6-1-2017

Solving the confusion of body sides problem in two-dimensional human pose estimation

Mohammad Hamdy Oreaba

The American University in Cairo

School of Sciences and Engineering

Solving the Confusion of Body Sides Problem in Two-Dimensional Human Pose Estimation

A Thesis Submitted to

The Department of Computer Science and Engineering

In partial fulfillment of the requirements for the degree of Master of Science

By

Mohammad Hamdy Oreaba

B.Sc. Computer Science

Under The Supervision of

Prof. Mohamed Moustafa

Spring 2017

Dedication

To my mom.

Acknowledgements

"... but over every possessor of knowledge is one [more] knowing [Allah]."

–Quran 12:76

My thanks go to my parents for their endless support. I also would like to extend my gratitude to my brothers and my sister for their caring and for always pushing me towards success. I am lucky to have you all in my life. I would not have made it without all your support.

My deepest gratitude goes to Prof. Mohamed Moustafa, my supervisor, for his continuous support and guidance throughout my thesis research. I cannot also forget to express my grateful feelings to Dr. Ahmed Rafea for his helpful suggestions and for his guidance throughout my study at AUC.

I would also like to recognize SAFRAN-France (MORPHO) for sponsoring a fundamental part of this work under the Research Award Program. My sincere thanks I give to every teacher, professor, colleague, relative or a friend who added to my knowledge, encouraged, helped, or even prayed for me.

All praise to Allah, the most gracious and the most merciful, lord of the worlds. To him my initial and final thanks. I am forever indebted.

"Praise to Allah, who has guided us to this; and we would never have been guided if Allah had not guided us." –Quran 7:43

Table of Contents

List of Tables

List of Figures

Abbreviations

2D HPE: Two-Dimensional Human Pose Estimation

3D HPE: Three-Dimensional Human Pose Estimation

Accuracy measure: FN False Negative (Actual Positive and Predicted Negative)

Accuracy measure: FP False Positive (Actual Negative and Predicted Positive)

Accuracy measure: TN True Negative (Actual Negative and Predicted Negative)

Accuracy measure: TP: True Positive (Actual Positive and Predicted Positive)

CBS: Confusion of Body Sides

CV: Computer Vision

DCNN: Deep Convolutional Neural Network

DFT: Discrete Fourier Transform

DoF: Degrees of Freedom

FPS: Frames per Second

HAR: Human Action Recognition

HMM: Hidden Markov Model

HoF: Histogram of Optical flow

HoG: Histogram of Oriented Gradients

HPE: Human Pose Estimation

NLP: Natural Language Processing

PCA: Principle Component Analysis

PE: Pose Estimation

PSM: Pictorial Structure Model

RoI: Regions of Interests

SHAPE: Smart Human Articulated Pose Estimation

SIFT: Scale-Invariant Feature Transform

SLAM: Simultaneous Localization and Mapping

SoA: State-of-the-Art

SVM: Support Vector Machine

Abstract

In this thesis, we address the problem of two-dimensional human pose estimation (HPE) from a single viewpoint. While many approaches to estimate the 2D human pose from a single viewpoint exist, the estimated joints' locations with respect to the viewpoint are often disregarded. This limits the overall accuracy of localizing the human body parts. To address this limitation, we define a novel problem in 2D HPE: the Confusion of Body Sides (CBS). We show the CBS problem in many 2D HPE approaches as well as in the state-of-the-art methods. In order to overcome the CBS problem, we introduce SHAPE: Smart Human Articulated Pose Estimation. We demonstrate how SHAPE can be plugged into a 2D HPE algorithm to solve the CBS problem. We report our qualitative and quantitative results on our proposed challenging dataset: 'Humans AUC' as well as on two popular HPE benchmark datasets: 'KTH Multiview Football dataset II' [1] and 'Image Parsing' [2]. Our approach is shown to make a notable 2D HPE approach [3] viewpoint-invariant and enhance the accuracy by 20% on average.

CHAPTER ONE

INTRODUCTION

In this chapter, first I will briefly talk about the vital role of computers in our lives and the emergence of the Computer Vision (CV) field. Second, I will discuss the problem definition, the motivation behind this work, and the challenges in this research. Finally, I will briefly discuss the objectives of this research and the thesis structure.

1.1 An Overview

Computers have been utilized as the foundation for many daily-use activities. The vital role computer systems play in this era endorse the undisputed fact of their importance. Such importance is an outcome of the high computational power those computer systems offer these days. Additionally, the quick accessibility, the ease of use, and the high accuracy of modern computer systems have all contributed to making such systems indispensable in most humankind activities. In addition, the ongoing improvements in computer hardware have been inspiring researchers to invest time and make persistent efforts in order to offer genuine solutions for many widely-known challenging problems.

Mankind has the ambition of making computers as capable as humans. This passion has led many scientists all over the world to pursue novel ways to try to make computers analyze and interpret data like humans. Researchers sometimes rely on simple hardware to feed in data, e.g. a mic to input sounds, and sometimes they rely on complex hardware, e.g. a sophisticated digital camera(s) to input images or videos. Therefore, the nature of the problem defines the complexity of the hardware used to achieve the mankind's dream.

There have been many explorations and experiments to advance Artificial Intelligence in Computers. On one hand, many recent advancements have been achieved in the field of Natural Language Processing (NLP) to make computers recognize speech, provide real-time translations, and take actions based on someone's voice tone. On the other hand, numerous studies have been done in the CV field to empower computers by making them perceive the surroundings. All such studies in both NLP and CV alike strive to provide intelligent systems in various areas.

CV is a field of developing techniques to acquire, process, analyze, and understand images. Such images are represented as high-dimensional data from the real world for the aim of producing numerical or symbolic information in the forms of decisions [4]. There are many applications of CV such as agriculture, augmented reality, autonomous vehicles, biometrics, character recognition, forensics, industrial quality inspection, face recognition, gesture analysis, geoscience, image restoration, medical image analysis, pollution monitoring, process control, remote sensing, robotics, surveillance, and transport [4].

In the last quarter of the twentieth century, significant interest in image and video analysis and understanding has increased. This is due to the fast pace of development in electronics and hardware which led to lowering the cost of image and video acquisition devices, imagery transmission, data storage and computational power. This rise in computational power and hardware advancements have opened new possibilities for image and video understanding.

Many fields have also enriched the field of computer vision. For example, to obtain images and videos, still-image cameras or video cameras are used, and the sensors of these cameras are based on optics and solid state physics. Another example,

accurate numerical methods are crucial to data processing. Furthermore, the field exploits the recent advances in other different, but related, fields, such as image and signal processing, computer graphics, pattern recognition, robotics, and artificial intelligence. All developments in these fields have stimulated researchers to provide novel answers for questions and problems that were difficult to provide solutions for two decades ago.

To give an insight on how CV applications are pushing researchers towards new developments, OpenCV Foundation with support from the Defense Advanced Research Projects Agency (DARPA) and Intel Corporation launched a community-wide challenge to update and extend the OpenCV library using state-of-the-art (SoA) algorithms [5].

As a prize, an award pool of $50,000 is provided to reward submitters of the best performing algorithms in 11 computer vision application [6]. These applications are as follows: (1) image segmentation, (2) image registration, (3) *human pose estimation*, (4) SLAM, (5) multi-view stereo matching, (6) object recognition, (7) face recognition, (8) gesture recognition, (9) action recognition, (10) text recognition, and (11) tracking.

Pose Estimation (PE), which was one of the 11 applications, aims to find an object's position and orientation. Considering the problem of Pose Estimation in the field of computer vision has mainly two points of view. On one hand, we may have a stationary scene while the camera is moving [7]. The purpose of this type is to find the camera location and orientation within the scene. This type of PE is useful in many applications such as navigation systems.

On the other hand, we may have a stationary camera and a moving object [7]. In a more complex scenario of the same model, it is possible to have many fixed

cameras and a set of moving objects [8]. The purpose of this type is to find the location and orientation of an object within the stationary scene. This model is needed and found beneficial in many applications, like pedestrian detection and surveillance applications. All these models of different scenarios and applications seek a common goal which is to solve the Pose Estimation problem [9].

Pose Estimation has many sub-domains like Object Pose Estimation [10]. For example, we may be interested in finding the location and the orientation of a ball in a sports game. Also, we may be interested in finding the pose of specific human body parts like hands, faces, or arms. Such problems in the field are referred as Hand Pose Estimation [11], Face Pose Estimation [12], and Arm Pose Estimation [13] respectively. Additionally, we may be interested in finding the position and the orientation of all the body joints of a human body in a still image or in a video sequence. Whether it is the upper body or the full body that we want to estimate the pose for, in the literature, this is referred as the *Human Pose Estimation* problem or HPE [14].

As discussed in this sections, there are various CV areas of great benefit to mankind nowadays. Each area has its own wide range set of applications. In this work, I focus on the full body Human Pose Estimation (HPE). In the following section, I will discuss the problem definition, motivation, challenges and the objectives of this study.

1.2 Problem Definition

Human Pose Estimation (HPE) is an important problem in Computer Vision [15]. According to Yang & Ramanan [3], the definition of HPE is "to report joint positions of articulated limbs". This means that, given an image or a video sequence, a successful HPE approach should be able to ***correctly*** find the location of the human body parts like the head, neck, torso, right arm, left leg, etc. as shown in Fig. 1.1.

The HPE problem is very broad. This is mainly because it comes in many different flavors depending on the final goal. I categorize the HPE problems into:

1. Estimate the human pose in a single frame or in a video sequence.
2. Estimate the human pose for a single person or for multiple persons.
3. Estimate the human pose from a single or multiple viewpoints.

In this thesis, we focus on estimating the human pose in a 2D image of a single person using a single viewpoint.

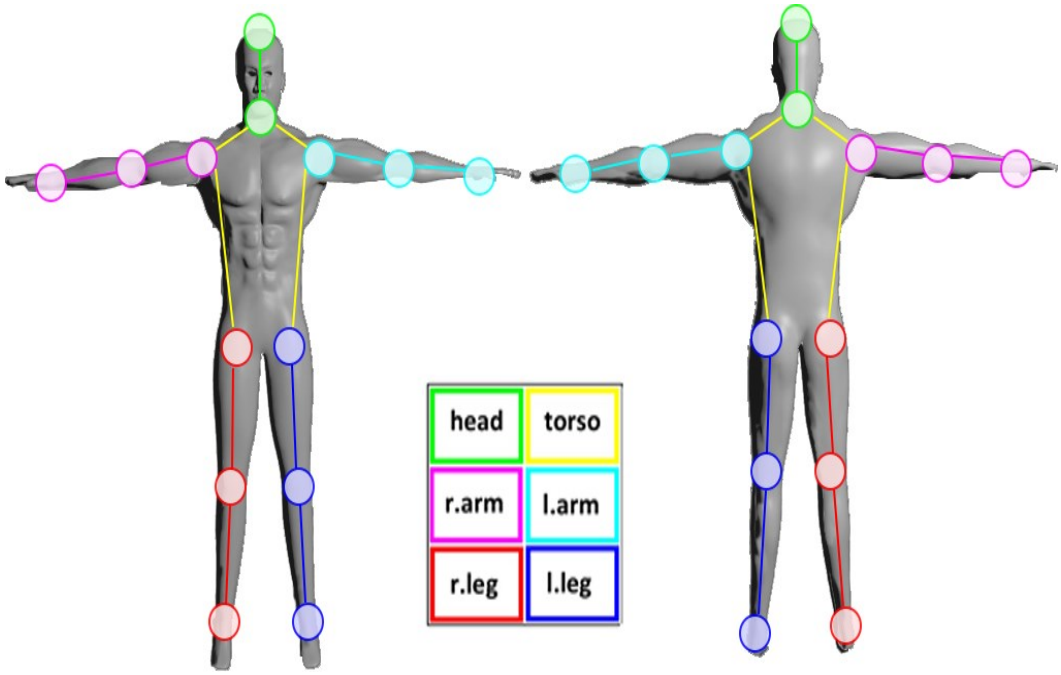


Figure 1.1: 14 joints of the human body from both views

In this section, I define a *novel* problem that current 2D HPE approaches suffer from [16][17], including the 2D HPE approach by Chen & Yuille [18] which was reported by Liu et al. to be the SoA 2D HPE [14]. I call the problem *Confusion of Body Sides* or CBS in short. CBS means that the 2D HPE approach, in certain situations, confuses the right side of the human body joints with its left symmetrical ones. That is because of the symmetrical structure of the human body. For example, a 2D HPE approach suffers from CBS when it sometimes reports the location of the left hand as if it is the location of the right one. In Fig 1.1, however, there is no confusion because, on both views, the 14 joints were *labeled correctly* irrespective of the human viewpoint.

An example of the CBS problem is shown in Fig. 1.2. when Chen & Yuille's algorithm [18] recognized the right leg as if it is the left leg as shown in Fig. 1.2 (b), and recognized the right arm as if it is the left arm as shown in Fig. 1.2 (e).

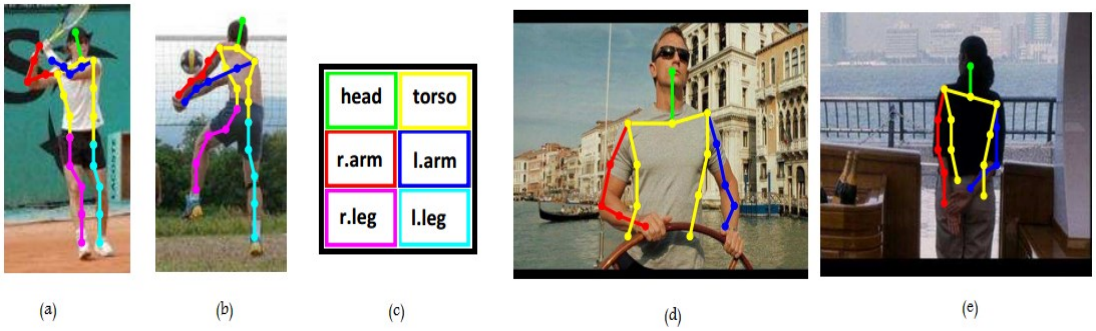


Figure 1.2: SoA 2D HPE approach suffers from the CBS problem

(a) In a full-body test, joints are found correctly.
(b) In a full-body test, [18] suffers from the CBS problem.
(c) The colored legend of body joints used in [18].
(d) In an upper-body test, joints are found correctly.
(e) In an upper-body test, [18] suffers from the CBS problem.

For many reasons, it is important to solve the CBS problem. One reason is to *adhere* to the definition provided earlier by Yang & Ramanan [3], which is to "report joint positions" ***correctly***. Another reason is that it will improve the accuracy of 2D HPE on challenging datasets that contains images of different view angles of the human body. Also, usually an HPE step is required before recognizing human actions in a Human Action Recognition system (HAR) [19]. Since the action to be recognized will be based on the joint locations found, it is important to not confuse the human body parts with each other [20].

Furthermore, for certain applications, it is critical to not confuse the right body parts with the left body parts. For example, in security applications and surveillance systems, it is costly sometimes to detect an action based on wrong inference of body parts [21]. In robot navigation systems [22], a robot should be able to locate the ***correct*** location of certain joints in order to take an appropriate action. Additionally, some

sports requires certain actions to be carried out by a specific leg or a specific arm [23]. Hence, classifying the action based on the current 2D HPE methods could make a great confusion.

Likewise, in Human-Computer Interaction (HCI) applications [13], no console input is provided to the computer, such as gaming applications and gesture recognition; Therefore, the identification of the correct pose and the correct limbs are necessary in order to interact with the human seamlessly. Imagine that, when you play a game or scan your body for a medical application, the computer detects your right arm as if it is the left one or detects the left leg as if it is the right one.

In this thesis, we propose a *novel* framework solution to the CBS problem and we call it *Smart Human Articulated Pose Estimation* or SHAPE. We chose the 2D HPE algorithm by Yang and Ramanan in [3] to improve. It is a notable 2D HPE algorithm and it was reported by [24] to be the SoA 2D HPE algorithm. Nevertheless, it suffers from the CBS problem.

1.3 Research Motivation

In this section, first, I will discuss the motivation for choosing to do my thesis in the field of HPE. Second, I will discuss why solving the CBS problem is essential. Finally, I will briefly mention the main motivations for choosing to improve the 2D approach of Yang and Ramanan [3].

First, there are many reasons that made me research the HPE field. One reason is because the demand for automated analysis of videos has been significantly increasing in recent years. Also, understanding humans' actions in images and video have gained more attention in numerous computer vision conferences this decade. For instance, many commercial applications nowadays rely on HPE as shown in table 1.1 [14].

Another important reason is because of the many areas HPE is involved in. Hence, there are many gaps need to be filled, and the room for improvement is widely open. To illustrate, there are many areas that HPE is primarily involved in, such as HAR, HCI, Augmented Reality, Behavior Understanding, and others [25]. Each of the aforementioned areas has its own wide range of applications. Thus, if it was not for human pose estimation solutions, we would not have been able to provide various applications in all these essential areas.

Systems	Principle	Application areas	Institution	Related URLs
Kinect Sensor	Structure light capture and machine learning	Motion Capture Multi-View Pose estimation	Microsoft	http://www.xbox.com/en-US/kinect
Leap Motion	Double sensors Infrared light Vision different	Gesture Recognition	Leap Motion Inc.	https://www.leapmotion.com/
Vicon	Reflected light based system	Industrial Robot Animation, Military Remote sensing, Bioinformation	Oxford Metrics Limited	http://www.vicon.com/
Wii	Bluetooth communication Infrared light	Games Physical treatment	Nintendo	http://www.nintendo.com/wii

Table 1.1: Main commercial systems for HPE

In addition to the aforementioned reasons, there is a wide range of applications that depend on 2D/3D human pose estimation. To give some examples, several key applications [19] are shown in Fig. 1.3.

Figure 1.3: HPE wide range of applications

Human Action Recognition (HAR)

In order to classify human actions, the first requirement is to detect a human in the image or the video sequence, and this is the human detection problem. The second requirement is required to estimate the pose of the detected human in order to understand the action being accomplished. Therefore, HPE basically is the core engine of any HAR system. For instance, the work accomplished by example [26] uses HPE as a pre-step to recognize actions done by humans.

Advanced human-computer interaction (HCI)

It is preferred to design and bring more natural interfaces between human and intelligent systems beyond the traditional medium like the keyboard and the mouse. Such interfaces should be able to understand natural communication methods like visual human gestures. For example, using specific hand movements to go forward and backward automatically in slides presented by a lecturer [27].

Video surveillance

Video surveillance is widely used in various places such as critical infrastructure, governmental buildings, public transportation, parking lots, homes, and office buildings. Because manually monitoring these cameras is becoming a hazard, there is a great need of approaches for automatic video surveillance including outdoor human activity analysis [28][29].

Video annotation

Nowadays a very large amount of video data can be saved easily due to the recent advances and development of hardware technology. Among such videos, we could have many human-related videos, such as sports videos, movies, and surveillance

videos. Human motion analysis can be used to annotate those videos instead of manually scanning through a large video database to get the needed information, e.g., methods to annotate video of a soccer game have been presented [30][31].

Second, the motivations to solve the CBS problem are several as discussed in section 1.2; In brief, some of these motivations are: to improve the accuracy of 2D HPE methods, to enable accurate Human Action Recognition system (HAR) [19], and to not confuse the right body parts with the left body parts in applications like surveillance, sports, medical and HCI applications .

Finally, Several reasons encouraged us to enhance the work of Yang and Ramanan [3]. One reason is because of the importance of the HPE problem. Also, the research in [3] has been cited in hundreds of research papers according to Google Scholar. Another reason is because they provide some programming libraries of their approach that make their results reproducible. Lastly, we found a potential for improvements that could advance the accuracy of 2D HPE methods on other datasets by making the algorithm sensitive to human viewpoints.

1.4 Research Challenges

CV grand challenge is video understanding. This includes dealing with four variables: Objects, Actions, Scene categories, and Geometry as shown in Fig 1.4. HPE deals with the first two variables: Objects and Actions.

Figure 1.4: CV Grand Challenge

Estimating the Human Pose can be carried out from a static image or from a video sequence. Also, HPE can be obtained from a single viewpoint or from multi-viewpoints. Determining the source input, as well as the desired viewpoint(s), define the challenges to be faced when estimating human poses. One example is the existence of many variations across human bodies, background clutter, high dimensionality, and complex appearance models.

Additionally, in order to estimate the human pose accurately in a monocular video sequence (single-view) or in a static 2D image, there are many challenges to overcome. Some of them are the high degrees of freedom (DoFs) of the human body, the large variations of human poses, changes of color cloths, change of lighting and illumination, various viewpoints of the same pose with respect to the mounted camera, and frequent of self-occlusions as shown in Fig 1.5.

Figure 1.5: Challenges to overcome in HPE

1.5 Research Objectives

The main objective of this thesis is to develop SHAPE; a solution to the CBS problem discussed in section 1.2 that would improve the accuracy of 2D HPE algorithms. Thus, we chose to improve the accuracy of a prominent full-body 2D HPE approach found in [3] by applying SHAPE to it and making it "smart" and viewpoint-invariant.

This goal will be accomplished through achieving a set of milestones. The first milestone is to reproduce the results obtained by [3]. The second milestone is to collect

our proposed data set of 2D images to test our approach on. The third milestone is to develop the CBS Solver and inject it to the 2D HPE baseline to produce SHAPE. The fourth and the final milestone is to evaluate SHAPE on two popular 2D HPE datasets as well as our proposed one. Another objective is to encourage other researchers to address and solve the CBS problem in their 2D HPE algorithms using our approach.

1.6 Thesis Roadmap

The rest of this thesis is organized as follows: **Chapter 2** builds an essential background of the different models used to estimate the human pose. It also reviews the model used by Yang and Ramanan in their work [3]. **Chapter 3** surveys related work on full body human pose estimation from a static single 2D image, keeping the focus on existent methods that suffer from the CBS problem. **Chapter 4** describes our proposed approach to solve the CBS problem and to develop SHAPE: Smart Human Articulated Pose Estimation. **Chapter 5** proposes our 2D HPE dataset (HUMANS AUC). **Chapter 6** discusses the experiments we conducted and the results we obtained. Finally, **Chapter 7** concludes the thesis and provide some insight into the future work.

1.7 Thesis Contributions

The major contributions of this thesis is that: **1)** we define a novel problem in 2D HPE approaches (CBS); **2)** we introduce a solution to the aforementioned problem (SHAPE); **3)** We improve the accuracy of a notable 2D HPE algorithm [3] – using SHAPE – that was reported by [24] to be the SoA; **4)** We present a challenging Humans dataset of 70 actors for the purpose of human detection, tracking, identity recognition, and pose estimation (4 synchronized cameras, 425 video sequences, each video is roughly one-minute length and 30 FPS); **5)** We provide the face annotation Ground truth for three datasets: PARSE, KTH, and Humans AUC (proposed).

CHAPTER TWO

BACKGROUND

In this chapter, I will briefly discuss the difference between single-view HPE and multi-view HPE. Afterward, I will give some background on the two model approaches used in HPE: Model-Based approaches and Model-Free approaches.

2.1 Single view HPE versus Multi-view HPE

Pose estimation systems can be categorized according to the number of views utilized. One of the factors that greatly determines the usage of a pose estimation system is the number of views. Basically, there are two main types: single-view human pose estimation and multi-view human pose estimation.

Single-view HPE methods: infers the human pose in a single static image or in a frame taken from a video sequence captured by one camera. According to [24], single-view HPE methods are classified into two main techniques: 2D HPE methods [2][16][17] and 3D methods [32][33][34].

On one hand, 2D single view HPE methods try to parse humans in 2D images in order to find the location and the orientation of each body parts i.e. limbs. This is the fundamental body part parsing problem in HPE, which is the core of most of the proposed work in HPE. More examples of these methods can be found in [14]. We use this class of HPE in our research.

On the other hand, 3D single-view HPE methods often referred to as Monocular depth methods, aim to find the location and the orientations of human body parts in 3D space. The 3D single view HPE uses one image from taken from a single viewpoint but it has depth information as well.

There are many ways of which 3D HPE can be inferred. One example is by using voxels or visual hull data directly when background information is [35]. Another example is by using un-calibrated configurations of cameras to infer 3D [36][37]. Also, in [38], the authors present a coarse labeling of depth pixels followed by a more precise joint estimation to estimate poses.

Multi-view HPE methods: These methods use a set of *calibrated* cameras to capture the human body from multi-views. Then, a projection of all views is performed to estimate the final human pose. Similarly, multi-view methods estimate the human pose in 2D or in 3D [39]. Different approaches fuse the multiple image sources using calibrated setups, then project models into these images as in [40]. Furthermore, orthogonal views approaches are utilized in [41].

For multiple cameras, there are two main drawbacks as mentioned by [42]. First, the multiple views of a scene are not always available. For example, it is difficult to obtain the multiple views when a pedestrian walk in public space. Second, the use of multiple cameras in a system requires camera calibration [43]. It is worth mentioning that it takes extra effort to have a good camera calibration for multiple cameras. Different cameras may have different lighting conditions, different angles of views, and different illuminations.

In conclusion, while multi-view approaches provide different views of the same person, which mitigate difficulties like self-occlusions and depth ambiguities, in interactive systems, a single view setup can be more practical. In this research, we focus on 2D static images taken from a single viewpoint.

2.2 Model-Based versus Model-Free

Body pose recovery approaches can be classified into 2 main categories: model-based and free-model approaches [24]. This classification depends on whether a prior kinematic body model is employed in the pose estimation process or not.

On the one hand, **Model-Free** class covers methods where there is no explicit prior model used [44]. These methods such as [44] and [45] try to learn some mapping between appearance and body pose. This leads to a fast performance and accurate results, but only for certain actions like walking. Also, these approaches have some limitations, such as the need of preprocessing stages like background subtraction. In addition, they are limited by a poor generalization about poses that can be detected.

On the other hand, **Model-based** approaches, which most of the HPE methods advocate, employ prior knowledge about the human body structure to recover the body posture. In these methods, the search space is reduced by taking into consideration the human body structure, its appearance, the viewpoint, and the human motion related to the activity which is carried out [24]. Because of the limitation mentioned within the model-free methods, we chose to focus on the model-based methods to estimate human poses.

2.3 Model-Based HPE categories

In this section, I will explain in details the classification of model-based methods as well as the recent techniques in each category. As proposed in [39], model-based methods are classified into *five* main techniques: appearance, viewpoint, spatial relations, temporal relation, and behavior as shown in Fig 2.1 [39]. In this research, we use the technique of spatial relations.

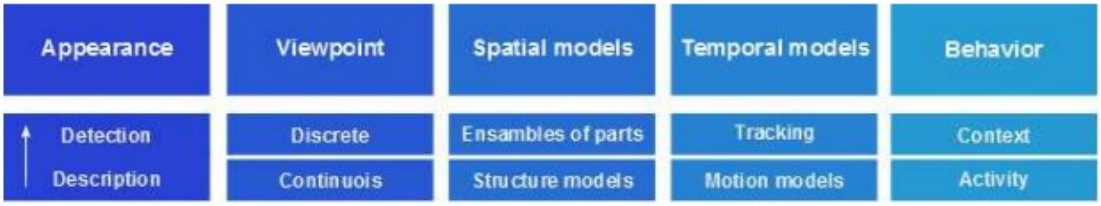


Figure 2.1: Taxonomy for model-based Human Pose Recovery approaches

2.3.1 Appearance

Appearance is considered to be image evidence related to human body and all of its possible poses. Not only image features and input data are considered evidence, but also pixel labels obtained from certain labeling procedures. Therefore, image evidence can be represented at different levels: from pixel to region and image. Image evidence has two main classifications: *description* of image features, and human *detection* i.e. human body part detection.

There are many variations of the appearance of people in images among human poses, such as clothing conditions, lighting, and changes in the viewpoint. The explanation described in this section tries to generalize over these kinds of variations because the final goal is to recover the kinematic configuration of a person.

In order to obtain accurate detections and tracking of the human body, prior knowledge of appearance and pose is required. This knowledge can be organized in two sequential stages: *description* of the image, and *detection* of the human body (or parts of the body) by applying some kind of a learning process. The procedure, starting from image description to the detection of some regions, can be performed at three different levels: pixel, local and global as shown in Fig. 2.2. These procedures lead to image segmentation, detection of some parts of the human body, and full body location [39].

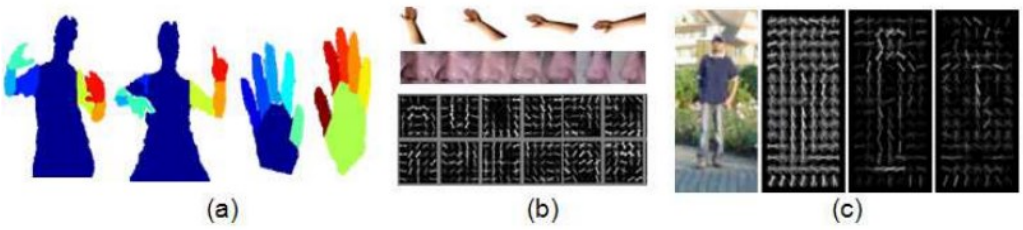


Figure 2.2: Descriptors applied at the pixel, local and global levels.

(a) Graph cut approach for body and hands segmentation (frame extracted from [39]).
(b) Steerable part basis (frame extracted from [39])
(c) Image of a person and its HOG descriptor (frame extracted from [39])

Despite the fact that, describing the human body as an ensemble of parts improves the recognition of the human body in complex systems, it increases the computational time dramatically. By contrast, in human detection field, global descriptors are successfully used, allowing fast detection of certain poses like pedestrian detections. It also works as initialization in human pose recovery approaches [46][3]. The sub-categories for both *description* and *detection* are explained in the following paragraphs.

- ***Description***

In the description phase, information is extracted from images, then they will be analyzed in the detection phase. Some typical methods that are applied for describing image cues are discussed below.

(a) Silhouettes and contours

The boundaries of the silhouettes whether they are edges or contours provide powerful descriptors invariant to changes in texture and color. Most of the body pose information remains in its silhouette. That is why silhouettes are used to fit the human body in images [47]. However, these methods have some limitations. For example, they suffer from bad and noisy segmentations in real-world scenes. In addition, because of

the lack of depth information, they suffer from the difficulty of recovering some Degrees of Freedom (DoF).

(b) Intensity, color, and texture

On the one hand, texture and color can be used as additional cues for a local description of regions of interests (RoI) [48]. For describing textures, usually Discrete Fourier Transform (DFT) is used [49], or wavelets such as Gabor Filters [50]. For describing colors, histograms or space color models are usually used to codify color information [51]. On the other hand, the most widely applied features for describing the appearance of a person are gradients on image intensities. Scale-Invariant Feature Transform (SIFT) descriptors and Histograms of Oriented Gradients (HoG) are considered [52].

Depth maps can now be obtained from the multi-sensor Kinect, which opened the door for human pose estimation to consider depth cues. This example of a cheap sensor provides near 3D information synchronized with RGB data. Examples exist in the literature, such as novel key point detectors based on the saliency of depth maps [53], and Gabor filters over depth maps for hand description [54]. Such approaches have the advantage of fast computing and discriminative descriptions by computing histograms of normal vectors distribution. However, these approaches require a specific image cue, and depth maps are not always available when needed.

(c) Motion Optical flow

To model path motion [55] is the most common feature used. It also can be used to classify human activities [56]. In addition, work in [57] codifies the motion provided by certain visual regions as an additional local cue. In this approach the same idea of

HoG is used, Histogram of Optical flow (HoF) can be constructed to describe regions as well as body parts movements.

(d) Logical

New descriptors including logical relations have been proposed in the following study [58]. In this work, local features are codified using logical operators. This allows intuitive and discriminative description of image context or RoI.

- ***Detection***

This phase refers to the output of classifiers which codify the human information in images. Below is a summary to discuss the four general areas in which this synthesis process can be performed.

(a) Discriminative classifiers

The first step is to describe image regions using standard descriptors like HOG, which is a common technique to detect people in images. The next step is to train a discriminative classifier like Support Vector Machines (SVM) as a global descriptor of the human body [52] or as a multi-part description and learning parts [59]. Spatial relations between descriptors in a second level discriminative classifier have been proposed by some authors, as in the case of poselets [32], to extend this kind of approaches.

(b) Generative classifiers

Generative approaches have been proposed to address person detection as in the case of discriminative classifiers. Nevertheless, in generative approaches, the problem of person segmentation is considered and dealt with. One example is the work by [60],

which learns a color model from an initial evidence of a person and background objects. They used Graph Cuts to optimize a probabilistic function.

(c) Templates

Another approach for human pose estimation is to use Example-Based methods [48] to compare the observed image with a database of samples.

(d) Interest points

In order to compute the pose or the behavior that is being carried out in a video sequence, salient points or parts in the images can be used [57]. A fair list of region detectors is described in [61].

2.3.2 Viewpoint

Viewpoint estimation significantly reduce the ambiguities in 3D body pose [48]. In most of the cases, body viewpoint is not directly estimated in pose recovery or human tracking. It is indirectly considered sometimes though. The possible viewpoints to be detected are sometimes constrained in the training dataset. For example, there exist datasets where upper body pose estimation or pedestrians are presented. In such cases, only front or side views are studies respectively. To illustrate, a detector presented in [62] can detect people in arbitrary views. However, only walking side views are considered in performance evaluation. In addition, some other works restrict their approaches explicitly to a reduced set of views. For example, in [63] frontal and lateral viewpoints are considered. In 3D viewpoint estimation, research can be divided in discrete classification and continuous viewpoint estimation as shown in Fig. 2.1.

In the **discrete approach**, the problem is considered to be viewpoint classification. In this approach, the viewpoint of a query image is classified into a limited set of possible initially known[64][65] or unknown [65] views. As in the work

done by [48], a discrete viewpoint is estimated for pedestrians by training eight viewpoint-specific people detectors a shown in Fig. 2.3 (a). In the following stage, the classification is used to refine the viewpoint in a continuous way as shown in Fig. 2.3 (b), estimating the rotation angle of the person around the vertical axis by the projection of 3D exemplars onto 2D body parts detections.

Where in the **continuous approach**, the problem refers to estimating the real-valued viewpoint angles for human in 3D or an object. Continuous viewpoint estimation is studied widely in the field of shape registration [66]. Some work has been done in this area, such as in [67][68], where authors modeled the possible camera poses as a Gaussian Mixture Model to provide a prior knowledge of the camera as shown in Fig. 2.3 (c).

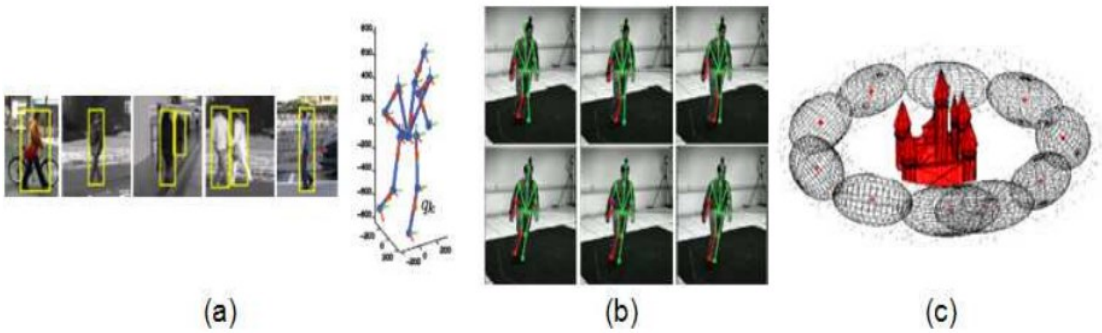


Figure 2.3: Viewpoint estimation examples

(a) First (discrete) (frame extracted from [39]).
(b) second (continuous) (frame extracted from [39]).
(c) Clusters of the camera pose space around the object which provide continuous viewpointImage of a person and its HOG descriptor (frame extracted from [39]).

2.3.3 Spatial Models

There are two main ways to encode the configuration of the human body in spatial models. First in a hard way, e.g., skeleton, bone lengths. Second in a soft way e.g., pictorial structures [67], grammars [68]. In order to encode *structure models*, 3D

skeletons and accurate kinematic chains are used [63][69] Also, in order to model the degenerative projections of the human body in the image plane, *ensembles of parts* are used as shown in Fig. 2.4. Regardless of the chosen strategy, HPE aims at estimating the full body structure, or upper body pose estimation [70][62][71]. Several works [72] [16] and datasets [71] have been restricted to upper body estimation because in TV shows and many scenes on films legs do not appear in the visible frame.

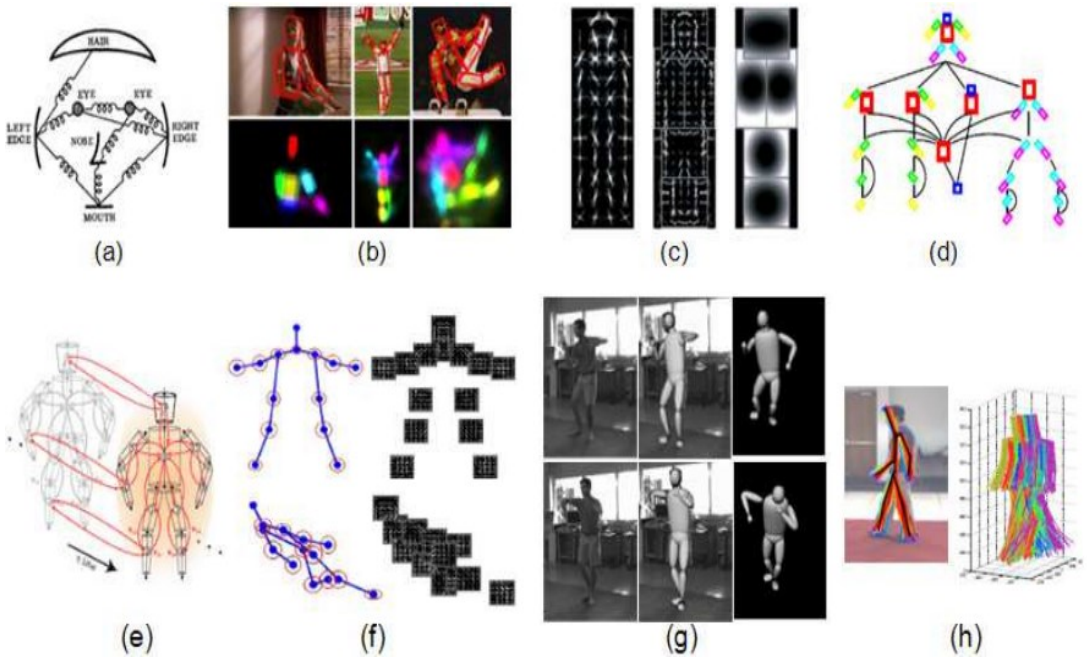


Figure 2.4: Examples of body models as ensembles of parts

(a) Original.
(b) Extended Pictorial Structures.
(c) Human model based on grammars.
(d) The hierarchical composition of body "pieces".
(e) Spatio-temporal loopy graph.
(f) Different trees obtained from the mixture of parts Structure models.
(g) Two samples of 3D pose estimation during a dancing sequence.
(h) Possible 3D poses.
- All frames are extracted from [24].

2.3.4 Temporal Models

Temporal consistency is studied when a video sequence is available in order to reduce the search space. To analyze the behavior that is being performed, the motion of body parts may be incorporated to refine the body pose.

Tracking

To ensure the coherence among poses over the time, tracking is applied. It can be applied either separately to all body parts or only a representative position for the whole body can be taken into account. In addition, 2D tracking can be applied to pixel positions, and also or world positions when the person is moving in 3D.

In tracking, there are two main subdivisions, the first one where a single hypothesis is maintained over the video sequence. For example, in [63] only the central part of the body is estimated through a Hidden Markov Model (HMM). Also in [72], a single hypothesis by each body joint is propagated in 2D. The second one is when there are multiple hypotheses propagated in time. In the end, the body pose is recovered in 2D from the refined position of the body.

A huge diversity of movements can be performed by the human body. However, smaller sets of movements can define specific actions (e.g., in cyclic actions as walking). Therefore, when a single action is performed, a set of motion priors can describe the whole body movements. However, in [73] the author establish the hard restrictions on the possible motions recovered. Motion models are introduced in [74]. Body models of walking and running sequences were combined. Also, to obtain an accurate tracking, a dimensionality reduction is performed by applying Principle Component Analysis (PCA) over sequences of joint angles from different examples. An extension of this work was presented in [75] for golf swings from monocular images

in a semi-automatic framework. In [76], more applications of such motion models related to human pose can be found.

2.3.5 Behavior

This category presents the methods that take into account context information or activity to provide feedback to previous pose recognition modules. The term *behavior* here means a general concept to include gestures and actions. Despite the fact that behavior analysis is not usual in the SoA of pose estimation, some works take into consideration the activity or behavior to accurately estimate a body pose. Some works go a step further in the literature and recover pose and behavior. For example, in [77] the authors include context information about human activity and its interaction with objects to improve both the final pose estimation and activity recognition. Ambiguities have been reported among classes though. In addition, in [78] Andriluka and Sigal extended their previous work in multi-people 3D pose estimation by modeling the human interaction context.

CHAPTER THREE

LITERATURE SURVEY

This chapter is organized as follows: in section 3.1, I will discuss the Pictorial Structure Model (PSM). This will include the history of PSM, the use of PSM in the Human Pose Estimation field, and the baseline by Yang and Ramanan [3]. Afterward, in section 3.2, I will give more insight on current 2D HPE methods that suffer from the CBS problem.

3.1 Pictorial Structure Model (PSM)

In this section, first, I will give a brief history about the birth of Pictorial Structure by Fischler in 1973 [67], followed by an early attempt to represent the human body by Marr in 1978 [79]. Second, I will give some background on the work done by Felzenszwalb in 2005 [80] which uses the Pictorial Structure Model (PSM) in Object Recognition. Finally, I will discuss the use of PSM in HPE by Yang [3].

3.1.1 History overview

Pictorial structures have been first proposed by Fischler and Elschlager [67] in 1973 as a simplified way to describe an object [34]. They represent the structure as a *graph*. Pictorial structures consist of two elements: 1) atomic object parts and 2) connections between these parts as shown in Fig. 3.1. In other words, it decomposes the appearance of objects into local part templates, together with geometric constraints on pairs of parts, often visualized as springs.

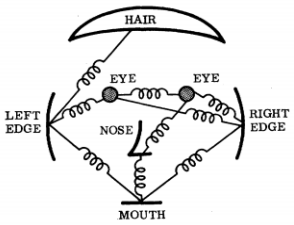


Figure 3.1: A face representation indicating components and their linkages [67]

An early attempt to model the entire human body was proposed by Marr & Nishihara in 1978 [79]. As shown in Fig. 3.2, the classic approach modeled the human as a set of parts, such as a head, torso, arm, and leg part [79]. In 3D, these parts can be modeled as cylinders.

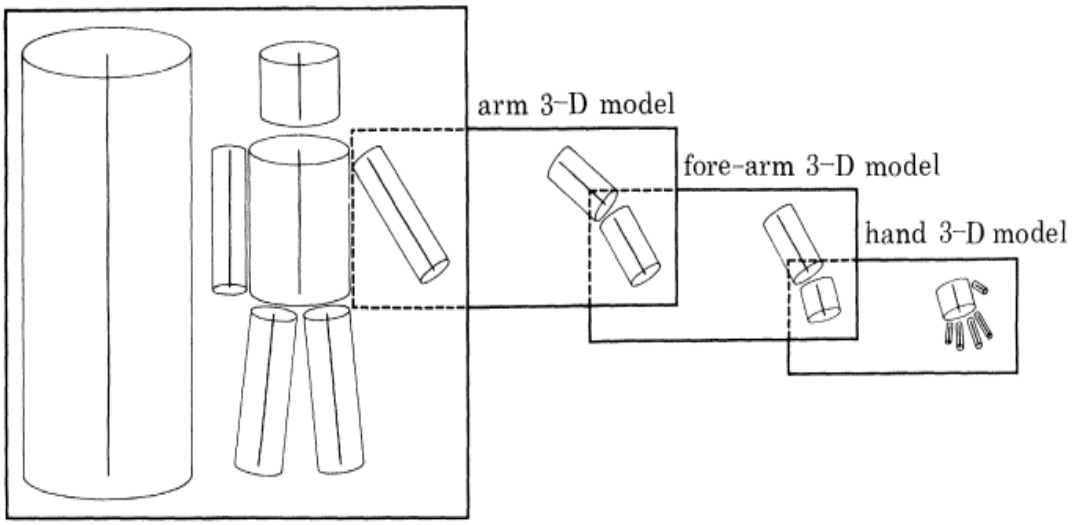


Figure 3.2: Human 3D model

3.1.2 PSM in HPE

In [80] Felzenszwalb & Huttenlocher presented a computationally efficient framework for part-based modeling and recognition of objects. Their work was motivated by the pictorial structure models introduced by Fischler and Elschlager in 2005 [67] more than forty years ago. The main idea is to represent an object by a *collection of parts* arranged in a deformable configuration.

Unlike Fischler & Elschlager [67], which have represented their structure as a graph, Felzenszwalb and Huttenlocher [80] have represented the underlying body model as a *tree* as shown in Fig 3.3 (b) due to inference facilities studied in [80]. Tree models are efficient and allow for efficient inference, yet they are plagued by the well-known phenomena of double-counting [80].

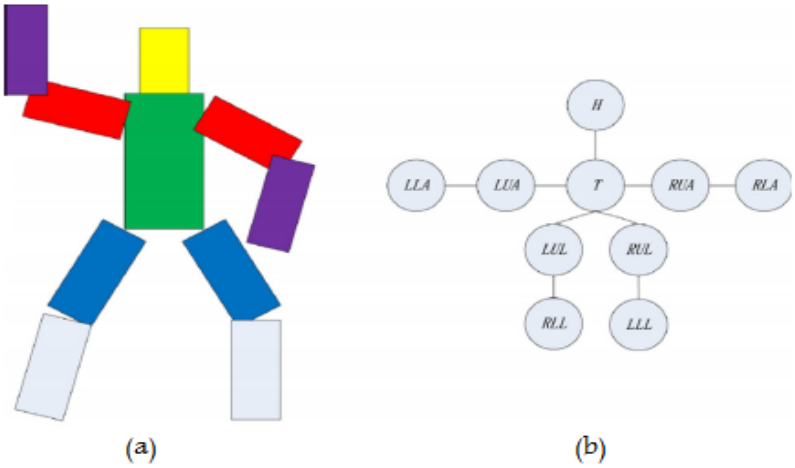


Figure 3.3: The pictorial structures model (PSM)

(a) the tree nodes representing each body parts and
(b) the connection between different body nodes.

Felzenszwalb & Huttenlocher model the human body as shown in Fig. 3.3 (a) as a collection of separate but elastically connected limb sections. Each limb is associated with its own detector where each part is detected with its specific detector. Each body part (limb) is represented by a tree node, and all the body parts are connected to its neighboring body parts. This collection of limbs is arranged in a tree structure as shown in Fig. 3.3 (b).

The approach works by searching for each individual body part. Afterward, the optimal pose is estimated by combining the detection results of individual parts efficiently. They showed how pictorial structures can be computed efficiently with dynamic programming if the representation has no cycles. Finally, they showed applications to 2D face detection and human pose estimation [80].

PSM is a special case of the tree model which was first introduced more than forty years ago by Fischler & Elschlager [67]. It differs from other tree models in that each of its nodes is modeled individually in a deformable form. Then, spring-like connections are used to connect different parts.

This special structure enables the PSM to have rich appearance variations. Also, the body parts in an articulated structure are inherently dependent on each other. Hence, imposing some *articulated constraints* on the tree model is helpful in body parts parsing. *Kinematic* Constraints between parts are modeled following Gaussian distributions [80].

Previously, in section 2.2 (Model-Based Vs. Model-Free) I discussed the difference between HPE model-based methods and mode-free methods. PSM is a model-based approach. Also, in section 2.3 (Model-Based HPE categories) I explained with examples the five classifications of model-based methods. PSM lies under spatial models.

The PSM had not been applied to HPE until investigated by Felzenszwalb and Huttenlocher [80] in 2005 as mentioned by [14]. Pictorial structures, which are generative 2D assemblies of parts, has now become the general framework for object detection. Currently, it is widely used for people detection and the most popular generative model in HPE according to [24].

3.1.3 PSM with "mini-part" of Yang & Ramanan

On one hand, the traditional models for ***object recognition*** parameterize parts solely by location, which simplifies both inference and learning. Such models have been shown in [80] to be very successful for object recognition. On the other hand, the dominant approach in Human Pose Estimation was to parameterize parts by both pixel location and orientation. In this way, the resulting structure can model ***articulation***. From this, Yang & Ramanan introduced a novel unified representation [3] which combines both models, and they produced SoA results for human pose estimation.

The key idea in their work is to divide every body part or *limb* into a set of ***mini-part model*** as shown in Fig 3.4. “Mini-Part” model can approximate deformations. They do this with a many miniature part model that models the appearance of a single limb using multiple parts connected with springs. For example, the lower leg of the panda can be modeled with two parts and the torso of the panda can be modeled with four parts.

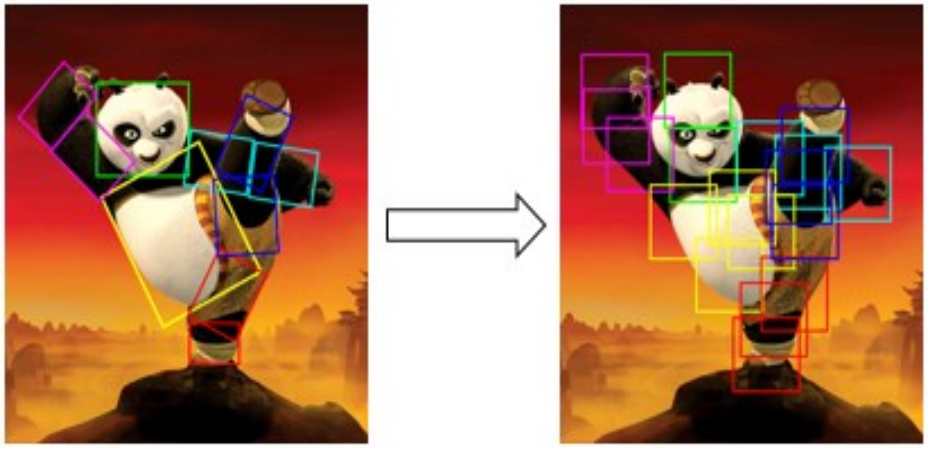


Figure 3.4: "mini part" model

Yang and Ramanan [3] proposed a mixture model in order to describe the body joints and their relationship. Each body joint is represented as a non-oriented mixture part as shown in Fig 3.5 (b), and each part is approximated to represent vertical or horizontal limbs as shown in Fig. 3.5 (c).

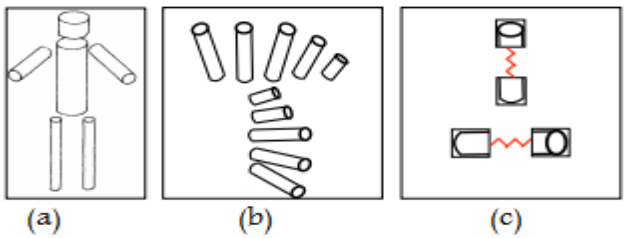


Figure 3.5: PSM Spatial Relations

(a) Shows the classic articulated limb model of Marr and Nishihara [79].
(b) Shows different orientation and foreshortening states of a limb, each of which is evaluated separately in classic articulated body models.
(c) Yang and Ramanan approximate these transformations with a mixture of non-oriented pictorial structures, in this case, tuned to represent near vertical and near horizontal limbs.

To illustrate, see their “mini” part model for modeling a lower arm. They visualize 3D transformations of an arm as 2D image foreshortening, as shown in Fig. 3.6, and rotation, as shown in Fig. 3.5 (b). They approximate these transformed images with a two-part model. If the arm rotates or foreshortens a lot, they use a different set of templates and springs. This means they use a pool of part templates and springs to capture such transformations.

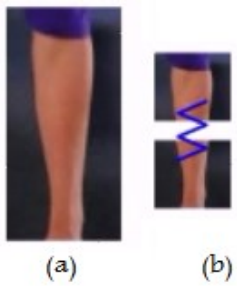


Figure 3.6: Lower arm mini part model

In order to deal with high deformations of human body and changes in parameters of the body, model and appearance were learned simultaneously. This mixture of parts would result in having different trees, see Fig. 3.7. Multi-view trees represent an alternative because a global optimum can be found using dynamic programming, or branch and bound algorithms. Yang and Ramanan use HOG feature and deal with a single person in a single image, and their main technique was Non-Maximum Suppression (NMS) SVM.

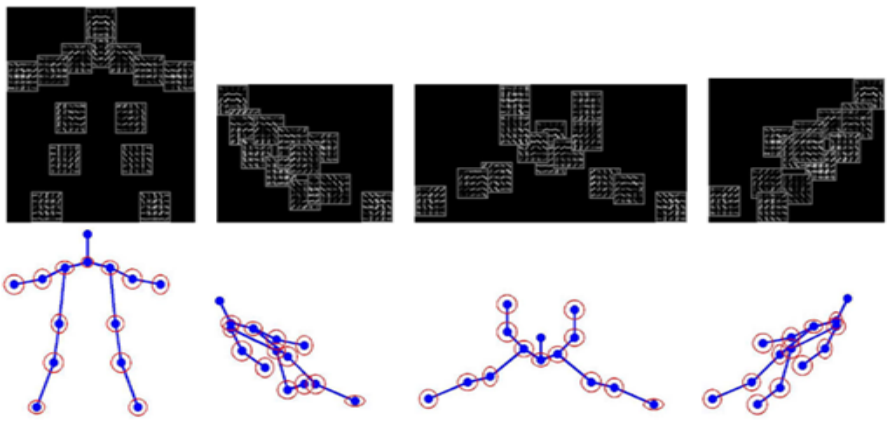


Figure 3.7: Different trees obtained from the mixture of parts

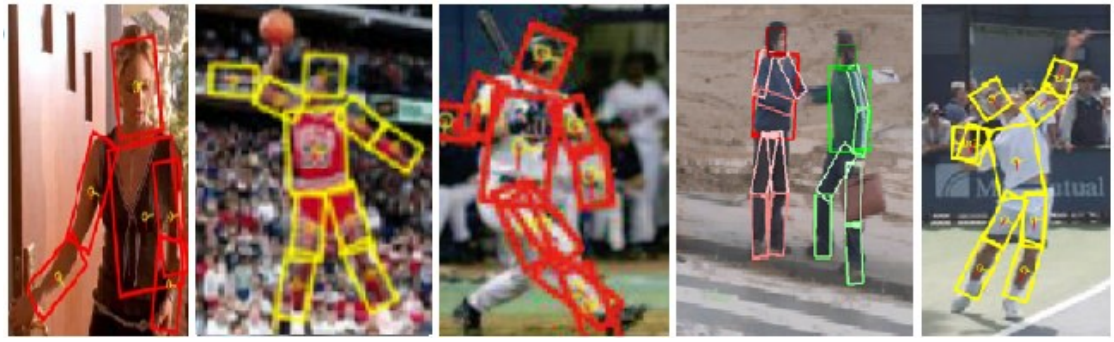
The deformable mixture-of-parts, proposed by Yang & Ramanan in [3] for estimating human pose in static single 2D images, is a fast approach based on strong body part detectors. Flexible tree configuration was proposed, and encoding pairwise relations between consecutive body parts were provided. According to [24], Yang & Ramanan [3] achieves the best results which make it the SoA in 2D single Human Pose Estimation. Nevertheless, their approach is not 'smart' enough and suffers from the CBS problem.

3.2 HPE methods suffer from the CBS

In this section, I will relate some of the previous work in 2D human pose to the *novel* problem discussed in section 1.2 (Problem Definition). Therefore, in the following sections, I will show that current 2D HPE approaches have not considered different viewpoints of human bodysides, and therefore, they suffer from the CBS problem.

3.2.1 Methods do not consider the CBS

Many of the 2D HPE surveyed methods escape from addressing the CBS problem. They do not consider the CBS problem by not considering any difference between the two human body sides. They focus on detecting the body parts, i.e. limbs, irrespective of the viewpoint and regardless of the body side as shown in Fig. 3.8.



(a) [62] (b) [81] (c) [82] (d) [83] (e) [84]

(f) [85] (g) [86] (h) [87] (i) [88] (j) [89]

Figure 3.8: Related work that do not consider the CBS problem

Some methods focus on detecting the body joints and representing all body parts with a single color as shown in the first row of Fig. 3.8. These methods have not considered the CBS problem since they do not differentiate between the two body sides, such as Andriluka et al. [62] as shown in Fig. 3.8 (a), and Fihl et al. [83], as shown in Fig. 3.8 (d). Other methods, recognizes the two body sides and set different colors for each joint, but they represent each symmetrical joint with the same color as shown in the second row of Fig. 3.8. These methods also have not taken into consideration the CBS, such as Eichner et al. [85] as shown in Fig 3.8 (f), and Kaliamoorthi et al. [86] as shown in Fig. 3.8 (g).

The methods that have not taken into account the different viewpoints of the human body sides often measure the accuracy by referring to both symmetrical body joints using one term. For example, they use the term ‘Upper arms’ to address both: the ‘right arm’ and the ‘left arm’. Similarly, they use the term ‘Lower Legs’ to refer to both:

the ‘left leg’ and the ‘right leg’. Hence, they also have not taken into consideration the CBS problem. All of the methods estimate the human pose in 2D but some take the input as an image and others provide the input as a video sequence. Also, some methods focus on Upper-Body human pose estimation, some focus on Full-Body HPE, and some can work in both situations. In table 3.1, I have summarized the related work that do not address the CBS problem and sorted them chronologically.

Year	First author	Dim	Images/ video	Upper/Full body	Do not consider CBS
2009	Andriluka [62]	2D	images	Full	✓
2009	Eichner [85]	2D	images	Upper	✓
2010	Fihl [83]	2D	Videos	Full	✓
2010	Eichner [88]	2D	Images	Upper	✓
2011	Vajda [82]	2D	Videos	Full	✓
2012	Fei [87]	2D	images	Upper	✓
2012	Schiele [81]	2D	images	Full	✓
2013	Kaliamoorthi [86]	2D	Videos	Full	✓
2013	Andriluka [84]	2D	images	Full	✓
2014	Luo [90]	2D	<Both>	<Both>	✓
2015	Pfister [89]	2D	Videos	Upper	✓

Table 3.1: Related work that do not consider the CBS problem

3.2.2 Methods suffer from the CBS

Although many 2D HPE methods have taken into account the two different human body side, they suffer from the CBS problem. This is mainly because of the symmetrical structure of the human body.

(a) [91] (b) [16] (c) [92] (d) [93] **(e)** [3]

(f) [94] (g) [95] (h) [96] (i) [18] (j) [97]

Figure 3.9: Related work that suffers from the CBS problem

An approach suffers from the CBS problem when it is insensitive to the viewpoint. Hence, it confuses a certain body part with its symmetrical one. For instance, an HPE algorithm suffers from CBS when it recognizes the 'Right hand' correctly in one situation, then recognizes it as if it is the 'Left hand' in another

situation. Similarly, the same confusion happens with the {shoulder, elbow, arm, wrist, hip, knee, and ankle}.

As shown in Fig. 3.9 (a-e), although the algorithm successfully recognizes both symmetrical body parts and assigns different colors to each symmetrical pairs on the first row, it fails in being consistent when the viewing angle of the human changed, and hence, it confuses the entire right body side with the left body side on the second row in Fig. 3.9 (f-j). At the end, this leads to an *imprecise* final pose of the human.

In Fig. 3.10, once can notice that the 3D HPE algorithm, which also uses the Pictorial Structure Model, is able to locate the joints' locations correctly. This is evident because the algorithm ***always*** colors the 'Right leg' with a **Cyan** color irrespective of the viewing angle of the person. This is because the problem is solved implicitly by providing multi-views of the same scene. Therefore, the 3D HPE methods do not suffer from the CBS problem by nature.

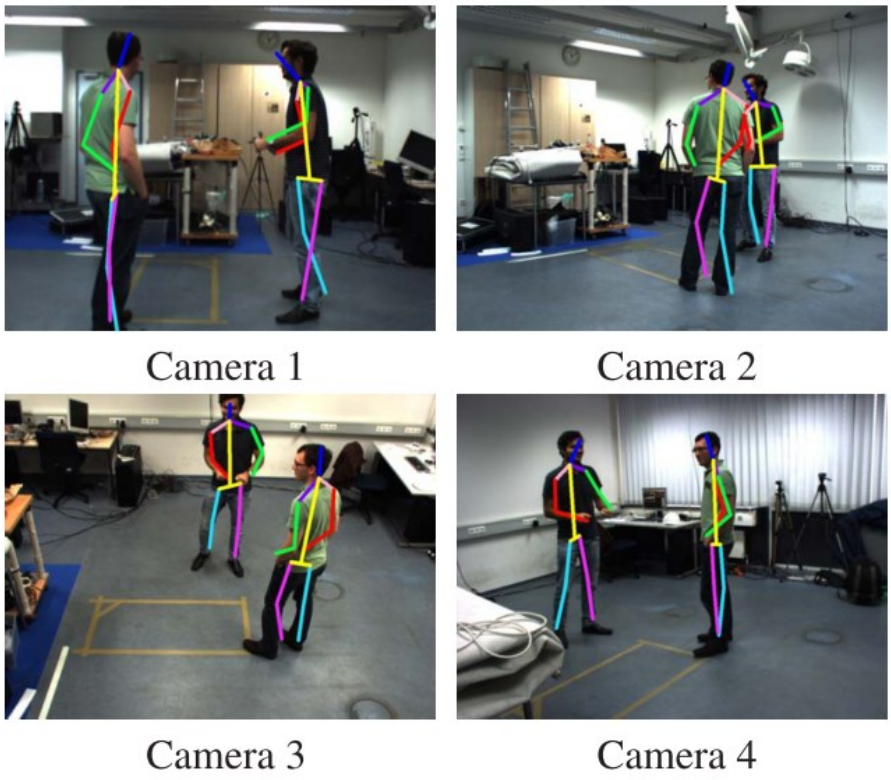


Figure 3.10: A 3D HPE method that does not suffer from the CBS problem [98]

The 2D HPE methods mentioned in this section have different input source. Some take the input as an image, some provide the input as a video sequence and other methods can work in both situations. Likewise, some methods focus on Upper-Body human pose estimation, some focus on Full-Body HPE, and some can work in both situations. In table 3.2, I have summarized the related work that suffers from the CBS problem and sorted them chronologically.

Year	First author	Dim	Images/ video	Upper/Full body	Suffer from CBS
2011	Sun [91]	2D	images	Upper	✓
2012	Eichner [16]	2D	<Both>	Upper	✓
2013	Dantone [92]	2D	images	Full	✓
2013	Wang [93]	2D	images	Full	✓
2013	Yang [3] (baseline)	2D	images	Full	✓
2013	Toshev [94]	2D	images	Full	✓
2014	Ouyang [95]	2D	images	Full	✓
2014	Ramakrishna [96]	2D	images	Full	✓
2014	Chen [18]	2D	images	<Both>	✓
2015	Bearman [99]	2D	images	Full	✓
2016	Guo [97]	2D	images	Full	✓
2017	This research	2D	images	Full	No
2013	Sikandar [100]	3D	Videos	<Both>	No
2016	Vasileios [101]	3D	Videos	Full	No

Table 3.2: Related work that suffers from the CBS problem

CHAPTER FOUR

PROPOSED APPROACH

In this research, I will propose a generic solution to solve the Confusion of Body Sides (CBS) problem in 2D Human Pose Estimation algorithms surveyed on section 3.2. The proposed approach consists of four main components: 1) Human Body Detection; 2) Human head detector; 3) Human Face Pose Estimation to be used as a face verifier 4) 2D HPE algorithm that is insensitive to the human viewing angle; The proposed approach is employed to solve the CBS problem and hence give very accurate human body parts localization and Pose Estimation in 2D images.

This chapter is organized as follows: On section 4.1 I will explain the proposed approach and its components; section 4.2 explains the evaluation methodologies we use to compare the results of different experiments conducted in chapter six. Finally, section 4.3, defines the used software libraries used in the proposed approach.

4.1 The Proposed System Architecture

According to the conducted research on the 2D HPE methods surveyed in chapter 3, one can note that many 2D HPE are insensitive to the human viewing angle. Because the 2D image viewing point is not taken into consideration, the algorithm suffers from the CBS problem. Therefore, our approach proposes a new methodology to tackle the viewpoint changes in 2D images when estimating the human pose. The proposed system consists of four main components as shown in Fig. 4.1.

4.1.1 Human Body Detection

As a first step, we need to locate the human in the 2D image. To do so, we have three options: 1) Use a separate human detection algorithm; 2) Train our own Human

Detector, or 3) Use the human detection that comes with the 2D HPE algorithm (if it has one). We have chosen the third option in our experiments for two main reasons: 1) to avoid adding time and space complexity on the system. 2) Our scope is not to provide the best accuracy in human detection; rather, it is to test the effectiveness of our proposed architecture with the minimal number of additional components.

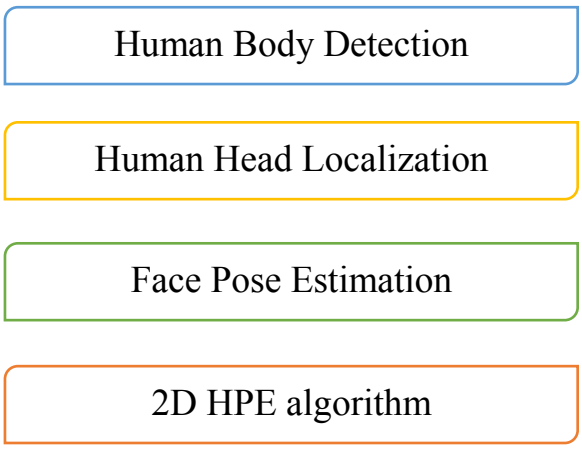


Figure 4.1: System Components for Viewpoint-Invariant 2D HPE

4.1.2 Human Head Localization

This stage takes input from the previous stage and feeds the output to Face Pose Estimation stage. For this component, we also have a design choice of 1) use an external Human Head Localization algorithm, such as VGG head detector by [102] or the SoA CNN head detector [103], 2) Train our own Human Head Detector, or 3) Use the human head localization that comes with the 2D HPE algorithm (if it has one). We chose the third option for the reasons discussed in section 6.9.5. The internal head detector in the PSM uses Histogram of Oriented Gradient (HoG) features and Support Vector Machine (SVM) as a classifier. The final human pose is highly dependent on this step. Therefore, it is better to perform single experiments to test the accuracy of the 'Head Localization' component before using it in the proposed system.

4.1.3 Face Pose Estimation

This stage takes input from the previous stage and feeds the output to the HPE algorithm. Also, the final pose is highly dependent on the previous component and on this component. Therefore, we will conduct a set of experiments first on this component before inserting it into the pipeline. Because nearly all 2D HPE datasets don't come with face annotation ground truth, we will label 3 different datasets to be able to evaluate this component. In this component, we chose to use 'Viola-Johns' Object Cascade detector as an external Face Detection algorithm due to the reasons discussed on session 6.12.5. It is used as a face verifier then we input the results to the next stage.

4.1.4 2D HPE algorithm

To provide a fair evaluation, we should first re-implement a prominent 2D HPE algorithm to be used as a baseline. We chose the 2D HPE algorithm of Yang and Ramanan [3] to improve its accuracy due to the reasons discussed in section 6.1.2. They use the Pictorial Structure Model (PSM) surveyed in section 3.1. They use HoG features to encode human appearance and a dedicated SVM classifier for each body part. Then, we will replicate its results on the author's data set using their evaluation method. Afterward, as shown in Fig. 4.2, we will run the baseline vs. [baseline +CBS solver] in a pipelined architecture on three different datasets to evaluate the final system.

In the pipelined architecture in Fig. 4.2, there are two main preprocessing stages. The first one is ***'Scale down the resolution'***. That preprocessing stage reduces the input image resolution to be of lower size e.g. 220x220 pixel. This stage is necessary to decrease the system running time; however, skipping this stage will not affect the accuracy for high resolution images. The second one is ***'Scale up the head patch'***. The extracted head patch is then scaled up by 4.0 bicubic interpolation. This preprocessing stage is necessary because 1) we want to look for faces of size 60x60 pixels in that

patch; 2) working with very lower resolution of head patches decrease the chance of finding faces in the head patch.

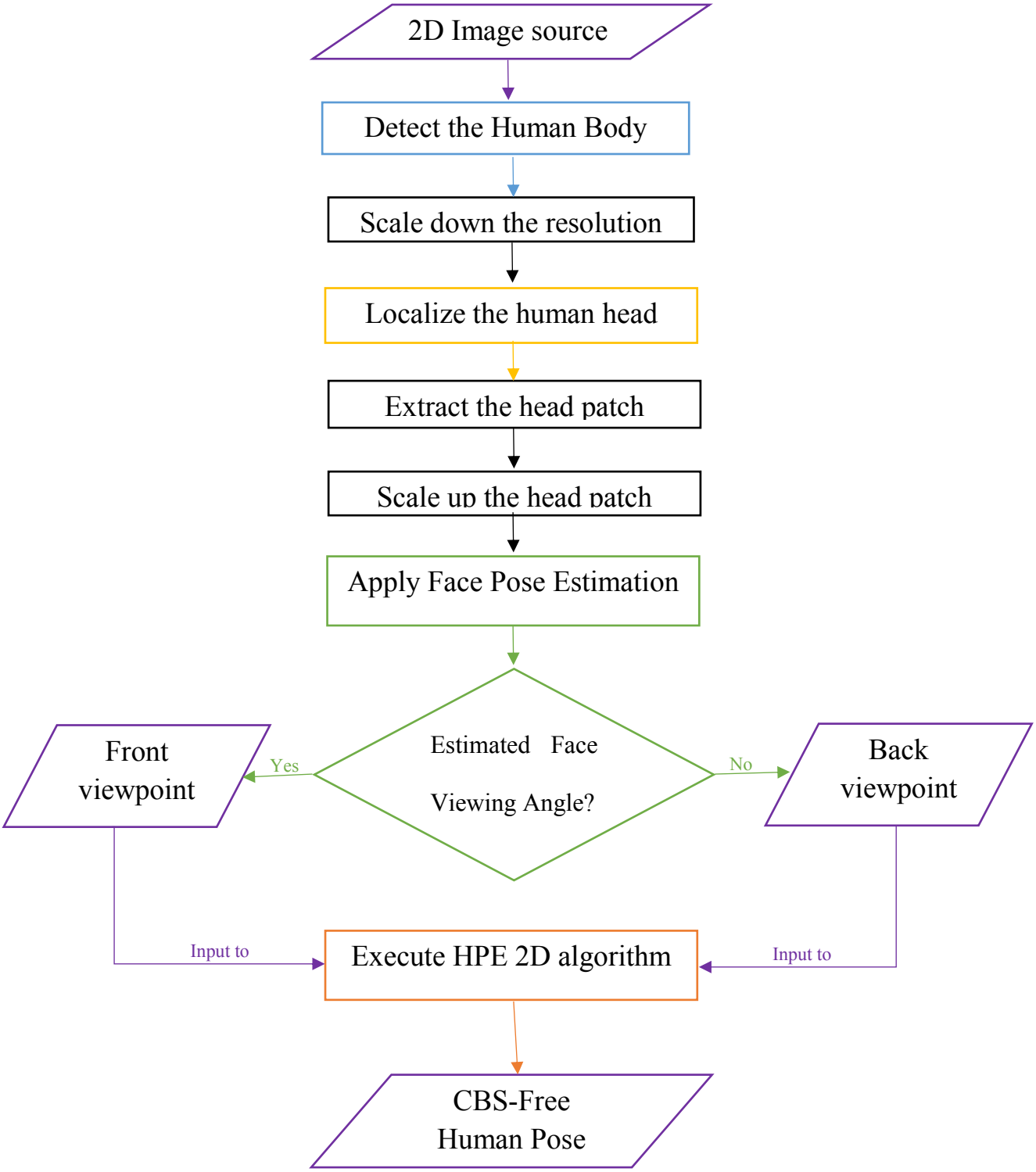


Figure 4.2: Processing Pipeline of SHAPE [2D HPE + CBS solver]

4.2 Evaluation Methodology

In this section, I will explain 3 Methods used in evaluation: 1) Confusion Matrix; 2) Percentage of Correctly estimated body Parts (PCP) which is used in many 2D HPE approaches in the literature; 3) Probability of Correct Keypoint (PCK). Finally, I will give a detailed example on how to compute the PCK evaluation.

4.2.1 Confusion Matrix

We use confusion matrix [104] in the evaluation of experiments 10, 11 and 12.

$$Sensitivity = \frac{TP}{TP + FN} \quad (1)$$

$$Specificity = \frac{TN}{TN + FP} \quad (2)$$

$$Precision = \frac{TP}{TP + FN} \quad (3)$$

$$Recall = \frac{TP}{TP + FP} \quad (4)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

All of these measures depend on the following four parameters:

TP: positive-labeled samples that were correctly classified.

TN: negative-labeled samples that were correctly classified.

FP: negative-labeled samples that were incorrectly classified.

FN: positive-labeled samples that were incorrectly classified.

	Predicted Negative	Predicted Positive
Actual Negative	TN	FP
Actual Positive	FN	TP

Table 4.1: Calculation of Confusion Matrix

4.2.2 PCP Evaluation

In this section, I will briefly mention a common evaluation criterion introduced in the Buffy Stickmen dataset [71], which is called ***P**ercentage of **C**orrectly estimated body **P**arts* (PCP). Afterward, I will discuss in details the evaluation criterion used to measure the accuracy of the experiments in this research. The evaluation method is called ***P**robability of **C**orrect **K**eypoint* (PCK) and it was first introduced by [17].

In PCP [71], an estimated body part is counted as correct if its segment endpoints lie within t% of the length of the ground-truth segment from their annotated location. This means a body part is considered correctly localized if its endpoints are closer to their ground-truth locations than a threshold (on average over the two endpoints). In [71] the authors recommend taking PCP at t=20% (strict) or t=50% (tolerant, this is the setting set *by default*). In other words, an estimated body part is labeled as correct if its segment endpoints lie within **50%** of the length of the ground-truth annotated endpoints.

Because of the aforementioned tolerance of the threshold, [17] concluded that PCP criterion was clearly crucial and influential in quantitative evaluation. Also, PCP is sensitive to the amount of foreshortening of a limb, and hence can be too loose a measure in some cases and too strict a measure in others. At last, PCP requires candidate and ground-truth pose to be placed in correspondence but does not specify how to obtain this correspondence. Therefore, in this research, I am evaluating the experiments using PCK evaluation method.

4.2.3 PCK Evaluation

PCK means Probability of Correct Keypoint. In PCK [17], given a bounding box of the human in the 2D image, a pose estimation algorithm must report the

estimated key point locations for body joints. An estimated key point is considered to be correct, i.e. true positive, if the key point falls within ***alpha* (*α*) * *scale* (*s*)** pixels of the ground truth (GT) key point, where ***α*** is a *threshold*, and the *scale* (*s*) is the maximum of the height (***h***) and the width (***w***) of the human bounding box respectively:

$$f(x) = max(h, w) \qquad (6)$$

This means that the PCK measure considers a detection as a correctly localized body joint if the *Euclidian* distance (***d***) between the detected position and the ground truth position is less than or equal to alpha **(*α*) *** scale **(*s*)** pixels.

$$f(x) = \begin{cases} 1, & d \leq \alpha * s \\ 0, & otherwise \end{cases} \qquad (7)$$

Therefore, the following are needed to obtain the PCK evaluation:

1. The ground truth key point of the body joint.
2. The detected key point of the body joint.
3. The threshold ***alpha*** (***α***).
4. The bounding box of the person to get the ***scale* (*s*)**.
5. The distance **(*d*)** between the GT key point and the estimated key point.

The first requirement, which is the ground truth location of the body joints, is given through manually annotating fourteen joints of each human in the image. Also, **the second requirement,** which is the detected locations of the body joint, is reported back by the human pose estimation algorithm.

The third requirement is the threshold ***alpha*** (***α***). ***α*** controls the relative threshold for considering correctness. Varying the PCK threshold corresponds to varying the desired accuracy. The less the value of the threshold the more strict the evaluation is. Hence, more accurate results could be obtained. For instance, when ***α*** = 0.1, the

evaluation is very strict, whereas when α = 0.9 the evaluation is very tolerant. In this research, I use α = 0.1 on both the 'Image PARSE' and 'Humans AUC' dataset evaluation.

The fourth requirement is to have a bounding box on the human in the image. This is needed to get the ***scale***. The scale is the maximum value of the bounding box width (***w***) and height (***h***). When the bounding box location is not available, we can infer it from the ground truth of the body joints as shown in Fig. 4.3.

The fifth requirement is the distance between the estimated key point and the ground truth key point. In PCK, the *Euclidian* distance is calculated to measure the distance between two points. Hence, when the Euclidian ***distance*** (***d***) is less than or equal ***thresh * scale***, the estimated joint location is deemed true.

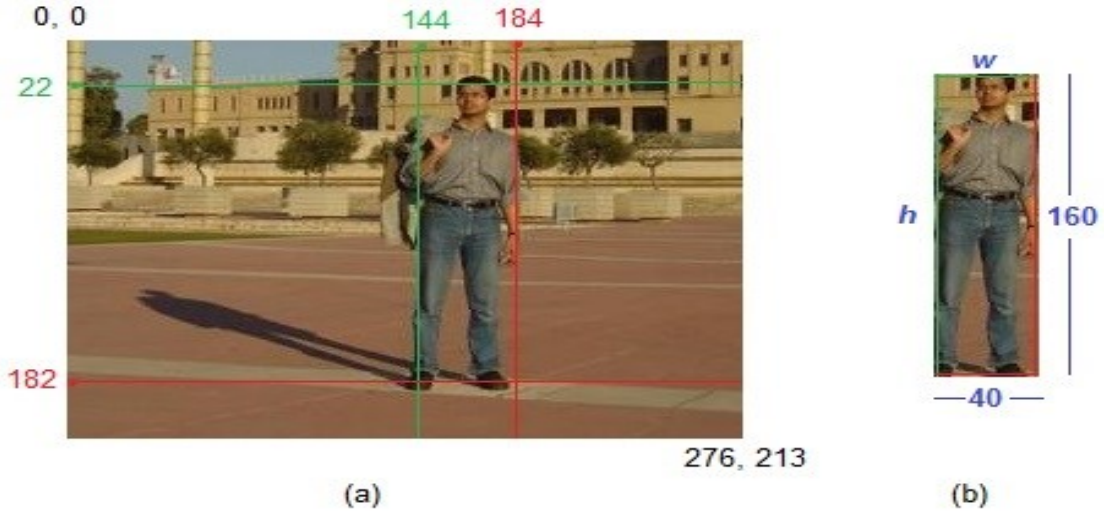


Figure 4.3: Obtaining the human bounding box from the ground truth

(a) A 2D image contains a human without a bounding box [2].
(b) Extracting the bounding box based on the joints' ground truth.
*The image is extracted from PARSE dataset number im0102.

Fig 4.3 illustrates how to calculate the bounding box of the human given the ground truth key points of the body joints. There was no bounding box on the human in Fig. 4.3 (a). However, since we have the XY location of the fourteen human joints {Head, Neck, Right Shoulder, Right Arm, Right Wrist, Left Shoulder, Left Arm, Left

Wrist, Right Hip, Right Knee, Right Ankle, Left Hip, Left Knee, Left Ankle} as ground truth data, we can infer the bounding box of the human.

For example, when we search for the greatest value of X and the greatest value of Y in all key points, we get 184 and 182 respectively. Likewise, when we search for the minimum values of X and Y, we get 144 and 122 respectively. To obtain the bounding box of the human body, and accordingly the size (h, w) of the bounding box, we subtract the great X from the minimum X to obtain the width 184-144 = 40, and the greatest Y from the minimum Y to obtain the height 182-22= 160. Thus, it is easy to get the largest value of the height and the width. In this case, the ***scale*** = *max*(160, 40) = 160.

4.2.3.1 Computing Euclidian Distance

In PCK evaluation criterion, *Euclidian* distance is calculated to measure the distance between two points. *Euclidian* distance can be obtained by calculating the Sum of Squared Differences (SSD) between the two points in the 2d image.

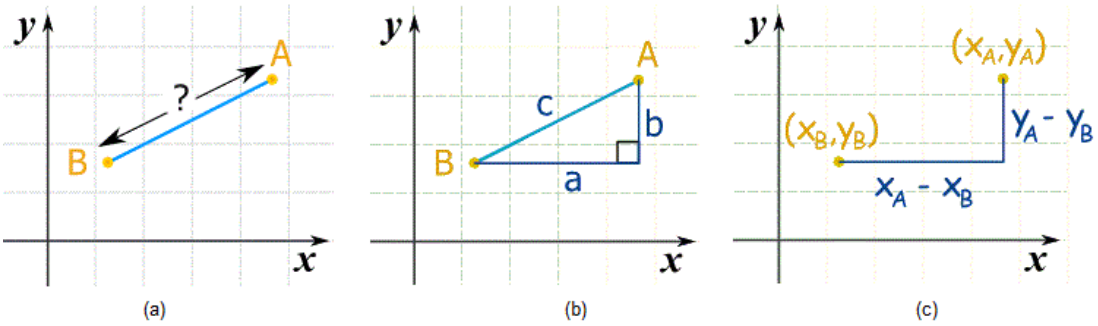


Figure 4.4: Calculation of Euclidian Distance

(a) Two point A and B with unknown distance.
(b) Drawing a right angled triangle between the two points
(c) Calculating the two other sides of the triangle.

Let us call the two points A and B as shown in Fig. 4.4 (a). We can run lines down from A, and along from B, to make a right angled triangle as shown in Fig. 4.4 (b). We know that: $C^2 = A^2 + B^2$ from Pythagoras. Now label the coordinates of points A and B. as shown in Fig 4.4 (c).

- X_A means x-coordinate of point A.
- Y_A means y-coordinate of point A.
- The horizontal distance **a** is ($X_A - X_B$).
- The vertical distance **b** is ($Y_A - Y_B$)

Now we can solve for **c** (the distance between the points):

$$C^2 = (X_A - X_B)^2 + (Y_A - Y_B)^2 \quad (8)$$

$$C = \sqrt{(X_A - X_B)^2 + (Y_A - Y_B)^2} \quad (9)$$

4.2.4 Computing PCK

As discussed in section 4.2.3, PCK evaluation needs ***five*** requirements: the ground truth position of joints, the detected position reported by the HPE algorithm, the threshold alpha, the scale, and the Euclidian distance between the ground truth position and the detected position. In the example shown in Fig. 4.5, there are two joints that we want to evaluate their results using PCK, which are: the left knee (called the right knee in PARSE dataset), and the left wrist (called the right wrist in PARSE dataset) [2].

First, the ground truth position values for 'left knee' and for the 'left wrist' are (170, 143) and (180, 115) respectively. They are colored in two green dots. ***Second,*** the detected positions reported by the HPE algorithm for the 'left knee' and for the 'left wrist' are (168, 136) and (181, 90) respectively. They are colored in two red dots. ***Third***, I always evaluate using the threshold alpha ***α = 0.1*** (strict evaluation). ***Fourth,*** the scale ***s*=160**, which is the maximum of the bounding box height and width as shown in Fig. 4.3 (a). ***Fifth,*** the *Euclidian* distance (SSD) between the ground truth position and the detected position of

the ‘left knee’ and for the ‘left wrist’ are 7 pixels and 25 pixels respectively. Now we are ready to compute the PCK.

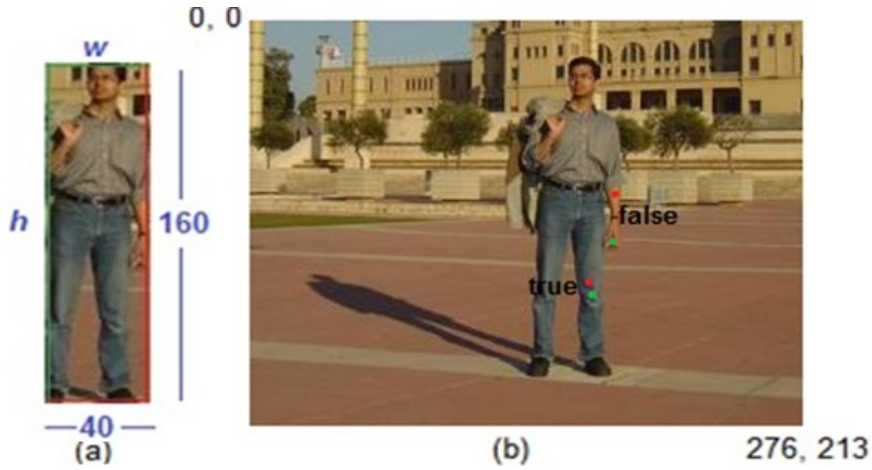


Figure 4.5: Computing PCK evaluation criterion
(a) The human bounding box inferred in Fig 4.3.
(b) The ground truth points and the detected points for 2 human joints.
*The image is extracted from PARSE dataset number im0102.

As discussed in section 4.2.3, the detected body joint is considered to be true positive if the distance between the ground truth position and the detected position is less than or equal a certain number of pixels as denoted by $f(x) = \begin{cases} 1, & d \leq \alpha * s \\ 0, & otherwise \end{cases}$

In Fig. 4.5 (b), the maximum allowed distance = **16** pixels (0.1 * 160). This means that the joint is considered successfully localized if the computed SSD is below or equal ***16*** pixels. This is more accurate than PCP [71] because the maximum allowed distance is computed for ***each*** human in the image depending on the bounding box size.

Hence, the ‘left knee’ detected point is considered true positive whereas the ‘left wrist’ detected point is considered false positive. When $\alpha = 0.1$, eleven joints were classified as true positive, i.e. their SSDs were less than or equal to 16 pixels while only three joints were classified as false positive, and they are 'left wrist', 'left elbow', and ‘right wrist'. When $\alpha = 0.2$, the maximum allowed distance will be 32 pixels (0.2*160) therefore we have only one false positive which is the ‘right wrist’ (SSD=40). In all experiments, I chose $\alpha = 0.1$.

4.3 Software libraries used

4.3.1 MATLAB 9.2

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment that allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python.

4.3.2 Open CV 3.0

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel. The library is cross-platform and free for use under the open-source BSD license.

4.3.3 CUDA 8.0

CUDA is a parallel computing platform and application programming interface (API) model created by Nvidia. It allows software developers and software engineers to use a CUDA-enabled graphics processing unit (GPU) for general purpose processing.

4.3.4 Visual Studio 14.0

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web apps, web services and mobile apps. It can produce both native code and managed code.

CHAPTER FIVE

PROPOSED DATASET: HUMANS AUC

In this chapter, I will discuss another contribution of this research which is the 'Humans AUC' dataset. First, I will mention some of the previous work in 2D human pose datasets. Afterward, I will discuss the dataset specifications, hardware and software used, video synchronization and annotation process, and the camera calibration setup orderly.

5.1 Related Work

There exist many 2D human pose estimation datasets in the field. Some focus on the 'Upper-Body' such as the Buffy Stickmen dataset [105] while others focus on the 'Full-Body' such as Leeds Sport Pose dataset (LSP) [106]. In table 4.1, I have summarized some of the widely used datasets by the human pose estimation approaches. The last row is the 'Humans AUC' dataset presented in this research.

Dataset	Video/ images	Type	Dim	Viewing angle	Single Person
MPII Human Pose [107]	Images	Full	2D	Monocular	<Both>
Buffy Stickmen [105]	Videos	Upper	2D	Monocular	✓
PASCAL Stickmen [108]	Images	Upper	2D	Monocular	✓
Synchronic Activities [109]	Images	Full	2D	Monocular	Multi
FLIC-motion [110]	<Both>	<Both>	2D	Monocular	Multi
Parse [111]	Images	Full	2D	Monocular	✓
Leeds Sports Pose [106]	Images	Full	2D	Monocular	✓
Human Pose in Wild [112]	Images	Upper	2D	Monocular	✓
Humans AUC Dataset	*<Both>*	*Full*	*2D-3D*	*Monocular-Multi-view*	*<Both>*

Table 5.1: 2D Human Pose Estimation Datasets

5.2 Dataset Specifications

In the ‘Humans AUC’ dataset, we present a set of video files that were recorded for 70 human participants. We have categorized them randomly across 17 groups. So, each group has three, four, or five volunteers. For each group, we ask them to do five scenarios while the cameras were recording. Four cameras were mounted in a laboratory room at height of roughly four meters. A fifth camera is used on front of the laboratory door, but it was not used in the experiments since the four cameras cover the four views well.

It would be a good idea to mount the cameras at various heights to capture different out of plan views. However, in some HPE datasets the cameras were mounted at fixed height to increase the overlap area as much as possible. An overlap area is where the entire body of the human body is visible in all mounted cameras. The overlap area is our setup is almost *one* meter square. There is only one entrance and two exits. We call the entry and the exit points: Entry 1, Exit 1 and Exit 2. We have been granted the permission by the Intuitional Review Board (IRB) at the American University in Cairo to start collecting the video samples of human participants.

The ‘Humans AUC’ dataset consists of 425 video files. The length of each video is roughly 30 to 90 seconds. The frame rate is 30 frame per second (FPS). The dataset videos are in the AVI file format. Most of the actions performed by the human participant are walking and sitting actions. Frames were extracted from the video files using the AUC annotation tool. For example, 425 frames were extracted from group 13 and used in chapter six.

5.3 Hardware Used

We have captured the frames using five High Definition (HD) cameras of type Bosch DINION IP dynamic 7000 HD with Power over Ethernet (PoE) feature. We used D-Link 8 ports Gigabit Web Smart PoE Switch. The lens used with these cameras is of type Ricoh FL-HC6Z0810 8-48mm. We have used 4 laptops of different specifications, such as core i3, core i5, and core i7 Intel processors. That is only to distribute the recording load so that we can guarantee we don't miss out frames. In order to synchronize the four cameras we must record all videos with the same FPS. Recording 4 cameras using one computer makes this FPS very difficult to achieve. Each laptop captures one video at a time so that no lag happens between the frames in the recorded video file, and hence, no frames will be dropped (See Fig. 5.1).

Figure 5.1: The environment setup shows a live feed from the 4 cameras

5.4 Software Used

We first experimented with the VLC player. Two main problems were encountered when using the VLC player and recorder. These problems are: first, the inaccurate functionality of setting the frame per second when recording. Second, the lag that is obvious between video frames although all hardware resources were available to the VLC player. Therefore, we have used a software by BOSCH to acquire image frames. It is called BOSCH Video Client version 1.7.1 and it works under Windows. Therefore, we have collected the dataset on Windows 10.

We have used a software called Format Factory. Format Factory is a free and multifunctional, multimedia file conversion tool. We have converted the entire dataset from the uncompressed MPG format to the AVI format, keeping the same FPS and the frame apparent quality. This process has reduced the size of a single video file from ~240 MB to only 40 MB, which is roughly 86% decrease in size. After converting the entire dataset, the 425 files are of size 12 GB. One last problem is that the 5 videos that were taken for each scenario are not synchronized, which is the issue to be discussed in the following section.

5.5 Video Synchronization

The dataset contains 425 video files. Each scenario was recorded using four concurrent, but not synchronized, cameras. It is important to synchronize the video files of each scenario in case of 3D Human Pose Estimation, but in 2D Human Pose Estimation, it is not required to synchronize the cameras. Therefore, in this research, there is no need to synchronize the video files but we are providing synchronous video files in this dataset to be useful to other domains as well.

Synchronizing the video files means that a frame F1 taken from video 1 (captured by Cam1) should be equivalent to frame F2 (captured by Cam2) at a given time t. For example, frame number 100 in the video sequence of Camera 1 should correspond to frame 100 in the video sequence of Camera 2 with no delay, and so on.

Two main challenges we faced to synchronize each 5 video files of each scene together. The first challenge is that, we should make sure of the five recorded video were captured and recorded at the same frame rate. Not having the same rate would make it very difficult to synchronize the video files. This challenge was resolved by using the BOSCH Video Client software and by assigning one computer to each camera. This was necessary because making one computer records data from 2 or 3 cameras lower the frame rate.

The second challenge is that the need for a synchronize method. Two methods were proposed, the first one is by using an audio signal. That is by using microphones for each camera then use a clapperboard to make a common start point for all the five cameras. However, the HD BOSCH DINION 7000 cameras did not contain build in microphones. Therefore, we resorted to the second method.

The second method is by using a visual signal. For example, using a laser pointer on the overlap area which is an area of size 1x1 meter that is visible by all of the four cameras that record the human participant from the different four angles. The BOSCH cameras, however, were not able to detect the wavelength of the laser pointer. Accordingly, we have resorted to another visual signal. That was by turning off the light then turning it back on again while the 5 cameras are recording and right before the volunteer enters through Entry 1. That would work as a visual mark on all videos for later processing.

After we finished recording the 17 groups with placing the visual marks on each scenario, we needed to manually synchronize each 5 videos together based on the visual mark we set earlier before the volunteer enters the room. The manual synchronization idea is applied by removing all the frames from the starting frame, which is frame zero, to the frame at which the light started to appear again. This process has been done manually to each of the 425 video files because we needed to search through each video frame by frame for our visual mark.

We used Adobe Premiere version 6.0 to provide us with three tasks: 1) Search in the video sequence frame by frame for the visual marker, 2) Remove the frames from frame 0 to the frame at which the visual signal appears, 3) Save the new processed video after cropping the starting frames. The only drawback is that Adobe Premiere increased the size of the input video file from ~40 MB to almost ~100 MB. That increased the dataset from 12 GB to about 25 GB. We have used Format Factory again to reduce the dataset size. ‘Humans AUC’ dataset final size was nearly 19 GB, and it was organized in 17 groups each group contains 5 scenarios, and each scenario has 5 synchronized video files. Annotating the frames is discussed in the following section.

5.6 Frame Annotation

We have developed a tailored utility to help us to annotate the 'Humans AUC' dataset. We call the utility: AUC Annotation Tool version 6.0 (see Fig. 5.2).

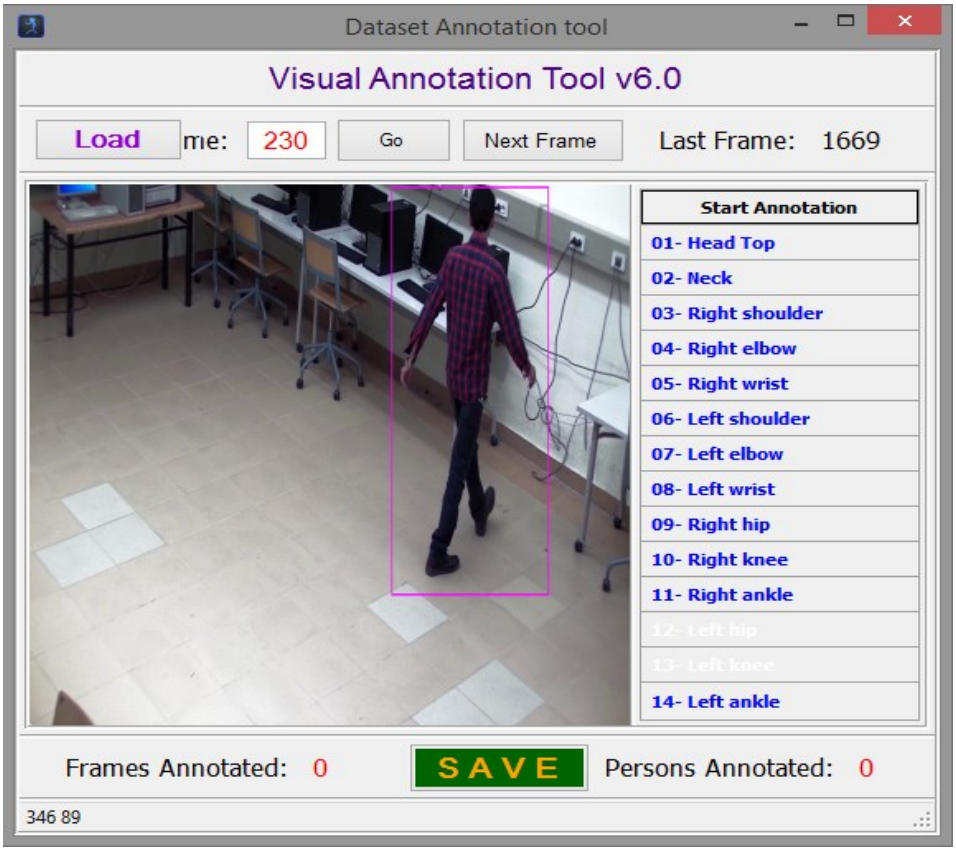


Figure 5.2: AUC Visual Annotation Tool version 6.0

The annotation tool has an easy-to-use Graphical User Interface, and it provides many functionalities. The annotation tool provides the ability to set 14 joint locations of the human body in the following order: head, neck, right shoulder, right elbow, right wrist, left shoulder, left elbow, left wrist, right hip, right knee, right ankle, left hip, left knee, left ankle. Many features were included as well.

In addition, we have developed another tool to do some preprocessing on the annotated ground truth before using them in any testing or evaluations. We called this tool AUC Pre-processing Batch Tool 5.0 as shown in Fig. 5.3.

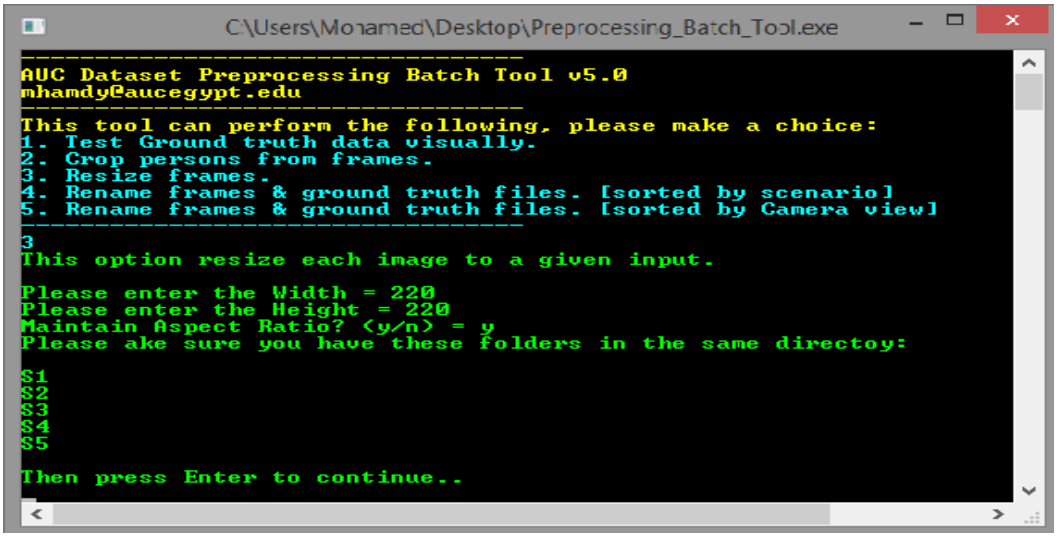


Figure 5.3: AUC Preprocessing Batch Tool version 5.0

The Batch Pre-processing tool is a console application that provides the following functionalities on the annotated ground truth images: 1) See the imposing of the ground truth limb locations on the image frame; 2) Crop volunteers from frames; 3) Resize images to the preferred width and height with or without maintaining aspect ratio; 4) Rename all frames and sort them by scenario; 5) Rename all frames and sort them by Camera view.

Using this small batch tool, we can manipulate the images frames with the ground truth and produce any size of human participants in the dataset. There is a batch script we have written in Matlab to parse these text files into Matlab ground truth according to ***any*** specific order.

5.7 Camera Calibration

Camera calibration is the first step towards computational computer vision. Even though some information concerning the measuring of scenes can be obtained by using uncalibrated cameras [113], calibration is essential when metric information is needed. The use of precisely calibrated cameras makes the measurement of distances in a real world from their projections on the image plane possible [114]. In 2D Human Pose Estimation, it is not required to calibrate the cameras. Therefore, in this research, there is no need to calibrate the cameras but we are providing calibrated camera parameters in this dataset to be useful to other domains as well.

We are providing the images that can be used to compute the camera calibration parameters for each camera. We are also providing the camera calibration parameters for each camera.

Figure 5.4: Checkerboard used in camera calibration

We used Matlab to do the camera calibration and to provide the camera parameters. Two requirements are needed to compute the camera parameters successfully. First, we need to capture at least 20 images of something that is easy to

be recognized like the checkerboard shown in Fig. 5.4. Second, we have to input the size of the checkerboard square in the real world so that the camera calibration can compute the distance from the camera and calculate the extrinsic and the intrinsic camera parameters. We have provided about 50 images taken by each camera, and the size of the checkerboard square in real world is 10 centimeters

After performing the camera calibration process, we will have the camera extrinsic and intrinsic parameters. Intrinsic parameters are the internal camera specifications like the focal length. The extrinsic parameters are like the rotation matrix. Knowing these parameters, we will be able to determine how far an object is from the camera. As shown in Fig. 5.5, the calibration process were able to plot where the camera is with respect to the 40 images that were taken for the checkerboard. The figure below shows camera 2 at the corner and the orientation of the checkerboard in each of the 40 images.

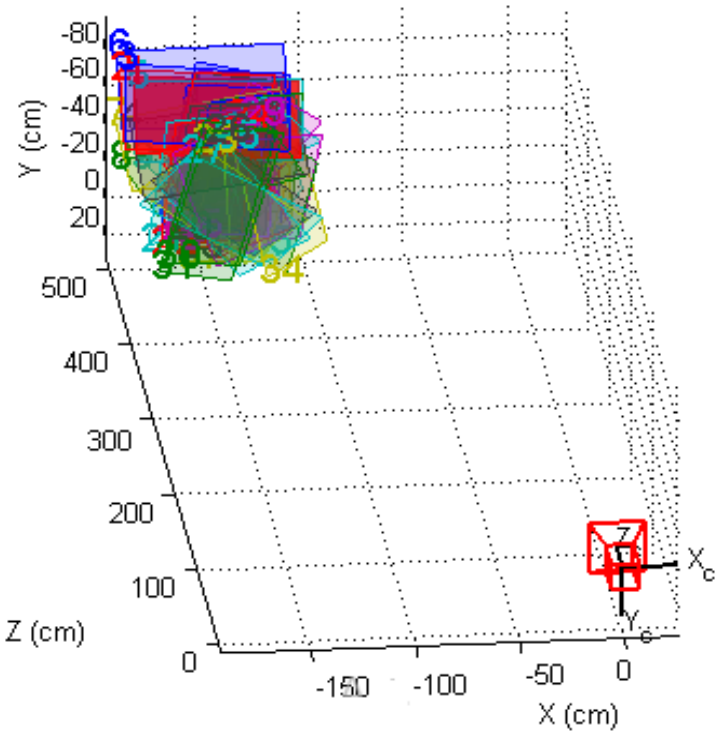


Figure 5.5: Camera calibration (Extrinsic Parameter Visualization

CHAPTER SIX

EXPERIMENTAL RESULTS AND EVALUATION

In this chapter, I will discuss 15 experiments categorized into three sets of experiments. I will demonstrate the results of building, testing, running, and evaluating a baseline approach in the experiments from 1 to 6. Afterward, I will discuss the conducted work of our proposed approach in experiments from 7 to 12. Finally, I will evaluate the final results of our proposed solution to the CBS problem in experiments from 13 to 15. Each experiment will have its objective, methodology, results, and discussion sections.

6.1 Experiment 1: Evaluating a 2D HPE Baseline

6.1.1 Objective

We need to choose a notable 2D HPE algorithm, re-implement it, replicate the author's results, and show qualitatively and quantitatively that it suffers from the confusion of Body Sides problem discussed in section 1.2.

6.1.2 Methodology

First, we need to find a notable 2D HPE approach, rebuild it, then test it qualitatively. Next, we want to reproduce the quantitative results of its authors using their dataset and their evaluation methodology (PCK). Hence, from the 2D HPE approaches surveyed on section 3.2.2, we chose the work of Yang and Ramanan [3]. They estimate the human pose in 2D images using the Pictorial Structure Model (PSM) discussed in section 3.1. Their approach was reported by a recent survey [24] to be the SoA approach for 2D HPE.

6.1.3 Results

Figure 6.1: Qualitative results of 2D HPE baseline [3] on random test images

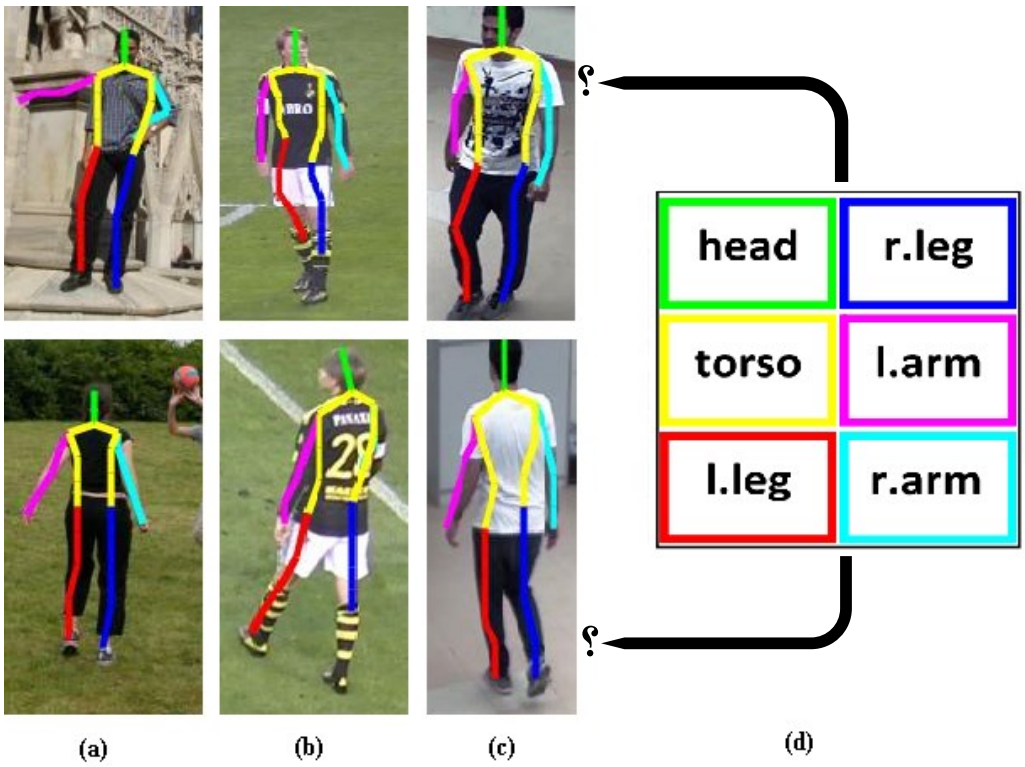


Figure 6.2: Implemented baseline [3] suffers from the CBS on three datasets
(a) Estimating pose on two images taken from 'Image PARSE' dataset.
(b) Estimating pose on two images taken from 'KTH Multiview Football' dataset.
(c) Estimating pose on two images taken from 'Humans AUC' dataset.
(d) Color legend of baseline [3] extracted from their paper [17].

Dataset: Image PARSE [205 images] Image Size: 150x150							
Points	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle
Accuracy	89.3	84.4	67.3	46.6	76.1	74.1	66.1
	Mean PCK		Total Time		Time/Image		
	72.0 %		7.0 min		2.1 sec		

Table 6.1: Our Quantitative Results of Baseline [3] on PARSE

6.1.4 Discussion

We have retrained the PSM model in baseline approach [3], then we tested it on a random set of 10 human images as shown in Fig. 6.1 (a). Because each body part has its own SVM classifier, the algorithm detects all the different body parts and mark a bounding box on the detected part as shown in Fig. 6.1 (b). After detecting all body parts, then a skeleton could be drawn based on the detected bounding boxes as shown in Fig. 6.1 (c).

The next step is to test the baseline approach with different viewpoints of humans to check if it suffers from the CBS problem. As shown in Fig. 6.2, the baseline approach is insensitive to the human viewpoint when estimating the human pose. Thus, it is incapable of correctly localizing the human body parts shown in the first row of Fig. 6.2 when compared to the second row. To elaborate, we cannot match the legend in Fig. 6.2 (d) to both rows of Fig. 6.2 at the same time. That means that the baseline approach confuses the left body side with the right one due to the symmetrical structure of the human body. Hence, the approach in [3] suffers from the CBS problem.

Afterward, we needed to reproduce all the results Yang and Ramanan have obtained on their dataset 'Image PARSE' using the same evaluation metric: Probability of Correct Keypoint (PCK) (see section 4.2.3 for more details). As reported in [3], the authors scored a total accuracy of 72.9% on a subset of Parse dataset. Particularly, they used images from 101 to 305 (205 images). In table 6.1, we show our obtained results on the same subset of the PARSE dataset. One can note that we scored a very close accuracy, which is **72.0%**. Now we have a running baseline that needs some enhancements.

6.2 Experiment 2: Speeding up the Baseline Approach

6.2.1 Objective

Speed up the human estimation process of the baseline approach by decreasing the time to estimate the human pose in a single image (Time/Image).

6.2.2 Methodology

The basic task which greatly consumes much time in detecting the human joint locations is the convolution process that takes place in the detecting function. I will utilize the Multi-threading convolution with SSE instruction presented by Ross Girshick in [115][116] instead of the basic sequential convolution.

6.2.3 Results

Dataset: PARSE [205 images] image size: 150x150							
Points	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle
Accuracy	89.3	84.4	67.3	46.6	76.1	74.1	66.1
Mean PCK		Total Time		Time/Image			
72.0 %		2.0 min		**0.6 sec**			

Table 6.2: PSM on PARSE [SSE]

6.2.4 Discussion

When using SSE convolution the time/image dropped drastically from 2.1 sec seconds as shown in table 6.1 to 0.6 seconds. That is nearly 4X the speed of the original baseline with basic convolution while maintaining the same accuracy. Therefore, all the following experiments will use SSE multi-threading convolution instead of the basic sequential convolution. For detailed PC specifications and software used to obtain these timings, please refer to section 6.16.

6.3 Experiment 3: Correcting the Baseline Ground Truth Labels

6.3.1 Objective

We discovered that the baseline approach by [3] suffers from the CBS problem in qualitative evaluation as shown in Experiment 1 Fig. 6.2. However, this degradation in accuracy is not shown in their quantitative results. So, we have analyzed their approach and found out that the ground truth labels of the 'Image PARSE' dataset are not consistent. This means that the order of human joints is not the same across the whole dataset. In 'Image PARSE' dataset, Ramanan uses one order for joints of people viewed from the front, and the opposite order for joints of people viewed from the back.

This provides unfair quantitative results if we want to differentiate between different human body sides. The objective here is to measure the *correct* accuracy with a *consistent* ground truth labels with one order across all images of the 'Image PARSE' dataset.

6.3.2 Methodology

We generated a consistent ground truth labels with one order along the whole ground truth labels. That is by reversing back the joints' order of people viewed from the back to have the same order to that of people viewed from the front. Then we test the baseline on PARSE dataset again using the corrected ground truth labels.

6.3.3 Results

Dataset: PARSE [205 images] image size: 150x150							
Points	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle
Accuracy	89.3	79.3	61.7	43.7	74.4	69.5	62.9
Mean PCK		Total Time		Time/Image			
68.7 %		2.0 min		0.6 sec			

Table 6.3: Baseline on PARSE with a Corrected Ground Truth Labels.

6.3.4 Discussion

In the 'Image PARSE' dataset [2], the author used the following order for the people viewed from front:

{lank, lkne, lhip, rhip, rkne, rank, lwr, lelb, lsho, rsho, relb, rwr, hbot, htop}

Whereas the author used the opposite order for the people viewed from the back:

{rank, rkne, rhip, lhip, lkne, lank, rwr, relb, rsho, lsho, lelb, lwr, hbot, htop}

Ordering the ground truth labels this way will trick the evaluation function in the baseline, which uses only one fixed order. It will make the baseline *mistakenly* not suffer from the CBS problem. For example, if the person is viewed from the back, the baseline evaluation will compare the estimated 'left ankle' location with the 'right ankle' location in the ground truth, and the baseline will mark this joint location correct if they match, which is inconsistent and imprecise.

I have set the ground truth order (with respect to the person in the image) to be: {rank, rkne, rhip, lhip, lkne, lank, rwr, relb, rsho, lsho, lelb, lwr, hbot, htop} for all images in the dataset, which is the same order of the baseline evaluation function. As expected, the accuracy dropped from **72.0**% to **68.7%** when we used a consistent ground truth for 'Image PARSE' dataset due to the fact that the baseline suffers from the CBS problem. That's the accurate accuracy that we should compare our results to. It worth mentioning that it only dropped few percentages because the dataset is not balanced. While the test set of 205 images contains roughly 187 images of people viewed from the frontal view, ***only*** 18 images contain people from the back view. We expect the accuracy to drop even more if the dataset is balanced and uses consistent ground truth label. Consequently, we have collected our own *balanced* dataset: Humans AUC to test the baseline approach on.

6.4 Experiment 4: The Baseline on 'KTH Multiview Football'

6.4.1 Objective

Report the accuracy of the baseline approach in [3] on another popular dataset in Human Pose Estimation that contains balanced data.

6.4.2 Methodology

We have run the baseline approach on 1000 images from a popular 2D HPE dataset from the literature which is called 'KTH Multiview football II' [1]. It contains 5907 annotated images from 3 different views of a single football player.

6.4.3 Results

Dataset: KTH [1000 images] image size: 250x250							
Points	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle
Accuracy	92.3	45.3	34.4	27.0	44.7	39.6	26.7
Mean PCK		Total Time		Time/Image			
44.3 %		15.5 min		0.9 sec			

Table 6.4: Baseline on KTH Multiview Football Dataset

6.4.4 Discussion

Three main reasons are behind choosing this dataset: 1) Unlike 'Image PARSE' dataset, it contains balanced data from 3 different views. 2) Unlike 'Image PARSE dataset', this dataset's ground truth is consistent. i.e., the player's right leg remains his right leg in the ground truth annotation regardless of the viewing angle, which is precise. 3) It uses the same order of annotation we use in the baseline evaluation, which is: {rank, rkne, rhip, lhip, lkne, lank, rwr, relb, rsho, lsho, lelb, lwr, hbot, htop}

It's important to note that we do not expect the accuracy to exceed 70s % since we are bound by the accuracy of the baseline approach itself as shown in table 6.1. However, as shown in table 6.4, the accuracy is 44.3%. That's because the baseline approach suffers from the CBS problem as demonstrated previously in Fig. 6.2 (b).

6.5 Experiment 5: The Baseline on 'Humans AUC' [All views]

6.5.1 Objective

Report the accuracy of the baseline approach in [3] on our challenging dataset 'Humans AUC' proposed in Chapter 5.

6.5.2 Methodology

We have run the baseline approach on a test set of images from 'Humans AUC' dataset. The dataset has been made balanced in order to provide a fair evaluation. Hence, we test on 425 annotated images from 4 different views, each view contains roughly one hundred test image. We use the same Ground Truth order of previous tests: {rank, rkne, rhip, lhip, lkne, lank, rwr, relb, rsho, lsho, lelb, lwr, hbot, htop}

6.5.3 Results

Dataset: Humans AUC [425 images] image size: 220x220							
Points	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle
Accuracy	95.5	39.1	34.6	31.9	47.9	47.8	46.4
Mean PCK		Total Time		Time/Image			
49.0 %		3.7 min		0.5 sec			

Table 6.5: Baseline on 'Humans AUC' Dataset [All Views]

6.5.4 Discussion

We do not expect the accuracy to exceed 70s % since we are bound by the accuracy of the baseline approach itself as shown in table 6.1. However, as shown in table 6.5, the accuracy reached only 49.0%. One can note that while the accuracy of estimating the head location remains quite satisfying, all the other body parts have very low accuracy. That is because the baseline approach is insensitive to the viewing angle of the person in the image. For example, the left hand is sometimes recognized as the right one, and so on. This sometimes results in confusing the entire left body side with the entire right body side as shown previously in Fig. 6.2 (c).

6.6 Experiment 6: The Baseline on 'Humans AUC' [Single View]

6.6.1 Objective

Report the accuracy of the baseline approach in [3] on each view independently. That is to analyze the situations in which the baseline suffers from the CBS problem.

6.6.2 Methodology

We have divided the 425 images into 4 groups according to the camera view. Then we ran the baseline approach on each view in a single experiment.

6.6.3 Results

Figure 6.3: Qualitative Results of the Baseline on Each View of 'Humans AUC'

(a) Estimating the pose on 5 actors viewed from Camera 1.
(b) Estimating the pose on 5 actors viewed from Camera 2.
(c) Estimating the pose on 5 actors viewed from Camera 3.
(d) Estimating the pose on 5 actors viewed from Camera 4.

Figure 6.4: Human Body Joints Color Legend of Baseline as in [17]

Dataset: Humans AUC [83 images] | image size: 220x220 | Camera One							
Points	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle
Accuracy	98.2	71.7	63.9	57.8	74.1	75.3	69.9
Mean PCK		Total Time		Time/Image			
73.0 %		0.6 min		0.5 sec			
Dataset: Humans AUC [91 images] | image size: 220x220 | Camera Two							
Points	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle
Accuracy	98.9	90.1	74.7	69.2	78.0	73.1	59.9
Mean PCK		Total Time		Time/Image			
77.7 %		0.7 min		0.4 sec			
Dataset: Humans AUC [130 images] | image size: 220x220 | Camera Three							
Points	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle
Accuracy	98.5	12.3	12.7	9.6	25.4	27.7	34.6
Mean PCK		Total Time		Time/Image			
31.5 %		1.0 min		0.5 sec			
Dataset: Humans AUC [121 images] | image size: 220x220 | Camera Four							
Points	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle
Accuracy	88.0	7.0	7.9	9.9	31.4	31.4	32.6
Mean PCK		Total Time		Time/Image			
29.8 %		0.9 min		0.5 sec			

Table 6.6: Quantitative Results of the Baseline on Each View of 'Humans AUC'

6.6.4 Discussion

As shown in table 6.6, while the baseline scores a good accuracy of 73% and 77.7% on views from camera 1 and 2 respectively, it scores very low accuracy of 31.5% and 29.8% on views from camera 3 and 4 respectively. By analyzing the qualitative and quantitative results of the experiments from camera 3 and 4, we conclude that the baseline approach is insensitive to the camera view and achieves its best result in frontal views only. As shown in Fig. 6.2 (c) and Fig 6.2 (d) the baseline confuses the right human joints with the left ones. Since the baseline is incapable of finding the correct locations of the human body parts when the view changes, therefore, is impossible to match the joint's legend of the baseline [3] in Fig 6.4 to all camera views in Fig 6.3. In the following experiments, we want to make the baseline viewpoint-sensitive and to find the correct locations of body joints.

6.7 Experiment 7: PSM Head Detector on 'Image PARSE'

6.7.1 Objective

Extracting the head patch of single persons from the 'Image Parse' dataset for the purpose of *Head Pose Estimation* and *Human Face Verification*.

6.7.2 Methodology

We combine the output of two SVM classifiers in the PSM model to extract the head patch: 'head top' and 'head bottom' trained body parts. Afterward, we scale the head patch up to 4.0 bicubic interpolation as a preprocessing step to the next phase in the pipeline.

6.7.3 Results

Qualitative and quantitative results are shown in Fig. 6.5 and table 6.7 respectively.

Figure 6.5: PSM Head Detector on PARSE

Dataset: PARSE [205 images]		image size: 150x150
Accuracy	Total Time	Time/Image
89.3 %	2.0 min	0.6 sec

Table 6.7: PSM Head Detector on PARSE

6.7.4 Discussion

The accuracy is only **89.3%** on this dataset because the human head is sometimes very small to be detected or it is deformed, like wearing a big hat or a mask. Refer to section 6.9.5 for more analysis.

6.8 Experiment 8: PSM Head Detector on 'KTH'

6.8.1 Objective

Extracting the head patch of single persons from the 'KTH Multiview Football' dataset for the purpose of *Head Pose Estimation* and *Human Face Verification*.

6.8.2 Methodology

Similarly, we combine the output of two SVM classifiers in the PSM model to extract the head patch: 'head top' and 'head bottom' trained body parts. Afterward, we scale the head patch up to 4.0 bicubic interpolation as a preprocessing step to the next phase in the pipeline.

6.8.3 Results

Qualitative and quantitative results are shown in Fig. 6.6 and table 6.8 respectively.

Figure 6.6: PSM Head Detector on KTH Multiview Football

Dataset: KTH [1000 images]		image size: 250x250
Accuracy	Total Time	Time/Image
92.3 %	15.5 min	0.9 sec

Table 6.8: PSM Head Detector on KTH Multiview Football

6.8.4 Discussion

Achieving **92.3%** accuracy on 1000 images in this dataset is satisfying and promising because we are bound by the head detection accuracy in our proposed approach. Refer to section 6.9.5 for more analysis.

6.9 Experiment 9: PSM Head Detector on 'Humans AUC'

6.9.1 Objective

Extracting the head patch of single persons from the 'Humans AUC' dataset for the purpose of *Head Pose Estimation* and *Human Face Verification*.

6.9.2 Methodology

Similarly, we combine the output of two SVM classifiers in the PSM model to extract the head patch: 'head top' and 'head bottom' trained body parts. Afterward, we scale the head patch up to 4.0 bicubic interpolation as a preprocessing step to the next phase in the pipeline.

6.9.3 Results

Qualitative and quantitative results are shown in Fig. 6.7 and table 6.9 respectively.

Figure 6.7: PSM Head Detector on Humans AUC

Dataset: Humans AUC[425 images]		image size: 220x220
Accuracy	Total Time	Time/Image
95.5 %	3.7 min	0.5 sec

Table 6.9: PSM Head Detector on Humans AUC

6.9.4 Discussion

95.5 % is the best head accuracy obtained so far. This is an encouraging result since the dataset contains human heads from the four different view angles.

6.9.5 Analysis

In the last three experiments of 'PSM Head detector' (Experiment 7, 8, 9), we had three options: 1) Use the PSM head detector in the baseline 2) Use a different Human Head Detector such as VGG head detector by [102] or the SoA CNN head detector [103] . 3) Train our own Human Head Detector.

We have experimented with VGG Head detector but we chose the first option mainly for three reasons: 1) the internal PSM head detector already has a satisfying accuracy that's above the 90s; 2) to avoid adding additional time complexity to our approach since we are going to run the PSM body parts SVM classifiers anyways; 3) Our objective in this thesis is to show the approach we are proposing can solve the CBS problem. Therefore, it is not the scope of this thesis to present the best accuracy of the human head detector.

6.10 Experiment 10: Cascade Face Detector on 'Image PARSE'

6.10.1 Objective

Identifying the viewing angle of a person in the 'Image Parse' dataset by analyzing the head pose using a *face verifier*.

6.10.2 Methodology

We use the Rapid Object-Cascade-Detector of Viola-John's algorithm [117] as a face pose estimator and a face verifier.

6.10.3 Results

Qualitative and quantitative results are shown in Fig. 6.8 and table 6.10 respectively.

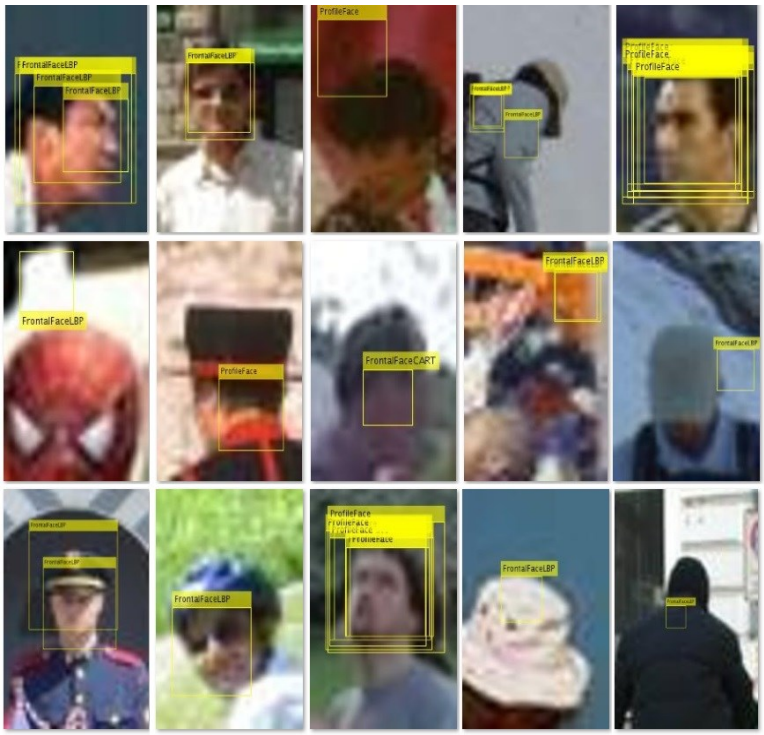


Figure 6.8: Cascade Face Detector on PARSE

Dataset: PARSE [205 images]			image size: 150x150		
TP	FP	TP+FP	Face	Non-Face	Total
130	21	151	187	18	205
Accuracy			Total Time		Time/Image
69.5 %			12 sec		0.05 sec

Table 6.10: Cascade Face Detector on PARSE

6.10.4 Discussion

'Image Parse' dataset [2] does not contain labels for face ground truth. It only contains 14 joints location of each image. Therefore, I had to annotate the entire dataset before performing any experiments. Consequently, we discovered that the dataset is ***unbalanced*** because it contains 283 images for front and profile viewed people and only 22 images for people viewed from the back. As shown in table 6.10, one can notice that the test set has only 18 people viewed from the back (non-Face). Having this unbalanced data will result in an unfair evaluation when we compare our final approach on this dataset. That is why we have supported it with two other balanced datasets: KTH Multiview Football and Humans AUC.

As shown in table 6.10, while the test set of 205 images has a ground truth of 187 people with a front face, upright, or profile face, only 130 samples were correctly classified as a face (true positives) although Viola-Johns face detector can achieve much higher accuracy than this. This is due to four reasons: 1) the average person height in the photos is only 100 to150 pixels. 2) The minimum face size to detect is 20 by 20 pixels because this is the smallest face size in the trained algorithm. That is why we scaled the patch up 4 times, which increased the *true positive* rate. 3) It still misses a lot of faces in PARSE dataset (false negatives) due to the fact that the face is actually very occluded, deformed or very distorted; 4) that the dataset actually cares about the human pose more than the face itself. For more analysis and parameter tuning used in the Viola-Johns Cascade Face Detector, please refer to section 6.12.5.

6.11 Experiment 11: Cascade Face Detector on 'KTH'

6.11.1 Objective

Identifying the viewing angle of a person in the 'KTH Multiview Football' dataset by analyzing the head pose using a *face verifier*.

6.11.2 Methodology

We use the Rapid Object-Cascade-Detector of Viola-John's algorithm [117] as a face pose estimator and a face verifier.

6.11.3 Results

Qualitative and quantitative results are shown in Fig. 6.9 and table 6.11 respectively.

Figure 6.9: Cascade Face Detector on KTH Multiview Football

Dataset: KTH [1000 images]			image size: 250x250		
TP	FP	TP+FP	Face	Non-Face	Total
413	31	444	447	553	1000
Accuracy			Total Time		Time/Image
92.4 %			36 sec		0.04 sec

Table 6.11: Cascade Face Detector on KTH

6.11.4 Discussion

This dataset does not contain labels for face ground truth, therefore, I have provided face annotation for those 1k images. For more analysis and parameter tuning used in the Viola-Johns Cascade Face Detector, please refer to section 6.12.5.

6.12 Experiment 12: Cascade Face Detector on 'Humans AUC'

6.12.1 Objective

Identifying the viewing angle of a person in the 'Humans AUC' dataset by analyzing the head pose using a *face verifier*.

6.12.2 Methodology

We use the Rapid Object-Cascade-Detector of Viola-John's algorithm [117] as a face pose estimator and a face verifier.

6.12.3 Results

Qualitative and quantitative results are shown in Fig. 6.10 and table 6.12 respectively.

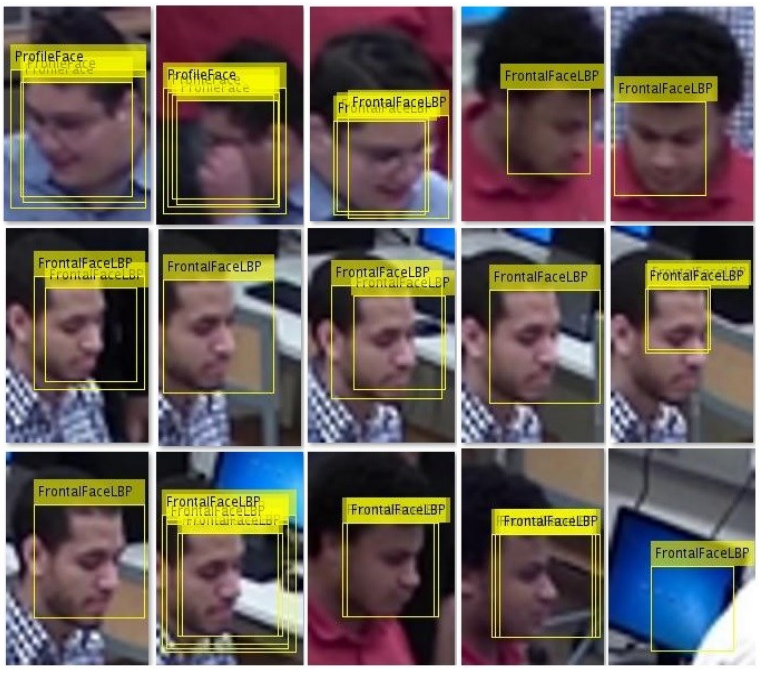


Figure 6.10: Cascade Face Detector on Humans AUC

Dataset: Humans AUC[425 images]			image size: 220x220		
TP	FP	TP+FP	Face	Non-Face	Total
204	13	217	210	215	425
Accuracy			Total Time		Time/Image
97.1 %			18 sec		0.04 sec

Table 6.12: Cascade Face Detector on Humans AUC

6.12.4 Discussion

Our proposed dataset: 'Humans AUC' contains both 14 joint locations for the person in the image as well as the person's face pose. We provide 4 DoFs (degrees of freedom) of face views. The accuracy is very high since we provide clear images of persons not deformed, occluded, or distorted images of persons in the dataset. Nevertheless, the algorithm here makes some mistakes because the head patches fed to the algorithm from the previous stage has already some false detections. Therefore, in this experiment, we are bound by the PSM baseline head detector in the experiment # 9.

Since the dataset is balanced i.e. almost half of the dataset has people viewed from front or profile views and the other half contains people viewed from the back, and face verifier algorithm here achieves a very good accuracy, I expect this dataset to achieve the best results if the bassline was able to resolve the CBS problem and become viewpoint-invariant.

6.12.5 Analysis

In the last three experiments of 'Cascade Face Detector' (Experiment 10, 11, 12), we had three options: 1) Use an existing Head Pose Estimator as a face verifier 2) Use a different Head Pose Estimator such as Zhu & Ramanan [118] Face Pose Estimation or the SoA Faster R-CNN Face Detector [119]. 3) Train our own Head Pose Estimator and Face Detector.

We have experimented with Zhu & Ramanan [118] Face Pose Estimation but we chose the first option mainly for four reasons: 1) Not only Zhu & Ramanan Face pose estimation requires large face patches to perform accurately, e.g. 500x600, but also it takes 3 seconds on average to estimate the face pose on one image 2) the Cascade Face

Detector already has a satisfying accuracy that's above 95; 3) to avoid adding additional time complexity to our approach since the Viola-Johns' algorithm is for rapid detection and can run in real time; 4) Our objective in this thesis is to show the approach we are proposing can solve the CBS problem. Therefore, it is not the scope of this thesis to present the best accuracy of human head estimation or face detector.

We use the Face Cascade detector for two goals: 1) to work as a face pose estimation 2) to work as a face verifier. We use three main classification models to find faces in the scaled up patch. These are: 'FrontalFaceCART', 'FrontalFaceLBP', and 'ProfileFace'. While the first and the third model use Haar features to encode facial features, the second model uses local binary patterns (LBP) features. We use the results of the first goal as a face verifier. That means if the extracted head patch was not recognized as any of the three classification models, it will be considered non-face, and consequently, it is classified as a person viewed from the back in our proposed approach.

Two parameters were tuned for best accuracy: 1) MergeThreshold was set to 0 to get all detections without performing any merging operation for the detected bounding boxed. 2) MinSize was set to [60, 60] for two main reasons: to not waste time in detecting smaller objects, since we already know it is a head patch and to avoid false positives as well, since we already know the minimum face size to be detected prior to processing which cannot be less than 60x60. For more analysis of the evaluation methodology used in these three experiments, please refer to section 4.2.

6.13 Experiment 13: SHAPE on 'Image PARSE'

6.13.1 Objective

Report the accuracy of the *enhanced* baseline on the 'Image PARSE' dataset.

6.13.2 Methodology

Evaluate the proposed approach discussed in chapter 4 and experimented in this chapter on this dataset. Hence, we executed *SHAPE* in a pipelined architecture. SHAPE (Smart Human Articulated Pose Estimation) consists of the Speeded up Baseline, Head Detector, and Face pose estimator as face verifier and feedback to the baseline.

6.13.3 Results

Qualitative and quantitative results are shown in Fig. 6.11 and table 6.13 respectively.

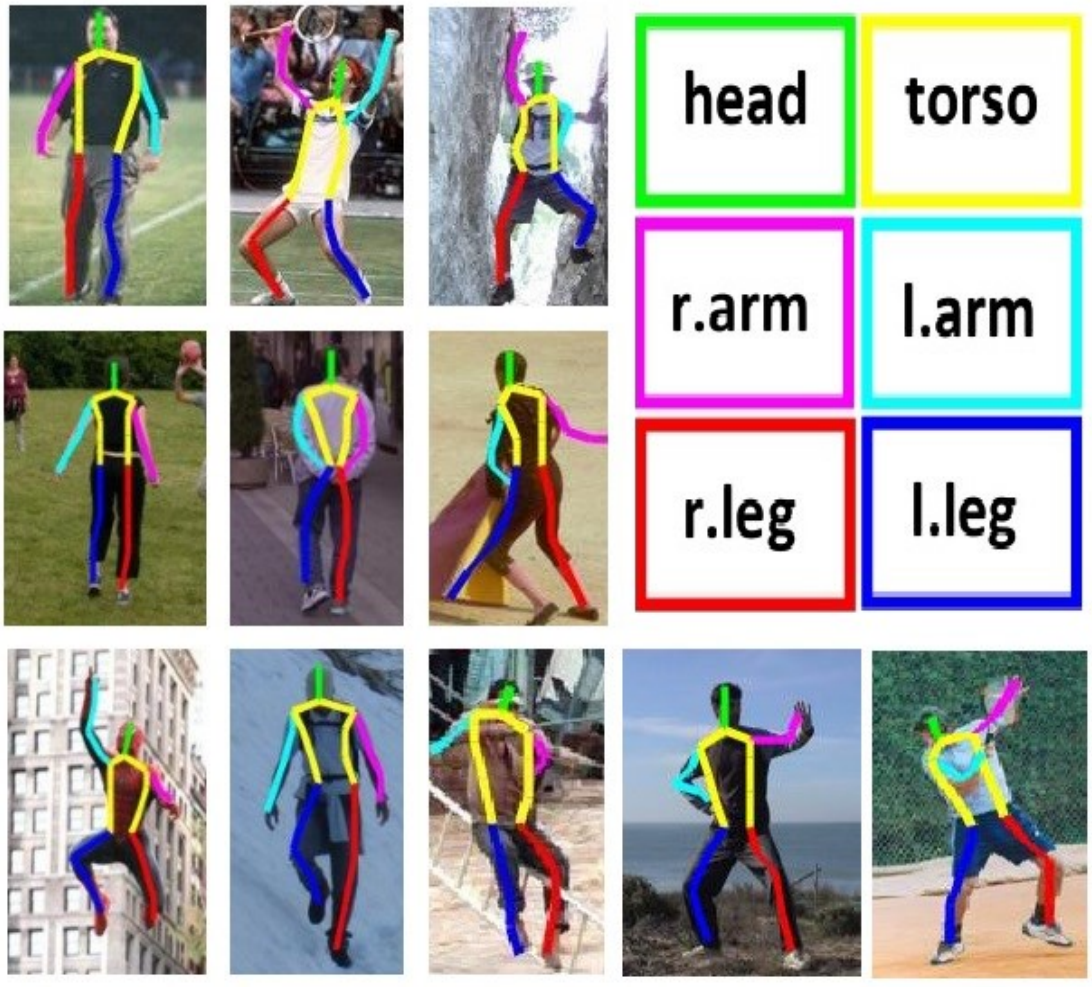


Figure 6.11: Qualitative results of SHAPE on PARSE dataset

Dataset: PARSE [205 images] image size: 150x150							
Points	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle
Accuracy	89.3	68.3	53.7	37.3	69.8	58.5	50.5
	Mean PCK		Total Time		Time/Image		
	61.0 %		2.0 min		0.6 sec		

Table 6.13: SHAPE on PARSE

6.13.4 Discussion

Running the speeded baseline with the corrected ground truth in Experiment 3 achieved the accuracy of **68%** on 'Image PARSE' dataset while when using SHAPE the accuracy dropped few percentages to **61%** as shown in table 6.13. This drop in accuracy was anticipated in experiment 10. That is because the Face Cascade detector used as a face verifier scores low on 'Image Parse' dataset as shown previously in table 6.10. That is because of the reasons discussed in details in section 6.10.4. One of them was that the dataset contains unbalanced data. Therefore, we cannot rely on this dataset only to provide a fair evaluation.

As shown in Fig. 6.11, the baseline has become sensitive to the human viewing angle on the first and seconds rows. Hence, it localizes the human body parts correctly. However, on the third row, it became insensitive to the viewing point because the faces were either occluded, deformed, covered by masks, or has a very low resolution.

Therefore, we conclude that it difficult to solve the CBS problem using our proposed approach (PSM Head Detector + Face Cascade detector as a face verifier) for the following two special cases: 1) the person's height in the image is very small such as 80 pixels; 2) the face is occluded, deformed, or covered by masks. Nevertheless, the room for improvement is open to use other algorithms or methods as suggested in section 6.10.4 to be plugged in the same pipelined architecture to solve the CBS.

6.14 Experiment 14: SHAPE on 'KTH'

6.14.1 Objective

Report the accuracy of the *enhanced* baseline on the 'KTH Multiview Football'.

6.14.2 Methodology

Evaluate the proposed approach discussed in chapter 4 and experimented in this chapter on this dataset. Hence, we executed *SHAPE* in a pipelined architecture. SHAPE (Smart Human Articulated Pose Estimation) consists of the Speeded up Baseline, Head Detector, and Face pose estimator as face verifier and feedback to the baseline.

6.14.3 Results

Qualitative and quantitative results are shown in Fig. 6.12 and table 6.14 respectively.

Figure 6.12: Qualitative results of SHAPE on PARSE dataset

Dataset: KTH [1000 images] image size: 250x250							
Points	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle
Accuracy	92.3	81.5	65.6	46.5	65.0	56.5	33.1
	Mean PCK		Total Time		Time/Image		
	62.9 %		15.5 min		0.9 sec		

Table 6.14: SHAPE on KTH

6.14.4 Discussion

Running the speeded baseline achieved the accuracy of 44.3% on 'KTH Multiview Football' dataset in Experiment 4. Now using SHAPE, the accuracy jumped drastically from **44.3%** (in Experiment 4) to **62.9%**. That is expected because of three reasons: 1) the CBS problem was resolved; 2) the dataset is balanced to some extent; 2) the PSM head detector and Face Cascade Detector as a face verifier score good results on this dataset. That's a very good accuracy improvements because we are bound by the baseline accuracy, PSM head detector accuracy, and Face Cascade Detector as a face verifier accuracy as discussed in the previous experiments.

As shown in Fig 6.12, the baseline is sensitive to the viewing point on the first and seconds rows. Hence, human body parts are localized correctly. On the third row, the CBS solver does succeed but what fails the system is its dependency of the 2D HPE algorithm. Therefore, the accuracy of **62%** could have been higher if the 2D HPE baseline was more accurate in estimating human poses for blurred and fast-motion 2D images.

We achieve **18.6%** increase in accuracy using SHAPE on 'KTH Multiview Football' dataset since it was **44.3%** in Experiment 4 and reached to **62.9%** in Experiment 14. That is mainly because we succeeded in making the 2D HPE baseline in [3] viewpoint-invariant. This is more accurate because the estimated human joints are not confused between each other.

6.15 Experiment 15: SHAPE on 'Humans AUC'

6.15.1 Objective

Report the accuracy of the *enhanced* baseline on the 'Humans AUC' dataset.

6.15.2 Methodology

Evaluate the proposed approach discussed in chapter 4 and experimented in this chapter on this dataset. Hence, we executed *SHAPE* in a pipelined architecture. SHAPE (Smart Human Articulated Pose Estimation) consists of the Speeded up Baseline, Head Detector, and Face pose estimator as face verifier and feedback to the baseline.

6.15.3 Results

Qualitative and quantitative results are shown in Fig. 6.13 and table 6.15 respectively.

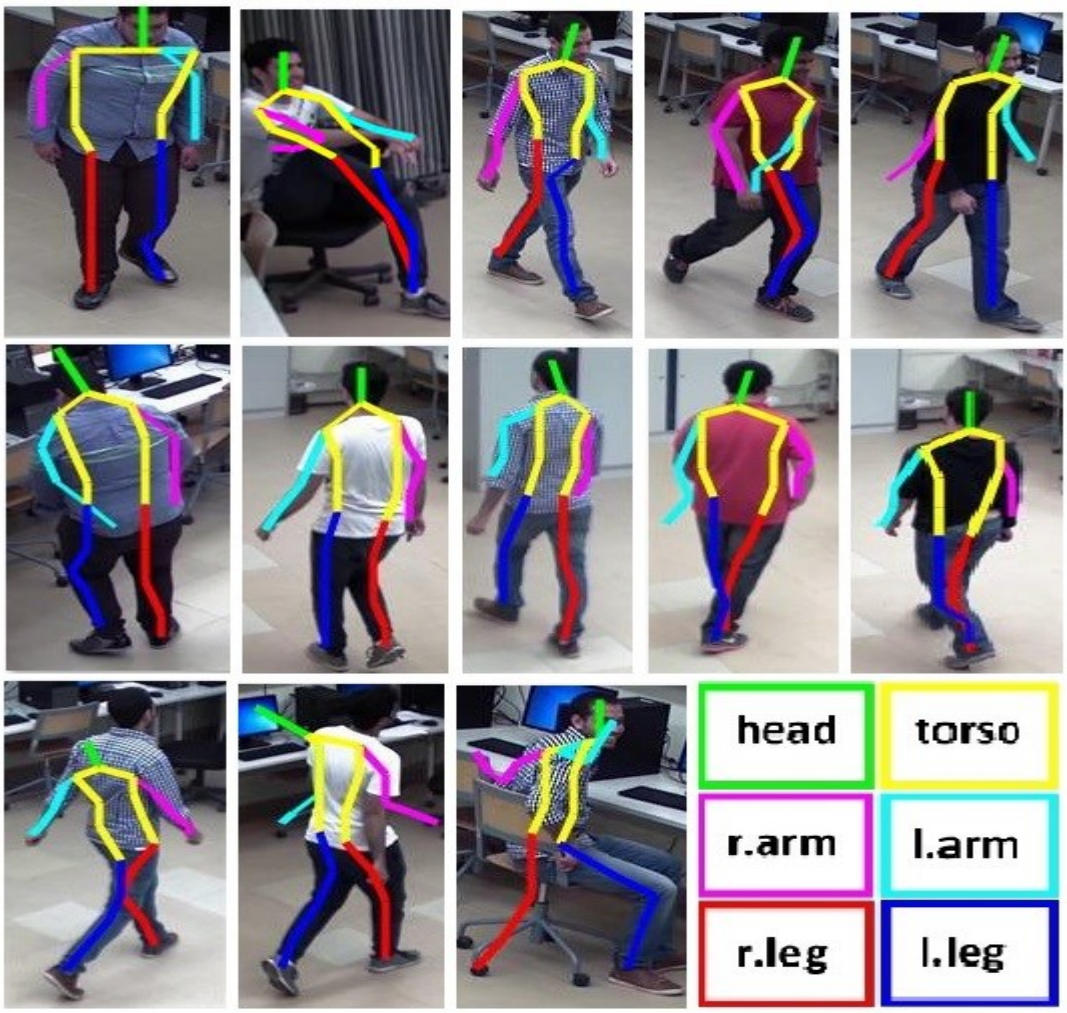


Figure 6.13: Qualitative results of SHAPE on PARSE dataset

Dataset: Humans AUC [425 images] image size: 220x220							
Points	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle
Accuracy	95.5	78.5	63.8	52.6	75.1	72.6	66.9
		Mean PCK		Total Time		Time/Image	
		72.1 %		3.7 min		0.5 sec	

Table 6.15: SHAPE on Humans AUC

6.15.4 Discussion

Running the speeded baseline with the corrected ground truth in Experiment 3 achieved the accuracy of **49.0%** on 'Humans AUC' dataset while when using SHAPE the accuracy jumped significantly to **72.1%**. That is expected because of three reasons: 1) the CBS problem was resolved; 2) the dataset is perfectly balanced; 2) the PSM head detector and Face Cascade Detector as a face verifier score very good results on this dataset. That's a very good accuracy improvements because the final accuracy is highly dependent on the baseline accuracy, PSM head detector accuracy, and Face Cascade Detector as a face verifier accuracy as discussed in the previous experiments.

As shown in Fig 6.13, the 2D HPE baseline is viewpoint-invariant. While it behaves normally in solving the CBS problem on the first two rows, it makes no mistakes on the third row although some of the human heads were localized incorrectly using the PSM head detector. Fortunately, these cases did not affect the total system accuracy since the face verifier reported no faces, and hence, the human body parts were localized precisely without confusion.

We achieve **23.1%** increase in accuracy using SHAPE on 'Humans AUC' dataset, since it was **49.0%** in Experiment 3 and reached to **72.1%** in Experiment 15. That is mainly because we succeeded in making the 2D HPE baseline in [3] viewpoint-invariant. This is more accurate because now the estimated human body parts are not confused between one another.

6.16 PC Specifications

All experiments were performed on Linux Mint 18.1 64-bit (Serena) with the following software libraries: OpenCV 3, MATLAB 2017a, and CUDA 8.0. The Operating system is running on a state of the art PC system equipped with a 64-bit Intel(R) Core™ i7-7700K CPU running at 4.5GHz, nVidia GeForce GTX 1080 graphics card with 2560 CUDA cores, 8 GB of internal GPU DDR memory, and a total of 16 GB DDRAM memory running at 3000 MHz. A demo of an experiment using SHAPE on this system is available on [120].

CHAPTER SEVEN

CONCLUSION AND FUTURE WORK

We have defined a novel problem in 2D Human Pose Estimation approaches, that is the Confusion of Body Sides (CBS). We have provided a running baseline approach of a notable 2D HPE algorithm by Yang and Ramanan [3] that uses Pictorial Structure Model to detect human body parts and estimate 2D human pose. The PSM approach provided by [3] which was reported to be the SoA 2D single human pose estimation by [24] does not perform well when it is applied to non-frontal views of humans and its accuracy decreases since the baseline is insensitive to the viewpoint.

We proposed and implemented a solution to solve the CBS problem in 2D HPE algorithms. In addition, we showed quantitative and quantitative results of our approach which confirms that we have solved the CBS problem in the baseline approach and succeeded in making the baseline viewpoint-invariant when estimating the 2D of a human body. Empirical results show that our approach increases the baseline accuracy by 20% on average. We demonstrate how our approach can be plugged in a 2D HPE algorithm that is insensitive to viewpoints and suffers from the CBS problem.

Moreover, we have proposed a challenging dataset called Humans AUC with ground truth annotation of joints and faces. We also provide quantitative and qualitative results by applying PSM on a subset of 'Human AUC' dataset.

In order to have an automated system, we recommend using a reliable full-body human detector with a high rate of detections. This human detection phase will be plugged in the pipeline after the building of the PSM and before pose estimation takes place. The human detector algorithm will detect every single human in a frame, then

apply some preprocessing like resizing, then feed the input to the HPE algorithm with SHAPE to estimate the human skeleton.

The final 2D estimated pose accuracy is highly dependent on three major factors: 1) the accuracy of the 2D HPE algorithm; 2) the accuracy of the head detector algorithm used; 3) the accuracy of the face detector algorithm used. SHAPE reshapes the future research to address the CBS problem and estimate the human pose estimation in 2D more accurately.

7.1 Summary of the Results

We summarize below the effectiveness of our approach in making the 2D HPE baseline approach [3] viewpoint-invariant as shown in Fig 7.1.

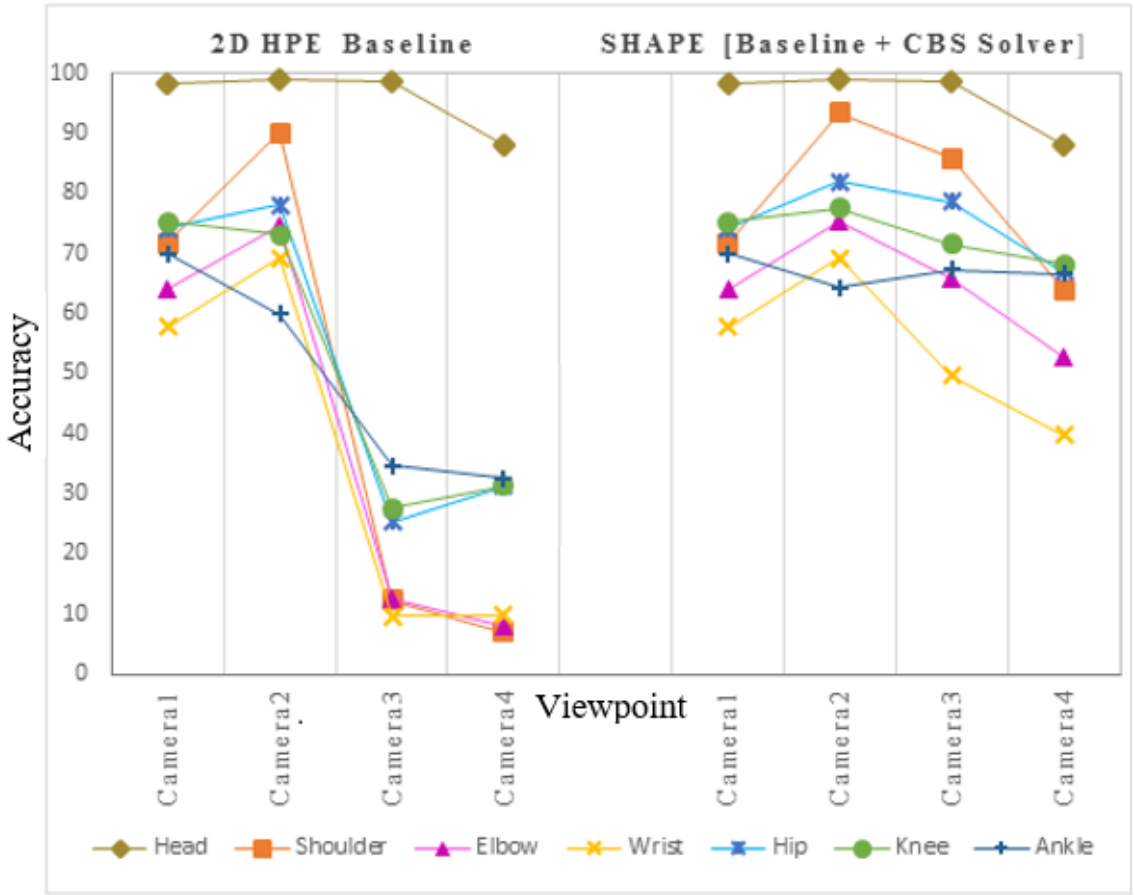


Figure 7.1: Accuracy on Humans AUC dataset According to the Viewpoint

Due to the fact that we could not find any research in the literature that solved the CBS problem in 2D HPE, we were unable to compare our results with previous work. Nevertheless, we compared the results of the baseline versus the SHAPE [baseline + CBS solver] on two popular HPE benchmarks and on our proposed challenging dataset as well, as shown in Fig 7.2. Finally, the total time added to the baseline to test an image is shown in Fig 7.3.

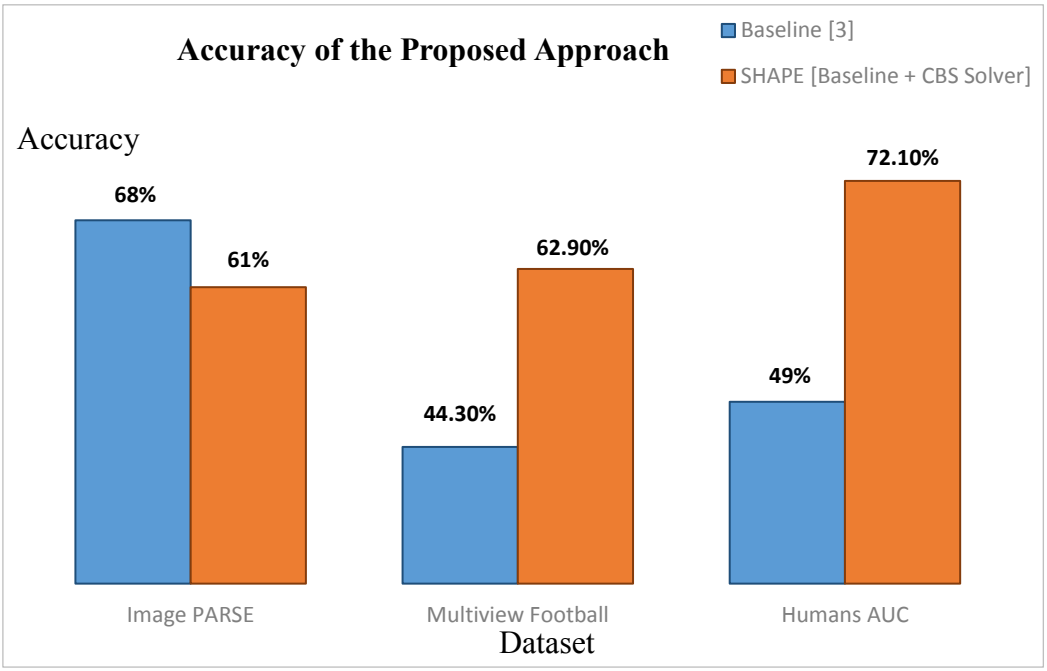


Figure 7.2: Accuracy of Baseline vs. SHAPE on Three Datasets

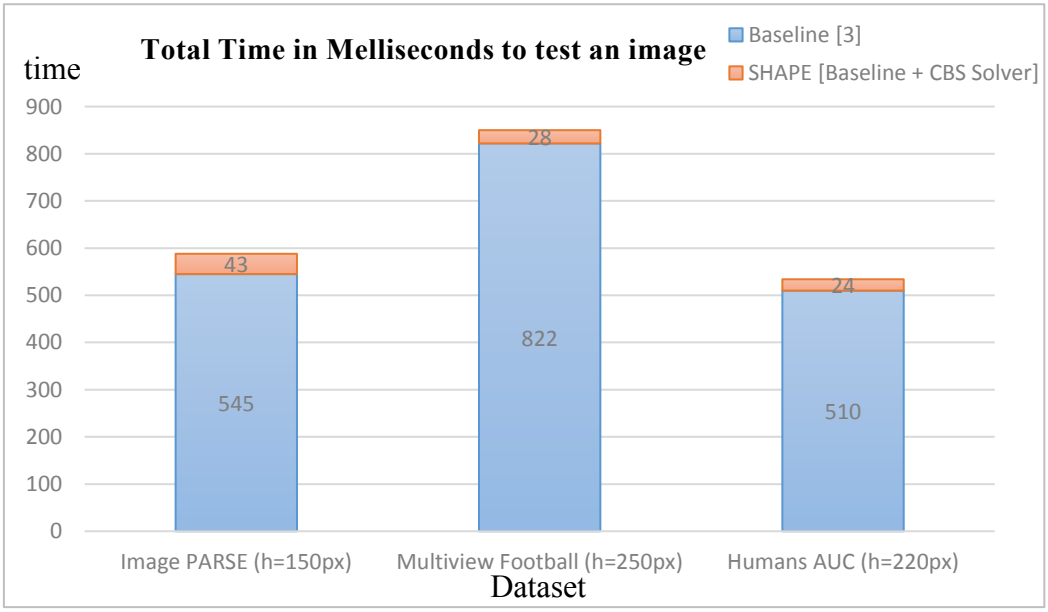


Figure 7.3: Additional Time Cost Added to the Baseline

7.2 Future Work

In the near future, I am looking forward to 1) adding SHAPE to 2D HPE approaches that suffer from the CBS problem; 2) using 2D HPE in multi-view instead of a single view and analyze the problem with occluded joints; 3) building on accurate 2D HPE approaches to perform Action Recognition in real-time.

REFERENCES

[1] V. Kazemi, M. Burenius, H. Azizpour, and J. Sullivan, "Multi-view Body Part Recognition with Random Forests," *Procedings Br. Mach. Vis. Conf. 2013*, p. 48.1-48.11, 2013.

[2] D. Ramanan, "Learning to parse images of articulated bodies," *Adv. Neural Inf. Process. Syst.*, vol. 19, pp. 1129–1136, 2007.

[3] Y. Yang and D. Ramanan, "Articulated Human Detection with Flexible Mixtures-of-Parts.," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–15, 2012.

[4] R. Klette, *Concise Computer Vision: An Introduction into Theory and Algorithms*. London: Springer London, 2014.

[5] P. Files *et al.*, "opencv : OpenCV." [Online]. Available: http://opencv.org/. [Accessed: 01-Apr-2015].

[6] "OpenCV Vision Challenge | OpenCV." [Online]. Available: http://opencv.org/opencv-vision-challenge.html. [Accessed: 01-Apr-2015].

[7] Matheen Siddiqui, "Human Pose Estimation from a Single View Point," University of Southern California, 2009.

[8] Ł. Kami and K. Kowalak, "Human Activity Recognition in Multiview Video," pp. 148–153, 2014.

[9] K. C. Chan, C. K. Koh, and C. S. George Lee, "Selecting best viewpoint for human-pose estimation," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 4844–4849, 2014.

[10] M. Ozuysal, V. Lepetit, and P. Fua, "Pose Estimation for Category Speci c Multiview Object Localization ∗ ¨," *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 778–785, 2009.

[11] J. Abella *et al.*, "Multi-modal descriptors for multi-class hand pose recognition in human computer interaction systems," *Proc. 15th ACM Int. Conf. multimodal Interact. - ICMI '13*, pp. 503–508, 2013.

[12] M. Shafi, "Face Pose Estimation in Monocular Images," *Ph.D Diss.*, p. 141, 2010.

[13] S. Salti, O. Schreer, and L. Di Stefano, "Real-time 3d arm pose estimation from monocular video for enhanced HCI," *Proceeding 1st ACM Work. Vis. networks Behav. Anal. - VNBA '08*, p. 1, 2008.

[14] Z. Liu, J. Zhu, J. Bu, and C. Chen, "A survey of human pose estimation: The body parts parsing based methods," *J. Vis. Commun. Image Represent.*, vol. 32, pp. 10–19, 2015.

[15] M. Burenius, J. Sullivan, and S. Carlsson, "3D Pictorial Structures for Multiple View Articulated Pose Estimation," *Cvpr*, pp. 3618–3625, 2013.

[16] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari, "2D Articulated Human Pose Estimation and Search in (Almost) Unconstrained Still Images," *Int. J. Comput. Vis.*, vol. 99, pp. 190–214, 2012.

[17] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," *Comput. Vis. Pattern ...*, 2011.

[18] X. Chen and A. L. Yuille, “Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations,” *Adv. Neural Inf. Process. Syst.*, pp. 1736–1744, 2014.

[19] M. B. Holte, C. Tran, M. M. Trivedi, and T. B. Moeslund, “Human pose estimation and activity recognition from multi-view videos: Comparative explorations of recent developments,” *IEEE J. Sel. Top. Signal Process.*, vol. 6, no. 5, pp. 538–552, 2012.

[20] S. Mukherjee, S. K. Biswas, and D. P. Mukherjee, “Human action recognition in video by ‘meaningful’ poses,” *Proc. Seventh Indian Conf. Comput. Vision, Graph. Image Process. - ICVGIP ’10*, pp. 9–16, 2010.

[21] R. Vezzani, D. Baltieri, and R. Cucchiara, “People reidentification in surveillance and forensics,” *ACM Comput. Surv.*, vol. 46, no. 2, pp. 1–37, Nov. 2013.

[22] M. E. M. RAGAB, “Multiple Camera Pose Estimation,” The Chinese University of Hong Kong, 2008.

[23] E. Cho and D. Kim, “Accurate Human Pose Estimation by Aggregating Multiple Pose Hypotheses Using Modi fi ed Kernel Density Approximation,” *Spl*, vol. 22, no. 4, pp. 445–449, 2015.

[24] X. Perez-Sala, S. Escalera, C. Angulo, and J. Gonzàlez, “A survey on model based approaches for 2D and 3D visual human pose recovery,” *Sensors (Basel).*, vol. 14, no. 3, pp. 4189–4210, 2014.

[25] E. Murphy-Chutorian and M. M. Trivedi, “Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness,” *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 300–311, 2010.

[26] Vivek Kumar Singh, “Monocular Human Pose Tracking And Action Recognition In Dynamic Environment,” University Of Southern California, University of Southern California, 2011.

[27] H. K. Lee and J. H. Kim, “An HMM-based threshold model approach for gesture recognition,” *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 21, no. 10, pp. 961–973, 2002.

[28] S. Park and M. M. Trivedi, “Understanding human interactions with track and body synergies (TBS) captured from multiple views,” *Comput. Vis. Image Underst.*, vol. 111, no. 1, pp. 2–20, 2008.

[29] Á. Utasi and C. Benedek, “A 3-D marked point process model for multi-view people detection,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011, pp. 3385–3392.

[30] J. Assfalg, M. Bertini, C. Colombo, A. Del Bimbo, and W. Nunziati, “Semantic annotation of soccer videos: Automatic highlights identification,” *Comput. Vis. Image Underst.*, vol. 92, no. 2–3, pp. 285–305, 2003.

[31] J. Kilner, J. Y. Guillemaut, and A. Hilton, “3D action matching with key-pose detection,” in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops 2009*, 2009, pp. 1–8.

[32] L. Bourdev, J. Malik, U. C. Berkeley, A. Systems, P. Ave, and S. Jose, "Poselets : Body Part Detectors Trained Using 3D Human Pose Annotations," *2009 IEEE 12th Int. Conf. Comput. Vis.*, no. Iccv, pp. 1365--1372, 2009.

[33] S.-R. Ke, L. Zhu, J.-N. Hwang, H.-I. Pai, K.-M. Lan, and C.-P. Liao, "Real-Time 3D Human Pose Estimation from Monocular View with Applications to Event Detection and Video Gaming," *7th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, pp. 489–496, 2010.

[34] A. Schick and R. Stiefelhagen, "3D pictorial structures for human pose estimation with supervoxels," *Proc. - 2015 IEEE Winter Conf. Appl. Comput. Vision, WACV 2015*, pp. 140–147, 2015.

[35] I. Mikić, M. Trivedi, E. Hunter, and P. Cosman, "Human body model acquisition and tracking using voxel data," *Int. J. Comput. Vis.*, vol. 53, no. 3, pp. 199–223, 2003.

[36] F. Guo and G. Qian, "Human pose inference from stereo cameras," in *Proceedings - IEEE Workshop on Applications of Computer Vision, WACV 2007*, 2007.

[37] R. Rosales, M. Siddiqui, J. Alon, and S. Sclaroff, "Estimating 3D body pose using uncalibrated cameras," *Proc. 2001 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition. CVPR 2001*, vol. 1, pp. 1–8, 2001.

[38] H. Wu *et al.*, "Computer Vision – ACCV 2007," *Comput. Vision–ACCV 2007*, vol. 4843, no. November 2015, pp. 688–697, 2007.

[39] X. Perez-Sala, S. Escalera, and C. Angulo, "Survey on 2D and 3D Human Pose Recovery.," *Ccia*, pp. 101–110, 2012.

[40] D. M. Gavrila and L. S. Davis, "3-D model-based tracking of humans in action: a multi-view approach," *Proc. CVPR IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 73–80, 1996.

[41] I. Science, "Three-Dimensional Human Body Model Acquisition from Multiple Views," *Ijcv*, vol. 30, no. 3, pp. 191–218, 1998.

[42] Shian-Ru Ke, "Recognition of Human Actions based on 3D Pose Estimation via Monocular Video Sequences," University of Washington, 2014.

[43] J. Salvi, X. Armangué, and J. Batlle, "A comparative review of camera calibrating methods with accuracy evaluation," *Pattern Recognit.*, vol. 35, no. 7, pp. 1617–1635, 2002.

[44] T. B. Moeslund, A. Hilton, and V. Kr??ger, "A survey of advances in vision-based human motion capture and analysis," *Comput. Vis. Image Underst.*, vol. 104, no. 2–3 SPEC. ISS., pp. 90–126, 2006.

[45] A. Agarwal and B. Triggs, "Recovering 3D human pose from monocular images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 44–58, 2006.

[46] G. Rogez, C. Orrite-Uru??uela, and J. Mart??nez-del-Rinc??n, "A spatio-temporal 2D-models framework for human pose recovery in monocular sequences," *Pattern Recognit.*, vol. 41, no. 9, pp. 2926–2944, 2008.

[47] A. Mittal, L. Zhao, and L. Davis, "Human body pose estimation using silhouette shape analysis," *Adv. Video Signal Based ...*, pp. 263–270, 2003.

[48] M. Andriluka, S. Roth, and B. Schiele, “Monocular 3D Pose Estimation and Tracking by Detection,” *2010 Ieee Conf. Comput. Vis. Pattern Recognit.*, no. 2, pp. 623–630, 2010.

[49] R. Navarathna, S. Sridharan, and S. Lucey, “Fourier Active Appearance Models,” *2011 Int. Conf. Comput. Vis.*, no. i, pp. 1919–1926, 2011.

[50] J. G. Daugman, “Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters.,” *J. Opt. Soc. Am. A.*, vol. 2, no. 7, pp. 1160–1169, 1985.

[51] K. Van De Sande, T. Gevers, and C. Snoek, “Evaluating color descriptors for object and scene recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1582–1596, 2010.

[52] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, 2005, vol. I, pp. 886–893.

[53] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun, “Real-time identification and localization of body parts from depth images,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2010, pp. 3108–3113.

[54] N. Pugeault and R. Bowden, “Spelling it out: Real-time ASL fingerspelling recognition,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 1114–1119.

[55] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, “Performance of optical flow techniques,” *Int. J. Comput. Vis.*, vol. 12, no. 1, pp. 43–77, 1994.

[56] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” in *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2008.

[57] I. Laptev, “On space-time interest points,” in *International Journal of Computer Vision*, 2005, vol. 64, no. 2–3, pp. 107–123.

[58] B. Yao and L. Fei-Fei, “Grouplet: A structured image representation for recognizing human and object interactions,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 9–16.

[59] D. Forsyth, “Object detection with discriminatively trained part-based models,” *Computer (Long. Beach. Calif).*, vol. 47, no. 2, pp. 6–7, 2014.

[60] C. Rother, V. Kolmogorov, and A. Blake, “‘GrabCut,’” *ACM Trans. Graph.*, vol. 23, no. 3, p. 309, 2004.

[61] K. Mikolajczyk *et al.*, “A comparison of affine region detectors,” *Int. J. Comput. Vis.*, vol. 65, no. 1–2, pp. 43–72, 2005.

[62] M. Andriluka, S. Roth, and B. Schiele, “Pictorial structures revisited: People detection and articulated pose estimation,” *2009 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work. CVPR Work. 2009*, pp. 1014–1021, 2009.

[63] I. A. Karaulova, P. M. Hall, and A. D. Marshall, “A Hierarchical Model of Dynamics for Tracking People with a Single Video Camera,” *Proc. Br. Mach. Vis. Conf.*, pp. 352–-361, 2000.

[64] S. Savarese and L. Fei-Fei, “3D generic object categorization, localization and pose estimation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2007.

[65] H. Su, M. Sun, L. Fei-Fei, and S. Savarese, “Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2009, pp. 213–220.

[66] M. Salzmann, F. Moreno-Noguer, V. Lepetit, and P. Fua, “Closed-Form Solution to Non-rigid 3D Surface Registration BT - Computer Vision–ECCV …,” in *Computer Vision–ECCV …*, vol. 5305, no. Chapter 43, 2008, pp. 581–594.

[67] M. A. Fischler and R. A. Elschlager, “The Representation and Matching of Pictorial Structures Representation,” *IEEE Trans. Comput.*, vol. C-22, no. 1, pp. 67–92, 1973.

[68] R. B. Girshick, P. F. Felzenszwalb, and D. Mcallester, “Object detection with grammar models,” *Adv. Neural*, pp. 1–9, 2011.

[69] I. I. Conference and I. Processing, “PICTORIAL STRUCTURES FOR OBJECT RECOGNITION AND PART LABELING IN DRAWINGS Amir Sadovnik and Tsuhan Chen Department of Electrical and Computer Engineering , Cornell University,” pp. 3613–3616, 2011.

[70] J. Śanchez-Riera, J. Östlund, P. Fua, and F. Moreno-Noguer, “Simultaneous pose, correspondence and non-rigid shape,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1189–1196.

[71] website, “Buffy Stickmen dataset,” 2009.

[72] R. Urtasun, D. J. Fleet, and P. Fua, “Temporal motion models for monocular and multiview 3D human body tracking,” *Comput. Vis. Image Underst.*, vol. 104, no. 2–3 SPEC. ISS., pp. 157–177, 2006.

[73] I. Rius, J. Gonzàlez, J. Varona, and F. Xavier Roca, “Action-specific motion prior for efficient Bayesian 3D human body tracking,” *Pattern Recognit.*, vol. 42, no. 11, pp. 2907–2921, 2009.

[74] R. Urtasun and P. Fua, “3D Human Body Tracking Using Deterministic Temporal Motion Models,” in *Computer Vision - ECCV 2004*, vol. 3023, 2004, pp. 92–106.

[75] R. Urtasun, D. J. Fleet, and P. Fua, “Monocular 3-D tracking of the golf swing,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, vol. 2, pp. 932–938.

[76] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh, “3D reconstruction of a moving point from a series of 2D projections,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, vol. 6313 LNCS, no. PART 3, pp. 158–171.

[77] B. Yao, “Modeling Mutual Context of Object and Human Pose in Human-Object Interaction Activities,” vol. 1, 2010.

[78] M. Andriluka and L. Sigal, “Human context: Modeling human-human interactions for monocular 3D pose estimation,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7378 LNCS, pp. 260–272, 2012.

[79] D. Marr and H. K. Nishihara, “Representation and recognition of the spatial organization of three-dimensional shapes.,” *Proceedings of the Royal Society of London. Series B, Containing papers of a Biological character. Royal Society (Great Britain)*, vol. 200, no. 1140. pp. 269–294, 1978.

[80] P. F. Felzenszwalb and D. P. Huttenlocher, “Pictorial structures for object recognition,” *Int. J. Comput. Vis.*, vol. 61, no. 1, pp. 55–79, 2005.

[81] L. Pishchulin, A. Jain, M. Andriluka, T. Thormahlen, and B. Schiele, “Articulated people detection and pose estimation: Reshaping the future,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 3178–3185, 2012.

[82] T. Vajda and ??br??m Zolt??n, “Pictorial structure based people detection and pose estimation in videos,” *Proc. - 2011 IEEE 7th Int. Conf. Intell. Comput. Commun. Process. ICCP 2011*, pp. 315–318, 2011.

[83] P. Fihl and T. B. Moeslund, “Pose estimation of interacting people using pictorial structures,” *Proc. - IEEE Int. Conf. Adv. Video Signal Based Surveillance, AVSS 2010*, pp. 462–468, 2010.

[84] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, “Strong appearance and expressive spatial models for human pose estimation,” *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 3487–3494, 2013.

[85] M. Eichner and V. Ferrari, “Better appearance models for pictorial structures,” *Procedings Br. Mach. Vis. Conf.*, pp. 1–11, 2009.

[86] P. Kaliamoorthi and R. Kakarala, “Parametric annealing: A stochastic search method for human pose tracking,” *Pattern Recognit.*, vol. 46, no. 5, pp. 1501–1510, 2013.

[87] L. Fei, “Combining pictorial structure and image features to estimate human pose,” *Proc. - 2012 9th Int. Conf. Fuzzy Syst. Knowl. Discov. FSKD 2012*, no. Fskd, pp. 1764–1768, 2012.

[88] M. Eichner and V. Ferrari, “We are family: Joint pose estimation of multiple persons,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6311 LNCS, no. PART 1, pp. 228–242, 2010.

[89] T. Pfister, J. Charles, and A. Zisserman, “Flowing ConvNets for Human Pose Estimation in Videos,” *ICCV, Int. Conf. Comput. Vis.*, 2015.

[90] Q. R. . Wanying Luo, “Human Pose Estimation Based on,” *ICSP2014 Proc.*, pp. 1257–1262, 2014.

[91] M. Sun and S. Savarese, “Articulated part-based model for joint object detection and pose estimation,” *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 723–730, 2011.

[92] M. Dantone, J. Gall, C. Leistner, and L. Van Gool, “Human Pose Estimation Using Body Parts Dependent Joint Regressors,” *Cvpr*, pp. 3041–3048, 2013.

[93] F. Wang and Y. Li, "Beyond Physical Connections: Tree Models in Human Pose Estimation," *2013 IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. cs.CV, pp. 596–603, 2013.

[94] A. Toshev and C. Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," *2014 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1653–1660, 2013.

[95] W. Ouyang, X. Chu, and X. Wang, "Multi-source Deep Learning for Human Pose Estimation," *2014 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2337–2344, 2014.

[96] V. Ramakrishna, D. Munoz, M. Hebert, J. Andrew Bagnell, and Y. Sheikh, "Pose machines: Articulated pose estimation via inference machines," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8690 LNCS, no. PART 2, pp. 33–47, 2014.

[97] S. R. and X. L. C. Guo, "A strong bilayer appearance model for human pose estimation from a high freedom still image," *IEEE Int. Conf. Image Pr*ocessing (ICIP), Phoenix, AZ, USA*, pp. 1284–1288, 2016.

[98] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic, "3D pictorial structures for multiple human pose estimation," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1669–1676, 2014.

[99] A. Bearman and C. Dong, "Human Pose Estimation and Activity Classification Using Convolutional Neural Networks," *Stanford CS231n*, 2015.

[100] S. Amin and M. Rohrbach, "Multi-view Pictorial Structures for 3D Human Pose Estimation," *Bmvc2013*, vol. 87, p. 2012, 2013.

[101] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic, "3D Pictorial Structures Revisited: Multiple Human Pose Estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 1929–1942, 2016.

[102] M. J. Marin-Jimenez, A. Zisserman, M. Eichner, and V. Ferrari, "Detecting people looking at each other in videos," *Int. J. Comput. Vis.*, vol. 106, no. 3, pp. 282–296, 2014.

[103] T. H. Vu, A. Osokin, and I. Laptev, "Context-aware CNNs for person head detection," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 11-18-NaN-2015, pp. 2893–2901, 2016.

[104] N. A. Thacker *et al.*, "Performance characterization in computer vision: A guide to best practices," *Comput. Vis. Image Underst.*, vol. 109, no. 3, pp. 305–334, 2008.

[105] website, "Buffy Stickmen dataset," 2009. [Online]. Available: http://www.robots.ox.ac.uk/.

[106] "Leeds Sports Pose." [Online]. Available: http://www.comp.leeds.ac.uk/mat4saj/lsp.html.

[107] "MPII Human Pose." [Online]. Available: http://human-pose.mpi-inf.mpg.de/.

[108] "PASCAL Stickmen." [Online]. Available: http://groups.inf.ed.ac.uk.

[109] "Synchronic Activities." [Online]. Available: http://groups.inf.ed.ac.uk/.

[110] "FLIC-motion." [Online]. Available: http://cs.nyu.edu/.

[111] "Parse." [Online]. Available: http://www.ics.uci.edu/.

[112] “Human Pose in Wild.” [Online]. Available: https://lear.inrialpes.fr/%5Cr%5Cn.

[113] M. Armstrong, A. Zisserman, and P. Beardsley, “Euclidean Structure from Uncalibrated Images,” *Proc. 5th Br. Mach. Vis. Conf. York, Engl.*, vol. 2, pp. 509–518, 1994.

[114] W. Goyert, R. Sagarin, and J. Annala, “The promise and pitfalls of Marine Stewardship Council certification: Maine lobster as a case study,” *Marine Policy*, vol. 34, no. 5. pp. 1103–1109, 2010.

[115] I. Over *et al.*, “Discriminatively trained deformable part models,” 2012. [Online]. Available: https://github.com/rbgirshick/voc-release4.01/blob/master/fconvsse.cc.

[116] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” *26th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR*, pp. 1–8, 2008.

[117] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” *Comput. Vis. Pattern Recognit.*, vol. 1, p. I--511--I--518, 2001.

[118] X. Zhu and D. Ramanan, “Face detection, pose estimation, and landmark localization in the wild,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 2879–2886, 2012.

[119] H. Jiang and E. Learned-Miller, “Face Detection with the Faster R-CNN,” *Corr*, pp. 1–6, 2016.

[120] M. H. Oreaba, “SHAPE Demo [Baselibe + CBS solver],” 2017. [Online]. Available: https://www.youtube.com/watch?v=YoZZLox8EwM. [Accessed: 18-Apr-2017].