Summer 8-2015

Pose Machines: Estimating Articulated Pose in Images and Video

Varun N. Ramakrishna
Carnegie Mellon University

Doctoral Dissertation

Pose Machines

Estimating Articulated Pose in Images and Video

CMU-RI-TR-15-27

Varun Ramakrishna
vramakri@cs.cmu.edu

The Robotics Institute
Carnegie Mellon University

Submitted in partial fulfillment
of the requirements of the degree of
Doctor of Philosophy in Robotics

August 2015

Doctoral Committee:

Yaser Sheikh, *Carnegie Mellon University* (Chair)	yaser@cs.cmu.edu
Takeo Kanade, *Carnegie Mellon University* (Chair)	tk@cs.cmu.edu
Drew Bagnell, *Carnegie Mellon University*	dbagnell@ri.cmu.edu
Deva Ramanan, *Carnegie Mellon University*	deva@andrew.cmu.edu
Andrew Fitzgibbon, *Microsoft Research, Cambridge*	awf@microsoft.com

Abstract

The articulated motion of humans is varied and complex. We use the range of motion of our articulated structure for functional tasks such as transport, manipulation, communication, and self-expression. We use our limbs to gesture and signal intent. It is therefore crucial for an autonomous system operating and interacting in human environments to be able to reason about human behavior in natural, unconstrained settings. This requires reliably extracting compact representations of behavior from possibly noisy sensors in a computationally efficient manner. The goal of this thesis is to develop computational methods for extracting compact keypoint representations of human pose from unconstrained and uncontrolled real-world images and video. Estimating articulated human pose from unconstrained images is an extremely challenging task due to complexity arising from the large number of kinematic degrees of freedom of the human body, large appearance and viewpoint variability, imaging artifacts and the inherent ambiguity when reasoning about three dimensional objects from two dimensional images.

A core characteristic of the problem is the trade-off between the complexity of the human pose model used and the tractability of drawing inferences from it: as we increase model fidelity by either incorporating structural and physical constraints or making fewer limiting assumptions, the problem of searching for the optimal pose configuration becomes increasingly difficult and intractable. Cognizant of this trade-off, in this thesis, we develop methods to reason about articulated human pose from single images by developing a modular sequential prediction framework called a *Pose Machine*. Pose Machines reduce the structured prediction problem of articulated pose estimation to supervised multi-class classification. The modular framework allows for integrating the latest advances in supervised prediction, incorporates informative cues across multiple resolutions, learns rich implicit spatial models by making fewer limiting assumptions, handles large real-world datasets, and can be trained in an end-to-end manner. Additionally we develop methods for estimating pose from image sequences and reconstructing pose in three dimensions by finding tractable substructures to incorporate physicial and structural constraints while maintaining tractability.

Acknowledgements

I would like to thank my advisors, Yaser Sheikh and Takeo Kanade, for teaching me everything I know about research. Their incredible patience, their ideas of what research is about, and their insistence on excellence and on playing the long game will stay with me for a lifetime. Their impact on my growth and thinking will fill pages. I hope it suffices to say: it has been an honor.

I am very grateful to my other committee members. Drew Bagnell for being a great teacher and collaborator, I have always come away from a discussion with you with a deeper understanding. Andrew Fitzgibbon, whose amazing mentorship during my summer at Microsoft Research Cambridge will remain a huge influence, and Deva Ramanan for his valuable advice and kind words of encouragement.

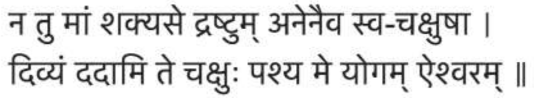
I originally came to Carnegie Mellon with the goal of pursuing research in speech recognition but—in what turned out to be a life-changing decision—I serendipitously enrolled in Martial Hebert's computer vision class. Thanks Martial, for being my first mentor at the RI and taking a chance on a fresh-off-the-boat, wet-behind-the-ears master's student all those years ago.

I would like to thank my collaborators over the years. Some of the important ideas in this thesis were developed during a train ride through Italy with Daniel Munoz. Thanks, Dan, for being a great collaborator and critic, the next Yuengling is on me. I would also like to thank Shih-En Wei, Kenny Marino, Daniel Huber, Kris Kitani, Hamid Izadinia, Michael Devyver and Dhruv Batra. Special thanks to Jamie Shotton for his mentorship and giving me the opportunity to work at Microsoft Research and Leonid Sigal for his mentorship at Disney Research. I would also like to thank all the members of the Perceptual Computing Lab including Hyun Soo Park, Hanbyul Joo, Tomas Simon, Eakta Jain, Yair Movshovitz-Attias, Minh Vo, Natasha Kholgade, Tim Godisart and Sean Banerjee. Thanks to the computer vision group members, past and present, for all the discussions, feedback, and friendship. Thanks, Dan, Abhinav I & II, Saurabh, Ishan, Aravindh, David, Scott, Ed, Narek, Jack, Kit, and everyone else. The

Robotics Institute provides an amazing environment for being a graduate student. Particular thanks to Suzanne Lyons Muth, Jessica Butterbaugh, Yukiko Kano, Suzette Gambone Rachel Burcin, and everyone else at the RI for making being a graduate student at the RI an absolute pleasure.

My time at Carnegie Mellon was amazing thanks to the company of some truly wonderful people. Thanks to Tomas Simon, Debadeepta Dey, and Jiuguang Wang for all the good times. Shout out to everyone who passed through the *Youth Hostel*. Satyajith Amaran, Utsav Prabhu, Aranya Venkatesh, Supreeth Achar, you guys were my family away from home. Thanks to everyone at the CMU Quiz Club, past and present, for the endless entertainment and great company. Thanks to Nishtha, Keshav, Chitra, Vinod, Sid “Julian” Garg, Vyas, Vas, Apurva, Ashwati, Divya, Toups, Ji, Samrat, Udaya, Lavan, Venkat, Abhijeet, Maddali, Mike (he’s in my team), and Erle. Thanks to Pandu, Bhatta, DC, Pavan, Charlie, Pota, KVM (hat-tip for the Vishwaroopa Darshana reference), Kuba, Khaj, Ashank, and Vada for being my closest friends and brothers-in-arms.

I’d like to thank my family and extended family. Thank you, Usha Aunty and Shankar Uncle, for always providing a home away from home during my stay in the United States. Thank you, Harini, for your love and support. I am a better, happier, less chaotic person since I met you. Finally, this thesis is dedicated to my parents, brothers, and my wife Harini, without whose unconditional love, unwavering support and selfless sacrifice, none of this would be possible. Thank you for always believing in me.



न तु मां शक्यसे द्रष्टुम् अनेनैव स्व-चक्षुषा ।
दिव्यं ददामि ते चक्षुः पश्य मे योगम् ऐश्वरम् ॥

“Thine own eyes are insufficient,
I will grant you divine vision—behold!”[1]

[1] Chapter 11, Verse 8 of the Vishwaroopa Darshana, *Bhagavad Gita*

Contents

CHAPTER 1

Introduction

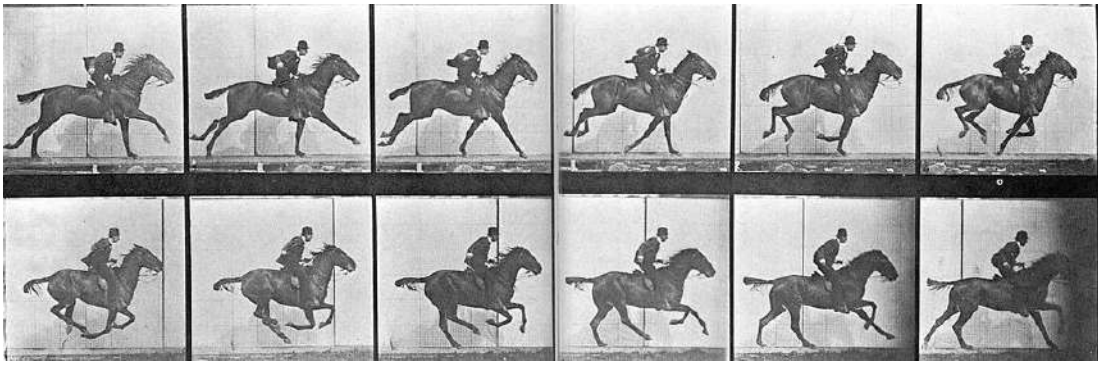
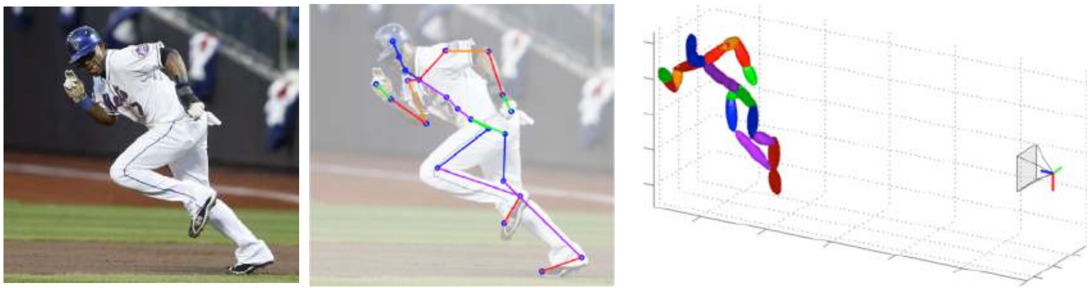


Figure 1.1: **The Horse in Motion**, *Eadweard Muybridge, 1899.* Muybridge settled a popular question of the day showing by means of photographs that a galloping horse indeed has all four hooves off the ground simultaneously, an event too fleeting to be captured by the human eye.

The visual study of natural articulated motion from images dates back at least to Muybridge [1899] and his study of human and animal locomotion. In possibly the first instance of visual inference of articulated motion from images, Muybridge conclusively proved that a galloping horse does indeed have all four hooves off the ground simultaneously (Figure 1.1), settling a popular debate of the day. Muybridge's [1899, 1901] early photographic studies showcased the rich range of complex motions that naturally occurring articulated structures can



Image

2D Anatomical Landmarks

3D Human Pose and Camera

(a) Input

(b) Output Representation

Figure 1.2: Our goal is to efficiently and accurately localize 2D and 3D joints from images and video.

execute.

Beyond answering questions regarding equine flight-worthiness, the visual study of articulated motion can be informative and revealing. In particular, because the articulated motion of humans is varied and complex; we use the range of motion of our articulated bodies for *functional* tasks, such as transport and manipulation, as well as for communication and self expression. When we interact, a large fraction of the information we convey is via non-verbal communication, using our limbs to gesture and signal intent. It is therefore *crucial* for an autonomous system, operating and interacting in cluttered and uncontrolled human environments, to understand human behavior in its natural setting. Reasoning usefully about such behavior requires efficiently extracting compact representations of the behavior from its possibly noisy sensing modalities, dealing with uncertainty in measurement, inherent power limitations, and a computational budget.

What constitutes a good representation for understanding natural articulated motion? We require that the representation (a) retains enough information such that desired task-specific, potentially higher level content can be inferred from it; (b) can be feasibly extracted from raw input (c) is compact and unambiguous, with few degrees of freedom. Simultaneously satisfying all these criteria is challenging because of trade-offs that arise between each of the desiderata. While one can conceive of detailed surface representations which capture every muscle movement and micro-expression, such a representation can be difficult to extract and unwieldy to perform down-stream computations with. Conversely, a representation such as a silhouette while allowing for higher-level reasoning, might be noisy and ambiguous. In this work we use a representation consisting of keypoints corresponding to anatomical landmarks on the articulated structure in 2D and 3D. Studies of the human visual perception system show that several higher-level percepts regarding behavior, intent and subject characteristics can be inferred from such a minimal representation[1]. The goal of this thesis is then to develop computational methods to automatically reason about the natural articulated motion and configuration of humans, from unconstrained images and image sequences by extracting such minimal keypoint representations. From an image, such as in Figure 1.2a, our goal is to localize the anatomical landmarks of the person in the image, assemble its configuration in three-dimensions, and position the camera at the relative vantage from where the image was captured. This task, while nearly effortless for humans, has proven to be a long-standing challenge for computers.

1.1 Scientific Challenges

The main challenges in estimating articulated human pose from images arise from the following sources of complexity: *kinematics*, *appearance*, *ambiguity* and

[1] In his seminal work, Johansson [1973] developed a minimal information display consisting of lights attached to a subject's joints to study the human perception of biological motion. From these *point light displays* alone, human observers have been shown to be able to infer higher level information such as arm movements [Pollick et al., 2001], American Sign Language [Poizner et al., 1981] identity [Perrett et al., 1985], gender ([Kozlowski and Cutting, 1977]), and the relative weight of lifted objects [Runeson and Frykholm, 1981].

imaging. We describe these sources of complexity in detail below.

Kinematics: The first challenge arises, in part, due to the large state space of articulated objects such as humans. The articulated structure results in a state space exponential in the number of kinematic degrees of freedom. For an articulated structure with d degrees of freedom and θ possible states, which could be locations in an image or discretized joint angles, there are θ^d possible configurations. As an example, in a simplified human body model with $d = 16$ degrees of freedom[2] and with each degree of freedom discretized into $\theta = 100$ states, we have 10^{32} possible configurations. Even with a conservative coarse estimate, we arrive at a large configuration space to reason over.

Appearance: The second challenge is due to the large variation in image evidence. The same articulated configuration can have varying appearance depending on local appearance factors such as clothing and skin color, and global appearance factors such as illumination, shadows etc. Additionally, the appearance of each part of an articulated object in an image is coupled with the configuration the object and the relative camera pose. Building upon our earlier counting argument, we can attempt to list the number of possible appearance states. Assuming a simple model with η_l states for local appearance properties, such as color and identity, and η_g for a global property such as lighting, this induces a total number of $(\theta\eta_l)^d\eta_g$ appearance states. As an example, setting local appearance $\eta_l = 10$ and global appearance $\eta_g = 10$, generates 10^{43} different appearances.

Ambiguity: In addition to the large variation, the monocular pose estimation problem is also riddled with ambiguities in geometry and appearance. *Geometrical ambiguity* arises because the problem of estimating the 3D configuration of points from their 2D projections is ill-posed, even when fitting

[2] By some estimates, the human body has up to 244 degrees of freedom [Zatsiorsky, 1998].

a known 3D skeleton[3]. *Appearance ambiguity* arises due to the fact that the human body has a bilateral plane of symmetry resulting in the symmetric appearance of parts on the left and right halves of the body. Additionally, in natural unconstrained environments with measurement noise and imaging artifacts, background clutter can often appear indistinguishable from the appearance of parts of the body and vice-versa.

Imaging: The process of projecting a scene onto the image plane results in a loss of information along the optical axis. In the pose estimation problem, this introduces complexity due to *self-occlusion* and *inter-person occlusion*, where parts of the same articulated structure or interacting articulated structures occlude each other along the camera axis. Reasoning about the presence of an occluded part is an extremely challenging task due to the absence of local evidence, and must be inferred only from context. The relative *viewpoint* of the camera with respect to the articulated object introduces additional complexity, as the number of possible appearances is multiplicative with the number of possible relative viewpoints.

In essence, the problem of articulated pose estimation could be distilled to one of *finding a valid configuration from an exponentially large number of possible configurations that explains ambiguous and uncertain image evidence.*

Approaches to tackle the articulated human pose estimation problem can be broadly separated into two phases. The first, **modelling**, usually takes the form of designing a scoring (objective) function that assigns each configuration a score, with plausible configurations being assigned higher scores and implausible configurations assigned lower scores. To design a scoring function, researchers have relied on approaches that involve scoring a configuration by measuring the

[3] As noted in [Lee and Chen, 1985], each 2D end-point of a limb subtends a ray in 3D space. A sphere of radius equal to the length of the limb centered at any location on one of these rays intersects the other ray at two points (in general) producing a tuple of possible 3D limb configurations for each location on the ray.

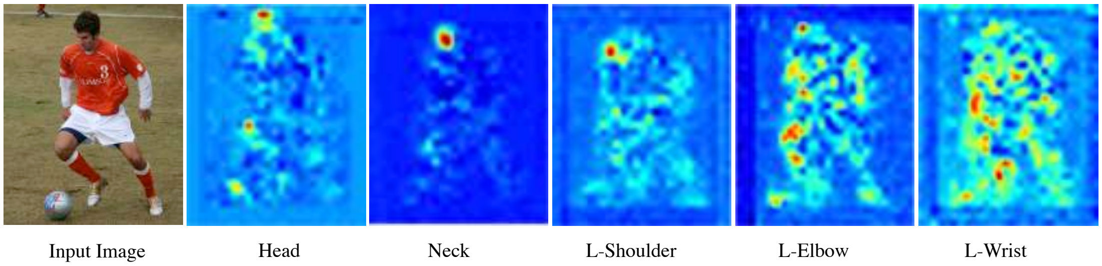


Figure 1.3: **Confidence maps from independent part detectors.** Parts with strongly discriminative appearance such as the head and shoulders have sharp peaks and unimodal confidence maps. Parts lower down in the kinematic chain of the human skeleton tend to be harder to detect due to large appearance variation.

agreement between learned visual appearance and the observed image appearance corresponding to the configuration being scored. A complete decomposition of the scoring function that reasons about each degree of freedom independently is usually not successful. In Figure 1.3 we show the confidences for detecting each part in an image independently using local appearance, we see that the confidence maps are noisy, ambiguous, with a large number of false positive detections. Relying purely on local appearance is therefore not a viable strategy. However, as far back as [Helmholtz et al., 1909] it has been hypothesized that in addition to relying on learned appearance of objects, humans also rely on *common sense* physical and structural constraints (Table 1.1) to aid visual reasoning. Relying purely on appearance can allow implausible configurations such as the classical problem of double counting in 2D pose estimation where two limbs are allowed to occupy the same region in the image. In order to prevent such configurations, it becomes necessary to penalize body configurations that violate such physical and structural constraints by incorporating them into the model. In Table 1.1, we list a set of cues and constraints that assist in the task of estimating articulated human pose.

The second phase of estimating articulated human pose, **inference**, involves designing algorithms to efficiently score and search through the set of possible

Appearance Cues	Structural Constraints	Physical Constraints
Local appearance	Kinematic chain structure	Temporal consistency
Mid-level Appearance	Inextensibility of limbs	Mutual Exclusion
Global Appearance	Structural symmetry	Rigid deformation
Appearance Symmetry		

Table 1.1: Useful constraints and cues in the articulated human pose estimation problem.

configurations. When the scoring function is such that it decomposes over states, algorithms can be designed for searching through the state space in an efficient manner. As an example, when the problem of estimating articulated human pose is modelled as a *tree structured* conditional random field [Lafferty et al., 2001], efficient dynamic programming algorithms can be employed. In continuous state spaces, if the model is designed such that the scoring function has certain geometric structure (e.g., convexity), efficient and provably optimal optimization methods can be employed.

A core characteristic of the articulated human pose estimation problem is the *trade-off* that arises between the *complexity* of the model (scoring function) and the *tractability* of drawing inferences from it—the more complicated the model, the harder it becomes to find exact answers to the questions we ask of it. In discrete state spaces, this could be because the addition of certain constraints disallows the scoring function from decomposing over states, thus preventing the use of efficient search algorithms. In the continuous case, certain constraints or priors can result in a non-convex optimization problem with multiple local minima making exact and efficient optimization difficult.

This trade-off defines a spectrum in the approach to these problems. On the one hand, one could prefer simple models that trade accurate modeling for exact inference, and on the other hand we have approaches that prefer accurate modeling but operate with inexact or approximate inference.

1.2 Core Contributions of this Thesis

Cognizant of the trade-off described above, this thesis develops solutions to articulated pose estimation problems with two distinct approaches: (i) tightly coupling modeling and inference to side-step the complexity-tractability trade-off and (ii) developing models which incorporate physical and structural constraints with tractable substructures that enable efficient inference. We discuss our contributions in the context of these approaches:

(i) Tightly coupled modeling and inference for pose estimation: In the context of predicting the 2D locations of anatomical landmarks, a key observation is that the spatial context of a landmark provides a strong cue for predicting its location. Classical approaches such as the pictorial structure approach [Fischler and Elschlager, 1973; Felzenszwalb and Huttenlocher, 2005; Yang and Ramanan, 2013] aim to capture such correlations and spatial dependencies between the parts via a graphical model. However, inference in graphical models is difficult and inexact in all but the most simple models such as a tree-structured or star-structured model. These simplified models are unable to capture important dependencies and interactions between parts and lead to characteristic errors. Models that incorporate additional constraints are difficult to learn and perform inference with [Kumar et al., 2005]. When people interact in images, this problem is compounded exponentially. We side-step this trade-off between modeling complexity and tractability of inference by directly learning an inference procedure for predicting the location of landmarks. We develop an inference machine [Munoz et al., 2010; Ross et al., 2011] architecture called a *Pose Machine* consisting of a sequence of predictors trained to predict the location at each stage in the sequence mimicing the mechanics of message-passing inference in graphical models. The Pose Machine architecture (Chapter 4) provides a modular framework for implicitly modeling complex spatial dependencies between parts and reduces the structured prediction problem of pose estimation to a sequence of simple classification problems. The modular nature of the architecture allows us to incorporate the latest advances in supervised prediction, including deep convolutional networks. Using the top-down design philosophy of a pose machine we develop a convolutional architecture called a *Convolutional Pose Machine. Convolutional Pose Machines* achieve state-of-the-art results on benchmark datasets for monocular pose estimation due to several design features suggested by the pose machine architecture, such as intermediate supervision, inter-mixed multi-resolution cues, and large receptive fields. For the problem of interacting people, we develop *Dyadic Pose Machines* (Chapter 5) to parse the articulated pose of two interacting objects. The contributions described here were published in the following papers:

> Varun Ramakrishna, Daniel Munoz, Martial Hebert, James A. Bagnell, and Yaser Sheikh. Pose Machines: Articulated Pose Estimation via Inference Machines. *In European Conference on Computer Vision, 2014.*
>
> Shih-En Wei, Varun Ramakrishna and Yaser Sheikh. Convolutional Pose Machines: A Deep Architecture with Intermediate Supervision. *Under review at IEEE International Conference on Computer Vision, 2015.*

(ii) Incorporating structural constraints with tractable substructures: We examine how to design models with richer structural and physical constraints that allow for tractable substructures that enable feasible inference. Tracking articulated human pose in image sequences is challenging due to the symmetric appearance of human body parts and due to self occlusions. In Chapter 6, we show how to incorporate mutual-exclusion constraints that, prevent double counting of body parts, provides a representation for occlusions, and encourages temporally smooth part-tracks. This is achieved by greedily solving tractable subproblems that model the motion of parts with symmetric appearance, drawing from ideas in the multi-target tracking literature.

The problem of reconstructing human pose in 3D is challenging due to inherent kinematic complexity and ambiguity. In Chapter 7, given 2D $(x, y) \in \mathbb{R}^2$ locations for landmarks in an image, we develop algorithms for reconstructing the 3D pose ($(X, Y, Z) \in \mathbb{R}^3$ locations of each anatomical landmark) of the articulated structure (human skeleton). We develop a model that is able to represent a wide variety of actions by relying on a large motion-capture dictionary while incorporating anthropometric constraints that preserve plausible limb-lengths. We show that inference can be broken down into a series of tractable subproblems solved in succession. The contributions described in this section were published in the following papers:

> Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Reconstructing 3D Human Pose from 2D Image Landmarks. *In European Conference on Computer Vision, 2012.*
>
> Varun Ramakrishna, Yaser Sheikh, and Takeo Kanade. Tracking Human Pose by Tracking Symmetric Parts. *In IEEE Conference on Computer Vision and Pattern Recognition, 2013.*

1.3 Broad Impact

The models and methods developed in this thesis have the potential for contributions towards research and development in a wide array of fields.

1.3.1 Pose Estimation from Passive Sensors

Fast and reliable articulated human pose estimation enables a wide spectrum of applications. The success of the Microsoft Kinect, which leverages a depth sensor to perform pose estimation, is testament to how accurate and real-time articulated human pose estimation can impact human-computer interaction. However, the Kinect addresses only a small fraction of the scenarios in which people are imaged. Images on the internet, images captured from hand-held devices and millions of frames of archival footage are still overwhelmingly in 2D (RGB). In scenarios where passive sensing is the only option (as opposed to active sensing such as the structured light sensor in the Kinect), which could be either due to on-board power constraints or when the environment prevents active sensing[4], RGB cameras are by far the most widely used sensor. Enabling fast and accurate articulated human pose estimation to work in unconstrained environments from low-cost, low-power sensors will push the boundaries of how we interact with machines.

1.3.2 Autonomous Reasoning in Human Environments

Autonomous agents deployed in and sharing real-world human environments need to reason and interact with humans. An autonomous car operating on and sharing roads with human drivers will need to understand a police officer's hand gestures and follow an indicated direction, or detect and understand that a cyclist's outstretched arm points to his intended direction of motion. In these scenarios, efficiently extracting a compact and accurate representation for the human's behavior is crucial to reacting in a safe and timely manner. Urban human environments are cluttered, unstructured and result in noisy sensor measurements. Performing articulated pose estimation in such scenarios will require the method to be robust to heavy background clutter and widely varying illumination. In this thesis, we develop and demonstrate methods that are robust to wide variation in background, illumination, and individual appearance. The key to achieving this is the fact that we are able to reduce the structured prediction problem to supervised prediction and then leverage powerful classification algorithms with high-inductive bias that are capable of learning sophisticated decision bound-

[4] In the case of outdoor sunlit environments, the sun's infrared emissions wash out the projected IR patterns from structured light sensors such as the kinect. Sensors which can project patterns bright enough to be sensed even in sunlight have untenable power requirements [Gupta et al., 2013].

aries. The methods developed in this thesis for efficient pose estimation from monocular images will enable reasoning about human motion, action, and intent in traditionally difficult urban human environments.

1.3.3 Understanding Social Scenes

A large fraction of the information conveyed when people interact is non-verbal. The body language of interacting people communicates emotion, intent and its study from video can reveal useful insights about behavior. Park et al. [2012, 2013] study the social dynamics of interacting people from ego-centric video, but limit their representation to gaze. Accurate pose estimation for multiple interacting people will advance our understanding of social dynamics by providing richer representations and reconstructions of such scenes. Current indoor pose estimation algorithms that rely on depth sensors make restrictive assumptions on the relative vantage of sensor with respect to the sensor. Inter person occlusions and interacting actors also further degrade performance. The methods developed in this thesis has the potential to overcome such limitations furthering research in this area.

1.3.4 Immersive Virtual Experiences

Virtual reality is moving from the pages of science fiction into everyday use and is beginning to show promise as a new medium of communication. True immersion in a virtual environment requires high fidelity in the behavior and motion of a virtual avatar for which accurate human pose estimation is critical. Traditionally this has only been possible with expensive motion capture setups such as the Vicon [Peak, 2005] system, or more recently, using active sensors such as the Microsoft Kinect$^{\text{TM}}$. Additionally, pose estimation when multiple actors interact in an environment can be challenging. The methods developed in this thesis for performing pose estimation from passive sensors have the potential to enable portable and immersive virtual reality in a wide variety of environments, and in complex multi-actor settings.

CHAPTER 2

Background

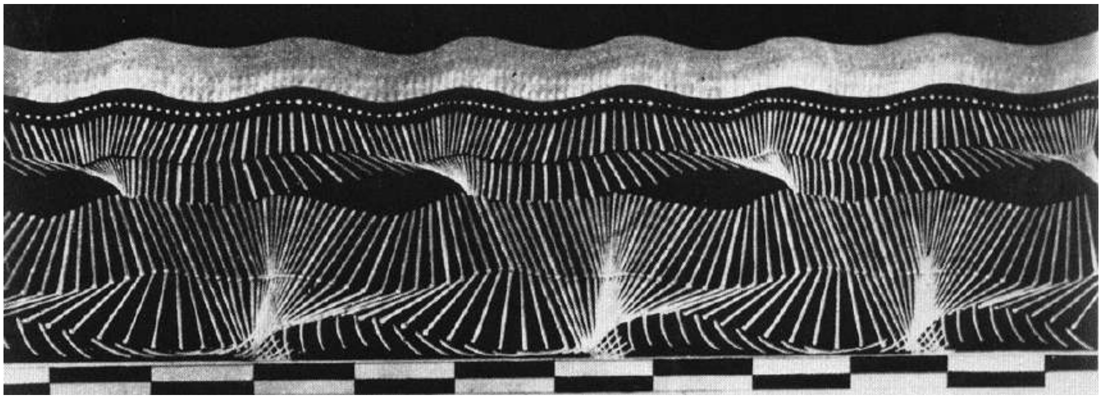


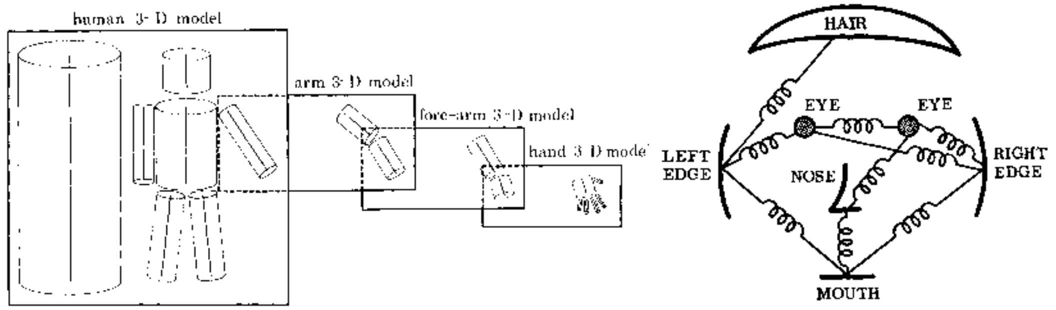
Figure 2.1: Joinville Soldier Walking, *Etiénne-Jules Marey, 1883.* Marey developed the technique of *chronophotography* to study the locomotion and physiology of human skeletal motion. Marey dressed his subject in a black suit with reflective tape attached to the clothing between the subject's joints.

The earliest visual studies of natural articulated motion were performed concurrently by Muybridge (see Figure 1.1) and the French physiologist Etiénne-Jules Marey. Marey's [1878, 1895] fascination of the human form and its biomechanics led him to develop a new photographic technique, *chrono-photography*, to capture spatial-temporal aspects of human skeletal motion. By dressing his subjects in black velvet suits and marking joint positions in white, he captured multiple instances of human motion on a single photographic plate clearly highlighting spatio-temporal skeletal motion (see Figure 2.1). Using this technique, Marey was able to gain previously unknown insights into articulated motion such as the motion of the center of gravity during complex body motions such as gymastics, the distribution of force when gymnasts land in different configurations and the positions of body parts and their effects on balance during complex motions.

The modern successor of Marey was the Swedish psychologist, Johansson [1973], who devised the dynamic point-light display[1] technique where actors were recorded wearing illuminated markers on the joints and head of the body while performing simple activities, such that only the illuminated markers were visible in the recording. Johannson was able to complellingly show that when human subjects were shown frames of the recording in quick succession they were able to easily discern the action being performed by the actor, but found it impossible to impute the configuration from a single frame suggesting the importance of motion in action understanding and the reliance on "learned" motion representations.

The core ideas in many current approaches for automatically understanding articulated objects from visual input trace their origins to theories regarding the representation and recognition of objects in the human visual perception system and early efforts to replicate them computationally. One of the essential ideas, that of representing an object by a collection of visual primitives or *parts*, is central to many current computational methods for understanding articulated pose. Proponents of Gestalt theory [Koffka, 1935], a movement in psychology, posited

[1] Point-light displays (PLD) revolutionized the study of human perception of biological motion. Research in the visual perception of biological motion has shown that many complex actions can be recognized solely from PLDs including facial expressions [Bassili, 1978], arm movements [Pollick et al., 2001], and American Sign Language [Poizner et al., 1981]. From only PLDs, subjects could infer identity [Perrett et al., 1985], gender ([Kozlowski and Cutting, 1977]), and the relative weight of lifted objects [Runeson and Frykholm, 1981]. We point the interested reader to [Blake and Shiffrar, 2007; Giese and Poggio, 2003] for a comprehensive review and summary of current findings in the area.



(a)

(b)

Figure 2.2: (a) Marr and Nishihara [1978] proposed an object-centred hierarchical model for the representation and recognition of a complex three-dimensional object. The model consists of a modular collection of volumetric primitives arranged in a hierarchy with increasing detail. (b) Fischler and Elschlager's [1973] pictorial structure model consists of a collection of *parts* arranged together with spring-like constraint relationships between them.

that the brain perceives objects as a whole by the *perceptual grouping* of visual elements that obey the principles of *proximity*, *similarity*, *symmetry* and *simplicity*. Marr and Nishihara [1978] proposed a hierarchical model composed of volumetric primitives for the object-centric representation of biological forms (see Figure 2.2a) and provided one of the first plausible computational models for performing inference of such a representation from images. Biederman's [1987] theory of *recognition-by-components* in the human visual system was based on assembling geometric primitives called "geons" such as generalized cylinders [Binford, 1971], blocks [Roberts, 1963], or ellipsoids [Pentland, 1986].

There is an extensive variety of work[2] in the computational understanding of human action, posture and behavior from images and image sequences. In this chapter, we provide a review of approaches to the problem of articulated pose estimation, where the goal is to automatically localize the positions of joints in an image and estimate their configuration in three dimensions.

[2] We point the reader to Gavrila [1999], Moeslund et al. [2011] and Forsyth et al. [2006] for comprehensive reviews of work in the visual analysis of humans

Some of the earliest work is by Fischler and Elschlager [1973] who provided the first computationally feasible and demonstrable algorithm for the understanding of complex objects in images. They developed the *pictorial structures* representation for complex objects which decompose the appearance of the object into a set of visual components or "parts", linked to each other via spring-like constraints on their relative deformations. Fischler and Elschlager [1973] also provided dynamic programming formulation that could be utilized to efficiently compute the optimal configuration of each of the parts that explains image evidence. In the following sections we discuss prior work in pose estimation in the context of monocular images, image sequences and approaches to dealing with the difficult case of multiple interacting articulated objects.

2.1 Pose Estimation from Monocular Images

In the pictorial structure or *parts-based* paradigm for articulated pose estimation, the goal is to localize joint locations or anatomical landmarks on the human body from a single image I. This is usually formulated as the problem of finding a configuration for each part $i \in \mathcal{V},\ \mathcal{V} = \{1, \ldots, P\}$, where P is the number of parts, that maximizes an objective function:

$$S(\mathbf{y}, I) = \sum_{i \in \mathcal{V}} \phi_i(y_i, I) + \sum_{(i,j) \in \mathcal{E}} \phi_{ij}(y_i, y_j, I), \tag{2.1}$$

where $\mathbf{y} = (y_1, \ldots, y_P) \in \mathcal{Y}$, is the configuration of the i^{th} part in the image, $\mathcal{Y}$ is the set of possible configurations in the image and $\mathcal{E}$ is a set of edges that link parts. The *appearance* term ϕ_i, scores the local image evidence of placing part i at the location y_i, while the *structure* term ϕ_{ij} scores the relative placement of parts i and j at the locations y_i and y_j respectively. The collection of parts $\mathcal{V}$ and the linkages between them $\mathcal{E}$ form a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Estimating the configuration

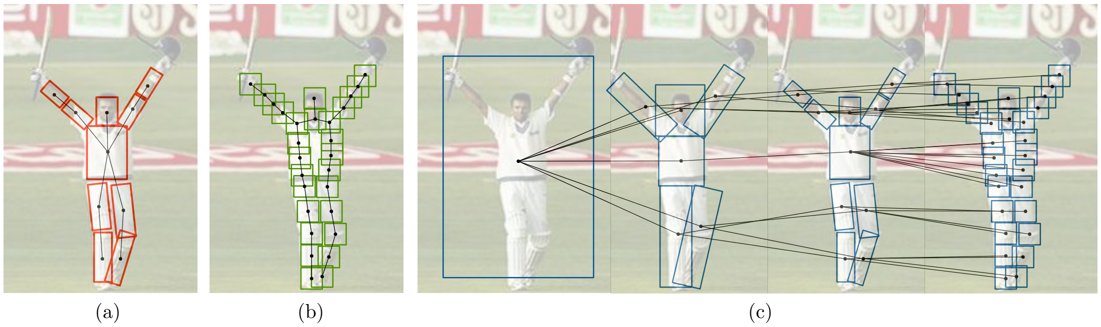


Figure 2.3: **Tree structured models.** (a) The tree-structured pictorial structure model of Felzenszwalb and Huttenlocher [2005] and extended by Andriluka et al. [2009]; Pishchulin et al. [2012] uses rectangular templates to model parts and a state representation that models the location, scale and orientation of each rectangular part, (b) the tree structured deformable parts model of Yang and Ramanan [2011] uses flexible templates around keypoints instead of limbs, and (c) a hierarchical tree model arranges parts in a hierarchy of increasing detail, versions of which have been used by [Sun and Savarese, 2011; Wang et al., 2011; Duan et al., 2012; Zhu et al., 2008].

of parts thus requires modeling and reasoning about both appearance (given by the choice of ϕ_i) and structure (the form of ϕ_{ij} and the topology of $\mathcal{G}$). We review related work in the context of design choices and models for appearance and structure.

2.1.1 Models for Structure

Tree Structured Models

The seminal work of Felzenszwalb and Huttenlocher [2005] built on the original pictorial structures [Fischler and Elschlager, 1973] and introduced the first practical algorithms for matching a pictorial structure approach to 2D images. The model expresses the articulated structure of the human body as a tree-structured graphical model with kinematic priors that couple connected limbs. By using a parametric quadratic function to model pairwise spatial term (ϕ_{ij}) that link coupled limbs they leverage efficient distance transforms [Felzenszwalb and Huttenlocher, 2004] to speed-up inference of the optimal configuration that explains image evidence.

Building on the success of the pictorial structures model, several approaches adopt a tree-structured graphical model to represent human pose. [Ramanan, 2007; Buehler et al., 2008; Andriluka et al., 2009; Pishchulin et al., 2013b] improve upon the original pictorial structures formulation by using expressive predictors for part detection and develop increasingly sophisticated models for kinematic relationships between parts. [Andriluka et al., 2009] use a parametric Gaussian distribution in a local co-ordinate frame to model the relative deformation of connected parts. Their model can be categorized as a *loose-limbed* model first introduced in [Sigal and Black, 2006], where limbs are not rigidly attached at joints allowing small deviations from the joint location. [Johnson and Everingham, 2010] showed improved pose estimation by clustering the training data by configuration and training a separate tree-model for each configuration cluster allowing for simpler spatial models in each cluster, but overall increasing expressivity.

Deformable Parts Model: The deformable part model [Felzenszwalb et al., 2008] derives from the pictorial structures formulation and has been one of the most successful methods for object detection. Objects are modelled as a collection of *parts* with spring constraints between adjacent parts. [Yang and Ramanan, 2011] extend the deformable parts model to perform articulated pose estimation by introducing smaller *flexible* parts arranged in a tree structured model whose parameters are learned jointly in a max-margin learning framework. They differ from previous pictorial structure models in using a mixture model to model small regions around keypoints as the representation for parts instead of a rigid rectanglular template for each whole limb as in pervious models. This allows them to learn simpler spatial models and allows for greater flexibility. [Wang and Mori, 2008] use a mixture of tree models to deal with the problem of *double counting* where the same region in the image can be used to explain more than one limb.

Search Space Reduction: Evaluating complex image features such as segmentation boundaries, optical flow, etc., can be expensive. Many techniques aim to reduce the search space so that expensive feature computations are performed only at a few pruned image locations. Ferrari et al. [2008] use a weak model followed by a grab-cut segmentation algorithm to prune the search space. Sapp et al. [2010] use a structured prediction cascade of pictorial structure models that learns a thresholding function to prune the state space. The thresholding function is designed to be aware of the optimal global configuration of all the parts and does not naively use only local image evidence to prune states.

Image Dependent Modeling: Chen and Yuille [2014a] achieve impressive results by using a convolutional neural network to learn image dependent unary and pairwise potentials of the tree-structured model of Yang and Ramanan [2011]. The use of image-dependent priors can potentially constrain the search space further allowing for more accurate inference and has been employed in work by Pishchulin et al. [2013a]; Sapp et al. [2010, 2011].

Hiearachical Tree Models: Tree models have also been adapted to incorporate a hierarchical representations of appearance. Sun and Savarese [2011] use a tree structured hierarchy of parts from larger composite parts such as whole limbs in coarse levels of the hierarchy to smaller parts modelling the appearance around joints in finer levels of the hierarchy. Pishchulin et al. [2013a] condition the spatial priors for detection of anatomical landmarks on the detection of larger composite parts. Tian et al. [2012] use a hierarchical tree model with latent variables to represent composite parts.

Beyond Trees

Many of the methods above have been successful on images where all the limbs of the person are visible, but are prone to characteristic errors such as double-counting image evidence, which occur because of correlations between parts, such as during self occlusion, that are not modeled by a tree-structured model. In order to get around the some of the deficiencies of tree models, many approaches propose to augment the tree structured model with edges to capture additional relationships between parts:

Occlusion Constraints Wang et al. [2011] use a hierarchical poselet representation with a non-tree structured graph but require the use of approximate loopy belief propagation for inference. While inference is exact in their model they rely on efficient bound computation which increases in complexity with the number of loops in the model. Sigal and Black [2006] use per-pixel hidden binary variables to encode occlusion relationships between parts resulting in a loopy graph and perform inference using a non-parametric variant of belief propagation.

Symmetric Part Constraints: Jiang and Martin [2008] incorporate mutual exclusion constraints by eschewing a graphical model and framing inference as a max-flow linear integer programming problem over a reduced number of locations for each part produced by an earlier state-space reduction step. Tian and Sclaroff [2010] use a loopy model enforce appearance symmetry constraints and uses a branch and bound algorithm to perform inference.

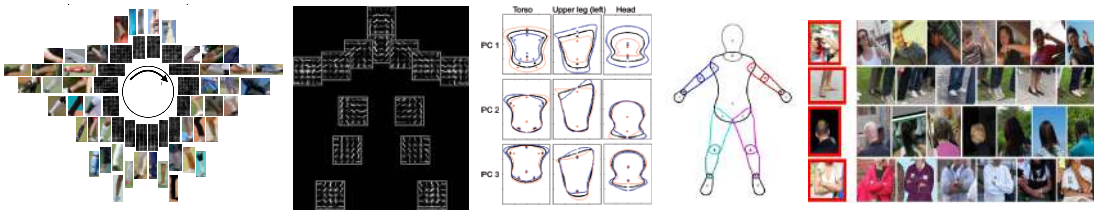
Limb Co-ordination Constraints: Limbs move in a co-ordinated fashion. While this information can be captured by tree-structured graphs, often weak local image evidence can prevent information flow. In order to accurately model limb co-ordination, edges between parts not connected in the standard tree structure need to be modeled. Lan and Huttenlocher [2005] use a common-factor model that augment the tree-structure with an additional latent factor that accounts for co-ordination between limbs. Kiefel and Gehler [2014] introduce the field of parts model where the presence or absence of each part at every image location is modeled with a large number of binary variables. Efficient inference in this large, loopy, model is performed by using high-dimensional gaussian filtering on a mean-field approximation of the model.

These models are usually difficult to perform inference on and rely on approximate inference methods at learning and test time. Moreover, the above tree and non-tree models usually involve some degree of careful modeling—for example, Andriluka et al. [2010] models deformation priors either by assuming a parametric form for the pairwise potentials or as in Yang and Ramanan [2011] who restrict the appearance of each part to belong to a mixture model. These trade-offs are usually required to allow for tractable learning and inference. Even so, learning the parameters of these models usually involves fine-tuned solvers or approximate piecewise methods.

Implicit Models for Structure

Implicit shape models eschew a parametric model of human shape for an implicit model that learns a spatial model directly from data. Some of the drawbacks of explicitly modelling articulated structure arise from the fact that simple parametric models are often ill-suited for modeling the complexity of the full range of human deformation. The choice for the particular parametric form is usually motivated largely by the ease of performing inference rather than the suitability of the model for capturing the variation in the data. Implicit models aim to overcome these drawbacks by not committing to a particular parametric form and by learning the correlations between parts in a data-driven manner using an expressive learning algorithm.

Bai and Tu [2009] introduced the *auto-context* algorithm for capturing contextual information via the use of context from neighboring pixel classifiers for



(a) Rectangular Templates (b) Parts (c) Contour Models (d) Poselets

Figure 2.4: **Appearance Models.** (a) Pictorial structure models traditionally have used rectangular templates, with a separate template modeling each discretized rotation state (Image courtesy, [Pishchulin et al., 2013a].), (b) the deformable part model of Yang and Ramanan [2011] use gradient histograms in patches around keypoints, (c) the deformable structures model of Zuffi et al. [2012] model the appearance of each part with a PCA contour model (Image courtesy, [Zuffi et al., 2012].) (d) poselets model mid-level patches which cluster together in appearance and configuration space (Image courtesy, [Bourdev and Malik, 2009]).

a variety of computer vision tasks. Dantone et al. [2013] learn better part detectors by using a sequence of multiple random forests. Models that attempt to learn a regressor that maps image features directly to a vector representing the articulated pose can also be thought of as performing implicit shape modeling. These methods have been popular in work that attempts to directly estimate 3D pose. Agarwal and Triggs [2004a] attempt to directly estimate 3D pose by using a relevance vector regressor on image silhouettes. Toshev and Szegedy [2013] recently used a standard deep convolutional architecture [Krizhevsky et al., 2012] to regress a vector of the 2D co-ordinates of anatomical landmarks.

2.1.2 Models for Appearance

Rigid Templates: The most popular method for capturing the appearance of parts of a articulated body model is the use of rigid rectangular templates. For models that use a limb and joint-angle based representation [Pishchulin et al., 2013b; Felzenszwalb and Huttenlocher, 2005; Andriluka et al., 2009], the parts correspond to a limb and use a rectangular template that is parameterized by, position, scale and orientation (see Figure 2.4a). Models that use a keypoint based representation[3] Yang and Ramanan [2011]; Chen and Yuille [2014a], use parts that are represented by a patch tightly cropped around each keypoint (see Figure 2.4b).

Contour Models: The work of Zuffi et al. [2012, 2013] introduced the deformable structures model that forgoes the traditional rigid rectangular template representation for parts in lieu of a more accurate contour model that accurately captures object boundaries (see Figure 2.4c). Points along the contour are represented as a linear combination of basis contour points that are computed using principal component analysis. The accurate contour model requires inference to be performed in a continuous domain, which is handled by using a non-parametric form of belief propagation. Sapp et al. [2010] also use contour support features in later stages of their cascaded model to provide richer image evidence for part location.

Poselet Models: While limbs and keypoints are semantically meaningful, they might not be best suited for detection. Combinations of parts that occur in a fixed configuration and therefore have a consistent discriminative visual structure can improve the probability of detection. The poselets [Bourdev and Malik, 2009] framework uses a mid-level representation that capture the appearance of geometrically consistent configurations (see Figure 2.4d). Mid-level representations for part appearance have also been used by Pishchulin et al. [2013a]; Wang et al. [2011]; Sun and Savarese [2011].

Global Models: Global models do not decompose the appearance of the articulated object into a collection of parts, instead try to learn a representation of the global appearance for regressing directly to matching exemplar in a dataset. Pose estimation is performed by transferring a smoothed pose estimate of the nearest matching exemplars. Popular features to capture global appearance have been silhouette features [Agarwal and Triggs, 2004b], shape context features [Mori and

[3] There is evidence in the cognitive neuroscience literature that suggests that points on joint locations are highly informative and provide a powerful representation.

Malik, 2006], HMAX features.

Learned Appearance: Many of the above approaches utilize gradient statistics, contour information or distance transform based features to encode the appearance of articulated objects. An alternative increasingly powerful approach is to directly learn a feature embedding for part appearance. The approaches by Taylor et al. [2010]; Tompson et al. [2014] and Jain et al. [2014] directly learn a representation for regions around anatomical landmarks using a deep convolutional architecture trained using backpropagation on a task-specific loss function.

2.2 Pose Estimation from Image Sequences

In image sequences, the temporal smoothness and articulated dynamics of natural motion provide additional useful cues for pose estimation. We discuss related work that use different approaches to incorporate temporal smoothness cues.

Tracking by Detection: In tracking by detection approaches, hypotheses for the pose are generated in each frame of the video independently followed by a data-association step which selects poses from each frame to form a complete pose track. One of the first models to track human pose over long natural image sequences was demonstrated by Ramanan et al. [2005]. A canonical pose is detected in the sequence from which the authors estimate a strong appearance model for each limb. Once an appearance model is obtained the authors follow a *tracking by detection* approach, reducing tracking to independent person-specific model detection in each frame.

In Park and Ramanan [2011], the model of Yang and Ramanan [2011] is sampled to obtain the N highest scoring diverse pose hypotheses in each frame, followed by dynamic programming routine to obtain a sequence of poses in an image sequence. The approach by Andriluka et al. [2008] detects people in each frame independently and uses a latent gaussian process dynamical model to infer the pose through the image sequence.

Graphical Model-based Approaches: Several methods adopt a graphical model approach to the problem and incorporate temporal continuity cues into the model via pairwise inter-frame potentials. The main challenge in this case is that the additional temporal edges in the graph between parts in adjacent frames introduce a large number of loops making inference slow and approximate. These approaches are also restricted to operate in a "batch" fashion. Ferrari et al. [2008] use loopy belief propagation in a simple temporal model where parts in successive frames are connected by temporal edges. Sigal and Black [2006] use non-parametric belief propagation with learned motion distributions for temporal edges. In Sapp et al. [2011], the authors use dual decomposition to perform inference where the loopy graph is broken up into a set of *slave* trees in which inference is feasible, and agreement is enforced via Lagrange multipliers.

Learned Dynamics: Several approaches learn dynamical models for human motion and use the learned dynamics to predict a configuration in the next frame given an estimate for the configuration in the current frame. Isard and Blake [1998] use 'factored sampling' along with a learned dynamical model to propagate distributions over position and shape through time. Approaches by Urtasun et al. [2006] learn a dynamical model directly from data using discriminative methods such as gaussian processes [Lawrence, 2004]. Instead of learning dynamical models from data, Brubaker et al. [2007] and Vondrak et al. [2008] use a detailed biomechanical characterization of human dynamics coupled with image based observation models to track human pose. A limitation of these models is that they usually require fairly accurate initialization and are prone to drift. In addition, human motion can be highly complex and learning a dynamical model from data that can generalize across human action, shape and identity remains a challenge.

Flow-based Filtering Models: In early work by Hogg [1983], projections of 3D primitives are tracked over image sequences. Bregler and Malik [1998] parametrized the kinematic tree of the human body using twists and exponential maps, and propagate pose estimates by performing least squares optimization to match measured and predicted flow of image gradients. Sheikh et al. [2008] use a Kalman filtering approach to tracking a set of articulated templates where predictions from a dynamical model are combined with detections from a per-frame body detector to obtain new location estimates.

2.3 Pose Estimation for Interacting Objects

When jointly reasoning about multiple interacting articulated objects in a scene, the state space for inference grows exponentially with the number of actors. Additionally, inter-person occlusions, non-canonical relative views and close proximity degrade standard pose estimation performance. A possible approach to dealing with the problem of interacting articulated objects is to reason about each object individually. Methods such as those proposed by Ghiasi et al. [2014] and Chen and Yuille [2014b] approach the problem by reasoning about each person individually. These methods reason about the pose of individuals by modeling local occlusions of body parts while remaining agnostic to the interacting individual. Reasoning about the joint configuration of the interacting object results in a more difficult inference problem. In work by Andriluka and Sigal [2012], interactions between individuals are modeled using a graphical model with additional connections between parts of the interacting objects, Eichner and Ferrari [2010] develops models for reasoning jointly about multiple people in a scene, by designing a graphical model that incorporates inter-person occlusion and inter-person exclusion terms. This results in a loopy graphical model, where inference is difficult. The authors either use a branch and bound approach to perform inference or approximate loopy belief propagation algorithms, which can potentially be intractable for models incorporating complex interactions and degrade as the number of interactions increase and the model becomes more “loopy”. Yang et al. [2012] study the problem of interacting people, but restrict their scope to the problem of detecting types of interactions, by fitting tree structured models trained for each interaction type and scoring the detections using the fitting error.

CHAPTER 3

Preliminaries

In this chapter, we provide a concise review of the tools from computer vision and machine learning that are employed in later chapters. Estimating articulated pose requires obtaining confidences for the location of each part in the image (represented by image patches around keypoints or rectangular limb templates). We begin by describing classic *sliding-window pipelines* for object detection and end-to-end *convolutional architectures*that perform feature learning directly from data. Articulated pose estimation requires the prediction of valid configurations of multiple parts simultaneously. We therefore review tools from machine learning for *structured prediction*, where the goal is to predict output objects that can be many-dimensional and possess certain regularities or *structure* in their output space.

3.1 Detection Pipelines for Rigid Objects

The standard object detection pipeline in computer vision consists of two stages: image **feature map computation** followed by **supervised classification** of

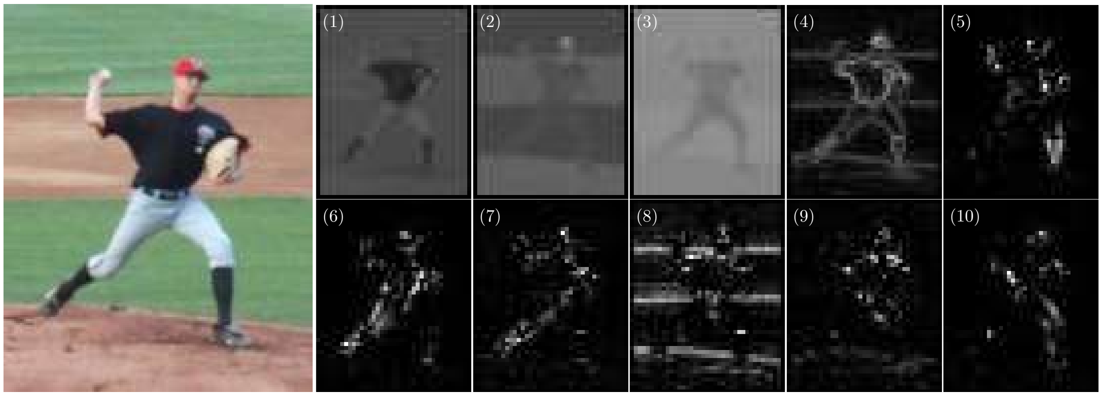


Figure 3.1: **Feature Maps**: On the left we show the input image, on the right are a set of feature maps computed from the image. The feature maps are registered to the image so that pixel correspondence can be established. Panels (1)-(3) correspond to the LAB feature maps, (4) gradient magnitude, (5)-(10) gradient orientation histograms for 6 orientation bins)

sliding windows. A common object detection pipeline as introduced by [Dalal and Triggs, 2005] uses a set of gradient histogram bin feature maps with a linear support vector machine classifier trained to indicate the presence or absence of an object at each location in the image. The pipeline used by [Viola and Jones, 2001] consisted of set of cascaded classifiers operating on haar features computed using an integral image of the original grayscale image. [Dollár et al., 2009] describe an object detection pipeline that consists of a set rich channel features followed by classifiers that operate on haar features, local sums and histograms that are computed efficiently using integral images.

Feature maps or *channel features* are non-linear transformations of the image registered so that a pixel location across the channel maps corresponds to the same pixel location in the image. Object presence can then be determined by a classifier that operates on each image location with spatial support on the feature maps, predicting a distribution over object classes, or determining the presence/absence of an object at that location.

3.1.1 Feature Maps

While there are a large number of different feature transformations that can be applied to an image, we describe a few that we use in later chapters for the task of part detection. We closely follow the feature channels as presented in [Dollár et al., 2009] as they provide a diverse set of informative features that are fast to compute.

LAB Colorspace: The RGB image is non-linearly transformed into the LAB color space that consists of a luminance (L) channel and two color channels (AB). The LAB colorspace is designed to approximate human vision and represent a color space that is perceptually uniform, i.e., perceptually similar colors occur closer together in this color space.

Gradients: Object boundaries and edges manifest as discontinuities in intensity values across the image plane. The magnitude and orientation of gradient of the image thus provide strong cues for object detection. Gradients can be computed by applying a set of image filters to a Gaussian smoothed image. The image filters f_x and f_y measure image differences in the x and y directions at an image location $z = (u, v) \in \mathbb{R}^2$. Convolving the filters with the image I results in the gradient responses,

$$g_x = I * f_x, \quad g_y = I * f_y, \tag{3.1}$$

where $*$ denotes a convolution operator. The magnitude and orientation of an image gradient at a location in the image can be computed as $G(z) = \sqrt{g_x(z)^2 + g_y(z)^2}$ and $O(z) = \tan^{-1}(\frac{g_y(z)}{g_x(z)})$ respectively.

Histograms: Statistics of gradient responses in small image regions also provide strong cues for object shape and have been one of the most successful features [Dalal and Triggs, 2005] for object detection. The quantization and binning of gradient orientations in image cells also provides some degree of invariance to small deformations and translation of object shape. A gradient histogram channel for orientation θ can then be computed by binning the orientation responses into one of τ bins, followed by counting pixels with specified orientation $\theta \in [\theta_1 \ldots \theta_\tau]$:

$$H_\theta(z) = \frac{1}{|\mathcal{N}_\sigma(z)|} \sum_{\delta \in \mathcal{N}(z)} G(\delta) \cdot \mathbb{1}(O(\delta) = \theta) \tag{3.2}$$

where $\mathcal{N}_\sigma(z)$ is a set of image co-ordinates in a neighborhood of the image pixel location z of size determined by pooling scale σ.

Feature Map Computation as a Convolutional Architecture: An interesting point to note is that the standard feature map computation pipeline can be abstracted to be comprised of a linear convolutional filtering step as in Equation 3.1 followed by a point-wise application of a non-linear function as in the computation of the gradient magnitude and orientation and the quantization step, followed by an aggregation or pooling step as in Equation 3.2. We will see later in the chapter that convolutional architectures for feature learning will use a similar pipeline for feature learning, but differ in that the parameters for the filters and non-linear operations are learned from data.

In the *sliding window* object detection pipeline, at each location z in the image, features are collected from a window of size $s_x \times s_y$ corresponding to the size of the object from each feature map. The features are collected into a vector and supplied to a classifier which is trained to predict whether an object is present at each location. In the following section, we review a popular choices for performing supervised multi-class classification and discuss the trade-offs and issues associated with them.

3.1.2 Supervised Classification

Supervised classification is the task of finding a discriminative function that classifies the training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$, where $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ and $y_i \in \mathcal{Y} \subset \mathbb{R}$ represent the features and corresponding label respectively of the i^{th} sample.

Algorithm 1 `train_random_forest`$(\mathcal{D}, N)$

```
Input: $N$    //*Number of trees*
Input: $\mathcal{D}$    //*Training dataset*
for $i = 1 \ldots N$ do
    Sample subset $\mathcal{D}_i$ from dataset $\mathcal{D}$.
    $f_i$ = grow_tree$(\mathcal{D}_i, 0)$
end for
Return: Learned forest $\{f_i\}_{i=1}^{N}$.
```

Often, the task of supervised learning is formulated as one of minimizing the empirical risk $\mathcal{R}$ over a dataset $\mathcal{D}$ given by:

$$\mathcal{R}(f) = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}} l(f(\mathbf{x}_i), y_i) \tag{3.3}$$

where $l(\cdot, \cdot)$ is a function that measures the *loss* incurred by making the prediction $f(\mathbf{x}_i)$ for a sample with ground truth label given by y_i. The empirical risk minimization problem is then to find the optimal f^* satisfying the following:

$$f^* = \min_{f} R(f) \tag{3.4}$$

We discuss some popular choices for the predictive function f and algorithms used for learning.

Random Forests: Random forests are a powerful and versatile supervised prediction algorithm introduced by [Breiman, 2001]. A random forest consists of a collection of decision trees, each trained on a random subset of the data and whose predictions are averaged at test time. Random forests have been empirically shown to be one of the most versatile and consistently high performing supervised learners on a variety of supervised learning tasks [Caruana and Niculescu-Mizil, 2006] in general and computer vision tasks in particular as in [Shotton et al., 2013].

Training a random forest proceeds by first selecting subsets of the data to

Algorithm 2 `grow_tree`$(\mathcal{D}, d)$

```
Input: $\mathcal{D} = \{(\mathbf{x}_j, \mathbf{y}_j)\}_{j=1}^{|\mathcal{D}|}$    //Training dataset
Input: $d$    //Current depth
Initialize: root //Root node of subtree
if $|\mathcal{D}| < n_{min}$ or $d > d_{max}$ then
    root→is_leaf = true
    root→ $\mathbf{y}_o \leftarrow \frac{1}{|\mathcal{D}|} \sum_{j=1}^{|\mathcal{D}|} \mathbf{y}_j$
else
    $\mathcal{D}_L$, $\mathcal{D}_R$, split_idx, thresh ← find_best_split($\mathcal{D}$)
    root→split_idx ← split_idx
    root→thresh ← thresh
    root→left  ← grow_tree($\mathcal{D}_L, d+1$)
    root→right ← grow_tree($\mathcal{D}_R, d+1$)
end if
Return: root
```

train each decision tree on: a procedure known as *bootstrap aggregation* (see Algorithm 1). A decision tree is then trained in a recursive manner as outline in Algorithm 2. The dataset is recursively split at each node of the tree by choosing a threshold on a feature value such that splitting on that feature results in distributions for the left and right children that maximize a particular information criteria. Popular choices for the splitting criteria include maximizing information gain, minimizing co-variance of the split distributions or maximizing the purity of each of the splits. At test time, each data sample is propagated down each of the trees eventually reaching a leaf node in each of the trees. The predicted label distribution $\mathbf{y}_{\text{pred}}$ is obtained by averaging the distributions $\mathbf{y}_o^i$ stored at the resulting leaf of each of the trees:

$$\mathbf{y}_{\text{pred}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{y}_o^i \tag{3.5}$$

Random forest inference can be implemented to operate efficiently in parallel on a GPU by converting the recursive tree traversal into an iterative traversal as shown in [Sharp, 2008][1]. While, efficient algorithms for training random forests exist, the key bottleneck in scaling up training to handle datasets with millions of data points is the fact that the learning algorithm requires $O(nd)$ memory. While online variants [Saffari et al., 2009; Denil et al., 2013] have been proposed, their performance has not yet been demonstrated to be competitive with their batch variants.

Boosting as Functional Gradient Descent: The empirical risk minimization problem in Equation 3.3 can be thought of as finding the optimal function $f \in \mathcal{F}$, where $\mathcal{F}$ is a function space with a well defined inner product [Friedman, 2001]. In this view, optimization of this objective corresponds to performing gradient descent in the function space $\mathcal{F}$ where in each iteration the predictor is updated by taking a gradient step,

$$f_t \leftarrow f_{t-1} + \alpha_t \nabla \mathcal{R}(f_t), \tag{3.6}$$

where α_t is the step-size and $\nabla \mathcal{R}(f_t)$ is the functional gradient of the empirical risk function $\mathcal{R}$. The gradient $\mathcal{R}(f_t)$ of the empirical risk function is defined only at the data points in the dataset $\mathcal{D}$. This results in updates to the predictive function only at data points seen during training which can lead to overfitting. To mitigate this, a smooth function $h_t \in \mathcal{H}$ which is defined in all of $\mathcal{X}$, is fit to the gradient $\nabla \mathcal{R}(f_t)$, by either projecting the gradient into $\mathcal{H}$ or finding the $h \in \mathcal{H}$ that is closest to the functional gradient according to a chosen metric. The resulting function after T steps of gradient descent is given by:

$$f_T = \sum_{t=0}^{T} \alpha_t h_t. \tag{3.7}$$

[1] Current graphics processing units (GPU) possess a single instruction scheduler which results in algorithms with high amounts of branching to execute slowly. The transformation from a recursive formulation to an iterative formulation drastically reduces the number of branching instructions allowing maximal utilization of the parallelization capabilities GPUs offer.

The functional gradient descent view of empirical risk minimization captures different variants of *boosting* depending on the choice of hypothesis class $\mathcal{H}$ and loss function $l(\cdot)$ used. For e.g., the *Adaboost* algorithm of [Freund and Schapire, 1997] use an exponential loss function and decision stumps as the class of weak learners. General convex loss functions can be optimized using a sub-gradient method variant of the above as shown by [Grubb and Bagnell, 2011]. In later chapters, we use *Boosted Random Forests*, which are a powerful class of predictors which use random forests as the hypothesis class $\mathcal{H}$, and have been shown to have strong empirical performance on a variety of learning tasks [Caruana and Niculescu-Mizil, 2006].

3.1.3 Convolutional Architectures for Object Detection

Deep convolutional networks also known as convolutional neural networks were made popular by LeCun et al. [1989] and are inspired by several early neural architectures such as Fukushima's [1980] *Neocognitron*. These architectures have achieved stellar performance on a variety of classification tasks in computer vision as first demonstrated by Krizhevsky et al. [2012] and many others [Girshick et al., 2014; Szegedy et al., 2014; Sermanet et al., 2013] subsequently. Deep convolutional networks are attractive as they perform feature learning, extraction and classification jointly in a single framework which can trained in an end-to-end fashion. These architectures have the advantage of learning features, directly from the data, that are tuned to the task being performed by backpropagating gradients of the task objective.

A deep convolutional network is comprised of successive layers of convolutional linear filtering followed by the application of a point-wise non-linear function (see Figure 3.2). Let $I \in \mathbb{R}^{w \times h \times 3}$ be a 3-channel color image of width and height equal to w and h respectively, and $W_l^i \in \mathbb{R}^{k \times k}$ and $\mathbf{b}_l^i$ be the weights and bias of the i^{th} linear filter at a layer l. We denote by $\mathbf{X}_{l-1} \in \mathbb{R}^{w' \times h' \times K}$ the

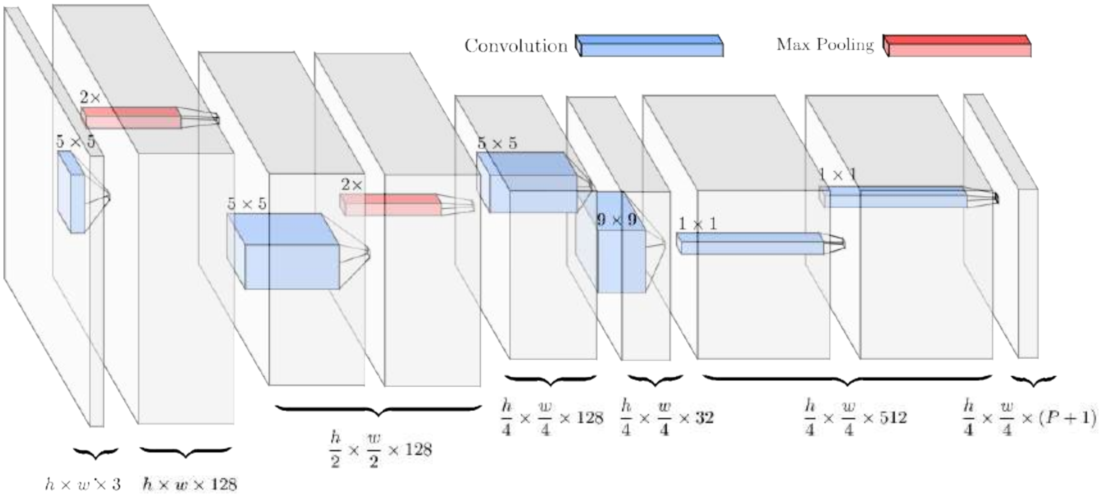

Convolution
Max Pooling
2×
5 × 5
5 × 5
2×
5 × 5
9 × 9
1 × 1
1 × 1
$h \times w \times 3$
$h \times w \times 128$
$\frac{h}{2} \times \frac{w}{2} \times 128$
$\frac{h}{4} \times \frac{w}{4} \times 128$
$\frac{h}{4} \times \frac{w}{4} \times 32$
$\frac{h}{4} \times \frac{w}{4} \times 512$
$\frac{h}{4} \times \frac{w}{4} \times (P+1)$

Figure 3.2: **Convolutional Architectures**: A deep convolutional architecture consists of a series of feature maps produced by a succession of filter convolution operations followed by the application of a point-wise non-linearity or a pooling operation. We show a *fully* convolutional architecture that produces dense pixel predictions for every location in the image.

collected set of responses or *feature maps* $\{X^i_{l-1}\}_{i \in (1 \dots K_{l-1})}$ to the K_{l-1} filters in the $l-1^{\text{th}}$ layer. The response at the l^{th} layer to the i^{th} filter is given as:

$$X^i_l = \sigma(\sum_j X^j_{l-1} * W^i_l + b^i_l), \tag{3.8}$$

where $\sigma(\cdot)$ is a non-linear function that is applied pointwise to each location. In the first layer, the filters are applied to the image, i.e., $\mathbf{X}_0 = I$. A popular choice for the non-linear function is the use of a clipped linear function called a rectified linear unit [Nair and Hinton, 2010], where σ is given by,

$$\sigma(u) = \max(u, 0) \tag{3.9}$$

In order to introduce a small degree of invariance to translations of the input, the feature responses are often aggregated and subsampled using a *pooling* layer. A popular choice for the pooling layer is the use of *max pooling*, which involves sliding a window across the feature map with a particular stride, and picking the maximum feature map response within each window, resulting in a new *subsampled* feature map (see Figure 3.2). Additional layers, such as *dropout* [Hinton et al., 2012] have been proposed to improve generalization. Dropout layers turn off units at random with probability $p = 0.5$ during training. The motivation is that during training, by turning off units at random, the network being learned is an average of several different sparser networks whose pathways are a subset of the full network thereby performing model averaging resulting in improved generalization.

In order to train the network, an objective function, $l(\mathbf{y}_n, \hat{\mathbf{y}}_n)$, can be defined in terms of the feature maps in the output layer L, $\mathbf{y}_n = \{X_L^j\}_{j=1}^{K_L}$, and desired outputs $\hat{\mathbf{y}}_n$ for the n^{th} image sample. The network is then learned by minimizing the average loss (empirical risk) across the training dataset, $\mathcal{D}$, given by:

$$\mathcal{L}_{\mathbf{W}}(\mathcal{D}) = \frac{1}{N}\sum_{n=1}^{N} l(\mathbf{y}_n, \hat{\mathbf{y}}_n), \tag{3.10}$$

where $\mathbf{W}$ is a vector of all the weights in all the layers of the network. Popular choices for the objective function are the euclidean loss function, soft-max regression loss or cross-entropy loss functions. Learning the parameters $\mathbf{W}$ of the network proceeds by performing gradient descent on the objective $\mathcal{L}_{\mathbf{W}}(\mathcal{D})$ as,

$$\mathbf{W}_t \leftarrow \mathbf{W}_{t-1} + \alpha_t \nabla \mathcal{L}_{\mathbf{W}}(\mathcal{D})$$

where α_t is a learning rate and the gradients $\nabla \mathcal{L}_{\mathbf{W}}(\mathcal{D})$ are computed using the *backpropagation* algorithm introduced by [Rumelhart et al., 1985] and later applied to convolutional networks by [LeCun et al., 1989]. For a simple and practical explanation on how to compute derivatives for deep architectures in general we point the interested reader to the introductory chapter of [Sutskever, 2013].

Object detection can be performed using convolutional architectures by training the network to produce, at the output layer, a set of confidence maps for different objects or background. The network is trained by supplying ideal-confidence maps $\hat{\mathbf{y}}_n$ with peaks in the confidence maps corresponding to the locations of the objects in the image I_n.

3.2 Structured Prediction

In the previous section we reviewed methods for localizing a single rigid object using a standard sliding window detection pipeline and alternatively by the use of a deep convolutional architecture. The problem of estimating human pose entails the joint detection of *multiple* parts of an object that can deform with respect to one another and is an instance of a *structured prediction* problem in computer vision. A structured prediction problem is a supervised learning problem in which the output space possesses underlying structure where only certain configurations of the output are valid.

While the precise definition varies according to domain, in computer vision, a structured prediction problem can be defined to be the problem of learning a predictor,

$$f(I) : \mathcal{I} \to \mathcal{Y}, \quad \mathcal{Y} \subset \mathbb{R}^d, \tag{3.11}$$

where $I \in \mathcal{I}$ is an image, $\mathcal{I}$ is the space of images and $\mathcal{Y}$ is a structured output space of dimension d. The problem of articulated pose estimation, is often framed as a structured prediction problem with output space $\mathcal{Y} \subset \mathbb{R}^{2P}$, corresponding to the (x, y) locations of P keypoints of the object. The output space of the articulated pose estimation problem is said to have *structure*, as only certain configurations of keypoint locations are considered valid. For example, the location of the head tends to co-occur with the location of the neck and shoulders in a subset of fixed configurations. A key challenge in such problems with large output spaces is the inference problem of searching over the exponential number of configurations for the optimal configuration.

Structured prediction techniques aim to learn a predictor that exploits the structure in the output space for reasoning about the exponentially many configurations. One of the most popular methods has been the use of a **graphical model**[2] which represents a joint probability distribution between the variables, represented as graph that encodes conditional independences. *Conditional random rields* have been the method of choice in the articulated pose estimation problem. A conditional random field is a discriminative undirected graphical model that learns a distribution over the output variables conditioned on the input variables. Inference in the graphical model setting involes finding the most probable configuration of variables or finding the marginal distributions of the joint distribution. The approach of using an **inference machine** [Munoz et al., 2010; Ross et al., 2011] for structured prediction eschews a probabilistic model for a sequential prediction approach that directly mimics the inference procedures used in standard graphical model based approaches. We provide a brief review of both these techniques in the context of articulated pose estimation.

3.2.1 Conditional Random Fields

In this section, we describe *conditional random fields* which are a class of undirected graphical models that has found wide application in computer vision. Conditional random fields model the conditional distribution between the output variables $\mathbf{y} = (y_1 \dots y_P), \quad \mathbf{y} \in \mathcal{Y}$ and the input features $\mathbf{x}$. In the articulated pose estimation problem each of the output variables $\mathbf{y}_i$ could refer to either the (u, v) location of a landmark or the joint-angle state of a limb. The conditional independences between the variables in the distribution are encoded in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} \in \{1 \dots P\}$ is the set of nodes for each variable and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$

[2] We point the reader to [Wainwright and Jordan, 2008] for a comprehensive overview of theory and algorithms for learning and inference in graphical models.

is the set of edges in the graph. The distribution can be expressed as a factorized Gibbs distribution, parametrized by θ, as follows,

$$P(\mathbf{y}|\mathbf{x};\theta) \propto \prod_{i \in \mathcal{V}} \psi_i(y_i|\mathbf{x}) \prod_{(i,j) \in \mathcal{E}} \psi_{ij}(y_i, y_j|\mathbf{x}). \tag{3.12}$$

where $\psi_i(\cdot)$ is a unary *compatibility* or *potential* function associated with each y_i and $\psi_{ij}(\cdot,\cdot)$ is a potential function associated with pairs of variables. The potential functions are constrained to be positive functions, and therefore are convenient to express as exponentials,

$$\psi_i(y_i|\mathbf{x}) = \exp\{-E_i(y_i;\theta_i)\}, \quad \psi_{ij}(y_i, y_j|\mathbf{x}) = \exp\{-E_{ij}(y_i, y_j;\theta_{ij})\} \tag{3.13}$$

where E_i and E_{ij} are functions parameterized by θ_i and θ_{ij} respectively. Note the dependence on the input $\mathbf{x}$ is implied, but dropped for notational convenience. The distribution in Equation 3.12 can then be expressed as $P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z}\exp\{-E(\mathbf{y},\mathbf{x};\theta)\}$ where Z, the partition function, serves to normalize the distribution and is given by, $Z = \sum\limits_{\mathbf{y} \in \mathcal{Y}} \exp\{-E(\mathbf{y},\mathbf{x};\theta)\}$. The energy function E is then given by,

$$E(\mathbf{y},\mathbf{x};\theta) = \sum_{i \in \mathcal{V}} E_i(y_i;\theta_i) + \sum_{(i,j) \in \mathcal{E}} E_{ij}(y_i, y_j;\theta_{ij}) \tag{3.14}$$

Thus, every conditional random field is associated with an *energy* function as in Equation 3.14. In this interpretation, the unary terms E_i can be thought of functions that measure compatibility between the input $\mathbf{x}$ and a particular choice for the variable y_i, with a low energy indicating high compatibility. The pairwise terms measure the compatibility between choices for pairs of variables y_i and y_j. In the context of the articulated pose estimation problem, the unary terms E_i measure the compatibility of the placement of a part i at a location y_i and can be obtained using the confidence maps from a detection pipeline for each part as described in Section 3.1. The pairwise term E_{ij} measures the compatibility of placing parts i and j at locations y_i and y_j respectively and is often modeled with a quadratic function that penalizes large displacements.

Learning: A natural objective for learning the parameters θ of the distribution in Equation 3.12 is that of **maximizing the likelihood** of the data $\{(\mathbf{x}^k, \mathbf{y}^k)\} \in \mathcal{D}$ under the distribution with parameters θ. This can also be written as equivalent to minimizing the negative log likelihood $\mathcal{L}_\theta$:

$$\theta^* = \arg\max_\theta \prod_{(\mathbf{x}^k, \mathbf{y}^k) \sim \mathcal{D}} P(\mathbf{y}^k | \mathbf{x}^k; \theta) = \arg\min_\theta \mathcal{L}_\theta(\mathcal{D}). \tag{3.15}$$

The gradient of the negative log likelihood with respect to the parameters can be shown to be given by,

$$\frac{\partial \mathcal{L}_\theta(\mathcal{D})}{\partial \theta_i} = \sum_{\mathcal{D}} P(y_i | \mathbf{x}^k) E_i^{'}(y_i; \theta_i) - \sum_{\mathcal{D}} E_i^{'}(y_i^k; \theta_i) \tag{3.16}$$

$$\frac{\partial \mathcal{L}_\theta(\mathcal{D})}{\partial \theta_{ij}} = \sum_{\mathcal{D}} P(y_i, y_j | \mathbf{x}^k) E_{ij}^{'}(y_i, y_j; \theta_{ij}) - \sum_{\mathcal{D}} E_{ij}^{'}(y_i^k, y_j^k; \theta_{ij}) \tag{3.17}$$

The learning rule given above aims to update parameters by moving in a direction that reduce the difference between the expected value of the energy terms under the model parameters and the empirical value of the energy terms. Computing the derivatives with respect to the parameters requires performing *marginal inference* to compute the marginal distributions $P(y_i, y_j | \mathbf{x}^k)$ and $P(y_i | \mathbf{x}^k)$ which can be NP-hard for general graphs. To deal with this, many approaches compute approximate marginals using loopy belief propagation or MCMC sampling based techniques.

An alternative objective for learning is the **maximum margin** learning objective. In this approach set forth by [Taskar et al., 2003] and [Tsochantaridis et al., 2005], the objective is to maximize the margin between the energy of the ground truth label for each sample $\mathbf{y}^k$ in the dataset $\mathcal{D}$ and all other possible labellings $\mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}^k$. This can be written as the constraint set:

$$E(\mathbf{y}^k; \theta) \leq E(\mathbf{y}; \theta) - \Delta(\mathbf{y}, \mathbf{y}^k) + \xi, \quad \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}^k,\ \xi > 0 \tag{3.18}$$

where $\Delta(\mathbf{y}, \mathbf{y}^k)$ measures the structured loss between the configuration $\mathbf{y}$ and the ground truth. The number of constraints in the set above is exponential, but can be replaced by a single constraint per sample in a formulation known as margin-scaling by observing that $\xi = \max_{\mathbf{y}} \Big(E(\mathbf{y}^k; \theta) - E(\mathbf{y}; \theta) + \Delta(\mathbf{y}, \mathbf{y}^k)\Big)$. The resulting optimization problem is then:

$$\min_{\theta} \ \frac{1}{2}\|\theta\|_2^2 + \sum_{(x^k, y^k) \in \mathcal{D}} \max_{\mathbf{y}} \Big(\Delta(\mathbf{y}, \mathbf{y}^k) - E(\mathbf{y}; \theta) +\Big) + E(\mathbf{y}^k; \theta) \tag{3.19}$$

The subgradient of the above loss function $\mathcal{F}$ can be expressed as:

$$\partial \mathcal{F}_\theta = E^{'}(\mathbf{y}^k; \theta) - E^{'}(\mathbf{y}^*; \theta), \tag{3.20}$$

where $\mathbf{y}^* = \arg\min_{\mathbf{y}} \Big(E(\mathbf{y}; \theta) - \Delta(\mathbf{y}, \mathbf{y}^k)\Big)$. Thus, during learning in the max-margin scheme we are required to compute the configuration $\mathbf{y}^*$ that minimizes the loss augmented energy function above. Computing the optimal configuration is an instance of *maximum a posteriori inference* and is NP hard in general. Approximate schemes include max-product message passing, graph cuts [Kolmogorov and Zabin, 2004], linear programming formulations [Wainwright et al., 2005] and dual decomposition [Komodakis et al., 2007]. It is useful to note that both *MAP* and *maximum likelihood* parameter estimation for graphical models require performing inference in an inner loop to compute gradients or sub-gradients during learning.

Inference: Given a model as in Equation 3.12 with parameters θ learned from data, and input features $\mathbf{x}$, there are usually two types of inference problems we are concerned with. The first type of inference corresponds to *marginal inference*, which can be computed by marginalizing over the remaining variables:

$$P(\mathbf{y}_i|\mathbf{x}) = \sum_{\mathbf{y}\backslash\mathbf{y}_i \in \mathcal{Y}} P(\mathbf{y}|\mathbf{x}). \tag{3.21}$$

The above equation forms the basis of the **sum-product belief propagation** algorithm, where the underlying graph is transformed into a factor graph [Kschischang et al., 2001]. A factor graph introduces new factor nodes for each clique in the graph. Belief propagation on the factor graph involves the recursive computation of the following equations that pass messages between factors and nodes, and between nodes and factors:

$$\mu_{v\to f}(y_i) = \prod_{f'\in\mathcal{N}(i)\backslash f} \mu_{f'\to i}(y_i), \tag{3.22}$$

$$\mu_{f\to i}(y_i) = \sum_{\mathbf{y}_f|y_i=y_i'} \psi_f(\mathbf{y}_f) \prod_{j\in\mathcal{N}(f)\backslash i} \mu_{j\to f}(y_j), \tag{3.23}$$

$$P(y_i|\mathbf{x}) = \prod_{f\in\mathcal{N}(i)} \mu_{f\to i}(y_i), \tag{3.24}$$

where $\mu_{i\to f}(y_i), \mu_{f\to i}(y_i)$ are messages from variables to factors and factors to variables respectively. $\mathcal{N}(i) \setminus f$ denotes the neighboring factor nodes of the variable i excluding the factor f, similarly $\mathcal{N}(f) \setminus i$ represents the neighboring variables of factor f excluding variable i and y_i' is an assignment of the variable y_i. When the graph $\mathcal{G}$ is a tree or a chain, exact marginals for each of the variables can be computed in two passes of message passing. For general graphs, exact marginal inference is NP-hard, but approximate marginals can be computed by passing messages described in Equations 3.22-3.24 until convergence in a procedure called *loopy belief propagation*.

The second type of inference is to find the configuration of output variables $\mathbf{y}$ that is most probable under the posterior distribution $P(\mathbf{y}|\mathbf{x})$ a*maximum a* *posteriori* inference where we seek to find the optimal configuration of the output:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}\in\mathcal{Y}} \; P(\mathbf{y}|\mathbf{x}) \tag{3.25}$$

The above equation reminds us of the definition of structured prediction in Equation 3.11, where the prediction function is now given by $f(I) = \min\limits_{\mathbf{y}\in\mathcal{Y}} E(\mathbf{y};\theta)$. Similar to Equations 3.22-3.24, the most probable configuration can be computed by using a variant of loopy belief propagation known as **max-product** message passing, where the summation in Equation 3.23 is replaced by a max operator.

In **variational mean field inference**, the intractable distribution $P(\mathbf{y}|\mathbf{x})$ is approximated by a tractable distribution $Q(\mathbf{y})$ that minimizes the Kullback-Liebler (KL) divergence $KL(Q\|P)$ between the two distributions. The distribution Q is chosen to be a distribution that can be factorized as the product of independent marginals $Q(\mathbf{y}) = \prod_i \mathbf{q}_i(y_i)$ [Wainwright and Jordan, 2008]. By differentiating the KL divergence objective and setting the gradient to zero, the following update rule can be obtained for each marginal $\mathbf{q}_i$ as[3]:

$$\mathbf{q}_i(y_i) \propto \; \psi_i(y_i) \prod_{j\in\mathcal{N}(i)} \exp\left[\phi_{j\to i}(y_i)\right]. \tag{3.26}$$

where $\phi_{j\to i}$, the mean-field message from variable y_j to y_i, is given by:

$$\phi_{j\to i}(y_i) = \sum_{y_j} \mathbf{q}_j(y_j) \log \psi_{ij}(y_i, y_j). \tag{3.27}$$

Inference is thus performed by iteratively computing Equations 3.26, 3.27 until convergence to obtain marginals $\mathbf{q}_i$ for each variable. The update rule in Equation 3.26 for each variable y_i is a function of the the associated unary term $\psi_i(y_i)$ and the the mean-field messages from its neighbors $\mathcal{N}(i)$. It is useful to note that when

[3] A detailed derivation for the mean field update equations can be found in [Koller and Friedman, 2009]

the pairwise potential takes the form $\psi_{ij}(y_i, y_j) = f(|y_i - y_j|)$, the computation of the message in Equation 3.27 reduces to a convolution, allowing the use of efficient computational techniques for fast convolutions[4]. In this view, we observe that each marginal is updated by the product of convolving neighboring marginals, $\mathbf{q}_j$, with a transformed pairwise term.

3.2.2 Message Passing Inference Machines

Graphical models aim to provide a distinct separation between modeling and inference. However, learning the parameters of a graphical model includes performing inference as the dominant subroutine to calculate gradients or subgradients of the learning objective. As inference is intractable for all but the simplest graphs [Cooper, 1990], approximate inference techniques are used during learning which can lead to multiple adverse effects. Kulesza and Pereira [2007] show that approximate inference during learning can reduce the expressivity of a model and lead the learning procedure astray by misleading the search for the correct model parameters. Kumar et al. [2005] show that, in practice, test time performance can be drastically effected by the particular type of approximation used and can be sensitive to a mismatch between the inference method used during testing and training.

Message passing inference machines [Munoz et al., 2010; Ross et al., 2011; Munoz, 2013] aim to overcome these problems by providing tight coupling between learning and inference. Inference machines are motivated by the observation that message passing inference as described in Section 3.2.1 can be viewed as a sequence of simpler prediction problems. Inference machines can be viewed as an unrolling of the sequence of computations in message passing inference and

[4] A compelling example is from Krähenbühl and Koltun [2011], who use mean-field inference on a densely connected CRF in the context of image segmentation. By using Gaussian pairwise terms, they reduce marginal updates to high-dimensional gaussian filtering, for which fast computational techniques exist (such as those proposed by Adams et al. [2010]).

directly training a predictor to produce the ideal message updates in each iteration. As a consequence, inference machines eschew probabilistic modelling of the data. As the goal in structured prediction is to obtain an accurate output configuration $\mathbf{y}^*$ that has lowest score compared to all possible configurations and we are not necessarily concerned with an accurately calibrated probability for each configuration [LeCun et al., 2006].

We describe a **mean field inference machine** which emulates the sequence of computations in variational mean field inference (see Section 3.2.1), where we observed that the updates consist of a aggregating information from estimates of marginals of neighboring variables as described in Equation 3.26. We observe that the mean-field inference procedure can be viewed as sequential classification, where, in each iteration, a predictor with a log-linear form produces an estimate for the marginal of each variable, using *features* computed from the marginals of neighboring variables as given in Equation 3.27. The mean field update of Equation 3.26 in t^{th} iteration can be written as:

$$\mathbf{q}_i^t(y_i) \propto \ \psi_i(y_i) \exp\left(\sum_{j \in \mathcal{N}(i)} \phi_{j \to i}^{t-1}(y_i) \right). \tag{3.28}$$

Written in this form, the mean field update for the marginal has the familiar log-linear form of maximum-entropy classifiers such as a logistic classifier, using messages from neighboring variables $\phi_{j \to i}(y_i)$ as input features. We summarize the procedure in Algorithm 3.

A mean field inference machine consists of a sequence of predictors $\{g_i^t\}$ trained to directly predict the "marginals" for each variable $\mathbf{y}_i$. Note that the term marginal is used in a loose sense, as an inference machine does not model an actual probability distribution; the marginals in this case can be thought of as a measure of confidence for each variable. Eschewing probabilsitic modeling, such as the use of an exponential family distribution or parametric potential functions,

Algorithm 3 `mean_field_inference`

```
Input: $\mathbf{x}$   //*Features*
Set: $\{\mathbf{q}_i^0\}$   //*Initialize uniform marginals*
for $t = 1 \dots T$ do
    for $i = 1 \dots P$ do
        $\phi_{j\to i}^t(y_i) \leftarrow \sum_{y_j} \mathbf{q}_j^{t-1}(y_j) \log \psi_{ij}(y_i, y_j) \quad \forall j \in \mathcal{N}(i)$   //*Compute "features"*
        $\mathbf{q}_i^t(y_i) \leftarrow \frac{1}{Z}\psi_i(y_i) \exp\left(\sum_{j\in\mathcal{N}(i)} \phi_{j\to i}^t(y_i)\right)$   //*"Predict" marginal*
    end for
end for
Return: $\{\mathbf{q}_i^T\}$
```

Algorithm 4 `mean_field_inference_machine`

```
Input: $\mathbf{x}$   //*Features*
Set: $\{\mathbf{q}_i^0\}$   //*Initialize uniform "marginals"*
for $t = 1 \dots T$ do
    for $i = 1 \dots P$ do
        $\phi_{j\to i}^t(y_i) \leftarrow \Psi(y_i, \mathbf{q}_j^{t-1}) \quad \forall j \in \mathcal{N}(i)$   //*Compute features*
        $\mathbf{q}_i^t(y_i) \leftarrow g_i^t(\{\phi_{j\to i}^t(y_i)\})$   //*Predict "marginal"*
    end for
end for
Return: $\{\mathbf{q}_i^T\}$
```

and viewing the inference procedure purely a sequence of supervised prediction problems allows us greater flexibility. We are now free to choose the form of the "predictor" allowing for high-capacity models and the "features" used to convey contextual information from neighboring variables. For a mean-field inference machine, the predictions in the t^{th} iteration would be written as:

$$q_i^t(y_i) = g_i^t\left(\{\phi_{j\to i}^t(y_i)\}_{j\in\mathcal{N}(i)}\right), \tag{3.29}$$

where g_i^t is a predictor for the variable y_i and $\{\phi_{j\to i}^t(y_i)\}_{j\in\mathcal{N}(i)}$ are features computed using a feature transform $\Psi(z, \mathbf{q})$, where z is a variable assignment and $\mathbf{q}$ is a marginal, and computed as:

$$\phi_{j\to i}^t(y_i) = \Psi(y_i, \mathbf{q}_j^{t-1}) \tag{3.30}$$

In this view, the predictors $\{g_i^t\}$ are not constrained to be of a particular parametric form and we are free to use any high-capacity predictor that is suited to the inference task such as a boosted random forest or a *universal function approximator* [Hornik et al., 1989] such as a convolutional neural network. Replacing the features with a general feature transform Ψ allows for modeling potentially richer interactions. We summarize the inference procedure for a mean-field inference machine in Algorithm 4. Note the similarity to the inference procedure of Algorithm 3. We see that the resulting procedure reduces structured prediction to a sequence of supervised classification problems. Learning an inference machine thus only requires training the supervised classifiers in each iteration to produce the ideal marginals (available to us during training) using features $\mathbf{x}$ and contextual features computed on neighboring marginals $\{\mathbf{q}_j^{t-1}\}_{j \in \mathcal{N}(i)}$ from the previous iteration. The message passing inference procedure in loopy belief propagation can be emulated in a similar fashion by an inference machine as described in Ross et al. [2011], where predictors are trained to produce the sequence of messages between factors and nodes in a factor graph.

In following chapters, we describe and apply inference machines developed for the task of articulated pose estimation. We find that the *pose machine* architecture provides a modular framework for a difficult structured prediction problem and a systematic design for composing powerful supervised predictors such as convolutional neural networks for the task of pose estimation.

CHAPTER 4

Pose Machines

In this chapter we describe our approach for localizing keypoints of a deformable articulated object in a single image. We address this problem in the context of human pose estimation where the keypoints correspond to anatomical landmarks on the body. Detecting landmarks by relying purely on local image evidence tends to perform poorly. In Figure 4.1, we show confidence maps for the detection of the head, elbow and wrist keypoints using only local image evidence. We see that for parts with discriminative appearance (such as the head), we obtain confidence maps with a strong unimodal peak, however for parts with large appearance variation such as the wrist and for parts without strongly discriminative appearance such as the hip, we obtain noisy confidences.

This is due to two primary sources of complexity in estimating the articulated pose of a human from an image. The first arises from the large number of degrees of freedom (nearly 20) of the underlying articulated skeleton which leads to a high dimensional configuration space to search over. The second is due to the large variation in appearance of people in images. The appearance of each part can vary with configuration, imaging conditions, and from person to person.

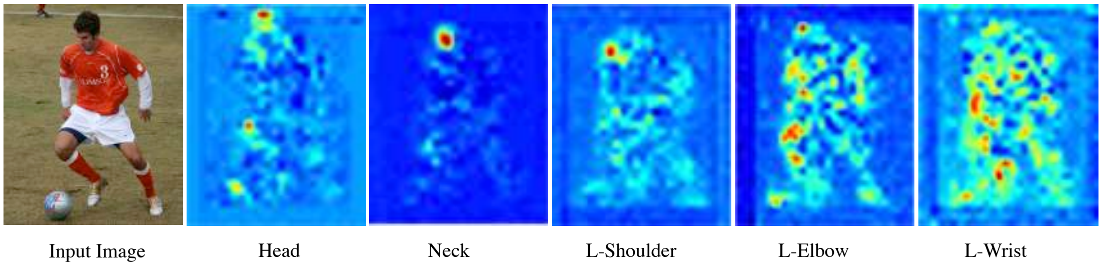


Figure 4.1: **Confidence maps from independent part detectors.** Parts with strongly discriminative appearance such as the head and shoulders have sharp peaks and unimodal confidence maps. Parts lower down in the kinematic chain of the human skeleton tend to be harder to detect due to large appearance variation.

A powerful cue for detecting a part is its spatial context: the detection of one part can provide cues for the detection of other parts. Detecting the face, for which proven algorithms exist, provides a strong cue for the location of other parts like the shoulders and torso. The intuition that the locations of parts in an image are spatially correlated can be captured in a graphical model framework. In such a model, we define a graph with the nodes representing the locations of parts and edges between nodes encoding conditional independences. The parameters of such a model can then be learned by maximizing a data likelihood or max-margin objective. Detecting part locations is then reduced to performing inference on the learned graphical model.

Pictorial structure approaches [Felzenszwalb and Huttenlocher, 2005; Ramanan et al., 2005; Andriluka et al., 2010, 2009; Yang and Ramanan, 2011; Johnson and Everingham, 2010] employ a tree-structured graphical model to capture the correlations and spatial dependencies between the locations of each of the parts. However, inference in graphical models is difficult and inexact in all but the most simple models such as a tree-structured or star-structured model. These simplified models are unable to capture important dependencies between locations of each of the parts and lead to characteristic errors. One such error—double counting (see Figure 4.2)—occurs when the same region of the image is used to explain more than one part. This error occurs because of the symmetric appearance of body parts (for e.g., the left and right arm usually have similar appearance) and that it is a valid configuration for parts to occlude each other.

Unfortunately, modeling this appearance symmetry and self-occlusion with a graphical model requires additional edges and induces loops in the graph. Such non-tree structured graphical models typically require the use of approximate inference (e.g., loopy belief propagation), which makes parameter learning difficult [Kulesza and Pereira, 2007]. Moreover, defining the potential functions in these models requires careful consideration when specifying the types of interactions. This choice is usually dominated by parametric forms such as simple quadratic models in order to enable tractable inference [Felzenszwalb and Huttenlocher, 2005]. Finally, to further enable efficient inference in practice, many approaches are also usually restricted to use simple classifiers such as mixtures of linear models for part detection [Yang and Ramanan, 2011], which are choices guided by tractabilty of inference rather than the complexity of the underlying data distribution. Such trade-offs result in a restrictive model that do not address the inherent complexity of the problem.

This chapter describes an approach that aims to side-step this complexity vs. tractability trade-off by directly training the inference procedure. Conceptually, the presented method, which we refer to as a *Pose Machine*, is a sequential prediction algorithm that emulates the mechanics of message passing to predict a confidence for each variable (part), iteratively improving its estimates in each stage. The inference machine architecture is particularly suited to tackle the main challenges in pose estimation. First, it incorporates richer interactions among multiple variables at a time, reducing errors such as double counting, as illustrated in Figure 4.2.

Second, it learns an expressive spatial model directly from the data without specifying the parametric form of the potential functions. Additionally, its mod-

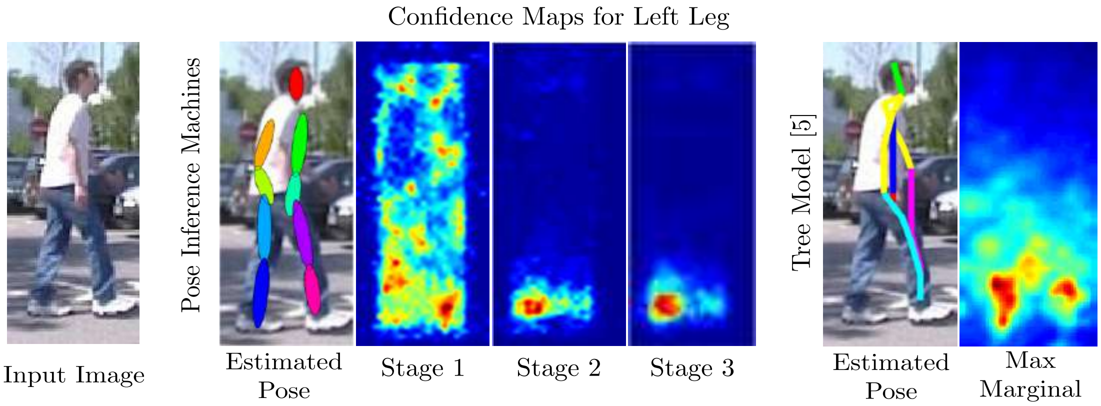


Figure 4.2: **Reducing double counting errors.** By modelling richer interactions we prevent the double counting error that occurs in tree models. On the left we show the belief for the left foot of the person in each stage of the pose machine. The belief quickly converges to a single sharp peak. On the right, we see that the tree-structured model has a max-marginal for the left foot with multiple peaks and resulting in both legs being placed on the same area in the image.

ular architecture allows the use of more expressive predictors which are better suited to deal with the highly multi-modal appearance of each part. Inspired by recent work [Pishchulin et al., 2013a; Sapp and Taskar, 2013] that has demonstrated the importance of conditioning finer part detection on the detection of larger composite parts in order to improve localization, we incorporate these multi-scale cues in our framework by also modeling a hierarchy of parts.

4.1 Model Overview

We view the problem of detecting anatomical landmarks as a structured prediction problem. That is, we model the pixel location of each anatomical landmark (which we refer to as a part) in the image, $y_p \in \mathcal{Z} \subset \mathbb{R}^2$, where $\mathcal{Z}$ is the set of all (u, v) locations in an image. Our goal is to predict the structured output $\mathbf{y} = (y_1, \ldots, y_p)$ for all P parts. A pose machine consists of a sequence of multi-class classifiers, $g_t(\cdot)$, that are trained to predict the location of each part. In each stage $t \in \{1 \ldots T\}$, the classifier predicts a confidence for each output variable assignment $y_p = z$, $\forall z \in \mathcal{Z}$ based on features of the image data $\mathbf{x}_z \in \mathbb{R}^d$ and contextual information from the preceeding classifier in the neighborhood around each y_p. In each stage, the computed confidences provide an increasingly refined estimate for the variable. For each stage t of the sequence, the confidence for the assignment $y_p = z$ is computed and denoted by

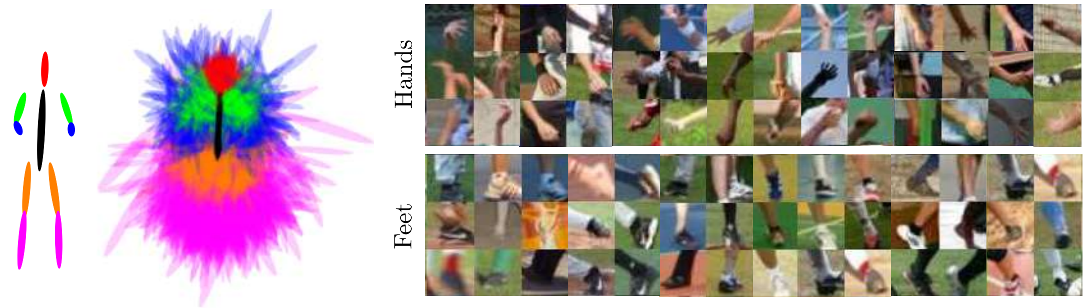
$$b_t(y_p = z) = g_t^p \left(\mathbf{x}_z ; \bigoplus_{i=1}^{P} \psi(z, \mathbf{b}_{t-1}^i) \right), \tag{4.1}$$

where

$$\mathbf{b}_{t-1}^p = \{b_{t-1}(y_p = z)\}_{z \in \mathcal{Z}}, \tag{4.2}$$

is the set of confidences from the previous classifier evaluated at every location z for the p^{th} part. The feature function $\psi : \mathcal{Z} \times \mathbb{R}^{|\mathcal{Z}|} \rightarrow \mathbb{R}^{d_c}$ computes contextual features from the classifiers' previous confidences, and $\bigoplus$ denotes an operator for vector concatenation. We denote the collection of $P+1$ confidence maps of stage t as $\mathbf{b}_t$

Unlike traditional graphical models, such as pictorial structures, the inference machine framework does not need explicit modeling of the dependencies between variables via potential functions. Instead, the dependencies are arbitrarily combined using the classifier, enabling potentially complex interactions among the variables. Directly training the inference procedure via a sequence of simpler subproblems, allows us to use any supervised learning algorithm to solve each subproblem. We are able to leverage the state-of-the-art in supervised learning and use a sophisticated predictor capable of handling multi-modal variation. In following sections, we describe the supervised classification framework used for detecting parts using only (a) only local image evidence in the first stage of the sequence and (b) using spatial context from a previous stage's predictions in



(a) Large Configuration Space

(b) Appearance Variation

Figure 4.3: **Complexity in configuration and appearance.** (a) The full set of spatial configurations of the human body in the LEEDS sports training dataset [Johnson and Everingham, 2010] are visualized. The highly articulated nature of the underlying skeleton results in a large number of possible configurations to search over. (b) Variation in appearance for hands and feet in the LEEDS sports dataset. Appearance varies greatly with configuration, imaging conditions and from instance to instance resulting in a multi-modal distribution.

subsequent stages of the sequence.

4.2 Keypoint Localization from Local Evidence

In the first stage of the sequence of predictors, the classifier g_1 uses only local image evidence around each location z to predict confidences each of the P parts and a background class:

$$g_1\left(\mathbf{x}_z\right) \rightarrow \{b_1^p(y_p = z)\}_{p\in 0\ldots P} \tag{4.3}$$

This is a challenging task, because, as shown in Figure 4.3, parts have large variation in appearance. The same configuration can have different appearance depending on local factors such as clothing and skin color, and global factors such as illumination. Additionally, the appearance of each part in the image is coupled with the configuration of the body and the relative camera location.

Using a simple counting argument, we can enumerate the number of appearance states possible. Assuming a simplified model with η_l states for local appearance properties, such as color and identity, and η_g for a global property such as lighting, a total number of $(\theta\eta_l)^d\eta_g$ appearance states are induced. As an example, setting local appearance $\eta_l = 5$ and global appearance $\eta_g = 10$, generates 10^{29} different appearances. Our goal then, is two-fold: first, to come up with a representation of local image evidence that is invariant to some of the factors described above, and second, to use a predictor that is able to effectively learn a decision boundary to separate the classes based on the above representation.

The aforementioned two-fold approach is the classical supervised classification pipeline of extracting manually designed image features at each image location followed by classification using a multi-class classifier. The features and classifier need to both be chosen carefully so as to effectively handle the inherent complexity of the data distribution. Additionally, in order to gain maximally from very large datasets, the learning algorithms for the chosen predictor should scale with the size of the dataset. We describe some of the trade-offs and design choices associated with such a pipeline in Section 4.2.1.

Alternatively, feature engineering and classification can be combined and learned jointly, directly from data. Deep architectures [Bengio, 2009] are a method for learning both the feature representation and the predictor simultaneously. Deep convolutional architectures with many layers have been effective for several vision tasks such as object detection [Sermanet et al., 2013], image segmentation [Long et al., 2015] etc. In Section 4.2.2 we describe a deep convolutional architecture for the task of part classification from local image evidence.

4.2.1 A Classical Supervised Classification Pipeline

In the classical supervised classification pipeline (see Section 3.1) the primary goals are to engineer features that represent the data effectively and to choose a classifier capable of learning accurate decision boundaries based on these features. We describe choices for both criteria in the context of problem of part classification.

Image Features

We extract a set of image features from a patch at each location in the image. We use *Histogram of Gradients (HOG)* features, *LAB* color features, and gradient magnitude which correspond to standard channel features as described in Section 3.1. We model a keypoint by a patch around its location with a spatial extent of 40 pixels assuming that scaled training samples have objects (persons) with a height 200 pixels. We choose a bin size of either 4 or 8 depending on the resolution of prediction required.

Choice of Predictor

The modular nature of the pose machine architecture allows us to insert any supervised learning classifier as our choice of multi-class predictor g. As the data distribution is highly multi-modal, a high-capacity non linear predictor is required. A good choice for the predictor is the *random forest* predictor [Breiman, 2001] and its boosted variant: gradient *boosted random forests* [Friedman, 2001]. Random forests and boosted random forests have been empirically shown [Caruana and Niculescu-Mizil, 2006] to consistently outperform other methods on several datasets. We learn a boosted random forest classifier (see Section 3.1.2) by optimizing the SVM hinge loss objective [Grubb and Bagnell, 2011]. We use 25 iterations of boosting, with a random forest classifier. Each random forest classifier consists of 10 trees, with a maximum depth of 15 and with a split performed only if a node contained greater than 10 training samples.

In the context of articulated human pose estimation we use 14 parts to model

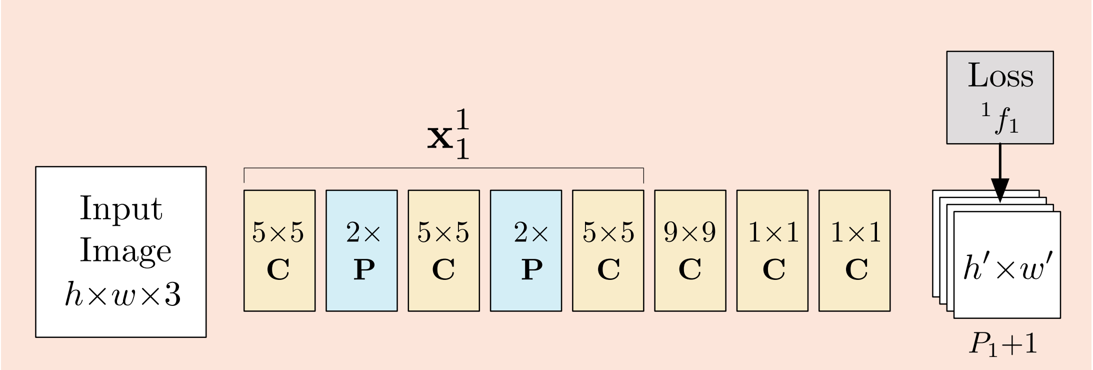


Figure 4.4: **Convolutional Architecture for Keypoint Localization** We show a deep convolutional architecture for performing keypoint localization that relies on local image evidence in a small region (receptive field) around each pixel location

a full body model and a 10 parts for an upper body model. At each location z in the image, features $\mathbf{x}_z$ are extracted and supplied to the classfiier g_1 to produce the beliefs $\{b_1(x_p = z)\}$ for each of the parts $p \in \{1 \dots P + 1\}$.

4.2.2 A Convolutional Architecture for Part Detection

Deep convolutional networks provide an architecture to perform both feature learning and classification in a single architecture trained jointly. The advantage of such an architecture is that the network is free to learn the most suitable representation for the given task, directly from the data instead of having to adapt to a manually designed representation. We describe a convolutional architecture for our task of part detection from local image evidence. Figures 4.4 shows the network structure for part detection from local image evidence using a deep convolutional network. The evidence is *local* because the receptive field of the network is constrained to a tightly cropped patch around the keypoint location. We use a network structure composed of 6 convolutional layers followed by two 1×1 convolutional layers which results in a fully convolutional [Long et al., 2015] architecture (see Figure 4.4) that allows inputs of an arbitrary size $h \times$

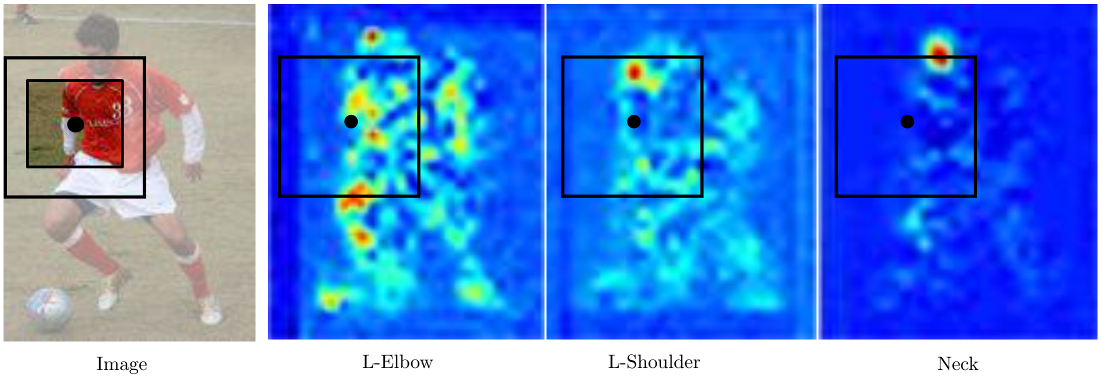


Figure 4.5: Spatial context from confidence maps can provide valuable cues for detection. The peak in the confidence map for easier to detect landmarks such as the shoulder can be a strong cue for the location of difficult to detect landmarks such as the left elbow.

w. The image is first contrast normalized and then passed through the three convolutional layers with kernels of size $5 \times 5 \times 128$, $5 \times 5 \times 128$, and $5 \times 5 \times 32$. We perform max-pooling with stride 2 after the first two convolutional layers resulting in an output layer of size $\frac{h}{4} \times \frac{w}{4} \times 32$ after the third layer. The fourth, fifth, and sixth layers have kernel sizes of $9 \times 9 \times 512$, $1 \times 1 \times 512$, $1 \times 1 \times (P_1 + 1)$ resulting $P + 1$ output confidence maps of size $h' \times w'$, corresponding to P body parts and background. The fourth layer of the network is equivalent to the first fully connected layer that takes the cascade of spatial features (into a long vector of length 512) in a typical deep network such as Krizhevsky et al. [Krizhevsky et al., 2012]. Similarly, the two 1×1 convolutional layers are the convolutional equivalent of fully connected layers. We use rectified linear units [Nair and Hinton, 2010] after each convolutional layer except the last one. The receptive field of the first level of the hierarchy is thus 64×64 pixels. The network can effectively be viewed as sliding a deep network across an image and regressing each 64×64 image patch to a $P_1 + 1$ sized output vector that represents a score for each part at that image location.

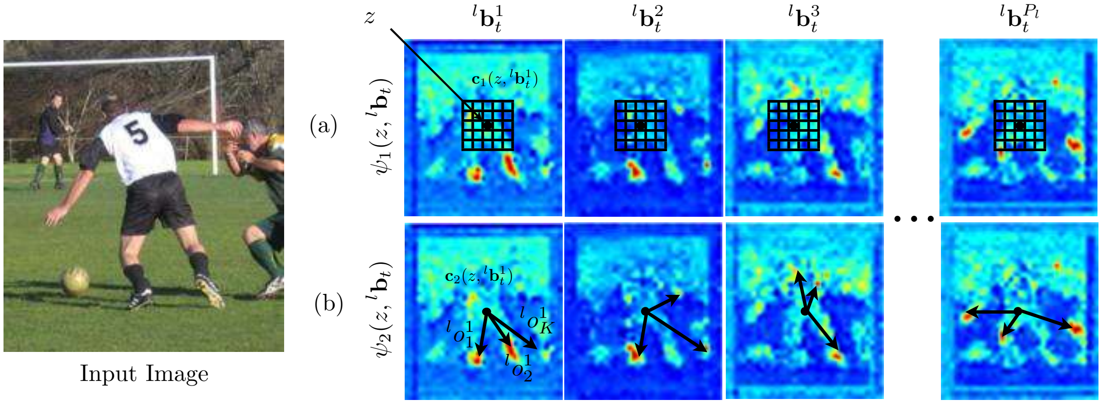
4.3 Sequential Prediction with Spatial Context

As discussed in the previous section, detecting landmarks from purely local image evidence performs poorly in general. While landmarks such as the head and shoulders that have consistent appearance, the detection rate is around 50%, but for landmarks lower down the kinematic chain, the accuracy is closer to 25%. However, the landscape of the confidence maps around a part location, albeit noisy, can be very informative. The fact that the confidence map for the shoulder has a sharp peak in the vicinity of the elbow can be used as a strong cue for predicting the location of the elbow (see Figure 4.5). A predictor in subsequent stages ($g_{t>1}$) can use the spatial context ($\psi(\cdot)$) of the noisy confidence maps in a region around the image location z and improve its predictions by leveraging the fact that parts that occur in consistent geometric configurations. In this section we describe how to design or learn a feature representation that encodes the spatial context of the confidence maps relative to a location in the image. We discuss designed features in Section 4.3.1, and describe a convolutional architecture for learnign such a representation in Section 4.3.2.

4.3.1 Designing Spatial Context Representations

To capture the spatial correlations between the confidences of each part with respect to its neighbors, we describe the design of two types of feature maps denoted by ψ_1 and ψ_2.

Context Patch Features. The feature map ψ_1 at a location z takes as input the confidence maps for the location of each landmark and produces a feature that is a vectorized patch of a pre-defined width extracted at the location z in the confidence map $\mathbf{b}_t^p$ (see Figure 4.6a). We denote the set of patches extracted and vectorized at the location z, from the beliefs of the parts in the hierarchy


z
$^l\mathbf{b}_t^1$
$^l\mathbf{b}_t^2$
$^l\mathbf{b}_t^3$
$^l\mathbf{b}_t^{P_l}$
(a)
$\psi_1(z, {}^l\mathbf{b}_t)$
$\mathbf{c}_1(z, {}^l\mathbf{b}_t^1)$
(b)
$\psi_2(z, {}^l\mathbf{b}_t)$
$\mathbf{c}_2(z, {}^l\mathbf{b}_t^1)$
$^l o_1^1$
$^l o_2^1$
$^l o_K^1$
...

Input Image

Figure 4.6: **Context Feature Maps** (a) Context patch features are computed from each scoremap for each location. The figure illustrates a 5 × 5 sized context patch (b) The context offset feature comprises of offsets to a sorted list of peaks in each scoremap.

level l, by $\mathbf{c}_1(z, \mathbf{b}_{t-1}^p)$. The feature map ψ_1 is therefore given by:

$$\psi_1(z, \mathbf{b}_{t-1}) = \bigoplus_{p \in 0 \ldots P_l} \mathbf{c}_1(z, \mathbf{b}_{t-1}^p). \tag{4.4}$$

In words, the context feature is a concatenation of scores at location z extracted from the confidence maps for each part in each level the hierarchy. The context patch encodes neighboring information around location z. Note that because we encode the context from all parts, this would be analogous to having a graphical model with a complete graph structure and would be intractable to optimize. The context patch feature can also be reduced in size by performing pooling in smaller windows in the grid, this has the effect of increasing spatial invariance, but at the cost of precise localization.

Context Offset Features. We compute a second type of feature, ψ_2, in order to encode long-range interactions among the parts that may be at non-uniform, relative offsets. First, we perform non-maxima suppresion to obtain a sorted list of K peaks from each of the P confidence maps $^l\mathbf{b}_{t-1}^p$ for all the anatomical landmarks. Then, we compute the offset vector in polar co-ordinates from location z

to each k^{th} peak in the confidence map of the p^{th} part denoted as as $o_k^p \in \mathbb{R}^+ \times \mathbb{R}$ (see Figure 4.6b). The set of context features computed on $\mathbf{b}_1$ is then given by:

$$\mathbf{c}_2(z, \mathbf{b}_{t-1}^p) = [o_1^p; \ldots ; o_K^p] \,. \tag{4.5}$$

The context feature ψ_2 is then formed by concatenating the offset features $\mathbf{c}_2(z, {}^l\mathbf{b}_{t-1}^p)$ from the confidence maps for each part in the the hierarchy:

$$\psi_2(z, \mathbf{b}_{t-1}) = \bigoplus_{p \in 1 \ldots P} \mathbf{c}_2(z, \mathbf{b}_{t-1}^p). \tag{4.6}$$

The context patch features (ψ_1) capture coarse information regarding the confidence of the neighboring parts while the offset features (ψ_2) capture precise relative location information. The final context feature ψ is computed by concatenating ψ_1 and ψ_2:

$$\psi = \psi_1 \bigoplus \; \psi_2 \tag{4.7}$$

In the second stage of a pose machine, the classifier g_2 accepts as input the image features $\mathbf{x}_z$ and features computed on the confidences via the feature function ψ for each of the parts in the previous stage. The feature function ψ serves to encode the landscape of the confidence maps from the previous stage in a spatial region around the location z of the different parts. As shown in Figure 4.7 we see that the context features proposed are complementary: the patch features outperform the offset features on the elbow joints, but the offset features outperform the patch features on the wrist joints. Using both sets of features together outperforms using just either further suggesting that they encode complementary information.

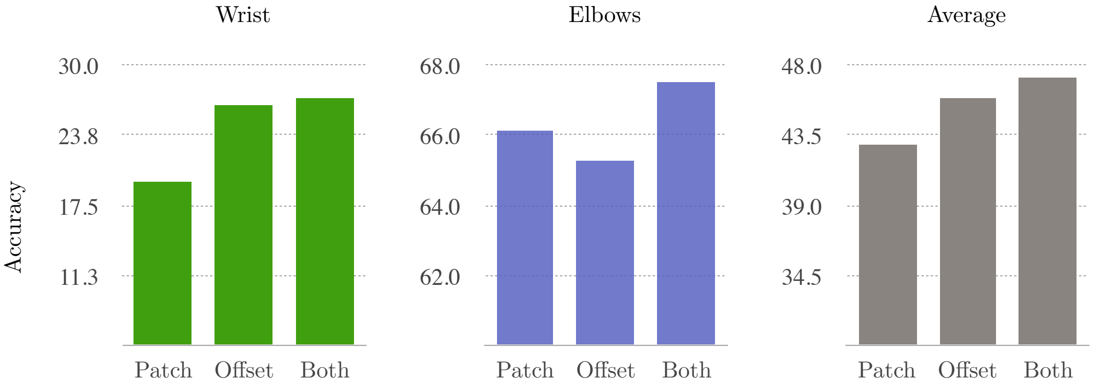


Figure 4.7: **Effect of context features on performance.** We compared variants of our model using the different types of designed context features for localizing the elbow and wrist landmarks on the FLIC dataset. We see that the context features proposed seem to be complementary: the patch features outperform the offset features on the elbow joints, but the offset features outperform the patch features on the wrist joints. Using both sets of features together outperforms using just either further suggesting that their effects are complementary

4.3.2 Learning Spatial Context Representations

Instead of hand-crafting features to capture contextual information from the previous stage beliefs/confidences, we could also use the feature learning capabilities of a convolutional architecture to directly learn a representation. A deep convolutional neural network is especially suited for the task as it can potentially learn arbitrarily complex functions of the input [Hornik et al., 1989].

For a pose machine, with a convolutional network as the prediction module, we do not define an explicit function that computes context features. Instead, we aim to learn ψ by designing a network with an appropriately sized receptive field on the confidences from the previous stage. The second stage predictor is designed such that the spatial support of the network's receptive field is large enough to capture information regarding spatial co-occurrences in the confidence maps of the different parts. In contrast to the method described above which uses handcrafted features to summarize contextual information, we directly learn

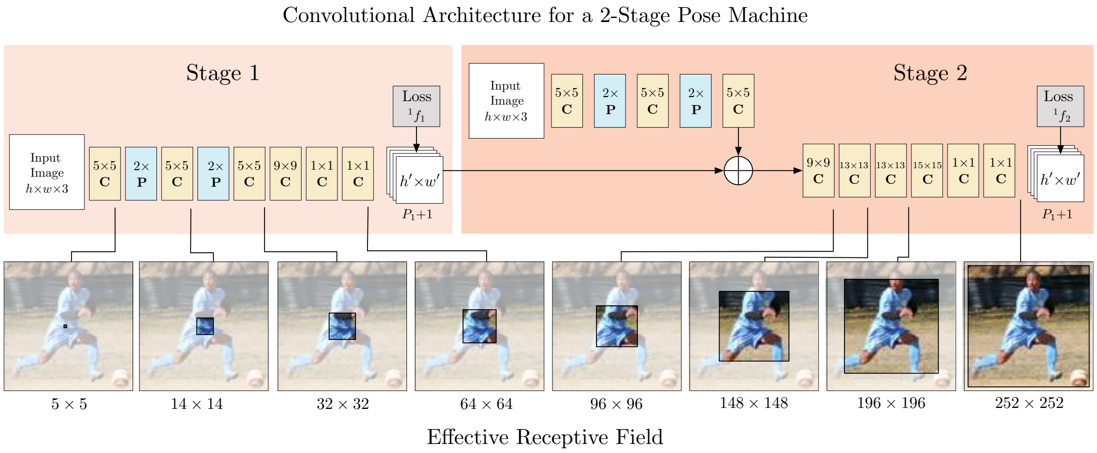


Figure 4.8: **Convolutional Architecture for a 2-stage Pose Machine** We show a deep convolutional architecture for a pose machine with two stages. In the second stage we use a convolutional architecture that uses both confidence maps from the previous stage as well as features learned directly from the image. Below, we also show the effective receptive field on an image at various parts of the architecture.

a feature representation on the combination of confidence maps and the original image via a deep convolutional network (see Figure 4.8). The subsequent stages therefore learn a image-dependent spatial model by combining information regarding the beliefs of parts from the previous stage and local image evidence from each patch.

Large receptive fields for learning spatial context: The design goal for the network in the second stage (and subsequent stages) is to provide a sufficiently large spatial region on the confidence maps (from the previous stage) as input so that the second stage network can learn potentially complex and long-range correlations between the locations of anatomical landmarks. The design of the network is guided by achieving a receptive field at the output layer of second stage network that meets this criterion.

Our design for the second stage network is shown in Figure 4.8 for the two hierarchy levels. The confidence maps from the first stage were generated from

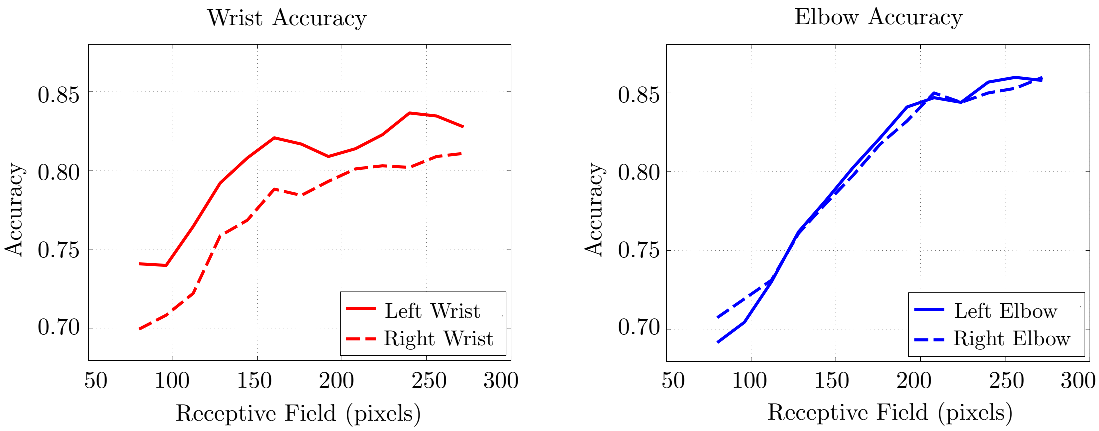
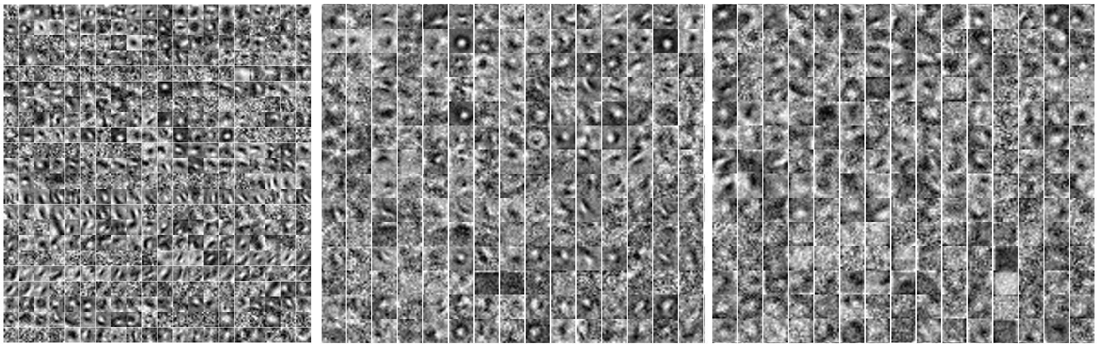


Figure 4.9: **Large receptive fields for spatial context** We show that networks with large receptive fields are effective at modelling long range spatial interactions between parts. We see that performance in terms of localization accuracy on the FLIC dataset increases for a network with increasing receptive field uptil around 250 pixels and saturates thereafter.

a network that examined the image locally with a receptive field of size 64 × 64. In the second stage, we design a network that drastically increases the equivalent receptive field. Large receptive fields can be achieved either by pooling, at the expense of precision, increasing the kernel size of the convolutional filters largely increasing the number of parameters or by increasing the number of convolutional layers at the risk of encountering vanishing gradients during training. We choose to use multiple convolutional layers as it allows us to be parsimonious with respect to the number of parameters of the model while the risk of vanishing gradients is offset thanks to the intermediate supervision (described in following sections) enforced during training.

As shown in Figure 4.8, the network predicting locations of parts consists of a network which recomputes image features (note the similarity in construction to the first stage network of Figure 4.8, except for the last three convolution layers) and subsequent layers that operate on the combined image features and confidence maps. The image feature maps (denoted by $\mathbf{x}_z$) are combined with the confidence maps for the parts from both hierarchy levels from the previous



(a) 9 × 9 Convolution Layer

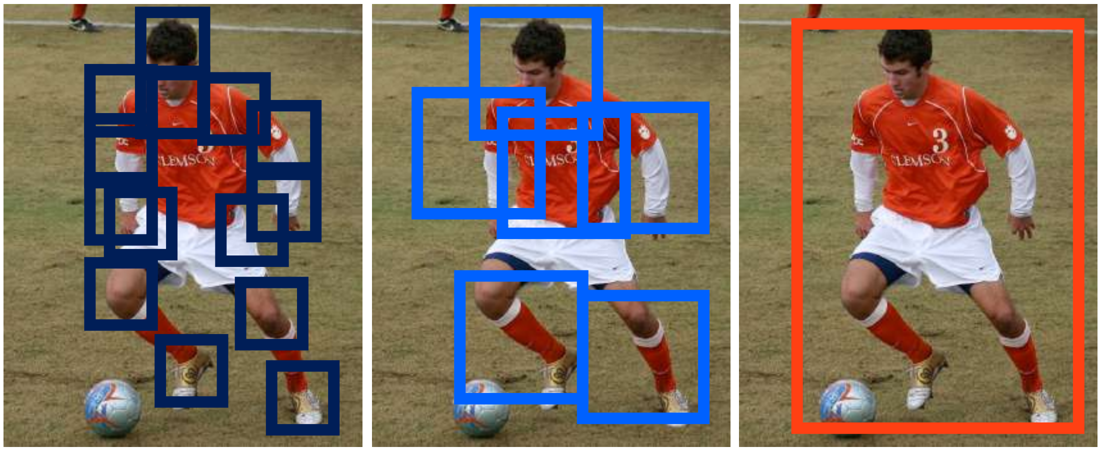
(b) 13 × 13 Convolution Layer

(c) 13 × 13 Convolution Layer

Figure 4.10: **Visualization of learned spatial context filters** We visualize the filter kernels for the convolutional layers of the second stage of our architecture. These filters correspond to layers that take as input both image feature maps and confidence maps from the preceding stage. The above layers are cascaded to achieve the desired receptive field on the confidence maps

stage. The confidence maps from preceding layers that are at a different resolution are correspondingly up-sampled using a deconvolutional layer [Long et al., 2015] or down-sampled using a max-pooling layer of the appropriate stride. The combination of image feature maps and confidence maps are then passed as input to a network with four convolutional layers which increase the receptive field followed by two 1×1 convolutional layers which effectively apply a fully connected network in a convolutional fashion across the image. We visualize some of the filters learned by our model in Figure 4.10. The receptive field on the confidence maps $\mathbf{b}_{t-1}$ is 45×45, which is equivalently 252×252 pixels on the original image.

We find that accuracy improves with the size of the receptive field. In Figure 4.9 we show the improvement in accuracy on the FLIC dataset [Sapp and Taskar, 2013] as the size of the receptive field on the original image is varied by varying the architecture. We see that the network achieves the best accuracy at an effective receptive field of between $200 - 250$ which also happens to be roughly the size of the object in the normalized training images. This improvement in accuracy with receptive field size also suggests that the network does indeed encode long range interactions between parts and that doing so is beneficial.



Level 1 Level 2 Level 3

Figure 4.11: **A Hierarchy of Parts**: Parts at various scales can be informative and can provide co-operative cues for detection.

4.4 Incorporating a Hierarchy

The visual structure around each landmark can provide discriminative information that is useful for prediction. Oftentimes larger regions around a landmark can contain more discriminative structure than a smaller tightly cropped region. For example, for landmarks such as the wrist joint, a patch tightly cropped around it has little distinguishable visual structure. Whereas, when we include additional visual context, we observe the consistent discriminative visual structure of the forearm. Multi-scale cues such as these can be useful for detection. In this section we describe how we incorporate such a hierarchy over scale in a pose machineby instantiating a separate set of predictors for each level in the hierarchy (see Figure 4.11. We also describe the corresponding convolutional architecture for a hierarchical pose machine that incorporates multi-scale cues by using networks with differing receptive fields.

4.4.1 Hierarchical Pose Machines

We define a hierarchy of scales from smaller atomic parts to larger composite parts. Each of the L levels of the hierarchy have parts of a different type. At the coarsest level, the hierarchy can be comprised of a single part that captures the whole body. The next level of the hierarchy is comprised of composite parts that model regions around landmarks that capture the visual structure of full limbs, while the finest level of the hierarchy is comprised of small parts that model a tightly cropped region around an anatomical landmark. We denote by $P_1, \ldots, P_L$, the number of parts in each of the L levels of the hierarchy. In the following, we denote ${}^{l}g_t^p(\cdot)$ as the classifier in the t^{th} stage and l^{th} level that predicts the score for the p^{th} part. While separate predictors could be trained for each part p in each level l of the hierarchy, in practice, we use a single multi-class predictor that produces a set of confidences for all the parts from a given feature vector at a particular level in the heirarchy. For simplicity, we drop the superscript and denote this multi-class classifier as ${}^{l}g_t(\cdot)$. To obtain an initial estimate of the confidences for the location of each part, in the first stage $(t = 1)$ of the sequence, a predictor ${}^{l}g_1(\cdot)$ takes as input features computed on a patch extracted at an image location z, and classifies the patch into one of P_l part classes or a background class (see Figure 4.12), for the parts in the l^{th} level of the hierarchy. We denote by $\mathbf{x}_z^l$, the feature vector of an image patch for the l^{th} level of the hierarchy centered at location z in the image. A classifier for the l^{th} level of the hierarchy in the first stage $t = 1$, therefore produces the following confidence values:

$$ {}^{l}g_1(\mathbf{x}_z^l) \rightarrow \left\{ {}^{l}b_1^p(y_p = z) \right\}_{p \in 0 \ldots P_l}, \tag{4.8} $$

where ${}^{l}b_1^p(y_p = z)$ is the score predicted by the classifier ${}^{l}g_1$ for assigning the p^{th} part in the l^{th} level of the hierarchy in the t^{th} stage at image location z.

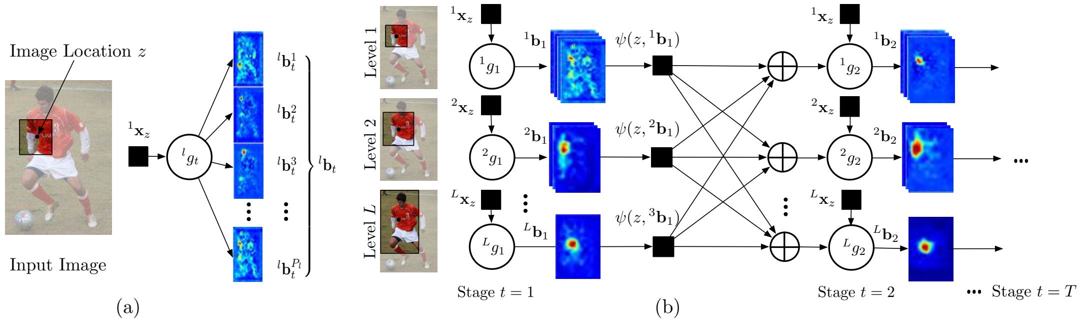


Figure 4.12: (a)A single multiclass predictor is trained to predict each image patch into one of $P+1$ classes. By evaluating each patch in the image, we create a set of confidence maps $\mathbf{b}_t^j$. In each stage, a predictor is trained to predict the confidence of the output variables. The figure depicts the message passing in an inference machine at test time. In the first stage, the predictors produce an estimate for the confidence of each part location based on features computed on the image patch. Predictors in subsequent stages, refine these confidences using additional information from the outputs of the previous stage via the feature maps ψ_1 and ψ_2.

Analogous to Equation 4.2, we represent all the confidences of part p of level l evaluated at every location $z = (u, v)^T$ in the image as ${}^l\mathbf{b}_t^p \in \mathbb{R}^{w\times h}$, where w and h are the width and height of the image, respectively. That is,

$$ {}^l\mathbf{b}_t^p[u,v] = {}^l b_t^p(y_p = (u,v)^T). \tag{4.9} $$

For convenience, we denote the collection of confidence maps for all the parts belonging to level l as ${}^l\mathbf{b}_t \in \mathbb{R}^{w\times h\times P_l}$ (see Figure 4.12).

In subsequent stages, the confidence for each variable is computed similarly to Equation 5.1. In the order to leverage the context across scales/levels in the hierarchy, the prediction is defined as

$$ {}^l g_t\left(\mathbf{x}_z^l, \bigoplus_{l\in 1\ldots L} \psi(z, {}^l\mathbf{b}_{t-1})\right) \rightarrow \left\{{}^l b_t^p(y_p = z)\right\}_{p\in 0\ldots P_l}, \tag{4.10} $$

As shown in Figure 4.12, in the second stage, the classifier ${}^l g_2$ takes as input the

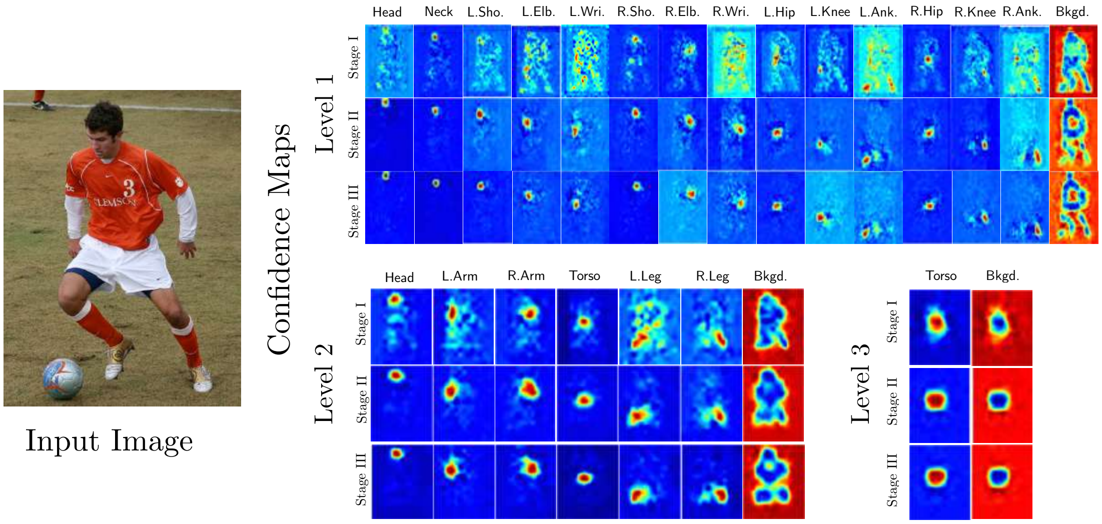


Figure 4.13: **Intermediate Predictions for a Hierarchical Pose Machine**: We show the confidence maps at each of the three stages and for three levels of a hierarchical pose machine. Detection cues are shared across hierarchy levels to improve prediction in each stage.

features $\mathbf{f}_z^l$ and features computed on the confidences via the feature function ψ for each of the parts in the previous stage. Note that the the predictions for a part use features computed on outputs of all parts *and* in all levels of the hierarchy ($\{{}^l\mathbf{b}_{t-1}\}_{l\in 1\ldots L}$). The inference machine architecture allows learning potentially complex interactions among the variables, by simply supplying features on the outputs of the previous stage (as opposed to specifying potential functions in a graphical model) and allowing the classifier to freely combine contextual information by picking the most predictive features. The use of outputs from all neighboring variables, resembles the message passing mechanics in variational mean field inference [Ross et al., 2011].

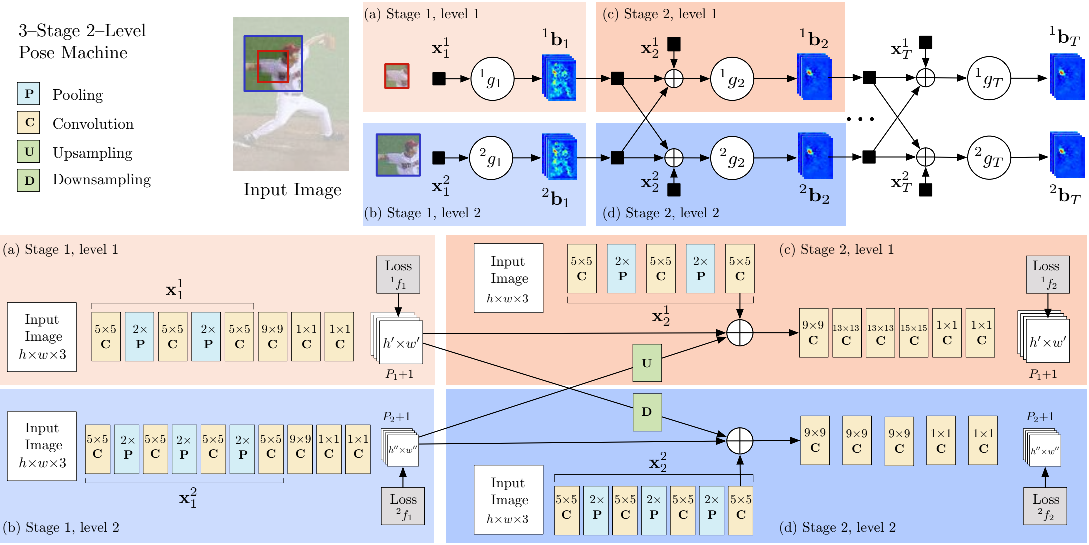


Figure 4.14: **Architecture for Convolutional Pose Machines.** We show a deep convolutional architecture for a pose machine with three stages and a two level hierarchy. The pose machine is shown in the top right with insets described below. Insets (a) and (b) show the architecture that operates only on image evidence in the first stage for each of the hierarchy levels. Insets (c) and (d) show the architecture for subsequent stages which operate both on image evidence as well as confidence maps from preceding stages. The architectures in (c) and (d) are repeated for all subsequent stages.The network is locally supervised after each stage using an intermediate loss layer that prevents vanishing gradients during training. (Best viewed in color)

4.4.2 Convolutional Architecture for a Hierarchical Pose Machine

We show the convolutional architecture for a hierarchical pose machine in Figure 4.14. The architecture consists of two types of sub-networks. The first type corresponds to the first stage of the pose machine and performs part detection from purely *local image evidence* as shown in the insets (a) and (b) of Figure 4.14 for the first and second levels of the hierarchy respectively. The second type of network (shown in insets (c) and (d) of Figure 4.14) corresponds to subsequent stages of the pose machine and is repeated for the number of stages specified.

These networks update the confidences for each part by leveraging both image evidence and *spatial context* learned from the confidence maps of preceding stages.

The design of these networks is motivated by the achieving the desired receptive field for each of the levels. For the first level of the hierarchy (Figure 4.14(a)) we use a network structure composed of 6 convolutional layers followed by two 1×1 convolutional layers which results in a fully convolutional [Long et al., 2015] architecture that allows inputs of an arbitrary size $h \times w$. The image is first contrast normalized and then passed through the three convolutional layers with kernels of size $5 \times 5 \times 128$, $5 \times 5 \times 128$, and $5 \times 5 \times 32$. We perform max-pooling with stride 2 after the first two convolutional layers resulting in an output layer of size $\frac{h}{4} \times \frac{w}{4} \times 32$ after the third layer. The fourth, fifth, and sixth layers have kernel sizes of $9 \times 9 \times 512$, $1 \times 1 \times 512$, $1 \times 1 \times (P_1 + 1)$ resulting $P_1 + 1$ output confidence maps of size $h' \times w'$, corresponding to P_1 body parts and background. The fourth layer of the network is equivalent to the first fully connected layer that takes the cascade of spatial features (into a long vector of length 512) in a typical deep network such as Krizhevsky et al. [Krizhevsky et al., 2012]. Similarly, the two 1×1 convolutional layers are the convolutional equivalent of fully connected layers. We use rectified linear units [Nair and Hinton, 2010] after each convolutional layer except the last one.

The receptive field of the first level of the hierarchy is thus 64×64 pixels. The network can effectively be viewed as sliding a deep network across an image and regressing each 64×64 image patch to a $P_1 + 1$ output vector. The network structure for the second level (Figure 4.14b) of the hierarchy is identical except for an additional $5 \times 5 \times 32$ convolutional layer followed by a pooling layer to approximately double the receptive field to 132×132.

4.5 Inference

At test time, inference proceeds in a feed-forward sequential fashion emulating message passing inference as discussed in Section 3.2.2.

For the architecture in Figure 4.12 the sequence alternates between image/context feature computation and prediction using the multi-class classifier. Features are extracted from patches of different scales (corresponding to each of the L levels of the hierarchy) at each location in the image and input to the first stage classifiers $\{^l g_1\}_{l=1}^L$, resulting in the output confidence maps $\{^l \mathbf{b}_1\}_{l=1}^L$. Context features are passed to the classifiers in the next stage via the feature maps ψ_1, ψ_2 on the confidences $^l \mathbf{b}_1$ from the previous stage. Updated confidences $\{^l \mathbf{b}_2\}_{l=1}^L$ are computed by the classifiers $^l g_2$ and this procedure is repeated for each stage. The computed confidences are increasingly refined estimates for the location of the part as shown in Figure 4.16. The location of each part is then computed as,

$$\forall l, \forall p, \ ^l y_p^* = \underset{z}{\operatorname{argmax}} \ ^l \mathbf{b}_T^p(z). \tag{4.11}$$

The final pose is computed by directly picking the maxima of the confidence map of each part after the final stage.

The architecture in Figure 4.14 can be thought of as a single convolutional network. Prediction in this architecture is simply evaluating the network output at the last layer by performing a forward-pass on the architecture. As before, the location of each part is computed by directly picking the location corresponding to the maximum of each confidence map as generated by the network.

Algorithm 5 `train_stage_wise`

```
Initialize: $\{{}^l\mathbf{b}_0 = \emptyset\}_{l \in 1,\ldots,L}$
for $t = 1 \ldots T$ do
    for $i = 1 \ldots N$ do
        Create $\{{}^l\mathbf{b}_{t-1}\}_{l=1}^{L}$ for each image $i$ using predictor ${}^l g_{t-1}$ using Eqn. 5.2.
        Append features extracted from each training image $i$, and from corresponding
        $\{{}^l\mathbf{b}_{t-1}\}_{l=1}^{L}$ (Eqns. 4.4 & 4.6), to training dataset $\mathcal{D}_t$, for each image $i$.
    end for
    Train ${}^l g_t$ using $\mathcal{D}_t$.
end for
Return: Learned predictors $\{{}^l g_t\}$.
```

4.6 Learning

Learning in our setting involves estimating the parameters of the predictors, $\{{}^l g_t\}$, in each level $l \in \{1, \ldots, L\}$, and for each stage $t \in \{1, \ldots, T\}$ from training data. Each of the predictors is simply a supervised classifier, and therefore learning reduces to simply training multiple supervised classifiers. Learning can either proceed sequentially in a stage-wise manner as described in Section 4.6.1 or jointly, where all the predictors are trained simultaneouly using backpropagation as described in Section 4.6.2. The stage-wise learning procedure is suitable when the predictors are not differentiable with respect to their parameters as is the case with classifiers such as random forests and boosted random forests, while joint training is suitable for when the classifiers are differentiable and gradients can be backpropagated as is the case for a deep convolutional architecture. We discuss both learning scenarios and address some of the challenges that arise.

4.6.1 Forward Stagewise Training

We describe the stage-wise training procedure in Algorithm 5. Training proceeds sequentially: the first set of predictors $\{{}^l g_1\}$ are trained using a dataset $\mathcal{D}_0$ consisting of image features on patches extracted from the training set of images at the annotated landmarks. For a subsequent stage t, the dataset $\mathcal{D}_t$ is created by extracting and concatenating the context features from the confidence maps of the previous stage, $\{{}^l\mathbf{b}_{t-1}\}_{l=1}^{L}$, for each image, at the annotated locations. The predictor in the next stage $\{{}^l g_t\}$ is then trained using dataset $\mathcal{D}_t$ and the procedure is iterated.

Stacked Training

Training the predictors of such an inference procedure can be prone to overfitting. Using the same training data to train the predictors in subsequent stages can cause them to rely on overly optimistic context from the previous stage, or overfit to idiosyncrasies of that particular dataset. Ideally we would like to train the subsequent stages with the output of the previous stages with noise characteristics similar to that as encountered at test time. In order to achieve this, we use the idea of stacked training [Wolpert, 1992; Carvalho and Cohen, 2005].

Stacked training aims to prevent predictors trained on the output of the first stage from being trained on same training data. Stacking proceeds similarly to cross-validation by making M splits of the training data $\mathcal{D}$ into training and held-out data $\{\mathcal{D}^m, \mathcal{D}/\mathcal{D}^m\}_{m=1...M}$. For each predictor we aim to train in the first stage, we make M copies, each trained on one of the M splits of the training data. To create the training data for the next stage, for each training sample, we use the copy of the predictor that has not seen the sample (i.e., the sample is in the held-out data for that predictor). Proceeding in this way creates a dataset to train the next stage on the outputs of the previous stage, ensuring that the outputs mimic test-time behavior. We repeat the stacking procedure for each subsequent stage. The stacking procedure is only performed during training to create a training dataset for subsequent stages. At test time, we use a predictor in each stage that is trained using all of the data.

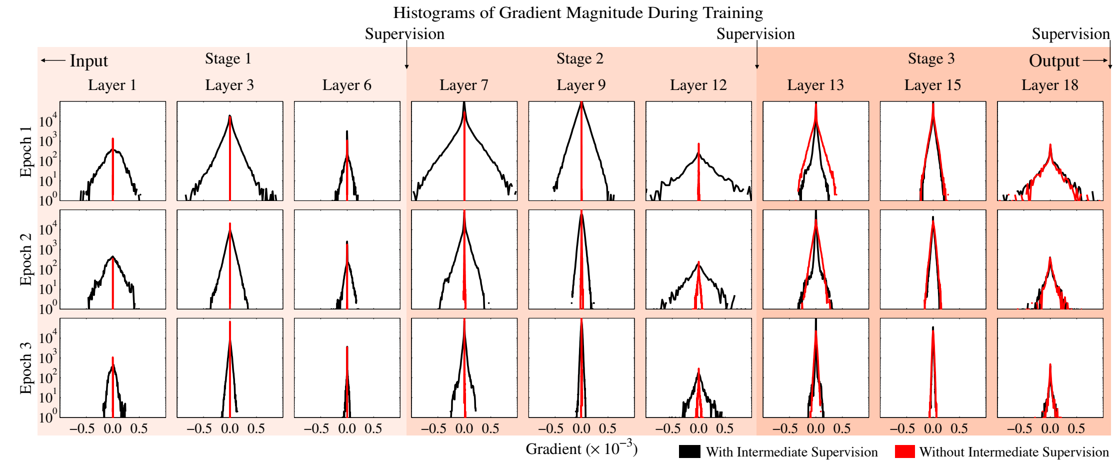


Figure 4.15: **Intermediate supervision addresses vanishing gradients.** We track the change in magnitude of weights in layers at different depths in the architecture, across training epochs, for models with and without intermediate supervision. We observe that for layers closer to the output the distribution has a large variance for both with and without intermediate supervision, however as we move from the output layer towards the input, the gradient magnitude distribution peaks tightly around zero with low variance (the gradients *vanish*) for the model without intermediate supervision. For the model with intermediate supervision the distribution has a moderately large variance throughout the network. At later training epochs, the variances decrease for all layers for the model with intermediate supervision and remain tightly peaked around zero for the model without intermediate supervision. (Best viewed in color)

4.6.2 Joint Training with Intermediate Supervision

The design described above for a pose machine results in a deep architecture that can have a large number of layers. Training such a network with many layers can be prone to the problem of *vanishing gradients* [Bradley , 2010; Glorot and Bengio, 2010; Bengio et al., 1994] where, as observed by Bradley [2010] and Glorot and Bengio [2010], the magnitude of backpropagated gradients decreases in strength with the number of intermediate layers between the output layer and the input layer.

Fortunately, the sequential prediction framework of the pose machine provides a natural approach to training our deep architecture that addresses this problem.

Each stage of the pose machine is trained to repeatedly produce the confidence maps or *beliefs* for the locations of each of the parts. We encourage the network to repeatedly arrive at such a representation by defining a loss function at the output of each stage t and hierarchy level l that minimizes the l_2 distance between the predicted and ideal confidence maps for each part. The ideal confidence map for a part p is written as ${}^{l}b_{*}^{p}(Y_p = z)$. The cost function we aim to minimize at the output of each stage at each level is therefore given by:

$$ {}^{l}f_t = \sum_{p=1}^{P_l} \sum_{z \in \mathcal{Z}} \| {}^{l}b_t^{p}(z) - {}^{l}b_{*}^{p}(z) \|_2^2. \tag{4.12}$$

The overall objective for the full architecture is obtained by adding the losses at each stage and is given by:

$$ \mathcal{F} = \sum_{t=1}^{T} \sum_{l=1}^{L} {}^{l}f_t. \tag{4.13}$$

The objective in Equation 6.6 describes a decomposable loss function that operates on different parts of the network (see Figure 4.14). Specifically, each term in the summation is applied to the network after each stage t effectively enforcing supervision in intermediate stages through the network. Intermediate supervision has the advantage that, even though the full architecture can have many layers, it does not fall prey to the *vanishing gradient* problem as the intermediate loss functions replenish the gradients at each stage.

We verify this claim by observing histograms of gradient magnitude (see Figure 4.15) at different depths in the architecture, across training epochs, for models with and without intermediate supervision. In early epochs, as we move from the output layer to the input layer, we observe that the gradient distribution is tightly peaked around zero because of vanishing gradients for the model *without intermediate supervision*. The model *with intermediate supervision* has a much larger variance across all the layers suggesting that learning is indeed occurring in all the layers thanks to intermediate supervision. We also notice that as training progresses, the variance in the gradient magnitude distributions decreases pointing to model convergence.

4.7 Analysis

4.7.1 Which Learning Method?

We compare different variants of training the network in Figure 4.17a and demonstrate the benefit of intermediate supervision with joint training across stages using a model trained in four ways: (i) using a global loss function that enforces intermediate supervision (ii) stage-wise; where each stage is trained in a feed-forward fashion and stacked (iii) as same as (i) but initialized with weights from (ii), and (iv) as same as (i) but with no intermediate supervision. We find that network (i) outperforms all other training methods, showing that intermediate supervision and joint training across stage is indeed crucial in achieving good performance.

4.7.2 Performance Across Stages

We show a comparison of quantitative performance across each stage on the LEEDS dataset in Figure 4.17b. We show that the performance increases monotonically across stages as the predictor in subsequent stages make use of contextual information in a large receptive field on the previous stage confidence maps to resolve confusions between parts and background. In Figure 4.16 we see the score maps for the wrist and elbows across three stages. In the first stage we find that the confidence for the wrists and elbows are noisy and multimodal. In subsequent stages, the confusions are resolved resulting in a single peak at the correct location of the anatomical landmark.

4.7.3 Does the Hierarchy Help?

We compare models with one and two levels in the hierarchy on LEEDS in Figure 4.18. We find that for the model using OC annotations the effect of using a two level hierarchy is marginal, however, we see a substantial improvement in performance with the two-level model when using PC annotations. The difficulty in learning person-centric pose estimation lies in the fact that the front-back ambiguity needs to be resolved. The favorable performance of the two-level model over the single-level model seems to suggest that information at a coarser scale assists the model in resolving this ambiguity.

4.7.4 Effects of Missing Context

In the design of a convolutional pose machine, the predictor networks in subsequent stages refine their estimates of part locations based on both image evidence and spatial context inferred from the confidence maps of the preceding stage. The goal of this experiment is an ablative analysis to learn which parts are most informative in terms of the spatial context they provide. We use a 3-stage convolutional pose machine on the LEEDS dataset with observer-centric annotation. We deliberately zero out the confidence map for each part in turn (one at a time) and observe the change in accuracy for each part at the final output layer.

Figure 4.19 shows that among all the body parts, the elbows and knees rely on the spatial relationship with other parts the most as they suffer in the largest drop in accuracy. Furthermore, there is a strong correlation between elbows and wrists, and between knee and ankles. Unsurprisingly, the detection of head seems to provide the most informative spatial context and zero-ing out the head's confidence map results in the largest average drop in accuracy across all the parts.

4.8 Evaluation

4.8.1 Leeds Sports Pose Dataset.

We evaluate our method on the Extended Leeds Sports Dataset that consists of 11000 images for training and 1000 images for testing with annotations provided for the full body. The LEEDS dataset consists of images of people performing a wide variety of complex sports actions.

We train a model with 3 stages and 2 hierarchy levels to predict 14 parts. We evaluate our method on both observer-centric (OC) and person-centric (PC) annotations using the Percentage Correct Keypoints (PCK) metric [Yang and Ramanan, 2013]. We see that for *observer-centric annotations* (see Figure 4.20) we outperform the nearest competing method by approximately 10 percentage points in the high precision regime (PCK@0.1) and approx. 5 percentage points in the lower precision regime (PCK@0.2). *Person-centric annotations* impose a harder problem on the pose estimation task since disambiguating the left from right limbs relies on observing the target's pose relative to the camera, in addition to the spatial relationship of parts in the image plane. Figure 4.16 shows that our our model develops a representation that is able to resolve the left-right ambiguity across the stages the with large receptive fields. Our method again outperforms all of the other methods, as shown in Figure 4.20.

In Figure 4.20 and Figure 4.21 we show complete quantitative comparisons between our method and the closes competing methods on LEEDS Sports Dataset with PCK metric. For observer-centric annotation, we outperform all other methods ([Chen and Yuille, 2014a], [Pishchulin et al., 2013b], [Ouyang et al., 2014], and [Ramakrishna et al., 2014b]) by a considerable margin, especially on difficult but important parts including the wrist, elbow, knee, and ankle. In Figure 4.22 we compare accuracy curves for the top-3 predictions for each part in an image with the single best prediction for each part in an image. The comparison curves in Figure 4.22 suggests that there is still additional improvements to be gained by disambiguating between the top-3 detections.

Table 4.1: Performance comparison of our method in three different precision regimes using the PCK metric on observer-centric annotations.

High precision: PCK@0.05 for LEEDS OC

	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Ours 3-Stage 2-Level	**49.1**	**32.9**	**35.3**	**31.2**	18.9	**38.0**	**36.2**	**34.5**
[Pishchulin et al., 2013a]	40.5	28.4	23.6	17.6	**21.0**	28.9	28.2	26.9
[Chen and Yuille, 2014a]	28.6	21.8	20.4	16.8	16.5	19.6	13.6	19.6
[Ouyang et al., 2014]	39.0	27.2	17.9	14.6	17.6	20.2	24.5	23.0
[Ramakrishna et al., 2014a]	18.0	13.8	9.8	7.7	12.9	12.8	12.3	12.5

Medium precision: PCK@0.1 for LEEDS OC

	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Ours 3-Stage 2-Level	**81.4**	**68.2**	**60.2**	**53.5**	53.5	**67.3**	**64.1**	**64.0**
[Pishchulin et al., 2013a]	74.9	56.1	44.3	33.0	**55.4**	56.1	51.6	53.1
[Chen and Yuille, 2014a]	68.2	56.6	48.1	43.7	49.0	51.0	42.7	51.3
[Ouyang et al., 2014]	73.7	57.5	41.3	34.2	48.0	48.5	50.8	50.6
[Ramakrishna et al., 2014a]	50.6	43.2	31.2	23.6	38.5	36.5	37.0	37.2

Low precision: PCK@0.2 for LEEDS OC

	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Ours 3-Stage 2-Level	**93.1**	**87.5**	**75.4**	**68.5**	**86.1**	**83.2**	**77.3**	**81.6**
[Pishchulin et al., 2013a]	87.5	77.6	61.4	47.6	79.0	75.2	68.4	71.0
[Chen and Yuille, 2014a]	91.5	84.7	70.3	63.2	82.7	78.1	72.0	77.5
[Ouyang et al., 2014]	86.5	78.2	61.7	49.3	76.9	70.0	67.6	70.0
[Ramakrishna et al., 2014a]	84.9	77.8	61.4	47.2	73.6	69.1	68.8	69.0

Table 4.2: Performance comparison of our method in three different precision regimes using the PCK metric on person-centric annotations.

High precision: PCK@0.05 for LEEDS PC

	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Ours 3-Stage 2-Level	**48.9**	29.0	**32.0**	27.4	**18.0**	**30.9**	27.0	**30.4**
[Tompson et al., 2014]	41.0	**32.2**	30.1	**28.2**	16.9	29.9	**30.1**	29.8
[Pishchulin et al., 2013a]	40.7	19.9	18.4	13.6	15.4	21.1	21.9	21.6
[Chen and Yuille, 2014a]	24.5	18.1	19.2	14.4	11.8	15.5	8.7	16.0
[Wang and Li, 2013]	30.2	14.6	10.1	10.0	9.9	12.9	15.9	14.8

Medium precision: PCK@0.1 for LEEDS PC

	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Ours 3-Stage 2-Level	**82.0**	61.0	**55.5**	49.8	**47.5**	**58.7**	48.8	**57.6**
[Tompson et al., 2014]	76.2	**63.2**	54.2	**50.5**	42.0	56.5	**53.4**	56.6
[Pishchulin et al., 2013a]	74.8	39.2	32.7	26.1	40.0	40.9	39.0	41.8
[Chen and Yuille, 2014a]	61.6	49.3	49.1	40.1	36.5	41.9	29.8	44.0
[Wang and Li, 2013]	65.6	36.7	26.4	24.7	29.8	34.0	36.3	36.2

Low precision: PCK@0.2 for LEEDS PC

	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Ours 3-Stage 2-Level	**94.0**	**79.8**	**70.6**	65.2	**77.9**	**74.1**	61.4	**74.7**
[Tompson et al., 2014]	90.6	79.2	67.9	63.4	69.5	71.0	**64.2**	72.3
[Pishchulin et al., 2013a]	87.2	56.7	46.7	38.0	61.0	57.5	52.7	57.1
[Chen and Yuille, 2014a]	91.8	78.2	71.8	**65.5**	73.3	70.2	63.4	73.4
[Wang and Li, 2013]	84.7	57.1	43.7	36.7	56.7	52.4	50.8	54.6

In Tables 4.1, we tabulate the accuracy of our method and competing methods at different precision regimes. Our method displays large performance gains in the high precision regime where we are better by up to 13.6 percentage points, against the closest competitor. We also report average performance over the whole precision range by including the area under curve (AUC) in the last column of Table 4.1.

In Tables 4.2, we tabulate the accuracy of our method and competing methods ([Tompson et al., 2014], [Pishchulin et al., 2013b], [Chen and Yuille, 2014a], and [Wang and Li, 2013]) at different precision regimes for person-centric annotations. While the gap in performance is smaller we still outperform all competing methods on average in all precision regimes. The most prominent error mode for the model trained with person-centric annotations is still the confusion between parts with symmetric appearance. This occurs because the front-back ambiguity needs to be resolved.

4.8.2 FLIC Dataset

We evaluate our method on the FLIC Dataset [Sapp and Taskar, 2013] that consists of 3987 images for training and 1016 images for testing with 9 annotations provided for the upper body. The FLIC dataset consists of frames taken from cinema. We report accuracy as per the metric introduced in Sapp et al. [Sapp and Taskar, 2013] for the elbow and wrist joints in Figure 4.25. Again we outperform all prior art in PCK metric in both high precision (PCK@0.05) by 7.5 percentage points on wrists and 7 percentage points on elbows, and in the lower precision regime (PCK@0.1) by 3 percentage points on wrists and 6.5 percentage points on elbows. We show qualitative results in Figure 4.24.

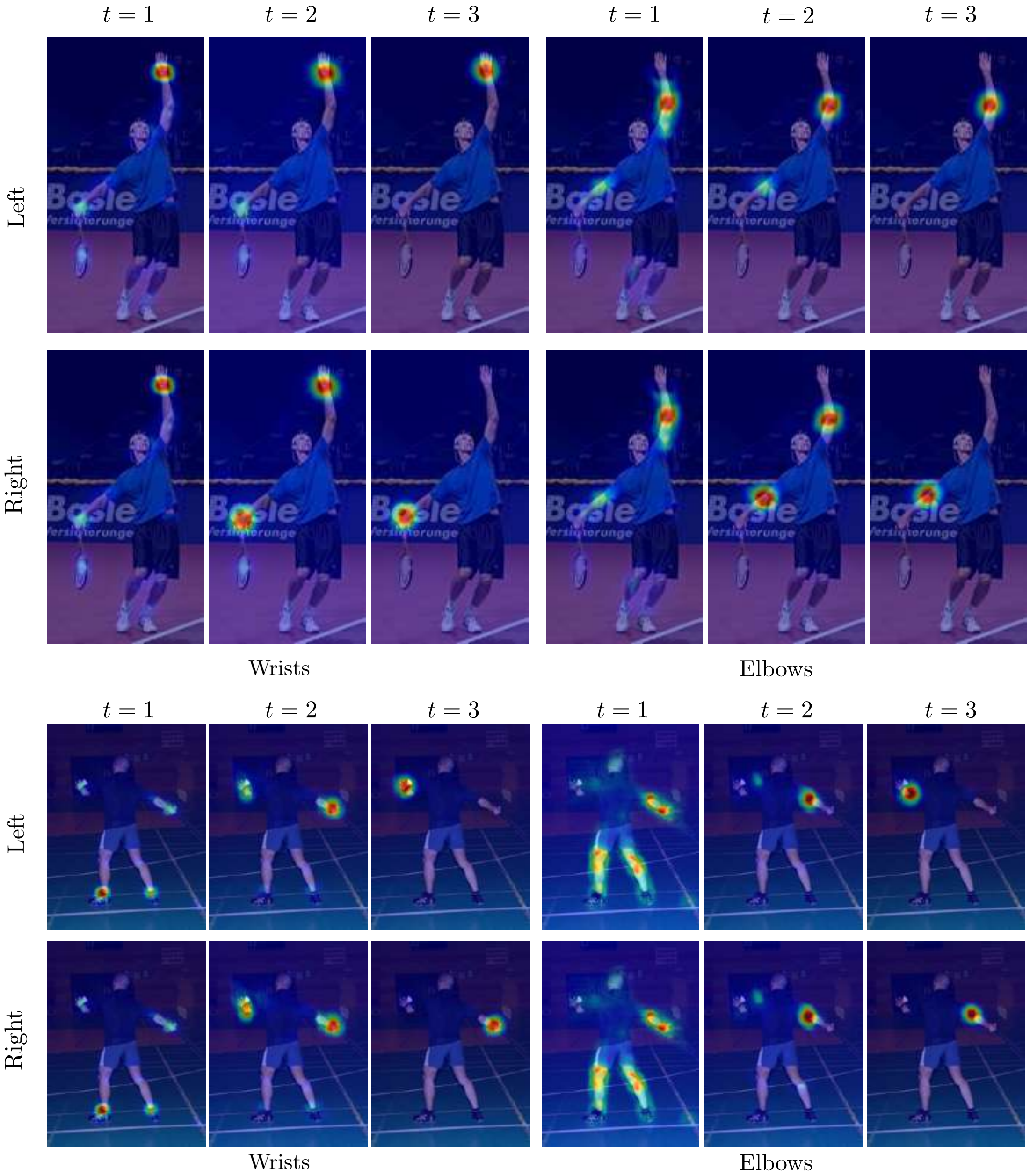
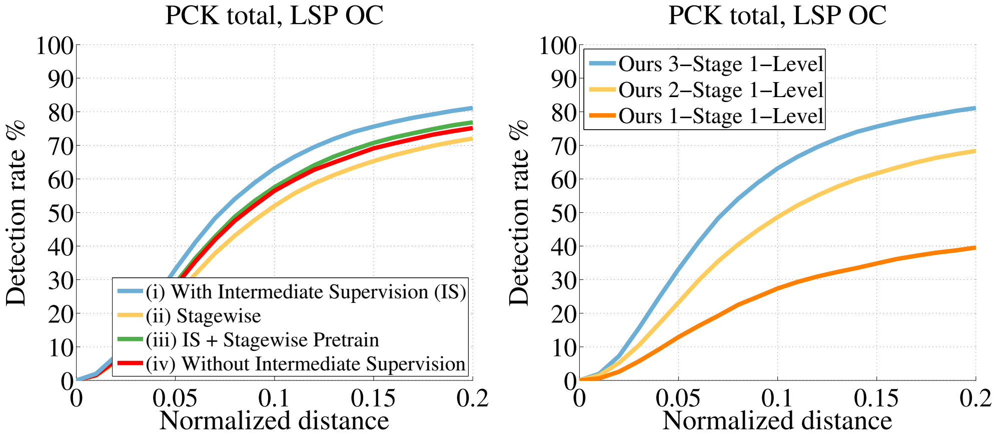


Figure 4.16: Comparison of confidence maps across stages for the elbow and wrist joints on the LEEDS dataset for a three stage deep pose machine. The subsequent stages of the convolutional pose machine learns a spatial model that aids in resolving confusions between parts. The first stage predictions for the wrist joints are often ambiguous or erroneous.



(a) Learning Methods

(b) Stagewise Comparison

Figure 4.17: (a) Comparisons on the LEEDS dataset between the different training methods. (b) Comparisons on the LEEDS dataset across each stage using joint training from scratch with intermediate supervision.

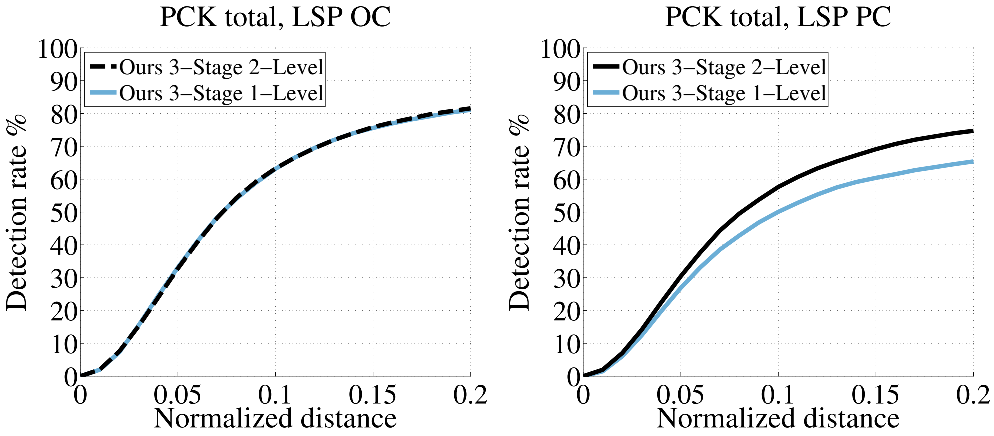
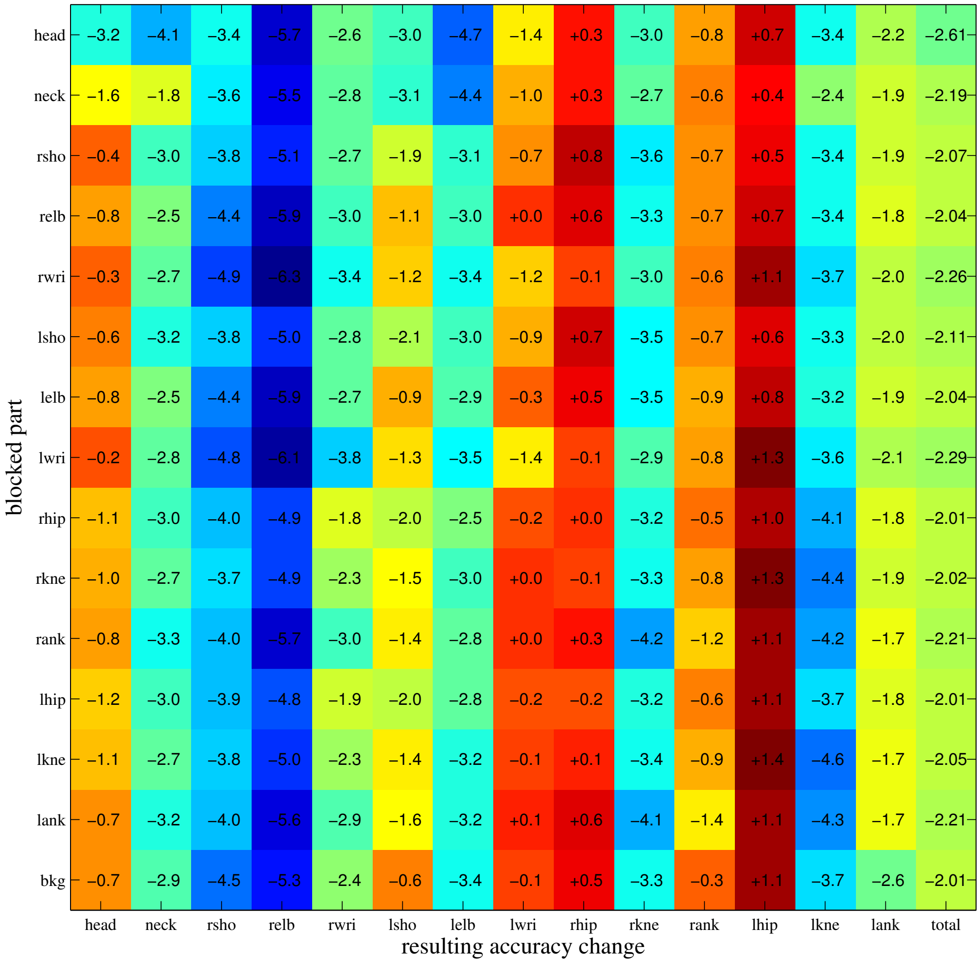


Figure 4.18: Hierarchy Levels Comparison: Comparisons on the LEEDS dataset across number of stages using training from scratch with OC and PC annotation, respectively.



	head	neck	rsho	relb	rwri	lsho	lelb	lwri	rhip	rkne	rank	lhip	lkne	lank	total
head	−3.2	−4.1	−3.4	−5.7	−2.6	−3.0	−4.7	−1.4	+0.3	−3.0	−0.8	+0.7	−3.4	−2.2	−2.61
neck	−1.6	−1.8	−3.6	−5.5	−2.8	−3.1	−4.4	−1.0	+0.3	−2.7	−0.6	+0.4	−2.4	−1.9	−2.19
rsho	−0.4	−3.0	−3.8	−5.1	−2.7	−1.9	−3.1	−0.7	+0.8	−3.6	−0.7	+0.5	−3.4	−1.9	−2.07
relb	−0.8	−2.5	−4.4	−5.9	−3.0	−1.1	−3.0	+0.0	+0.6	−3.3	−0.7	+0.7	−3.4	−1.8	−2.04
rwri	−0.3	−2.7	−4.9	−6.3	−3.4	−1.2	−3.4	−1.2	−0.1	−3.0	−0.6	+1.1	−3.7	−2.0	−2.26
lsho	−0.6	−3.2	−3.8	−5.0	−2.8	−2.1	−3.0	−0.9	+0.7	−3.5	−0.7	+0.6	−3.3	−2.0	−2.11
lelb	−0.8	−2.5	−4.4	−5.9	−2.7	−0.9	−2.9	−0.3	+0.5	−3.5	−0.9	+0.8	−3.2	−1.9	−2.04
lwri	−0.2	−2.8	−4.8	−6.1	−3.8	−1.3	−3.5	−1.4	−0.1	−2.9	−0.8	+1.3	−3.6	−2.1	−2.29
rhip	−1.1	−3.0	−4.0	−4.9	−1.8	−2.0	−2.5	−0.2	+0.0	−3.2	−0.5	+1.0	−4.1	−1.8	−2.01
rkne	−1.0	−2.7	−3.7	−4.9	−2.3	−1.5	−3.0	+0.0	−0.1	−3.3	−0.8	+1.3	−4.4	−1.9	−2.02
rank	−0.8	−3.3	−4.0	−5.7	−3.0	−1.4	−2.8	+0.0	+0.3	−4.2	−1.2	+1.1	−4.2	−1.7	−2.21
lhip	−1.2	−3.0	−3.9	−4.8	−1.9	−2.0	−2.8	−0.2	−0.2	−3.2	−0.6	+1.1	−3.7	−1.8	−2.01
lkne	−1.1	−2.7	−3.8	−5.0	−2.3	−1.4	−3.2	−0.1	+0.1	−3.4	−0.9	+1.4	−4.6	−1.7	−2.05
lank	−0.7	−3.2	−4.0	−5.6	−2.9	−1.6	−3.2	+0.1	+0.6	−4.1	−1.4	+1.1	−4.3	−1.7	−2.21
bkg	−0.7	−2.9	−4.5	−5.3	−2.4	−0.6	−3.4	−0.1	+0.5	−3.3	−0.3	+1.1	−3.7	−2.6	−2.01

Figure 4.19: **Ablative Analysis of Missing Context.** We list the change in accuracy using PCK with a threshold of 0.05 when each part is removed in turn. We observe that among all the body parts, the elbows and knees rely on the spatial relation with other parts the most as they suffer in the largest drop in accuracy. Furthermore, there is a strong correlation between elbows and wrists, and between knee and ankles. The detection of head provides the most informative spatial context and resulting in the largest average drop in accuracy across all the parts, when zero-ed out.

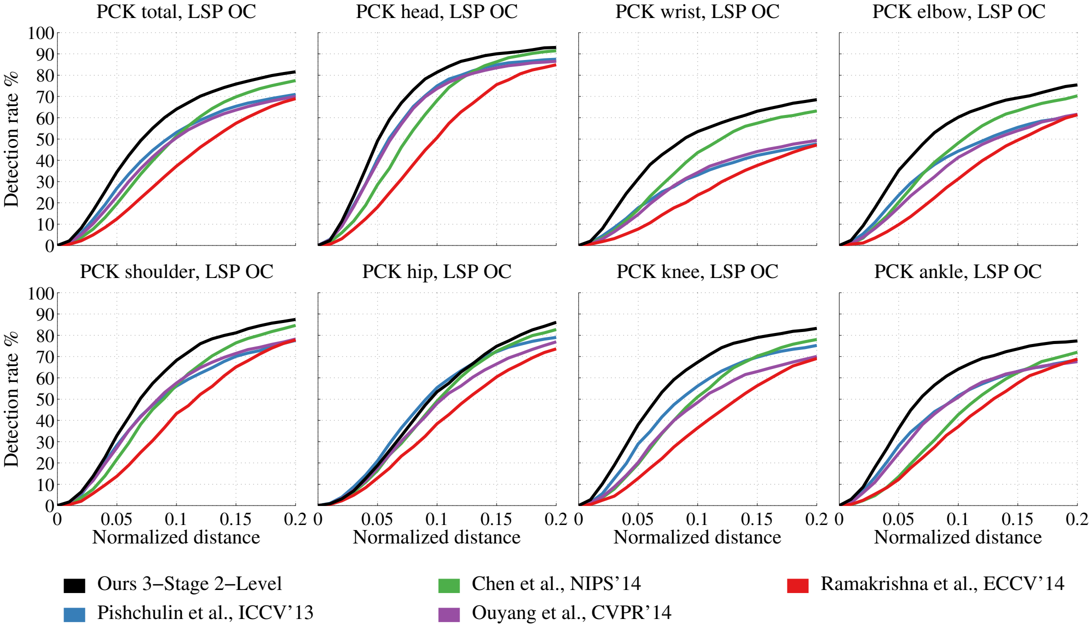


Figure 4.20: PCK Performance on LEEDS Sport Data with Observer-centric Annotation

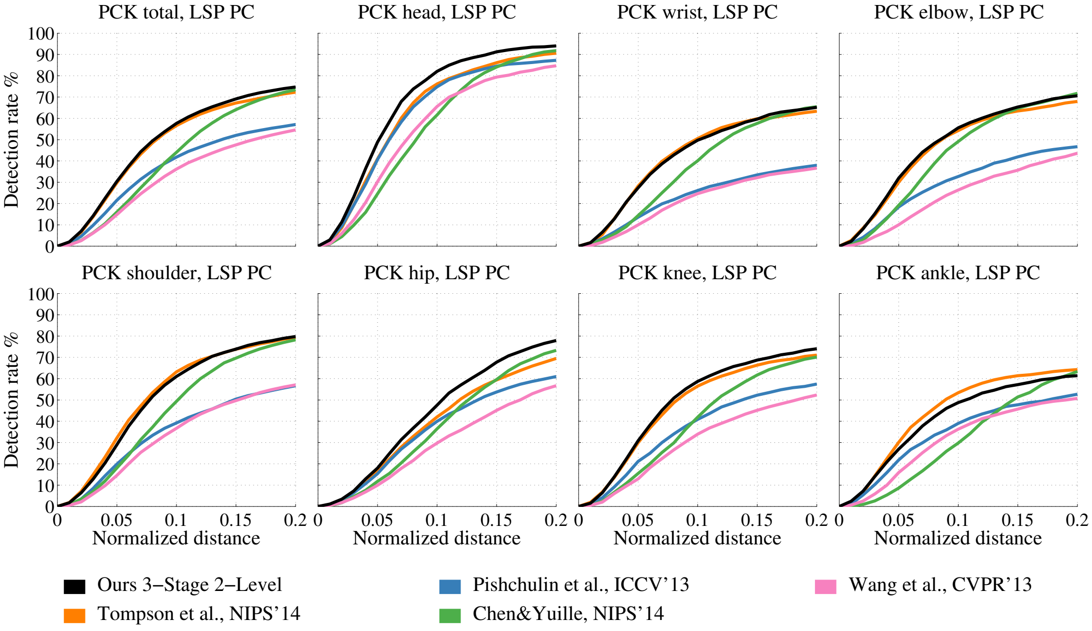


Figure 4.21: PCK Performance on LEEDS Sport Data with Person-centric Annotation

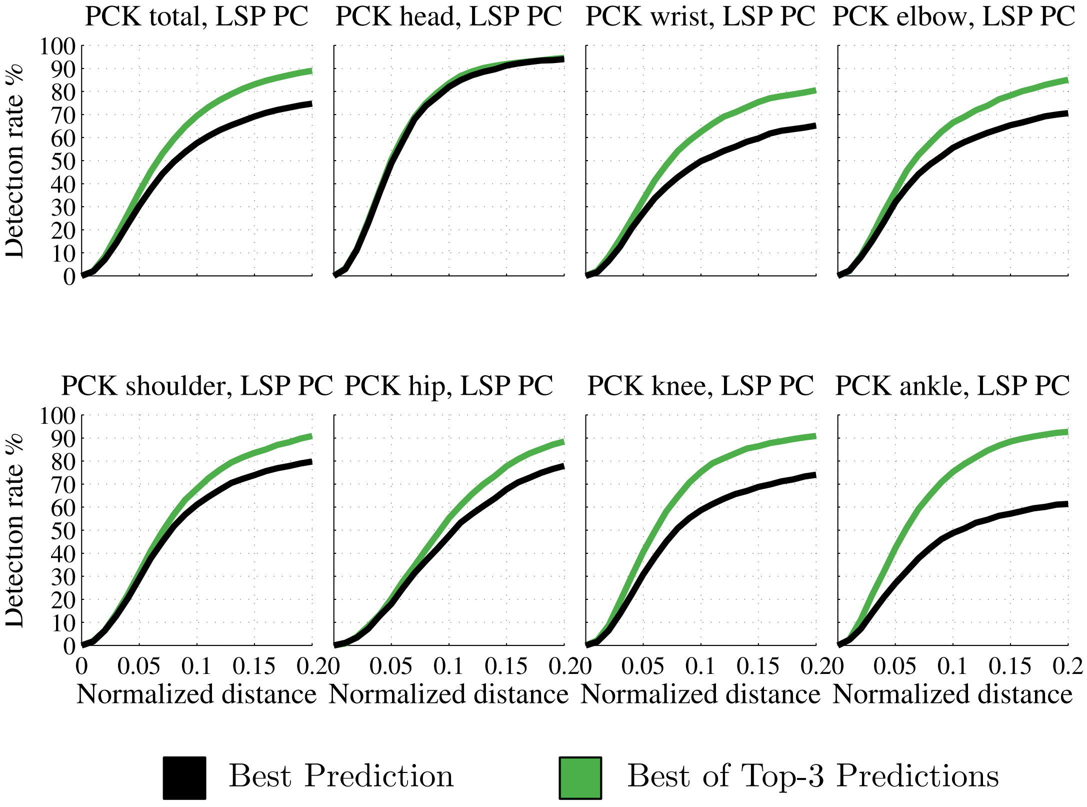


Figure 4.22: PCK Performance on LEEDS Sport Data with Person-centric Annotation for the Top-3 predictions for each part. In **green** we show the accuracy curves using top-3 predictions and in **black** we show the accuracy curves for the best prediction in each image.

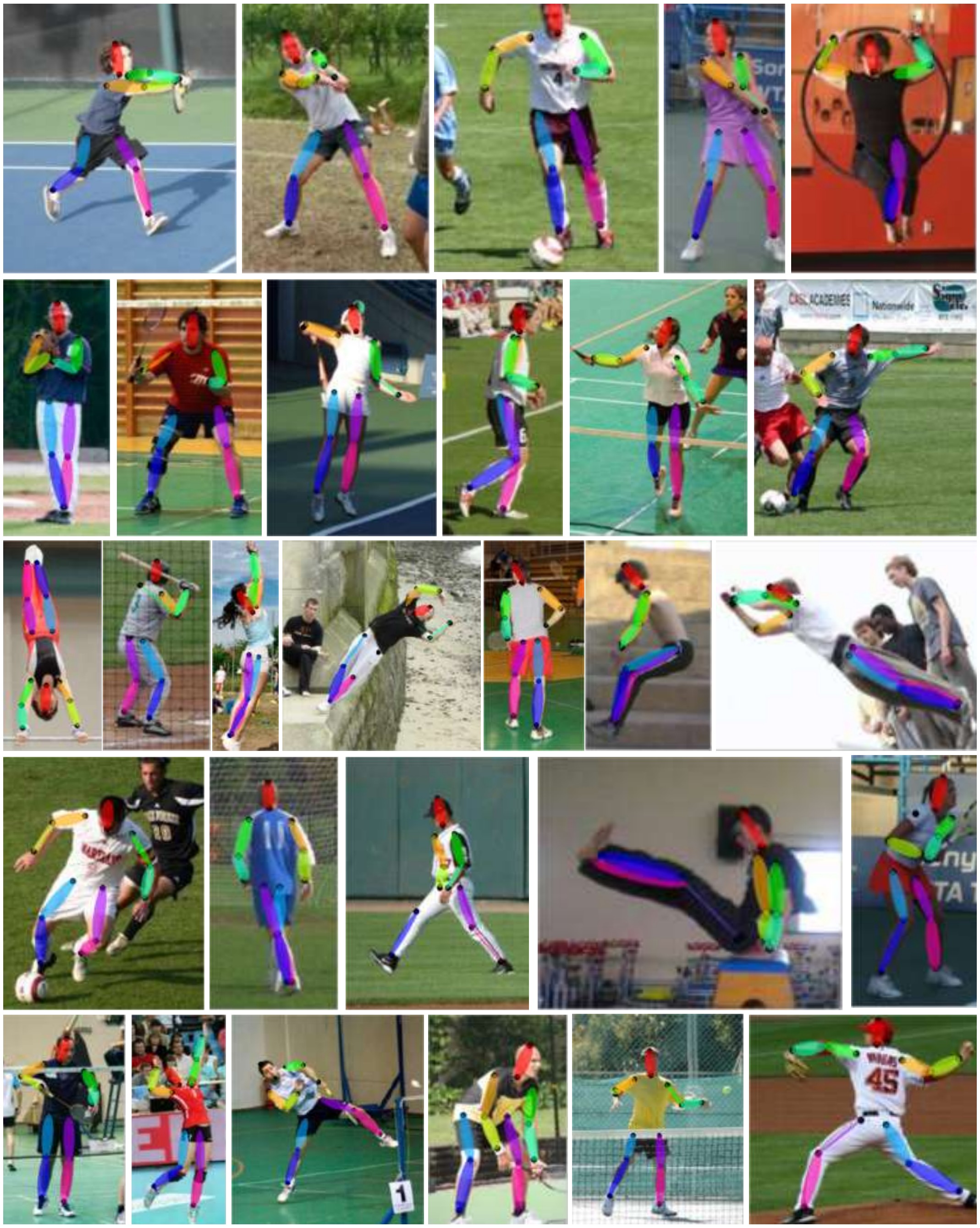


Figure 4.23: Qualitative results of our method on the LEEDS dataset using person-centric annotations. We see that the method is able to handle non-standard poses and resolve ambiguities between symmetric parts for a variety of different relative camera views (best viewed in color).

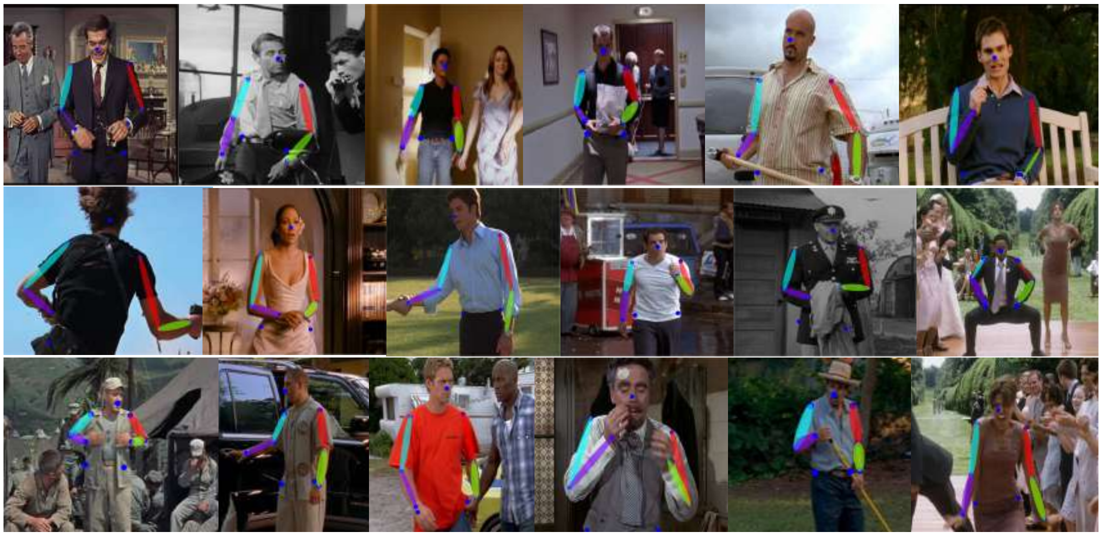


Figure 4.24: Qualitative results of our method on the FLIC dataset (best viewed in color)

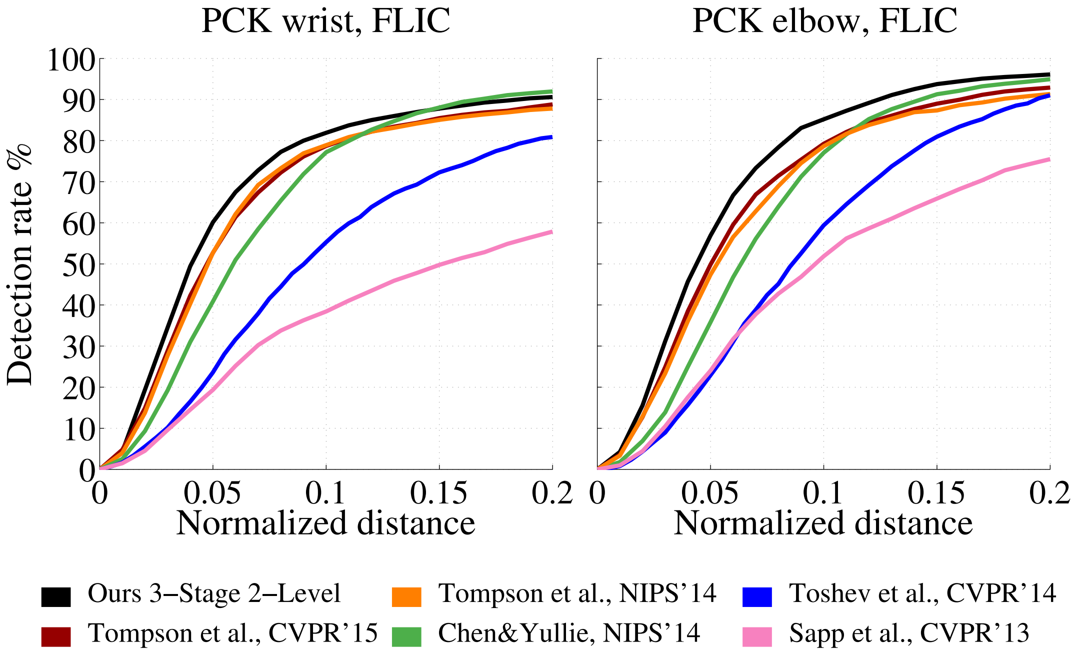


Figure 4.25: Quantitative results for the elbow and wrist joints on the FLIC dataset for a convolutional pose machine with three stages and two levels. We outperform all competing methods.

CHAPTER 5

Parsing Visual Dyads

When people physically interact they convey crucial non-verbal information. The articulated pose of interacting people can convey information regarding social status, the relationship between the actors and the intent of the interaction. However, reasoning about articulated pose for interacting articulated objects from images is extremely challenging. In this chapter we discuss these challenges and focus on the problem of reasoning about the articulated pose of *two* interacting individuals from a single image. We term such an interaction in an image a *visual dyad*[1]. A visual dyad consists of a *visual* interaction between a pair of individuals. A visual interaction can include the result of physical interactions between the actors which manifest as proximity, contact and the relative configuration of the objects in the image, and interactions that are induced by the imaging process such as occlusion and illumination changes.

The usual challenges of performing articulated pose estimation are compounded when dealing with interacting objects. Estimating articulated pose in the presence of dyadic interactions requires reasoning about combinatorially many

[1] We borrow the term *dyad* from sociology, where it refers to a pair of individuals linked via a social interaction [Macionis and Gerber, 2010].

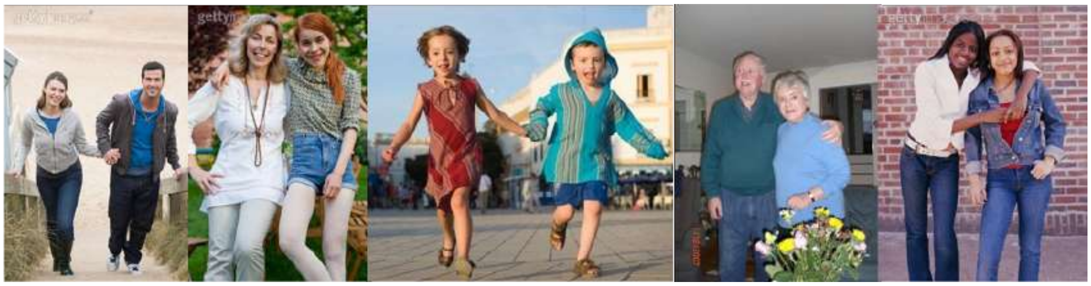


Figure 5.1: Interacting people present a special challenge for pose estimation algorithms. We see that in the images shown above from the *Proxemics* dataset [Yang et al., 2012], parts from each of the individuals are in close proximity to each other making assigning ownership of each part to the individuals a challenging task. Additionally, parts from each of the individuals occlude tend to occlude each other.

configurations and combinatorially many interactions. For an articulated body model with d degrees of freedom and K possible states for each degree of freedom, we have $O\left(d^K\right)$ possible configurations. For M interacting objects, the number of possible total configurations of the ensemble of objects increases to $O\left((Md)^K\right)$. A model attempting to deal with such a large state-space must be equipped to take advantage of statistical regularities in the way such objects interact. The imaging process introduces additional challenges when dealing with interacting objects. Due to differing depths of the objects relative to the camera, *inter object occlusions* are one of the main difficulties faced when parsing visual dyads. Limbs or parts of one individual can occlude parts of the other individual in the pair and vice-versa. Occluded parts complicate reasoning about full configurations as assumptions about part-connectivity are broken. A second challenge is that of resolving *ambiguous part ownership*. As both objects in the dyad belong to the same class (i.e., people), their parts tend to have similar appearance. When the objects are in close proximity, it becomes difficult to assign ownership of detected parts to each of the individuals in the dyad. Resolving this ambiguity requires reasoning about the relative configuration and correlating the appearance of the part with the appearance of the individual.

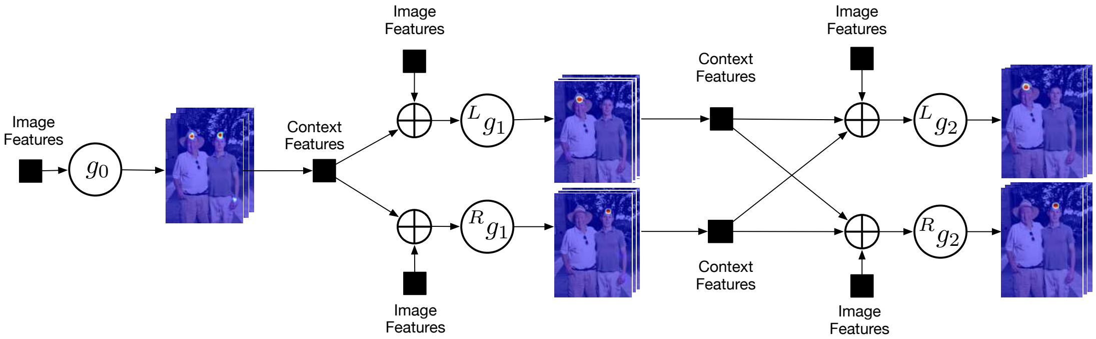


Figure 5.2: **Pose Machine for Parsing Visual Dyads**: Two parallel sequences of predictors are trained to estimate the pose of each participating object in the visual dyad. Each sequence specializes in predicting the object in the left and right parts of the image plane respectively.

A possible approach to dealing with the problem of interacting articulated objects is to reason about each object individually. Methods such as those proposed by [Ghiasi et al., 2014] and [Chen and Yuille, 2014b] approach the problem in an interaction-agnostic fashion. These methods reason about the pose of individuals by modeling local occlusions of body parts while remaining agnostic to the interacting individual. In work by [Andriluka and Sigal, 2012], interactions between individuals are modeled using a graphical model with additional connections between parts of the interacting objects. This results in a loopy graphical model, where inference is difficult. The authors use a branch and bound approach to perform inference, which can potentially be intractable for models incorporating complex interactions. [Yang et al., 2012] study the problem of interacting people, but restrict their scope to the problem of detecting types of interactions, by fitting tree structured models trained for each interaction type and scoring the detections using the fitting error. In this chapter we make the case for reasoning about interacting objects jointly and not in isolation. We describe a model for jointly estimating the articulated pose of both interacting objects in a visual dyad that leverages the success of the pose machine architecture for modelling complex spatial interactions via a sequential prediction procedure.

5.1 Model Overview

As in the previous chapter we view the problem of detecting the anatomical landmarks of interacting objects as a structured prediction problem. We model the pixel location of each anatomical landmark (which we refer to as a part) in the image, $y_p \in \mathcal{Z} \subset \mathbb{R}^2$, where $\mathcal{Z}$ is the set of all (u, v) locations in an image. Our goal is to predict the structured outputs $\mathbf{y}^l = (y_1^l, \ldots, y_p^l)$ for all P parts of the *left* object and *right* object where $l \in \{L, R\}$ refer to the left and right objects. Our basic assumption regarding the interacting objects is that they occur in a horizontal relative geometric configuration, thus allowing us to classify each object as belonging to the left or the right of the image plane. A dyadic pose machine consists of an initial part predictor $g_0(\cdot)$ that is trained to predict the location of the P anatomical landmark s regardless of object ownership from local image evidence and a sequence of predictor pairs $\{g_t^l(\cdot)\}_{l\in\{L,R\}}$ that are trained to predict the location of parts belonging to the left and right persons respectively from both local image evidence and contextual information from preceding classifiers. In each stage $t \in \{1 \ldots T\}$, the classifier predicts a confidence for each output variable assignment $y_p^l = z\, \forall z \in \mathcal{Z}, l \in \{L, R\}$ based on features of the image data $\mathbf{x}_z \in \mathbb{R}^d$ and contextual information from the preceeding classifiers of both objects in the neighborhood around each y_p. In each stage, the computed confidences provide an increasingly refined estimate for the variable. The confidence for the i^{th} part agnostic of object ownership is given by:

$$b_0(y_p = z) = g_0^p\left(\mathbf{x}_z\right), \tag{5.1}$$

where $\mathbf{x}_z$ refer to local image evidence around the location z. In subsequent stages the predictors $\{g_t^l(\cdot)\}_{l\in\{L,R\}}$ resolve part ownership based on local image evidence and contextual information from previous predictions:

$$^{l}g_t\left(\mathbf{x}_z^l, \bigoplus_{l\in\{L,R\}} \psi(z, {}^{l}\mathbf{b}_{t-1}), \psi(z, \mathbf{b}_0)\right) \rightarrow \left\{{}^{l}b_t^p(y_p = z)\right\}_{p\in 0\ldots P_l}. \tag{5.2}$$

For convenience, we denote the collection of confidence maps for all the parts belonging to level l as ${}^{l}\mathbf{b}_t \in \mathbb{R}^{w\times h\times P}$.

The predictors in subsequent stages use object-agnostic part beliefs ($\mathbf{b}_o$, local image evidence $\mathbf{x}_z^l$ and the object-specific part beliefs ${}^{l}\mathbf{b}_{t-1}$ from preceding stages to compute updated confidences for the object-specific part beliefs (see figure 5.3. As before the inference machine architecture allows the learning of complex spatial interactions between both parts of the same object as well as parts of the interacting object as each predictor is presented context from both objects. This allows the model to implicitly reason about the relative configurations of interacting objects. We see that in subsequent stages the predictors are able to resolve the part ownership starting from the object-agnostic detections produces by the predictor $g_0(\cdot)$. As in the previous chapter, the procedure models a fully connected graph where contextual information is shared between all parts of all objects. This allows the model to reason about inter-object occlusions as relationships between all parts are modeled. It also enables the model to learn to reason about complex joint configurations of the interacting objects.

We use deep convolutional networks as the prediction modules for the predictors for object-agnostic part detection (g_0) and the object-aware part predictors $\{{}^{l}g_t\}_{l\in\{L,R\}}$. As described in section 4.3.2, contextual information on the output of preceding stages is captured by designing a network with the appropriate receptive field.

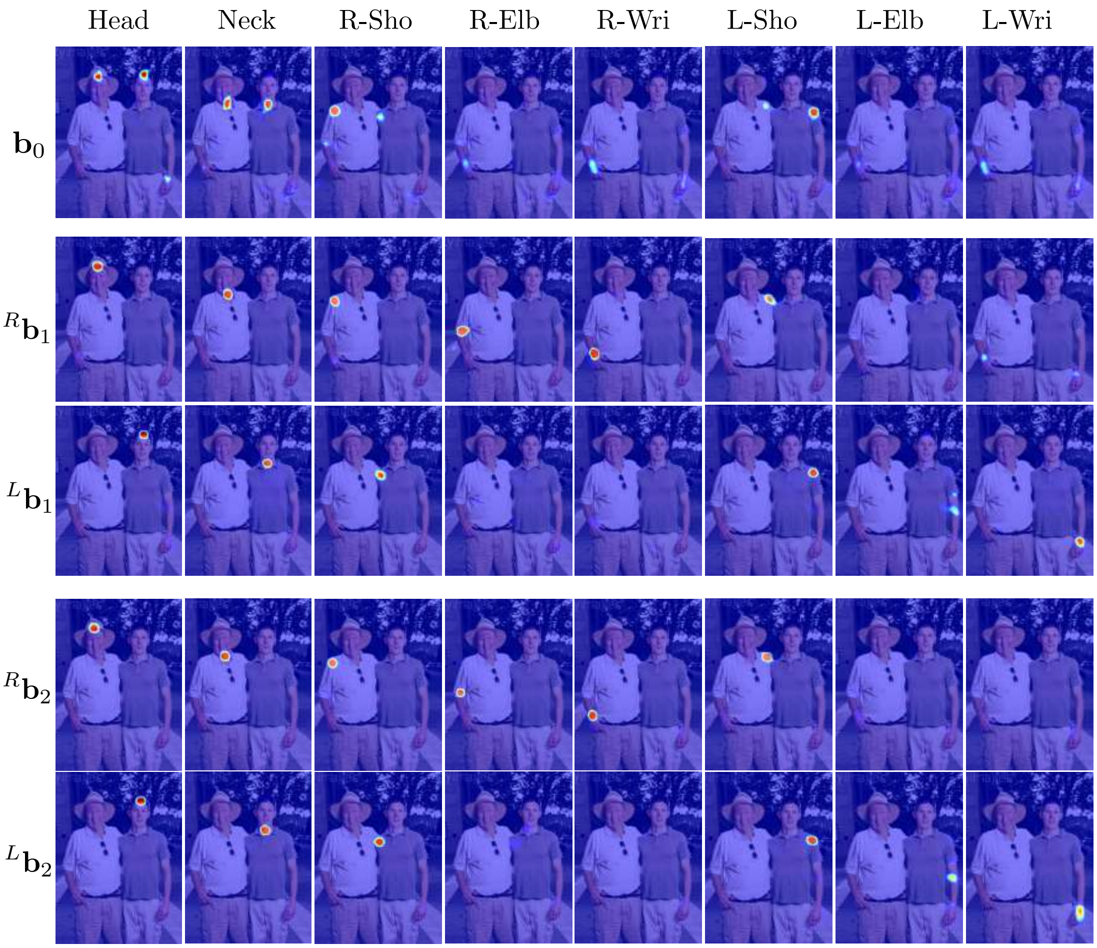


Figure 5.3: **Intermediate outputs of a dyadic pose machine:** In the first row we show the object-agnostic part detection confidences. We see multiple peaks for each part on both interacting people. The second and third row show outputs for the left and right object-aware part predictors. We see that the bimodality of the part detections is attenuated with the detection on the corresponding object being strengthened. The fourth and fifth row show an additional stage of the sequence.

5.2 Learning

Each stage of the dyadic pose machine is trained to repeatedly produce the confidence maps or *beliefs* for the locations of each of the parts for each of the interacting objects. Similar to the training procedure described in Section 4.6.2, we encourage the network to repeatedly arrive at such a representation by defining a loss function at the output of each stage. In the order-agnostic stage, we define a cost function that minimizes the euclidean distance between the order-agnostic part predictions, ${}^0b^p(z)$, and the ideal scoremaps, ${}^0b^p_*(z)$, for each location z in each training image:

$$f_0 = \sum_{p=1}^{P} \sum_{z \in \mathcal{Z}} \| {}^0b^p_0(z) - {}^0b^p_*(z) \|_2^2. \tag{5.3}$$

In subsequent stages t, for each object $l \in \{L, R\}$, we define a loss function that penalizes the l_2 distance between the predicted and ideal confidence maps for each part. The ideal confidence map for a part p is written as ${}^lb^p_*(Y_p = z)$. The cost function we aim to minimize at the output of each stage at for each object is therefore given by:

$$ {}^lf_t = \sum_{p=1}^{P} \sum_{z \in \mathcal{Z}} \| {}^lb^p_t(z) - {}^lb^p_*(z) \|_2^2. \tag{5.4}$$

The overall objective for the full architecture is obtained by adding the losses at each stage and summing over every image in the training dataset $\mathcal{D}$, and is given by,

$$\mathcal{F} = \sum_{\mathcal{D}} \left(f_0 + \sum_{t=1}^{T} \sum_{l=1}^{L} {}^lf_t \right). \tag{5.5}$$

As the full architecture is differentiable, the above loss function can be minimized using a first order method such as stochastic gradient descent. Gradients throught the entire architecture can be computed by the backpropagation algorithm [Rumelhart et al., 1988; LeCun et al., 1989]. The performance of the convolutional architecture improves with the data used for training. We perform data augmentation by rotating, flipping and cropping the image to generate a large training corpus.

5.3 Inference

At test time, *inference* comprises of performing a feedforward pass through the convolutional architecture of the dyadic pose machine to produce the beliefs at the final stage ${}^{l}\mathbf{b}_T$ for each object in the ordering $l \in \{L, R\}$. The location for anatomical landmarks y_p^l for each object in the ordering, $l \in \{L, R\}$, are then obtained by finding the location in each of the scoremaps that corresponds to the maximum score:

$$y_p^l = \arg\max_{z} \; {}^{l}\mathbf{b}_T^p(z). \tag{5.6}$$

In the dyadic pose machine, we model *part presence* during training. Parts that are not visible in training images do not contribute to peaks in the ideal training score maps. Therefore, at test-time, we only predict the presence of parts by assigning parts a location as in Equation 5.6 if the score corresponding to the location exceeds a visibility threshold τ_{vis}. The visibility threshold is calibrated using validation data, by finding the threshold value which provides the fewest misclassifications of part visibility. Therefore we have:

$$y_p^l = \begin{cases} \arg\max_{z} \; {}^{l}\mathbf{b}_T^p(z), & \text{if } \max_{z} \; {}^{l}\mathbf{b}_T^p(z) \geq \tau_{vis} \\ \text{not visible} & \text{if } \max_{z} \; {}^{l}\mathbf{b}_T^p(z) < \tau_{vis} \end{cases} \tag{5.7}$$

5.4 Results and Analysis

We analyze and evaluate our model for dyadic pose prediction on the *Proxemics* dataset introduced by [Yang et al., 2012]. The datasets consists of 578 images of people interacting, with two or three people per image. We use 300 images for

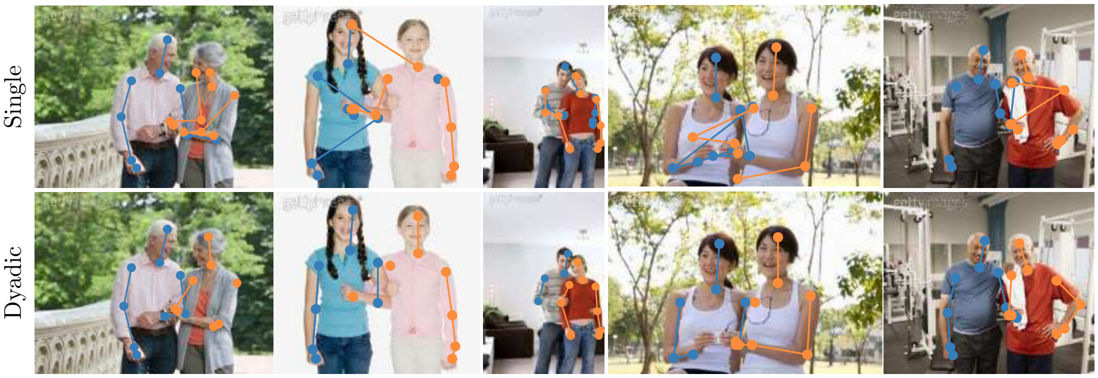


Figure 5.4: Comparison between dyadic pose prediction and single pose prediction. We see that the pose predictor operating on individuals is prone to incorrectly assigning part ownership and suffers when there is heavy inter-person occlusion. In contrast the dyadic pose machine is able to predict pose in the presence of a large degree of inter-person occlusion and complex interactions.

training and the rest for testing. We show some representative image samples in Figure 5.1. The dataset includes annotations for 10 keypoints on the upper body. We train a model as described in the previous section to predict the 10 keypoints for the *left* (L) person and the *right* (R) person. We evaluate accuracy using the percentage correct keypoints (PCK) metric introduced in [Yang and Ramanan, 2013]. The PCK metric computes the accuracy of a keypoint prediction as a function of a threshold distance from the ground truth keypoint.

5.4.1 Comparison with Monadic Prediction Baseline:

In Figure 5.5 we plot the PCK accuracy comparing our baseline model that consists of an architecture trained to predict a single person (monadic pose prediction) as described in Chapter 4, with the dyadic pose prediction model as described in the previous sections over all poses in the dataset. We find that the dyadic pose prediction model outperforms the monadic model by a large margin. We show qualitative comparisons between the individual pose predictions and the dyadic pose predictions in Figure 5.4.

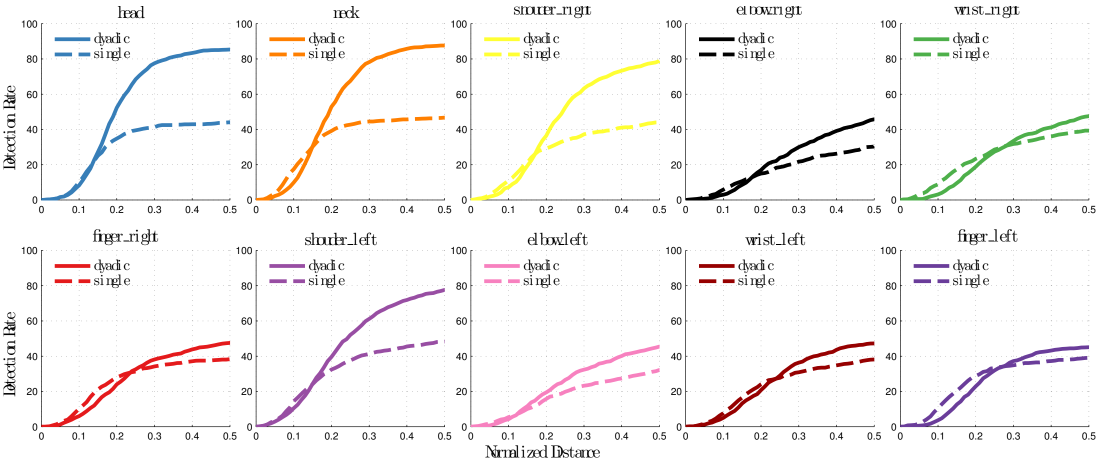


Figure 5.5: We compare the dyadic pose machine model with a baseline that consists of a convolutional pose machine as described in the previous chapter in sequential fashion, the pose is predicted for each person separately anchored on a crop of the image based on a face detection result.

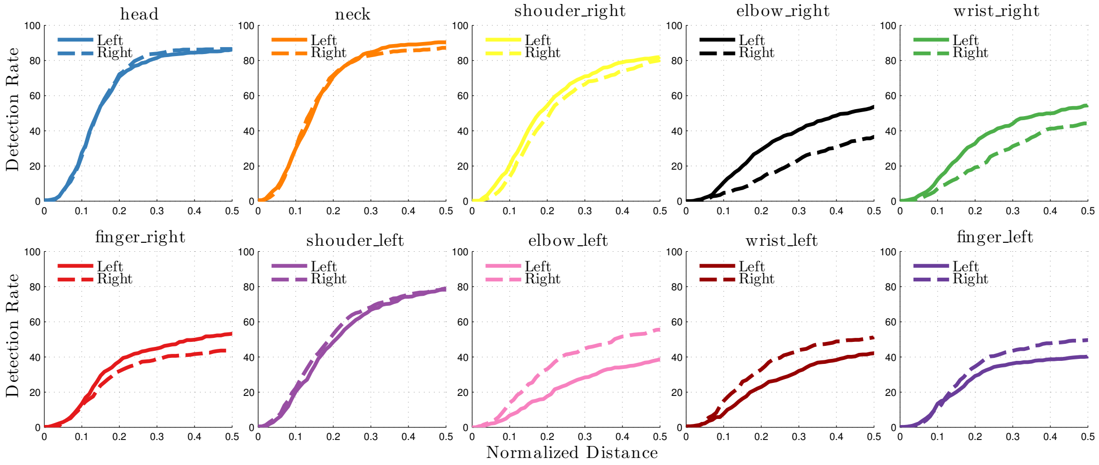


Figure 5.6: PCK evaluation comparing performance of the left person and right person predictors of a dyadic pose machine. Note the keypoints on the right arm of the left person predictor and left arm of the right person predictor. We see that these parts tend to occlude each other resulting in lower accuracy.

5.4.2 Comparison of Left/Right Predictions:

We compare the pose estimation accuracy for the left person with the accuracy for the right person in Figure 5.6. We note that they perform comparably, except for a drop in accuracy for keypoints on the left arm of the right person, and keypoints on the right arm of the left person. This is because these parts are highly prone to inter-person occlusions.

5.4.3 Accuracy based on Interaction Type

: We show the pose estimation accuracy using the PCK metric based on each type of interaction as coded in the Proxemics dataset [Yang et al., 2012] in Figure 5.8. The dataset is divided into six different types of interactions: *hand-hand*, *hand-shoulder*, *hand-torso*, *shoulder-shoulder*, *elbow-shoulder* and *hand-elbow*. We see that

5.5 Discussion

We developed a pose-machine architecture for reasoning about visual dyads. Dyadic pose prediction is challenging due to the large variation in the coupled configuration of the dyad. We find that when people occur in configurations that obey the ordering assumptions (left/right) we are able to parse each object's pose successfully, reasoning about inter-person occlusions and asssigning correct part ownership. The primary failure mode for the method is when ordering assumptions are violated in the image, and the relative ordering is not captured by a simple left/right designation. In Figure 5.9 we show common failure modes. We see that in images when the subjects have a top/down ordering, our method does not perform favorably. We also find that certain poses with intricate interactions and complex single configurations can also prove troublesome if similar examples haven't been seen previously during training. Developing a consistent and stable ordering method coupled with training with large datasets that cover the large variation in dyadic configurations is an avenue for future work.

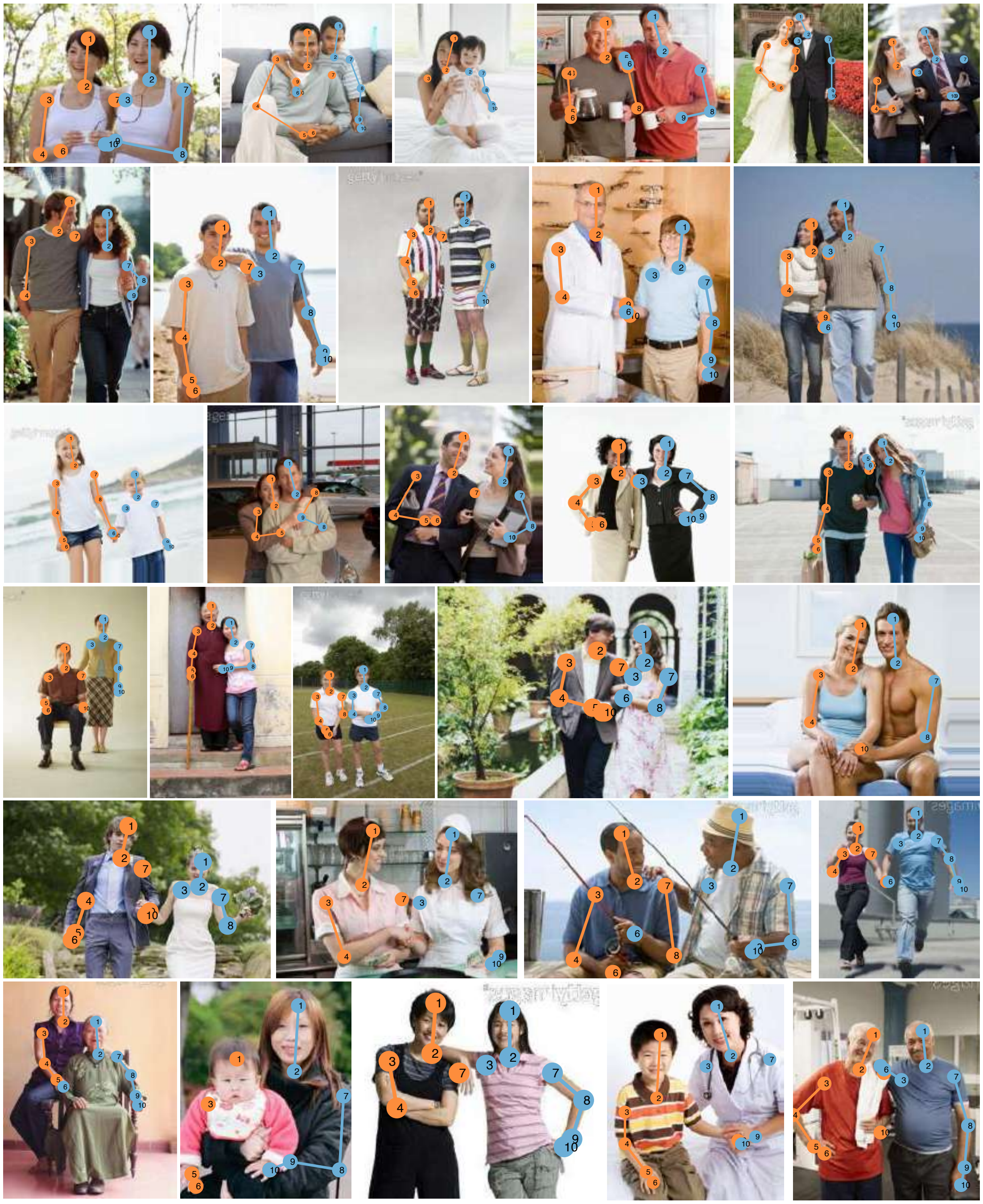


Figure 5.7: Qualitative examples of dyadic pose prediction on the PROXEMIC dataset[Yang et al., 2012]. Keypoints and limbs are only overlaid if the detection confidence is above a threshold for visiblility.

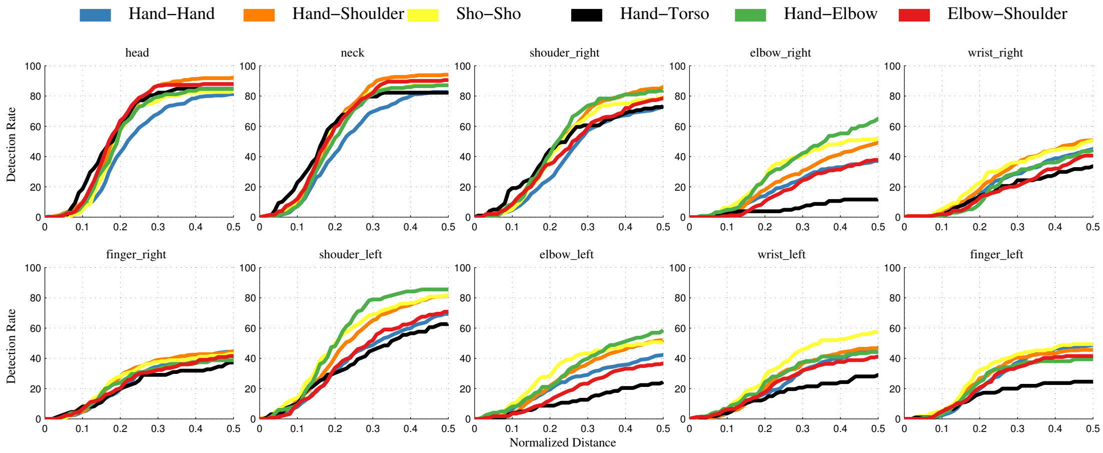


Figure 5.8: Comparison of PCK performance across types of interactions as described in Yang et al. [2012].

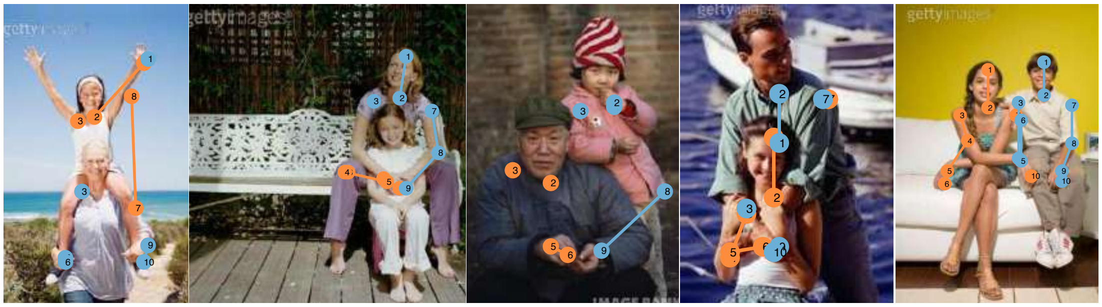


Figure 5.9: Failure examples. The primary failure mode is when the ordering assumption is violated. We see that in most failure cases the left-right ordering of subjects does not hold.

CHAPTER 6

Tracking Articulated Human Pose

As far back as [Gibson et al., 1969], researchers have noted the importance of having a representation for occlusion to reason about motion. Representing occlusion is particularly important in estimating human motion because, as the human body is an articulated structure, different parts occlude each other frequently. The human body is structurally symmetric and parts tend to be occluded by their symmetric counterparts, such as left knees by right knees. This occurs because the viewer's optical axis is often perpendicular to the body's bilateral plane of symmetry.

Spatial representations for reasoning about occlusion require evaluating a large set of possible spatial configurations [Sigal and Black, 2006], which scales combinatorially as we move from images to videos. Spatial representations also rely on weak cues; for example, the location and appearance of a shoulder provides only a weak cue as to whether the elbow is occluded. Temporal representations can make use of strong temporal continuity priors to reason about occlusions. It has been noted that even in the human visual system [Remus and Engel, 2003], temporal motion continuity serves occlusion reasoning. A part that is visible and has a smooth trajectory before and after a period of non-visibility must be

Algorithm 6 `sym_track`

```
Compute max-marginals $\mu^*(x_i^f)$ using Equation eq:maxmarg for root part in each frame.
Sample root part proposals $\mathcal{X}_p^f \sim \mu^*(x_i^f)$
Track root part by minimizing the objective in Equation 6.3.
while In breadth first fashion, select next part(s) do
    Compute max-marginals $(\mu^*(x_p^f))$ for current part(s) conditioned on the tracked locations of parent parts.
    if is_symmetric_pair then
        Sample part proposals $\mathcal{X}_p^f, \mathcal{X}_q^f$ from corresponding max marginals
        Track symmetric parts using formulation in Equation 6.6
    else
        Sample part proposals $\mathcal{X}_p^f \sim \mu^*(x_p^f)$
        Track part using formation in Equation 6.3
    end if
end while
```

occluded for that period. If a system cannot reason about occlusion temporally, motion consistency will force it to struggle to find image evidence to support a smooth path when occlusion occurs. This can corrupt tracking even outside the duration of occlusion.

In this chapter, we argue that temporal reasoning about occlusion is essential to tracking human pose and handling double counting. We divide the body into a set of singleton parts and pairs of symmetric parts. Our key insight is that tracking human pose can be cast as a multi-target tracking problem where the "targets" are related by an underlying articulated structure. Our contributions are: (1) an occlusion-aware model for tracking human pose that enforces both spatial and temporal consistency; (2) a method for jointly tracking symmetric parts that is inspired by optimal formulations for multi-target tracking.

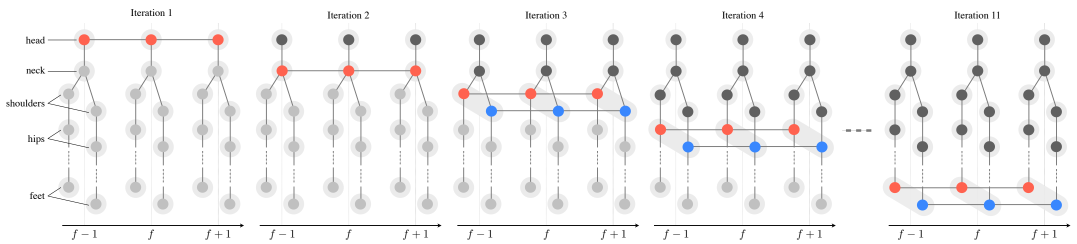


Figure 6.1: **Graphical representation of the algorithm**. We use a tree-structured deformable parts model in each frame to generate proposals for each part. In the first iteration, we track the head node using an LP tracking formulation. Proposals for the next symmetric pair in the tree are generated by conditioning each tree on the tracked locations computed in the previous iteration. Symmetric parts are tracked simultaneously with mutual exclusion constraints. The method proceeds by sequentially conditioning the tracking of parts on their parents until all the parts are tracked.

6.1 Tracking Human Pose

The (u, v) location of a part p in a frame at time instant f is denoted by x_p^f. We denote by $\mathbf{u}_p = [x_p^1 \;\ldots\; x_p^F]$, the locations of part p in frames 1 to F and by $\mathbf{u}$ the set of tracks for all parts $(1, \ldots, P)$. A symmetric part pair is a pair of parts (p, q) that share the same appearance. The goal of human pose tracking is to estimate the location of each part of the person in every frame of the image sequence. We write this as maximizing the following scoring function over the full model:

$$\mathbf{u}^* = \underset{\mathbf{u}}{\operatorname{argmax}}\; \mathbf{E}(\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_P). \tag{6.1}$$

Optimizing the above scoring function over the full model requires a search over an exponential number of configurations and is NP-hard in general.

To bypass the intractability of the objective, we proceed by approximating the function and making stage-wise locally optimal decisions (see Figure 6.1). We begin with a root node for which the false positive rate is the lowest [Yang and Ramanan, 2011]. For human pose, this root node is the head for which we are able to get reliable detections. Given a set of proposals for the location of the head in each frame (Section 6.3.1), we solve for the optimal track $\mathbf{u}_1^*$,

$$\mathbf{u}_1^* = \operatorname*{argmax}_{\mathbf{u}_1} \mathbf{E}(\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_P). \tag{6.2}$$

s

6.2 Tracking a Singleton Part

Given a set of proposals denoted by $\mathcal{X}_p^f$ for part p in the image at each frame f, we first augment the proposal sets with an occlusion state o_p^f for each frame. We form tracklets ${}^p t_{ijk}$ for each part by combining triplets $({}^i x_p^{f-1}, {}^j x_p^f, {}^k x_p^{f+1})$ where ${}^i x_p^f \in \mathcal{X}_p^f$ is a proposal at location i in the image or an occlusion state o_p^f.

We denote by ${}^p\mathbf{U}_{ijk}^f$ the indicator variable that is associated with tracklet ${}^p t_{ijk}$ that takes values $\in \{0, 1\}$ corresponding to the tracklet being selected or not. We associate with each tracklet, a score u_{ijk}^f based on appearance, detection, and foreground likelihood cues, which is described in Section 6.3.2. Our goal then is to maximize the following objective subject to constraints:

$$\begin{aligned}
\max_{\{{}^p\mathbf{U}\}} \quad & \sum_{\forall i,j,k,f} {}^p u_{ijk}^f \, {}^p\mathbf{U}_{ijk}^f \\
\text{s.t.} \quad & \{\mathbf{U}_{ijk}^f\} \in \{0,1\} \\
& \forall f, \forall (j,k) \;\; \sum_i {}^p\mathbf{U}_{ijk}^f = \sum_l {}^p\mathbf{U}_{jkl}^{f+1} \qquad (\textit{Continuity}) \\
& \forall f, \;\; \sum_{i,j,k} {}^p\mathbf{U}_{ijk}^f = 1 \qquad\qquad\quad\;\; (\textit{Uniqueness})
\end{aligned} \tag{6.3}$$

The above optimization problem corresonds to finding the single best path in a lattice graph and can be solved efficiently using dynamic programming.

Continuity Constraints enforce conservation of flow by stating that the flow entering the nodes j and k should be equal to the flow emanating from those nodes. These constraints essentially encode the connectivity of a track, preventing fragmented tracks.

Uniqueness Constraints limit the flow at each time instant to be 1. This implies that one object is being tracked in the network graph.

6.2.1 Conditioned Tracking

Once the optimal track $\mathbf{u}_1^*$ has been obtained (Section 6.2), we generate proposals and track the next set of nodes conditioned on the optimal parent track $\mathbf{u}_1^*$.

$$(\mathbf{u}_2^*) = \underset{\mathbf{u}_2}{\operatorname{argmax}} \; \mathbf{E}(\mathbf{u}_1 = \mathbf{u}_1^*, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4 \ldots \mathbf{u}_P). \tag{6.4}$$

We use the same formulation as in Section 6.2 to obtain the optimal track $\mathbf{u}_2^*$.

Next, for a symmetric pair of parts whose tracks are given by $(\mathbf{u}_3, \mathbf{u}_4)$ we simultaneously estimate the optimal tracks (See Section 6.3):

$$(\mathbf{u}_3^*, \mathbf{u}_4^*) = \underset{\mathbf{u}_3, \mathbf{u}_4}{\operatorname{argmax}} \; \mathbf{E}(\mathbf{u}_1 = \mathbf{u}_1^*, \mathbf{u}_2 = \mathbf{u}_2^*, \mathbf{u}_3, \ldots \mathbf{u}_P). \tag{6.5}$$

Tracking is conditioned on the optimal parent track by fixing the location of the parent in each of the frames to the tracked locations and re-running dynamic programming inference in each of the trees in each frame (Section 6.3.1).

We proceed in this manner, by conditioning the tracking of the child nodes on the optimal tracks of their parents and by tracking symmetric parts using a joint formulation, until all the parts have been tracked.

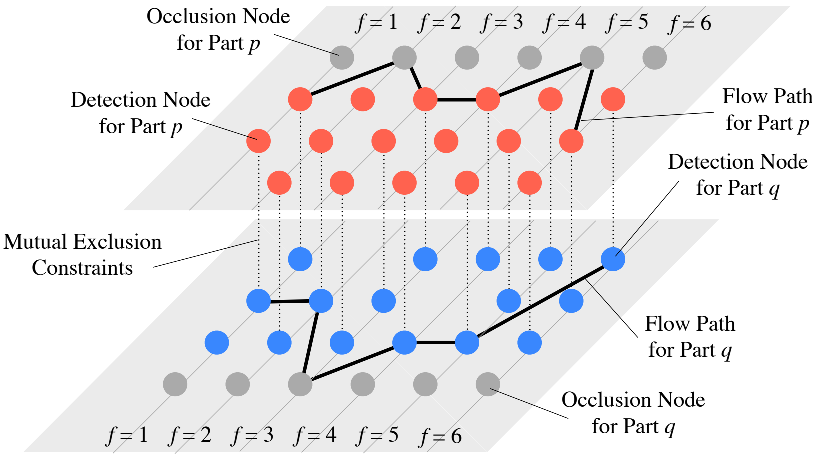


Figure 6.2: Max-flow formulation for symmetric part tracking. The blue and red dots denote detections for each of the parts separately in each frame. The gray nodes denote occlusion nodes for each frame. The dotted lines depict mutual exclusion constraints between certain sets of nodes. The symmetric tracking problem is to find the best scoring path in each of these graphs subject to the mutual-exclusion constraints.

6.3 Tracking a Pair of Symmetric Parts

Our approach treats the problem of tracking symmetric pairs of parts as a multi-target tracking problem. In multi-target tracking, the goal is to track multiple objects that share the same appearance and hence the same generic detector (typically pedestrians). The objects move in the scene in an unconstrained fashion with mutual occlusions. Recent methods have modeled multi-target tracking as a network flow problem [Andriyenko and Schindler, 2010; Jiang et al., 2007; Berclaz et al., 2011] where finding tracks is equivalent to pushing K-units of flow through a graph where K is the number of objects to be tracked.

Our formulation is as follows: we denote by ${}^{p}\mathbf{U}$ and ${}^{q}\mathbf{U}$ the set of all indicator variables for tracklets p and q respectively. Our full objective is now the following optimization problem:

$$\begin{array}{lll}
\max\limits_{\{{}^{p}\mathbf{U},{}^{q}\mathbf{U}\}} & \sum\limits_{i,j,k,f} {}^{p}u^{f}_{ijk}\,{}^{p}\mathbf{U}^{f}_{ijk} + \sum\limits_{i,j,k,f} {}^{q}u^{f}_{ijk}\,{}^{q}\mathbf{U}^{f}_{ijk} & \\
\text{s.t.} & \{\mathbf{U}^{f}_{ijk}\} \in \{0,1\} & \\
 & \forall f,\;\; \sum\limits_{i} {}^{p}\mathbf{U}^{f}_{ijk} = \sum\limits_{l} {}^{p}\mathbf{U}^{f+1}_{jkl} & (\mathit{Continuity}) \\
 & \forall f,\;\; \sum\limits_{i} {}^{q}\mathbf{U}^{f}_{ijk} = \sum\limits_{l} {}^{q}\mathbf{U}^{f+1}_{jkl} & \\
 & \sum\limits_{i,k} {}^{p}\mathbf{U}^{f}_{ijk} + \sum\limits_{i,k} {}^{q}\mathbf{U}^{f}_{ijk} \leq 1 & (\mathit{Mutual\ Exclusion}) \\
 & \forall f,\;\; \sum\limits_{i,j,k} {}^{p}\mathbf{U}^{f}_{ijk} = 1 & (\mathit{Uniqueness}) \\
 & \forall f,\;\; \sum\limits_{i,j,k} {}^{q}\mathbf{U}^{f}_{ijk} = 1 &
\end{array} \tag{6.6}$$

Mutual Exclusion Constraints. We enforce mutual exclusion constraints that prevent the symmetric parts from occupying the same location in the image. In a typical self-occlusion scenario the score of a particular location in the image will be high for both the symmetric parts. In such a case the mutual-exclusion constraints enforce that only one part can occupy the location, while the symmetric counterpart is either pushed to an occlusion node or to another location in the image that is consistent with the constraints and has a high score. We enforce these constraints by limiting the total flow at nodes in both networks that share the same location in the image.

This formulation corresponds to maximizing the flow through two separate networks that interact via the mutual exclusion constraints. The above optimization problem is an integer linear program and solving it is NP-complete. However, we can relax the problem by replacing the integral constraints by allowing $0 \leq {}^{p}\mathbf{U}^{f}_{ijk} \leq 1$ and $0 \leq {}^{q}\mathbf{U}^{f}_{ijk} \leq 1$. The relaxation can be shown to be tight for most practical cases [Andriyenko and Schindler, 2010].

We solve this linear program using a commercially available solver [MOSEK]. In the case of non-integral solutions, we use a branch and cut method to find the integral optimum as suggested in [Andriyenko and Schindler, 2010].

Occlusion Interpolation Once a solution is obtained, the location of the occluded part is estimated by interpolating between the image location of the node preceding and following occlusion using cubic B-spline interpolation.

6.3.1 Generating Part Proposals via Max-Marginals

Human pose in a frame at each time instant is modeled with a tree-strutured deformable part model as in recent work by [Yang and Ramanan, 2011]. A deformable part model is a tree-structured CRF that maximizes the following score, given an image:

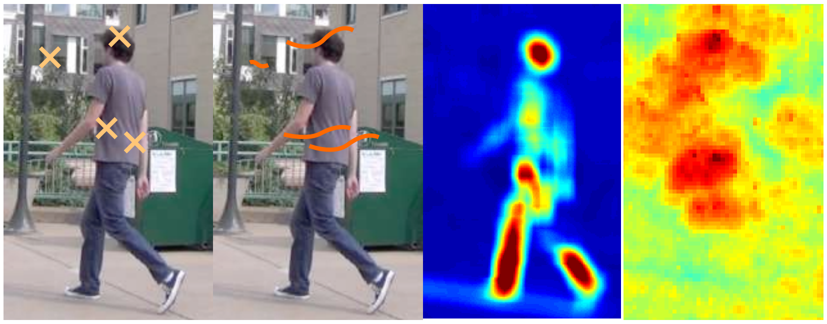
$$\mathbf{S}(\mathbf{x}_f) = \sum_{i=1} w_i^F \phi(I_t, x_i^f) + \sum_{i,j} w_{ij} \psi(x_i^f, x_j^f) \tag{6.7}$$

where $\mathbf{x}_t = [x_1^f \dots x_P^f]$ is the pose in frame f, $\phi(I_f, x_i^f)$ are a set of image features computed at location x_i^f, $\psi(x_i^f, x_j^f)$ is a quadratic function that measures the displacement between parts i and j. The weights w_i and w_{ij} are the parameters of the CRF that are learned as described in [Yang and Ramanan, 2011].

To generate proposals for part locations in each frame, we compute the max-marginal of the above scoring function at each part. The max-marginal for part i in frame f is given by:

$$\mu^*(x_i^t = s) = \max_{x^t : x_i^t = s} \mathbf{S}(\mathbf{x}_t), \tag{6.8}$$

which is the maximum of the scoring function with the part i clamped to location s. The max-marginal provides a peaky approximation of the true marginal distribution. We compute max-marginals for each tree in each frame separately. The max-marginals for a tree-structured graphical model can be computed efficiently



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

Figure 6.3: **Scoring Tracklets**. (a) Proposals for the head are generated from the max-marginal score map shown in (d). (b) Proposal sets are augmented by tracking each proposal forwards and backwards to ensure smooth tracks. (c) Foreground likelihood used to score tracklets (d) The detection likelihood for the head part.

for all the parts by performing two passes of max-sum message passing inference. We perform non-maxima suppression on the max-marginal score map for each part to generate a set of location proposals in each frame.

We expand the proposal set by tracking each proposal forwards and backwards using a Lucas-Kanade template tracker [Baker and Matthews, 2004] to obtain extended proposal sets $\mathcal{X}_i^t$. This ensures smoother tracks and makes the proposal generation robust to frame-to-frame inconsistencies of the detector.

Once a parent part has been tracked, the max-marginals for the child nodes are recomputed by conditioning on the tracked locations of the parent nodes. The conditioned max-marginals for part i in frame f with a set of parent nodes $pa(i)$ with tracked locations $\mathbf{x}^*_{pa(i)}$ can be written as:

$$\mu^*(x_i^f = s) = \max_{\substack{x^f : x_i^f = s, \\ \forall j \in pa(i),\ x_j^f = x_j^{f*}}} \mathbf{S}(\mathbf{x}_f). \tag{6.9}$$

This can be efficiently computed for a tree, as before, by performing dynamic programming max-sum inference.

6.3.2 Scoring Part Tracklets

Each tracklet is assigned a likelihood score that consists of terms that measure the detection likelihood, the foreground likelihood and motion prior:

$$\begin{aligned} s_{ijk}^{f} = \; & \alpha_s \mathbf{s}_{\text{fore}}(\mathbf{U}_{ijk}^{f}) + \; \alpha_f \mathbf{s}_{\text{det}}(\mathbf{U}_{ijk}^{f}) \\ & + \alpha_m \mathbf{s}_{\text{mot}}(\mathbf{U}_{ijk}^{f}). \end{aligned} \tag{6.10}$$

The weighting co-efficients of the different terms were set by performing a grid search on validation data.

Detection Likelihood. The likelihood of detection for a particular part is obtained by using the max-marginal score of the tree-structured CRF model. We normalize the max-marginal score and obtain a likelihood of detection of part p at location i as:

$$\mathbf{l}_{det}({}^{i}x_{p}^{f}) \propto \frac{\exp(-\mu^{*}(x_{p}^{f} = i))}{\sum_{s=1}^{L} \exp(-\mu^{*}(x_{p}^{f} = s))}. \tag{6.11}$$

For a tracklet with occlusion nodes we assign a constant score for the occlusion nod $\mathbf{l}_{det}({}^{i}o_{p}^{f}) \propto p_{det}^{o}$. This constant needs to be calibrated in relation to the scores of the detector and is found by performing a grid search on validation data. The detection score for the tracklet $\mathbf{U}_{ijk}^{f}$ is obtained as:

$$\mathbf{s}_{det}(\mathbf{U}_{ijk}^{f}) = \mathbf{l}_{\text{det}}({}^{i}x_{p}^{f-1}) \cdot \mathbf{l}_{\text{det}}({}^{j}x_{p}^{f}) \cdot \mathbf{l}_{\text{det}}({}^{k}x_{p}^{f+1}). \tag{6.12}$$

Motion Likelihood. We use a constant velocity motion model. In order to check for constant velocity, we require two motion vectors, and therefore we use three consecutive sites in our formulation (similar to [Andriyenko and Schindler, 2010]). We denote the two motion vectors as $\mathbf{v}_{ij} = x_{i}^{f-1} - x_{j}^{f}$ and $\mathbf{v}_{jk} = x_{j}^{f} - x_{k}^{f+1}$. Our motion score is now given by:

$$\mathbf{s}_{mot}(\mathbf{U}^f_{ijk}) = e^{-\left(\frac{\|\mathbf{v}_{ij} - \mathbf{v}_{jk}\|}{\sigma_m}\right)^2}. \tag{6.13}$$

The constant velocity model allows us to enforce smoother trajectories and penalize large deviations.

Foreground Likelihood. The foreground likelihood is estimated by computing a background model by median filtering the image sequence. The foreground likelihood is estimated as:

$$\begin{aligned}\mathbf{s}_{mot}(\mathbf{U}^f_{ijk}) = (1 - p_b(x^{f-1}_i)) \cdot (1 - p_b(x^f_j)) \\ \cdot(1 - p_b(x^{f+1}_k))\end{aligned}$$

where $p_b(x^f_j)$ denotes the probability of the location x^f_j of belonging to the background, as given by:

$$p_b(x) = \frac{1}{\sqrt{2\pi}\sigma_b} e^{-\left(\frac{\|I(x) - I_b(x)\|}{\sigma_b}\right)^2} \tag{6.14}$$

where I_b is the computed background model. As before, we assign a constant score to occlusion nodes.

6.4 Experimental Analysis

We perform qualitative and quantitative experiments on two challenging datasets to determine the performance of the proposed algorithm. In order to test the tracking method we model human pose with the tree-structured CRF model of [Yang and Ramanan, 2011]. For all experiments, we train the model on the PARSE dataset introduced in [Ramanan et al., 2007]. We model the human body with 26 parts as in [Yang and Ramanan, 2011]: 2 singleton parts for the head and neck and a total of 12 symmetric pairs of parts for the shoulders, torso, legs, and upper arms.

Comparisons. As our baseline, we compare the method of [Park and Ramanan, 2011] that also uses a detector for pose in each frame [Yang and Ramanan, 2011] that is trained on the same training data. The n-Best pose configurations are generated for each frame and tracking is performed by modeling pose tracking with a chain-CRF and performing viterbi-decoding like inference.

6.4.1 Datasets.

We test our method on a variety of challenging datasets consisting of both indoor and outdoor sequences.

Human Eva-I: We evaluate our method on a standardized dataset that comprises of sequences of actors performing different actions in a indoor motion capture environment. We report results on the 250 frames each of the sequences *S1_Walking, S1_Jog, S2_Jog* for camera 1. We show qualitative results in Figure 6.7.

Outdoor Pose Dataset: This dataset consists of 6 sequences collected by us comprising of 4 different actors performing varied actions outdoors with a natural cluttered background. The actors perform complex actions and switch between actions within the same video. The poses they assume include many with significant self-occlusion.

Sequences from [Park and Ramanan, 2011]: We also test our method on the *walkstraight* and *baseball* sequences used in [Park and Ramanan, 2011] for evaluation and report PCP scores on these videos. We show qualitative results in Figure 6.4.

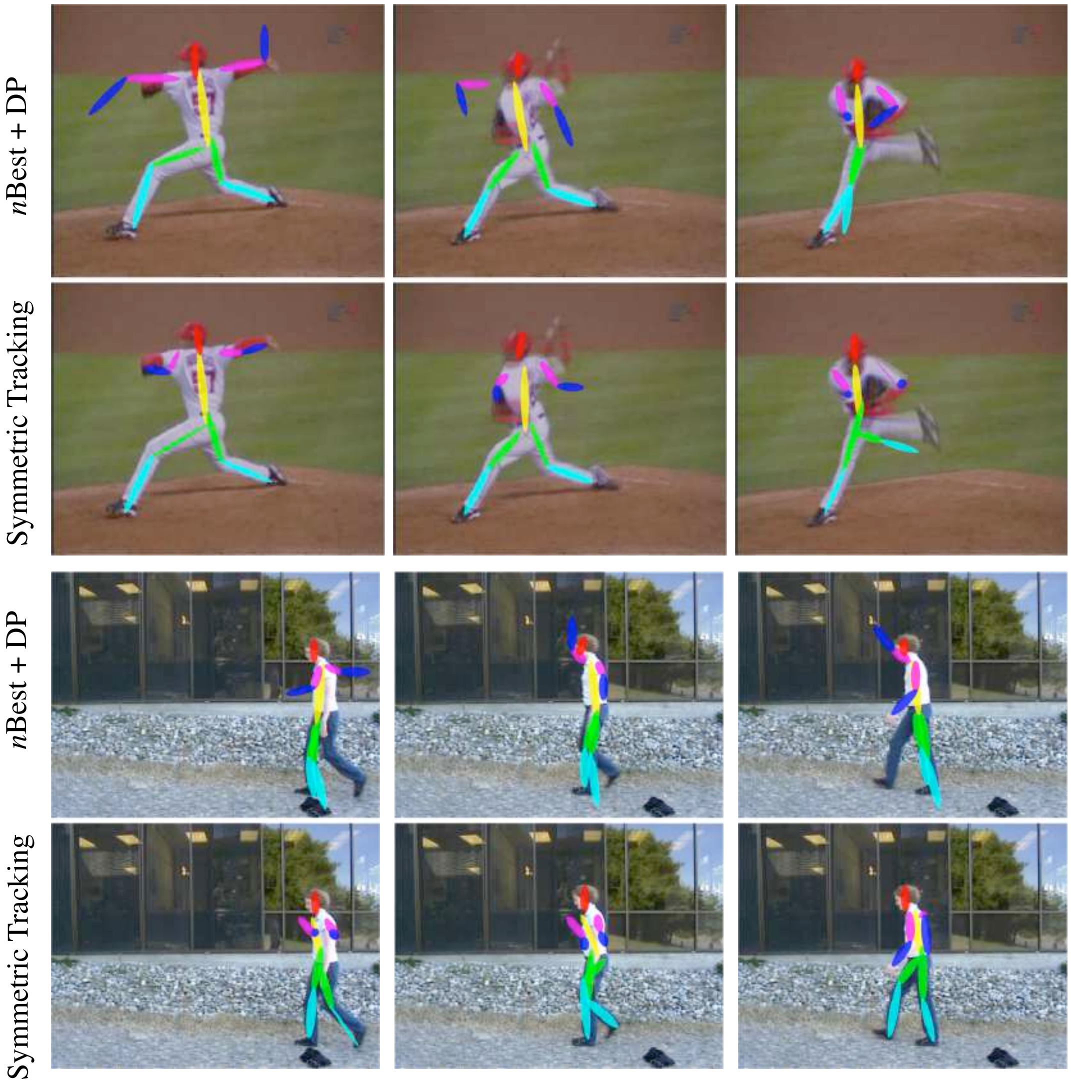


Figure 6.4: **Qualitative Comparison**. We show improvement frames on two of the sequences used in [Park and Ramanan, 2011].

6.4.2 Detection Accuracy

We use two metrics to evaluate our algorithm. We use the *PCP* criterion as in [Ferrari et al., 2008] and *keypoint localization error* (KLE). Keypoint localization

Metric	Method	Head	Torso	U.L.	L.L.	U.A.	L.A.
PCP	Ours	1.00	0.69	0.91	0.89	0.85	0.42
	[Park and Ramanan, 2011]	1.00	0.61	0.86	0.84	0.66	0.41
KLE	Ours	0.53	0.88	0.67	1.01	1.70	2.68
	[Park and Ramanan, 2011]	0.54	0.74	0.80	1.39	2.39	4.08

Table 6.1: PCP scores and keypoint localization error for the *baseball* and *walking* videos. We outperform the baseline due to better temporal consistency and occlusion handling.

error measures the average euclidean distance from the ground truth keypoint normalized scaled by the size of the head in each frame to correct for scale changes. As our method (and most 2D pose estimation methods) cannot distinguish between left and right limbs we report the score of the higher scoring assignment. We obtain significantly better results than our baseline [Park and Ramanan, 2011] on the outdoor pose dataset as reported in Table 6.2. The main improvements are in the tracking of the lower limbs which are especially susceptible to double counting errors. Our method reduces the double counting artifacts and enforces temporal smoothness for each part resulting in smoother and more accurate tracks. We also show improvments on the sequences used in [Park and Ramanan, 2011], PCP and KLE accuracies are reported in Table 6.1.

Metric	Method	Head	Torso	U.L.	L.L.	U.A.	L.A.
PCP	Ours	0.99	0.86	0.95	0.96	0.86	0.52
	[Park and Ramanan, 2011]	0.99	0.83	0.92	0.86	0.79	0.52
KLE	Ours	0.39	0.58	0.48	0.48	0.88	1.42
	[Park and Ramanan, 2011]	0.44	0.58	0.55	0.69	1.03	1.65

Table 6.2: PCP scores and keypoint localization error for the six sequences of the outdoor pose dataset. We obtain a significant improvement over the baseline due to better temporal consistency and occlusion handling.

Metric	Method	Head	Torso	U.L.	L.L.	U.A.	L.A.
PCP	Ours	0.99	1.00	0.99	0.98	0.99	0.53
	[Park and Ramanan, 2011]	0.97	0.97	0.97	0.90	0.83	0.48
KLE	Ours	0.27	0.48	0.13	0.22	1.14	1.07
	[Park and Ramanan, 2011]	0.23	0.52	0.24	0.35	1.10	1.18

Table 6.3: **HumanEvaI evaluation.** PCP scores and keypoint localization error for sequences from the HumanEva-I dataset. We obtain significant improvement over the baseline due to better temporal consistency and occlusion handling. We particularly perform well on the lower and upper legs which typically are difficult because of mutual occlusions.

6.4.3 Double counting errors

We observe a significant decrease in the number of double counting errors of our method over the baseline (Figure 6.6). In the outdoor pose dataset we reduce the number of double counting errors by substantially by around 75 %, while we observe a decrease of approximately 41 % on the HumanEva-I sequences.

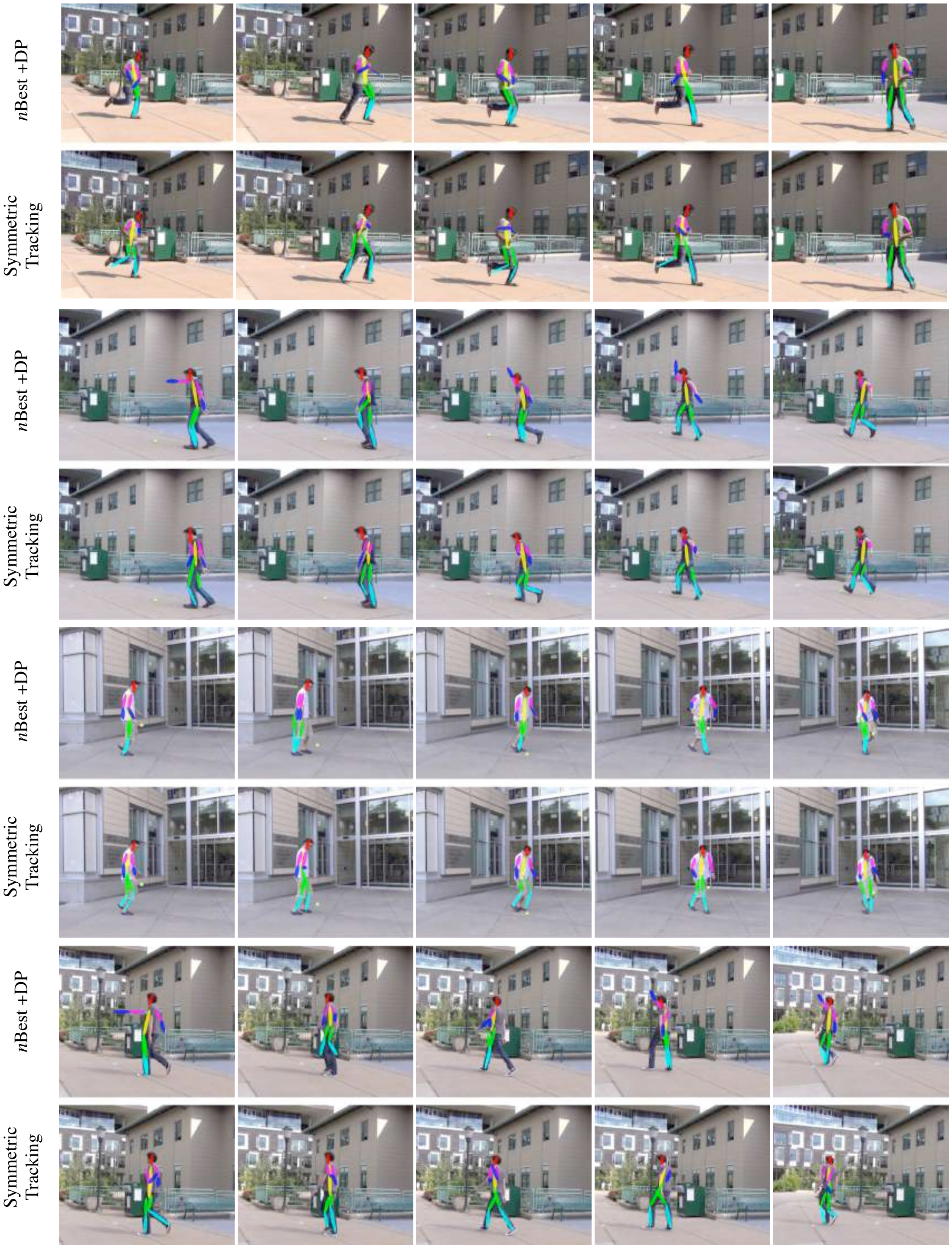


Figure 6.5: **Qualitative Comparison**. We show frames of symmetric tracking of human pose in comparison to the baseline [Park and Ramanan, 2011] on outdoor pose dataset. Note that our method reduces double counting errors especially on frames when the person is entering a profile view with mutual occlusion.

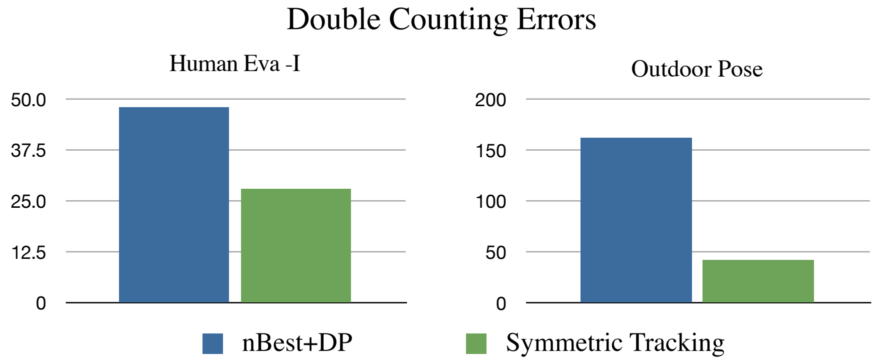


Figure 6.6: **Reduction in double counting**. We achieve a reduction in double counting errors on both our evaluation datasets due to better occlusion reasoning and mutual exclusion constraints.

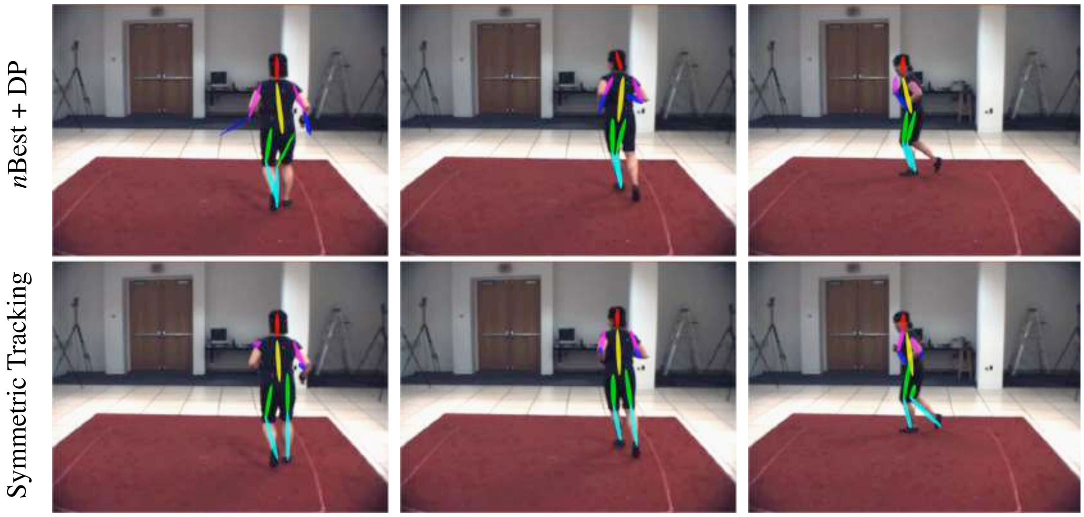


Figure 6.7: **Qualitative Comparison**. We show improvement frames on a sequence from the HumanEva-I dataset. We reduce double counting errors by reasoning about occlusion and enforcing mutual exclusion constraints.

CHAPTER 7

Reconstructing Articulated Human Pose

Geometrically, the problem of estimating the 3D configuration of points from their 2D projections is ill-posed, even when fitting a known 3D skeleton[1]. With human observers, the ambiguity is likely resolved by leveraging vast memories of likely 3D configurations of humans [Peelen and Downing, 2007]. A reasonable proxy for such experience is available in the form of motion capture libraries [MoCap], which contain millions of 3D configurations. The computational challenge is to tractably generalize from the configurations spanned in the corpus, ensuring anthropometric plausibility while discouraging impossible configurations.

Kinematic representations of human pose are high-dimensional and difficult to estimate directly. Allowing only statistically plausible configurations leads to compact representations that can be estimated from data. Linear dimensionality reduction (such as PCA) is attractive as it yields tractable and optimal estimation methods. It has been successfully applied to constrained deformable objects, such as faces [Matthews and Baker, 2003] and action-specific body reconstruction,

[1] As noted in [Lee and Chen, 1985], each 2D end-point of a limb subtends a ray in 3D space. A sphere of radius equal to the length of the limb centered at any location on one of these rays intersects the other ray at two points (in general) producing a tuple of possible 3D limb configurations for each location on the ray.

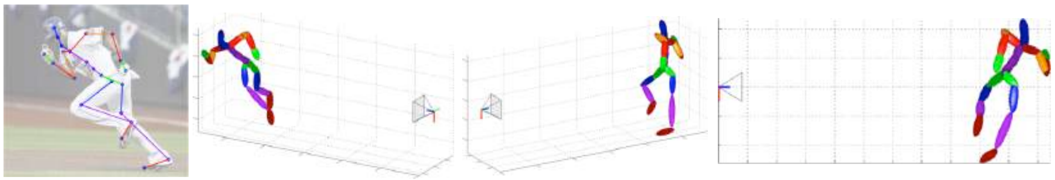
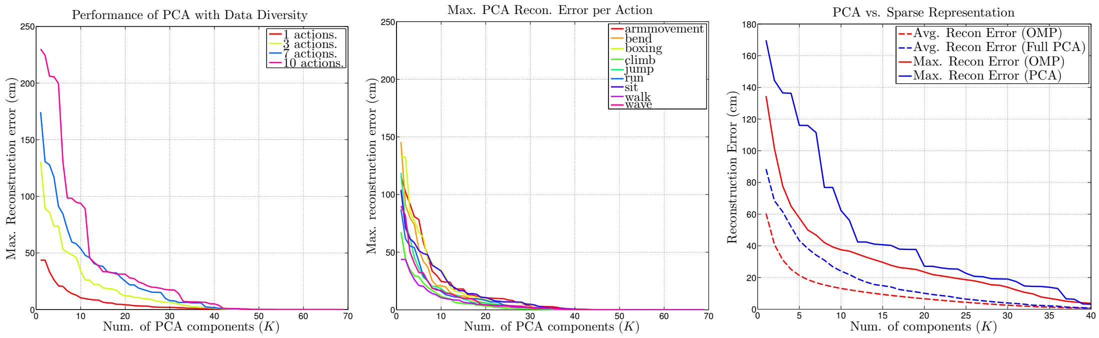


Figure 7.1: Given the 2D location of anatomical landmarks on an image, we estimate the 3D configuration of the human as well as the relative pose of the camera.

such as walking [Safonova et al., 2004]. However, as we add poses from varied actions, the complexity of the distribution of poses increases and, consequently, the dimensionality of the reduced model needs to be increased (see Figure 7.2). If we expand the dimensionality, linear models increasingly allow configurations that violate anthropometric constraints such as limb proportions, yet yield a projection in 2D that is plausible. The goal is therefore to develop an activity-independent model while ensuring anthropometric regularity.

We present a method to reconstruct 3D human pose while maintaining compaction, anthropometric regularity, and tractability. To achieve compaction, we separate camera pose variability from the intrinsic deformability of the human body (because combining both leads to an approximately six-fold increase in the number of parameters [Xiao et al., 2004]). To compactly model the intrinsic deformability across multiple actions, we use a sparse linear representation in an overcomplete dictionary. We estimate the parameters of this sparse linear representation with a matching pursuit algorithm. Enforcing anthropometric regularity through strict limb length constraints is intractable because satisfying multiple quadratic equality constraints on a least squares system is nonconvex [Boyd and Vandenberghe, 2004]. Instead, we encourage anthropometric regularity by enforcing a necessary condition (i.e., an equality constraint on the sum of squared lengths) as a constraint that is applied in closed form [Gander, 1981]. We solve for the model coefficients and camera pose within the matching pursuit iterations, decreasing the reprojection error objective in each iteration.



(a) (b) (c)

Figure 7.2: **Data Complexity**. (a) As more actions and, consequently, diverse poses are added to the training corpus, the maximum reconstruction error incurred by a linear dimensionality reduction model increases. (b) Maximum reconstruction error for each action separately using PCA. Each action can be compactly modeled with a linear basis. (c) Using a sparse representation in an overcomplete dictionary estimated using Orthogonal Matching Pursuit (OMP) achieves lower reconstruction error for 3D pose.

Our core contributions are: (1) a new activity-independent representation of 3D human pose variability as a sparse embedding in an overcomplete dictionary, and (2) an algorithm, Projected Matching Pursuit, to estimate the sparse model from only 2D projections while encouraging anthropometric regularity. Within the matching pursuit iterations, we explicitly estimate both the 3D camera pose and the 3D body configuration. We evaluate our method to test generalization, and robustness to noise and missing landmarks. We compare against a standard linear dimensionality reduction baseline and a nearest neighbor baseline.

7.1 Sparse Representation of 3D Human Pose

A 3D configuration of P points can be represented by $\mathbf{X} = \left(\mathbf{X}_1^T, \ldots, \mathbf{X}_P^T\right)^T \in \mathbb{R}^{3P\times 1}$ of stacked 3D coordinates. Under weak perspective projection, the 2D coordinates of the points in the image are given by

$$\mathbf{x} = \left(\mathbf{I}_{P\times P} \otimes \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{R} \right) \mathbf{X} + \mathbf{t} \otimes \mathbf{1}_{P\times 1}, \tag{7.1}$$

where $\mathbf{x} \in \mathbb{R}^{2P\times 1}$, $\otimes$ denotes the Kronecker product, $\mathbf{s} \in \mathbb{R}^{2\times 2}$ is a diagonal scale matrix with s_x and s_y being the scales in the x and y directions , $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^{2\times 1}$ denote the rotation and translation parameters of the weak perspective camera that we collectively denote as $\mathcal{C}$. We assume the camera intrinsic parameters are known. Estimating $\mathbf{X}$ and $\mathcal{C}$ from only the image evidence $\mathbf{x}$ is, fundamentally, an ill-posed problem. We see from Equation 7.1 we have $3P + 7$ parameters that we need to estimate from only $2P$ equations.

If the points form a semantic group that deform in a structured way, such as anatomical landmarks on a human body, we can reduce the number of parameters that need to be estimated using dimensionality reduction methods that learn the correlations between the points [Cootes et al., 2001]. Linear dimensionality reduction methods (e.g., Principal Component Analysis (PCA)) can be used to represent the points as a linear combination of a small number of basis poses,

$$\mathbf{X} = \boldsymbol{\mu} + \sum_{i=1}^{K} \mathbf{b}_i \omega_i, \tag{7.2}$$

where K is the number of basis poses, $\mathbf{b}_i$ are the basis poses, ω_i are the coefficients, and $\boldsymbol{\mu} \in \mathbb{R}^{3P\times 1}$ is the mean pose computed from training data. Under this model, we now have to estimate only $K + 7$ parameters instead of the original $3P + 7$ parameters.

A direct application of PCA to all the poses contained in the corpus[2] raises difficulties as shown in Figure 7.2a. For a single action, PCA performs well.

[2] We use the Carnegie Mellon Motion Capture Database [MoCap] to obtain a large corpus of 3D human poses.

As the diversity in actions in the data increases, the number of PCA components required for accurate reconstruction increases, and the assumption of a low dimensional linear subspace becomes strained. In particular, the *maximum* reconstruction error increases as the diversity in the data is increased because PCA inherits the occurrence statistics of poses in the corpus and not just the extent of variability.

7.1.1 Sparse Representation in an Overcomplete Dictionary

In Figure 7.2b we see that each individual action is compactly representable by a linear basis. Therefore, an arbitrary pose can be compactly represented by some subset of the set of all bases,

$$\begin{aligned} \mathbf{X} = \quad & \boldsymbol{\mu} + \textstyle\sum_{i=1}^{K} \mathbf{b}_i \omega_i, \\ & \{\mathbf{b}_i\}_{i \in I_{\mathbf{B}^*}} \in \mathbf{B}^* \subset \mathcal{B}, \end{aligned} \tag{7.3}$$

where $\boldsymbol{\mu}$ is the mean pose, $\mathcal{B} \in \mathbb{R}^{3P \times (\sum_{i=1}^{N_a} N_b^i)}$ is an overcomplete dictionary of basis components created by concatenating N_b^i bases computed from N_a different actions, $\mathbf{B}^*$ is an optimal subset of $\mathcal{B}$, and $I_{\mathbf{B}^*}$ are the indices of the optimal basis $\mathbf{B}^*$ in $\mathcal{B}$. We validate this observation in Figure 7.2c by using Orthogonal Matching Pursuit (OMP) [Pati et al., 1993; Tropp and Gilbert, 2007] to select a sparse set of basis vectors to reconstruct each 3D pose in a test corpus. The sparse representation is able to achieve lower reconstruction error with higher compaction on the test set than using a full PCA model. It is instructive to note the behavior in Figure 7.2c of the maximum reconstruction error, which usually correspond to atypical poses. For human poses, we conclude that the sparse representation demonstrates greater generalization ability than full PCA.

7.1.2 Anthropometric Regularity

Linear models allow cases where the 2D projection appears to be valid (i.e., the reprojection error is minimized), but the configuration in 3D violates anthropometric quantities such as the proportions of limbs. Enforcing anthropometric regularity (i.e., that limb lengths follow known proportions) would discourage such implausible configurations. For a limb[3] between the i^{th} and j^{th} landmark locations, we denote the normalized limb length as l_{ij}. The normalized limb lengths are set by normalizing with respect to the longest limb of the mean pose ($\boldsymbol{\mu}$). For a 3D pose $\mathbf{X}$, we can ensure anthropometric regularity by enforcing

$$\begin{gathered} \|\mathbf{X}_i - \mathbf{X}_j\|_2 = l_{ij}, \\ \forall (i,j) \in \mathcal{L} \end{gathered} \tag{7.4}$$

where $\mathcal{L} = \{(i,j)\}_{i=1}^{N_l}$ is the set of pairs of joints between which a limb exists and N_l is the total number of limbs in the model. Unfortunately, applying quadratic equality constraints on a linear least squares system is nonconvex. A *necessary* condition for anthropometric regularity is

$$\sum_{\forall (i,j)\in\mathcal{L}} \|\mathbf{X}_i - \mathbf{X}_j\|_2^2 = \sum_{\forall (i,j)\in\mathcal{L}} l_{ij}^2. \tag{7.5}$$

This constraint limits the sum of the squared distances between valid landmarks to be equal to the sum of squares of the limb lengths[4]. The feasible set of the constraint in Equation 7.5 contains the feasible set of the constraints in Equation 7.4. The necessary condition on the sum of squared limb lengths is therefore a relaxation of the constraints in Equation 7.4. As shown in [Gander, 1981], this necessary condition can be applied in closed form.

[3] We loosely define a limb to be a rigid length between two consecutive anatomical landmarks in the tree.

[4] Note that since we are using normalized limb-lengths, these constraints become constraints on limb proportions rather than on limb lengths.

Algorithm 7 Projected Matching Pursuit

```
1. Initialize $\mathbf{r}_0 = \mathbf{x} - (\mathbf{I} \otimes \mathbf{sR})\,\boldsymbol{\mu} - \mathbf{t} \otimes \mathbf{1}$
2. While $(\|\mathbf{r}_t\| \geq tol)$
   3. $i_{\max} = \arg\max_{\mathrm{i}} \langle \mathbf{r}_t,\ (\mathbf{I} \otimes \mathbf{s}_t\mathbf{R}_t)\,\mathbf{B}_i \rangle$
   4. $\mathbf{B}^* = [\mathbf{B}^*\ \mathbf{B}_{i_{\max}}]$
   5. Solve: $\{\mathcal{C}^*, \boldsymbol{\Omega}^*\} = \arg\min \|\hat{\mathbf{x}} - (\mathbf{I} \otimes \mathbf{sR})\,\mathbf{B}^*\boldsymbol{\Omega}\|_2$
      subject to constraints in Equation (7.8) using Section 7.2.2 & Section 7.2.3
   6. Recompute residual $\mathbf{r}_{t+1} = \mathbf{x} - (\mathbf{I} \otimes \mathbf{s}^*\mathbf{R}^*)\,(\mathbf{B}^*\boldsymbol{\Omega}^* + \boldsymbol{\mu}) - \mathbf{t}^* \otimes \mathbf{1}$
   7. Set $\boldsymbol{\Omega}_{t+1} = \boldsymbol{\Omega}^*$
8. Return $\{\mathcal{C}^*, \boldsymbol{\Omega}^*, \mathbf{B}^*\}$
```

7.2 Matching Pursuit under Camera Projection

We solve for the pose and camera by minimizing the reprojection error in the image. The resulting optimization problem can be stated as follows

$$\begin{aligned} \min_{\boldsymbol{\Omega}, \mathcal{C}, I_{\mathbf{B}^*}} \quad & \|\mathbf{x} - (\mathbf{I} \otimes \mathbf{sR})\,(\mathbf{B}^*\boldsymbol{\Omega} + \boldsymbol{\mu}) - \mathbf{t} \otimes \mathbf{1}\|_2 \\ \text{s.t.} \quad & \sum_{\forall(i,j)\in\mathcal{L}} \|\mathbf{X}_i - \mathbf{X}_j\|_2^2 = \sum_{\forall(i,j)\in\mathcal{L}} l_{ij}^2, \\ & \mathbf{B}^* \subset \mathcal{B}. \end{aligned} \tag{7.6}$$

Although the problem is non-linear, non-convex, and combinatorial, it has the following useful property in the set of arguments $(\mathcal{C}, \boldsymbol{\Omega}, I_{\mathbf{B}^*})$: we can solve optimally, or near-optimally, for each subset of the arguments given the rest. This property suggests a coordinate descent-style algorithm. Algorithm 7 describes a matching pursuit algorithm we refer to as *Projected Matching Pursuit* for coordinate descent on the reprojection error objective.

7.2.1 Algorithm

The combinatorial challenge of picking the optimal set of basis vectors from an overcomplete dictionary to represent a given signal is NP-hard. However, techniques exist to solve the sparse representation problem approximately with guarantees [Tropp and Gilbert, 2007; Tropp, 2004]. Greedy approaches such as orthogonal matching pursuit (OMP) [Mallat and Zhang, 1993; Tropp and Gilbert, 2007] reconstruct a signal $\mathbf{v}$ with a sparse linear combination of basis vectors from an overcomplete dictionary $\mathcal{B}$. It proceeds in a greedy fashion by choosing, at each iteration, the basis vector from $\mathcal{B}$ that is most aligned with the residual $\mathbf{r}$ (the residual is set equal to $\mathbf{v}$ in the first iteration). The new estimate of the signal $\hat{\mathbf{v}}$ is computed by reconstructing using the basis vectors selected at the current iteration and the new residual ($\mathbf{r} = \mathbf{v} - \hat{\mathbf{v}}$) is computed. The iterations proceed on the residual until K basis vectors are chosen or a tolerance on the residual error is reached.

In our scenario, we do not have access to the signal of interest, namely the 3D pose $\mathbf{X}$. Instead, we are only given the projection of the original 3D pose in the image $\mathbf{x}$. We present a matching pursuit algorithm for reconstructing a signal from its projection and an overcomplete dictionary. At each iteration of our algorithm, the optimal basis set $\mathbf{B}^*$ is augmented by matching the image residual with basis vectors projected under the current camera estimate and adding the basis vector which maximizes the inner product to the optimal set. Given the current optimal basis set $\mathbf{B}^*$, the pose and camera parameters are re-estimated as outlined in Section 7.2.2 and Section 7.2.3. The algorithm terminates when the optimal basis set has reached a predefined size or the image residual is smaller than a tolerance value. The procedure is summarized in Algorithm 7. We have an intuitive and feasible initialization in the mean 3D pose computed from the training corpus.

7.2.2 Estimating Basis Coefficients with Anthropometric Regularization

To encourage anthropometric regularity we enforce the necessary constraint from Equation 7.5 which limits the sum of squared limb lengths. We can write each 3D landmark $\mathbf{X}_i = \mathbf{E}_i\mathbf{X}$, where $\mathbf{E}_i = [\cdots\ \mathbf{0}\ \mathbf{I}_{3\times 3}\ \mathbf{0}\ \cdots]$ is a $3 \times 3P$ matrix that selects out the i^{th} landmark.

We can write $\mathbf{E}_{ij} = \mathbf{E}_i - \mathbf{E}_j$, and express each limb length as $\|\mathbf{E}_{ij}\mathbf{X}\| = l_{ij}$. Equation 7.5 can now be rewritten in matrix form as:

$$\|\mathbf{C}\mathbf{X}\|_2^2 = \sum_{\forall(i,j)\in\mathcal{L}} l_{ij}^2, \tag{7.7}$$

where $\mathbf{C}$ is a $3N_l \times 3P$ matrix of the N_l stacked $\mathbf{E}_{ij}$ matrices. Where N_l is the number of limbs.

Given the optimal basis set $\mathbf{B}^*$ and the camera $\mathcal{C}$, solving for the coefficients of the linear model $\mathbf{\Omega}$ can now be formulated as the following optimization problem:

$$\begin{aligned} \min_{\mathbf{\Omega}} \quad & \|\hat{\mathbf{x}} - \mathbf{sR} \otimes \mathbf{I}_{P\times P}\mathbf{B}^*\mathbf{\Omega}\|_2 \\ \text{s.t.} \quad & \|\mathbf{C}\mathbf{B}^*\mathbf{\Omega} - \mathbf{C}\boldsymbol{\mu}\|_2^2 = \sum_{\forall(i,j)\in\mathcal{L}} l_{ij}^2, \end{aligned} \tag{7.8}$$

where $\hat{\mathbf{x}} = \mathbf{x} - \mathbf{sR} \otimes \mathbf{I}_{P\times P}\boldsymbol{\mu} - \mathbf{t} \otimes \mathbf{1}_{P\times 1}$. The above problem is a linear least squares problem with a single quadratic equality constraint that can be solved optimally in closed form as shown in [Gander, 1981].

There also exists a natural lower bound on the length of the limb between the estimated joint locations, $\mathbf{X}_i^*$ and $\mathbf{X}_j^*$, in terms of the image projections $\mathbf{x}_i$ and $\mathbf{x}_j$. Using the triangle inequality we can show that

$$\|\mathbf{X}_i^* - \mathbf{X}_j^*\| \geq \|\mathbf{s}^{-1}(\mathbf{x}_i - \mathbf{x}_j)\|. \tag{7.9}$$

The above inequality shows that the estimated limb lengths are bounded by the length of the limbs in the image. Thus we can guarantee that the estimated limb length will not collapse to zeros as long as the limb has finite length in the image.

7.2.3 Estimating Camera Parameters

Given the current estimate of the pose $\hat{\mathbf{X}} = \mathbf{B}^{*}\boldsymbol{\Omega} + \boldsymbol{\mu}$, and the image projections $\mathbf{x}$, we need to recover the weak perspective camera parameters $\mathcal{C}$. This can be written as the following

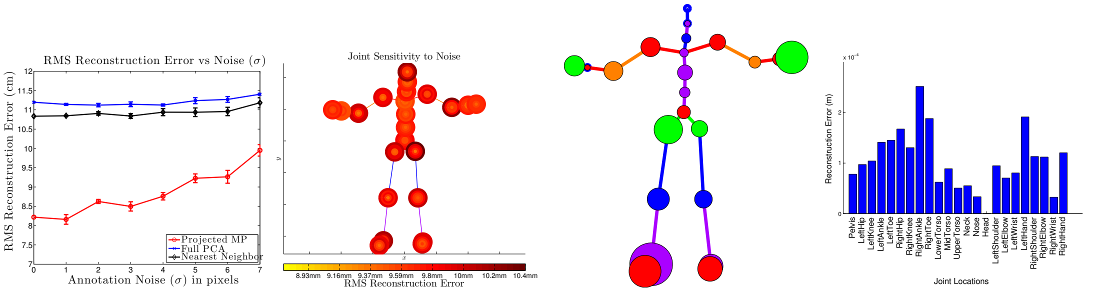
$$\begin{aligned} \min_{\mathcal{C}} \quad & \|\mathbf{x} - (\mathbf{I} \otimes \mathbf{sR})\,\hat{\mathbf{X}} - \mathbf{t} \otimes \mathbf{1}\|_2 \\ \text{s.t.} \quad & \mathbf{R}^T\mathbf{R} = \mathbf{I} \end{aligned} \tag{7.10}$$

This can be solved as an instance of the orthogonal procrustes problem [Schonemann, 1966] or by using an off-the-shelf non-linear least squares solver.

7.3 Experimental Analysis

We perform quantitative and qualitative evaluation of our method. We use the Carnegie Mellon motion capture database for quantitative tests and compare our results against using a representation baseline (direct PCA on the entire corpus) and a non-parametric nearest neighbor method.

For all experiments, an overcomplete shape basis dictionary was constructed by concatenating the shape bases learnt for a set of human actions. We use a model with 23 anatomical landmarks. Each pose in the motion capture corpus was aligned by procrustes analysis to a reference pose. Shape bases were then learnt for the following motion categories- *‘running’, ‘waving’, ‘arm movement’, ‘walking’, ‘jumping’, ‘jumping jacks’, ‘run’, ‘sit’, ’boxing’,’bend’* by collecting



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

Figure 7.3: Quantitative evaluation on optical motion capture. (a) We compare our method against two model baselines - a nearest neighbor approach and a linear model that uses PCA on the entire corpus. Reconstruction error is reported against annotation noise σ on a test corpus. (b) We evaluate the sensitivity of the reconstruction to each anatomical landmark annotation. (c) We show the sensitivity in reconstruction to missing landmarks. The radius of each circle indicates the relative magnitude of error in 3D incurred when the landmark is missing (d) The additional reconstruction incurred when the landmark is missing.

sequences from the CMU Motion Capture Dataset and concatenating PCA components which retained 99% of the energy from each motion category.

7.3.1 Quantitative Evaluation

Optical Motion Capture. To evaluate our methods we test our algorithm on a sequence of mixed activities from the CMU motion capture database. We take care to ensure that the motion capture frames come from sequences that were not used in the training of the shape bases. We project 30 frames of motion capture of diverse poses into 4 synthetically generated camera views. We then run our algorithm on the 2D projections of the joint locations to obtain the camera location and the pose of the human. We report 3D joint position error with increasing annotation noise σ in Figure 7.3a.

We compare our method against two baselines. The first baseline uses as a linear model, a basis computed by performing PCA on the entire training

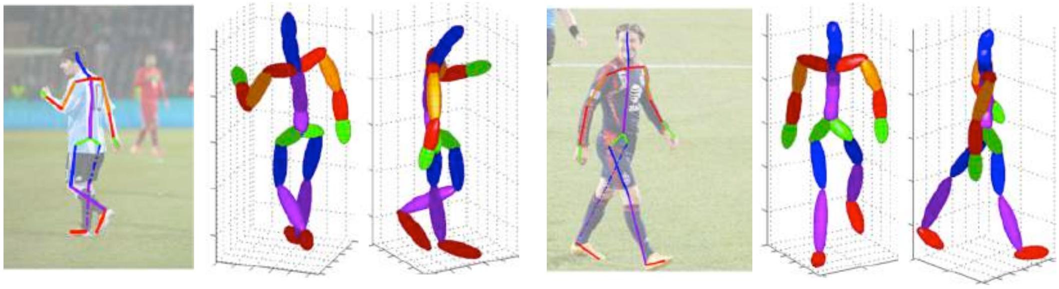


Figure 7.4: Our method is able to handle missing data. We show examples of reconstruction with missing annotations. The missing limbs are marked with dotted lines. We are able to reconstruct the pose and impute the missing landmarks in 3D.

corpus. Anthropometric constraints are enforced as in Section 7.2.2. The second baseline uses a non-parametric, nearest neighbor approach that retains all the training data. The 2D projections in each test example are matched to every 3D pose in the corpus by estimating the best-fit camera using the method in Section 7.2.3. The 3D pose that has the least reprojection error under the best-fit camera estimate is returned. The results are reported in Figure 7.3. We find that our method that used Projected Matching Pursuit achieves the lowest RMS reconstruction error. We also tested the effect of imposing an equality constraint on the sum-of-squared limb length ratios and find that we deviate from the ground truth on our test set by 13.1% on average.

We evaluate the comparative importance of the anatomical landmarks by performing two experiments:

Joint Sensitivity. We test the sensitivity of the reconstruction to each landmark individually. Each pose in the testing corpus is projected into 2D with synthetically generated cameras and each landmark is perturbed with Gaussian noise independently. Figure 7.3b shows the sensitivity of the reconstruction to each landmark. The maximum length of a limb in the image is 200 pixels, the minimum limb length is 20 pixels, and the average length of a limb in the image is 94.5. pixels The noise is varied to about 10% of the average limb length in the image.

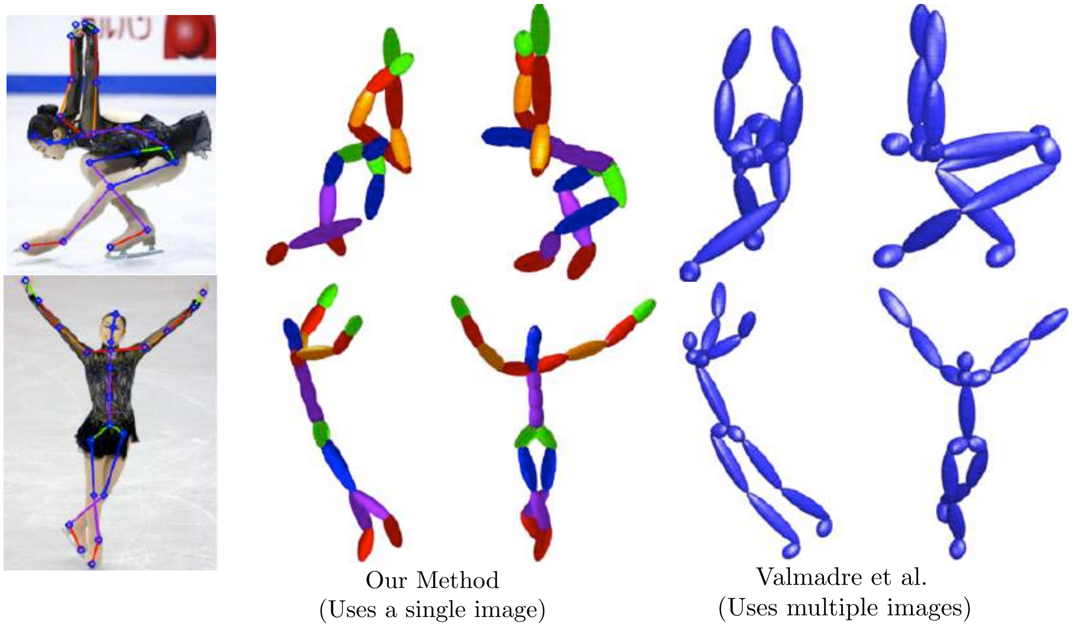
Missing Data. An advantage of our formulation is the ability to handle missing data. In Figure 7.4 we show examples of reconstructions obtained with incomplete annotations. We perform an ablative analysis of the joint annotations by removing each annotation in turn and measure the increase in the reconstruction error. We plot our results in Figure 7.3d. The radius of each circle is indicative of the error incurred when the annotation corresponding to that joint is missing. We find that the extremal joints are most informative and help in constraining the reconstruction.

7.3.2 Qualitative Evaluation

Comparison with recent work. We compare reconstructions obtained by our method to work by [Valmadre and Lucey, 2010]. Their method requires multiple images of the same person and requires a human annotator to resolve depth ambiguities. We present our comparative results in Figure 7.5. Our method is applied per frame to images of the ice skater Yu-Na Kim and compared to the reconstructions obtained by Valmadre et al. We can see in Figure 7.5 that we are able to obtain good reconstructions per image, without the requirement of a human annotator resolving the depth ambiguities.

Internet Images. We downloaded images of people in a variety of poses from the internet. The 2D joint locations were manually annotated. We present the results in Figures 7.7a and 7.6. In Figure 7.6 we first obtained individual camera and pose estimates for each of the annotated human figures. We then fixed the camera upright at an arbitrary location and placed the human figures using the estimated relative rigid pose. It can be seen that the camera estimates are consistent as the actors are placed in their correct locations.

Non-standard viewpoints. We also test our method on images taken



Our Method
(Uses a single image)

Valmadre et al.
(Uses multiple images)

Figure 7.5: **Comparsion with recent work**. Valmadre et al., estimate human pose using multiple images and requires additional annotation to resolve ambiguities. Our method achieves realistic results operating on a single image and does not require additional annotation

from non-standard viewpoints. We reconstruct the pose and relative camera from photographs downloaded from the internet taken from viewpoints that have generally been considered difficult for pose estimation algorithms. We are able to recover the pose and the viewpoint of the algorithm for such examples as shown in Figure 7.7b.

Monocular video. We demonstrate our algorithm on a set of key frames extracted from monocular video in Figure 7.7c. The relative camera estimates are aligned to a single view-point to obtain a sequence of the person performing an action. Note that we are able to estimate the relative pose between the camera and the human correctly resulting in a realistic reconstruction of the sequence.

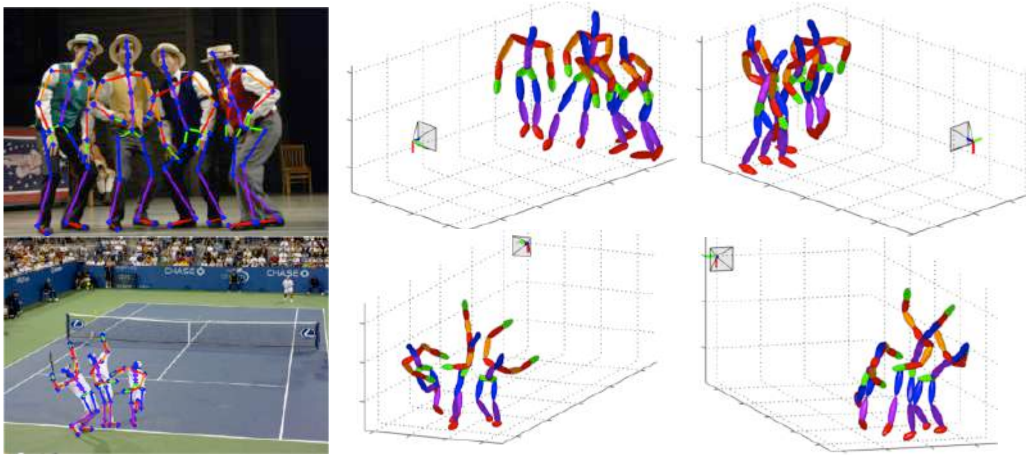
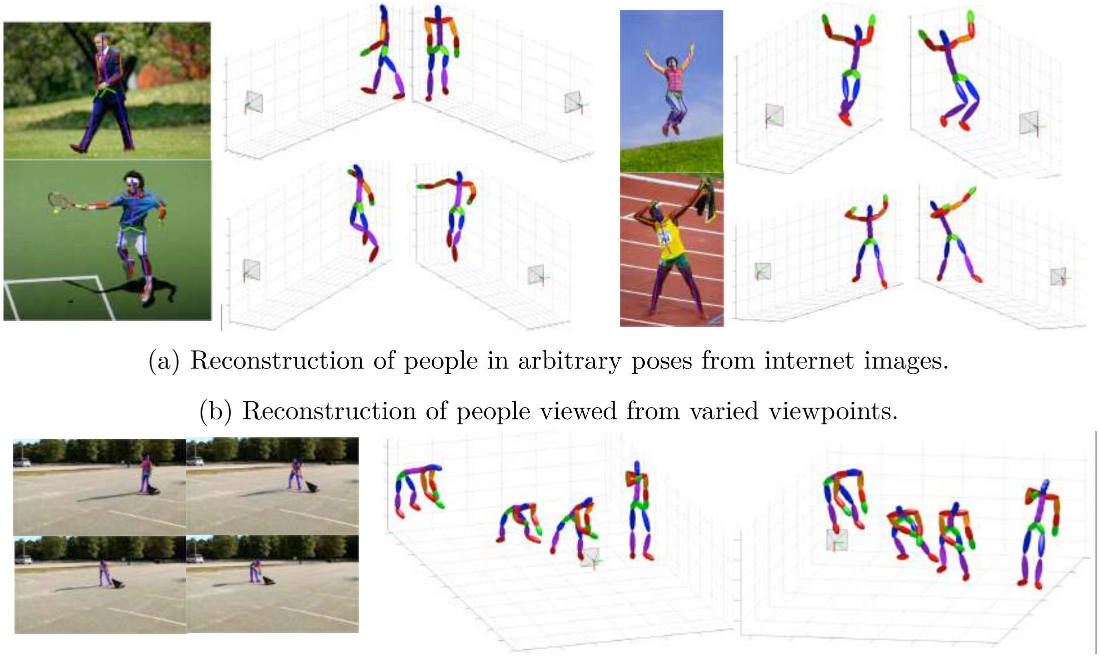


Figure 7.6: Reconstruction with multiple people in the same view. The camera estimation is accurate as the people are placed consistently.



(a) Reconstruction of people in arbitrary poses from internet images.

(b) Reconstruction of people viewed from varied viewpoints.

(c) Our algorithm applied to four frames of an annotated video.

Figure 7.7: We acheive realistic reconstructions for people in (a) arbitrary poses, (b) captured from varied viewpoints and (c) monocular video streams.

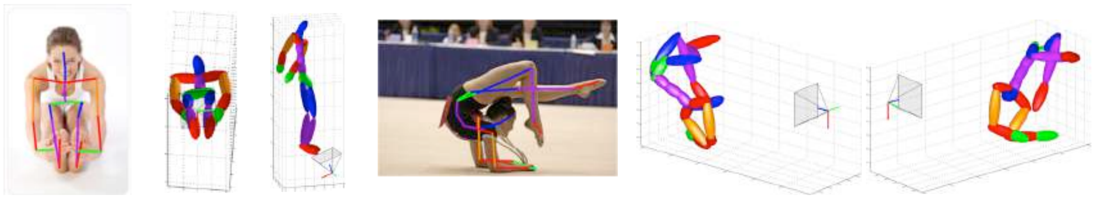


Figure 7.8: **Failure Cases.** The method does not recover the correct pose when there are strong perspective effects and if the mean pose is not a good initialization.

7.4 Discussion

We presented a new representation for human pose as a sparse linear embedding in an overcomplete dictionary. We develop a matching pursuit algorithm for estimating the sparse representation of 3D pose and the relative camera from only 2D image evidence while simultaneously maintaining anthropometric regularity. Every step in the matching pursuit iterations is computed in closed form, therefore the algorithm is efficient and takes on average 5 seconds per image to converge. We are able to achieve good generalization to a large range of poses and viewpoints. A case where the algorithm does not result in good reconstructions are in images with strong perspective effects where the weak perspective assumptions on the camera model are violated and in poses where the mean pose is not a reasonable initialization (See Figure 7.8).

CHAPTER 8

Conclusion

Natural articulated motion results in complex configurations of the articulated structure. Additionally, complexity arising due to appearance variation and imaging pose further challenges in the pose estimation problem. Traditional approaches have attempted to handle this complexity by building simplistic models, such as tree-structured models, where inference is tractable and exact. However, obtaining good performance in the pose estimation tasks requires incorporating global cues and non-tree like interactions. In this thesis, we identify that the core challenge in estimating the articulated pose of objects is the trade off that arises between modeling and inference. As we attempt to increase model fidelity by incorporating additional constraints, richer interactions between variables, and sophisticated features, we find that performing inference is difficult and learning the parameters of such models is a challenge.

In this thesis we addressed this trade off in two ways: (i) by incorporating physical and structural constraints by leveraging tractable substructures in modeling human pose and (ii) by enforcing tighter coupling between learning and inference via a sequential prediction procedure called a pose machine. In the former, we showed how we can incorporate constraints such as limb-length constraints and symmetric mutual exclusion constraints while solving tractable subproblems during inference. In the latter we showed that the sequential prediction procedure of a pose machine is well suited for combining complex interactions between variables with little or no overhead as the number of these interactions increase.

Modeling complex relationships between variables and incorporating global cues is key to improving the performance of structured prediction problems such as pose estimation. Important contextual interactions in the pose estimation problem include:

- **Spatial Context**: The spatial location of parts provide a strong cues for part detection. Parts which can be easily detected provide anchors for the detection of difficult parts, additionally, parts co-occur in a geometrically consistent fashion, which can be leveraged to improve detection.
- **Visual Context**: The visual appearance of nearby parts and the appearance at parts at a different resolution provide strong cues for part detection. The appearance of a neighboring part, for e.g., an upraised upper arm provides cues for the appearance and location of the elbow.
- **Semantic Context**: Labels of parts in a spatial vicinity also provide useful information for reasoning about articulated pose. The fact that the left knee occurs in a particular location should prevent the prediction of the right knee at the same location. A semantic hierarchy of parts organized into anatomical landmarks at the finest level of the hierarchy and parts corresponding to limbs in coarser levels of the hierarchy aids in pose estimation. Additionally, when objects interact, knowledge of interacting object provides additional information that can prevent characteristic errors.
- **Temporal Context**: Detection of parts in adjacent frames of a image sequence provide strong cues for part detection in the current frame. Tem-

poral consistency provides valuable features for reasoning about occlusion and prevents errors due to weak image evidence in particular frames.

The pose machine architecture developed in this thesis enables the incorporation of all the above types of contextual interactions between variables. In addition to being able to incorporate rich contextual and global cues, pose machines have the following distinct advantages:

Modular Feedforward Architecture: The pose machine reduces the structured prediction problem of pose estimation into a sequence of supervised classification problems. As a result learning does not require optimizing a complicated structured loss or the need for specialized solvers. The modular architecture allows the *plug and play* of any supervised predictor, including powerful feedforward predictors such as convolutional architectures. Convolutional pose machines allow the learning of both image and context features directly from raw input. The feedforward

Implicit Spatial Modeling: The pose machine architecture eschews probabilistic modeling in favor of a sequential prediction procedure. Classifiers in subsequent stages of the sequence use and combine cues in whatever fashion is most predictive of part locations and do not have to rely on hand-crafted spatial modeling.

Scalability and Extensibility: The architecture allows us to easily incorporate additional cues with very little additional computational overhead. The pose machine architecture can be easily extended to incorporate hierarchical cues, temporal cues (see Section 8.1.3) and cues from interacting objects (see Chapter 5). Additionally, when used in concert with convolutional architectures, the convolutional pose machine can be trained in an end-to-end manner on large datasets by leveraging first order online training methods.

Further advances in the state-of-the-art in pose estimation are likely to come from the use of massive internet-scale datasets for training. The architecture of a convolutional pose machine leaves us well poised to deal with large scale datasets. The online training of a convolutional pose machine allows for streaming data during learning thereby enabling the use of massive datasets.

The architecture of a convolutional pose machine allows for dense pixel-wise predictions that rely on rich local and global contextual cues and can be extended to a variety of different structured prediction problems in computer vision. The work in this thesis opens up several different avenues for research and potential future applications. In the following section, we outline areas for future study.

8.1 Future Work

8.1.1 MAP Inference

Pose machines in particular, and message passing inference machines in general, are currently limited to producing "marginal" like output confidence maps by emulating *marginal inference* in graphical models. Recall from Section 3.2.1 that marginal inference for a variable $\mathbf{x}_i$ of a graphical model with a joint distribution $P(\mathbf{x}_1, \ldots, \mathbf{x}_P)$ is performed by summing over the remaining variables (marginalization) using the sum-product algorithm, resulting in the message updates in Equations 3.22-3.24. The mean-field inference machine of [Munoz et al., 2010], the inference machines of [Ross et al., 2011] and the work presented in this thesis for pose estimation all focus on producing belief maps for each variable that resemble marginals.

An alternative method of inference is maximum a-priori inference, where the goal is to compute the highest scoring configuration for the variables. Computing the highest scoring configuration requires performing *max-product* inference on a graphical model, where the summations of Equation 3.22-3.24 are replaced by max operators and result in the computation of *max-marginals*:

$$\mu_{i\to f}(y_i) = \prod_{j\in\mathcal{N}(i)\setminus f} \mu_{f\to i}(y_j), \tag{8.1}$$

$$\mu_{f\to i}(y_i) = \max_{\mathbf{y}_f} \left(\psi_f(\mathbf{y}_f) \prod_{j\in\mathcal{N}(f)\setminus i} \mu_{j\to f}(y_j) \right), \tag{8.2}$$

$$m(y_i|\mathbf{x}) = \prod_{f\in\mathcal{N}(i)} \mu_{f\to i}(y_i), \tag{8.3}$$

where $m(y_i|\mathbf{x})$ is the max-marginal for variable y_i, $\mu_{i\to f}(y_i)$, $\mu_{f\to i}(y_i)$ are messages from a variables to factors and factors to variables respectively. $\mathcal{N}(i)\setminus f$ denotes the neighboring factor nodes of the variable i excluding the factor f and similarly $\mathcal{N}(f)\setminus i$ represents the neighboring variables of factor f excluding variable i. For a variable $\mathbf{y}_i$,

$$m(y_i = y) = \max_{\mathbf{y}\in\mathcal{Y}\setminus y_i : y_i = y} P(y_1, \ldots, y_P) \tag{8.4}$$

A max-marginal for a variable is the score of the configuration, for each setting of that variable, given all other variables have been set optimally. Designing an inference machine to emulate MAP inference, remains an open problem and an interesting direction for future research.

8.1.2 Inference over Latent Variables

A limitation of the pose machine architecture that currently limits its deployment for traditional bounding box style parts-based object detection is the requirement of part annotations. In the deformable part model of [Felzenszwalb et al., 2008], the star-structured graphical model is trained with latent part locations, which are initialized with mean positions. During learning, only the bounding box of the full object is given, while the part locations are updated latently. Additionally, for some objects, it is not clear what good "keypoints" are. Interesting future work would be to automatically find good keypoints and to incorporate latent

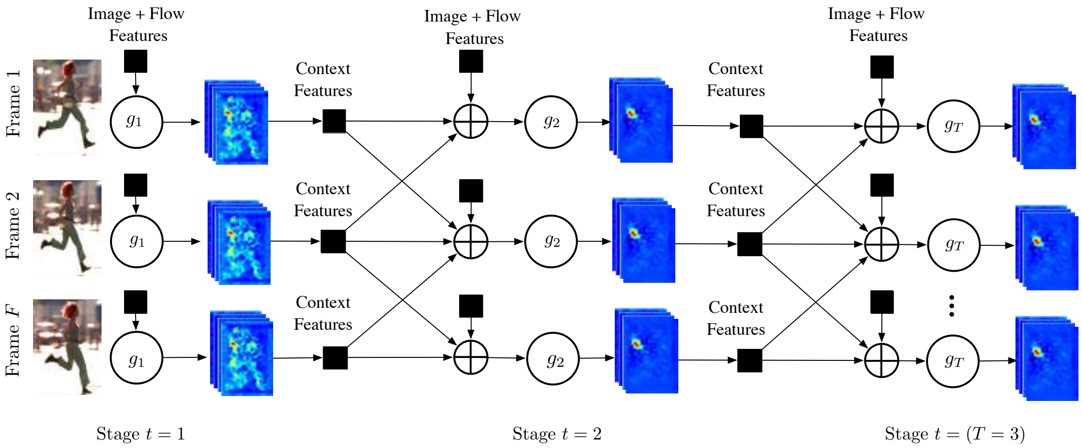


Figure 8.1: **Temporal Pose Machines.** Predictions in each frame take advantage of predictions in adjacent frames in additional to temporal continuity features between adjacent frames.

part location updates into the training procedure removing the need for detailed part annotations. This would allow us to perform object-detection that leverages the rich spatial modeling capacity of the pose-machine architecture for reasoning about complex interactions between object parts.

8.1.3 Incorporating Temporal Context

As natural motion is smooth, beliefs for the location of a part in adjacent frames inform the prediction of the location of a part in the current frame. Additionally, when parts move in and out of occlusion, temporal continuity provides strong cues for part location under occlusion. Interesting future work would be to extend the pose machines architecture to handle video and incorporate temporal cues from adjacent frames.

In Figure 8.1 we show a proposed model for a temporal pose machine in which context features are passed between adjacent frames. The proposed architecture operates on sliding window of F frames. In the first stage each predictor makes

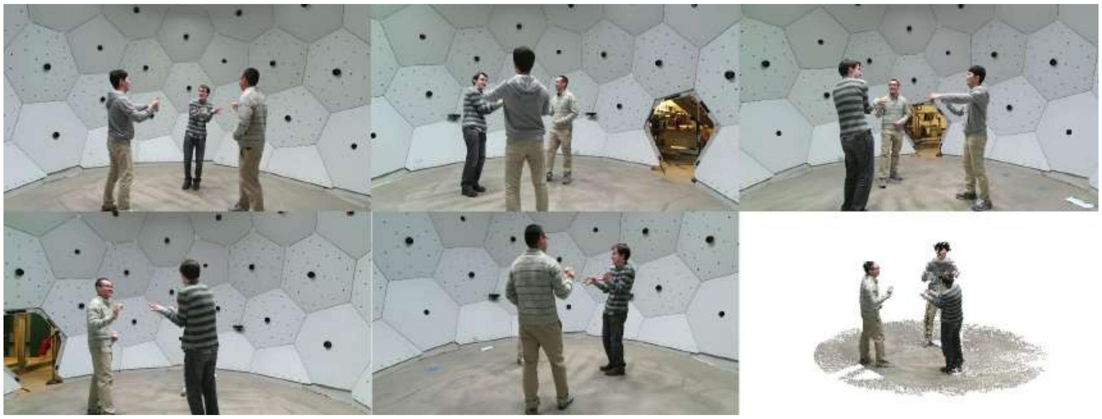


Figure 8.2: When multiple people interact, there is often inter-person occlusions, non-canonical relative views and close proximity which degrade standard pose estimation performance. Image courtesy, Tomas Simon.

a pose prediction independent of adjacent frames. In subsequent stages, each predictor combines context features computed from the present, past and future frames in addition to the original image features to produce an updated confidence for the location of each part.

8.1.4 3D Pose Machines

When multiple people interact, inter-person occlusions, non-canonical relative views and close proximity degrade standard pose estimation performance. In Figure 8.2, we show a point cloud reconstructed from multiple commodity depth-sensors in a capture environment. We see that although no one view is completely unoccluded, from multiple views, we are able to assemble a complete raw 3D representation where the objects are separated in 3D. Operating on such a point-cloud representation may mitigate some of the challenges of standard pose estimation algorithms. The design of a pose machine to perform landmark localization on 3D point clouds is an interesting direction of future work. In the same vein, extension of pose machines to incorporate multiple modalities of data. In early work by [Munoz et al., 2012], additional modes of data, namely registered LIDAR data was incorporated into the hierarchical inference machine model of [Munoz et al., 2010]. Interesting future work would include extending the pose-machine model to incorporate addtional input modes such as point-clouds to perform pose estimation directly in 3D.

Bibliography

Adams, A., J. Baek, and M. A. Davis
2010. Fast high-dimensional filtering using the permutohedral lattice. In *Computer Graphics Forum*. Wiley Online Library. 4

Agarwal, A. and B. Triggs
2004a. 3D Human Pose from Silhouettes by Relevance Vector Regression. In *IEEE Conference on Computer Vision and Pattern Recognition, IEEE*. 2.1.1

Agarwal, A. and B. Triggs
2004b. 3d human pose from silhouettes by relevance vector regression. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.1.2

Andriluka, M., S. Roth, and B. Schiele
2008. People-tracking-by-detection and people-detection-by-tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.2

Andriluka, M., S. Roth, and B. Schiele
2009. Pictorial Structures Revisited: People Detection and Articulated Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.3, 2.1.1, 2.1.2, 4

Andriluka, M., S. Roth, and B. Schiele

2010. Monocular 3D Pose Estimation and Tracking by Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.1.1, 4

Andriluka, M. and L. Sigal

2012. Human context: Modeling human-human interactions for monocular 3d pose estimation. In *Articulated Motion and Deformable Objects*. Springer. 2.3, 5

Andriyenko, A. and K. Schindler

2010. Globally optimal multi-target tracking on a hexagonal lattice. *European Conference on Computer Vision*. 6.3, 6.3, 6.3.2

Bai, X. and Z. Tu

2009. Auto-context and its application to high-level vision tasks and 3d brain image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2.1.1

Baker, S. and I. Matthews

2004. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*. 6.3.1

Bassili, J. N.

1978. Facial motion in the perception of faces and of emotional expression. *Journal of experimental psychology: human perception and performance*. 1

Bengio, Y.

2009. Learning deep architectures for AI. *Foundations and trends in Machine Learning*. 4.2

Bengio, Y., P. Simard, and P. Frasconi

1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*. 4.6.2

Berclaz, J., F. Fleuret, E. Turetken, and P. Fua

2011. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 6.3

Biederman, I.

1987. Recognition-by-components: a theory of human image understanding. *Psychological review*. 2

Binford, T. O.

1971. Visual perception by computer. In *IEEE conference on Systems and Control*. 2

Blake, R. and M. Shiffrar

2007. Perception of human motion. *Annu. Rev. Psychol.* 1

Bourdev, L. and J. Malik

2009. Poselets: Body part detectors trained using 3d human pose annotations. In *International Conference on Computer Vision*. (Link). 2.4, 2.1.2

Boyd, S. and L. Vandenberghe

2004. *Convex Optimization*. Cambridge University Press. 7

Bradley , D.

2010. *Learning In Modular Systems*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA. 4.6.2

Bregler, C. and J. Malik

1998. Tracking people with twists and exponential maps. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.2

Breiman, L.

2001. Random forests. *Machine learning*. 3.1.2, 4.2.1

Brubaker, M. A., D. J. Fleet, and A. Hertzmann
2007. Physics-based person tracking using simplified lower-body dynamics. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.2

Buehler, P., M. Everingham, D. P. Huttenlocher, and A. Zisserman
2008. Long term arm and hand tracking for continuous sign language tv broadcasts. In *British Machine Vision Conference*. 2.1.1

Caruana, R. and A. Niculescu-Mizil
2006. An empirical comparison of supervised learning algorithms. In *ICML*. 3.1.2, 3.1.2, 4.2.1

Carvalho, V. and W. Cohen
2005. Stacked sequential learning. In *IJCAI*. 4.6.1

Chen, X. and A. Yuille
2014a. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in Neural Information Processing Systems*. 2.1.1, 2.1.2, 4.8.1, 4.1, 4.2, 4.8.1

Chen, X. and A. Yuille
2014b. Parsing occluded people by flexible compositions. *arXiv preprint arXiv:1412.1526*. 2.3, 5

Cooper, G. F.
1990. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial intelligence*. 3.2.2

Cootes, T., G. Edwards, and C. Taylor
2001. Active Appearance Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE*. 7.1

Dalal, N. and B. Triggs

2005. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*. 3.1, 3.1.1

Dantone, M., J. Gall, C. Leistner, and L. Van Gool

2013. Human pose estimation using body parts dependent joint regressors. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.1.1

Denil, M., D. Matheson, and N. de Freitas

2013. Consistency of online random forests. *arXiv preprint arXiv:1302.4853*. 3.1.2

Dollár, P., Z. Tu, P. Perona, and S. Belongie

2009. Integral channel features. In *British Machine Vision Conference*. 3.1, 3.1.1

Duan, K., D. Batra, and D. J. Crandall

2012. A multi-layer composite model for human pose estimation. In *British Machine Vision Conference*. 2.3

Eichner, M. and V. Ferrari

2010. We are family: Joint pose estimation of multiple persons. In *European Conference on Computer Vision*. Springer. 2.3

Felzenszwalb, P. and D. Huttenlocher

2004. Distance transforms of sampled functions. Technical report, Cornell University. 2.1.1

Felzenszwalb, P., D. McAllester, and D. Ramanan

2008. A discriminatively trained, multiscale, deformable part model. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.1.1, 8.1.2

Felzenszwalb, P. F. and D. P. Huttenlocher
2005. Pictorial structures for object recognition. *International Journal of Computer Vision*. 1.2, 2.3, 2.1.1, 2.1.2, 4

Ferrari, V., M. Marin-Jimenez, and A. Zisserman
2008. Progressive search space reduction for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.1.1, 2.2, 6.4.2

Fischler, M. A. and R. A. Elschlager
1973. The representation and matching of pictorial structures. *IEEE Transactions on Computers*. 1.2, 2.2, 2, 2.1.1

Forsyth, D. A., O. Arikan, and L. Ikemoto
2006. *Computational Studies of Human Motion: Tracking and Motion Synthesis*. Now Publishers Inc. 2

Freund, Y. and R. E. Schapire
1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*. 3.1.2

Friedman, J. H.
2001. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*. 3.1.2, 4.2.1

Fukushima, K.
1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*. 3.1.3

Gander, W.
1981. Least Squares with a Quadratic Constraint. *Numerische Mathematik*. 7, 7.1.2, 7.2.2

Gavrila, D. M.
1999. The visual analysis of human movement: A survey. *Computer vision and image understanding*. 2

Ghiasi, G., Y. Yang, D. Ramanan, and C. C. Fowlkes
2014. Parsing occluded people. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.3, 5

Gibson, J., G. Kaplan, H. Reynolds, and K. Wheeler
1969. The change from visible to invisible. *Attention, Perception, & Psychophysics*. 6

Giese, M. A. and T. Poggio
2003. Neural mechanisms for the recognition of biological movements. *Nature Reviews Neuroscience*. 1

Girshick, R., J. Donahue, T. Darrell, and J. Malik
2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (IEEE Conference on Computer Vision and Pattern Recognition), 2014 IEEE Conference on*. 3.1.3

Glorot, X. and Y. Bengio
2010. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*. 4.6.2

Grubb, A. and J. A. Bagnell
2011. Generalized boosting algorithms for convex optimization. In *ICML*. 3.1.2, 4.2.1

Gupta, M., Q. Yin, and S. K. Nayar
2013. Structured light in sunlight. In *IEEE International Conference on Computer Vision*. 4

Helmholtz, H. v. et al.

1909. Handbuch der physiologischen optik. *Hamburg: Voss*. 1.1

Hinton, G. E., N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov

2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*. 3.1.3

Hogg, D.

1983. Model-based vision: a program to see a walking person. *Image and Vision Computing*. 2.2

Hornik, K., M. Stinchcombe, and H. White

1989. Multilayer feedforward networks are universal approximators. *Neural networks*. 3.2.2, 4.3.2

Isard, M. and A. Blake

1998. Condensation: conditional density propagation for visual tracking. *International Journal of Computer Vision*. 2.2

Jain, A., J. Tompson, M. Andriluka, G. W. Taylor, and C. Bregler

2014. Learning human pose estimation features with convolutional networks. In *International Conference on Learning Representations*. 2.1.2

Jiang, H., S. Fels, and J. Little

2007. A linear programming approach for multiple object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*. 6.3

Jiang, H. and D. R. Martin

2008. Global pose estimation using non-tree models. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.1.1

Johansson, G.

1973. Visual perception of biological motion and a model for its analysis. *Attention, Perception, & Psychophysics*. 1, 2

Johnson, S. and M. Everingham

2010. Clustered pose and nonlinear appearance models for human pose estimation. In *British Machine Vision Conference*. 2.1.1, 4, 4.3

Kiefel, M. and P. V. Gehler

2014. Human pose estimation with fields of parts. In *European Conference on Computer Vision*. 2.1.1

Koffka, K.

1935. Principles of gestalt psychology. 2

Koller, D. and N. Friedman

2009. *Probabilistic graphical models: principles and techniques*. MIT press. 3

Kolmogorov, V. and R. Zabin

2004. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 3.2.1

Komodakis, N., N. Paragios, and G. Tziritas

2007. Mrf optimization via dual decomposition: Message-passing revisited. In *IEEE International Conference on Computer Vision*. 3.2.1

Kozlowski, L. T. and J. E. Cutting

1977. Recognizing the sex of a walker from a dynamic point-light display. *Perception & Psychophysics*. 1, 1

Krähenbühl, P. and V. Koltun

2011. *Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials*. Advances in Neural Information Processing Systems. 4

Krizhevsky, A., I. Sutskever, and G. E. Hinton
2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. 2.1.1, 3.1.3, 4.2.2, 4.4.2

Kschischang, F. R., B. J. Frey, and H.-A. Loeliger
2001. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*. 3.2.1

Kulesza, A. and F. Pereira
2007. Structured learning with approximate inference. In *Advances in Neural Information Processing Systems*. 3.2.2, 4

Kumar, S., J. August, and M. Hebert
2005. Exploiting inference for approximate parameter learning in discriminative fields: An empirical study. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Springer Berlin Heidelberg. 1.2, 3.2.2

Lafferty, J., A. McCallum, and F. C. Pereira
2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*. 1.1

Lan, X. and D. P. Huttenlocher
2005. Beyond trees: Common-factor models for 2d human pose recovery. In *IEEE International Conference on Computer Vision*. 2.1.1

Lawrence, N. D.
2004. Gaussian process latent variable models for visualisation of high dimensional data. *Advances in Neural Information Processing Systems*. 2.2

LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel
1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*. 3.1.3, 3.1.3, 5.2

LeCun, Y., S. Chopra, R. Hadsell, M. Ranzato, and F. Huang
2006. A tutorial on energy-based learning. *Predicting structured data.* 3.2.2

Lee, H.-J. and Z. Chen
1985. Determination of 3D Human Body Postures from a Single View. *Computer Vision, Graphics, and Image Processing.* 3, 1

Long, J., E. Shelhamer, and T. Darrell
2015. Fully convolutional networks for semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition.* 4.2, 4.2.2, 4.3.2, 4.4.2

Macionis, J. and L. Gerber
2010. Sociology, seventh canadian edition. *Don Mills: Pearson Education Canada.* 1

Mallat, S. and Z. Zhang
1993. Matching Pursuits with Time-Frequency Dictionaries. *IEEE Transactions on Signal Processing.* 7.2.1

Marey, E.-J.
1878. *La méthode graphique dans les sciences expérimentales et principalement en physiologie et en médecine.* G. Masson. 2

Marey, E.-J.
1895. *Movement.* D. Appleton. 2

Marr, D. and H. Nishihara
1978. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B, Biological Sciences.* 2.2, 2

Matthews, I. and S. Baker
2003. Active Appearance Models Revisited. *International Journal of Computer Vision.* 7

MoCap, C.
. Carnegie Mellon University Graphics Lab Motion Capture Database. http://mocap.cs.cmu.edu. 7, 2

Moeslund, T. B., A. Hilton, V. Krüger, and L. Sigal
2011. *Visual analysis of humans*. Springer. 2

Mori, G. and J. Malik
2006. Recovering 3d human body configurations using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*. 2.1.2

MOSEK
. The MOSEK optimization software. (Link). 6.3

Munoz, D.
2013. *Inference Machines: Parsing Scenes via Iterated Predictions*. PhD thesis. 3.2.2

Munoz, D., J. A. Bagnell, and M. Hebert
2010. Stacked hierarchical labeling. In *European Conference on Computer Vision*. 1.2, 3.2, 3.2.2, 8.1.1, 8.1.4

Munoz, D., J. A. Bagnell, and M. Hebert
2012. Co-inference machines for multi-modal scene analysis. In *European Conference on Computer Vision*. 8.1.4

Muybridge, E.
1899. *Animals in motion: an electro-photographic investigation of consecutive phases of animal progressive movements*. Chapman & Hall. 1

Muybridge, E.
1901. *The human figure in motion: an electro-photographic investigation of consecutive phases of muscular actions*. Chapman and Hall. 1

Nair, V. and G. E. Hinton

2010. Rectified linear units improve restricted boltzmann machines. In *ICML*. 3.1.3, 4.2.2, 4.4.2

Ouyang, W., X. Chu, and X. Wang

2014. Multi-source deep learning for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 4.8.1, 4.1

Park, D. and D. Ramanan

2011. N-best maximal decoders for part models. In *IEEE International Conference on Computer Vision*. 2.2, 6.4, 6.4.1, 6.4, **??**, **??**, 6.4.2, **??**, **??**, **??**, **??**, 6.5

Park, H. S., E. Jain, and Y. Sheikh

2012. 3d social saliency from head-mounted cameras. In *Advances in Advances in Neural Information Processing Systems*. 1.3.3

Park, H. S., E. Jain, and Y. Sheikh

2013. Predicting primary gaze behavior using social saliency fields. In *IEEE International Conference on Computer Vision*. 1.3.3

Pati, Y., R. Rezaiifar, and P. Krishnaprasad

1993. Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition. In *1993 Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*. 7.1.1

Peak, V.

2005. Vicon motion capture system. 1.3.4

Peelen, M. V. and P. E. Downing

2007. The Neural Basis of Visual Body Perception. *Nature Reviews Neuroscience*. 7

Pentland, A. P.
1986. Perceptual organization and the representation of natural form. *Artificial Intelligence*. 2

Perrett, D., P. Smith, A. Mistlin, A. Chitty, A. Head, D. Potter, R. Broennimann, A. Milner, and M. Jeeves
1985. Visual analysis of body movements by neurones in the temporal cortex of the macaque monkey: a preliminary report. *Behavioural brain research*. 1, 1

Pishchulin, L., M. Andriluka, P. Gehler, and B. Schiele
2013a. Poselet conditioned pictorial structures. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.1.1, 2.4, 2.1.2, 4, 4.1, 4.2

Pishchulin, L., M. Andriluka, P. Gehler, and B. Schiele
2013b. Strong appearance and expressive spatial models for human pose estimation. In *IEEE International Conference on Computer Vision*. 2.1.1, 2.1.2, 4.8.1, 4.8.1

Pishchulin, L., A. Jain, M. Andriluka, T. Thormahlen, and B. Schiele
2012. Articulated people detection and pose estimation: Reshaping the future. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.3

Poizner, H., U. Bellugi, and V. Lutes-Driscoll
1981. Perception of american sign language in dynamic point-light displays. *Journal of experimental psychology: Human perception and performance*. 1, 1

Pollick, F. E., H. M. Paterson, A. Bruderlin, and A. J. Sanford
2001. Perceiving affect from arm movement. *Cognition*. 1, 1

Ramakrishna, V., D. Munoz, M. Hebert, J. Bagnell, A., and Y. Sheikh
2014a. Posemachines: Articulated pose estimation via inference machines. In *European Conference on Computer Vision*. 4.1

Ramakrishna, V., D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh
2014b. Pose Machines: Articulated Pose Estimation via Inference Machines. In *European Conference on Computer Vision*. 4.8.1

Ramanan, D.
2007. Learning to parse images of articulated bodies. *Advances in Advances in Neural Information Processing Systems*. 2.1.1

Ramanan, D., D. Forsyth, and A. Zisserman
2007. Tracking people by learning their appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 6.4

Ramanan, D., D. A. Forsyth, and A. Zisserman
2005. Strike a Pose: Tracking people by finding stylized poses. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.2, 4

Remus, D. and S. Engel
2003. Motion from occlusion. *Journal of Vision*. 6

Roberts, L. G.
1963. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology. 2

Ross, S., D. Munoz, M. Hebert, and J. A. Bagnell
2011. Learning message-passing inference machines for structured prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1.2, 3.2, 3.2.2, 3.2.2, 4.4.1, 8.1.1

Rumelhart, D. E., G. E. Hinton, and R. J. Williams
1985. Learning internal representations by error propagation. Technical report, DTIC Document. 3.1.3

Rumelhart, D. E., G. E. Hinton, and R. J. Williams

1988. Learning representations by back-propagating errors. *Cognitive modeling.* 5.2

Runeson, S. and G. Frykholm

1981. Visual perception of lifted weight. *Journal of Experimental Psychology: Human Perception and Performance.* 1, 1

Saffari, A., C. Leistner, J. Santner, M. Godec, and H. Bischof

2009. On-line random forests. In *IEEE International Conference on Computer Vision Workshops.* 3.1.2

Safonova, A., J. K. Hodgins, and N. S. Pollard

2004. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Transactions on Graphics (SIGGRAPH 2004).* 7

Sapp, B. and B. Taskar

2013. MODEC: Multimodal Decomposable Models for Human Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition.* 4, 4.3.2, 4.8.2

Sapp, B., A. Toshev, and B. Taskar

2010. Cascaded models for articulated pose estimation. In *European Conference on Computer Vision.* Springer. 2.1.1, 2.1.2

Sapp, B., D. Weiss, and B. Taskar

2011. Parsing human motion with stretchable models. In *IEEE Conference on Computer Vision and Pattern Recognition.* 2.1.1, 2.2

Schonemann, P.

1966. A Generalized Solution of the Orthogonal Procrustes Problem. *Psychometrika.* 7.2.3

Sermanet, P., D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun
2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*. 3.1.3, 4.2

Sharp, T.
2008. Implementing decision trees and forests on a gpu. In *European Conference on Computer Vision*. Springer. 3.1.2

Sheikh, Y. A., A. Datta, and T. Kanade
2008. On the sustained tracking of human motion. In *Face & Gesture*. 2.2

Shotton, J., R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, et al.
2013. Efficient human pose estimation from single depth images. In *Decision Forests for Computer Vision and Medical Image Analysis*. Springer. 3.1.2

Sigal, L. and M. J. Black
2006. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.1.1, 2.2, 6

Sun, M. and S. Savarese
2011. Articulated part-based model for joint object detection and pose estimation. In *IEEE International Conference on Computer Vision*. 2.3, 2.1.1, 2.1.2

Sutskever, I.
2013. *Training recurrent neural networks*. PhD thesis, University of Toronto. 3.1.3

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich
2014. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*. 3.1.3

Taskar, B., C. Guestrin, and D. Koller
2003. Maximum-margin markov networks. *Advances in Neural Information Processing Systems*. 3.2.1

Taylor, G. W., R. Fergus, G. Williams, I. Spiro, and C. Bregler
2010. Pose-sensitive embedding by nonlinear nca regression. In *Advances in Neural Information Processing Systems*. 2.1.2

Tian, T.-P. and S. Sclaroff
2010. Fast globally optimal 2d human detection with loopy graph models. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.1.1

Tian, Y., C. L. Zitnick, and S. G. Narasimhan
2012. Exploring the spatial hierarchy of mixture models for human pose estimation. In *European Conference on Computer Vision*. 2.1.1

Tompson, J. J., A. Jain, Y. LeCun, and C. Bregler
2014. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems*. 2.1.2, 4.2, 4.8.1

Toshev, A. and C. Szegedy
2013. DeepPose: Human pose estimation via deep neural networks. *IEEE Conference on Computer Vision and Pattern Recognition*. 2.1.1

Tropp, J.
2004. Greed is Good: Algorithmic Results for Sparse Approximation. *IEEE Transactions on Information Theory*. 7.2.1

Tropp, J. A. and A. C. Gilbert
2007. Signal Recovery from Random Measurements via Orthogonal Matching Pursuit. *IEEE Transactions on Information Theory*. 7.1.1, 7.2.1

Tsochantaridis, I., T. Joachims, T. Hofmann, and Y. Altun

2005. Large margin methods for structured and interdependent output variables. In *Journal of Machine Learning Research*, Pp. 1453–1484. 3.2.1

Urtasun, R., D. J. Fleet, and P. Fua

2006. 3d people tracking with gaussian process dynamical models. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.2

Valmadre, J. and S. Lucey

2010. Deterministic 3D Human Pose Estimation using Rigid Structure. In *European Conference on Computer Vision*. 7.3.2

Viola, P. and M. Jones

2001. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition*. 3.1

Vondrak, M., L. Sigal, and O. C. Jenkins

2008. Physical simulation for probabilistic motion tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.2

Wainwright, M. J., T. S. Jaakkola, and A. S. Willsky

2005. Map estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory*. 3.2.1

Wainwright, M. J. and M. I. Jordan

2008. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1. 2, 3.2.1

Wang, F. and Y. Li

2013. Beyond physical connections: Tree models in human pose estimation. 4.2, 4.8.1

Wang, Y. and G. Mori
2008. Multiple tree models for occlusion and spatial constraints in human pose estimation. In *European Conference on Computer Vision*. 2.1.1

Wang, Y., D. Tran, and Z. Liao
2011. Learning hierarchical poselets for human parsing. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.3, 2.1.1, 2.1.2

Wolpert, D. H.
1992. Stacked Generalization. *Neural Networks*. 4.6.1

Xiao, J., S. Baker, I. Matthews, and T. Kanade
2004. Real-Time Combined 2D+3D Active Appearance Models. In *IEEE Conference on Computer Vision and Pattern Recognition, IEEE*. 7

Yang, Y., S. Baker, A. Kannan, and D. Ramanan
2012. Recognizing proxemics in personal photos. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.3, 5.1, 5, 5.4, 5.4.3, 5.7, 5.8

Yang, Y. and D. Ramanan
2011. Articulated pose estimation with flexible mixtures-of-parts. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.3, 2.1.1, 2.4, 2.1.2, 2.2, 4, 6.1, 6.3.1, 6.3.1, 6.4

Yang, Y. and D. Ramanan
2013. Articulated human detection with flexible mixtures of parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1.2, 4.8.1, 5.4

Zatsiorsky, V. M.
1998. *Kinematics of human motion*. Human Kinetics. 2

Zhu, L. L., Y. Chen, Y. Lu, C. Lin, and A. Yuille
2008. Max margin and/or graph learning for parsing the human body. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.3

Zuffi, S., O. Freifeld, and M. J. Black
2012. From pictorial structures to deformable structures. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.4, 2.1.2

Zuffi, S., J. Romero, C. Schmid, and M. J. Black
2013. Estimating human pose with flowing puppets. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2.1.2