag{2.22}$$

where $\mathcal{M}$ denotes a sequence of weight-sharing convolutions for the mean field approximation. W^k denotes the spatial correlation kernel. W^k is shared across different time steps. s indicates an initial convolution with the target feature map. In this network, they used three steps of recursive convolution within the mean field approximation.

To make the branching of the network the features are computed with the following expressions:

$$\begin{aligned} s_p &= g(\mathrm{W}_p^a * h_1^{att} + b), \\ \Phi_p &= \mathcal{M}(s_p, \mathrm{W}_p^k) \end{aligned} \tag{2.23}$$

where $p \in \{1, ..., P\}$ represents the P body joints. W_p^a are the parameters for obtaining the summarization map s_p of part p. Then the part attention map Φ_p is

combined with the refined feature map h_1^{att} to obtain the refined feature map for part p as follows:

$$h_p^{att} = h_1^{att} \star \Phi_p \tag{2.24}$$

Finally the heatmap for the p body joint is obtained from the refined features h_p^{att} from the expression $y_p = w_p^{cls} * h_p^{att}$, where w_p^{cls} is the classifier. This guarantees that each body part has it own attention map. The attention map of each body part is shown in Figure 2.7.

Multi-Context attention for human pose estimation focuses on retrieving contextual interpretation from the whole process which means that they have a strong focus on taking into consideration multiple resolutions and neighboring regions. The analysis goes from local detection to a global understanding of the human structure. This architecture has a strong relation to the work of this thesis which adds a further global analysis by the computation of the most probable sequence of movement of a given body joint.

2.6 Chapter summary

The proposed method requires the description of theoretical concepts based on Graphical Models and Convolutional Neural Networks. Individually, the concepts involved in this thesis are the following: Hidden Markov Models (HMMs), Viterbi Algorithm, Coupled Hidden Markov Models (CoHMMs), N-Heads algorithm, Convolutional Neural Networks (CNN), Conditional Random Fields (CRFs) and the base part-detector Multi-Context attention for human pose estimation.

Based on the previous concepts our architecture follow up the HMMs. An HMM

could be seen as a double stochastic process, a hidden stochastic process that we cannot directly observe and a second stochastic process that produces the sequence of observations given the first process (Sucar, 2015). This model performs three basic tasks. i) estimation of the probability of a sequence of observations, (ii) estimation of the most probable state sequence that produces the given sequence of observations, and (iii) adjustment of the model parameters given a sequence of observations.

If our model represents the states of a process within a single variable when the problem that is been tackled requires the interaction of more random variables this state interaction represented within a single variables explodes since at a glance it is required to consider the cartesian product of the state space. However, it is possible to make a relaxation of the cartesian product and therefore split the states on one variable per interacting process and only taking into consideration a subset of all the possible states combination.

To achieve this efficiency, we use Coupled Hidden Markov Models (CoHMM). This model has multiple state variables that are temporally coupled via matrices of conditional probabilities. In this case, we work with two random variables X and Y which represents spatial coordinates. The task of the model is focused in the estimation of the most probable sequence of states considering the interaction of X and Y given the part-detector observations, this task translates into the most probable movement of a body joint within an interval of time.

The algorithm that helps to compute the most probable sequence is the N-Heads algorithm. It makes a relaxation of the assumption that all possible paths must be visited. Improving the complexity from a naive approach $O(TN^{2C})$ to an improved one $O(T(CN)^2)$ where N is the number of states, C is the number of coupled chains, and T is the number of observations.

As a source of initial estimations, we use the implementation done by Multi-

Context Attention model for human pose estimation (Chu et al., 2017). This part-detector is implemented upon a Stacked Hourglass architecture (Newell et al., 2016). It also includes an implementation of Conditional Random Fields (CRFs) within the CNN architecture. The main idea behind the part-detector is the retrieval of contextual information from the features extraction process at different granularities and also take into consideration neighboring regions through CRFs instead of a softmax function. Within our model, we built upon the part-detector architecture a further contextual process by working with the estimation of complete sequences.

Chapter 3

Related Work

In this Chapter we review the relevant research about human pose estimation. The organization of the chapter is as follows. Section 3.1 gives an overview about the entire pose estimation problem to highlight where this work takes place into the taxonomy of solutions. Section 3.2 summarizes contributions on still images. Section 3.3 describes the solutions that work with sequence of images including approaches based on CNNs, graphical models, and hybrid methods. Finally Section 3.3 presents the most common datasets developed in recent years.

3.1 Human pose estimation overview

Figure 3.2 shows a taxonomy of the approaches followed by researchers for the human pose estimation problem. The classification of this approaches is taken from the survey developed by (Gong et al., 2016). The main division of the solutions is based on three main branches: features, human body models and methodologies. We will describe first this three components in order to build a clear context about the path followed by researchers and then we will give more details about the topics and works

to which our approach concerns.

The solutions based on features focus on the extraction of key points and descriptors of them. These points have to be the most representative ones of the image, since from them the skeleton is estimated, they also need to be robust about noise and variance. They could be divided on low-level, mid-level, high-level and motion features. For example, some of the low-level features are edges, corners, contours, etc; mid-level features like Fourier descriptors, shape contexts, Poisson features, etc; some of high-level features are geometry descriptors, body part patches, HOG, etc; and motion features are optical flow, motion boundaries, edge energy, etc. These features help with the description of an image or a sequence of images. Once they are extracted, an evaluation step takes place to preserve only the most valuable ones before they are fed to another process. The descriptors help to get rid of redundant information and noise. On higher levels, features provide more useful information for semantic interpretation. In recent years, it has become very popular the use of convolutional neural networks as an alternative to automate the feature extraction process (Bulat and Tzimiropoulos, 2016; Newell et al., 2016; Umer Raf and Bastian Leibe et al., 2016; Wei et al., 2016).

Further representations of the human body try to represent the inherent structure of the skeleton to lead the estimation process with more information about the skeleton constraints. Human body models help with this task; they include structural kinematic information, human body shape information and texture information. The main types of models are kinematic models, planar models, volumetric models and prior models. Figure 3.1 illustrates these models.

Kinematic models follow a skeletal structure of the human body. We can imagine these models similar to a stickman draw. They are composed with a set of body joints positions and angles. These models allow the inclusion of prior belief about the angle of each body joint. It is possible to build them in a predefined way or they

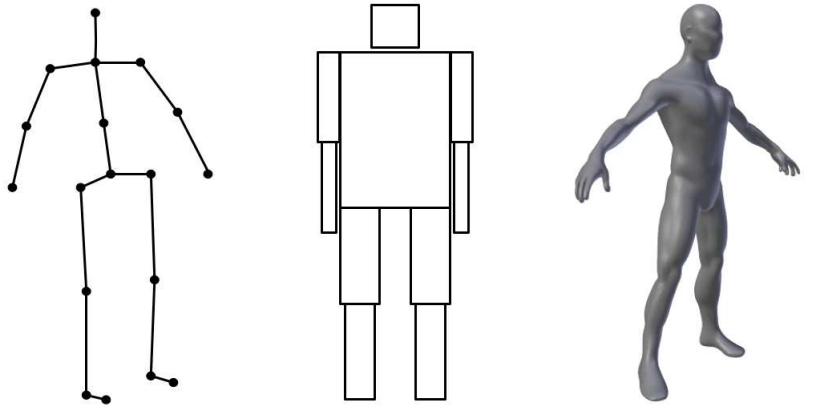

Figure 3.1: Human body models: The kinematic representation follows the skeletal structure of the human body. The cardboard model add shape and appearance to an skeletal structure. The volumetric models provide a more realistic 3D body (Gong et al., 2016).

can be learned directly from images or with additional processes (Chu et al., 2017, 2016a; Chen and Yuille, 2014; Shi et al., 2016a,b). In this work we use this kind of structures through hidden Markov models to represent a body joint trajectory and to take advantage of the temporal information.

Planar models, in addition to the properties of the kinematic models, also help to represent the shape and appearance of the body parts. For example, the cardboard model represents each body limb in a rectangle shape by the average RGB color, like the work done by (Jiang, 2010).

Volumetric models provide us with more realistic information through the inclusion of 3D body shapes and poses. They are based on geometric shapes and meshes. One option of geometric shape is a connected structure compound of cylinders representing each body limb (Hedvig Kjellström and Fernando De la Torre and Michael J. Black). The meshes are triangulated deformable models that can represent non-rigid human bodies (de Aguiar et al., 2007).

The human body pose is constrained with the kinematics, operational limits of

joints and behavioral patterns of motion in certain activities. Temporal information allows to learn pose priors from data. With the availability of motion capture techniques it is possible to explore the human pose possibilities. These priors are important to constraint the inference of pose sequences in order to improve monocular human pose tracking (Insafutdinov et al., 2017; Iqbal et al., 2017).

The methodologies are based in two ways to model the human pose estimation problem: as a geometric projection problem or as a specific image processing problem. (Gong et al., 2016) divided the related work in generative methods and discriminative methods. The main difference between each other is that while the generative methods take a 3D model of the human body and make projection in a 2D space to verify the evidence, the discriminative models take the images first and try to build a model that reconstructs the human body pose. These last models could be learned from training data; once the model is trained, the testing phase is usually faster than generative methods. Since our approach is based on taking videos as an input, and through a CNN and an HMM compute the body joint estimations and compute the most probable sequence; it is a combination of discriminative models, generative models and learning-based methods.

3.2 Human pose estimation in still images

Several contributions have been develop on the HPE problem for the configuration based on still images (Chen and Yuille, 2014; Fang et al.; Toshev and Szegedy, 2014; Khungurn and Chou, 2016; Chu et al., 2017; Wei et al., 2016; Newell et al., 2016; Yang et al., 2016). A top-down multi-person framework was developed by (Fang et al.), it consist of an initial phase of localization of bounding boxes around the target people, then it performs pose estimations within these bounding boxes. Similar to our work, this framework uses a base part-detector which is the hourglass network (Newell

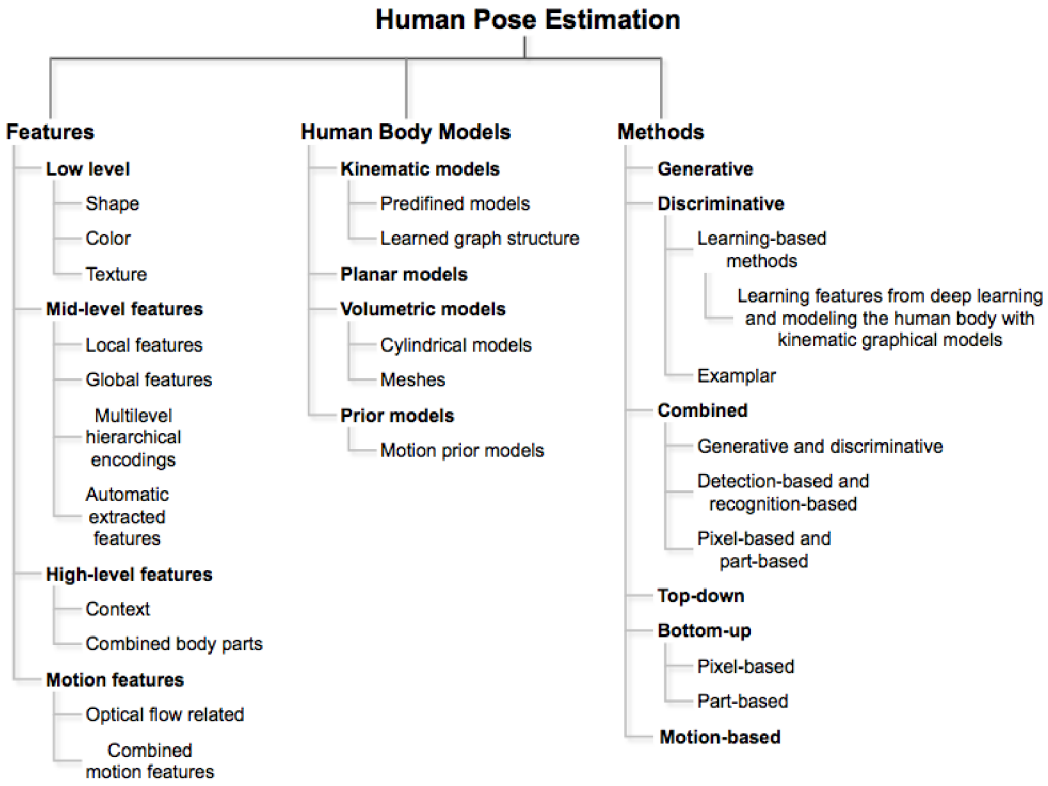

Figure 3.2: Taxonomy of the approaches followed by researchers for the human pose estimation problem (Gong et al., 2016).

et al., 2016) described in section 2.3.1. In contrast, our part-detector is an improved implementation that retrieves information from different contexts (Chu et al., 2017). Top-down architectures present failure cases on crowded scenes or where the target person is not accurately detected. They try to tackle this problem by the generation of new poses and later discrimination of the best ones. Nevertheless, the occlusion challenge within scenes remains.

A combined methodology based on CNNs and graphical models that works with still images is introduced in (Chen and Yuille, 2014). This method exploits local measurements to predict spatial relationships. The contribution is made around the idea that these local measurements could help, besides the part-detection task, to

retrieve dependencies of pairs of body joints. The spatial information is treated by a mixture model and CNNs to learn conditional probabilities focus on part-detection and spatial relations within image patches. They tested this combined method on the LSP dataset, FLIC dataset and Buffy dataset. This method relies on Non-Maximun suppresion as a post-processing phase for merging the independent detected hypotheses. However the selection of each maximum part not always leads to a correct skeletal structure (Fu et al., 2015). In contrast, our implementation seeks the most probable sequence.

DeepPose is a framework developed by (Toshev and Szegedy, 2014). As Deep Neural Networks (DNN) framework, Toshev and Szeged were the first to adapt the pose estimation problem into the neural networks field. They modeled the problem of pose estimation as a regression problem. The architectures consists of 7 layers and takes images of size 220 x 220 and regresses a coordinates vector of size $2k$, where k is the number of target keypoints. They encode the full pose into a one dimensional vector to represent the x and y coordinates. In spite of the simple architecture they use, they showed good performance on state-of-the-art datasets at that time. However this method do not consider occlusion handling. This method has been used for features extraction within other architectures.

Within the work done by (Yang et al., 2016) is proposed an end-to-end architecture based on a CNN and Graphical Model. It focuses on full body estimations to maintain a proper skeleton structure. Similar to Chu et al. (2017), they attached at the end of the CNN architecture a graph structure to model spatial constraints. However, they treat the spatial relationships differently. While in our part-detector there is a substitution of the softmax function to a CRF model, in (Yang et al., 2016) there are message passing layers to learn the dependencies between body joints. In both models the CNN is branched to focus on each body joint. In the latter method, the messages are passed between this branches accordingly to a kinematic model. The structure message passing phase is also applied in (Chu et al., 2016b).

3.3 Human pose estimation in videos

In this section we review the most relevant and recent work that focus on image sequences. Several efforts has been made to take advantage of the temporal informations in videos, this review is mainly focus on Convolutional Neural Networks based models, Graphical Models based solutions and a combination of both.

3.3.1 Graphical model based solutions

Graph optimization is a technique commonly applied to human pose estimation with temporal information. We can apply this methodology from two approaches. The first approach generates human poses as candidates on each frame and selects the best one at a time. The efficiency of the inference in this case is high, however, it is hard to get correct estimations for all parts due to a considerably large space of pose configurations; on the other hand, the second approach treats each part separately. Therefore, the candidates are body parts on each frame which are filtered to keep the consistency in the skeletal structure. Through the selection the process it takes all frames together. This last formulation improves the human pose configuration diversity. Although, the problem is NP-hard due to the loopy graph structure (Zhang and Shah, 2016).

As an alternative solution of graph optimization, which is NP-hard, the model proposed in (Zhang and Shah, 2016) is based on a tree structure model to increase efficiency and compute an exact solution. Through their tree-based model they built an approximation to the fully connected model. They introduce the concept of "abstract body part" that tries to obtain constraints between symmetric parts. Hence, the model is founded on two principal ideas, abstraction and association. The abstraction of body parts combines those which are symmetrical while the association concept consists on the generation of optimal tracklets for each abstract

body part. Through these concepts they make a simplified representation of the commonly used tree structure body in space and time. This method would fail or would have grater error due the generation of hypotheses, if the initial hypotheses have a large error it would propagate. Also it tends to fail in cases with occlusion or complex poses where the limbs are considerably separated from the torso, since they are spatially penalized.

Within a graphical model the extension of a temporal domain adds edges between the same parts in different frames. This model could have loops which turns the inference intractable. Common solutions to this problem are approximate inference, ensemble of tree-structured sub-models or changing the model structure. The required methods depends on the granularity of the models. The models that take as unit a single body part often leads to loops, thereby they apply loopy belief propagation, sampling or variational methods. The models that its granularity includes spatially the entire human pose or temporarily the trajectory of a body part, are simplified to a chain structure, which can be efficiently solved by dynamic programming. The work done by (Shi et al., 2016a) follows the latter approach based on the propagation of the body parts detected on each frame using global motion estimations. Its approach is divided in 3 modules: (i) single image pose detector on each frame, (ii) pose propagation to the whole sequence using motion estimation by elastic motion tracking, and (iii) connection of the body part tracklets by inferring the optimal pose sequence. The human body is modeled as a full body graph. This method have achieved considerable results, however, it is sensitive to the variations of body parts and it only enforce local temporal consistency between adjacent frames.

Also based on tracklets, in (Shi et al., 2016b) is introduced a spatio-temporal graphical model to select an optimal tracklet for each body part. They combine Markov networks and a Markov chain for spatial and temporal parsing, respectively. To break the loops, they combine the symmetric parts in a single node. Inference is performed separately in each model. As both models are tree-structured, the

marginal distribution for each node can be obtained efficiently by belief propagation. Inference in the Markov networks gives the number of candidate tracklets for each part which are fed in the next iteration. Then the tracklets with high temporal scores within the Markov chain are merged to generate the optimal pose. This work performs well on isolated sequence of bodies but it has limitations on crowded scenes or self-occlusion.

3.3.2 CNN based solutions

The work done by (Pfister et al., 2015) propose a CNN architecture that includes spatial fusion layers to learn spatial relationships within the target images. The predictions are refined and aligned according to the optical flow. This network takes frames around the target time t. Then at later layers of the network, confidence heatmaps are generated that gives information about each joint. The optical flow around time t helps to guide the process at later layers. The whole process is performed within the neural network. This network instead of regressing directly the x and y coordinates for each joint position, it regresses a heatmap of each joint to preserve neighbouring information. The ground truth is built with synthetic heatmaps composed of fixed Gaussians around every labeled position. The architecture of (Pfister et al., 2015) has low resolution heatmaps which is not sufficient to represent accurately a human pose. This work also lack of pose diversity and may fail on complex poses (Kawana et al., 2018).

(Wei et al., 2016) introduces Pose Machines, a framework to perform sequential predictions by exploiting spatial informations. The model is based on a CNN to learn image features and spatial dependencies. It achieves this task by preserving beliefs maps from a current time step to the next as an alternative to a graphical model. Their model is divided on two stages, in the first one the data source is only the image evidence in order to retrieve the initial belief maps, then the second stage

is composed of an architecture that considers both the image evidence and the belief maps previously computed to propagate the dependencies across the sequence. They tested the models on MPII, LSP, FLIC. Despite their good results on these datasets, the cases where multiple people are on the scene remain a challenging task. In same way as (Pfister et al., 2015) this work lack of resolution to represent a human pose, which could lead to wrong estimation in the original image context.

3.3.3 Combined solutions

CNNs in combination with a Markov Random Field (MRF) is proposed by (Tompson et al., 2014). They try to retrieve the articulated human body from monocular images. They argue that their model takes advantage of the geometric relationship between the locations of the body parts. In the same manner as (Chu et al., 2017), they use the graphical model within the neural network process to work with spatial dependencies. Hence, the CNN main function is part detection. Within the MRF the union of the parts are performed in a pairwise fashion. The unary potentials are provided by the part-detector for each body part, while the pair-wise are computed using convolutional priors, which basically represent the conditional distribution of two different body joints. This work uses a single filter to model all pairwise relations, this limitation could lead to a wrong representation of complex poses.

Another configuration for the human body pose estimation problem is the multi-person approach that consists on an extension of the problem constraints, which opens the estimations for multiple individuals, within a single image or video sequences. Some of the approaches followed in this configuration of the problem are: a non parametric representation called Part Affinity Fields (Cao et al., 2017) that associates body parts with each individual on the image by using a set of 2D vector fields that encode the location and orientation of limbs over the image domain. The work ArtTrack (Insafutdinov et al., 2017) models the set of estimations on the image

for all people as analogous to the subgraph multicut problem which is NP-hard, but recent work has shown an efficient approximate inference. The work PoseTrack (Andriluka et al., 2017) represents the problem as a graph that is composed of spatial and temporal edges and the nodes being the body joints candidates on each frame, then the association of body joints and people is solved by linear programming.

3.4 Datasets

Table 3.1 shows some of the common datasets used for evaluation in the works mentioned in previous sections. These datasets are synthetic, captured or manually labeled. These datasets are labeled, whether they support still images or video sequences (Varol et al., 2017). Within the table it is indicated the scale variation of the data source, the skeleton size variation, the amount of people and if it is a synthetic dataset.

The work introduced by Andriluka et al. (2014) is a well known benchmark within the HPE problem. The MPII Human Pose. benchmark introduces more diversity and difficulty within this research field. It is composed of over 800 human activities including recreational, occupational and householding tasks. They are also captures from different viewpoints. The Posetrack dataset described in Section 5.1 is based on an extension of this dataset.

Previous research has shown the convenience and advantages of using synthetic data in the human pose estimation problem, thereby some work done about this field are the following. The dataset SURREAL was built by (Varol et al., 2017) as a solution to the difficulty in the construction of in-the-wild datasets. They showed how based on a 3D human body model it is possible to render a variety of scenes that achieve a good performance compared to real datasets. Similar to (Chen et al., 2016) but with additional information, the SURREAL dataset includes 2D and 3D poses,

Table 3.1: Video and image datasets

Dataset	video	Multi-person	Large scale variation	Variable skeleton size	# of individuals	Synthetic
Leed Sports (Johnson and Everingham, 2010)			✓	✓	2,000	
MPII Pose (Andriluka et al., 2014)		✓			40,522	
We are familiar (Eichner and Ferrari, 2010)		✓	✓	✓	3,131	
MPII Multi-Person Pose (Pishchulin et al., 2016)		✓	✓	✓	14,161	
MS-COCO Keypoints (Lin et al., 2014)		✓	✓	✓	105,698	
J-HMDB (Jhuang et al., 2013)	✓		✓	✓	32,173	
Penn-Action (Zhang and Shah, 2016)	✓		✓		159,633	
VideoPose (Cherian et al., 2014)	✓				1,286	
Poses in-the-wild (Cherian et al., 2014)	✓				831	
YouTube Pose (Charles et al., 2016)	✓				5,000	
FYDP (Shen et al., 2014)	✓				1,680	
UYDP (Shen et al., 2014)	✓				2,000	
Multi-person PoseTrack (Iqbal et al., 2017)	✓	✓	✓	✓	16,219	
SURREAL (Varol et al., 2017)	✓	✓	✓	✓		✓
HuPBA 8k+ (Escalera et al., 2014)	✓	✓			14	

surface normals, optical flow, depth images, and body-part segmentation maps for rendered people. On the other hand, Non-parametric Bayesian Network Prior of Human Pose (Lehrmann et al., 2013) introduce a generative probabilistic model of static human pose. The model is able to synthesize realistic poses and it is able to score any given pose by how a priori likely it is. The sampling of poses allows the generation of new poses and enrich an existing dataset in combination with a rendering tool as described above. This methodology is been commonly used.

3.5 Chapter summary

In this chapter, we discuss the relevant approaches that have contributed to the improvement of the human pose estimation problem. To set a context and highlight where this thesis takes place, we describe a taxonomy and give a complete overview of the different techniques that have been employed during the past years. From low-level features extraction to higher level methodologies such as neural networks and graphical models.

Feature extraction techniques involve the use of features such as shape, color, textures, optical flow, etc. These techniques provide useful descriptions of the target image, and they also could retrieve global semantics. However, they lack the inherent human body structure information, leading the research to the inclusion of structural models such as kinematic models, planar models, volumetric models and prior models. These models help on the representation of the constraints and dependencies of the human skeleton. The information that these models could handle varies from, the influence of the body joints through their correlation, to a detailed description of the shape of a given person through tridimensional meshes. Due to the complexity of the human body, to represent its exact behavior arises high computational complexity, therefore within these models, it is common to relax the skeleton constraints, and it is also common to use approximation algorithms.

As an alternative to the feature extraction process, recent approaches use Convolutional Neural Networks (CNNs). These models have reported significant improvements. From this point, emerges solutions that combine the advantages of the different methods such as those based on CNNs and graphical models. Several approaches have implemented combined methodologies achieving great results. However due to the complexity of the human body there are still challenging problems such as: not visible body parts, occlusion, variability of poses, complex poses,

etc. This opportunity of improvements guides this work to the inclusion of a further global analysis on top of a multi-context part-detector through the computation of the most probable sequence which is intended to fill gaps where an initial estimation fails on the problematic cases.

From the state of the art we follow up ideas similar to (Shi et al., 2016b) and (Shi et al., 2016a). However we rely on a global optimization by treating complete sequences. To achieve retrieve information from low level to global movement, we use (Chu et al., 2017) as par-detector, since its approach takes into consideration analysis at from low to high resolution retrieving different semantics at each level therefore obtaining different context information which leads the idea of the inclusion of a further global optimization.

Chapter 4

CoHMM based method for human pose estimation in videos

In this chapter, we describe the proposed method for human pose estimation in video sequences. Due to the complexity of the HPE problem, it is common that an estimator misses some estimations within specific frames during a sequence. These wrong estimations could be the result of occlusion, complicated clothing, crowded scenes, etc. To enhance current implementations and datasets our model is designed as a correction module that could be attached as a post-processing phase of a complex architecture or it could be applied directly to a dataset to improve later computations. These corrections allow maintaining a correct skeleton structure along a sequence of movements.

The method takes a sequence of images from a base skeleton estimator as an input, and then it applies a set of CoHMMs to each body part to maintain temporal consistency across the complete sequence to track the human pose. The objective is to correct the skeleton structure in cases where the estimation fails.

The coupled model allows working with interacting processes efficiently by ap-

plying the N-Heads algorithm. Since the model is focused on independent joint trajectories, the graph structure of the body model is split keeping the temporal edges on each joint. This representation allows the reduction of the state space and the computational overhead.

The organization of the chapter is as follow. Sections 4.4 and 4.5 define the base human body representation that our model uses. These definitions include the base skeleton model and the base sequence structure to deal with temporal information. In Section 4.2 we give a description of the CoHMM parameters. Section 4.3 describes the Viterbi algorithm for coupled processes. Section 4.6 describes the reconstruction process of the human skeleton from the most probable sequences previously computed.

4.1 CoHMM for human pose estimation

This method is intended to keep the skeleton structure across time. We work with complete sequences obtained from a base part-detector, which not always gives correct estimations. So we need to keep track of the body movement in order to recover a certain pose from the incorrect body joint estimations.

The part-detector is taken from the Multi-Context work (Chu et al., 2017). They introduced an approach which takes a set of images and outputs a set of estimated coordinates, one for each body joint. Multi-Context was designed as a single-person and single-frame estimator. We modified the test module of the framework to allow multiple sets of frames and to generate the base sequences. We retrieved from the Multi-Context work the estimations computed by feeding the estimator with the PoseTrack dataset. This dataset is labeled for different kind of pose configurations. The most general case is multi-person, although, our work is constrained to single-person cases. It is a challenging test due to the several scenes problems involved in

crowded scenes, such as occlusion.

We adjust a Coupled Hidden Markov Model to work with the skeleton movement, the approach we follow is to attach a single model to each body joint. To avoid the exponential growth of complexity, we assume independence between joints. We merge the initial sequence and our estimation at the final stage to preserve spatial and temporal consistency through the Equation 4.6.

We propose to maintain the structural body consistency through time by exploiting the computation of the most probable sequence of movements. To perform this task, the first problem that arises is the representation of the states. Since we are working with spatial information (body joint positions), the representation of coordinates in one variable though one HMM leads to a high state space. We propose to keep each variable separate (X, Y) and optimize their movement through their coupling as interacting processes.

The computation of the most probable sequence for coupling processes is done through the N-Heads algorithm, which is an extension of the Viterbi algorithm. To maintain efficiency, we define the interacting processes as the Cartesian axis, x, and y. To reduce the state space, instead of using directly the coordinates, we perform the modifications explained in Section 4.4 and Section 4.5. Similar to (Shi et al., 2016b) we break up the spatial and the temporal models by treating the temporal process through the CoHMMs, but we compute the most probable interaction between the axis on each body part sequence. After running the N-Heads algorithm, a reconstruction phase takes place. We merge the original sequence with the most probable sequence for each process since the original sequence gives key spatial information.

The complete process is divided in the following stages: i) trajectories representation iii) most probable sequence by CoHHMM, and iii) body reconstruction. The

complete architecture is illustrated in the Figure 4.1.

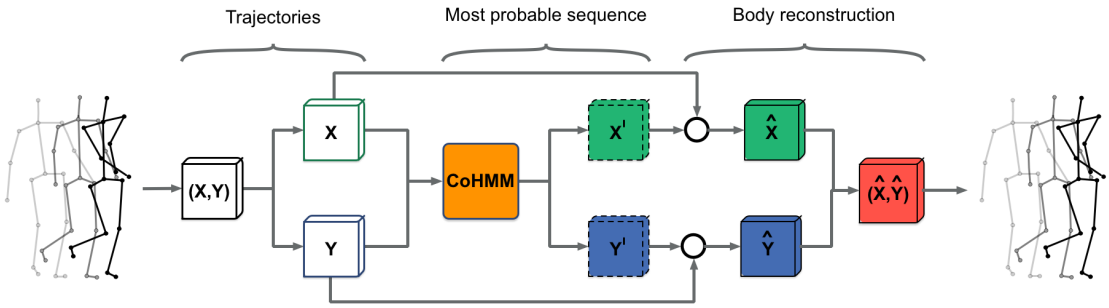

Figure 4.1: Main architecture. i) the representation of a sequence is given by the transformation of the initial sequences of tuples (X, Y) (extracted from the part-detector) to a set of normalized relative position sequences for each axis X and Y. ii) through a CoHMM, X' and Y' represents the most probable sequence of states given the X and Y observations. ii) the initial sequences and the most probable ones help to rebuild the body skeleton.

4.2 CoHMM parameter learning

At this stage, we aim to find the parameters for each joint to feed the associated CoHMMs for the X and Y processes for each joint. The parameters of the model are the following: i) the set of states, ii) the set of observations, iii) the prior probabilities, iv) the transition probabilities, and v) the observation probabilities. We will describe in more detail the computation of each set of parameters.

4.2.1 Observations

The dimensions of the input images are 256 x 256, and the output is a set of heatmaps of dimensions P x 64 x 64, where P is the number of joints. From these heatmaps, we extract the maximum values for each joint. The part-detector (Chu et al., 2017) also requires additional parameters such as the center of every target person and a scale factor that takes as reference 200px to calculate its bounding box. Since this

approach was designed for a single-person and a single-frame we made a minor modification to support sequences of frames. This modification organizes by sequence all the initial parameters and outputs to ensure consistency with the PoseTrack dataset.

The estimated skeletons, therefore, are defined by a structure of dimensions N_f x $2P$ for each sequence. Where N_f is the number of frames, P is the number of body joints and the second dimension contains the predicted coordinates (x, y). The data is transformed according to our base representation, so we end up with a data structure of the same size but with relative, normalized and centered trajectories.

These modifications give us the base observations of our model. The main idea of this representation is to isolate the movement behavior for a joint around its father. It focuses the attention on a proper joint movement while the part detector gives us information about spatial displacements.

Instead of a matrix of probabilities, we include a continuous emission function to obtain the probability $b_{ik} = P(O_t = o_k|S_t = q_i)$. The emission function is defined with the following expression:

$$P(O_t = o_k|S_t = q_i) = \frac{1}{\sqrt{(2\pi)^k det\Sigma_{o,s_t}}} exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu}_{o,s_t})^T \Sigma_{o,s_t}^{-1}(\vec{x} - \vec{\mu}_{o,s_t})\right)$$

where $\vec{x}$ is the vector that contains x and y observations. $\vec{\mu}_{o,s_t}$ is the mean vector of $\vec{x}$ values that are labeled with the state S_t in the training set. In the same way, Σ_{o,s_t} is the covariance matrix of those coordinates.

For each subset of data the covariance matrix Σ is computed from the following equation:

$$\Sigma = 1/(|O| - 1) \sum_{i=1}^{n} (\vec{x}_i - \mu)(\vec{x}_i - \vec{\mu})^T$$

For simplicity we omit the notations $\cdot_{o,s_t}$ which indicates the subset of observations o that are labeled with the state s_t.

4.2.2 States

To define the states of the CoHMM we represent the sequences in a restricted state space to avoid high computational complexity. The set of states $Q = \{q_1, q_2, ..., q_n\}$ is defined from a uniform spaced range of bins while the state labeling is performed by a discretization process guided by those bins. We end up with two trajectories, one from the part-detection process and another discrete sequence that leads the training process. We assign a state label to the closest bin i through the expression $bin_{i-1} < k \leqslant bin_i$ where k is a value from X or Y process and $i = 1...N$, the number of states as N. The number of bins is passed as a parameter.

4.2.3 CoHMM Transition matrix

The transition matrix for both processes X and Y is calculated directly from the state sequences of the training set. In CoHMMs, X and Y, as interactive processes, have conditional probabilities which make the joint state set a cartesian product with transitions of dimension NM x NM. To tackle this increase of complexity, we follow the approach described in (Brand et al., 1997), that explains a factorization to compute the conditional probabilities and the transition probabilities. Let x_i and y_j, for $i = 1...N$ and $j = 1...M$, be the states for each process. Then the cartesian product leads us to the a quadratic state table with joint states $C_{ij} = \{x_i, y_j\}$.

We obtain the transition parameters $P_{x_i|x_j}, P_{y_k|y_l}$ and the conditional parameters

$P_{x_i|y_l}, P_{y_k|x_j}$ through the following expressions:

$$P_{x_i|x_j} = \sum_l P_{y_l} \sum_k P_{c_{ik}|c_{jl}} \tag{4.1}$$

$$P_{y_k|y_l} = \sum_j P_{x_j} \sum_i P_{c_{ik}|c_{jl}} \tag{4.2}$$

$$P_{x_i|y_l} = \sum_j P_{x_j} \sum_k P_{c_{ik}|c_{jl}} \tag{4.3}$$

$$P_{y_k|x_j} = \sum_l P_{y_l} \sum_i P_{c_{ik}|c_{jl}} \tag{4.4}$$

where in the absence of posterior probabilities $P_{x_l} = \frac{1}{|\{X\}|}$ and $P_{y_j} = \frac{1}{|\{Y\}|}$. This projection factors the transition table dimensions $(|\{X\}| \cdot |\{Y\}|)^2$ into two transition tables of dimensions $(|\{X\}|)^2$ and $(|\{Y\}|)^2$ and two conditional tables of dimensions $(|\{X\}| \cdot |\{Y\}|)$. The transition tables denote the change of state within the same process while the conditional tables denote the interaction between the two processes.

4.3 Viterbi algorithm

The Viterbi algorithm give us the most probable sequence of states given a sequence of observations. The problem is modeled with the states being discrete trajectories over time while the observations are the part-detector estimated trajectories. Therefore, this algorithm finds the best movement behavior given the base estimations. This method has the goal to preserve consistency between frames by taking advantage of the behavior of a joint around its father.

The Viterbi algorithm for CoHMMs has more parameters (conditional probabilities) than a standard HMM. The state space and the complexity of Viterbi is bounded by $O(TN^{2C})$ for a naive approach. The approximation algorithm N-Heads, imple-

mented using dynamic programming, reduces the complexity of both the forward-backward and Viterbi algorithm to a bound of $O(T(CN)^2)$.

The N-Heads algorithm for the coupled model performs the maximization of Equation 2.8. In this case, it maximizes both chains X and Y in the same process. As we described earlier, the parameters of this equation are obtained from the observations of each chain, and the factorization of the transition matrix reduces the complexity of the algorithm. In this problem there are N states for process X and N states for process Y, thereby the complexity of the algorithm is $O(T(N)^2)$, which improves the exponential growth in the naive cartesian product approach. The same state space size for both processes helps with the efficiency of the operations.

4.4 Human body representation

The Multi-Context Attention Model approach works with ankles, knees, hips, pelvis, thorax, wrists, elbows, shoulder, neck and head. While the PoseTrack dataset works with ankles, knees, hips, wrists, elbows, shoulders, neck, nose and head. To maintain compatibility with both structural body models, we get rid of the thorax and pelvis from the Multi-Context work, and also we get rid of the nose from the PoseTrack dataset. These models are illustrated in Figure 4.2. So the target structure is defined as the graph $G = (V, E)$ where V are the matched joints in both models and E the links as it is shown in Figure 4.2 C.

This representation is split to focus on each body joint at a time. The separation is composed of tuples of child and parent. The relative positions help to reduce the spatial area in which the model focus. Since the target window requires fewer states to represent the articulation movement, it reduces the computational overhead. Besides, the model is intended to learn a proper articulation behavior around its parent, which will lead the correction process on cases where the part-detector

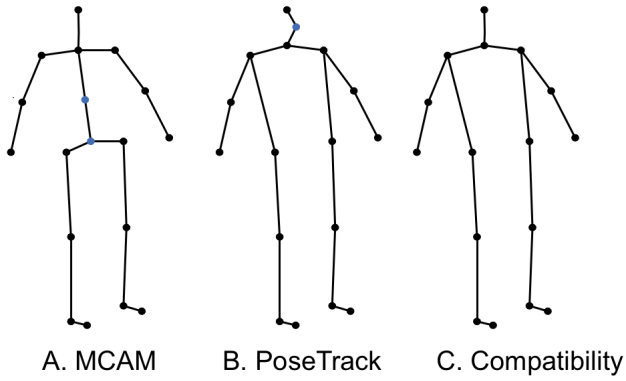

Figure 4.2: Skeleton models. The part-detector and the datasets we use have a different skeleton structure. To make a structure compatible with both approaches we are focus on the body joint that are present in both models, as it is illustrated in C.

fails. Let the set of all children be the base representation of the skeleton, X be a random variable associated to the x coordinates and Y be a random variable associated to y coordinates of every joint. Thereby, the coordinates of a joint i are defined as $X_i = x_i - x_j$ and $Y_i = y_i - y_j$ where j is the parent joint according to the edges E. Then $V = \{(X_0, Y_0), (X_1, Y_1), ..., (X_P, Y_P)\}$ where P is the numbers of joints. This computation is done from leave nodes to the root node which is the head. Figure 4.3 shows the relative positions sets.

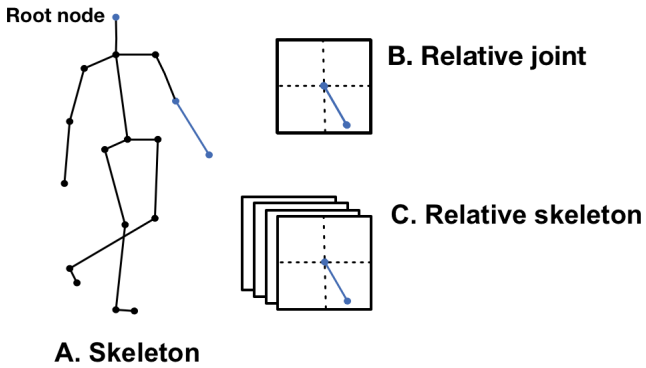

Figure 4.3: Relative skeleton representation. The base human body representation is modeled from the relative position of each joint. This representation helps to highlight the proper behavior of a human body joint and also to reduce the state space for the CoHMM. So a set C, composed of all the body joints B of the skeleton A, is the base representation of a human body in this work

4.5 Human body sequence representation

A skeleton sequence is represented by a set of relative positions across time. Let H be the graph that connects each joint in the skeletons across time within T frames with their trajectory. Since we assume spatial independence between joints, we delete the spatial edges, and we add temporal ones connecting each joint over time. Hence, the graph $H = (V_i, E_i)$ for $i = 1...T$ turns into joint trajectories.

The visualization of the coordinates within a sequence is shown in Figure 4.4. The kinematic body model is transformed into relative coordinates by taking the parent of each joint as a reference. The resulting set of positions are considered as the new body representation across time. As we mentioned, the set of the children joints relative positions is the representation of a skeleton, then the representation of a sequence would be a set of skeletons for every time t. Since we are interested in the articulation trajectories, this last set is grouped by joint. So the skeleton sequence in Figure 4.4 A is represented through the set of every body joint trajectory over time as it is illustrated in Figure 4.4 D.

To properly represent the articulations trajectories within a reduced state space, each joint sequence is translated and centered on the mean value of each axis x and y. The mean value is computed separately for each axis taking into consideration all frames. Then every trajectory is normalized to keep them all within a target bounding box. This transformation helps to avoid the positional bias due to the skeleton spatial structure and to emphasize the body part behavior. This representation also allows the model to work within a given resolution, which is useful for the generation of the state. Figure 4.5 illustrates this representation.

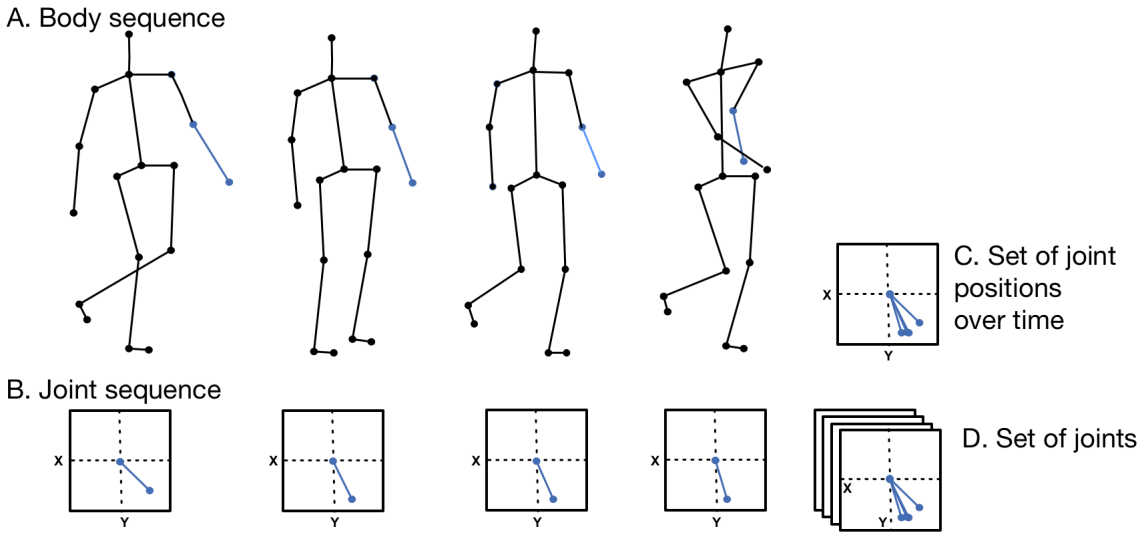

Figure 4.4: Human body sequence representation. A) Target body joint highlighted to illustrate the joint movement. B) Relative position of the joint over time. C) set of relative positions of a selected joint. D) The complete sequence is represented by a set of joint sequences

.

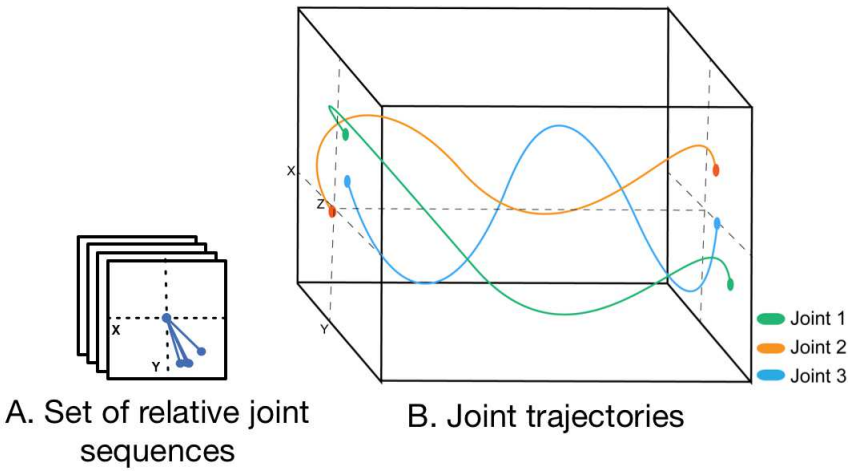

Figure 4.5: Set of body joint trajectories normalized on a bounding box. The set of the articulations relative coordinates in A is represented as a spatial trajectory. All the trajectories are translated from its skeleton base position to be centered on the mean value of axes x and y during the whole sequence. This translation helps in the representation of the spatial states.

4.6 Sequence correction

The sequence correction is performed through the reconstruction of the skeleton structure since we split the sequences to work with each joint independently. The rebuilding is divided in the three following steps: i) trajectories combination, ii) transfer of trajectories and iii) back to absolute coordinates.

The trajectories combination helps to reduce the error from both sequences, the initial sequence, and the estimated sequence. It is performed with the following expressions:

$$\hat{X} = \alpha(kX') + (1-\alpha)X,$$
$$\hat{Y} = \alpha(kY') + (1-\alpha)Y$$

where the combination is weighted by $\alpha = [0, 1]$ and k is a constant to adjust the area of movement. X' and X denote the estimated sequence and the initial sequence respectively, in the same way for Y' and Y.

This process allows us to control how strong would be the modifications on the estimated skeletons since we want to make a robust estimation by the combination of both trajectories. The output of this phase is a set of corrected trajectories $(\hat{X}, \hat{Y})$.

To reconstruct the skeleton, we use the mean of each observation trajectory as a reference. Hence, we move the corrected sequence to match its mean with the mean of the observations through the following equations:

$$\hat{X} = \hat{X} - \mu(\hat{X}) + \mu(X),$$
$$\hat{Y} = \hat{X} - \mu(\hat{Y}) + \mu(Y)$$

where $\mu(X)$ and $\mu(Y)$ are the mean values for the initial trajectories. $\mu(\hat{X})$ and

$\mu(\hat{Y})$ are the mean values of the estimated trajectories. Therefore, this equation could be seen as a translation of the estimated sequences $\hat{X}$ and $\hat{Y}$ to the origin, then to the mean of the initial estimations. This translation allows us to preserve the spatial structure of the human body while the most probable sequence gives the temporal consistency.

Once the trajectories are placed, the last step is to return from relative position to absolute ones. We propagate the relative positions following the tree structure of Figure 3.1 C from the root to the leaves.

4.7 Chapter summary

The objective of human pose tracking is to maintain consistency in the body joints trajectories. In this chapter, we present a method based on CoHHMs to deal with temporal information. Within the scope of this work, a set of CoHHMs is attached independently on each body joint to compute its most probable sequence. Our main contribution is the implementation of a post-processing method combined with a part-detector to take advantage of spatio-temporal dependencies. This work could enhance current implementations and datasets by maintaining a correct skeleton structure along a sequence of movements.

The part-detector is based on an architecture built upon a CNN and a CRF. The based model as part-detector is Multi-Context Attention for Human Pose Estimation (Chu et al., 2017), which uses a CRF to deal with spatial correlations between neighboring regions.

The kinematic models of the human body are split to keep each articulation independent as a representation of a skeleton and limbs trajectories. Then the movement of each limb is translated relative to its parent. This transformation helps to high-

light the behavior of a child around its parent and to reduce the resolution required to represent the body joint's movement.

The parameters of the CoHMMs are obtained directly from the dataset. The states represent a sequence within a discrete space. The emission function is composed of a Gaussian mixture model, where each Gaussian parameter is calculated from the observations labeled as a given state. The transition matrixes are computed from a factorization of the coupled transition matrix (Brand et al., 1997).

We describe the N-Heads algorithm which through the parameters of the CoHMM model computes the most probable sequence of coupled states from a given sequence of observations. We model the coupled states as the axes X and Y from the coordinates of the body joints trajectories.

Finally, we rebuild the body structure by placing the most probable sequence of each joint on the mean of each initial sequence. The initial sequence is taken as a reference to maintain the target body motion.

Chapter 5

Experiments and results

This chapter is focused on the experiments and their evaluation. To address the experiments we chose the challenging dataset PoseTrack (Andriluka et al., 2017) which has become a benchmark for video-based human pose estimation and articulated tracking in recent years. This dataset allow us to compare our work to recent state of the art implementations.

The organization of the chapter is as follows: in Section 5.1, we describe the PoseTrack dataset. Section 5.2 gives the details about the experiments configurations. Section 5.3 discusses the findings from the experiments and compares the results of the proposed method and the base part-detector.

5.1 PoseTrack dataset

The PoseTrack dataset (Andriluka et al., 2017) is a benchmark proposed for the human pose and articulated tracking problem through videos. This work is focused on three main challenges: i) single-frame multi-person pose estimation, ii) multi-person

detection in videos, iii) multi-person articulated tracking. The video sequences are composed of 5 seconds of activity with 30 labeled frames. The test annotations are dense with annotations every four frames. The whole dataset contains 23,000 labeled frames with 153,615 pose annotations.

The annotations contain an id for every person in the scene besides the coordinates for all joints. They include a bounding box for the head for every person. The joints included in the labels are the following: head, nose, neck, shoulders, elbows, wrists, hips, knees and ankles.

We take advantage of the multi-person approach of the dataset to split the annotation for every person as a new sequence. Hence, there are more observations per joint, and as we work with the sequences by joint, this is useful. Also, in the case where not all the skeleton is visible, we work only with those valid body joints.

Figure 5.1 illustrates some image examples extracted from the sequences of frames. This dataset includes a variety of activities such as sports, jobs, dances, daily life tasks, etc. It also includes a variety of scenarios within indoor and outdoor places. The complex poses arise occlusions, incomplete bodies, and cluttered bodies.

5.2 Experiments

To perform the experiments with the CoHMM implementation, we use the Posetrack dataset split for training and test. The test dataset contains 380 sequences. The only parameter we need to set at the beginning of the process is the number of states. To speed up the algorithm implementation this parameter is the same for both processes, and because we work on a spatial window, our method gives the same resolution on both axis. We tested our model with three different number of states: 21, 41 and 51 states.

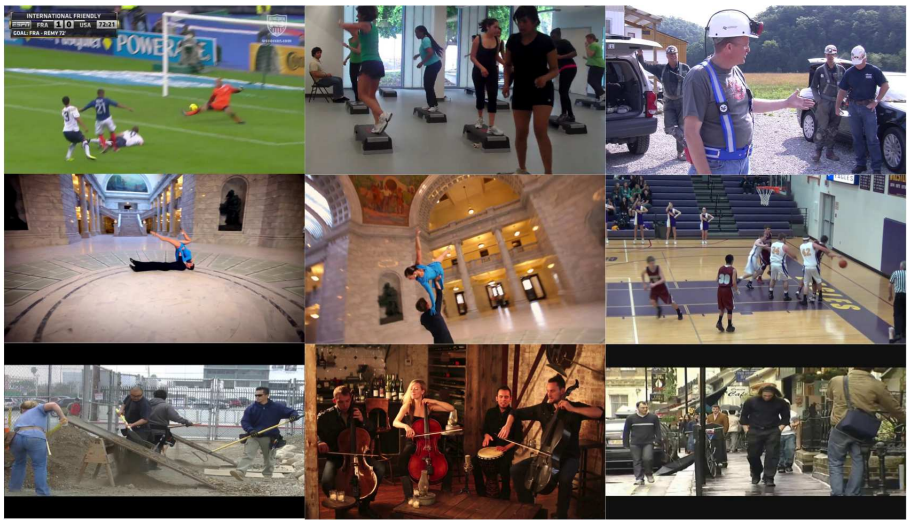

Figure 5.1: PoseTrack image examples. The PoseTrack dataset is composed of a wide range of complex scenarios and activities. The images are taken from real-life environments and activities such as sports, workplaces, dance, daily life tasks, etc. Since the dataset is focused on different configurations of the pose estimation problem, from this images, it is possible to work with single person estimations and multi-person estimations.

The Multi-Context implementation is composed of an 8-stack hourglass network. The input image dimensions are 256 x 256, and the output heatmaps are P x 64 x 64, where P is the number of body parts. From these heatmaps, the maximum points are the initial body joint's coordinates.

As an evaluation metric, we use the mean Euclidean distance between the ground truth, the part-detector points, and the post-processed part-detector points. The results present the mean and the standard deviation as a comparison of the performance of both models.

5.3 Discussion and results

In Figures 5.2, 5.3, and 5.4 we show the mean error computed on each body joint from the experiments with 21, 41 and 51 states, respectively. The graph shows

the comparison of this metric between both models, the base part-detector and the CoHMM attached as post-processing. It is possible to see that the mean error from the model remains closer to the error of the part-detector. However, the standard deviation decreases. The decrement of the standard deviation is due to the correction of outliers, which could be interpreted as missed body joints caused by occlusion, cluttered bodies or incomplete bodies. To support this argument, the error distribution in Figure 5.5 illustrates how its distribution has moved, and the number of outliers decreased. For better visualization the fit error distribution graph in Figure 5.6 presents more details. Therefore, the set of CoHMMs include the missed points in a proper body joint behavior.

In Figure 5.7 we report two examples of sequences where our model makes a skeleton sequence correction. Within the figure, (A) and (D) are the input images labeled by the part-detector, we took only a sample of a sequence for illustration purposes. (C) and (E) isolate the labeled skeleton for a better comparison with our model. Hence, (B) and (F) are the skeletons computed from our models. From these examples, we can say that our model preserves the skeleton structure in those cases where the part-detector fails. Since our model also uses the part-detector estimation, the best results are on the cases in which the base estimation is not completely lost. A limitation of our model is the inclusion of an additional error on some estimations that already were correct. This behavior is illustrated in the fit error distribution graph in Figure 5.6 and in Table 5.1. Another consideration to take into account for this behavior is that we are based on the base estimation to reconstruct the body since in this approach is our only reference. We compute the Average Precision (AP) metric (Andriluka et al., 2017) to compare our implementation with other works that use the Posetrack dataset. The results are shown in Table 5.1. Despite there is a decrease in the AP with the inclusion of the CoHMM, the results show that our model helps to reduce the standard deviation, which is relevant for some applications. The published scores of the challenge do not provide the reference for

all works.

In Table 5.2 we show the mean and the standard deviation of the execution time for the key phases of our implementation during 10 iterations of the experiments. These times show the efficiency of the N-Heads algorithm.

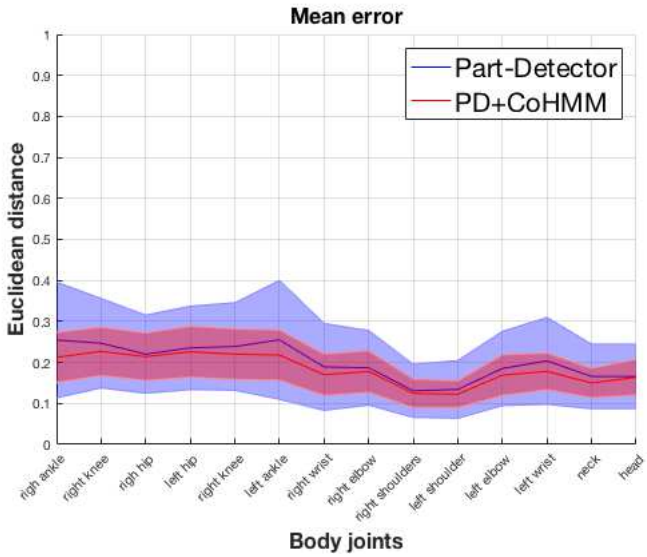

Figure 5.2: 21 states mean error comparison. Comparison between the error computed from the part-detector model and the part-detector with the inclusion of our model. The error is computed through the Euclidean distance from the relative trajectories and the ground truth.

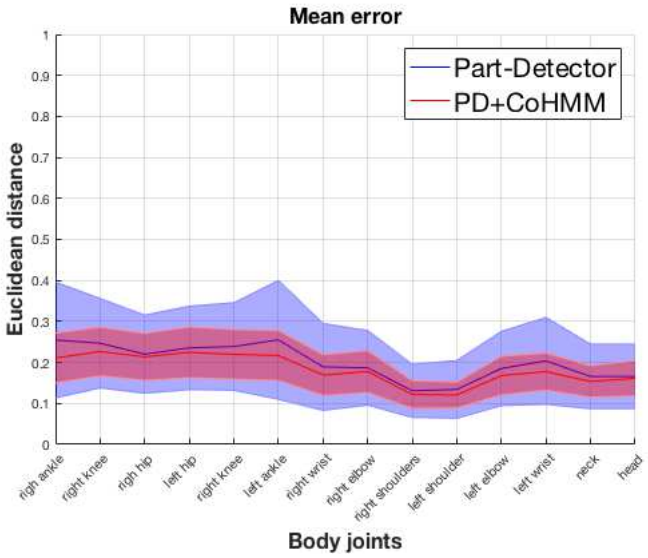

Figure 5.3: 31 states model mean error comparison. Comparison between the error computed from the part-detector model and the part-detector with the inclusion of our model. The error is computed through the Euclidean distance from the relative trajectories and the ground truth.

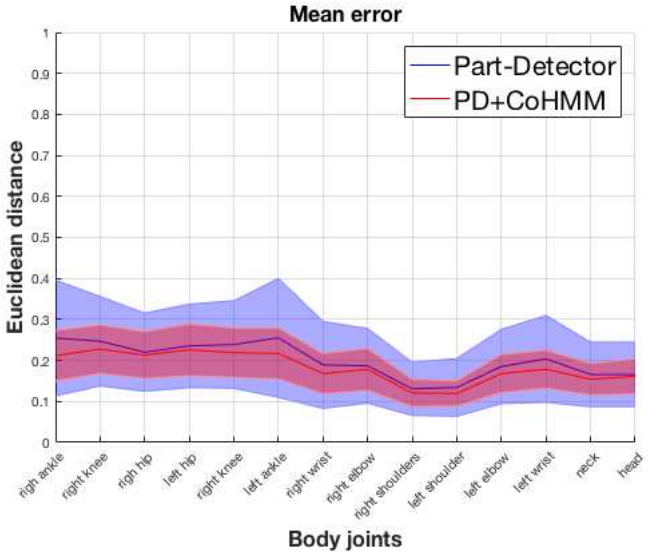

Figure 5.4: 51 states model mean error comparison. Comparison between the error computed from the part-detector model and the part-detector with the inclusion of our model. The error is computed through the Euclidean distance from the relative trajectories and the ground truth.

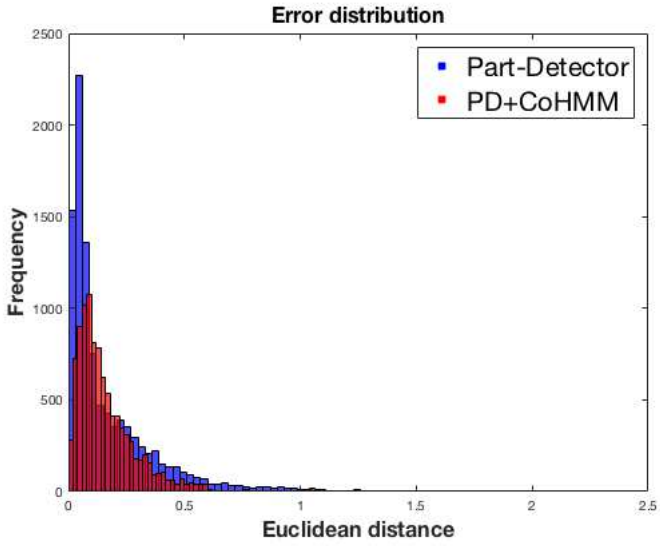

Figure 5.5: 41 states model error distribution. The distribution of the error shows the reduction of the outliers

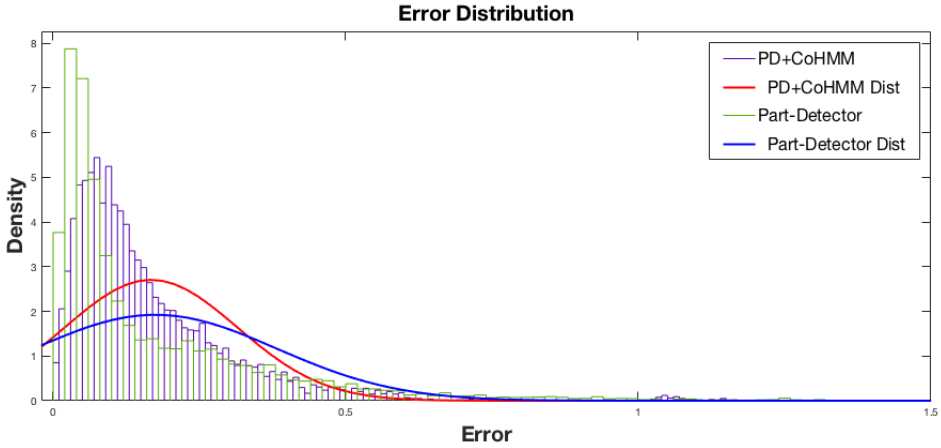

Figure 5.6: 41 states model fit error distribution. The distribution of the error shows the reduction of the outliers

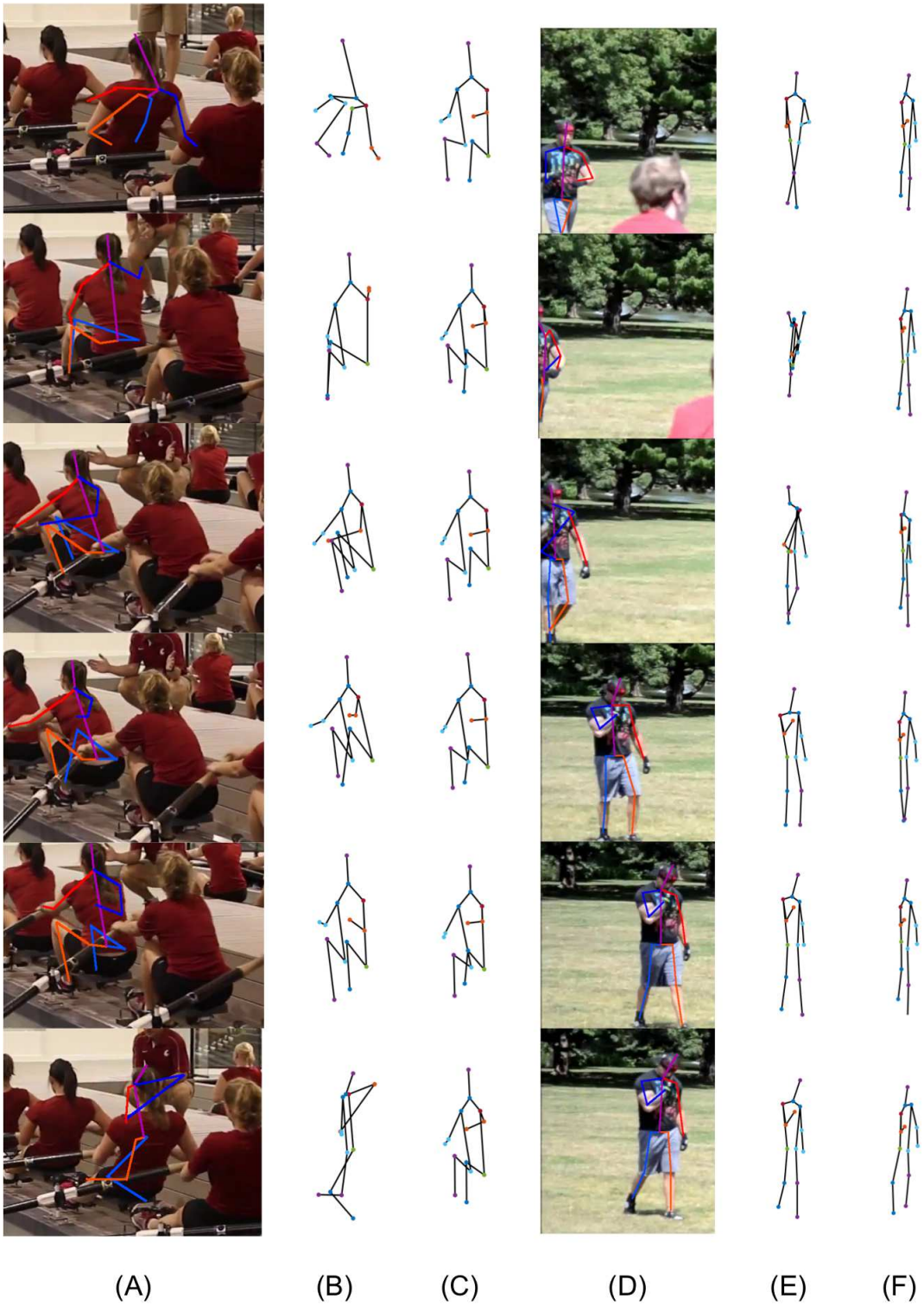

Figure 5.7: Examples of estimated poses. (A) and (D) are the source images annotated by the part-detector. (B) and (E) are the extracted skeletons to better visual comparison with our poses. (C) and (F) are our estimated poses.

Table 5.1: Posetrack 2017 Challenge: AP results of the pose estimation challenge for the multi-frame approach. Despite there is a decrease in the AP with the inclusion of the CoHMM, the results show that our model helps to reduce the standard deviation, which is relevant for some applications. The published results do not provide a reference for all works (Andriluka et al., 2017).

Work Name	Wrist AP	Ankle AP	Total AP
T-Net	72.04	66.96	74.95
FlowTrack	71.52	65.69	74.57
STAF	65.02	60.72	70.28
HMPT	60.99	60.11	63.73
MVIG	59.37	58.13	63.23
PoseFlow	59.03	57.90	62.95
BUTD2	52.92	42.65	59.16
MPR	52.29	49.47	57.55
IC_IBUG	35.21	32.59	47.56
Multi-Context Attention Model (Part-Detector) (Chu et al., 2017)	67.23	63.23	71.32
Part-Detector + CoHMM	61.50	54.75	62.88

Table 5.2: Execution times: in this table are listed the execution times for the relevant phases of our work after 10 iterations of the experiments.

Process	Mean seconds	STD
Data formatting	30.63	4.04
CoHMM parameters learning	51.48	5.91
N-Heads	0.0069	0.0042
Reconstruction	$1.69\text{x}10^{-6}$	$2.33\text{x}10^{-6}$

Chapter 6

Conclusions and future work

6.1 Summary

This thesis proposed a method for markerless 2D human pose tracking on monocular video sequences. The monocular and markerless constraints have received increasing attention because of the low-cost setup and the availability of data. The convenience of relying on this approach increases the range of real-life applications such as video surveillance, human action recognition, activity analysis, gesture recognition, etc.

Although many approaches have been followed to solve this problem, it is still a challenging task. The high complexity is due to the large number of degrees of freedom, the interdependency between the body joints, the non-rigidity of the human body; also it is due to external factors such as different types of body shapes on each person, resolution of the acquired images, different shapes of clothes, etc. Besides, the scene composition could lead to occlusion, cluttered bodies, and moving backgrounds.

The proposed method, described in chapter 4, treats the pose estimation problem

through a spatio-temporal approach. The initial skeleton estimations are extracted from Multi-Context Model for Human Pose Estimation implementation, which is composed of a combination of a CNN and a CRF. Then this data is divided by joint, and it is fed to a set of CoHMMs to process each joint separately. This division allows increasing the computational efficiency since this method keeps dependencies only between axes of each joint. As a final step, the initial estimation serves as a reference to rebuilding the body skeleton.

The CoHMM method described in Section 4.1 focus on a post-processing phase attached to a part-detector. Our main contribution consists on the inclusion of temporal analysis to maintain the skeleton structure over a sequence of movements. The parameter learning process is described in Section 4.2.

To avoid high computational complexity, we used the sequence representation described in Sections 4.4 and 4.5. This representation leads to a reduced state space. The complexity also decreased with the use of the N-Head approximation algorithm of Section 4.3.

The N-Heads algorithm generates the most probable sequence of interacting processes. The chosen processes that represent the body joint behavior are the axes X and Y. Using the sequences of each joint the method performed a skeleton correction, it is described in Section 4.6. The correction merged the initial estimation with the computed sequence to rebuild the body skeleton and to reduce the error in both sequences.

The evaluation of the proposed method, described in the Experiments Chapter 5, was performed on the PoseTrack dataset with 380 test sequences. The results show that in such cases in which the part-detector fails to keep the body structure between frames properly our models helps to fill in these gaps.

6.2 Conclusions

This work tackles the human pose estimation though different perspectives. Firstly from the state of the art we took advantage of a model that retrieves different semantic meaning by the inclusion of neighboring regions and multiple resolutions. This approach follows our approach by the consideration of further available information within the constraints of the input data. As a spatial approach, to perform the body joints estimations, they retrieve spatial contextual information within the scenarios to make a coordinate prediction.

Our contribution took the temporal information as a help to improve the estimations, due to great results within the state of the art implementations of the HMMs for sequential data we applied this model to preserve global consistency of the body structure. Following the thoughts that the body has a complex configuration but at the end, it relies on a structure that always preserves and therefore the estimation of each joint through time may follow a path that could be modeled, considering cases of common life activities.

The human body has a huge space of possible poses, and with the addition of time, the inference of the sequences turns intractable. For this reason, we show that the N-Heads algorithm could help to maintain efficiency within a reasonable running time by a relaxation of the paths that must be followed, within the Viterbi trellis, to compute the most probable sequence. Also to help in the reduction of the computational overhead our implementation treated the sequences as relative to its parent to allow a representation within a part size resolution instead of a complete scene.

In the final step we took advantage of the work on the spatial information within the part-detector and our implementation with the temporal information therefore we proposed a simple merge step two preserve the contribution of both works.

We have shown that the inclusion of temporal analysis by the computation of the most probable sequence helps to maintain the consistency of the human body structure. Challenging cases such as occlusion or cluttered bodies could be corrected by considering complete sequences. Since each body joint is constrained with the parent it is attached, the proposed method is intended to learn the dependencies that this parenting structure has on the child movement behavior. For instance, this refinement could be applied to the generation of datasets or as a post-processing phase for more complex architectures.

We could summarize the following statements as learnings from this work:

- Cases with occlusion or cluttered bodies could be corrected considering complete sequences.
- Temporal information fills gaps where the initial estimates are missed.
- The N-Heads algorithm helps to avoid high complexity, leading to an efficient post-processing phase.
- The proposed model could be treated as a portable module useful for body structure corrections.

6.3 Limitations

Our implementation relies on the base estimation to compute the most probable sequence and to reconstruct the body structure at the end of the process for each body joints. Therefore it has a strong dependency on the quality of the initial estimations, for both training and testing stages. We transformed every detected joint coordinate of the part-detector trough or architecture to maintain a body structure. For this

reason, the results present decay in the error computation, since the predictions that were already correct are modified.

We design this architecture as a module that could be attached as a post-processing phase or a pre-processing phase and it is not intended to replace other pose estimation techniques but to enhance those in which it is important to preserve the human body structure.

Additionally, to maintain efficiency, we assume independence between body joints in the computation of the most probable sequence.

6.4 Future work

According to the previously described findings and limitations, the following improvement could be followed. The treatment of the body structure as a set of independent articulations has a significant advantage on the efficiency of the inference procedures; however, to rebuild the skeleton, the method uses the initial estimation as a reference. Further research work could include dependencies between joints to maintain the complete structure at the same process and therefore get rid of the part-detector at the end of the process which could stop the error propagation. These dependencies could be represented by adding more than one joints within the coupling in the HMMs. One case where this change could impact is such one where the initial sequence has several wrong estimated frames.

References

Andriluka, M., Iqbal, U., Milan, A., Insafutdinov, E., Pishchulin, L., Gall, J., and Schiele, B. (2017). PoseTrack: A Benchmark for Human Pose Estimation and Tracking. *arXiv*.

Andriluka, M., Pishchulin, L., Gehler, P. V., and Schiele, B. (2014). 2d human pose estimation: New benchmark and state of the art analysis. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693.

Brand, M., Oliver, N., and Pentland, A. (1997). Coupled hidden Markov models for complex action recognition. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Bulat, A. and Tzimiropoulos, G. (2016). Human pose estimation via Convolutional Part Heatmap Regression. In *ECCV*.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1302–1310.

Charles, J., Pfister, T., Magee, D. R., Hogg, D. C., and Zisserman, A. (2016). Personalizing Human Video Pose Estimation. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3063–3072.

Chen, W., Wang, H., Li, Y., Su, H., Wang, Z., Tu, C., Lischinski, D., Cohen-Or, D., and Chen, B. (2016). Synthesizing training images for boosting human 3D pose

estimation. In *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*, pages 479–488.

Chen, X. and Yuille, A. L. (2014). Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*.

Cherian, A., Mairal, J., Alahari, K., and Schmid, C. (2014). Mixing body-part sequences for human pose estimation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2361–2368.

Chu, X., Ouyang, W., Li, H., and Wang, X. (2016a). CRF-CNN: Modeling Structured Information in Human Pose Estimation. *Conference on Neural Information Processing Systems (NIPS)*, (Nips):316–324.

Chu, X., Ouyang, W., Li, H., and Wang, X. (2016b). Structured Feature Learning for Pose Estimation. *Cvpr2016*, pages 4715–4723.

Chu, X., Yang, W., Ouyang, W., Ma, C., Yuille, A. L., and Wang, X. (2017). Multi-Context Attention for Human Pose Estimation.

de Aguiar, E., Theobalt, C., Stoll, C., and Seidel, H.-P. (2007). Marker-less deformable mesh tracking for human shape and motion capture. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Eichner, M. and Ferrari, V. (2010). We are family: Joint pose estimation of multiple persons. In *ECCV*.

Escalera, S., Baró, X., Gonzàlez, J., Bautista, M. Á., Madadi, M., Reyes, M., Ponce-López, V., Escalante, H. J., Shotton, J., and Guyon, I. (2014). Chalearn looking at people challenge 2014: Dataset and results. In *ECCV Workshops*.

Fang, H.-s., Xie, S., Tai, Y.-w., and Lu, C. RMPE : Regional Multi-Person Pose Estimation.

Fu, L., Zhang, J., and Huang, K. (2015). Mirrored non-maximum suppression for accurate object part localization. *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 051–055.

Gong, W., Zhang, X., Gonzàlez, J., Sobral, A., Bouwmans, T., Tu, C., and Zahzah, E. H. (2016). Human pose estimation from monocular images: A comprehensive survey. *Sensors (Switzerland)*, 16(12).

Hedvig Kjellström and Fernando De la Torre and Michael J. Black, booktitle=FG, y. A framework for modeling the appearance of 3d articulated figures.

Insafutdinov, E., Andriluka, M., Pishchulin, L., Tang, S., Levinkov, E., Andres, B., and Schiele, B. (2017). Arttrack: Articulated multi-person tracking in the wild. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1293–1301.

Iqbal, U., Milan, A., and Gall, J. (2017). PoseTrack: Joint multi-person pose estimation and tracking. In *CVPR*.

Jhuang, H., Gall, J., Zuffi, S., Schmid, C., and Black, M. J. (2013). Towards understanding action recognition. *2013 IEEE International Conference on Computer Vision*, pages 3192–3199.

Jiang, H. (2010). Finding human poses in videos using concurrent matching and segmentation. In *ACCV*.

Johnson, S. and Everingham, M. (2010). Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*.

Kawana, Y., Ukita, N., Huang, J.-B., and Yang, M.-H. (2018). Ensemble convolutional neural networks for pose estimation. *Computer Vision and Image Understanding*, 169:62–74.

Khungurn, P. and Chou, D. (2016). Pose Estimation of Anime / Manga Characters : A Case for Synthetic Data.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.

Lehrmann, A. M., Gehler, P. V., and Nowozin, S. (2013). A Non-parametric Bayesian Network Prior of Human Pose. *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1281–1288.

Lin, T.-Y., Maire, M., Belongie, S. J., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *ECCV*.

Linna, M., Kannala, J., and Rahtu, E. (2016). Real-time human pose estimation from video with convolutional neural networks. *CoRR*, abs/1609.07420.

Newell, A., Yang, K., and Deng, J. (2016). Stacked Hourglass Networks for Human Pose Estimation. *ECCV*.

Pfister, T., Charles, J., and Zisserman, A. (2015). Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE International Conference on Computer Vision*.

Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P. V., and Schiele, B. (2016). Deepcut: Joint subset partition and labeling for multi person pose estimation. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4929–4937.

Shen, H., Yu, S.-I., Yang, Y., Meng, D., and Hauptmann, A. G. (2014). Unsupervised video adaptation for parsing human motion. In *ECCV*.

Shi, Q., Di, H., Lu, Y., Lv, F., and Tian, X. (2016a). Video Pose Estimation with Global Motion Cues. *Neurocomputing*, 219:269–279.

Shi, Q., Di, H., Lu, Y., Qin, M., and Tian, X. (2016b). Video pose estimation via medium granularity graphical model with spatial-temporal symmetric constraint part model. *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1299–1303.

Singh, D. (2017). Human pose estimation: Extension and application.

Sucar, L. E. (2015). *Probabilistic Graphical Models: Principles and Applications*. Springer Publishing Company, Incorporated.

Tian, J., Li, L., and Liu, W. (2015). A robust framework for 2D human pose tracking with spatial and temporal constraints. In *2014 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2014*.

Tompson, J., Jain, A., LeCun, Y., and Bregler, C. (2014). Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. *Advances in neural information processing systems*.

Toshev, A. and Szegedy, C. (2014). Deeppose human pose estimation via deep neural networks. *Computer Vision and Pattern Recognition (CVPR)*, pages 1653–1660.

Umer Raf and Bastian Leibe, J. G., Rafi, U., and Leibe, B. (2016). An Efficient Convolutional Network for Human Pose Estimation. *BMVC, 2016*, pages 1–11.

Varol, Romero, J., Martin, X., Mahmood, N., Black, M. J., Laptev, I., and Schmid, C. (2017). Learning from synthetic humans. *CoRR*, abs/1701.01370.

Viterbi, A. J. (1967). Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory*.

Wei, S. E., Ramakrishna, V., Kanade, T., and Sheikh, Y. (2016). Convolutional Pose Machines. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4732.

Yang, W., Ouyang, W., Li, H., and Wang, X. (2016). End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3073–3082.

Zhang, D. and Shah, M. (2016). A framework for human pose estimation in videos. *CoRR*, abs/1604.07788.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. H. S. (2015). Conditional Random Fields as Recurrent Neural Networks. *Iccv*, pages 1529–1537.

Zou, B., Chen, S., Shi, C., and Providence, U. M. (2009). Automatic reconstruction of 3D human motion pose from uncalibrated monocular video sequences based on markerless human motion tracking. *Pattern Recognition*, 42(7):1559–1571.