1-1-2012

Efficient Human Pose Estimation with Image-dependent Interactions

Benjamin John Sapp
University of Pennsylvania, bensapp@cis.upenn.edu

Efficient Human Pose Estimation with Image-dependent Interactions

Abstract
Human pose estimation from 2D images is one of the most challenging and computationally-demanding problems in computer vision. Standard models such as Pictorial Structures consider interactions between kinematically connected joints or limbs, leading to inference cost that is quadratic in the number of pixels. As a result, researchers and practitioners have restricted themselves to simple models which only measure the quality of limb-pair possibilities by their 2D geometric plausibility.

In this talk, we propose novel methods which allow for efficient inference in richer models with data-dependent interactions. First, we introduce structured prediction cascades, a structured analog of binary cascaded classifiers, which learn to focus computational effort where it is needed, filtering out many states cheaply while ensuring the correct output is unfiltered. Second, we propose a way to decompose models of human pose with cyclic dependencies into a collection of tree models, and provide novel methods to impose model agreement. Finally, we develop a local linear approach that learns bases centered around modes in the training data, giving us image-dependent local models which are fast and accurate.

These techniques allow for sparse and efficient inference on the order of minutes or seconds per image. As a result, we can afford to model pairwise interaction potentials much more richly with data-dependent features such as contour continuity, segmentation alignment, color consistency, optical flow and multiple modes. We show empirically that these richer models are worthwhile, obtaining significantly more accurate pose estimation on popular datasets.

EFFICIENT HUMAN POSE ESTIMATION WITH IMAGE-DEPENDENT INTERACTIONS

Benjamin John Sapp

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2012

Ben Taskar, Associate Professor
Computer and Information Science
Supervisor of Dissertation

Val Tannen, Professor
Computer and Information Science
Graduate Group Chairperson

Dissertation Committee

Kostas Daniilidis, Professor
Computer and Information Science
University of Pennsylvania

Jianbo Shi, Associate Professor
Computer and Information Science
University of Pennsylvania

Camillo J. Taylor, Associate Professor
Computer and Information Science
University of Pennsylvania

David Forsyth, Professor
Computer Science
University of Illinois

EFFICIENT HUMAN POSE ESTIMATION WITH IMAGE-DEPENDENT INTERACTIONS

2012

Benjamin John Sapp

To my family:

Mom and Dad, for giving me a lifelong place to call home;

Joe, John, and Scott for lifelong friendship.

Acknowledgements

I have been more successful and satisfied with my Ph.D. career than I had dared to hope when I started, and I can't imagine it happening anywhere but Penn. This has not so much to do with The City of Brotherly Love, the "greene Country Towne", the so-called American Paris, despite its prideworthy grit and soul, its food trucks, diners, and BYOBs, its parks and squares filled with dogs and children, its brick and rainbow row houses, its Chinatown and Italian Market, Schuykill River Trail and Wissahickon. Philadelphia has made a lasting impression on me, but it is the revolving cast of characters around me these last five years that have made all the difference:

I'd like to thank Kostas Daniilidis for convincing me to come to Penn with his generous hospitality while introducing me to the GRASP lab. I have continued to enjoy his hospitality and skillful diplomacy throughout my career, including while heading my thesis committee. I'd also like to thank Jianbo Shi for further persuading me to come to Penn, and more importantly for lighting the computer vision fire inside of me; something I will keep for the rest of my life. I am also indebted to the rest of my thesis committee: C.J. Taylor for his tenacious and thorough following of both my Written Preliminary Examination II and this thesis, which produced thoughtful comments and corrections. Thanks also to David Forsyth for his warmth, enthusiasm for my research, and inspiring high-level vision.

It is hard to imagine working with any other advisor than Ben Taskar. I am grateful for his unwavering optimism and confidence that things will work out. His hands-on approach, hard work ethic, and adherence to the highest possible quality and rigor are

now an indelible part of my own approach to research. I also cherish our friendship: cracking jokes, playing soccer, enjoying conferences, and knowing that he has my best interests at heart in work and life.

Instrumental in my formative Ph.D. years was Timothee Cour, with whom I cut my teeth on my first serious vision projects, who showed me the way of MATLAB hacking, and how to cut quick to the heart of a problem. Timothee, Alex Toshev and Katerina Fragkiadaki are all equal parts colleagues, confidants and comrades, at equal ease proposing and critiquing research ideas, enjoying drinks, and exploring new cities and countries to the fullest. Thanks to David Weiss for his enthusiasm, deep discussions about life and work, and easy access to cats; best of luck in the future.

I can't thank Mike Felker and Charity Payne enough—Mike has always made me feel that someone is personally looking out for me in the Ph.D. program; Charity makes all administrative tasks seem effortless.

Thanks to BUGS for many enjoyable reading groups and outings—I have especially fond memories of the Poconos retreat and NIPS workshops (special credit due to the masterful organizer Alex Kulesza). Thanks to the rest of the GRASP lab for giving me a home-away-from-home. Thanks to the Automatons, the FC Inter Penn All Stars, and Penn-pickup-soccer for giving me something to look forward to every week. Finally, thanks to Yuri for keeping me grounded to the important things in life, especially laughter.

ABSTRACT

EFFICIENT HUMAN POSE ESTIMATION WITH IMAGE-DEPENDENT INTERACTIONS

Benjamin John Sapp

Ben Taskar

Human pose estimation from 2D images is one of the most challenging and computationally demanding problems in computer vision. Standard models such as Pictorial Structures consider interactions between kinematically connected joints or limbs, leading to inference cost that is quadratic in the number of pixels. As a result, researchers and practitioners have restricted themselves to simple models which only measure the quality of limb-pair possibilities by their 2D geometric plausibility.

In this talk, we propose novel methods which allow for efficient inference in richer models with data-dependent interactions. First, we introduce structured prediction cascades, a structured analog of binary cascaded classifiers, which learn to focus computational effort where it is needed, filtering out many states cheaply while ensuring the correct output is unfiltered. Second, we propose a way to decompose models of human pose with cyclic dependencies into a collection of tree models, and provide novel methods to impose model agreement. Finally, we propose a local linear approach that learns bases centered around modes in the training data, giving us image-dependent local models which are simple and accurate.

These techniques allow for sparse and efficient inference on the order of minutes per image or video clip. As a result, we can afford to model pairwise interaction potentials much more richly with data-dependent features such as contour continuity, segmentation alignment, color consistency, optical flow and multiple modes. We show empirically that these richer models are worthwhile, obtaining significantly more accurate pose estimation on popular datasets.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

> Because it's there.
>
> — George Mallory, on why he wanted to climb Mount Everest.

Why human pose estimation?

The idea of an intelligent robot performing a variety of tasks, extraordinary and mundane, up to and exceeding human performance, has captured the hearts and minds of people since at least the European Renaissance. A key feature of much of this romantic vision is that robots can interact with *us*—working with, around and for humans. An understanding of human pose is a crucial component to making this compelling dream become a reality.

In the more practical and not-too-distant future, understanding human pose from images has enormous potential to help in many computer vision tasks: semantic indexing of images and video (Ferrari et al., 2009), action recognition (Tran et al., 2011), human-object interaction (Yao and Fei-Fei, 2012), and scene understanding (Gupta et al., 2011), to name a few.

The problem of human pose understanding is also interesting in its own right. It defines part of the boundary of what can and cannot be accomplished by artificially intelligent systems. Infants and even other species can understand human pose—why can't a computer?

Pose estimation subsumes one of the holy grails of computer vision: general object recognition. It serves as useful vehicle to demonstrate computer vision techniques that can be used in other subfields. Humans can be considered a collection of related objects (body parts), or a single, highly deformable object. The parts themselves are some of the most difficult to detect in the literature. Typical objects that researchers work on recognizing—faces, bicycles or even potted plants (Everingham et al., 2009)—have distinguishing features, reliable patterns and limited intra-class variability. A body part such as a lower arm, on the other hand, is far more generic. It has a generic shape— at best it can be described as a projection of a cylinder or frustum—and is subject to much higher intra-class variability due to clothing, articulated pose, body type, and severe foreshortening. Features developed must be invariant to pose, lighting, texture and color and still discriminate parts from clutter, or efficient search procedures over these variations need to be developed. These types of techniques are valuable for computer vision in general.

Human pose estimation is also one of the most computationally demanding problems in computer vision, as the set of possible outputs is combinatorial in the number of parts. It can be posed as a graph assignment or graphical model inference problem with an enormous set of possible labels (for each part, determine which pixel it is associated with). This makes it an interesting testbed for advancements in graphical model and matching algorithms for and beyond computer vision—in language understanding, computational biology, statistics and physics.

Finally, the problem of pose estimation is timely. In the computer vision community, statistical machine learning tools and supervised datasets give us principled protocols to learn effective recognition models which didn't exist a decade ago. Robots, cameras, and automated systems are more and more pervasive in everyday life. The demand for reliable

pose estimation is already strong in the entertainment and defense industries.

For all these reasons—practical applications, the dream of artificial intelligence, the general applicability to vision and machine learning, the convergence of technology to make it all possible—human pose estimation is an excellent problem upon which to focus.

1.1 Problem Statement

Here we formalize our problem definition in terms of input, output and computational requirements as follows:

Problem 1 (2D human upper-body pose estimation)**.**

Input: *A single RGB image or RGB video sequence containing the rough location and scale of a person in every frame, with no additional information.*

Output: *Line segments describing the major anatomical parts* {*left and right upper arms, left and right lower arms, torso, head*} *in pixel coordinates.*

Requirements: *Computation time and space polynomial in the number of input pixels and number of output parts.*

Importantly, we concern ourselves only with 2D (two dimensional) pixel array input. This makes the task much more challenging than when using additional sensors, such as in Microsoft's Kinect capture system (Shotton et al., 2011) where depth information and hence reliable knowledge of the background can be used. However, our limited-sensor problem also means it can be applied in more general settings: we can apply such pose estimation methods outdoors and on the wealth of archival images and footages already stored on personal computers, libraries, and photo and video sharing web sites.

Furthermore, we do not assume any additional information, such as knowledge of the foreground, background, clothing, lighting, indoor versus outdoor, etcetera. All these factors work to confound estimation by introducing appearance artifacts. We refer to our

Figure 1.1: An example illustrating the pose estimation problem, formalized in Problem 1.

general setting as pose estimation *in the wild*, to stress the fact that the datasets we consider are from unconstrained foreground and backgrounds settings (or nearly unconstrained, when dealing with TV shows).

Also of note, we only consider the upper body, although all methods and models discussed in this work can be extended to full body processing (*i.e.* including hips and upper and lower legs). In fact, most of the models and tools developed in this work can be applied to other articulated objects, and in general, other domains in which estimating the instantiation of interacting parts (*e.g.*, handwriting recognition, or gene sequencing). We focus on upper body human pose in this work because (1) most interesting pose variation occurs in the upper body, (2) there is a vast amount of data of people's upper bodies from TV shows, movies and images where lower halves are not visible, and (3) there is little extra knowledge to be learned about pose estimation by including the lower body parts, while increasing the computation time of all models at least linearly.

Finally, we restrict ourselves to polynomial running time. The space of all possible poses is exponential in the number of parts. The ideal approach, if computation were not an issue, would be to enumerate all possible poses and score them all using any scoring function of arbitrary complexity. However, this is simply not feasible, and we are forced to

make conditional independence assumptions between certain parts to achieve tractability. In practice, we wish to estimate pose on the order of a few minutes or seconds per frame.

1.2 Intrinsic difficulties

Human pose estimation in the wild is an extremely challenging problem. It shares all of the difficulties of object detection, such as confounding background clutter, lighting, viewpoint, and scale. In addition, there are significant difficulties unique to human poses. We are forced to reason over an enormous number of plausible poses for each image, making this a very computationally demanding problem. In this section we go over the intrinsic difficulties of this problem, both from perceptual and computational standpoints.

1.2.1 Perceptual issues

One of the primary difficulties in human pose estimation is that appearance of pose is largely unconstrained, making it highly variable with multiple appearance modes. The following issues are illustrated in Figure 1.2.

Lighting: Images of pose can be taken indoor or outdoor, making not only the mean intensity of the image variable (signal bias), but also contrast (signal gain). This issue is well studied in computer vision, and to an extent, features have been developed to be invariant to lighting, *e.g.*, HoG (Dalal and Triggs, 2005b), but require harsh quantization and local normalization of edge energy information.

Viewpoint and pose: Humans can look very different depending on where they are with respect to the imaging plane. The global "twist" (rotation about the length-of-body axis) which determines the degree of frontal versus profile stance of the person can be somewhat mitigated by coarse person detectors (Andriluka et al., 2010). However, the body can also go through radical appearance changes due to the articulation of the limbs, forcing practical systems to decompose the modeling into the most basic, articulation-invariant components as atomic units: limbs and joints.

Figure 1.2: Some of the perceptual challenges in human pose estimation. Large variations in lighting, pose, viewpoint, foreshortening, relative scale and clutter all work to confound pose estimation. See §1.2.1

Relative scale: We assume our input is a detected person at a rough global scale. However, we still have a large variation in the scale of parts in two different ways: In any particular person, the ratio of limb lengths may not be consistent; *e.g.* a baby's proportions are very different than an adult's. Across people, there are also a large differences in the geometries of parts, based on gender, body type (fat, skinny, muscular), and age. These further contribute to the variability in appearance.

2D projection: The fact that we are working with images that are projections of the real world lead to further difficulties. Foreshortening makes estimating the length of the limb in 2D coordinates even more difficult, and changes the appearance. Self-occlusions and

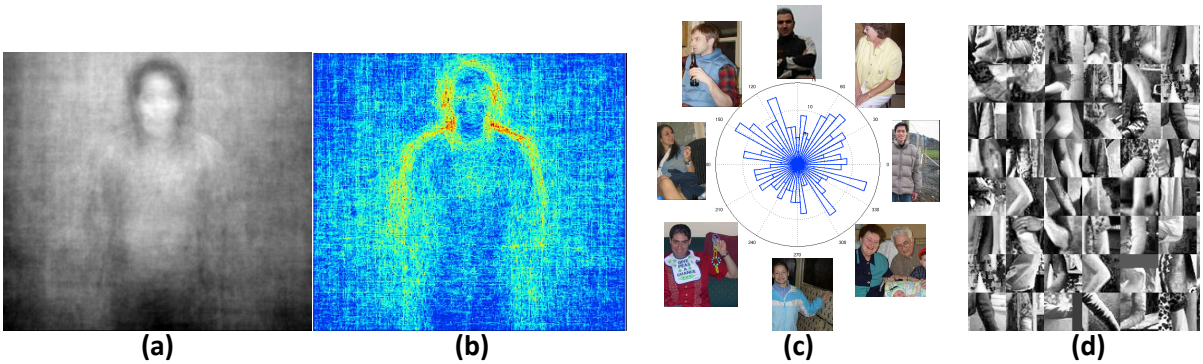


Figure 1.3: Some illustrations of variation in appearance in the PASCAL Stickmen dataset. (a) An average of the dataset in grayscale. (b) Average of Sobel edges over dataset. (c) Polar histogram of the inner angle made between upper and lower arm, with examples for $0°, 45°, 90°, 135°, 180°, 225°, 270°$ and $315°$. (d) A random sampling of 100 left elbows from the Buffy Stickmen pose dataset, removing color and intensity bias, to illustrate the huge variety of appearance due to clutter, motion blur, clothing, body type, and pose.

foreground occlusions make a part invisible and are very hard to determine without further scene or depth knowledge. Finally, it is inherently ambiguous to map from 2D pose to 3D real world coordinates (even up to an unknown global scale factor), discussed further in §3.2. This makes it difficult to model priors on arm length, as we are forced to measure and reason about lengths on the 2D pixel grid.

Clothing: Clothing contributes a near-infinite space of foreground variation. Not only is clothing responsible for foreground clutter, it also can be considered an occluder which hides parts (*e.g.* baggy clothing, skirts, ponchos) and can break assumptions about left-right appearance symmetry (*e.g.* an asymmetric shirt).

Background clutter: Background clutter accounts for roughly half the errors in pose estimation performance. Often it is extremely difficult to separate edges in the background from lower arms. Lower arms appear as little more than a pair of roughly parallel lines in an image, as do many man-made structures and natural objects in backgrounds: walls, tables, chairs, posts, trees, etcetera.

1.2.2 Computational issues

To operationalize Problem 1, let x be the input image pixels, and y be a representation of the output predicted pose. Then a general solution to Problem 1 would take the form of a *scoring function* $s(x, y)$ which evaluates the quality of any estimated pose y in the image x. We can define the "best" pose as the highest scoring: $y^\star = \arg\max_y s(x, y)$ (or $y^\star = \arg\sup_y s(x, y)$ if y is infinite dimensional, *i.e.* continuous). Then Problem 1 is satisfied if this determination of the maximizer can be done in polynomial time. There are two sources of intrinsic computational complexity within this framework.

Complexity of the input: For the reasons outlined in §1.2.1, there are an astronomical number of different inputs x that can map to the same true pose y—the same layout of body parts can look very different from image to image. The problem is inherently multimodal, in the sense that radically different appearances are equally valid input representations of any particular pose. For a few different illustrations of the variability of a dataset, see Figure 1.3.

To deal with this complex problem, we are forced to either design features that are invariant to the multimodality (*e.g.*, a generic patch-based arm detector based on coarse edges, or geometric features based on relative part coordinate systems), or to partition the space and model multiple modes separately. In the case of the latter approach, we are faced with other difficult decisions regarding model complexity: how to define modes and a notion of locality, and what the right trade-off is between the richness of the model and the error in fitting the model at different modes with a finite amount of training data.

Complexity of the output: The enormous combinatorial space of possible output poses is a second source of computational complexity. A typical discretization of the state space of human poses is an 80×80 spatial grid of part locations at 24 possible angles (Felzenszwalb and Huttenlocher, 2005), resulting in an output space that is roughly 150000 possibilities for each part, and thus $150000^6 \approx 10^{30}$ for the joint output space of all 6 upper body parts [1].

[1] In the case of continuous spaces, the output space is infinite (infinitely precise), and to maintain

Enumerating all possibilities for a joint configuration of all parts is clearly not feasible. At the other extreme, we could ignore part interactions, and estimate the pose of each part separately—a task which instead has 6×150000 possibilities, which is computationally very cheap with modern computers. However, individual part detection is extremely difficult for body parts (Andriluka et al., 2009) due to the wide range of appearances and lack of discriminating features.

Between the two extremes of (1) estimating parts in isolation and (2) enumerating all possible joint pose configurations, there lies a family of models $s(x, y)$ that consider *some* part interactions, but not all. The simplest of these is a *first-order*, or *pairwise model*, which looks at pairs of part interactions at a time, and the graph of part interactions forms a tree structure. This compromise between a full model of every part and a decoupled model of independent parts will be the basic model building block throughout this work.

In such a pairwise model, the basic bottleneck operation is to evaluate the quality of a pair of parts at a time. The model combines all such pairwise scores together to determine the optimal global pose. This scoring requires $150000^2 \approx 1$ billion possibilities to consider for a pair of parts in our example $80 \times 80 \times 24$ state space, which is large but just small enough for modern machines to handle with some additional model restrictions, detailed in §3.

1.3 Contributions of this thesis

Due to the computational issues discussed above, previous work in pose estimation has resorted to a model of pose that considers, at most, pairwise interactions between parts, in a specially restricted form: the network of part-pair interactions is described by a tree structured graph, and the interactions are described by simple kinematic consistency. This is known as the basic *pictorial structure* (PS) model, also referred to as a *spring model*—see §3, and Figure 1.5.

tractability the form of $s(x, y)$ is typically analytical with a closed-form maximum, mean and/or mode, or approximate sampling techniques are used for inference. See §4.

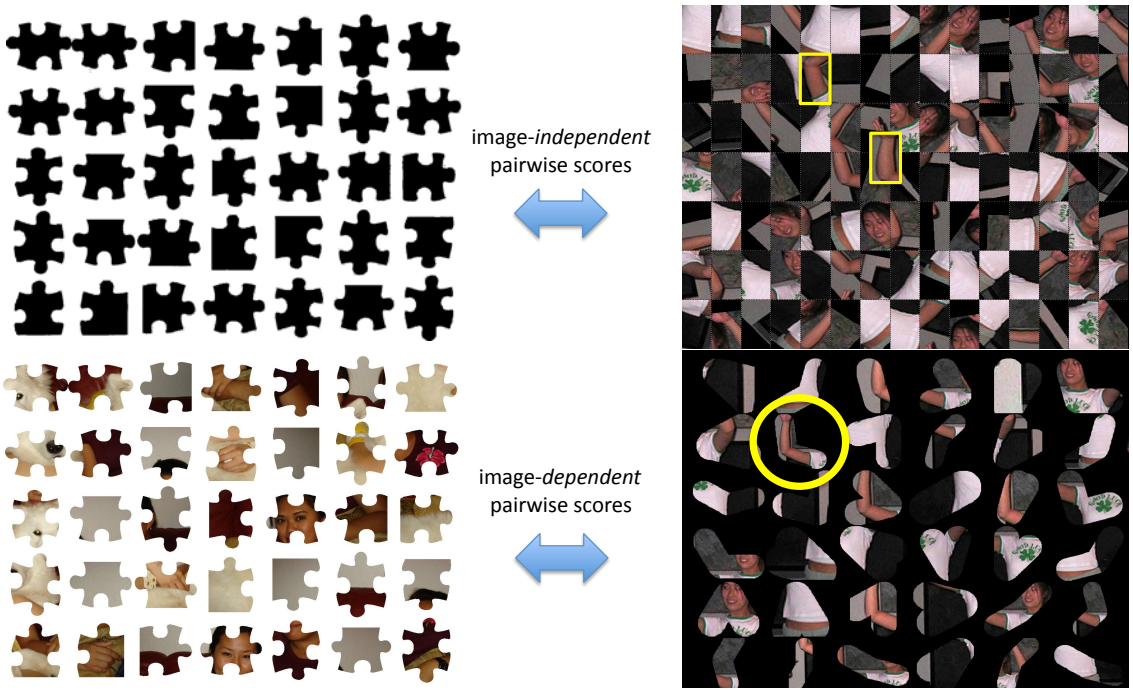


Figure 1.4: Puzzle analogy of pose estimation. Classical approaches only use individual part detectors and geometric plausibility to determine the pose of a person. This is analogous to attempting to put together a puzzle without looking at appearance of the pieces —only the plausibility of them fitting together. On the other hand, models with data-dependent interactions are analogous to using the appearance of the puzzle piece faces as well as their fit when constructing the puzzle. Even for humans, it is easier to spot the correct pair of upper/lower arms when they are examined jointly.

In PS models, each part has an individual score for placement at any location in the image, which must be balanced with part-pair penalties for deforming from the default model positions ("pose prior"). For example, the deformation penalty between an upper arm and lower arm expresses the fact that they should roughly agree on where the elbow is. The deformation penalties can be thought of as springs at rest at default positions and stretched by moving the parts to new positions. The parts themselves are attracted individually to likely spots in the image. Thus a balance is sought between individual part

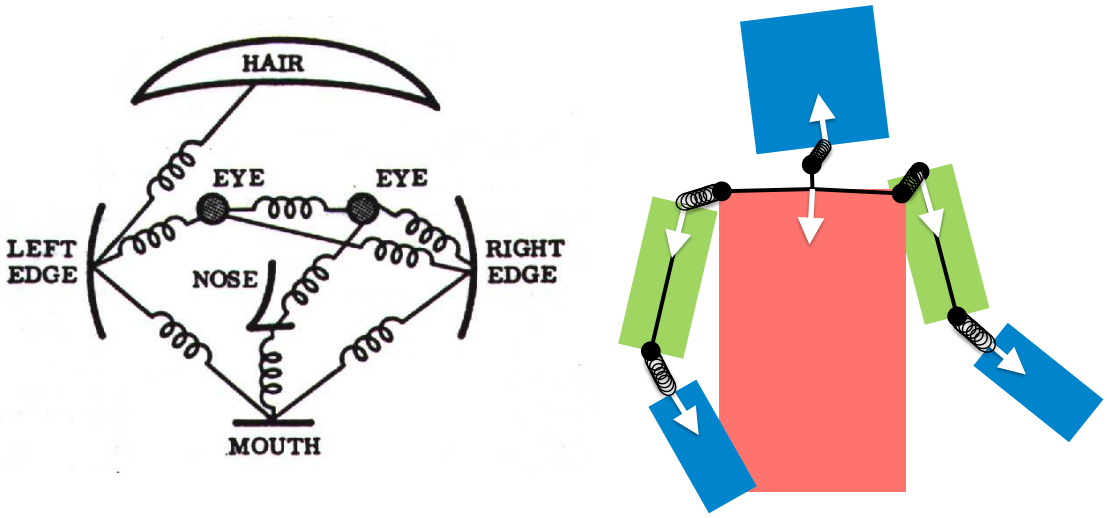


Figure 1.5: Spring model of pose. At left, the original spring pictorial structure model that appeared in Fischler and Elschlager (1973). At right, the standard PS model for 2D human pose. The states are shown as unit vectors indicating the position of joints and their direction. The mean displacement between joints are shown as solid black circles, connected by solid black lines to show the kinematic tree structure. The displacement from mean positions are shown as springs stretching. This figure is repeated again for convenience in §3.

beliefs and respecting the prior notion of what a pose should look like.

An important property of PS models is that the pairwise, spring stretch, termsare *blind to the image content*. The problem with this is that individual part detector scores are extremely weak (§1.2.1): they must work in isolation and generalize to limbs in all settings of backgrounds, foregrounds, articulation, and environment.

As a result, the above model is effectively trying to piece together parts of a person, when the parts themselves are extremely ambiguous. As an extreme analogy of this, it is similar to attempting to put together a jigsaw puzzle in which the pieces themselves are very generic. One has many plausible candidates for where each puzzle piece should go individually (like individual part scores), but to determine whether two pieces are adjacent,

one can only see how well they fit together, and *not* the image content on their faces; see Figure 1.4-top row.

1.3.1 Image-dependent interactions in a tree structured model, §5

As an improvement over the basic PS model, we wish to actually *exploit* image content when modeling pairwise interactions. In the puzzle analogy, this would allow us to fit pieces together based on their color similarity and continuous contours across the connection boundary; see Figure 1.4-bottom row. We exploit these same cues for determining whether limbs go together in an image, as well as additional cues such as region support and multimodal descriptions of geometry.

Unfortunately, this turns out to be computationally infeasible using standard tools and techniques. In light of this, we propose a *cascade* of models to focus computation on pose possibilities that are more promising. There are many pose possibilities that are easy to reject as incorrect with a simple model (like a basic PS model, or even simpler), and we are then able to freely apply a richer model on the possibilities that remain. This is illustrated in Figure 5.4.

To employ a cascade approach, we develop and analyze *structured prediction cascades*, and apply them to the problem of 2D pose estimation. Importantly, we provide a novel training objective for the cascade so that parameters of the models are learned to specifically to filter out a significant proportion of possibilities at every cascade level.

1.3.2 Image-dependent interactions in a general graph, §6

The cascade approach we proposed in the previous section works for any tree-structured model. However, in any tree model, we fail to capture important interactions between parts, both *within* a single frame (*e.g.*, color constancy between left/right symmetric parts to model clothing) and *across* frames when dealing with tracking multiple parts and their interactions over time in video. Determining the best possible answer $\arg\max_y s(x, y)$

over a general cyclic graph of part relationships is known to be $\#P$-hard (Koller and Friedman, 2009)—exponential in the number of frames of video. We provide an approximate approach which decomposes a cyclic model of pose into a collection of subgraph trees, whose union of edges covers all the relationships we care to model. This allows us to exploit all the interesting interaction terms in the original model with efficient inference in each subtree, thanks to the structure and the use of our cascade approach. Then, we can exploit all cues used in the tree-based model, and in addition, cues based on color symmetry across the body, and temporal appearance and location persistence information. We propose and investigate empirically different methods of reaching a consensus between the subtrees.

We evaluate our approach on a new video dataset, the first of its kind in tracking human pose in the wild without any assumed extra knowledge. We show our proposed model and approximation scheme is beneficial, beating the state-of-the-art in pose estimation systems.

1.3.3 Multimodal interactions, §7

The aforementioned models focus attention on increasing the quality of features and increasing the number of modeled part interactions. The hope is that these more expressive models do a better job at capturing the inherently multimodal appearance space of poses, by better separating the true pose configurations from false alarms. This somewhat addresses the issue of non-linearity in lower-dimensional feature spaces, *e.g.*, using only edge information.

Complementary to the models described in the previous sections, we propose to capture this nonlinearity directly with an explicitly multimodal model. Now the goal is to determine not only the best pose layout, but also which *mode* the pose belongs to. Each mode need only model a portion of the pose space. Instead of fitting the parameters of one monolithic model to cover all possible modes, here we learn separate parameters for each mode, allowing us to learn more precise descriptions of appearance and geometry,

for, *e.g.*, an arms crossed mode, an arms raised mode, etcetera.

1.3.4 Technical summary of models

This section is intended for readers who already have an understanding of pairwise structured models and the parts-based pose estimation literature. It provides a quick overview of the model formulations discussed in depth in the rest of this thesis. It does not give a self-contained explanation, and the unfamiliar reader is encouraged to skip this section.

The basic pictorial structure model takes the form

$$s(x, y) = \sum_{i \in \mathcal{V}_\Upsilon} \phi_i(x, y_i) + \sum_{i,j \in \mathcal{E}_\Upsilon} \phi_{ij}(y_i - y_j)$$

where the network of part pairwise interactions is described by a tree structured graph $\Upsilon = (\mathcal{V}_\Upsilon, \mathcal{E}_\Upsilon)$. The first terms $\phi_i(x, y_i)$ score how likely a part is to be placed at location y_i, independent of other parts. The pairwise terms $\phi_{ij}(y_i - y_j)$ score the geometric compatibility between pairs of parts. Importantly, this pairwise term is *blind to the image content*.

Instead, we wish to actually *exploit* image content when modeling pairwise interactions. To this effect, in §5 we propose a more general model of human pose, of the form

$$\boxed{s(x, y) = \sum_{i \in \mathcal{V}_\Upsilon} \phi_i(x, y_i) + \sum_{i,j \in \mathcal{E}_\Upsilon} \phi_{ij}(x, y_i, y_j)} \tag{1.1}$$

This turns out to be computationally infeasible using standard tools and techniques. We develop and analyze structured prediction cascades as a tool to deal with this. The cascade works by progressively filtering the state space of a sequence of models with coarse-to-fine resolution. We pose a learning objective so that the models prune both aggressively and accurately at test time.

The model we propose Equation 1.1 works for any tree-structured model, but cannot be applied to cyclic models. We address this with a generalization of Equation 1.1:

$$\boxed{s(x,y) = \sum_{i \in \mathcal{V}_G} \phi_i(x, y_i) + \sum_{i,j \in \mathcal{E}_G} \phi_{ij}(x, y_i, y_j)} \tag{1.2}$$

where $G = (\mathcal{V}_G, \mathcal{E}_G)$ is a general graph, not just a tree. Determining the best possible answer $\arg\max_y s(x,y)$ over a general graph G is known to be $\#P$-hard (Koller and Friedman, 2009)—exponential in the number of frames of video. To deal with this, we explore state-of-the-art approximation techniques, such as dual decomposition, and propose a simpler approximation technique the performs at least as well with orders of magnitude less computation. See §6.

The aforementioned models focus attention on increasing the quality of features and increasing the number of modeled part interactions. The hope is that these more expressive models do a better job at capturing the inherently multimodal appearance space of poses, by better separating the true pose configurations from false alarms. This somewhat addresses the issue of non-linearity in lower-dimensional feature spaces, *e.g.*, using only edge information.

Complementary to the models described by Equation 1.1 and Equation 1.2, in §7 we propose to capture this nonlinearity directly as follows:

$$\boxed{s(x,y,z) = \sum_{i \in \mathcal{V}_G} \phi_i(x, y_i, z) + \sum_{i,j \in \mathcal{E}_G} \phi_{ij}(x, y_i, y_j, z)} \tag{1.3}$$

where now the goal is to determine not only the best pose layout y, but also which *mode* z the pose belongs to: $y^\star, z^\star = \arg\max_{y,z} s(x,y,z)$. Each mode need only model a portion of the pose space. Instead of fitting the parameters of one monolithic model to cover all possible modes, here we learn separate parameters for each mode, allowing us to learn more precise descriptions of appearance and geometry.

1.3.5 Summary of contributions

In summary, the work detailed in this thesis contributes the following to the fields of machine learning and computer vision, and especially their intersection for human pose estimation:

- New models of human pose that capture image-dependent interactions.
- Computational innovations that enable learning and inference in these models, which are naïvely intractable: structured cascades and tree ensemble methods.
- A variety of new features and feature types not typically applied to pose estimation. Some of these are bottom-up type features complementary to the traditional edge-based cues.
- State-of-the-art results on the public Buffy and Pascal Stickmen single frame datasets, and our introduced MoviePose single frame and VideoPose video sequence datasets.

1.3.6 Published work supporting this thesis

The Cascaded Pictorial Structure model (§5) first appeared in Sapp et al. (2010b) and introduced the concepts of a coarse-to-fine cascade that allows one to use arbitrary features efficiently. This extended the original Structured Prediction Cascades approach from Weiss and Taskar (2010), with a journal version under review—Weiss et al. (2012). Using the cascade approach for an ensemble of models was developed in Sapp et al. (2010c). This idea was then extended beyond cascade filtering for prediction in the Ensembles of Stretchable Models framework (§6) for pose estimation in video in Sapp et al. (2011). The complementary non-parametric approach of Local Linear Pictorial Structures (§7) is currently to be submitted—Sapp and Taskar (2012).

Part I

Preliminaries

Chapter 2

Structured Prediction

In this section we lay out the basic definitions and tools necessary for data-driven modeling of *classification problems*. In the most general setting, the goal is to learn a mapping, or *classifier*, h in a hypothesis class $\mathcal{H}$ from a set of input examples x to a set of discrete output variables y. We concern ourselves in this work with a specific setting: x lies in a (possibly high-dimensional) vector space (most commonly in this work, the input image pixels) and the target lies in a discrete n-dimensional space:

$$y \in \{0, 1, \ldots, k\}^n \triangleq \mathcal{Y}.$$

We will refer to the set $\mathcal{Y} = \{0, 1, \ldots, k\}^n$ as the set of labels, *label set*, or *state space* for a dimension of y. The i^{th} dimension of y will be indexed y_i, and referred to as *variable* i. We refer to the state space of part i as $\mathcal{Y}_i$, and the size of the state space as $|\mathcal{Y}_i|$. The size of the full state space is $|\mathcal{Y}| = |\mathcal{Y}_1 \times \ldots \times \mathcal{Y}_n|$, where here $\times$ denotes a Cartesian product.

For example, in a simple image classification task to determine whether an image has a dog in it, x could be an image's pixels, and $y \in \{dog, not\ dog\} \cong \{0, 1\}$; an instance of *binary classification*. In this case, $k = 2$ and $n = 1$. In *multiclass classification*, $k > 2$ and $n = 1$, *e.g.*, predicting the handwritten digits $0, \ldots, 9$ or the weather $\{sunny, cloudy, rainy\} \cong \{0, 1, 2\}$. Finally, in *structured prediction*, the output y is a vector: $n > 1$.

We will discuss four major components necessary for learning and applying machine learning classifiers in this chapter:

- An **inference procedure** to determine the most likely label for a fixed test example x:

$$h(x) \triangleq \arg\max_{y \in \mathcal{Y}} h(x, y). \tag{2.1}$$

- The **hypothesis class** $\mathcal{H}$ from which to obtain our classifier h.

- A **learning loss function** $\mathcal{L}(\cdot)$ which assesses the quality of any particular classifier $h \in \mathcal{H}$.

- A **learning algorithm** to find the minimizer of $\mathcal{L}$:

$$h = \arg\min_{h' \in \mathcal{H}} \mathcal{L}(h'). \tag{2.2}$$

2.1 Generalized linear classifiers

There are many possible parametric and non-parametric choices of hypothesis classes $\mathcal{H}$ to consider. One of the easiest to represent, learn, visualize and analyze is the hypothesis class of *generalized linear models*, of the form

$$h(x, y) = \sum_{i=1}^{d} w_i f_i(x, y) \ = \mathbf{w} \cdot \mathbf{f}(x, y) \tag{2.3}$$

where $\mathbf{w} \in \mathbb{R}^d$ is a vector of linear parameters of our model and $\mathbf{f}(x, y) \in \mathbb{R}^d$ is a vector of features that depend on the input and output variables. Thus the model is simply a weighted sum of features to obtain a real-valued score, and $\mathcal{H} = \mathbb{R}^d$.

2.1.1 Binary and multi-class classifiers

In binary ($y \in \{0, 1\}$) and multi-class ($y \in \{0, \ldots, k\}$) classification problems, there is a simple and intuitive geometric interpretation to linear classifiers. In two dimensions, the

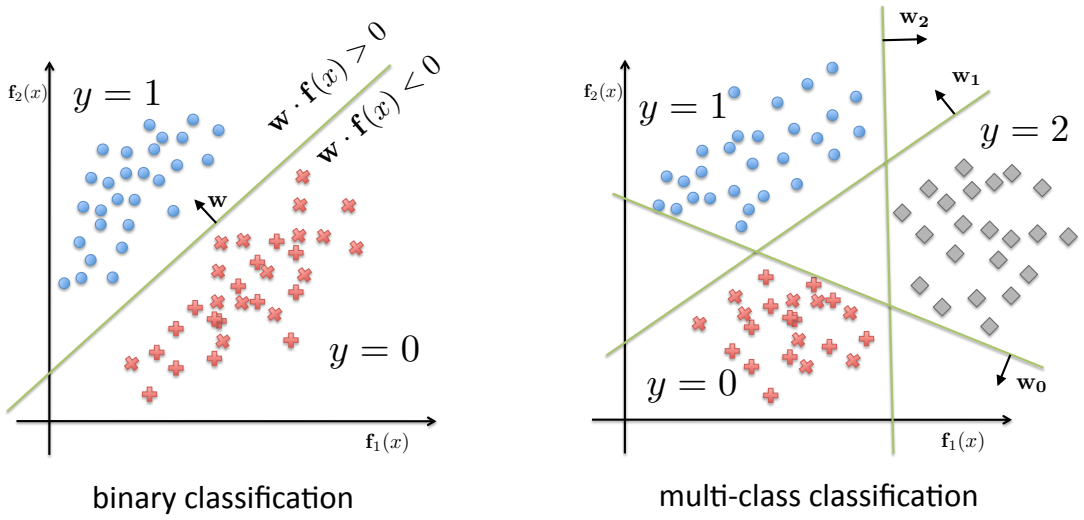


Figure 2.1: Binary and multi-class classification. On the left is shown a binary labeling problem in 2 dimensions, with a linear classifier which perfectly classifies the data. On the right is shown a 3 class classification problem, with a linear classifier for each class, each of which perfectly separates the correct label from the two incorrect labels.

classifiers describe a line, in three dimensions, they describe a plane, and in general, they describe a *hyperplane* which partitions the feature space (for a particular label) into two halfspaces.

Observe that for binary classification, we need only look at which side of a hyperplane a point lies on in a particular feature space to determine the label:

$$h(x) = \mathbf{1}\left(h(x,1) > h(x,0)\right) \tag{2.4}$$

$$= \mathbf{1}\left(\mathbf{w} \cdot \mathbf{f}(x,1) > \mathbf{w} \cdot \mathbf{f}(x,0)\right) \tag{2.5}$$

$$= \mathbf{w} \cdot \left(f(x,1) - f(x,0)\right) > 0 \tag{2.6}$$

$$= \mathbf{w} \cdot \tilde{f}(x) > 0 \tag{2.7}$$

The vector $\mathbf{w}$ describes a hyperplane via its surface normal. The test $\mathbf{w} \cdot \tilde{f}(x) > 0$ determines which side of the hyperplane an example lies on, which also corresponds to its predicted label, as in Figure 2.1-left. When $k > 2$, we can also transform feature

spaces and interpret the linear classifier as k separating hyperplanes in d dimensions, one hyperplane to separate each label from the rest. Figure 2.1-right illustrates an example with $k = 3$ and $d = 2$.

2.2 Pairwise structured models

In structured prediction, the output space $|\mathcal{Y}| = |\{0, \ldots, k\}^n|$ is exponential in n and typically enormous. Due to computational and modeling considerations to be discussed, we assume that the full model $\mathbf{w} \cdot \mathbf{f}(x, y)$ *decomposes* into *factors*, or *cliques* which involve overlapping sets of variables:

$$\mathbf{w} \cdot \mathbf{f}(x, y) = \sum_{c \in \mathcal{C}} \mathbf{w}_c \cdot \mathbf{f}_c(x, y_c) = \sum_{c \in \mathcal{C}} \phi_c(x, y_c), \tag{2.8}$$

where we use the shorthand for factors $\phi_c(\cdot) \triangleq \mathbf{w}_c \cdot \mathbf{f}(\cdot)$. The index c represents a set of components of y and $\mathbf{f}$, *e.g.*, $y_c = \{y_1, y_2, y_3\}$. The set $\mathcal{C}$ is the set of all factors c in our model. We can encode the sets of factors and their overlap relations in general using a *factor graph* (Koller and Friedman, 2009). The number of different settings and representations for structured models is vast and varied and most are outside the scope of this work.

For our purposes, we focus on structured problems with at most *pairwise structure*, where $|c| \leq 2 \quad \forall c \in \mathcal{C}$[1]. We can represent a pairwise structured model as a graph $G = (\mathcal{V}_G, \mathcal{E}_G)$ where each variable y_i corresponds to a vertex in the graph's vertex set $\mathcal{V}_G$, and each edge in $\mathcal{E}_G$ corresponds to two variables y_i, y_j being involved in a pairwise factor

[1]This simplification does not result in loss of generality, as factors involving 3 or more variables can always be converted into pairwise or unary factors with an expanded label set consisting of the Cartesian product of each variable's state space. For example, a factor involving the variables y_1, y_2 and y_3 with state space $\{1, \ldots, k\}^3$ can be converted into a single variable $y_{123} \in \{1, \ldots, k^3\}$. This transformation is, however, exponential in $|c|$.

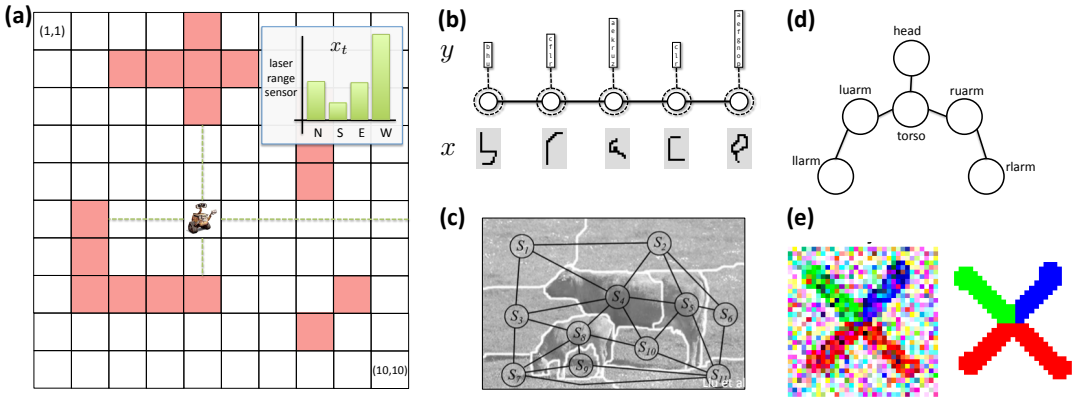


Figure 2.2: MRF examples. (a) A robot localization problem on a 10×10 grid. (b) Handwriting recognition. (c) Scene labeling. (d) Human pose estimation. (e) Image denoising. See text for details.

ϕ_{ij}. We can then decompose our classifier into the special form

$$\mathbf{w} \cdot \mathbf{f}(x, y) = \sum_{i \in \mathcal{V}_G} \mathbf{w}_i \cdot \mathbf{f}(x, y_i) + \sum_{i,j \in \mathcal{E}_G} \mathbf{w}_{ij} \cdot \mathbf{f}(x, y_i, y_j) \tag{2.9}$$

$$= \sum_{i \in \mathcal{V}_G} \phi_i + \sum_{i,j \in \mathcal{E}_G} \phi_{ij}. \tag{2.10}$$

There is a huge amount of research dedicated to models of this form originally stemming from statistical mechanics, where it was first used to determine the spin of particles arranged in a grid. From this perspective, Equation 2.10 describes the log of the energy of the particle system. When given a probabilistic interpretation (as we will see in §2.2.1) this type of model is known as a *pairwise Markov Random Field* (MRF). Because of such historical intuitions as a model for energy, we refer to the different terms as *potentials*, and Equation 2.10 as the negative of an *energy function* which we seek to minimize. Terms of the form $\phi_i = \mathbf{w}_i \cdot \mathbf{f}_i$ we will refer to as *unary potentials*; $\phi_{ij} = \mathbf{w}_{ij} \cdot \mathbf{f}_{ij}$ are *pairwise potentials*.

Pairwise MRF examples

Our primary application of interest in this work is human pose estimation, whose modeling as an MRF will be discussed in detail in §3. However, to first give motivation and intuition about how and why to model problems via a pairwise MRF, we present a few vision- and robotics-related examples here.

Wandering robot Imagine we are tracking the location of a robot on a map with 100 locations over 100 time steps as in Figure 2.2-a. We have no idea what the robot's intent is or where he starts out in the world (the "kidnapped robot" scenario), only that he is restricted to moving adjacent grid positions at each time step (*i.e.*, cannot teleport). We have noisy sensor readings $x = [x_1, \ldots, x_{100}]$ for every time step. The output y is a sequence of locations the robot visited in all 100 time steps, one out of $\mathcal{Y} = 100^{100} = 10^{1000}$ possibilities. By comparison, the estimated number of atoms in the universe is only 10^{80}. A naïve approach would be to model this as a multi-class problem with 10^{1000} labels, and try to directly learn a mapping $h(x, y) = \mathbf{w} \cdot \mathbf{f}(x, y)$ for all possible y. Importantly, there is a lot of *structure* to this problem, the most obvious being that many outputs y are just not valid, due to the fact that the robot can only move to an adjacent location at each step.

The key insight to be made is the following: knowing where the robot is at time t is tremendously useful for determining the robot's next location at $t + 1$—the robot must be somewhere in the vicinity. In addition, knowing where the robot is at $t - 1$ also helps localize it at $t + 1$, but with diminishing returns—we now know its previous velocity. Going further into the past, there is increasingly less information to help localize where the robot is at time $t + 1$. In general, we can make the simplifying assumption that *given the recent past, the future is independent of the distant past*. Using this intuition, we can model the problem as a *chain* MRF:

$$\mathbf{w} \cdot \mathbf{f}(x, y) = \sum_{t=1}^{100} \phi_t(x, y_t) + \sum_{t=1}^{99} \phi_{t,t+1}(y_t, y_{t+1}) \tag{2.11}$$

where the graph structure is a node for the output variable at each time step y_t, and we

only consider pairwise factors over adjacent time steps $\phi_{t,t+1}$, creating a chain of variable interactions. The unary potentials ϕ_t can model the likelihood of being in each grid location at time t based on sensor readings, and the pairwise potentials can model how likely it is to transition from any location y_t to any location y_{t+1}.

The pairwise term in such tracking problems is typically represented as a *transition matrix*. In this robot localization problem the transition matrix would be very sparse, since any location can only transition to adjacent grid locations. It might be comprised of the values $\{-\infty, 0\}$ indicating which transitions are or aren't possible, or more generally, log-probabilities of how likely different transitions are: $\phi_{t,t+1}(y_t, y_{t+1}) \propto \log p(y_{t+1}|y_t)$.

In other structured problems, it also makes sense to make *independence assumptions* about which dimensions of y are assumed dependent on each other and their interactions should be directly modeled. For example:

Handwriting recognition In this problem, the output at each position is which letter of the alphabet is written given handdrawn letter images x (in this problem, $k = 26$ if only considering 'a',...,'z', or roughly 70 if considering all alphanumerics plus punctuation). Clearly adjacent letters in a document are highly correlated (*e.g.*, adjacent outputs such as 'ab' and 'he' are likely, but 'hb' is not), but depend very little on letters far away in the word, sentence or even document. Practitioners again typically model this as a chain, seen Figure 2.2-b.

Scene labeling In scene labeling, the goal is to label coarse segments of an image with scene types ("sky","grass","building","cow", etcetera; k is on the order of tens of labels) given the image as x. The typical assumption made is that adjacent regions' labels directly depend on each other, whereas the label of spatially distant segments have weak or no interactions and are not modeled (Cour et al., 2005). The MRF model thus connects adjacent segments in the image, giving us a cyclic planar graph as in Figure 2.2-c.

Human pose estimation Knowing the location of the left shoulder is a strong cue for where the left elbow should be, since they are kinematically coupled in the real world, but only a weak indicator of where the right wrist should be. The typical representation is to

describe the human layout as a tree graph corresponding to the kinematic skeleton, see Figure 2.2-d and §3.

Image denoising Here the input x is an image with a small set of labels that are corrupted by noise; the goal is to determine the uncorrupted original image y the same size as x. In this setting k is typically 2 or a small set of indexed colors. Practitioners typically model this problem with a grid graph, where variable $y_{(r,c)}$ corresponds to pixel label at row r, column c, and is in pairwise factors with $y_{(r+1,c)}$, $y_{(r-1,c)}$, $y_{(r,c+1)}$, and $y_{(r,c-1)}$. Pairwise terms model the assumption that adjacent pixels are likely to have the same label, *e.g.*, $\phi_{(r,c),(r+1,c)} \propto \mathbf{1}\left[y_{(r,c)} = y_{(r+1,c)}\right]$. See Figure 2.2-e.

You may object that some of the decomposition assumptions made in the above examples are somewhat extreme. However, they are typically seen as forgivable thanks to the greater simplicity of modeling only local interactions. Even more enticing is the reduction in computation they allow, discussed in §2.3.

2.2.1 Probabilistic interpretation

Much of the development of machine learning models and methods originally came from the probabilistic modeling and statistics literature (Bishop, 2006; Friedman et al., 2001; Koller and Friedman, 2009). From this perspective, our model describes a *log-linear* probabilistic form of the posterior probability $p(y|x)$:

$$p(y|x) = \frac{\exp\left[\mathbf{w} \cdot \mathbf{f}(x, y)\right]}{\sum_{y \in \mathcal{Y}} \exp\left[\mathbf{w} \cdot f(x, y)\right]} = \frac{1}{Z(x)} \exp\left[\mathbf{w} \cdot \mathbf{f}(x, y)\right] \tag{2.12}$$

It can be shown that this particular form of $p(y|x)$ is the distribution with maximum entropy, subject to the constraints that feature expectations with respect to $p(y|x)$ match empirical feature expectations in a training set (Jaynes, 1963). This principle is based on the desire to have our model make as few assumptions as possible (be most entropic) about the observed data.

Because we are typically only interested in the most likely y—here, the *maximum a posteriori* or MAP assignment—it is sufficient to find the $\arg\max$ of a simplified quantity

and not worry about normalization by $Z(x)$:

$$\arg\max_y p(y|x) \tag{2.13}$$

$$= \arg\max_y \log p(y|x) \tag{2.14}$$

$$= \arg\max_y \mathbf{w} \cdot \mathbf{f}(x, y) - \log Z(x) \tag{2.15}$$

$$= \arg\max \mathbf{w} \cdot \mathbf{f}(x, y) \tag{2.16}$$

A distribution $p(y|x)$ modeling a structured y that decomposes over factors as in §2.2 takes the form:

$$p(y|x) \propto \exp\left[\sum_{i \in \mathcal{V}_G} \phi_i + \sum_{i,j \in \mathcal{E}_G} \phi_{ij}\right] = \prod_{i \in \mathcal{V}_G} \exp \phi_i \prod_{i,j \in \mathcal{E}_G} \exp \phi_{ij} \tag{2.17}$$

Historically and in other settings, terms $\exp \phi_c$ were assumed to be themselves proper joint $p(y_c, x)$ or posterior $p(y_c|x)$ distributions over subsets of random variables c, but we assume no such restriction in our case.

Sometimes people make distinctions between fitting parameters for a *generative model* of the form $p(x, y)$ versus a *discriminative model* of the form by referring to the former as a Markov Random Field, and the latter as a *Conditional Random Field* (CRF) (Lafferty et al., 2001) or *Discriminative Random Field* (Kumar and Hebert, 2003). The differences between MRFs and CRFs lie in the way in which models of the form Equation 2.17 are trained, and restrictions on the form of the potentials allowed.

In all our work, we assume no restrictions on the form of potentials, always assume x is given and y is to be estimated, and learn a discriminative model, like a CRF (see §2.5). However, we never use a probabilistic interpretation of our model nor compute the distribution normalizing constant. We prefer to use the generic term MRF for the rest of the paper, under the view that all these models are simply general linear models in a high dimensional space, with decomposable structure. At test time, the inference algorithms are the same.

2.3 Inference

The inference problem is to determine the highest scoring possibility out of all possible outputs in $\mathcal{Y}$:

$$y^\star = h(x) = \arg\max_{y \in \mathcal{Y}} h(x, y). \tag{2.18}$$

We use the convention h to mean *hypothesis* or *hypothesis class*, originating from the machine learning community. In the spirit of optimization, we also view this as a *scoring function* and use the synonymous notation s to denote the score:

$$s(x, y) \triangleq h(x, y) \tag{2.19}$$

When $\mathcal{Y}$ is low-dimensional, brute-force search is easy enough: evaluate $s(x, y)$ for all possible y, and return the highest scoring. This is possible for both binary and multi-class classification, where we evaluate up to k possible labels to determine the highest scoring.

However, in the most general setting, the size of $\mathcal{Y}$ is exponential in k: $|\mathcal{Y}| = k^n$, and we can't hope to enumerate all possible outputs in $\mathcal{Y}$ to find the highest-scoring. Thankfully, the decomposable structure we impose on our models, discussed in §2.2, allows us to find a solution time polynomial in k and linear in n.

As a simple example of this, consider our wandering robot problem (§2.2) where we make the common assumption that we only model temporally adjacent pairs of output variables together, making a chain of variable dependencies:

$$\mathbf{w} \cdot \mathbf{f}(x, y) = \sum_{t=1}^{100} \phi_t(x, y_t) + \sum_{t=1}^{99} \phi_{t,t+1}(x, y_t, y_{t+1})$$

Observe now how computing the maximum score of the classifier *also decomposes* for this simple problem:

$$\max_{y \in \mathcal{Y}} \mathbf{w} \cdot \mathbf{f}(x, y) = \max_{y_1, \ldots, y_{100}} \sum_{t=1}^{100} \phi_t(x, y_t) + \sum_{t=1}^{99} \phi_{t,t+1}(x, y_t, y_{t+1}) \tag{2.20}$$

$$= \max_{y_{100}} \left[\phi_{99,100} + \phi_{100} + \ldots \max_{y_3} \left[\phi_{3,4} + \phi_3 + \max_{y_2} \left[\phi_{2,3} + \phi_2 + \max_{y_1} \phi_{1,2} + \phi_1 \right] \right] \ldots \right] \tag{2.21}$$

In Equation 2.21, we see that the max over each dimension of y can be nested and needs only reason over at largest a pairwise factor at a time. In our robot problem, each pairwise factor is only 100^2 numbers to consider and each unary factor is of size 100. Thus the total amount of numbers that need to be examined in this problem is roughly 10^6, an astronomical improvement over the complete space of 10^{1000}. In general the bottleneck of inference is the max over the pairwise factors having size k^2, and thus inference is $O(n \cdot k^2)$.

When an MRF model forms a tree, we can always exploit the structure in a similar way for exact efficient inference (Koller and Friedman, 2009). Using the same trick we used in the specific example of the robot localization problem Equation 2.21, we can perform inference by nesting the $\max$ operator over pairs of variables at a time, as well as storing the $\arg\max$ maximizer of each $\max$ operation. For completeness, the algorithm is in Algorithm 1, commonly referred to as max-sum *message passing, belief propagation*, or *viterbi decoding*. The algorithm results in inference that is $O(n \cdot k^2)$ computation rather than k^n.

When the MRF interactions do *not* form a tree, inference becomes $\#P$-hard. For example, if we could determine the maximizer in a loopy MRF, we could in polynomial time transform it to determine how many variable assignments satisfy a given 3-SAT formula (Koller and Friedman, 2009; Valiant, 1979). Thus, no known algorithm exists in general for inference that is polynomial in the number of variables. As a sanity check, Algorithm 1 breaks down for loopy graphs: we have no topological ordering of the network for which we can pass belief messages up to a root, and then backtrack in reverse to obtain the best solution.

2.4 Max-marginals

Often we are only interested in the best scoring complete assignment $y^\star = \arg\max_y s(x, y)$ and its corresponding score $s^\star(x) = \max_y s(x, y)$, both obtainable from Algorithm 1.

Algorithm 1: Max-sum message passing to solve

$$\arg\max_{y \in \mathcal{Y}} s(x, y) = \arg\max_{y \in \mathcal{Y}} \sum_{i \in \mathcal{V}} \phi_i + \sum_{ij \in \mathcal{E}} \phi_{ij}$$

Input: Factors $\{\phi_i\}, \{\phi_{ij}\}$, tree graph $G = (\mathcal{V}, \mathcal{E})$ with (arbitrary) root node index r and topological ordering π, where $\pi_n = r$.

Output: $y^\star = \arg\max_y s(x, y)$

for $i = \pi_1, \pi_2, \ldots, \pi_n$ **do**
- $m_i = \phi_i + \sum_{j \in \text{kids}(i)} m_{j \to i}$
- **if** $i == r$ **then**
  - break
- $p = \texttt{parent}_\pi(i)$
- $m_{i \to p} = \max_{y_i} \phi_{ip} + m_i$
- $a_i = \arg\max_{y_i} \phi_{ip} + m_i$

$y_r^\star = \arg\max m_r$

for $i = \pi_{n-1}, \pi_{n-2}, \ldots, 1$ **do**
- $y_i^\star = a_i \left[y^\star_{\texttt{parent}_\pi(i)} \right]$

However, another extremely useful quantity which we rely on often in this work is the notion of a *marginal* score. When treating our model as a log-linear conditional distribution $p(y|x)$ (see §2.2.1), we can consider the marginal distribution of a particular variable

$$p(y_i|x) = \sum_{y_j \text{ s.t. } j \neq i} p(y|x) \tag{2.22}$$

as a measure of the model's belief over the value of variable i.

Similarly, we explore the notion of a *max-marginal*, an analogous quantity for the $\max$ operator:

$$s_x^\star(y_i) \triangleq \max_{y'} s(x, y'), \text{ subject to: } y'_i = y_i \tag{2.23}$$

In words, the max-marginal score for y_i is the score of the best full assignment restricted to fixing variable i to be y_i.

A naïve way to compute $s_x^\star(y_i)$ for all states in our model is as follows: for each i and each y_i (in total nk possibilities), set $\phi_{ij}(x, y'_i, y_j) \leftarrow -\infty$ for all $y'_i \neq y_i$, and all j attached to i, then run Algorithm 1. This ensures that y_i is the "chosen" state for variable i, and we get its max-marginal score $s_x^\star(y_i)$. This strategy would cost in total $O(nk \cdot nk^2) = O(n^2k^3)$, which is not very satisfying.

It turns out we can compute max-marginal quantities for all nk variable-state possibilities in $O(nk^2)$ time using a forward *and* backward pass of message passing and careful bookkeeping, similar to Algorithm 1. In addition, we can also collect *witness statistics*, which keep track of the number of times different states have been used in any max-marginal satisfying assignment (and thus are a "witness" to the assignment). The *witness* for $s_x^\star(y_i)$ is

$$y^\star(y_i) \triangleq \arg\max_{y'} s(x, y'), \text{ subject to: } y'_i = y_i, \tag{2.24}$$

which is the maximizer which corresponds to maximum $s_x^\star(y_i)$.

For completeness, the algorithm to compute max-marginal values and witness statistics is included in §B in Algorithm 2.

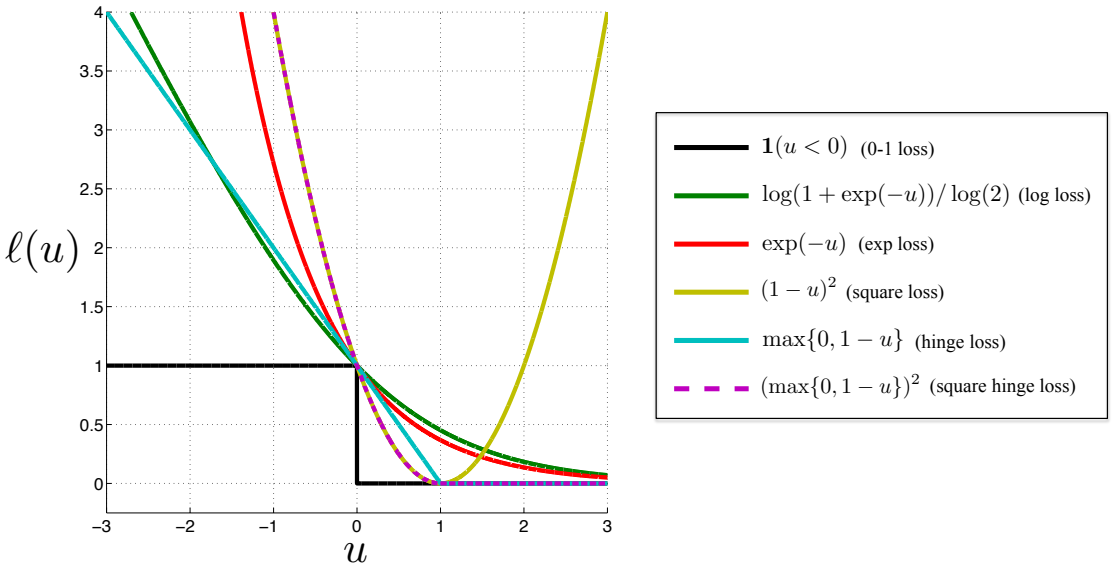


Figure 2.3: Convex surrogate supervised loss functions.

2.5 Supervised learning

In the supervised learning setting, we have access to a training set $S = \{(x^{(j)}, y^{(j)})\}_{j=1}^m$ of examples assumed to be sampled independently, identically distributed (i.i.d.) from some true joint distribution $(x, y) \sim P(X, Y)$. The standard supervised learning task is to learn a hypothesis $h : \mathcal{X} \times \mathcal{Y} \mapsto \mathcal{Y}$ that minimizes the average misclassification error (0/1 error) on the training set:

$$\mathcal{L}_{0/1}(h, S) = \frac{1}{m} \sum_{j=1}^{m} \mathbf{1}\left(h(x^{(j)}) \neq y^{(j)}\right) \tag{2.25}$$

The linear hypothesis class we consider is of the form $h(x) = \arg\max_y h(x, y)$, where the scoring function $h(x, y) \triangleq \mathbf{w} \cdot \mathbf{f}(x, y)$ is the inner product of a vector of parameters $\mathbf{w}$ and a feature function $\mathbf{f} : X \times Y \mapsto \mathbb{R}^d$ mapping (x, y) pairs to d real-valued features, as discussed in §2.1.

Using a linear representation, we have the following optimization problem to learn our

model from training data:

$$\underset{\mathbf{w}}{\textbf{minimize}}\ \mathcal{L}_{0/1}(h, S) = \tag{2.26}$$

$$\underset{\mathbf{w}}{\textbf{minimize}}\ \frac{1}{m}\sum_{j=1}^{m} \mathbf{1}\left(\arg\max_{y}[\mathbf{w}\cdot\mathbf{f}(x^{(j)}, y)] \neq y^{(j)}\right) \tag{2.27}$$

The above optimization is extremely difficult to solve in high dimensions because it is non-convex and discontinuous, leading to many poor local minima. In light of this, we introduce a *convex surrogate* to the indicator loss function $\ell(\cdot)$ to replace $\mathbf{1}(\cdot)$ with an upper-bound (Bishop, 2006):

$$\underset{\mathbf{w}}{\textbf{minimize}}\ \mathcal{L}_{\ell}(h, S) = \tag{2.28}$$

$$\underset{\mathbf{w}}{\textbf{minimize}}\ \frac{1}{m}\sum_{j=1}^{m} \ell\left(\max_{y}\mathbf{w}\cdot\mathbf{f}(x^{(j)}, y), y^{(j)}\right) \tag{2.29}$$

There are many common choices for $\ell(\cdot)$, shown in Figure 2.3, leading to many different structured and non-structured machine learning algorithms. We choose to use the *hinge loss* for most of this work, which leads to a simple and intuitive stochastic optimization algorithm. The hinge loss is of the form $\ell(u) = \max(0, 1-u) \triangleq [1-u]_+$, giving us the following optimization problem:

$$\underset{\mathbf{w}}{\textbf{minimize}}\ \frac{1}{m}\sum_{j=1}^{m}\left[\max_{y}\mathbf{w}\cdot\mathbf{f}(x^{(j)}, y) - \mathbf{w}\cdot\mathbf{f}(x, y^{(j)}) + 1\right]_+ \tag{2.30}$$

Depending on the training set, this form of learning function may have issues. Particularly, if the training set is *separable*, there are many possible $\mathbf{w}$'s that will achieve an optimum value of 0 in Equation 2.30. For example, if a $\mathbf{w}$ is a minimizer, so is $10\cdot\mathbf{w}$. Thus, we add an additional regularization term that seeks to find a low error $\mathbf{w}$, as well as a simple $\mathbf{w}$—one whose weights are small. To this effect, a regularized form is

$$\underset{\mathbf{w}}{\textbf{minimize}}\ \frac{\lambda}{2}||\mathbf{w}||_2^2 + \frac{1}{m}\sum_{j=1}^{m}\left[\max_{y}\mathbf{w}\cdot\mathbf{f}(x^{(j)}, y) - \mathbf{w}\cdot\mathbf{f}(x, y^{(j)}) + 1\right]_+ \tag{2.31}$$

which encourages $\mathbf{w}$ to be small in an L_2-norm sense. The meta-parameter λ balances how much we care about model simplicity (first term) versus training accuracy (second term).

Model simplicity implies a better chance of generalizing to new data, at the risk of being too restrained to learn an effective classifier. Such concerns about model underfitting or overfitting and the tradeoff between model bias and model variance are well-studied in the machine learning literature, especially for linear classifiers. See Bishop (2006); Friedman et al. (2001) for rigorous treatments.

2.5.1 Learning algorithms

Any convex learning function of the form of Equation 2.29 lends itself to a variety of optimization algorithms. Because we are guaranteed to reach a global optimum (modulo numerical precision and computational limitations), we focus on choosing a method that is quick and simple. Stochastic first-order descent methods estimate the gradient of the function one data point at a time, and are attractive due to their low memory requirements and fast convergence rates per example examined (Shalev-Shwartz et al., 2007).

Using Equation 2.31, the second term is piecewise-linear, and we take stochastic subgradient steps to optimize it, also known as structured perceptron (Collins, 2002). Given a single supervised example (x, y), we make the following update if the second, hinge loss term of Equation 2.31 (*i.e.*, the subgradient) is non-zero:

$$\mathbf{w}' \leftarrow (1 - \eta\lambda)\mathbf{w} + \eta \left(\mathbf{f}(x, y) - \mathbf{f}(x, y^\star)\right). \tag{2.32}$$

where, again, $y^\star = \arg\max'_y \mathbf{w} \cdot \mathbf{f}(x, y')$.

Chapter 3

Pictorial structures: Pose estimation meets structured prediction

Classical pictorial structure models (PS models) are a class of structured models specifically proposed to represent 2D objects with parts which can articulate in a kinematically plausible way. The model was first proposed by Fischler and Elschlager (1973), and has been hugely popular in computer vision since computational innovations by Felzenszwalb and Huttenlocher (2005).

Pictorial structures take the form of a pairwise structured model as described in §2.2:

$$s(x, y) = \sum_{i \in \mathcal{V}_G} \phi_i(x, y_i) + \sum_{ij \in \mathcal{E}_G} \phi_{ij}(y_i - y_j) \tag{3.1}$$

with the important exception that *the pairwise terms $\phi_{ij}(y_i - y_j)$ are image-independent*—they are not a function of x.

In PS models, the variables correspond to the major body parts. For upper-body pose estimation, y typically corresponds to {head, torso, left upper arm, left lower arm, right upper arm, right lower arm}, and $y_i \in \{1, 2, 3, \ldots, 80 \times 80 \times 24\}$ represents a typical 80×80 discretized grid of spatial locations, and 24 discretized angles for each part. Thus

one way to think about the representation for each part is a unit vector describing position and orientation of each limb, and the state space as a 3D cuboid.

Edge structure: In order to support tractable inference, the part interaction structure G must be a tree (§2.3). The most common tree is formed by considering kinematic connections only: the lower-arm is connected to the upper-arm, the upper-arm to the torso, etcetera.

Unary potentials: The unary potentials $\phi_i(x, y_i)$ encode how likely a limb is at location/orientation y_i in the image. This can be thought of as a part detector applied to an image patch located at y_i. In practice these detectors are patch-based sliding window object detectors, applied at every location and orientation. The typical representation is based on edge information in order to be invariant to lighting and color. Edge orientation and magnitude are often locally quantized and histogrammed to be more numerically and spatially stable, and invariant to local signal gain. Feature representations are discussed in detail in §8.

Pairwise potentials: Pictorial structures assumes a restricted form of pairwise potential the depends only on the deformation between kinematially attached parts. In general, this is a quadratic stretching cost of the form

$$\phi_{ij}(y_i - y_j) = -||T_{ij}y_i - T_{ji}y_j - \delta_{ij}||_2^2 \tag{3.2}$$

where T_{ij} are rigid transformations (rotation, translation and scale) to place neighboring parts in a local coordinate frame, and δ_{ij} is the expected displacement between the parts. For example, $\phi_{ij}(y_{\text{luarm}} - y_{\text{llarm}})$ measures the squared Euclidean distance between the elbow locations according to the left upper arm and left lower arm variables.

Spring model interpretation: The reasons for restricting the pairwise potential to be only a function of spatial deformation are primarily computational, as discussed in §3.1. In addition, this type of model is simple and intuitive to interpret as a "spring model" of object layout: the PS model can be interpreted as a set of springs at rest in default positions δ_{ij} stretched by displacement $T_{ij}y_i - T_{ji}y_j$. The spring tightness is encoded by warping transformations T_{ij} which can scale the dimensions of the spatial Euclidean coordinate

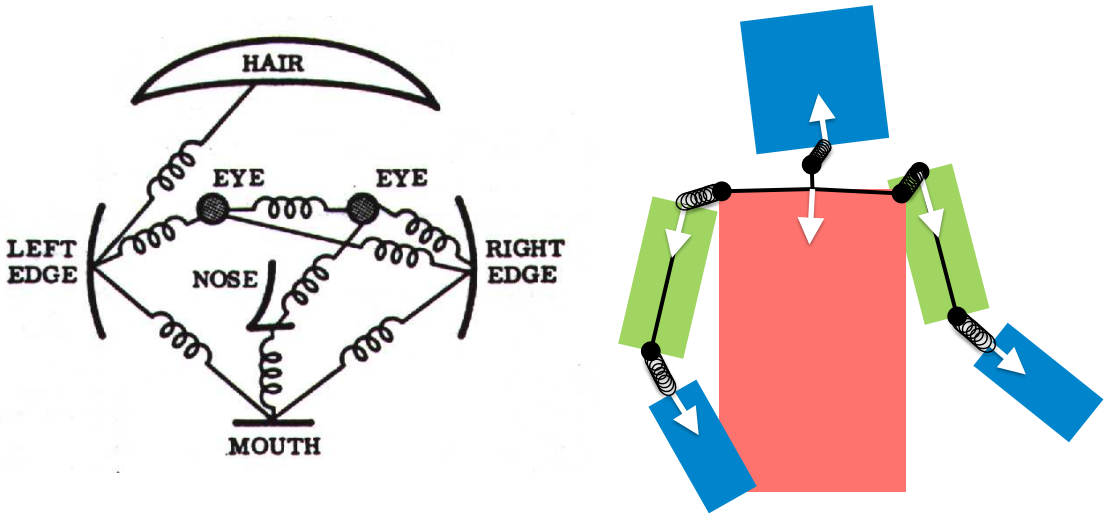


Figure 3.1: Spring model of pose. At left, the original spring pictorial structure model that appeared in Fischler and Elschlager (1973). At right, the standard PS model for 2D human pose. The states are shown as unit vectors indicating the position of joints and their direction. The mean displacement between joints are shown as solid black circles, connected by solid black lines to show the kinematic tree structure. The displacement from mean positions are shown as springs stretching.

space. The unary terms pull the spring ends towards locations y_i with higher scores ϕ_i which are more likely to be a location for part i. Thus the spring model seeks a balance between confidence in individual part detectors, and the amount of deformation from a default, 2D geometric prior of the human layout. Figure 3.1 provides an illustration.

3.1 Sub-quadratic inference

As discussed in §2.2, the computation required to obtain the highest-scoring pose out of the exponentially-many possibilities according to Equation 3.1 is $O(kn^2)$, using Algorithm 1. In pose estimation, the reason is intuitive: for each possible location of an upper arm, we need to exhaustively search to find the best location of a corresponding lower arm,

checking all such possibilities. Doing this for all left or right upper arms yields an exact n^2 local search.

In practice, there is no need to check *all* possibilities of lower arms for each upper arm location; it is sufficient to check only in a reasonable spatial window around the mean displacement location δ_{ij}, because it is impossible for kinematically-connected object parts to be stretched too far. Even so, we must search over a fixed fraction of neighbor states for each part, which still scales quadratically with the resolution of the state space. Typically a state space window of $80/5 \times 80/5 \times 24$ possibilities need to be evaluated for each location, yielding on the order of $(80 \times 80 \times 24)^2/25 = 943,718,400$ computations—nearly a billion—for a pair of parts.

The search operation described here is exactly that performed in the following lines of Algorithm 1, where i indexes a part, and j its parent in a topological ordering of the tree graph[1]:

$$m_{i \to j} = \max_{y_i} \phi_{ij} + m_i \tag{3.3}$$

$$a_i = \arg\max_{y_i} \phi_{ij} + m_i \tag{3.4}$$

The quantity $a_i(y_j)$ is the best placement of part i when placing part j at location y_j, using all the information from predecessor parts in the topological ordering. The quantity $m_{i \to j}(y_j)$ is the corresponding score for that placement.

It turns out that when the pairwise term ϕ_{ij} takes the form it does for classical PS, then we can apply a *generalized distance transform* procedure to compute a_i and $m_{i \to j}$ over all possibilities for y_i and y_j in time linear in the number of states. This makes the total cost of inference $O(kn)$ instead of $O(kn^2)$. This was introduced by Felzenszwalb and Huttenlocher (2005).

The generalized distance transform solves the problem

$$\mathcal{D}_q(p; f) = \min_q ||p - q||_2^2 + f(q) \tag{3.5}$$

[1] We substituted variable index p with j here in keeping with the notation in this section.

for any discrete function $f(\cdot)$ where p and q are discrete vector quantities with bounded domain. The solution to the above can be found via dynamic programming to keep track of the lower envelope of parabolas of the form $(p_i - q_i)^2 + f(q_i)$ for a single dimension i, operating over a dimension at a time (Felzenszwalb and Huttenlocher, 2006). By manipulating Equation 3.3, we can massage it into the form of Equation 3.5.

$$\max_{y_i} \phi_{ij} + m_i(y_i) \tag{3.6}$$

$$= \max_{y_i} -||T_{ij}y_i - T_{ji}y_j - \delta_{ij}||_2^2 + m_i(y_i) \tag{3.7}$$

$$= -\min_{y_i} ||T_{ij}y_i - T_{ji}y_j - \delta_{ij}||_2^2 + (-m_i(y_i)) \tag{3.8}$$

$$= -\min_{y_i} ||\tilde{y}_i - \tilde{y}_j||_2^2 + (-m_i(y_i)) \tag{3.9}$$

$$= -\mathcal{D}_{\tilde{y}_i}(\tilde{y}_j; -m_i) \tag{3.10}$$

Where $\tilde{y}_{i,j}$ are transformed versions of the 2D state space for $y_{i,j}$ via the rigid transforms of T_{ij}, T_{ji} and a translation by δ_{ij}. In fact, the tricks used to compute the generalized distance transform efficiently work for any unimodal functions of geometric displacement, *e.g.* L_1-norm.

3.2 Limitations

There are some severe limitations to the classical PS model. Some of these are primary motivations for the contributions of this thesis.

3.2.1 2D representation for a 3D object

Inferring 3D pose from 2D input is inherently ambiguous. Even when the real world geometric specifications of an object are known (*e.g.*, the true lengths of limbs in centimeters, or their length ratios), there is ambiguity due to camera projectection—does the imaged

limb appear shorter because it is pointing towards or away from the camera? In a solution provided by Taylor (2000), knowing the locations of n limbs' image coordinates and real world proportions results in a family of 2^n possible real-world solutions, up to an unknown scale factor. User-guided interfaces aided with this technology still take minutes of human effort in trial and error adjustments to get a plausible 3D pose from 2D (Bourdev and Malik, 2009).

It is intrinsic to the 2D pose problem that we must reason in 2D with 2D input (although when estimating pose in video, temporal consistency may help resolve some of the ambiguities). As a result, the 2D geometric priors are inherently weak. When a rough global scale is known, we can determine that a limb length ranges from some maximum distance (when parallel to the camera plane) down to 0 pixels (when the pointing straight at the camera). This also makes background and self-occlusion extremely difficult to model.

3.2.2 Cardboard people

Representing pose as limbs parametrized by location and orientation has been called a "cardboard people" representation, since at best limbs can be described by rectangles of fixed dimensions (from idealized cylinder representations of limbs projected into the image) (Ju et al., 1996). As such, the representation leaves no room for modeling foreshortening, and is forced to discretize the set of orientations into coarse increments—$10°$ to $15°$. In fact, the system originally proposed by Felzenszwalb and Huttenlocher (2005) also considered 10 discretized values of scale for each part to handle foreshortening, but no recent publicly available system (Andriluka et al., 2009; Eichner and Ferrari, 2009; Sapp et al., 2010a,b, 2011) includes a scale for each part. The reasons are two-fold.

First, it is computationally more expensive. Ten scales per part results in a state space for each part that is 10 times larger. When using distance transform inference tricks as in §3.1, this makes inferences 10 times slower. If using standard max-sum inference (§2.2), inference would be 100 times slower.

Second, there is very little signal in real world scenarios from which to infer foreshortening. In Felzenszwalb and Huttenlocher (2005) the system operated on foreground sillhouettes, making background clutter a non-issue. When severely distorted, it is extremely difficult to determine, even to the human eye, what is a foreshortened arm and what is simply background clutter. This problem can be ameliorated when parsing human pose in video, in which we can track joints as the arm undergoes a foreshortening over a sequence of frames. We address these issues in §6.

3.2.3 Unimodal potentials

In order to achieve fast inference, modeling sacrifices have been made. The linear time max-sum inference for PS described in §3.1 via distance transforms is supported only if the pairwise potentials are a unimodal function of geometric displacement. This is quite a strong assumption, especially for representing angles, *e.g.*, between upper and lower arms. Figure 1.3-c shows that the inner angle between lower and upper arms is highly multi-modal.

Unary potentials are also typically unimodal, in the sense that they are modeled as individual linear classifiers on an edge representation, of the form $\mathbf{w}_i \cdot \mathbf{f}_i(x, y)$. As a consequence, all the variations of part appearance due to pose, deformation, body type, lighting, foreshortening and clothing are being fit with a single linear model, resulting in a harsh quantization of the rich space of poses. The reasons for this are not only computational (the weights can be re-interpreted as a 2D linear filter, and $O(n \log n)$ convolution can be used to evalute them), but also statistical: using a more complex model requires more training data for accurate estimation. This issue is the motivation for the model proposed in §7.

3.2.4 Image-independent interactions

Maybe the most unsatisfying property of the pictorial structure model is its effective blindness to image content when determining how to "piece patches together" in a geometrically plausible way, due to the image-independent pairwise terms $\phi_{ij}(y_i - y_j)$. This is a necessity in order to achieve sub-quadratic inference, as discussed in §3.1. This means we are unable to encode in the pairwise term the likelihood that a pair of part hypotheses go together because they *look like* they go together; only that they go together because they fit together geometrically, like putting together a jigsaw puzzle without looking at the puzzles piece faces (Figure 1.4). Thus we are unable to express color, shape or region compatibility for a pair of parts in our model. Overcoming this limitation is a major contribution of this thesis, addressed in §5 and §6.

Chapter 4

Related work

In this section we survey a variety of methods for 2D human pose estimation in both single frames and video. The majority of this work is on *part-based* models, which decompose and reason about basic units of pose and their relationships. A few non-part-based approaches are mentioned in §4.4. We omit work using other more powerful sensors such as motion capture environments or multiple cameras, since many of the significant issues in 2D human pose estimation are not present in these domains (*e.g.*, clutter, viewpoint and occlusion issues).

4.1 Single-frame parts-based models

The idea that real-world objects captured in 2D images can be modeled as collections of interrelated atomic parts goes back to the foundational work of Binford (1971). The term "pictorial structures" was introduced by Fischler and Elschlager (1973). In this work, the authors lay out the basic PS model described in §3, an aggregation of individual local part scores and pairwise part-part scores which depend only on the displacement between connected parts. The standard dynamic programming solution is presented here (see §2.3), quadratic in the number of parts.

The pictorial structures model was made popular in recent literature by Felzenszwalb

and Huttenlocher (2005), who contributed a linear time message passing algorithm using distance transforms (explained in §3.1) which made tree-structured models tractable. This led to an explosion of work using and refining this basic PS model:

Ramanan (2006) introduced discriminative training of edge template filters to improve upon previously hand-crafted templates, and proposed an iterative estimation procedure which uses the current guess at the foreground color distributions in subsequent parses. Ferrari et al. (2008) provide an end-to-end multi-step pipeline to parse upper body pose in video and whittle down the number of possibilities: first detect people, estimate their color, reject background hypotheses using a conservative color-based graphcut, then track remaining candidates through time. Eichner et al. (2010) builds on Ferrari et al. (2008) and Ramanan (2006) to make a competitive state-of-the-art system. Andriluka et al. (2009) improve upon the basic PS model by learning more powerful part filters using Adaboost classifers on top of shape context to represent each part.

All of these models rely heavily on the facts that (1) the part interactions form a tree and (2) these interactions must be simple functions of geometry only. If these are invalidated, obtaining the best solution from the model is intractable (§2.3). From a certain viewpoint, the rest of the literature discussed here is an innovation in response to at least one of these limitations, which lead to the following shortcomings:

S-A: *Lack of important part-pair relationships.* For example, to mantain acyclicity, typically the left and right arms are left unconnected. However, this omits useful cues for parsing—*e.g.*, to express the fact that arms are likely the same color, or do not often lie on top of each other.

S-B: *Lack of larger and richer relationships.* Modelers would also like to design cost functions that consider more than two parts at a time (and informed by more than just geometry). For example, one would like to embed in the cost function answers to questions such as ”is this a realistic half-body with respect to countours in the image?” Representing larger collections of parts naïvely leads to a combinatorial explosion of possibilities.

S-C: *A "one-size-fits-all' representation of appearance and pose.* In a single tree with parameters describing part appearance and part geometry, it is difficult to capture the wide variability in human pose found in nature. For example, a single tree model must have a very general template for the lower arm to capture all appearance changes due to clothing, lighting, body size and articulation. It must also represent the wide range of lower arm-upper arm geometries with a single (typically unimodal) angular distribution.

4.1.1 Dealing with cyclic models

In a desire to overcome shortcoming S-A, some researchers consider cyclic, single frame graphs. In a cyclic, or loopy, graph, inference is $\#P$-complete (Koller and Friedman, 2009), without a known polynomial-time algorithm. Tran and Forsyth (2010) consider a fully connected graph of limb interactions, and use local greedy search as an approximate inference technique. Quite recently Sun et al. (2012) use a fully-connected graph using the output of our cascade system (§5), and improve upon our results which use a tree model. They use branch-and-bound techniques and an order of magnitude more time for inference to obtain exact solutions (roughly 20 minutes versus 10 seconds).

Many other approximate inference techniques have been applied to cyclic graphs in video, see §4.3.

4.1.2 A family of trees

Describing pose with multiple trees can solve issues with shortcomings S-A and S-C. Using multiple trees, we can cover all part relationships desired in at least some tree, and maintain acyclicity in all trees. Or, multiple trees can share the same graph structure, but each tree (or subcollection) can model a subspace of the pose space (*e.g.*, an arms-crossed model; a profile model). Tree beliefs are typically accumulated or maxed over to determine a final pose. In either case, such models are sometimes called "mixture-of-trees" models.

Wang and Mori (2008) use multiple tree models to capture previously unused part relationships to reason about occlusion and "double counting" (left and right symmetric parts both attaching to the most likely part image evidence). Our ESM model §6 uses these types of cues and more. Ioffe and Forsyth (2001) also use multiple trees to reason about occlusion in video—one tree for each common occlusion scenario. Johnson and Everingham (2011) and Zhu and Ramanan (2012) use dozens of tree models for full body and face parsing, respectively, to capture broad pose categories such as "upside down" body or "left 3/4 profile" face. In practice, mode definitions are data- driven and not necessarily semantically interpretable.

4.1.3 Multimodal compositional tree models

To address shortcomings S-B, many works have begun to look at additional "parts" which are actually compositions of atomic parts (*e.g.* lower arm plus upper arm induce a full arm part). The compositional parts connect to simpler parts and model geometric consistency constraints (Sun and Savarese, 2011; Wang et al., 2011b; Yang and Ramanan, 2011). By virtue of necessity, the appearance terms of compositional parts must be multimodal—-it would be impossible to capture all appearances of an articulated half-body with a single part template. The typical approach is to pre-define a handful of discrete modes based on training data and thus have a collection of mode possibilities for each part. During inference, the correct mode is inferred as a state in the model or maxed out. This type of multimodal modeling extends and has shown to be very valuable even for only atomic parts (Yang and Ramanan, 2011). In fact, even a basic PS model which searches over rotations for each part can be thought of as an atomic part multimodal model, where each part orientation is a mode, and all modes share the same appearance parameters (the rotated template).

One can think of a family of trees that captures different portions of pose space per tree (§4.1.2) in this way: it is a multimodal model at the most global level. We take this view in §7.2 and give a more in-depth discussion of recent work in this area there, including our

LLPS model.

4.2 Bottom-up methods

Decomposing the body into parts and modeling only local interactions is one way of dealing with the combinatorial number of possibilities of human layout. An orthogonal approach is to keep the number of full pose possibilities manageable by growing them incrementally from bottom-up information, and rejecting unlikely pose proposals along the way. This is attractive over parts-based models in that, in intermediate stages, larger context can be leveraged to evaluate the quality of a larger collection of parts. This is a remedy to shortcomings S-A and S-B described in §4.1.

One of the earliest approaches in this vein was the idea of "body plans" (Forsyth and Fleck, 1997), models of object layout defining what parts of an object can be grouped together, at what stage of grouping, and how. This idea was revisited in (Mori et al., 2004), which supports a heuristic set of rules to group superpixels into limbs, then combine limbs into partial and finally full configurations. Srinivasan and Shi (2007) furthered this line of work by evaluating intermediate proposals by shape matching against a database of exemplar shapes for each stage.

The bottom-up approach holds potential for more powerful representations of large collections of parts than local parts-based models. On the other hand, the inference procedure in all these works is greedy and hand-tuned or learned at each grouping stage separately. It's not clear what exact cost function is being minimized, or whether they reach a global or local solution.

4.3 Temporal Models

Estimating pose in a sequence of images allows us to exploit inter-frame cues such as smoothness of motion and appearance. This has the potential to improve upon pose estimation independently in each frame. However, it introduces an additional challenge: tracking multiple parts over time leads to intractability. In the graphical model literature, this is known as *entanglement* (Koller and Friedman, 2009), a property which implies that inference becomes exponential in the number of parts times the length of the image sequence. This is a major computational hurdle to overcome. The works listed here provided different approximate solutions to this exponential roadblock, in order to exploit temporal continuity cues for better pose estimation.

4.3.1 Approximate inference in cyclic networks

Some rely on standard approximate inference techniques, which often result in expensive inference and poorly understood behaviors. Ferrari et al. (2008) use loopy belief propagation to incorporate geometric continuity through time. Loopy belief propagation involves iteratively propagating beliefs around the network until convergence is reached. Convergence, however, is not guaranteed, and the number of belief propagation steps required in practice is orders of magnitude more than required for exact inference in a tree model.

Loopy belief propagation is a *variational inference* method which can be interpreted as attempting to maximize a simpler distribution than the true Markov Random Field distribution of the model (Jaakkola and Jordan, 2000). A related approach proposed by Ioffe and Forsyth (2001) is direct coordinate ascent of the distribution. In their case, their coordinate ascent is guided by the problem: they alternate between maximizing assignments within frames, holding inter-frame beliefs fixed, then maximizing between frames, holding intra-frame beliefs fixed. This can be viewed as a application-specific constrained loopy belief propagation to achieve a local maxima.

Sigal et al. (2004a) use non-parametric belief propagation with learned motion distributions for temporal edges. In Sigal and Black (2004), a continuous state space model is employed, and they also resort to sampling. The other alternative with continuous state space modeling is to require analytical representations with convenient manipulations for clique scores, such as the Gaussian modeling of Kalman Filters (Roweis and Ghahramani, 1999).

Particle filtering is another form of approximate inference. Most particle filtering methods only support *forward*, or *online* inference in which the whole video sequence cannot be taken into account (and thus future information cannot be used to help in earlier video frames). The Condensation algorithm by Isard and Blake (1998) tracks multiple contours (parametrized by B-splines) via particle filtering. Singh and Nevatia (2011) consider a cyclic intra-frame model in conjunction with pose and action-part modes propagated through time, also approximately tracked with particles, using the same algorithm as Sigal and Black (2004).

In §6, we study another form of approximate inference, dual decomposition (Komodakis et al., 2007), a principled approximation framework with convergence guarantees. Dual decomposition decomposes inference in a cyclic network to inference in a set of tree "slave" networks, and iterates to make the slaves agree on a solution. We also provide a similar but much simpler approximation method that is computationally cheaper with no reduction in performance in §6.2.3.

4.3.2 Tracking-by-detection

A second kind of approximate inference in video is to make hard decisions in every frame to limit the set of pose hypotheses. This keeps the number of hypotheses from growing exponentially with time, and allows us to use much more powerful features between a pair of frames. However, it is impossible to recover from mistakes made in early frames.

Ren and Malik (2007) is a good example of this paradigm, using powerful reasoning (graph matching with color coherence) between frames to establish correspondences between the assumed correct pose in the current frame and segments in the next frame. In this system, there is no top-down model for the tracked object, so drifting away from the correct pose eventually is inevitable. In earlier work in the same spirit, Bregler and Malik (1998) propagate an assumed correct pose in the current frame to the next by doing local gradient descent to fit an exponential map representation of a kinematic chain of parts.

In the "strike a pose" work by Ramanan et al. (2005), a rigid anchor pose is first detected with high confidence. From it, more discriminative detectors based on color are learned from it to better detect poses in neighboring frames. The detection in each frame is done independently, but informed by the hard decision made by already detected poses. Andriluka et al. (2008) propose a "people-tracking-by-detection" method which first detects people independently in every frame, then reasons about people tracks and their latent walking modes through a hierarchical Gaussian process latent variable model.

4.4 Holistic approaches

Despite limitations, parts-based models allow for decomposing the problem from exponentially many possibilities to a quadratic number for each pair of parts. Atomic parts can employ appearance models with some degree of genericness, enabling the models to generalize to unseen poses in the future.

An entirely different approach is to tackle the problem holistically without decomposing the problem into smaller, conditionally independent pieces. These methods attempt to map directly from image to part locations, typically through the use of exemplars. Possibly the simplest is the work of Mori and Malik (2002) which finds a nearest neighbor based on shape context, and transfers joint locations. Toyama and Blake (2002) consider modeling pose as a mixture of exemplar similarities using chamfer distance to define a metric space. Shakhnarovich et al. (2003) treat exemplar distance as a locality-sensitive

hashing problem. More recently, Taylor et al. (2010) learn a non-linear embedding so that nearest-neighbor pose classification is effective at predicting hand locations, using a nearest neighbor database of $40,000$ images.

Finally, Ionescu et al. (2011) learns a non-linear regression from figure-ground hypotheses directly to a vector of 3D joint locations. The regression of different joints is non-linearly decorrelated by embedding in a latent space via Kernel Dependency Estimation, with a kernel based on similarity between estimated figure-ground masks and groundtruth masks generated from a synthetic 3D model rendered in a variety of poses.

All of the works mentioned here have not been applied to recent "in the wild" datasets that we consider in this thesis. It remains an open question whether such exemplar-based methods can capture all the variations present in such challenging datasets. Parts-based models can achieve a much higher degree of generality due to their decomposable nature.

Part II

Models and methods

Chapter 5

Cascaded Pictorial Structures

5.1 Introduction

Pictorial structure models, first proposed by Fischler and Elschlager (1973) and outlined in §3, are the standard method for human body pose estimation. The search over the full pose space is linear time in the number of parts when the part-part dependencies form a tree. However, the individual part state spaces are too large (typically hundreds of thousands of states) to allow complex appearance models to be evaluated densely. Most appearance models are therefore simple linear filters on edges, color and location Andriluka et al. (2009); Felzenszwalb and Huttenlocher (2005); Ferrari et al. (2008); Ramanan and Sminchisescu (2006).

Similarly, because of quadratic state-space complexity, part-part relationships are typically restricted to be image-independent deformation costs that allow for convolution or distance transform tricks to speed up inference Felzenszwalb and Huttenlocher (2005), see §3. A common problem in such models is poor localization of parts that have weak appearance cues or are easily confused with background clutter (accuracy for lower arms in human figures is almost half of that for torso or head Andriluka et al. (2009)). Localizing these elusive parts requires richer models of individual part shape and joint part-part appearance, including contour continuation and segmentation cues, which are prohibitive

to compute densely.

In order to enable richer appearance models, we propose to learn a cascade of pictorial structures (CPS) of increasing pose resolution which progressively filter the pose state space. Conceptually, the idea is similar to the work on cascades for face detection Fleuret and Geman (2001); Viola and Jones (2002), but the key difference is the use of structured models. Each level of the cascade works on a certain spatial and angular resolution. A level refines the set of candidates from the previous level and then runs inference to determine which poses to filter out. For each part, the model selects poses with the largest *max-marginal* scores (§2.4), subject to a computational budget. Unlike conventional pruning heuristics, where the possible part locations are identified using the output of a detector, models in our cascade use inference in simpler structured models to identify what to prune, taking into account global pose in filtering decisions. As a result, at the final level the CPS model has to deal with a much smaller hypothesis set which allows us to use a rich combination of features.

In addition to the traditional part detectors and geometric features, we are able to incorporate object boundary continuity and smoothness, as well as shape features, discussed in detail in §8. The former features represent mid-level and bottom-up cues, while the latter capture shape information, which is complementary to the traditional HoG-based part models. The approach is illustrated in the overview Figure 5.4. We apply the presented CPS model combined with the richer set of features on the Buffy and PASCAL Stickmen benchmark, improving the state-of-the-art on arm localization, as discussed in §IV.

We choose to model part configurations as a general linear MRF over arbitrary pairwise and unary terms:

$$s(x, y) = \mathbf{w} \cdot \mathbf{f}(x, y) = \sum_{i \in \mathcal{V}_\Upsilon} \mathbf{w}_i \cdot \mathbf{f}_i(x, y_i) + \sum_{ij \in \mathcal{E}_\Upsilon} \mathbf{w}_{ij} \cdot \mathbf{f}_{ij}(x, y_i, y_j) \tag{5.1}$$

where $\Upsilon = (\mathcal{V}_\Upsilon, \mathcal{E}_\Upsilon)$ defines a tree structured graph of part interactions. The parameters of our model are the pairwise and unary weight vectors $\mathbf{w}_{ij}$ and $\mathbf{w}_i$ corresponding to the pairwise and unary feature vectors $\mathbf{f}_{ij}(x, y_i, y_j)$ and $\mathbf{f}_i(x, y_i)$. The key differences with the

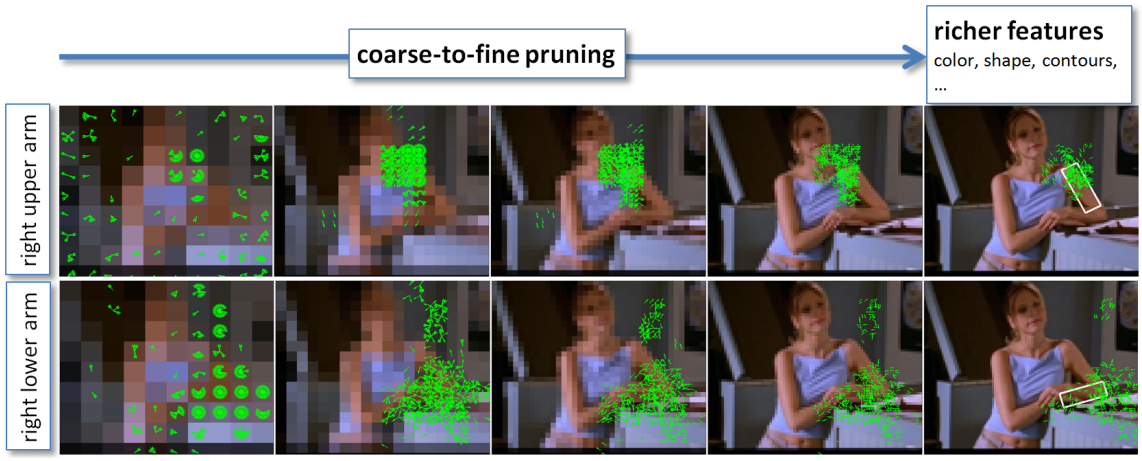


Figure 5.1: Overview of CPS: A discriminative coarse-to-fine cascade of pictorial structures filters the pose space so that expressive and computationally expensive cues can be used in the final pictorial structure. Shown are 5 levels of our coarse-to-fine cascade for the right upper and lower arm parts. Green vectors represent position and angle of unpruned states, the downsampled images correspond to the dimensions of the respective state space (but *not* the resolution at which features are computed), and the white rectangles represent classification using our final model.

classical PS model are (1) our pairwise costs allow data-dependent terms, and (2) we do not constrain our parameters to fit any parametric distribution such as a Gaussian distribution, as is done in Andriluka et al. (2009); Eichner and Ferrari (2009); Felzenszwalb and Huttenlocher (2005); Ramanan and Sminchisescu (2006). This is strictly more general. For example, we can express the pairwise features used in the classical model as $y_i \cdot y_i$, $y_j \cdot y_j$, and $y_i \cdot y_j$ without requiring that their corresponding weights can be combined into a positive semi-definite covariance matrix.

In this general form (Equation 5.1), inference can *not* be performed efficiently with distance transforms as discussed in §3.1, and we rely on standard $O(nk^2)$ dynamic programming techniques to compute the best scoring assignment $\arg\max_y s(x, y)$. However, this remains efficient and sub-quadratic in practice through the use of cascades.

5.2 Related work

For unstructured, binary classification, cascades of classifiers have been quite successful for reducing computation. Fleuret and Geman (2001) propose a coarse-to-fine sequence of binary tests to detect the presence and pose of objects in an image. The learned sequence of tests is trained to minimize expected computational cost. The extremely popular Viola-Jones classifier (Viola and Jones, 2002) implements a cascade of boosting ensembles, with earlier stages using fewer features to quickly reject large portions of the state space.

Our cascade model is inspired by these binary classification cascades. In natural language parsing, several works (Carreras et al., 2008; Petrov, 2009) use a coarse-to-fine idea closely related to ours and Fleuret and Geman (2001): the marginals of a simple context free grammar or dependency model are used to prune the parse chart for a more complex grammar.

Recently, P. Felzenszwalb (2010) proposed a cascade for a structured parts-based model. Their cascade works by early stopping while evaluating individual parts, if the combined part scores are less than fixed thresholds. While the form of this cascade can be posed in our more general framework (a cascade of models with an increasing number of parts), we differ from P. Felzenszwalb (2010) in that our pruning is based on thresholds that adapt based on inference in each test example, and we explicitly learn parameters in order to prune safely and efficiently. In Fleuret and Geman (2001); P. Felzenszwalb (2010); Viola and Jones (2002), the focus is on preserving established levels of accuracy while increasing speed. The focus in this paper is instead developing more complex models—previously infeasible due to the original intractable complexity—to improve state-of-the-art performance.

5.3 Structured Prediction Cascades

The recently introduced Structured Prediction Cascade framework (Weiss and Taskar, 2010) provides a principled way to prune the state space of a structured prediction problem

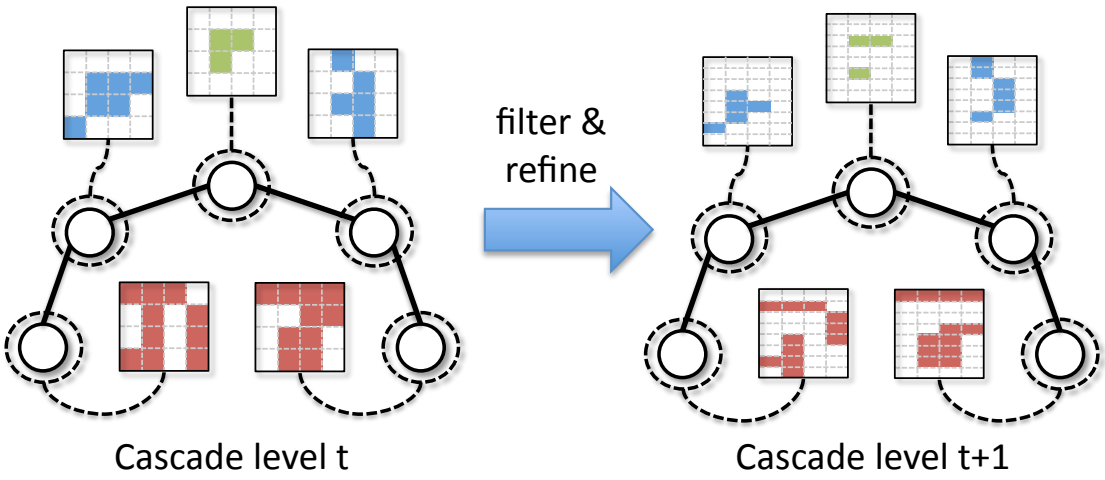


Figure 5.2: Two consecutive stages of a cascade, showing a model with a sparse set of states in a coarsened state space (top) filtering out states, and then passing them on to the next model (bottom) which works on a finer, upsampled version of the state space.

via a sequence of increasingly complex models. There are many possible ways of defining a sequence of increasingly complex models. In Weiss and Taskar (2010) the authors introduce higher-order cliques into their models in successive stages (first unary, then pairwise, ternary, etc.). Another option is to start with simple but computationally efficient features, and add more complex features downstream as the number of states decreases. Yet another option is to geometrically coarsen the original state space and successively prune and refine.

We use a coarse-to-fine state space approach with simple features until we are at a reasonably fine enough state space resolution and left with few enough states that we can introduce more complex features. We start with a severely coarsened state space and use standard pictorial structures unary detector scores and geometric features to perform quick exhaustive inference on the coarse state space.

5.3.1 Inference

The procedure for CPS described above is as follows. Two intermediate steps of the process are conceptualized in Figure 5.2.

- For an input x, initialize a coarse state space $\mathcal{S}_0 = \mathcal{Y}_0$ by spatially pooling states in the original space (downsampling the original state space volume).
- Repeat for each cascade level $t = 0, \ldots, T-1$:
  - Run sparse, exact inference over $\mathcal{S}_t$ using the t^{th} cascade model, computing max-marginal scores.
  - Filter states based on max-marginal scores to obtain $\tilde{\mathcal{S}}_t$:
    For each i, filter y_i if $s^\star_x(y_i) < t_x$, a data-dependent threshold (see §5.3.2).
  - Refine the state space of $\tilde{\mathcal{S}}_t$ to obtain $\mathcal{S}_t$ for the next cascaded model.
- Predict using the final level: $y^\star = \arg\max_{y \in \mathcal{Y}_T} s(x, y)$.

The key ingredient to the cascade framework is that states are pruned using *max-marginal* scores $s^\star_x(y_i)$, introduced in §2.4, computed efficiently using dynamic programming techniques. The notion of a max-marginal is intuitively explained in our model of pose estimation: The max-marginal for a part i at location y_i is the score of the highest scoring pose with part i fixed or "pinned" to location y_i. Importantly, max-marginals are a *global* quantity of a complete model of pose, rather than a local one: A part could have weak individual image evidence of being at location y_i but still have a high max-marginal score if the rest of the model believes this is a likely location for the part.

5.3.2 Filtering threshold

When applying a cascade, we have two competing objectives that we must trade off: accuracy and efficiency. We want to minimize the number of errors incurred by each level of the cascade and maximize the number of filtered max-marginals. A natural strategy is to

prune away the lowest ranked states based on max-marginal scores. For reasons explained below, we approximate a ranking threshold by pruning the states whose max-marginal score is lower than a data-specific threshold t_x: y_i is pruned if $s^\star_x(y_i) < t_x$. This threshold is defined as a convex combination of the highest score $s^\star_x = \max_y s(x, y)$ and the *mean max-marginal score*, defined as:

$$\bar{s}^\star_x = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{|\mathcal{Y}_i|} \sum_{y_i \in \mathcal{Y}_i} s^\star_x(y_i). \tag{5.2}$$

which is just the average $s^\star_x(y_i)$ over all parts and all states for each part. Our thresholding function is thus

$$t_x(s, \alpha) = \alpha s^\star_x + (1 - \alpha)\bar{s}^\star_x \tag{5.3}$$

where $\alpha \in [0, 1]$ is a parameter to be chosen that determines how aggressively to prune. When $\alpha = 1$, only the best state is kept, which is equivalent to finding the best, unconstrained assignment. When $\alpha = 0$ approximately half of the states are pruned (if the median of max-marginals is equal to the mean). The reasons for choosing this particular form of $t_x(s, \alpha)$ are (1) it is a function of the image x, which allows a threshold that adapts to the difficulty of the problem, and (2) it leads to a convex learning formulation with additional guarantees, as opposed to sorting max-marginals and choosing a cutoff.

5.3.3 Learning

The goal of learning is to fit parameters to our models $\mathbf{w}$ such that they are optimized for the task of filtering states efficiently and accurately. This is notably different than standard supervised learning, which attempts to find $\mathbf{w}$ to separate the right answer from wrong. In some sense, the filtering learning objective is easier to learn—the correct answer does not have to be the highest scoring, but only above the threshold value. To wit, we pose the following hard-constraint learning objective, assuming a training set $\{(x^{(j)}, y^{(j)})\}_{j=1}^m$:

$$\underset{\mathbf{w}}{\textbf{minimize}} \ \frac{1}{2}||\mathbf{w}||_2^2 \tag{5.4}$$

$$\textbf{subject to}: \ s(x^{(j)}, y^{(j)}) \geq t_{x^{(j)}}(s, \alpha) + 1, \quad \forall\, j \tag{5.5}$$

In words, the above is attempting to find a regularized set of weights $\mathbf{w}$ such that, in every training example, the score of the correct pose is above our image-adaptive threshold. We convert the hard constraint objective into an unconstrained hinge-loss form typical of a max-margin structured learning problem (see §2.5):

$$\underset{\mathbf{w}}{\mathbf{minimize}} \frac{\lambda}{2}||\mathbf{w}||_2^2 + \frac{1}{m}\sum_{j=1}^{m}\left[t_{x^{(j)}}(s,\alpha) - s(x^{(j)}, y^{(j)}) + 1\right]_+ \tag{5.6}$$

The learning formulation uses a simple fact about max-marginals and the definition of $t_x(s,\alpha)$ to get a handle on errors of the cascade: if $s(x,y) > t_x(s,\alpha)$, then for all i, $s_x^\star(y_i) > t_x(s,\alpha)$, so no part state of y is pruned. Given an example (x,y), this condition $s(x,y) > t_x(s,\alpha)$ is *sufficient* to ensure that no correct part is pruned, which is our justification for using max-marginals as the pruning measure. Note that probabilistic marginals do *not* have this property: $p(y|x)$ being above a threshold does not guarantee that $p(y_i|x)$ are above a threshold for all i.

We solve (5.6) using stochastic subgradient descent. Given an example (x,y), we apply the following when the term $[t_x(s,\alpha) - s(x,y) + 1]_+$ (and the subgradient) is non-zero:

$$\mathbf{w} \leftarrow \mathbf{w} + \eta\left[-\lambda\mathbf{w} + \mathbf{f}(y,x) - \alpha\mathbf{f}(y^\star,x) - (1-\alpha)\bar{\mathbf{f}}^\star(x)\right]. \tag{5.7}$$

Above, η is a learning rate parameter, $y^\star = \arg\max_{y'} s(x,y')$ is the highest scoring assignment and $\bar{\mathbf{f}}^\star(x)$ are the average features used by all max-marginal witnesses $y^\star(y_i)$ (explained in §2.4):

$$\bar{\mathbf{f}}^\star(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|\mathcal{Y}_i|}\sum_{y_i\in\mathcal{Y}_i}\mathbf{f}(x, y^\star(y_i)). \tag{5.8}$$

The key distinguishing feature of this update as compared to structured perceptron (§2.5.1) is the last term which subtracts features included in all max-marginal assignments $y^\star(y_i)$.

Sequential cascade learning The stages of the cascade are learned sequentially, from coarse to fine, and each has a different set of parameters $\mathbf{w}$ and $\mathcal{Y}_i$ for each part, as well as α. The states of the next level are simply refined versions of the states that have not been pruned. We describe the refinement structure of the cascade in §11.1.

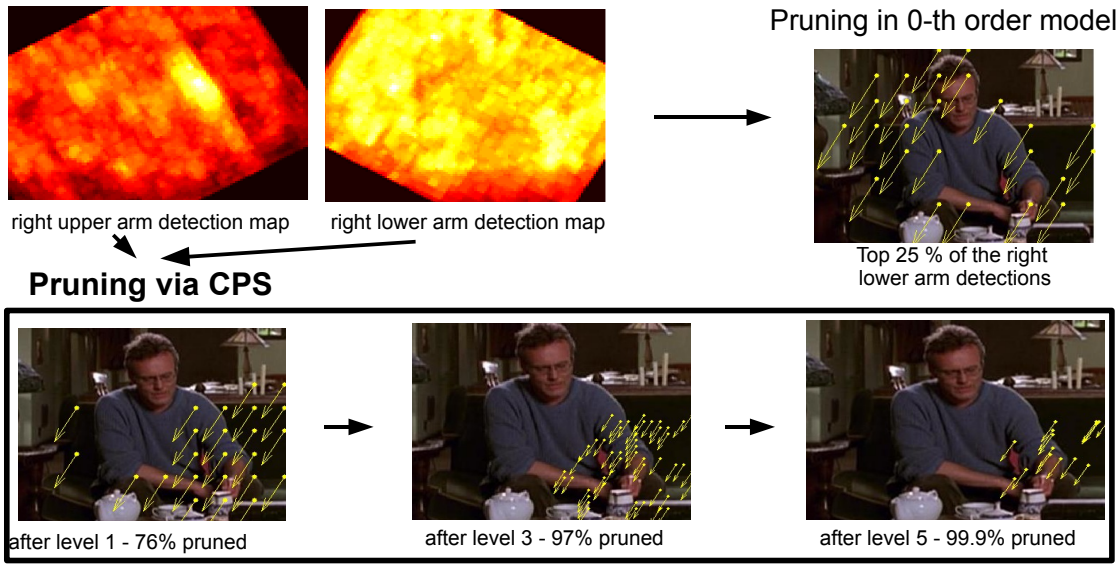


Figure 5.3: Cascade filtering example. Detector-based pruning (0^{th}-order model) by thresholding yields many hypotheses far way from the true one for the lower right arm. The CPS (bottom row), however, exploits global information (such as certainty in the shoulder location) to perform better state pruning.

5.3.4 Why not just detector-based pruning?

A naïve approach used in a variety of applications is to simply subsample states by thresholding outputs of part or sparse feature detectors, possibly combined with non-max suppression. Our approach, based on pruning on max-marginal values in a first-order model, is more sophisticated: for articulated parts-based models, strong evidence from other parts can keep a part which has weak individual evidence, and would be pruned using only detection scores. The failure of pre-filtering part locations in human pose estimation is also noted by Andriluka et al. (2009), and serves as the primary justification for their use of the dense classical PS. This is illustrated in Figure 5.3. In the results in §11.1 we confirm this empirically.

5.4 System summary

Figure 5.4 shows the cascade in action on a real world example. In the end we are left with approximately 500 states for each part, which allows us to add in a variety of useful, data-dependent features in a tractable manner. Figure 5.5-left shows how two of the features help in practice over simply using geometry alone. Details of all the features are in §8. Figure 5.5-right shows some of the good results achieved on test data; see §11 for quantitative and §A for more qualitative results.

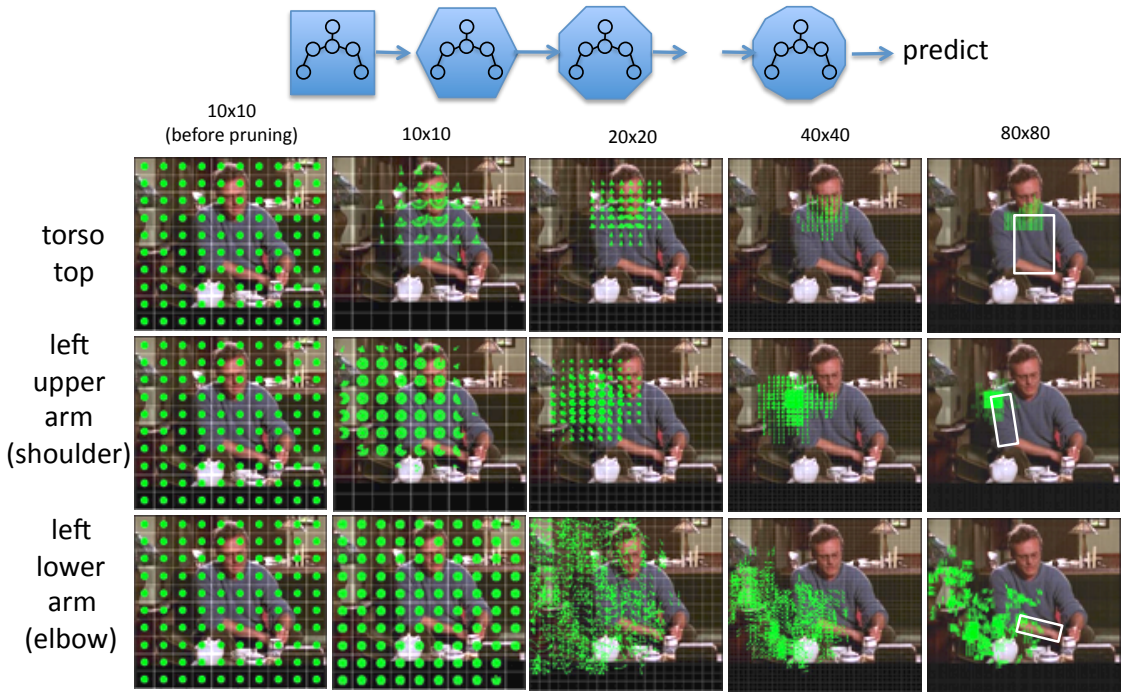


Figure 5.4: Cascade filtering on a real example. Each column shows a stage of the cascade. The states remaining for the torso, left upper arm and left upper arm are shown in the three rows, represented as small green unit vectors. Before pruning (first column) all states are present in a coarse state space. As the cascade progresses, initially it is easier to discard torso and upper arm hypotheses as these are more discriminative parts. The cascade automatically determines to postpone decisions about lower arms until near the end stage of the cascade.

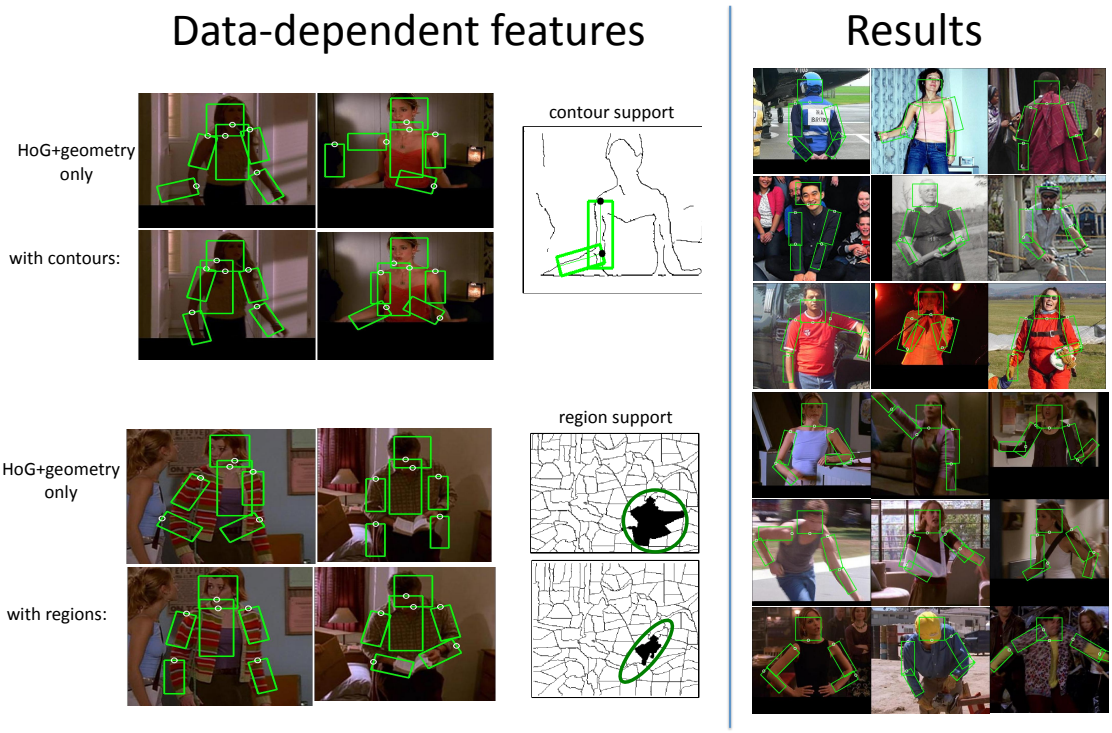


Figure 5.5: Left: some examples of where data-dependent features are beneficial. At top-left, using only geometry as a pairwise term, geometrically plausible false positives arise from hard edges in the background. However, these false positives are not supported by long contours in the image, which help determine the true answer. Similarly, at bottom-left false positives arise due to the highly-textured sweater. However, these are not supported by realistic region shapes. At right, a sampling of results from the Buffy and Pascal datasets which leverage all our rich features. See §11 and §A for more.

Chapter 6

Ensembles of Stretchable Models

6.1 Introduction

In this chapter, we focus on the task of estimating and tracking articulated 2D human pose in videos "in the wild": single-view, uncontrolled settings typical in movies, television and amateur video. Reliable parsing of human motion in such videos could vastly improve and refine action recognition and semantic retrieval. This task is made difficult by the considerable background clutter, camera movement, motion blur, poor contrast, body pose and shape variation, as well as illumination, clothing and appearance diversity. There has been an explosion of recent work on articulated pose estimation from single images of this type (Andriluka et al., 2009; Eichner and Ferrari, 2009; Felzenszwalb and Huttenlocher, 2005; Ferrari et al., 2008; Mikolajczyk et al., 2004; Ramanan, 2006; Ronfard et al., 2002), and the model proposed in §5.

Despite steady advances, localization of the most interesting parts (lower arms and hands) remains very inaccurate. Video provides additional motion cues, yet most work on articulated tracking requires manual initialization from a few frames (Balan and Black, 2006; Bregler and Malik, 1998; Buehler et al., 2008; Ju et al., 1996; Ren and Malik, 2007; Sminchisescu and Triggs, 2003). Several recent papers (Ferrari et al., 2008; Lan and Huttenlocher, 2005; Ramanan et al., 2005; Sapp et al., 2010c; Sigal et al., 2004b),

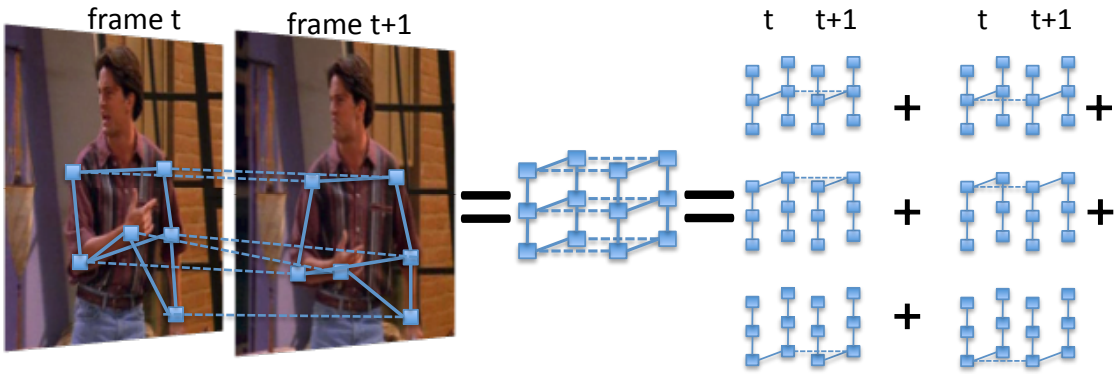


Figure 6.1: Overview. On the left, we show a full model with edges representing all pairwise relationships we want to capture within and between frames. We approximate this full, intractable loopy model by decomposing it into an ensemble of six models which cover the edge relationships of the full model. Our joint-centric representation is considerably more flexible in modeling pose variability than rigid limb-based representations (Andriluka et al., 2009; Ferrari et al., 2008; Sapp et al., 2010b).

however, combine tracking and estimation without any supervised initialization; our work in this chapter follows this setting.

In the "strike-a-pose" parsing method of Ramanan et al. (2005), an easily detectable canonical pose (e.g., limbs spread out) is automatically found in a long stretch of video and used as initialization for person-specific part appearance models. While this intuitive idea works well when every person strikes the easy canonical pose at least once in every video, many motion sequences are short and all the poses are difficult. Most work to date on short, difficult motions does not show significant improvement over single frame parsing (Ferrari et al., 2008; Sapp et al., 2010c), and sometimes causes actual degradation in accuracy when pose estimation is coupled across frames. One crucial reason for this is that joint parsing of multiple articulated parts over time involves intractable inference and learning problems, since part location/orientation variables have very large state spaces and the models are highly inter-connected (high-treewidth). Because of this barrier, previous work has resorted to approximate inference using sampling (Isard and

Blake, 1998; Sigal et al., 2004b; Sminchisescu and Triggs, 2003; Wang et al., 2007) and variational (Ferrari et al., 2008; Sigal et al., 2004a) methods, which often introduce poorly understood error and/or bias. The computational complexity of learning such models often limits the ability to learn rich features, resulting in using only simple, image-independent location-persistence coupling (Ferrari et al., 2008). One notable contrast to this is the work of Ren and Malik (2007), which uses powerful image cues and inference to establish correspondences between frames and within frames. However, their method makes greedy decisions as it proceeds through frames in order to handle the intractability of maintaining a distribution of beliefs throughout the video sequence, and hence has no way of recovering from bad choices in earlier frames.

Computational considerations also usually lead to restrictive simplifying assumptions on geometry: 2D limb lengths are fixed (given global scale) (Andriluka et al., 2009; Eichner and Ferrari, 2009; Ferrari et al., 2008; Ramanan et al., 2005). In typical video sequences this assumption is almost always violated because of foreshortening and body type variation, especially for the lower arms. For this paper, we collected a new challenging video dataset which extends the one in Sapp et al. (2010c), which we call VideoPose2.0. In VideoPose2.0, roughly 30% of all lower arms are significantly foreshortened; see Figure 6.2 for a summary of the dataset variability.

In this work, we overcome these computational and modeling limitations using an ensemble of tractable submodels which couple locations of body joints within and across frames using rich image-dependent cues. We address the problems of foreshortening and part scale variation by using joint centers (shoulders, elbows, wrists) as parts (Lee and Nevatia, 2006; Rogez et al., 2008; Urtasun and Darrell, 2008), instead of limbs. Each submodel in the ensemble tracks a single joint through time (e.g., left elbow), and also models the spatial arrangement of all joints in a single frame. Because of the tree structure of each submodel, we can perform efficient exact inference and use rich temporal features that depend on image appearance, e.g., color tracking and optical flow contours. The models are trained discriminatively using a large-margin loss. We experiment with

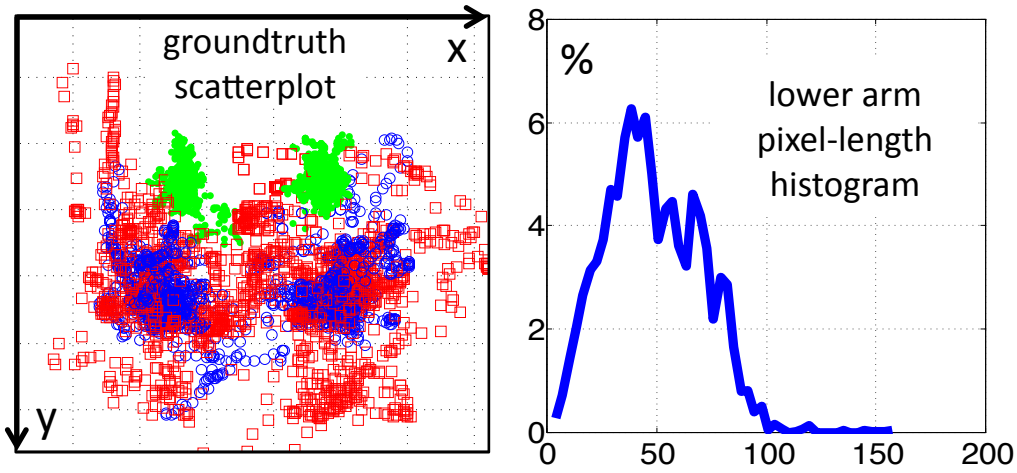


Figure 6.2: A summary of our new video dataset, VideoPose2.0. **Left:** A scatter plot of groundtruth joints (green dot = shoulder, blue circle = elbow, red square = wrist). **Right:** Histogram of lower arm lengths in our dataset, illustrating how important it is to model relative part length in a real world setting.

a range of inference techniques that introduce increasing coupling between models (and thus increasing computational cost) at test time. Intriguingly, we find that a highly efficient method of enforcing agreement on single variables that we introduced in Sapp et al. (2010c) outperforms costly approximate inference using dual decomposition.

We apply our motion parsing model on a new video dataset of highly varied and articulated poses from TV shows containing over 1,200 frames. We show significant quantitative and qualitative improvements over state-of-the-art single-frame pose estimation approaches (in particular, wrist and elbow accuracy improves by greater than 15%). In summary, the novel contributions of this chapter are: **(1)** an efficient ensemble of tree models for parsing human motion which covers a complex set of pairwise relationships and includes rich, image-dependent features; **(2)** stretchable 2D layout models, as opposed to the rigid parts typically used in pose estimation; **(3)** significant improvement over single frame parsing results (arguably a first in this setting) **(4)** a challenging new video dataset, VideoPose2.0, with a high degree of motion pose variability which shows

limitations of state-of-the-art methods. The VideoPose2.0 dataset, code and videos are available at *http://vision.grasp.upenn.edu/video*.

6.2 Modeling

As discussed, due to the intractability of jointly tracking multiple objects through time, previous work has made limiting assumptions in both the representation of pose (i.e., the inability to model foreshortening or fine angular granularity), and the interactions between parts, which do not capture rich, image-dependent relationships. We address each of these issues in a principled framework.

6.2.1 Stretchable models of human pose

One of the biggest limitations of current models of human pose is the "cardboard people" assumption (Ju et al., 1996): that body parts are rigid patches of fixed size (up to a global scale). In the pictorial structures framework, the human is represented as a collection of parts with fixed lengths, which along with the position and angle of each joint, completely determines the pose of the person [1]. Thus, wrists are always a fixed distance away from elbows in these models. This is a frequently violated assumption in realistic video, due to foreshortening and variation in body type. In the VideoPose2.0 dataset we introduce (Section IV), 30% of lower arms are significantly foreshortened, see Figure 6.2.

Two joints $>$ one limb: Rather than model human limbs as a position and orientation, we propose a model *directly on the pixel coordinates of each joint*. Although this choice introduces more variables (one for each joint rather than for each part), the state space for each variable is drastically reduced because we no longer need to reason over a finely discretized set of angles in the state space, and we can now implicitly represent nearly any angle between parts. In current PS implementations, this is a $24\times$ reduction in the state

[1] The seminal work of Felzenszwalb and Huttenlocher (2005) uses 10 discretized scales per part, but all modern implementations of PS use one fixed scale (Andriluka et al., 2009; Ferrari et al., 2008; Sapp et al., 2010b), partly due to its prohibitive increase in the state space.

space of each variable. In addition to this, because the length of the limb is now determined implicitly by the pixel distance between neighboring joints, the model naturally lends itself to capturing large variability in the part length. Because of its ability to represent finely discretized limb lengths, we refer to this model as *stretchable*, in contrast to the typical rigid, rectangle-based representation.

One side-effect of switching from a limb-centric to joint-centric model of pose is that unary attributes of the limb-centric model are now pairwise attributes of a joint-centric model. Furthermore, pairwise attributes in a limb-centric model correspond to ternary attributes in a joint-centric model, which we do not use. However, in a standard PS model, pairwise attributes are only image-independent functions of geometry, whereas in our model, pairwise potentials all incorporate image information, giving us overall a more expressive model than standard PS.

6.2.2 Ensembles of stretchable models (ESM)

Ideally we want a model of human motion which captures important relationships between all correlated parts. This includes parts that are connected kinematically (e.g., left elbow, left wrist), parts that are left/right symmetric (e.g., left elbow, right elbow), and instantiations of the same part in consecutive frames (e.g., left elbow at time t, left elbow at time $t+1$). Clearly, modeling all these relationships together leads to cyclic dependencies within a single frame (due to the three symmetry edges) and between consecutive frames (due to the six tracking edges); see Figure 6.1-left.

However, in general, it is always possible to express the score of a given state assignment in a full, intractable model as the sum of scores under a set of tree sub-models that collectively cover every edge in the full model. This is the key insight that allows us to include all the rich relationships we desire: we *decompose* our model of all interesting relationships related to parsing human motion into an ensemble of submodels, all of which are trees (and therefore tractable). Each tree submodel is responsible for tracking a single joint through time and additionally models the corresponding set of pairwise interactions

between joints in a single frame (Figure 6.1-right).

Formally, we pose this problem as a structured prediction task, where the input x is a video sequence of ℓ images and the output y is a sequence of $n\ell$ variables, where n is the number of parts (joint locations) included in the model. Each output y_i is the 2D coordinate of some part joint (defined in a 80×80 discretization of the pixel space) in some frame. We also use the shortcut notation $y_{\mathbf{t}} = \{y_i \mid y_i \text{ is in frame } t\}$ to index all n joint variables in frame t.

Instead of using the full 80×80 state space for every joint, we make use of Cascaded Pictorial Structures (§5) which is trained to prune unlikely portions of the state space away in a single frame. This leaves us with $|\mathcal{Y}_i| \leq 500$ possibilities for each joint in practice. Details are in §10.4.

Let $G = (\mathcal{V}, \mathcal{E})$ be our full graphical pose model; we further assume a general pairwise MRF model that decomposes over the vertices $\mathcal{V}$ and edges $\mathcal{E}$, so that

$$s(x, y) = \sum_{i \in \mathcal{V}} \mathbf{w}_i \cdot \mathbf{f}_i(x, y_i) + \sum_{(i,j) \in \mathcal{E}} \mathbf{w}_{ij} \cdot \mathbf{f}_{ij}(x, y_i, y_j). \tag{6.1}$$

Note that edges (i, j) may connect variables between consecutive frames. Let there be P tree sub-models, and $G_p = (\mathcal{V}, \mathcal{E}_p)$ be the sub-graph of G corresponding to the p'th one as in Figure 6.1-right. Then we decompose the score $s(x, y)$ into the sum of the scores of the P constituent sub-models: $s(x, y) = \sum_{p=1}^{P} s^p(x, y)$, the score of the p'th model is as in Equation 6.1, restricted to the edges $\mathcal{E}_p$, i.e.

$$s^p(x, y) = \sum_{i \in \mathcal{V}} \mathbf{w}_i^p \cdot \mathbf{f}_i(x, y_i) + \sum_{(i,j) \in \mathcal{E}_p} \mathbf{w}_{ij}^p \cdot \mathbf{f}_{ij}(x, y_i, y_j). \tag{6.2}$$

Note that we do not couple parameters across different models $\mathbf{w}^p$, so that different models can learn different parameters and effect different behaviors accorded to strengths and weaknesses dictated by their graph structures.

6.2.3 Inference

We explore several methods for combining the P independent models during test time to make a single final decision. The methods form a hierarchy of agreement criteria between the submodels: at one extreme, we enforce the constraint that all submodels must agree on the maximizing assignment to all variables, and at the other, inference is completely decoupled across submodels. Note that there is an inherent trade-off between the degree of agreement imposed on the models and the computational cost of the corresponding inference. We present our inference methods by order of decreasing agreement below.

Full Agreement via Dual Decomposition. A natural goal is to find the argmax decoding of joint locations throughout the entire sequence of frames using our original model in Equation 6.1. However, solving the argmax decoding problem exactly is prohibitively expensive, due to the high treewidth of this cyclic graph. We use the method of Dual Decomposition (DD) (Bertsekas, 1999; Komodakis et al., 2007) to solve a linear programming relaxation of the decoding problem as follows. Observe that the argmax decoding problem of our full model can be decomposed into P subproblems if those problems are coupled through a global equality constraint:

$$\arg\max_{y,y^1,\ldots,y^P} \sum_{p=1}^{P} s^p(x, y^p) \text{ s.t. } y^p = y \qquad (DD) \tag{6.3}$$

Although the optimization (6.3) is still intractable because of the integral constraints, optimizing the *dual* of (6.3) is always tractable (if we first drop the integrality requirement), but no longer guaranteed to return an optimal integral solution. We solve the dual problem with sub-gradient descent. Once complete agreement between all y^p is reached, then inference has converged to the exact solution of (6.3) (although eventual convergence is not guaranteed). In our experiments, if dual decomposition did not converge (after 500 iterations in practice, each requiring inference in all P models), we used the maximum scoring primal variables found during any descent iteration.

To solve the dual problem with sub-gradient descent, we introduce Lagrange multipliers ν_{ik}^p for every possible state assignment $y_i = k$ in every part model. We then alternate between updating the dual variables ν^p and the primal variables y^p:

$$y^p \leftarrow \arg\max_y \left(s^p(x, y) + \sum_{i,k} \nu_{ik}^p \mathbf{1}\left[y_i = k^p\right] \right) \tag{6.4}$$

$$\nu_{ik}^p \leftarrow \nu_{ik} - \alpha \left(\mathbf{1}\left[y_i^p = k\right] - \frac{1}{P} \sum_{p=1}^{P} \mathbf{1}\left[y_i^p = k\right] \right), \tag{6.5}$$

where α is a rate parameter chosen according to the scheme given in Komodakis et al. (2007).

Single Frame Agreement. An alternative to computing the MAP decoding of the entire sequence is to find the argmax solution while constraining model agreement to only a subset of the variables at a time. If we restrict our attention to the variables in a single time frame only, inference is considerably simpler. For each frame t, we solve

$$\arg\max_{y'_{\mathbf{t}}} \sum_{p=1}^{P} \max_{y : y_{\mathbf{t}} = y'_{\mathbf{t}}} s^p(x, y) \qquad (SF) \tag{6.6}$$

to obtain the joint configuration for that frame (remember $y_{\mathbf{t}}$ denotes the *set* of n joint variables in frame t). In words, the inner maximization finds the highest scoring sequence y subject to the constraint that the variables in frame t are fixed at positions $y'_{\mathbf{t}}$. The outer $\arg\max$ is over all possibilities of single frame configurations $y'_{\mathbf{t}}$. This extends the notion of a max-marginal of a variable (see §2.4) to a max-marginal over *a set* of variables. This method requires first computing the max-sum messages in a forward-backward message passing algorithm that incident to the variables in frame t, in each of the submodels. Finding the argmax decoding of $y_{\mathbf{t}}$ is then equivalent to inference in a grid with n variables (in our case $n = 6$ joints). This can be solved exactly by forming cliques of size 3, for which the message passing clique-tree forms a chain. The construction is illustrated in Figure 6.3. In each clique, there are at most pairwise potentials, and since the state space of each part is relatively small ($|\mathcal{Y}_i| \leq 500$, from first running CPS), the

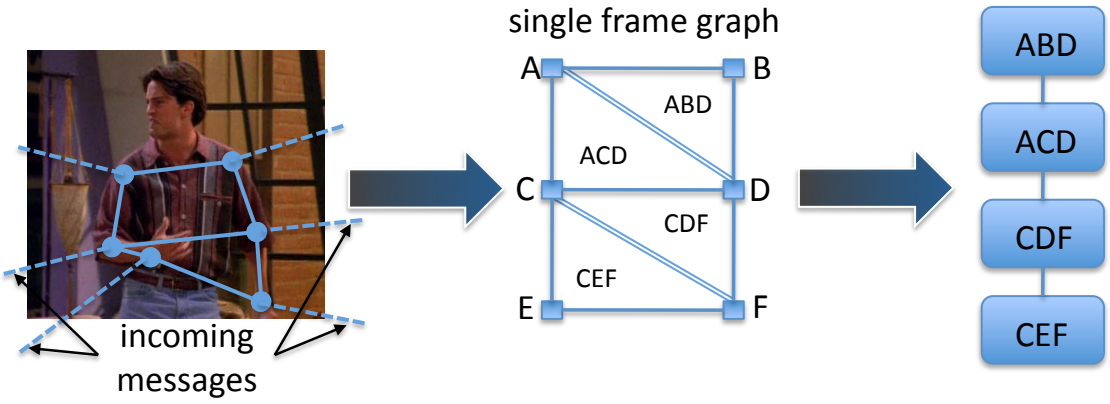


Figure 6.3: Single frame agreement construction. We first precompute messages coming into frame t from all other frames (left). We then combine 3-clique state spaces via Cartesian product to merge cliques into nodes in a larger state space (middle) to obtain a chain graph (right) over which we can perform exact inference.

$O(\sum_i |\mathcal{Y}_i|^3)$ cost of this inference takes less than a second per frame in practice, and in our experiments took about as long as performing inference in all P tree submodels (i.e., twice as slow overall).

Single Variable Agreement. We can further narrow our subset of interest down to a single variable at a time (Sapp et al., 2010c). This is a weaker criteria for model agreement, but yields cheaper and simpler inference. This gives us the following inference problem for the i^{th} variable:

$$\arg\max_{y_i'} \sum_{p=1}^{P} \max_{y:y_i=y_i'} s^p(x,y) \qquad (SV) \tag{6.7}$$

This can be solved by computing max-marginals for each model using standard forward-backward message passing, summing the P max-marginal scores, and taking the highest scoring sum. Note that this is actually equal to the best assignment decoding in the full model Equation 6.1 *when all sub-models agree on the argmax*. However, this rarely occurs in practice.

Independent / No Agreement We also compared the above methods to predicting each joint using the single model in the ensemble that incorporated temporal dependencies for that specific part, which we call the $Independent$ decoding scheme.

6.2.4 Learning

Although we enforce agreement during ensemble inference at *test time*, coupling inference during training for more than one variable is prohibitively expensive to use as part of a learning procedure. Thus we learn parameters $\mathbf{w}^p$ using *decoupled* inference separately for each model. To learn parameters, we optimize a convex hinge-loss objective for each model p separately:

$$\min_{\theta_m} \frac{\lambda}{2}||\mathbf{w}^p||^2 + \frac{1}{n}\sum_{j=1}^{m}\left[\max_{y} s^p(x^{(j)}, y) - s^p(x^{(j)}, y^{(j)}) + 1\right]_+ \tag{6.8}$$

We optimize this problem via stochastic subgradient descent, as explained in §2.5, and we choose λ and the number of training epochs using a held-out development set to minimize error.

6.3 System summary

Figure 6.4 shows a summary depiction of the Stretchable Ensembles process. Because we decompose our cyclic graph, we are able to use a variety of rich image-dependent features for tracking. However, this also requires us to reconcile different tree submodels' beliefs on the final predicted pose, shown on a few sample frames of a video at the bottom of the figure. For full test set results, we refer the reader to §11 and this video:

http://www.youtube.com/watch?v=vnOh8_D3RhQ.

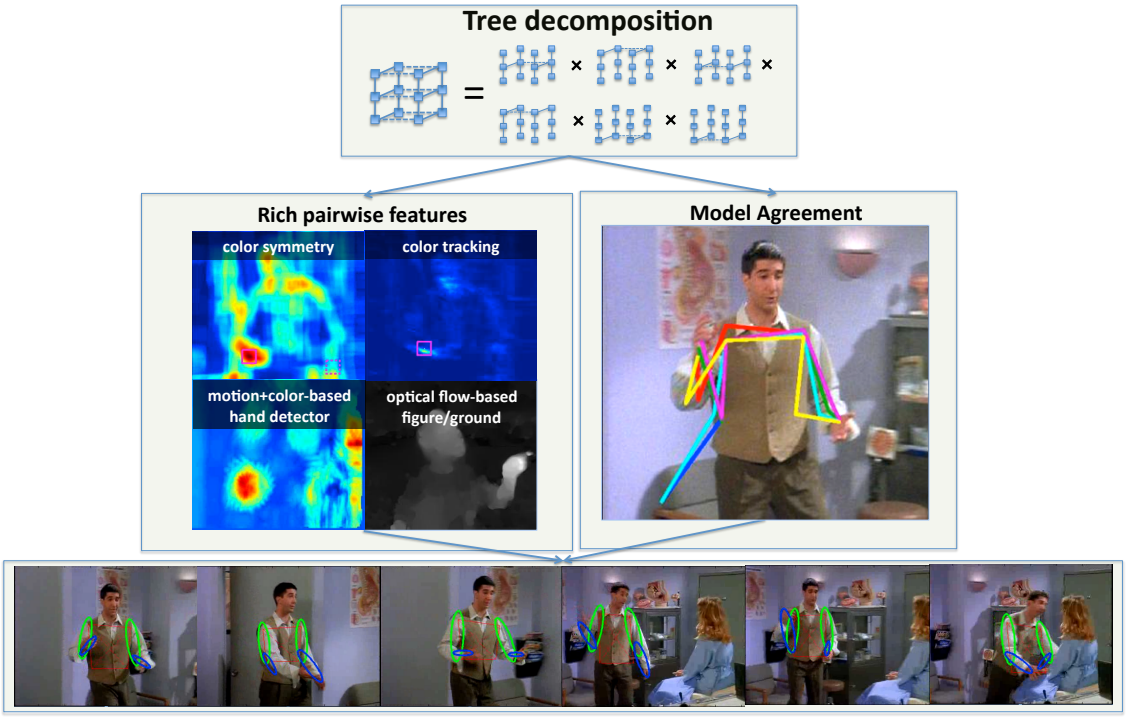


Figure 6.4: Overview of the stretchable ensembles system. The desired complete graph is decomposed into tree submodels, allowing us to design a variety of rich features, but requiring us to coordinate agreement between submodels. Middle-left: a few of the rich features employed—color symmetry shown for a left wrist patch; color tracking of a left wrist patch; hand detectors based on motion and color; optical flow motion estimate to obtain a rough cue of figure/ground separation. Middle-right: the six tree submodels' beliefs about the pose in one frame of video. While sometimes in agreement, in general they are not. Bottom row: sample frames of predicted pose once single-variable agreement is imposed. Not the tracking failure in the last frame of the left lower arm due to occlusion.

Chapter 7

Locally-linear pictorial structures

7.1 Introduction

In this chapter, we focus explicitly on the multimodal nature of the 2D pose estimation problem. As discussed in §1.2.1, there are enormous appearance variations in images of humans. This is due to foreground and background color, texture, viewpoint, and body pose. The shape of body parts is further varied by clothing, relative scale variations, and articulation (causing foreshortening, self-occlusion and physically different body contours).

Most models developed to estimate human pose in these varied settings use *only a single, linear model—e.g.*, Andriluka et al. (2009); Eichner and Ferrari (2009); Ramanan and Sminchisescu (2006); Tran and Forsyth (2010) and the models described in §5 and §6. Such models make it very difficult to capture the part variations discussed. Instead, many researchers have focused attention on improving features in hopes that a better feature space will make a linear model discriminate correct poses from incorrect ones well. However, this comes at a price of considerable feature computation cost—CPS, for example, requires computation of Pb contour detection and Normalized Cuts (see §8 for details), each of which takes minutes.

Recently, especially the past two years, there has been an explosion of successful work focused on increasing the number of modes in pose models. This line of work in general

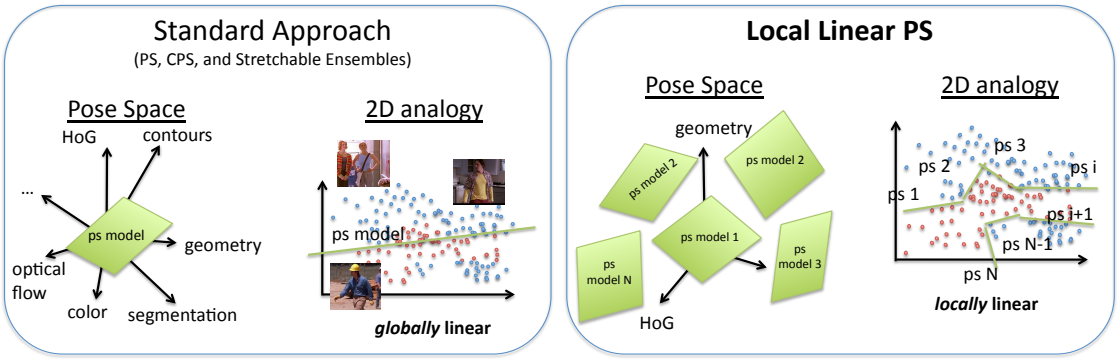


Figure 7.1: **Left:** Most pictorial structures researchers have put effort into better and larger feature spaces, in which they fit one linear model. The feature computation is expensive, and still fails at capturing the many appearance modes in real data. **Right:** We take a different approach. Rather than introduce an increasing number of features in hopes of high-dimensional linear separability, we model the non-linearities in simpler, lower dimensional feature spaces, using a collection of *locally* linear models.

can be described as instantiations of a *family of compositional, hierarchical pose models*. The family of models can be parametrized by the number of *part levels* and the number of *part modes* captured per part. The part levels can encode different granularities and scope of pose—*e.g.*, { *wrist, lower arm, full arm, half-body, full body* } are example parts in a 5-level hierarchy. The higher-level parts can leverage more image context but must generalize larger within-mode variations in pose. They serve to inform lower-level parts of likely locations. Part modes at any level can capture different poses (*e.g.*, elbow crooked, lower arm sideways) and appearance (*e.g.*, thin arm, baggy pants). Also of crucial importance are details such as how models are trained, the computational demands of inference, and how modes are defined or discovered. Details for all recent work is in Table 7.1, which we discuss at length in §7.2.

Our approach: In this chapter, we propose a particular instantiation of a multimodal model with a focus on simplicity, speed and accuracy. We capture multimodality at the global level, which gives us the ability to capture a variety of pose modes, each modeled

as a discriminative linear classifier. Thanks to the rich, multimodal nature of the model, we see performance improvements even with only computationally-cheap image gradient features.

The model features an explicit mode selection variable, the value of which is jointly inferred along with the best layout of body parts in the image at test time. Unlike some previous work, our method is trained jointly (thus avoiding difficulties calibrating different submodel confidences) and includes both holistic and local part-level cues (thus allowing it to effectively predict which mode to use). Finally, we employ an initial structured cascade mode selection step which cheaply discards unlikely modes up front, yielding a $5\times$ speedup in inference and learning over considering all modes for every example. This makes our model faster than state-of-the-art approaches (on average, 1.31 seconds vs. 1.76 seconds for Yang and Ramanan (2011)), while being significantly more accurate. It also suggests a way to scale up to even more modes as larger datasets become available.

7.2 Related work

We discuss here in depth the differences in recent works concerning multimodal models of pose. We then briefly go into other related models in the computer vision and machine learning communities.

7.2.1 Multimodal modeling

As can be seen in Table 7.1, compositional, hierarchical modeling of pose has enjoyed a lot of attention recently. In general, works either consider only global modes, local modes, or several multimodal levels of parts.

Global modes but local cues. Some works consider models with a number of distinct global modes (Johnson and Everingham, 2011; Wang and Mori, 2008; Zhu and Ramanan, 2012), enumerating over tens of disjoint models and taking the highest scoring. One characteristic of all previous models in this category is that they only employ local part

model	# part levels	# global modes	# part modes	training obj.	graph type	latent modes?
basic ps	1	1	1	parsing	tree	no
Wang and Mori (2008)	1	3	1	parsing	tree	no
Johnson and Everingham (2011)	1	16	4*	parsing	tree	yes
Yang and Ramanan (2011)	1	1	4 to 6	detection	tree	yes
Wang et al. (2011b)	4	1	5 to 20	parsing	cyclic	no
Sun and Savarese (2011)	4	1	4	parsing	tree	yes
Zhu and Ramanan (2012)	1	18	1**	detection	tree	yes
Duan et al. (2012)	4	9	4 to 6	detection	cyclic	no
Tian et al. (2012)	3	5	5 to 15	detection	tree	yes
LLPS (ours)	2	32	1	both	tree	no

* Part modes are not explicitly part of the state, but instead are maxed over to form a single detection.

** A variety of parameter sharing across global models is explored, thus there is some multimodal information used during training.

Table 7.1: In the past few years, there have been many instantiations of the family of multimodal models. The recent models listed here and their attributes are described in the text.

cues (*e.g.* wrist patch or lower arm patch), making it difficult to adequately represent and predict the best global mode. A second issue is that some sort of model calibration is required, so that scores of the different models are comparable. One way to do this is to train the models jointly by explicitly reasoning over a mode variable, as we do in our model. Johnson and Everingham (2011) instead calibrate the models post-hoc using cross-validation data, similar to Malisiewicz et al. (2011).

Local modes. A second approach is to focus on modeling modes only at the part level, *e.g.* Yang and Ramanan (2011). If n parts each use k modes, this effectively gives up to k^n different instantiations of modes for the complete model through mixing-and-matching part modes. Although combinatorially rich, this approach also lacks the ability to reason

about larger structure, not just from lack of global cues, but the inability of the representation to reason about larger structures such as global modes. A second issue is that to reason about a pair of neighboring nodes, one must consider a quadratic number of local mode combinations—*e.g.* for each of k wrist types, k elbow types must be considered, resulting in inference message passing that is k^2 larger than unimodal inference.

Additional part levels. A third category of models consider both global, local and intermediate level modes (Duan et al., 2012; Sun and Savarese, 2011; Tian et al., 2012; Wang et al., 2011b). At all levels, cues for the level are used, allowing the model to effectively capture mode information at different levels of part granularity. The biggest downside to this approach is that it is slow: First, quadratic mode inference is necessary, as with any local mode modeling. Second, inference grows linearly with the number of additional parts, and becomes intractable when part relations are cyclic, as in Duan et al. (2012); Wang et al. (2011b).

Defining modes. The mode definitions are typically obtained by clustering in the space of part configurations. Johnson and Everingham (2011) include appearance in the definition of a mode. In roughly half of the previous literature, local modes are considered latent variables to be re-estimated during training. This renders otherwise convex learning formulations non-convex, iterative learning procedures, where good mode initialization is crucial to converge to a good local minima.

Contrast with our model. In contrast to the above, our model supports multimodal reasoning at the global level, as in (Johnson and Everingham, 2011; Wang and Mori, 2008; Zhu and Ramanan, 2012). Unlike those, we explictly reason about, represent cues for, and jointly learn to predict the correct global mode as well as location of parts. Unlike local mode models such as Yang and Ramanan (2011), we do not require quadratic part-mode inference and can reason about larger structures. Finally, unlike models with additional part levels, ours is fast: a simple two-level model with no local modes and tractable inference. This also relieves us of the guesswork inherent in learning latent modes. We also apply a structured prediction mode filtering step to reduce the number of modes considered

for each test image.

7.2.2 Other models

Throwing features at the problem: Equipped with a linear model of human pose, most researchers focus attention instead on increasing the quality of features, to be robust to the variety of modes discussed above—lighting-invariant texture modeling (Andriluka et al., 2009), foreground and skin color (Eichner and Ferrari, 2009; Ramanan and Sminchisescu, 2006), left-right appearance similarity (Sapp et al., 2011; Tran and Forsyth, 2010), contour and segmentation support (Sapp et al., 2010b, 2011), and more detailed description of 2D geometry (Sapp et al., 2011; Tran and Forsyth, 2010), to name a few. The hope in all of these models is that a linear model in a larger-dimensional feature space will be better able to separate the true pose configurations from false alarms, and mitigate the issue of non-linearity in lower-dimensional feature spaces, *e.g.*, using only edge orientation information.

Other local modeling methods: In the machine learning literature, there is a vast array of local methods for prediction. Many of these require costly parameter estimation at test time, such as locally weighted logistic regression and KNN-SVM (Zhang et al., 2006). Although we call our method locally linear, it does not estimate parameters at test time like these techniques.

Different from those, nearest-neighbor methods are powerful, but live and die by the right choice of distance function. Standard norms fare poorly in high-dimensional spaces. Learning distance functions for nearest neighbors has achieved some success, *e.g.* Large-Margin KNN (Weinberger et al., 2006), which seeks to learn a global distance function for the whole sample space. A refinement of this is to learn local distance functions— (Frome et al., 2007) and the recent Exemplar-SVM (Malisiewicz et al., 2011) both learn distance functions per example. These works are quite similar in spirit to ours, but focus on object classification and detection, not structured, articulated part localization.

7.3 LLPS

We pose the problem of 2D human pose estimation as a structured prediction task. Let x represent a given input image, and y represent the location of P parts in image coordinates. Each variable y_i denotes the pixel coordinates (row, column) of part i in image x. For "parts" we choose to model joints and their midpoints (*e.g.*, left wrist, left forearm, left elbow) which allows us fine-grained encoding of foreshortening and rotation, as is done in (Sapp et al., 2011; Yang and Ramanan, 2011).

The standard pictorial structures model described in §3 is a linear model which decomposes into a sum of unary and pairwise linear terms. We choose a general pairwise MRF form in which the score for a part configuration specified by y in image x is:

$$s(x, y) = \sum_{i \in \mathcal{V}} \mathbf{w}_i \cdot \mathbf{f}_i(x, y_i) + \sum_{(i,j) \in \mathcal{E}} \mathbf{w}_{ij} \cdot \mathbf{f}_{ij}(x, y_i, y_j), \tag{7.1}$$

When the graph $G = (\mathcal{V}, \mathcal{E})$ describes a tree, we can use efficient dynamic programming techniques to infer the best scoring configuration of all parts (§2.3), $y^\star = \arg\max_{y \in \mathcal{Y}} s(x, y)$. The set $\mathcal{Y}$ denotes the entire set of possible poses, which is exponential in the number of model parts: $|\mathcal{Y}| = |\mathcal{Y}_i|^P$, where $\mathcal{Y}_i$ is the set of possible placements of part i in the image (and is the same for all i).

Instead of a single, linear model as in Equation 7.1, we model human pose with a collection of linear models which describe different local neighborhoods. Let us consider M such models, which we index $z = 1 \ldots M$:

$$s^z(x, y) = \mathbf{w}^z \cdot \mathbf{f}(x, z) + \sum_{i \in \mathcal{V}} \mathbf{w}_i^z \cdot \mathbf{f}_i(x, y_i, z) + \sum_{(i,j) \in \mathcal{E}} \mathbf{w}_{ij}^z \cdot \mathbf{f}_{ij}(y_i, y_j, z), \tag{7.2}$$

These submodels include the additional, global term $\mathbf{w}^z \cdot \mathbf{f}(x, z)$ which help discriminate the submodels from others using global features of the image. Each model shares the same tree structure $\mathcal{E}$ for the sake of simplicity; it is easy to extend this to a heterogenous collection of tree models. Note the pairwise term we express only as a function of the state-space, which allows us to perform inference linear in each part's state-space using a

distance transform (§3.1). The M models' parameters $\mathbf{w}^z = [\mathbf{w}_i^z;\ \mathbf{w}_{ij}^z]$ are learned to fit a local neighborhood around each mode center, discussed in §7.4.

At test time, the full LLPS models infers both the best local model z to use, and the best placement of joints given the corresponding submodel:

$$s(x, y, z) = s^z(x, y) \tag{7.3}$$

$$z^\star, y^\star = \underset{z\in[1,M],\ y\in\mathcal{Y}}{\arg\max}\ s(x, y, z) \tag{7.4}$$

Thus, given a test example, the test time inference procedure is straightforward: evaluate all M local submodels (in practice, we use 32 modes for upper body pose estimation), via max-sum inference (§2.3), saving the score and highest scoring output sequence of each. Then, we take the highest scoring of the M local models and its output sequence as a prediction of the pose. This procedure is linear in M. In the next section we show a superlinear speedup using cascaded prediction.

7.3.1 Cascaded mode filtering

The use of structured prediction cascades has been a successful tool for drastically reducing state spaces in structured problems. In §5 and §6, the cascade method was used to reduce the number of locations to consider for human parsing. Here we employ a simple multiclass cascade step to reduce the number of modes considered in the full $LLPS$ model. We employ a cascade model of the form

$$c(x, z) = \theta^z \cdot \phi(x, z) \tag{7.5}$$

whose purpose is to score the mode z in image x, in order to filter unlikely mode candidates. The features of the model are $\phi(x, z)$ which capture the pose mode as a whole instead of individual local parts, and the parameters of the model are a linear set of weights for each mode, θ^z. Following the cascade framework, we retain a set of mode possibilities $\bar{M} \subseteq [1, M]$ after applying the cascade model:

$$\bar{M} = \{z \mid c(x, z) \geq \alpha \max_{z\in[1,M]} c(x, z) + \frac{\alpha - 1}{M} \sum_{z\in[1,M]} c(x, z)\} \tag{7.6}$$

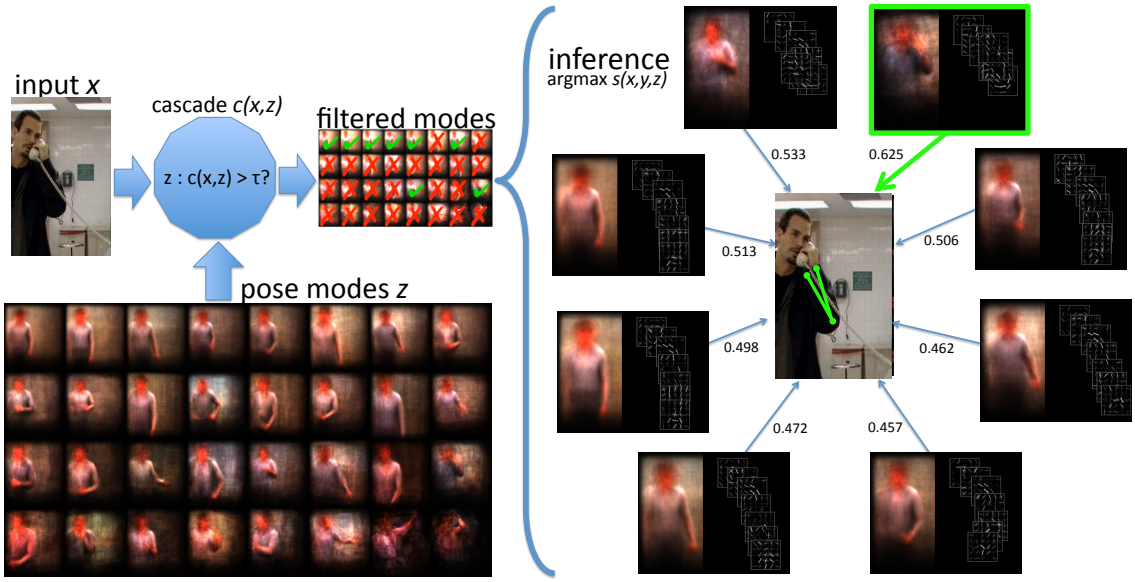


Figure 7.2: An illustration of the inference process. First, modes are cheaply filtered via a cascaded prediction step. Then each remaining local submodel can be run in parallel on a test image, and the argmax prediction is taken as a guess.

The metaparameter $\alpha \in [0, 1)$ is set via cross-validation and dictates how aggressively to prune—between pruning everything but the max scoring mode to pruning everything below the mean score. Applying this cascade before running the final level LLPS model results in the inference task

$$z^\star, y^\star = \arg\max_{z \in \bar{M},\ y \in \mathcal{Y}} s(x, y, z) \tag{7.7}$$

where $|\bar{M}|$ is considerably smaller than M. In practice it is on average 5 times smaller, giving us a $5\times$ speedup.

The full system pipeline is laid out in Figure 7.2. Given an input image, we first quickly discard most of the possible modes via the cascade model. The remaining modes are then each applied independently on the test image. The models are trained jointly and thus their inference scores are well-calibrated. We take the highest scoring position of part layout y from submodel/mode z as the final prediction.

7.3.2 Image-adaptive pose priors and a non-parametric perspective

One important perspective of the above is as a mode *switching model*, popular in speech synthesis (Rosti and Gales, 2004). In this setting, z can be thought of as a switch which chooses between different appearance and pose priors. The particular instantiation of z depends on the image content. This is related to another, locally-linear method called Adaptive Pictorial Structures (APS) (Sapp et al., 2010a). In this work, the pose prior terms $\mathbf{f}_{ij}$ were adjusted based on a kernel-weighted sum of exemplar similarities to the image. LLPS, on the other hand, forces a discrete choice from a dictionary of pose priors, and this procedure is learned discriminatively. From a more practical standpoint, APS relied on very costly and heuristic nearest-neighbor computation. It searched densely through a large number of deformations to handle articulation, rather than making use of a structured decomposable similarity (*i.e.*, the LLPS submodel scores). Each submodel of LLPS can be thought of as a decomposable similarity function, learned discriminatively to separate true poses from wrong ones locally, as we will see next.

7.4 Learning

During training, we have access to a training set of images with labeled poses $\mathcal{D} = \{(x^t, y^t)\}_{t=1}^T$. As described, we choose to model a local neighborhood in pose and appearance centered around predefined modes in pose space. This allows us to capture fine variations of appearance and part layout due to pose.

Mode definitions. Modes are obtained from the data by finding centers $\{\mu_i\}_{i=1}^M$ and example-mode membership sets $S = \{S_i\}_{i=1}^M$ in pose space that minimize reconstruction error under squared Euclidean distance:

$$S^\star = \arg\min_S \sum_{i=1}^M \sum_{t \in S_i} ||y^t - \mu_i||^2 \tag{7.8}$$

where μ_i is the Euclidean mean of the examples in mode cluster S_i. We approximately minimize this objective with the k-means algorithm with 100 random restarts. We take

the cluster membership as our supervised definition of mode membership in each training example, so that we augment the training set to be $\mathcal{D} = \{(x^t, y^t, z^t)\}$.

Parsing constraints. We seek to learn to correctly identify the correct mode *and* location of parts in each example. Intuitively, for each example this gives us hard constraints of the form

$$s^{z^t}(x^t, y^t) - s^{z^t}(x^t, y') \geq 1, \qquad \forall y' \neq y^t \tag{7.9}$$

$$s^{z^t}(x^t, y^t) - s^{z'}(x^t, y) \geq 1, \qquad \forall z' \neq z, \forall y \tag{7.10}$$

In words, Equation 7.9 states that the score of the true configuration for local model z must be higher than z's score for any other (wrong) configuration in example t—the standard max-margin parsing constraint for a single model. Equation 7.10 states that the score of the true configuration for z must also be higher than any score a "stranger" model z' has for any pose configuration in example t. We refer to these constraints as *parsing constraints* as they deal with scoring the correct parse y^t above others in a single example t.

Detection constraints. A popular variant recently in the pose estimation literature is to use detection-style constraints instead of parsing constraints; enforcing the notion that the model score should be low when no valid parse exists in images without people (Duan et al., 2012; Tian et al., 2012; Yang and Ramanan, 2011). These are typically encoded as

$$s^{z^t}(x^t, y^t) \geq +1 \qquad \forall t \tag{7.11}$$

$$s^{z}(x^{neg}, y) \leq -1 \qquad \forall x^{neg} \in \mathcal{D}^{neg}, \forall y, z \tag{7.12}$$

where now $\mathcal{D} = \mathcal{D}^{pos} \cup \mathcal{D}^{neg}$, with $\mathcal{D}^{neg}$ a database of unannotated images with no people in them. Simply put, these constraints enforce that groundtruth parses should have a score at least 1, and parses on negative images from any model should score at most -1. Notably absent from these constraints is an explicit way to calibrate different submodels z with each other as in Equation 7.10. This type of calibration is typically done in a post-hoc heuristic manner, *e.g.* (Johnson and Everingham, 2011; Malisiewicz et al., 2011).

Detection-style constraints are easily put into our framework by adding a negative class mode z^{neg} modeled with a single constant feature. Then $s^{z^{neg}}(x, y) = \tau^{neg}$ is a learned

scalar threshold for the negative class, and for cases when z^t or $z^{t'} = z^{neg}$, Equation 7.10 can be interpreted as detection constraints

$$s^{z^t}(x^t, y^t) \geq \tau^{neg} + 1, \qquad \forall t \in \mathcal{D}^{pos}, \tag{7.13}$$

$$s^{z}(x^{neg}, y) \leq \tau^{neg} - 1, \qquad \forall x^{neg} \in \mathcal{D}^{neg}, \forall y, z \tag{7.14}$$

Learning formulation. We consider all constraints from Equation 7.9 and Equation 7.10, and include a negative mode z^{neg} to incorporate detection constraints of the form in Equation 7.13 and Equation 7.14. Adding slack and regularization on the parameters, we get a convex, large-margin structured loss jointly over all M local models

$$\min_{\{\mathbf{w}^z\},\{\xi_t\}} \quad \frac{1}{2}\sum_{z=1}^{M} ||\mathbf{w}^z||_2^2 + \frac{C}{T}\sum_{t=1}^{T}\xi_t \tag{7.15}$$

subject to: (7.16)

$$s^{z^t}(x^t, y^t) - s^{z^t}(x^t, y') \geq 1 - \xi_t \qquad \forall y' \neq y^t \tag{7.17}$$

$$s^{z^t}(x^t, y^t) - s^{z'}(x^t, y) \geq 1 - \xi_t \qquad \forall z' \neq z \in \bar{M}^t, \forall y \tag{7.18}$$

Note the use of $\bar{M}^t$, the subset of modes unfiltered by our mode prediction cascade for each example. This is considerably faster than considering all M modes in each training example.

The number of constraints listed here is prohibitively large: in even a single image, the number of possible outputs $y' \in \mathcal{Y}$ is exponential in the number of parts. Standard techniques to minimize structured SVM objectives like this are stochastic gradient descent and cutting plane methods. We use a cutting plane technique where we find the most violated constraint in every training example via structured inference. We then solve Equation 7.15 under the active set of constraints using fast off-the-shelf QP solver LIBLINEAR (Fan et al., 2008).

7.5 Modeling human pose with LLPS

In §7.3 and §7.4, we detailed a general learning and inference framework for local linear structured models. We now fill in necessary details on the graph structure, neighborhood function, and features to describe how we apply this model to human pose estimation.

7.5.1 Local graph structure

In order to effectively use training data, we model only one side of a human— the nose, left shoulder, left elbow and left wrist. This allows us to re-use training data for the left and right sides by horizontal mirroring, and also makes inference faster since we have a simple chain graph connecting each joint. We also insert midpoint nodes into the graph between the semantic joints—between the nose and shoulder (capturing neck curvature), the upper arm, and the forearm, similar to Yang and Ramanan (2011), and thus have $n = 7$ parts in each local linear structured model.

The decision to split the modeling into sides is also justified empirically. Localization accuracy for detecting the torso and head of a person in pose datasets is near-perfect, thus there is virtually no useful signal for the left side of a person to send to the right side through the torso and head, regarding beliefs of where parts should go. In other words, variables on the left and right side of the person are *d-separated* (Koller and Friedman, 2009) given the torso and head variables, which are observed essentially deterministically.

7.5.2 Features

Appearance. Unlike our previous CPS and Stretchable Ensemble models, we utilize only histogram of gradients (HoG) descriptors, using the implementation from Felzenszwalb et al. (2011). For local cues $\mathbf{f}_i(x, y_i, z)$ we use a 5×5 grid of HoG cells, with a cell size of 8×8 pixels. For global cues $\mathbf{f}(x, z)$ we capture larger structure with a 9×9 grid and a cellsize of 16×16 pixels. The cascade mode predictor uses the same features in a 17×15 grid with a cell size of 8×8 pixels.

Geometry. We use the same quadratic deformation cost features as Felzenszwalb and Huttenlocher (2005), allowing us to use distance transforms for message passing (§3.1):

$$\mathbf{f}_{ij}(y_i, y_j, z) = \left[(y_i(r) - y_j(r) - \delta_{ij}^z(r))^2 \;\; (y_i(c) - y_j(c) - \delta_{ij}^z(c))^2\right]^T$$

where $(y_i(r), y_i(c))$ denote the row and column represented by state y_i, and δ_{ij}^z is the mean displacement between parts i and j in mode z. In order to make the deformation cue a convex, unimodal penalty (and thus computable with distance transforms), we need to ensure that the corresponding parameters on these features $\mathbf{w}_{ij}^z$ are positive. We enforce this by adding additional positivity constraints in our learning objective: $\mathbf{w}_{ij}^z \geq \epsilon$, for small ϵ strictly positive. It may still be the case that the constraints are not respected (due to slack variables), but this is rare. In the unlikely event that this occurs, we project the deformation parameters onto the feasible set: $\mathbf{w}_{ij}^z \leftarrow \max(\epsilon, \mathbf{w}_{ij}^z)$

Part III

Representations of 2D human pose

Chapter 8

Feature Sources

A major goal of this work was to introduce rich features into models of human pose. The standard approach is to represent limbs with an edge-based rigid template invariant to color, lighting, and minor deformations. In addition to this, we use color, shape and geometry.

In this chapter, we go over feature *sources* - the basic channels of input and how they are obtained. For implementation details of how these are used in the various models, see §9.

8.1 Edges

The standard cue for pictorial structures is rigid templates based on edges (Andriluka et al., 2009; Eichner and Ferrari, 2009; Ferrari et al., 2008; Ramanan and Sminchisescu, 2006; Tran and Forsyth, 2010; Yang and Ramanan, 2011). Edges provide invariance to lighting bias and color. Robust descriptors on top of edges provide additional invariance to lighting changes and slight translations, by coarsely binning edge magnitude and orientation, possibly with local edge energy normalization. Andriluka et al. (2009) used Shape Context (Belongie et al., 2001) on top of Canny edge detection for a sparse representation which handles angular deformation well by binning radially.

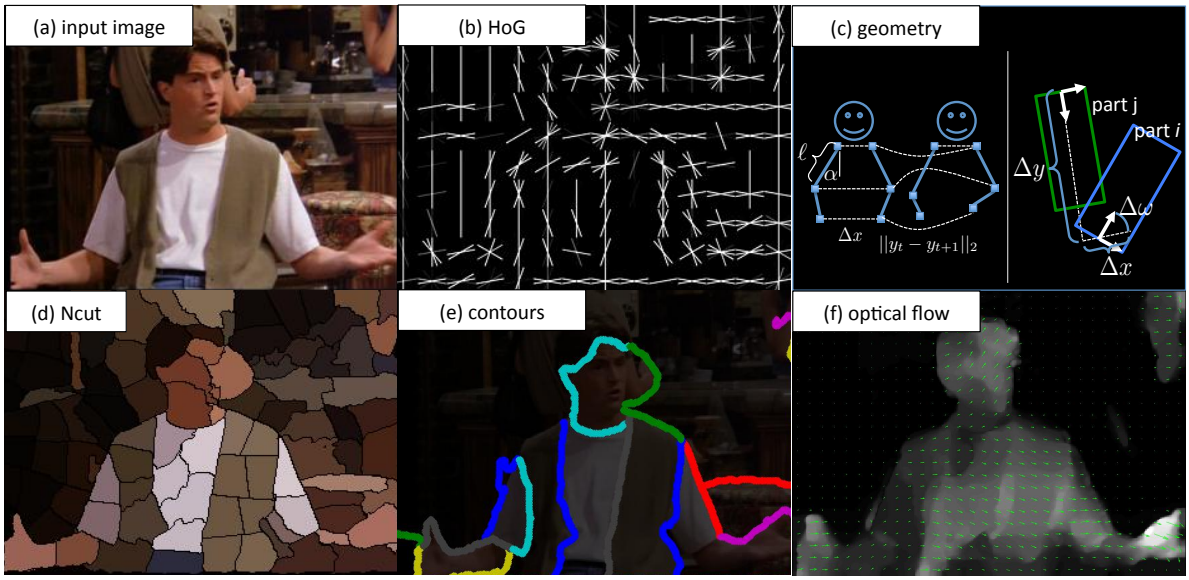


Figure 8.1: Feature sources. (a) Running example input image. (b) Coarse HoG representation. (c) Different geometric quantities we compute for our models. (d) Normalized cuts. (e) Long contours derived from Ncut segment boundaries. (f) Optical flow field shown sparsely as a quiver plot over the dense magnitude of the optical flow field at every pixel.

In all our work, and many others, we use an unsigned histogram of gradients (HoG) descriptor, which partitions the image into a coarse uniform grid of cells, and measures the edge energy discretized into the uniform grid of spatial bins (cells) and 9 unsigned orientation bins. HoG was originally proposed by Dalal and Triggs (2005a) for pedestrian detection, and has been hugely popular in the object recognition community as a whole, *e.g.* Felzenszwalb et al. (2008). We use a modified version of the implementation by Felzenszwalb et al. (2008), changed to use only unsigned orientations and no color. See Figure 8.1-b.

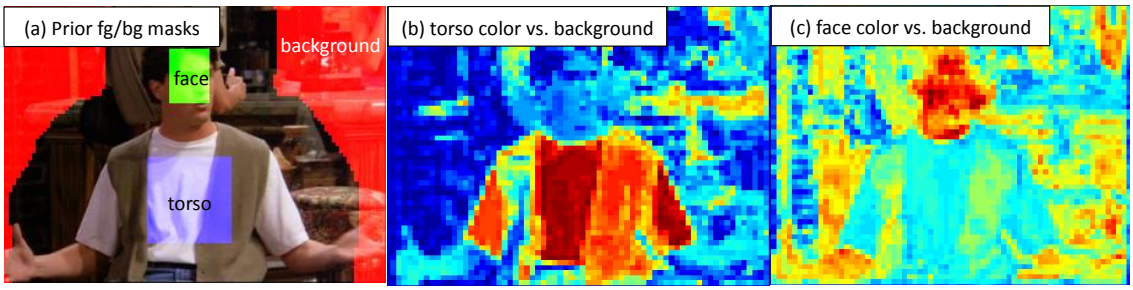


Figure 8.2: Foreground color estimation. (a) Masks obtained from groundtruth locations of face, torso and background *a priori*, overlaid on an example image. (b,c) Output of classifier learned to discriminate face versus background and torso versus background independently for every pixel.

8.2 Color

Color is a tricky cue to exploit, being highly variable from image to image. However, having knowledge of the foreground and/or background color is powerful information that enables us to separate unlikely poses from likely ones—we can exploit such assumptions as figure/ground color separation and foreground constancy across the body and through time.

Foreground/Background modeling It is difficult to model the foreground or background color without first *knowing* where the foreground and background are—"a chicken and egg problem". Some works such as Eichner and Ferrari (2009); Ramanan and Sminchisescu (2006) focus on this issue directly by iteratively re-estimating color and re-estimating pose.

We choose a simple and effective method inspired by Ramanan et al. (2005) to obtain foreground skin and clothing color models in a single pre-processing step. We exploit the fact that humans are quite well-localized by people detectors (Ferrari et al., 2008), and assume we have samples of pixels from the face, torso and (importantly) background. Figure 8.2 shows a mask of pixels which we sample and corresponding labels. Treating these samples as *groundtruth*, we model color using a supervised binary classifier (§2)

for skin and torso color separately, using the assumed {face, torso} pixels as positive examples, and assumed background pixels as negative examples. Each pixel is its own example, classified independently.

Because we have few samples that do not completely describe the range of foreground and background colors, and because the feature space is relatively small (the space of colors), it is quite easy to overfit a discriminative foreground/background color model to the training samples. This leads to poor generalization when we apply the learned model. Using validation data, we found that a scheme of learning the models using an Adaboost classifier composed of 10 decision trees each of maximum depth 2 (Friedman et al., 2001) worked well in practice. To complete our input foreground/background color estimation feature source, we apply the learned face and torso color classifiers to all pixels in the image independently (including the training sample pixels).

Color similarity and tracking It is also useful to *compare* the color of parts in a relative sense, for the purposes of color constancy between limbs and joints within a frame and across time. For this, we need a description of color and a notion of distances that are robust to fluctuations in lighting, hue and saturation. Because we are only concerned with local color similarity here (between patches in a single or consecutive frames), we choose a vector quantization to partition the colorspace into a coarse colormap. Specifically, we choose the method of Heckbert (1982)—`rgb2ind(·)` in MATLAB—which greedily chooses bisections of RGB space to minimize variance. This representation is not only robust to color fluctuations, it is also fast to compute and lends itself easily to robust histogram comparison distances, as explained below.

8.3 Shape

Shape representations have the potential to model very generic yet discriminative descriptions of objects. However, they have seen limited success in the object recognition community, typically because the representations are fragile and confounded by real-world

clutter, *e.g*. Sebastian et al. (2004). Furthermore, much of the shape community is inspired by the study of early biological vision and psychophysics, where the focus is on *bottom-up processing*: finding objects or pieces of objects in a generic way—without an explicit model of any specific object—based on perceptual grouping principles such as symmetry, contour continuity and parallelism, *e.g*. Biederman (1987).

In contrast, the object recognition community has relied heavily on data-driven discriminative models on top of edge template representations (Everingham et al., 2010), such as the HoG representation discussed in §8.1, and others (Dalal and Triggs, 2005a; Felzenszwalb et al., 2011; Viola and Jones, 2002). These models leverage powerful machine learning machinery to distinguish foreground from background clutter, but the rigid edge template representation is brittle to intrinsic deformations and extrinsic rotation, translation and scale.

We seek to leverage the representational power of bottom-up grouping cues in our top-down model of human pose. We look at shape descriptors based on region and contour information. Recently other works have focused on using purely bottom-up shape representations in a top-down matching framework, such as Carreira et al. (2011); Toshev et al. (2010); Zhu et al. (2008). Also, some of the early work on human pose estimation was purely bottom-up driven, based on heuristics and hand-crafted rules to propose and connect limbs, *e.g*. Mori et al. (2004); Srinivasan and Shi (2007).

We choose to use Multiscale Normalized Cut (Ncut) (Cour et al., 2005) to obtain regions. Ncut seeks a balanced partitioning of the image into meaningful regions via a spectral decomposition of a graph over image pixels, based on pairwise pixel affinities. See Figure 8.1-d.

We obtain long continuous contours in the image from the regions obtained by Ncut, as follows. We define a graph whose nodes are all boundaries between Ncut segments with edges linking touching boundaries. Each contour is a path in this graph. To reduce the number of possible paths, we restrict ourselves to all shortest paths. To quantify the smoothness of a contour, we compute an angle between each two touching segment

boundaries[1]. The smoothness of a contour is quantified as the maximum angle between boundaries along this contour. Finally, we find among all shortest paths those whose length exceeds 18% of the estimated person's upper body and whose smoothness angle is less then $45°$ to obtain a final set of contours, typically 15 to 30 per image. See Figure 8.1-e.

8.4 Geometry

As discussed in §3, most models of human pose consider only a unimodal function of the displacement between parts of the form $d(y_i - y_j - \delta_{ij})$, where δ_{ij} is the expected displacement between parts, and $d(\cdot)$ is typically squared Euclidean distance in a linearly transformed space. In our CPS and Stretchable Ensembles models we are not restricted to any analytical form. Thus we encode features based on limb lengths (Euclidean distance in pixel coordinates), relative angles between parts, absolute angles that limbs make with the horizontal, and absolute locations of limbs and joints. The "absolute" measurements are really with respect to the detected person bounding box, which can be thought of as a simple virtual "root" part which is always instantiated (observed) in our models. See Figure 8.1-c.

8.5 Motion

Motion gives us cues to separate foreground from background, and fast moving parts (*e.g.*, hands) from more stationary ones (*e.g.* head). When consecutive frames of video are available, we obtain motion estimation from optical flow. Specifically, we use the fast implementation from Liu (2009). See Figure 8.1-f.

[1]The angle computed is between straight lines fit to the last third of each segment boundary

Chapter 9

Features

There are several possible ways to organize the list of features we use in our systems—by model, by modality, by importance. We choose to group them here by variable interactions, in keeping with the story of this thesis. To make clear which features are used in which models, we mark each feature with [CPS] (§5), [ESM] (§6), [LLPS] (§7), or some combination of the three.

9.1 Discretization

All the features described in this chapter come in a variety of units, types and ranges, both categorical and real. Typically in linear models performance suffers significantly when features are at very different scales (Fan et al., 2008). In order to incorporate these into a linear pairwise MRF model in a flexible manner, we *uniformly discretize* all the features discussed, mapping them to one of b bins. When a feature is real-valued, $b = 10$ with bin boundaries uniformly spaced between the 10^{th} and 90^{th} percentile of the feature's value over the training set. When a feature is categorical, b is equal to the number of categories. In all cases, each scalar feature gets mapped to a binary vector of length b, with exactly one entry that is 1, indicating bin membership, and the rest 0. This allows us to learn non-linear, multi-modal mappings of scalar features, as the weights of each

bin are unconstrained. The sparse representation allows us to compute $\mathbf{w}_i \cdot \mathbf{f}_i(x, y_i)$ and $\mathbf{w}_{ij} \cdot \mathbf{f}(x, y_i, y_j)$ quickly for all factors.

The last stage of CPS uses its own form of discretization, by way of learning decision tree stumps on raw feature values in a boosting procedure. Details are in §10.4.

9.2 Limb-pair features

Color similarity [CPS:]: We measure the color similarity of kinematically-connected upper and lower arms via χ^2-distance between histograms of quantized RGB color, described in §8.2. The histograms are accumulated within rectangles describing the limb extent. For histograms q and p, with bins indexed by p_i and q_i, the distance is defined as

$$\chi^2(p, q) = \frac{1}{2} \frac{\sum_i (p_i - q_i)^2}{\sum_i p_i + q_i}.$$

We discretize the colorspace into only 8 bins to histogram, giving us a stable but crude representation of color. The coarseness of the quantization is permissible because we only need to model a local region of a single image around a detected person.

Contour support [CPS]: We can use the set of contours $\{c_k\}$ obtained as described in §8.3 to define features for each pair of lower and upper arms, which encode the notion that those two parts should share a long smooth contour, which is parallel and close to the part boundaries. For each arm part y_i and a contour c_k we can estimate the edges of c_k which lie inside one of the halves of the supporting rectangle of y_i and whose edge normals build an angle smaller than ω with the normal of the part axis. We denote the number of those edges by $q_{ik}(\omega)$. Intuitively, a contour supports a limb if it is mostly parallel and enclosed in one of the limb sides, i.e. the value $q_{ik}(\omega)$ is large for small angles ω. A pair of arm limbs y_i, y_j should have a high score if both parts are supported by a contour c_k, which can be expressed as the following two scores

$$\text{cc}^{(1)}_{ijk}(\omega, \omega') = \frac{1}{2} \left(\frac{q_{ik}(\omega)}{h_i} + \frac{q_{jk}(\omega')}{h_j} \right) \quad \text{and} \quad \text{cc}^{(2)}_{ijk}(\omega, \omega') = \min \left\{ \frac{q_{ik}(\omega)}{h_i}, \frac{q_{jk}(\omega')}{h_j} \right\}$$

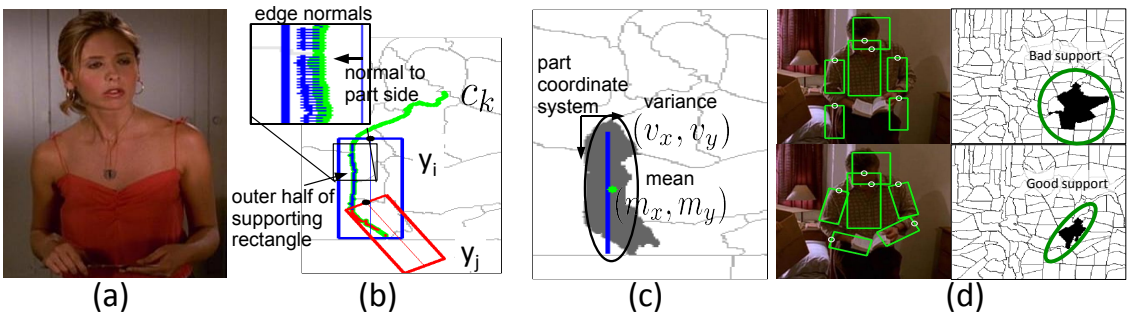


Figure 9.1: Shape features. (a) Input image. (b) Measuring a pair of limbs co-likelihood by checking attachment to long contours in the image, as described in the "Contour support" feature. (c) Measuring the likelihood of a single limb's region support, as described in the "Region support" feature. (d) Examples of right lower arm hypotheses that have poor support from underlying regions (top), and good support (bottom).

where we normalize q_{ik} by the length of the limb h_i to ensure that the score is in $[0, 1]$. The first score measures the overall support of the parts, while the second measures the minimum support. Hence, for y_i, y_j we can find the highest score among all contours, which expresses the highest degree of support which this pair of arms can receive from any of the image contours:

$$\text{cc}_{ij}^{(t)}(\omega, \omega') = \max_k \ \text{cc}_{ijk}^{(t)}(\omega, \omega'), \quad \text{for} \quad t \in \{1, 2\}$$

By varying the angles ω and ω' in a set of admissible angles Ω defining parallelism between the part and the contour, we obtain $|\Omega|^2$ contour features[1] The features are illustrated in Figure 9.1-(b).

Geometry [CPS]: We measure the interior angle made between parts, keeping the part identity so that the angle is in the full range $[0, 2\pi]$. We also look at difference from the expected position δ_{ij} for a pair of parts, using the signed x and y components as features.

[1] We set $\Omega = \{10°, 20°, 30°\}$, which results in 18 features for both scores.

9.3 Joint-pair features

HoG limb detectors [CPS, ESM]: For each of the 6 body parts – head, torso, upper and lower arms – we learn an individual Gentleboost classifier Friedman et al. (2000) on HoG features (§8.1) using the Limbs Annotated in Movies Dataset[2]. We then apply these detectors densely over all orientations and locations and downsample by max-pooling or upsample via nearest-neighbor to get a heatmap of beliefs over our state space.

Region support [CPS, ESM]: We compute the first and second order moments of the segments lying under the major part axis[3] to coarsely express shape of limb hypotheses as a collection of segments R, the union of all segments that support the part hypothesis. To achieve rotation and translation invariance, we compute the moments in the part coordinate system. We include convexity information $|conv(R)|/|R|$, where $conv(\cdot)$ is the convex hull of a set of points, and $|R|$ is the number of points in the collection of segments. We also include the number of points on the convex hull, and the number of part axis points that pass through R to express continuity along the part axis. See Figure 9.1-b,c.

Foreground color [CPS, ESM]: Using likelihood heatmaps of estimated skin and torso color (§8.2), we measure the likelihood that a limb is present at a particular location by a mean heatmap value inside the rectangular extent of a part, for each color model (face and torso). This can be computed efficiently densely via convolution with an all-ones filter, separable over dimensions, for each orientation. If desired, the feature can be computed efficiently over a sparse set of locations using integral images (Viola and Jones, 2002).

Geometry [CPS, ESM, LLPS]: To describe the geometry of a limb in CPS, we use as features its absolute position and orientation with respect to the image axes. Position is discretized into a coarse spatial grid, and orientation is discretized into 24 possible values; both categorical features.

In the Stretchable Ensemble Model, we use the following pairwise geometric features for limbs in a single frame: (i) length of arms, (ii) unsigned difference in angle that the

[2]LAMDa is available at *http://vision.grasp.upenn.edu/video*

[3]We select segments which cover at least 25% of the part axis.

upper arm makes with respect to the vertical axis, (iii) difference in x-coordinate between left-right symmetric joints, from which our model can learn a type of repulsion behavior (e.g., the left shoulder should be far from the right shoulder) as well as a left-right order of parts (e.g., the left shoulder should be to the left of the right shoulder). Note that we do *not* express features describing the angle of the lower arm, leaving it free to rotate and stretch. See Figure 9.2-d.

To express joint motion continuity through time in ESM, we use one simple geometric feature: the Euclidean distance between the joint in consecutive frames, discretized into 10 binary values. In LLPS, we use standard quadratic deformation features as in §3.1 (Felzenszwalb and Huttenlocher, 2005).

Contour support [ESM]: For each arm joint-pair, we measure its support from long contours extracted in the image (§8.3) as follows: we take the number of contour points that are roughly parallel to the joint-pair line segment (angle less than $12°$) and within a spatial support region (approximately 25% of the length of the average limb). We then use as a feature the number of supporting contour points, normalized by the length of the limb hypothesis. Due to the sparse contour set, this feature can be computed extremely efficiently, by quickly discarding hypotheses whose endpoints are not near any contour. See Figure 9.2-c.

Color consistency [ESM]: To capture the fact that the color of pairs of joints is often similar due to clothing and/or skin color, we describe the color consistency of pairs of joints via the χ^2-distance between color histograms obtained from small image patches (with side length 10% of image dimensions) extracted around each joint. See Figure 9.2-b.

Color tracking [ESM]: We capture the persistence of appearance over time with a simple and effective patch-based color tracker for each joint. We jointly quantize each pair of consecutive images into a small number of color indices, using minimum variance quantization (§8.2)—here with 32 colors. We then compare patches (of side length 10% of the image dimensions) around the joint in each frame using an L_0-norm (Hamming

loss) distance function: Let P_t and P_{t+1} represent quantized color image patches around the joint in frame t and $t+1$, with N pixels. Then the color tracking distance feature is $||P_t - P_{t+1}||_0 = \frac{1}{N} \sum_{(r,c)} \mathbf{1}[P_t(r,c) = P_{t+1}(r,c)]$ where (r,c) indexes rows and columns in the patch. See Figure 9.2-b.

The coarse quantization gives us a robust way to compare the color in consecutive frames that is tolerant to changes in lighting and pixel fluctuation. The L_0 distance is more robust than Euclidean distance/radial basis similarity scores often used (*e.g.* Gould et al. (2008)), in which the distance grows with the distance in the color space and can thus be corrupted by a few large outlying pixels. Furthermore, it is a much more discriminative cue than color histogram distances across time (as used in *e.g.* Ren and Malik (2007)) because it requires that the *pixel structure* of the patch be similar in consecutive frames.

Figure from flow [ESM]: We use several features based on dense optical flow from Liu (2009), which we compute between adjacent frames. We obtain a rough estimate of the foreground of each clip by assuming there are only 2 planes of motion: foreground (figure) and background. Given a detected person, we estimate the background motion by computing the median flow vector μ_{bg} outside the detected person bounding box, and not considering outlier flow with magnitude greater than the 75^{th} percentile. We then subtract off μ_{bg} from the flow field and take the magnitude of the flow as an estimate of foreground likelihood. We incorporate this as a limb feature by computing the average foreground likelihood sampled at 5 points evenly spaced along the line segment between joint pairs. See Figure 9.2-e.

9.4 Single joint features

Histogram of Gradients [ESM, LLPS]:

In ESM we have no specific learned model of joints, so we project down trained limb detectors from §9.3, taking a max over all possible angles and over 3 possible limb lengths

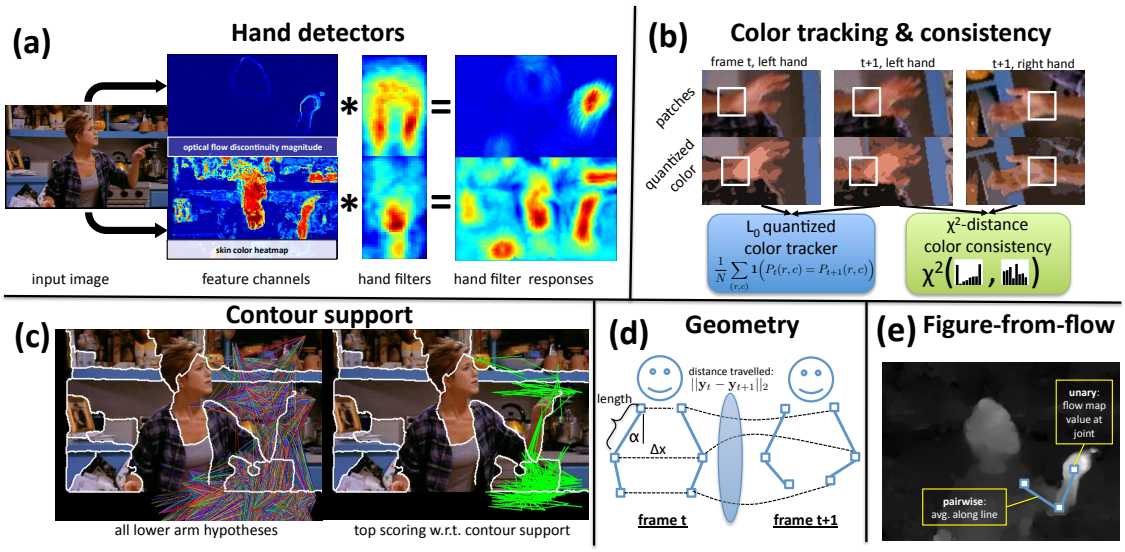


Figure 9.2: Joint and joint-pair features. See §9.3 and §9.4. (a) Hand detectors. (b) Color distances for left-right symmetry and tracking, using a quantized color space (shown). (c) Visualization of arm hypotheses that are supported by contours in the image. (d) Joint-joint geometry we employ in our Stretchable Ensemble model. (e) Figure-ground estimation by estimating two flow clusters.

(not foreshortened, somewhat foreshortened, extremely foreshortened; estimated by quantiles of dataset groundtruth limbs).

In LPPS, we represent joints with 5×5 cell Histogram of Gradients descriptors around each joint, using the implementation from Felzenszwalb et al. (2011), modified to use only 9 un-signed edge orientations. Using these features, learned parameters capture the local shape around each joint, *specific to* the variations of pose and appearance in a small local neighborhood. We supplement the HoG channels with an extra "out-of-image" channel, to jointly learn if it's likely that a body part lies at least partially outside the image coordinates.

Color and motion-based hand detectors [ESM]: Detecting the hand location is an extremely useful cue in anchoring all joint locations. Unfortunately, traditional template-based part detectors such as HoG detectors fail at this task due to hands' high variability

in appearance, pose, and motion blur. We instead learn discriminative linear filters (Fan et al., 2008) from two different input modalities: (1) Skin detection heatmaps. (§8.2). (2) The gradient of the magnitude of the optical flow field between consecutive images (§8.5). The latter exploits the fact that hands are often in motion (and naturally the fastest moving body part). Both linear filters can be evaluated efficiently densely via convolution. See Figure 9.2-a.

Figure from flow [ESM]: We again use an estimate of foreground via optical flow magnitude as in §9.3, and Figure 9.2-e. Here we report a mean foreground likelihood in a small window around the joint as a feature.

Part IV

Experiments

Chapter 10

Methodology

> What I cannot create, I do not understand.
>
> — Richard Feynman

The main contributions of this thesis are new models of pose which allow us to incorporate new image representations. In this part we show empirically that the new representations are worthwhile by comparing them to state-of-the-art methods on competitive benchmark datasets. Furthermore, we also analyze the trade-offs of the different models and the features' effectiveness.

10.1 Datasets

We evaluate our methods on four publicly available datasets; two developed through the course of this research.

10.1.1 Buffy Stickmen

This dataset was first introduced in Ferrari et al. (2008). We use version 2.1 in our experiments, consisting of 748 frames from the TV show Buffy the Vampire Slayer, from

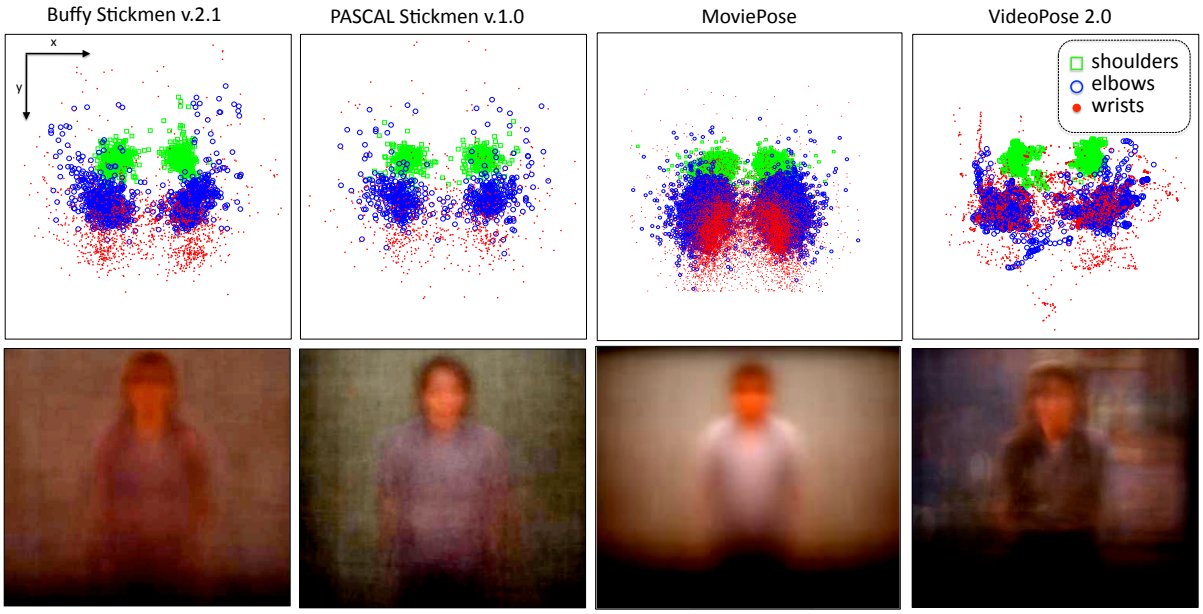


Figure 10.1: Dataset joint scatterplots and pixel averages. Here we show the spatial distribution of wrist, elbow and shoulder joints for our three datasets, both training and test, top row. Bottom row, the average images from each of the datasets, with manually enhanced brightness and contrast for better visualization.

episodes 2, 4, 5 and 6 from season 5. The standard protocol is to test on a given set of 235 frames that were correctly localized by an upper body detector in Ferrari et al. (2008)—within 50% overlap with the groundtruth. For training images where an upper body detector did not detect a person for which we have annotated limbs, we loosely annotated the upper body manually to simulate test time behavior.

10.1.2 PASCAL Stickmen

This dataset was first introduced by Eichner and Ferrari (2009) and contains 360 examples (version 1.0) obtained from amateur photographs culled from the PASCAL VOC 2008 challenge (Everingham et al., 2009). This is purely a test dataset; the standard protocol is to train a model on Buffy images and test on PASCAL Stickmen. We follow this protocol

for CPS, and for LLPS we train using the larger MoviePose dataset, discussed next.

10.1.3 MoviePose

Large datasets are crucial when we want to learn rich models of realistic pose. The Buffy and PASCAL Stickmen datasets contain only a few hundred examples for training pose estimation models. Other datasets exist with a few thousand images, but are lacking in certain ways. The H3D (Bourdev and Malik, 2009) and PASCAL VOC (Everingham et al., 2009) datasets have thousands of images of people, but most are of insufficient resolution, significantly non-frontal or occluded. The UIUC Sports dataset (Wang et al., 2011b) has 1299 images but consists of a skewed distribution of canonical sports poses, *e.g*. croquet, bike riding, badminton.

Due to these shortcomings, we collected a 5003 image dataset automatically from popular Hollywood movies, which we dub MoviePose. The images were obtained by running a state-of-the-art person detector (Bourdev and Malik, 2009) on every tenth frame of 30 movies, see Figure 10.2. People detected with high confidence were then sent to the crowdsourcing marketplace Amazon Mechanical Turk to obtain groundtruth labeling. Each image was annotated by five Turk users for $\$0.01$ each to label 10 upperbody joints. The median labeling was taken in each image to be robust to outlier annotation. Finally, images were rejected manually by us if the person was occluded or severely non-frontal. We set aside 20% (1016 images) of the data for testing.

10.1.4 VideoPose

We also apply our methods to a video dataset, which we collected ourselves: VideoPose 2.0. Clips in the dataset were hand-selected (before developing our algorithm) to highlight natural settings where state-of-the-art methods fail: a highly varied (yet realistic) range of poses, rapid gesticulation, and a significant portion of frames (30%) with foreshortened lower arms. The dataset consists of 44 short clips, each 2-3 seconds in length, with a total

Figure 10.2: A sampling of the 30 movies used in MoviePose. The full list: *12 O'Clock High, 2 Fast 2 Furious, Along Came Polly, American Wedding Unrated, Basic Instinct, Batman Returns, Battle Cry, Bend Of The River, Bourne Supremacy, Casino Royale, Collateral, Daredevil, Diehard 2, Flight Plan, Funny Girl, Giant Side, Goldeneye, Hitch, Irma La Douce, Italian Job, Million Dollar Baby, Monster In Law, Mr. & Mrs. Smith, National Treasure, Ocean's Eleven, Princess Diaries, Schindler's List, Ten Commandments, The Departed.*

of 1,286 frames. We use 26 clips for training, recycle 1 training clip for a development set, and use 18 for testing. As in the Buffy and PASCAL datasets, we fix global scale and translation of the person—in this case by manual annotation.

10.1.5 Discussion

The datasets Buffy, Pascal, MoviePose and VideoPose have different biases, as can be seen in Figure 10.1. First, note the distributions of joint locations. The least varied is Buffy, followed by Pascal, MoviePose and then VideoPose. In VideoPose in particular, the wrist locations have much more spread, and often lie on top of elbows, indicating more foreshortening in this dataset. The hands are often raised up, gesticulating. MoviePose has 5 to 10 times more exmaples than the other datasets.

Both Buffy and Pascal were collected with the 2D pictorial structure model in mind, where frames could be plucked individually to satisfy some of the 2D-based assumptions of the model. In the VideoPose dataset, we collected clips only based on duration, scale

and viewpoint of the person. MoviePose frames were selected automatically via a person detector. Note that Buffy and Pascal both have less than 20 examples with elbows raised above shoulders, whereas VideoPose has none—this is an unusual pose for sitcoms and amateur photographs.

Second, observe the average images of these datasets in Figure 10.1-bottom. We can see plain biases—Buffy and VideoPose are darker (being primarily interior settings), the Buffy dataset is primarily female, and VideoPose has fewer unique background settings (the Friends' apartment, coffee shop, etcetera) which bias the average image. MoviePose has a significantly smoother average image than the rest, as it contains 5 to 10 times more data.

10.2 Evaluation Measures

Based on Problem 1, we wish to measure how well we recover "line segments describing the major anatomical parts". Ideally, we want to measure how often our guessed pose matches a groundtruth pose. However, this is nearly impossible in practice, even on the training set. It is very difficult to achieve anything near pixel-level precision—in practice, 1 pixel is approximately the size of a person's pupil at the resolution we use. This lack of precision is due to uncertainty in human labeling, fixed-length limb models (in the case of classical PS and CPS), and translation invariant features which cannot guide limbs to extremely precise fits.

We consider the following relaxed measures of performance:

10.2.1 Root-Mean-Square Error (RMSE)

Because our datasets can be scale normalized (to the precision of an upper-body detector or groundtruth torso), pixel distances are meaningful across examples, and semantically meaningful. As such, we could measure accuracy via root-mean-square error, *i.e.* the average Euclidean distance between predicted joints and the groundtruth joints on the test

Figure 10.3: Joint error matching limits. We measure a range of pixel error distance thresholds to make a performance curve.

set. However, when a guess is incorrect, it can be arbitrarily far away from the groundtruth, and arbitrarily skew the RMSE.

10.2.2 Pixel error threshold

In light of the outlier skewing problem of RMSE, we examine pixel accuracy at a range of thresholds. At each threshold, we report the percent of test joint guesses that are within that Euclidean distance of the groundtruth, scaled by the size of the groundtruth torso. We explore a range of thresholds, from very precise (the semantic distance of a average palm's width in pixels), to somewhat loose (the semantic distance of an average head's height in pixels). This is illustrated in Figure 10.3.

The resulting performance curve gives us a good idea of accuracy at different operating points. This is useful when assessing system for different applications. For example, it may be that for action recognition only a coarse notion of pose is required (Wang et al., 2011b), but for automatic sign language understanding (Buehler et al., 2009) we require near-certainty.

10.2.3 Percentage of Correct Parts (PCP)

Other works have proposed a measure Percentage of Correct Parts (PCP) that is a limb-based measure of correctness: a guess limb is correct if its endpoints lie within radii of the groundtruth endpoints, where the radii are half the length of the groundtruth limb (Eichner and Ferrari, 2009; Ferrari et al., 2008). The public implementation of this error measure (Eichner and Ferrari, 2009) has some peculiarities: It uses a max-norm in a coordinate space aligned to the groundtruth limb, and allows arbitrary matching of endpoints (*e.g.*, matching wrist to elbow is permissible). This even allows correct matches that are perpendicular to the true limb, as long as the guess limb length is no longer than the true limb length and intersects the groundtruth limb at its midpoint.

We prefer not to use PCP because (1) its criteria for a matched guess is too relaxed (2) is discontinuous (3) only considers one operating point instead of a range and (4) there are different interpretations of the metric leading to discrepancies in implementations (see the code release of Eichner et al. (2010)). However, for historical reasons, many works have reported results in terms of PCP, and here we do the same for compatibility.

10.3 Competitor Methods

The field of 2D human pose estimation has exploded in the last 5 years. The public datasets we report numbers on are highly competitive, with absolute performance rising each year since 2008. We compare to several competing models. Whenever possible, we report numbers in §11 from the publicly available implementations of competitors' code; these numbers are typically different than numbers reported in papers. When public code is not available, we include PCP measures reported by the authors.

Note that the numbers reported here for our models are directly from our publicly available code, meaning any practitioner should be able to replicate results exactly.

- Mean pose baseline

One reasonable sanity check is to compare against guessing the average pose in every te. As can be seen in Figure 10.1, there is some centrality to the data, such that guessing the mean joint position will get some fraction of joints correct. The mean pose is obtained by empirical averages of joint locations on the training sets; each joint estimated independently.

- Andriluka et al. (2009)

  This is a classical dense PS method, as described in §3. The unary limb detectors are trained Adaboost ensembles of Shape Context on top of Canny edges. The pairwise potentials are standard unimodal geometric displacement costs, computed efficiently with distance transforms (§3.1).

- Eichner and Ferrari (2009)

  This is a complex system built off of Ramanan and Sminchisescu (2006). The initial unary term is linear filters on image edges. It then iteratively re-parses using color estimated from the initial parse. It also uses graphcut (Boykov et al., 2001) to rule out some of the background clutter, and post-processing of probabilistic marginals to obtain final limb segments.

- Yang and Ramanan (2011)

  This recent work uses HoG as its only image cue. Like our models, it is trained jointly in a discriminative learning framework. Contemporary with our proposed Stretchable Ensemble model, Yang and Ramanan (2011) also use joints as a basic unit of inference. Their work focuses on modeling a several appearance modes for each part, which they treat as latent variables to be estimated during training.

10.4 Implementation Details

Here we give various additional details about the implementation of our models. All the details of the various feature implementations are provided in §8. Code is available online

at

http://vision.grasp.upenn.edu/video

10.4.1 CPS

Coarse-to-Fine Cascade While our fine-level state space has size $80 \times 80 \times 24$, our first level cascade coarsens the state-space down to $10 \times 10 \times 12 = 1200$ states per part, which allows us to do exhaustive inference efficiently. We train and prune with $\alpha = 0$, effectively learning to throw away half of the states at each stage. In practice we adjust α's per part after a cascade stage is learned via cross-validation error, to prune as much as possible while retaining 95% of the groundtruth validation hypotheses.

After pruning we double one of the dimensions (first angle, then the minimum of width or height) and repeat. In the coarse-to-fine stages we only use standard PS features. HoG part detectors are run once over the original state space, and their outputs are resized to for features in coarser state spaces via max-pooling. We also use the standard relative geometric cues as described in Sec. 4. We bin the values of each feature uniformly, which adds flexibility to the standard PS modelrather than learning a mean and covariance, multi-modal pairwise costs can be learned.

Final stage To obtain segments, we use NCut[19]. For the contour features we use 30 segments and for region moments 125 segments. As can be seen in §11, the coarse-to-fine cascade leaves us with roughly 500 hypotheses per part. For these hypotheses, we generate all features mentioned in §8. For pairs of part hypotheses which are farther than 20% of the image dimensions from the mean connection location, features are not evaluated and an additional feature indicating this is added to the feature set. We concatenate all unary and pairwise features for part-pairs into a feature vector and learn boosting ensembles which give us our pairwise clique potentitals. This method of learning clique potentials has several advantages over stochastic subgradient learning: it is faster to train, can determine better thresholds on features than uniform binning, and can combine different features in a

tree to learn complex, non-linear interactions. This approach is also a major selling point of Nowozin et al. (2011).

10.4.2 Stretchable Ensembles

Our chosen ensemble of tree models is a collection of six models that captures time persistence of each of the six joints, as well as left/right symmetric joint edges for left/right shoulder, elbows and wrists. The decomposition is shown in Figure 6.1. This covers all reasonable connections we could conceive of modeling, and allows us to incorporate all features mentioned in §8.

As input to our method, we use potential shoulder and elbow locations generated by the coarse-to-fine cascade of CPS (§5), independently for each frame. This typically yields 300-500 possible shoulder and elbow locations per image. For each of the 24 discrete elbow orientations predicted by CPS, we project possible wrist locations at 4 different lengths, chosen from the 5th, 25th, 50th, and 75th lower arm length quantiles on the training set. We then take the top 500 wrist locations scored according to the foreground color features for each frame (§8). The result is a sparse set of locations for each joint with high recall.

10.4.3 LLPS

The MoviePose dataset contains 5003 images, of which 1016 are set aside for testing. This means we have $2 \times 3987 = 7974$ half-image examples to learn our mode centers. Based on cross-validation, we found 32 modes to be most accurate in an end-to-end system.

Chapter 11

Results

In this chapter we go through quantitative results on the datasets Buffy, Pascal and Video-Pose (§10.1), analyzing behavior and design choices of our methods. We also compare our models—CPS, ESM and LLPS—against competing models (§10.3). For much of the analysis we focus on upper and lower arms only—in particular, elbow and wrist localization accuracy. The reasons for this are that (1) torso and head localization are near-perfect given a detected person (Yang and Ramanan, 2011), (2) arms are the most interesting parts, involved in actions, hand-held objects and object-person interactions.

11.1 Coarse-to-fine cascade evaluation

In Table 11.1 we show the progress of our CPS model's coarse-to-fine cascade, in the Buffy dataset. As explained in §10.4, we start with a small state space and continue pruning and refining until we reach a somewhat fine $80 \times 80 \times 24$ grid. At the end of the cascade we are left with on average 492 states for each part, 99.67% fewer states than the original $80 \times 80 \times 24$ state space. We see that after one level of the cascade, we have already pruned more than half of the full state space away. This is intuitive because there are many easy decisions of states to reject based on even geometry alone, *e.g.* the left elbow does not ever appear in the upper right corner of the person's bounding box.

cascade level	state dimensions	# states in the original space	# states in the pruned space	state space reduction %	near-correct lower arms unpruned (PCP)
0	10x10x12	153600	1200	00.00	100
1	10x10x24	72968	1140	52.50	76.6
3	20x20x24	6704	642	95.64	72.3
5	40x40x24	2682	671	98.25	70.5
7	80x80x24	492	492	99.67	68.4
detection pruning	80x80x24	492	492	99.67	58.6

Table 11.1: Coarse-to-fine cascade progression analysis. We show the progression of state spaces in the cascade, as well as reduction in the state space at each level (measuring efficiency), and in the last column, how many arm hypotheses remain closely matched, considering the closest match to groundtruth remaining from the unpruned hypotheses (measuring accuracy).

As the cascade progresses, we do lose arm hypotheses close to the groundtruth arms, seen in the last column of Table 11.1. However, the percent of hypotheses close to groundtruth after the cascade process is still higher than any current system's accuracy on lower arms (see Table 11.2). Thus this number (68.4%) is an upper bound on how well we could do with our small, pruned set of states.

To verify that our pruning is better than heuristic pruning, we compare to heuristic pruning in Table 11.1, last row. The heuristic is to sample states proportional to their unary potential scores (*i.e.*, HoG limb detectors), with non-max suppression. At the same number of states sampled as we obtain from the cascade, the heuristic pruning misses 10% more lower arm hypotheses.

Finally, it could be the case that the benefits of our rich features in the last stage of CPS make discrepencies in accuracy of pruning strategies neglible. In other words, even

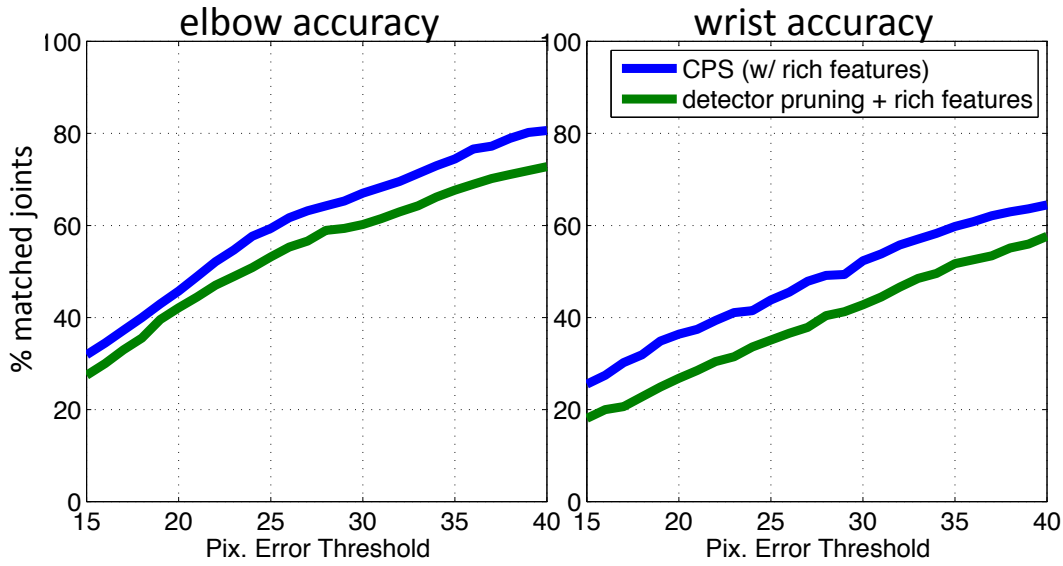


Figure 11.1: Cascade versus heuristic pruning. Here we see that our cascade progression, coupled with a rich set of features does better than heuristic pruning with the same rich features used afterwards. This indicates that not only do the rich features matter, but also the quality of the states retained from CPS.

the pruning heuristic retains 58.6% of lower arms in its hypothesis set, and it is possible that it could perform equally well at final-level prediction when using the same features as CPS. We see in Figure 11.1 that this is not the case. CPS performs 5 to 10% better than simple detector pruning coupled with the rich features we use in CPS. This makes a strong case that the CPS state filtering strategy is important.

11.2 Feature analysis

One of the main contributions of this thesis is technical innovations that allow us to include "everything and the kitchen sink" (in the house of vision features). At this point we wish to verify that the work done implementing, computing and learning parameters for features makes a difference in performance. This is actually quite difficult to do in general,

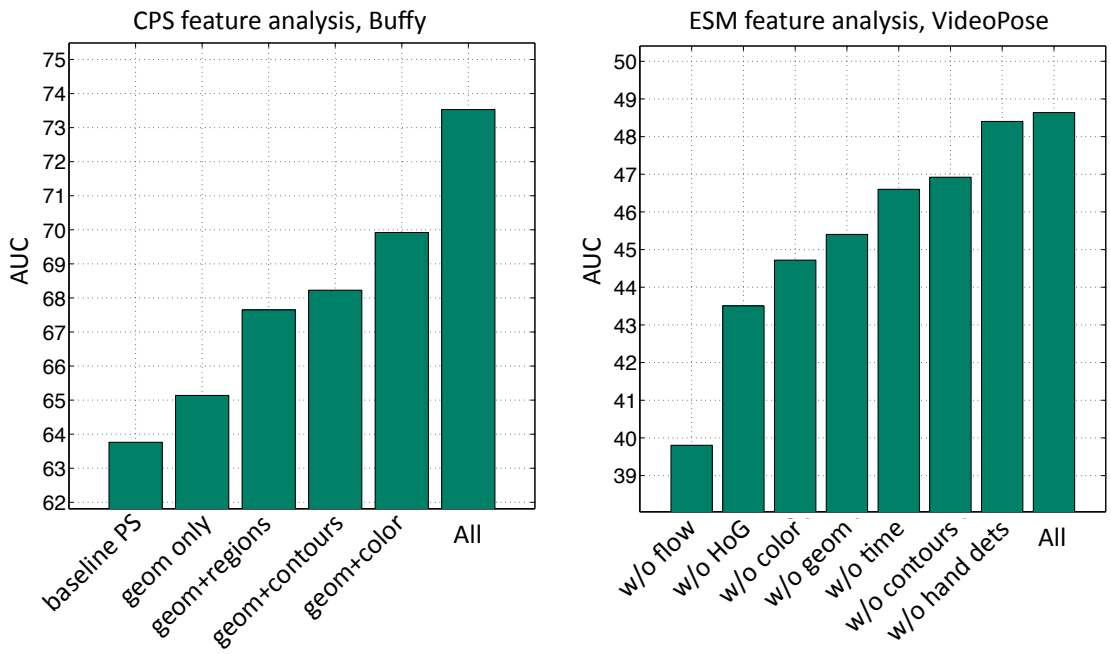


Figure 11.2: Feature analysis. On the left, we observe how adding features contributes to final system performance of CPS on Buffy, measuring the Area Under the Curve of pixel error distance. On the right, we observe how removing single feature modalities from our Ensemble of Stretchable Models affects performance.

because it is computationally infeasible to explore the combinatorially many possibilities of feature sets. Non-additive interactions between feature types may occur. We analyze the importance of features by grouping them by modality, and adding or removing them from our full systems in turn, measuring the change in performance.

In Figure 11.2 we do a feature analysis for CPS (left) and ESM (right), grouping features by coarse modalities. The first important thing to note is that the rich features we include over standard edge template and geometry information lead to better performance. For CPS, including rich feature types individually in conjunction with geometry features helps performance, in particular pairwise cues such as color and contours that were infeasible to compute without our cascade approach. In the bar marked "baseline PS" we

—	MoviePose		Buffy v2.1		Pascal Stickmen		all
method	uarms	larms	uarms	larms	uarms	larms	mean
Andriluka et al. (2009)	—	—	79.3	41.2	—	—	60.3
Eichner et al. (2010)	90.40	52.85	92.77	53.40	67.92	30.56	57.19
Yang and Ramanan (2011)	94.05	67.08	92.77	65.53	65.83	37.92	70.53
CPS	93.95	52.12	95.11	67.02	86.39	59.17	75.62
LPPS	95.57	71.85	88.94	70.64	68.61	44.58	73.37
mean pose	78.25	33.22	93.19	43.83	72.92	34.03	59.24
mean cluster prediction	95.13	63.73	96.81	70.85	85.83	53.75	77.68

Table 11.2: PCP Evaluation of single frame pose estimation. PCP is a fairly loose measure of accuracy and only reveals one precision operating point. We include the measure for historical reasons; for a more detailed picture see Figure 11.3.

evaluate a classical unimodal PS, whereas the geometric parameters in our cascade models ("geom only") are learned bin weights that can achieve non-linearity. All features together do significantly better than any one feature modality in isolation.

On the right side of Figure 11.2, we do an ablative analysis of our ESM model. The most important individual feature modality is optical flow, which gives us a fairly good estimate of foreground/background separation in many video frames. Importantly, many of these feature modalities are not used in pose estimation models because they require joint interactions which lead to loopy, cyclic models.

11.3 System results

In this section we analyze end-to-end system results, using the publicly available code for our systems and competitors, for reproducibility's sake.

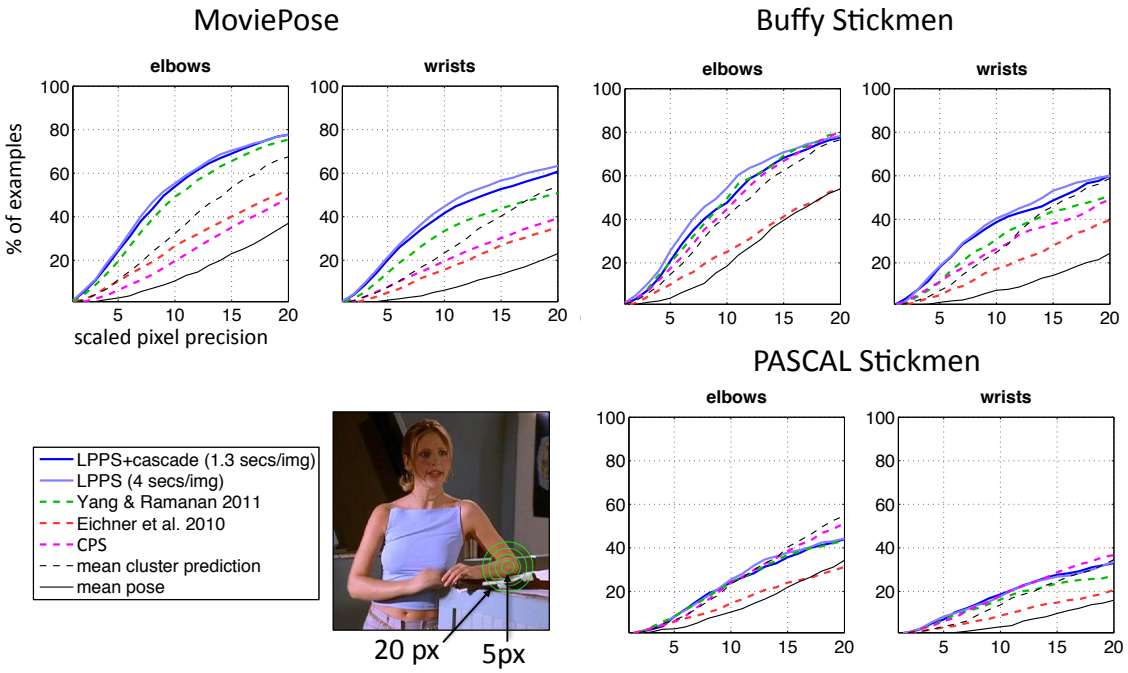


Figure 11.3: Single frame pose estimation results. Shown are our single frame models CPS and LPPS against state-of-the-art competitor models (§10.3)

.

11.3.1 Single frame pose estimation

The performance of all single-frame models are shown on the MoviePose, Buffy and Pascal datasets in Figure 11.3. The LLPS model outperforms the rest across the three datasets. Yang and Ramanan (2011) is the closest competitor overall, but CPS outperforms the others slightly on the most difficult Pascal dataset. Eichner et al. (2010) (and Andriluka et al. (2009), whose code we were unable to run; see Table 11.2) is uniformly worse than the other models, most likely due to the lack of discriminative training and the unimodal modeling. Localization accuracy is not the only way to measure the quality of a model (*e.g.* speed)—see §12 for more discussion.

Surprisingly, the two simple prior pose baselines perform comparatively well. The "mean pose" baseline is a lower bound on performance, but is competitive with Eichner

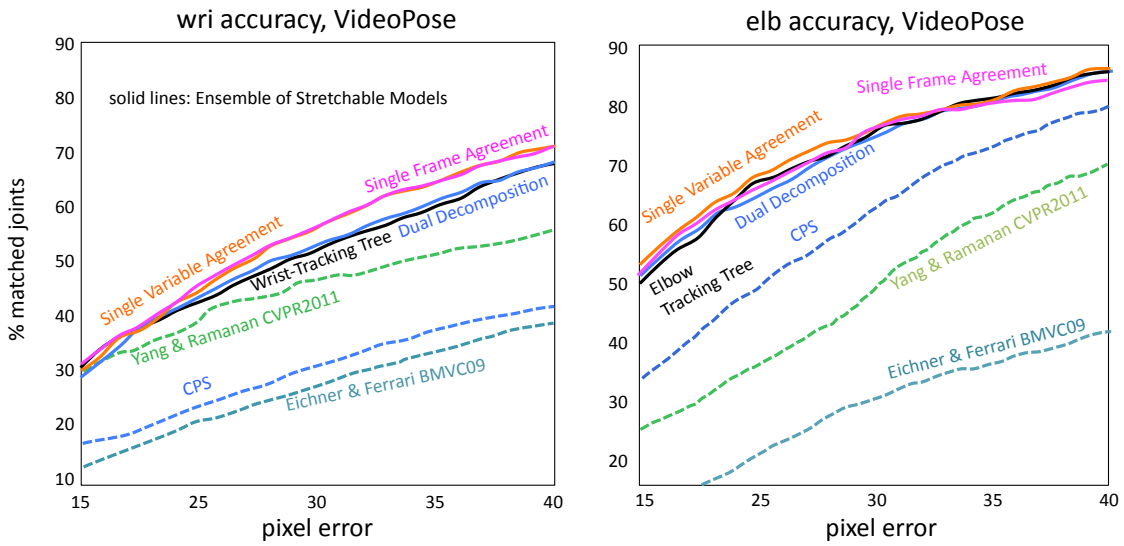


Figure 11.4: Video pose estimation results. Shown are our single frame model CPS, various forms of agreement for Ensembles of Stretchable Models, and other single frame competitors.

et al. (2010) in some cases. The "mean cluster prediction" baseline actually outperforms or is close to CPS and Eichner et al. (2010) on the three datasets, at very low computational cost—0.0145 seconds per image. This surprising result indicates the importance of multimodal modeling in even the simplest form. The decent performance of these mean pose baseline is also an indication of either the difficulty or lack of pose variation in these datasets, or a combination of both. In fact, the scatterplots in Figure 10.1 do show that most elbows are very tightly grouped in these single frame datasets.

We also report PCP scores in Table 11.2. Comparing the rankings of methods by Table 11.2 and Figure 11.3, we see that PCP does not give a complete story of performance.

11.3.2 Video pose estimation

Quantitative results for pose estimation in video are shown in Figure 11.4. All three methods we explore for inference in our pose estimation video model (Ensemble of Stretchable

Models) outperform the state-of-the-art single-frame methods by a significant margin. Using just a single one of our Stretchable Model trees already does significantly better than single frame models. This shows the usefulness of our stretchable model (joint-centric) representation of pose, as well as some of the rich pairwise interactions we use that other models do not. It is important to note that previous work has found that incorporating time persistence into models actually *hurt* performance (Ferrari et al., 2009; Sapp et al., 2010c)—hence single frame models are the most competitive models for which to compare.

Furthermore, we explore the different agreement methods discussed in §6.2.3. From Figure 11.4, we see that the very fast, approximate decoding schemes (A single tree and SV require only a single inference pass) were comparably accurate to more expensive methods (SF takes about $2\times$ as long while we ran DD for $500\times$ iterations, each requiring inference in each submodel)[1]. We found two important trends: (1) On average, completely decoupled inference ($Independent$) was consistently about 0.75-1.5% worse than the inference methods that aggregated information across models, and (2) solving *partial* agreement problems *exactly* (SV, SF) performed better than solving *complete* agreement *approximately* with Dual Decomposition.

The absolute accuracy values are also a testament to the difficulty of the dataset: At the tightest matching criterion, we only correctly localize half the elbows. This suggests that there is plenty of future progress to be made on this dataset, and in this domain in general.

[1]Specifically, running inference in all 6 models sequentially takes about 16 seconds per 30 frame clip (trivial to parallelize) on a Intel Xeon CPU, E5450 @ 3.00GHz. SF inference takes an additional 19 seconds.

Part V

Discussion and Conclusions

Chapter 12

Discussion

In this chapter, we discuss advantages and disadvantages of the different models developed and compared to in this thesis, and justify our contributions. In addition, we address possible criticisms and other tradeoffs and design choices that are of importance beyond pure accuracy. Performance of the different models in terms of part localization accuracy is in §11.3.

12.1 The case for image-dependent interactions

At the time of publication, CPS was significantly better than all modern "in the wild" pose estimation approaches (Andriluka et al., 2009; Eichner and Ferrari, 2009; Ferrari et al., 2008; Ramanan and Sminchisescu, 2006). CPS was the only one to use image-dependent pairwise cues. These other systems focused on either pre-processing to reduce the state space or improving unary potentials via foreground color estimation, and all performed inference relying ultimately on edge-based unary potentials and simple geometric displacement pairwise cues as in the classical PS model (§3). In a controlled setting, we see that a classical PS model coupled with just one image-dependent interaction feature, such as contour continuity, significantly improves performance (Figure 11.2). This makes a strong case that image-dependent interactions improve performance.

The benefits of image-dependent interactions is further reinforced by the performance by our video pose estimation model. The Ensemble of Stretchable Models exploits a variety of across-body and across-frame image interactions and significantly outperfoms methods with fewer interactions.

Finally, we see effective data-dependent switching of modes in our LLPS model. The decision of which local model $s^z(x, y)$ to choose depends on large-context interactions considering a whole half-body in scope. These holistic cues are crucial in determining an effective mode to use, and serve as one of the main reasons LLPS outperforms Yang and Ramanan (2011).

We believe our cascade approach is a very useful tool that helps us design models with larger image-dependent interactions. It is a key component to all models presented here to achieve a high degree of accuracy and/or efficiency. The cascade framework also holds potential to improve the richness of models in the future. In general, our approach is a principled framework that frees us from restricting our models to limited pairwise potentials of the form $||y_i - y_j||$. This allows much more flexibility in the future to incorporate more pairwise and even higher-order features.

12.2 More features or more modes?

As discussed in §7, there are two major approaches for improving pose accuracy in recent years: adding more features, *e.g.* CPS, ESM and Eichner and Ferrari (2009); Ferrari et al. (2008); Tran and Forsyth (2010), or adding more modes, *e.g.* LLPS and Felzenszwalb et al. (2011); Johnson and Everingham (2011); Wang et al. (2011b); Yang and Ramanan (2011).

LLPS and Yang and Ramanan (2011) are both multimodal models that outperform the feature- and computation-heavy CPS. It is unclear given the current state-of-the-art if the multimodal HoG model approach is yet saturated, or more data and more nuanced mode definitions will continue to yield increased performance in coming years. Zhu et al. (2012)

has an excellent study of whether these models are near saturation; see Figure 12.1 for our own trend analysis. Both suggest that we are not near saturation yet, in terms of number of parts, modes and training data. There are also no limit to the performance gains to be had by adding more and better features, only computational hurdles.

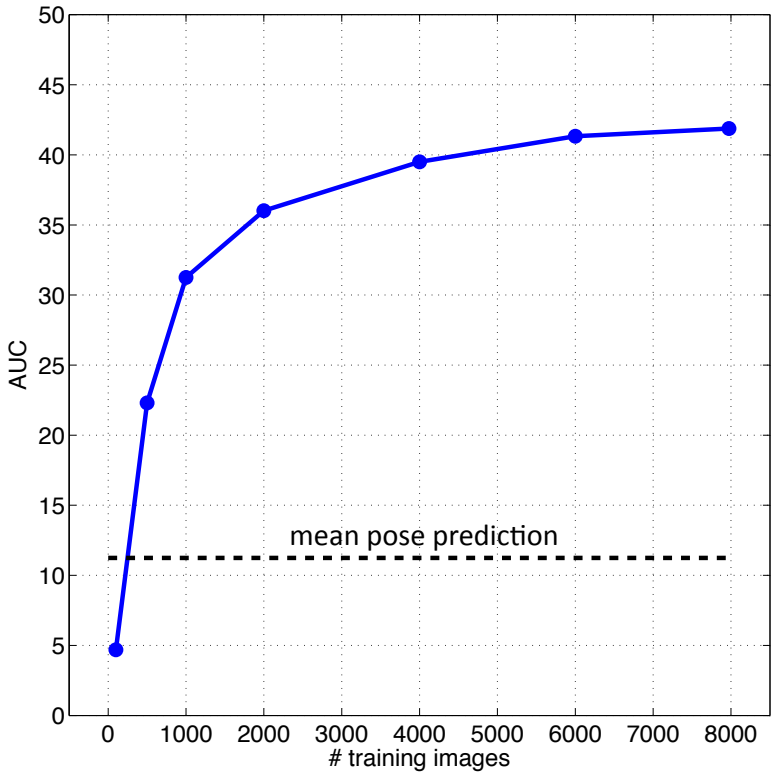


Figure 12.1: LLPS learning curve: test set accuracy for wrists and elbows combined for LLPS models trained with different amounts of data. The mode definitions, number of modes, and cascaded mode prediction step were held fixed and used for training.

Ultimately, these research directions are complementary, and an ideal model would use a combination of rich features and multiple modes. Each additional feature type (*e.g.*, segments, contours, optical flow, depth) incurs an additional cost to obtain, but adds to the generalization capabilities of the model. Additional modes allow for more specific modeling of different scenarios, but require more training data to estimate parameters accurately. This leaves a large space of possible models combining the two approaches.

12.3 Joints or limbs?

The ESM model introduced a joint-based 2D representation of pose. At the same time, Yang and Ramanan (2011) also introduced a model based on joints and limb midpoints as basic units of inference. This approach has clear benefits for easily capturing foreshortening and scale. It has the seeming disadvantage of not being able to capture limb-pair features with a pairwise model. However, this is not a fundamental limitation. Especially using cascaded inference techniques we should not shy away from describing higher order cliques in the future. In general, the scope of the basic atomic unit for inference (the inference variables) need not be dictated by the scope of the largest clique we capture in our model. Joint-based models are worth exploring further. In particular, we expect an Ensemble of Stretchable Models approach applied to *single frame* pose estimation to work well.

12.4 Detection, localization, or both?

Some pose estimation systems advertise that they work well for both person detection *and* pose estimation—in particular Andriluka et al. (2009) and Yang and Ramanan (2011). One system that does both is beneficial in its simplicity—one function to both find a person and find their body parts. Our approach, on the other hand, requires first detecting a person with a dedicated person detector, and then running our pose models on the detected person.

We believe that detection and localization are fundamentally different tasks and should be decoupled. In detection, we wish to generalize over all poses and determine how to discriminate any pose from background clutter—a detection is correct even when a pose is incorrect, and a detector must also have some notion of global confidence to determine, over all possible image patches, whether it is a person or not. A pose estimator works under the assumption that a pose is present, and is correct only if it predicts the right pose versus combinatorially many wrong poses.

One model that attempts to perform both tasks is bound to perform only as well as it

could tuned to each task independently, and probably worse. It may be that PS models are the right *family* of models for both detection and localization—they have attractive benefits for generalizing over poses with deformations and obstructions in addition to localizing pose—and they should be used for both. However, models should be trained evaluated specific to a single task.

12.5 Everything and the kitchen sink: a bug or a feature?

One of the selling points of our models is the ability to include a multitude of features. The goal is to include as many feature modalities as possible in our CPS and ESM models. Having so many features makes it difficult to determine exactly what is contributing to the success of our model.

From a machine learning standpoint, this is an attractive aspect of our system: given training data, we can try everything and see what works. From a computer visionist's (or perceptual scientist's) perspective, this is a disadvantage—it is difficult to gain insight into why the model is performing well.

We take a functional, application-driven approach towards computer vision, and consider our problem one of engineering rather than perceptual science. The inability to measure the individual performance of components in any complex system is inevitable—the whole is greater than the sum of its parts. We provide individual feature analysis in Figure 11.2, and make convincing arguments that the features and interactions we include are beneficial. We make no statement as to which features are the "best", in any sense other than their contribution to final system performance.

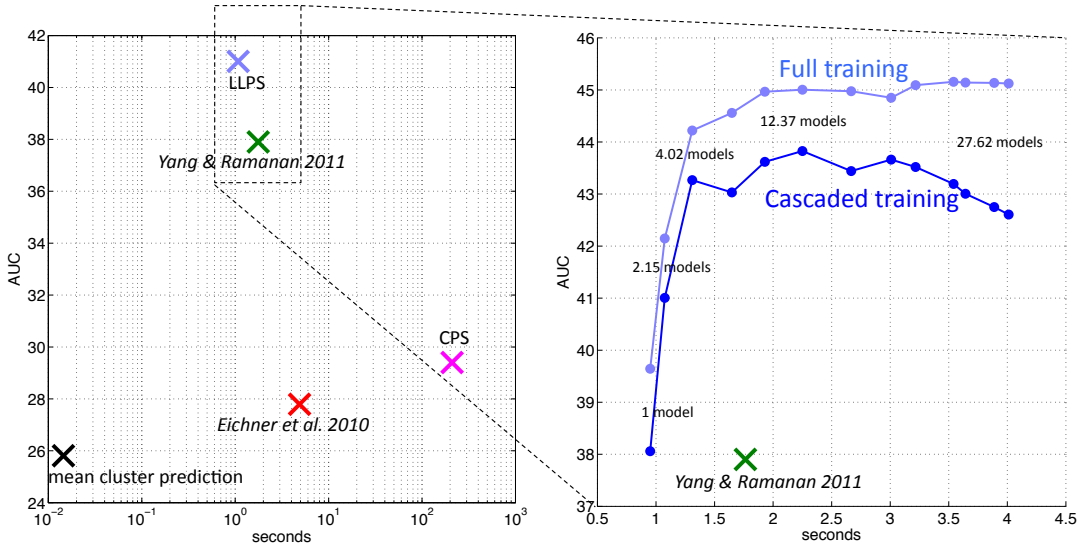


Figure 12.2: Test time speed versus accuracy. Accuracy is measured as area under the pixel error threshold curve (AUC). Speed is in seconds on an AMD Opteron 4284 CPU @ 3.00 GHz with 16 cores. On the left, we compare different methods with a log time scale. The upper left corner is the most desirable operating point. LLPS is strictly better than other methods in the speed-accuracy space. On the right, we zoom in to investigate our cascaded LLPS approach. By tuning the aggressiveness (α) of the cascade, we can get a curve of test time speed-accuracy points. Also, we see that "full training"—considering all modes in every training example—rather than "cascaded training"—just the ones selected by the cascade step—leads to roughly a 1.5% performance increase (at the cost of $5\times$ slower training).

12.6 Accuracy, speed, simplicity

When developing our CPS and ESM models, we focused our attention on obtaining the best performing, computationally tractable system. Besides raw performance, practitioners care as much about *speed*—does the system run quickly?—and *simplicity*—how long does it take to download, compile, understand, run and/or re-implement? One of the motivations of LLPS and attractiveness of Yang and Ramanan (2011)'s model is its speed and simplicity, in terms of image features (only HoG) and lines of code.

LLPS is strictly better than other models in terms of speed and simplicity, according to Figure 12.2. Yang and Ramanan (2011) is strictly better than all but LLPS. Among the other models, there is no clear winner—CPS is more accurate but slower. Predicting cluster means is extremely fast but less accurate.

We believe that, moving forward, models with any combination of the three contributions *accuracy, speed, simplicity* are all worthwhile, even independent of the others. Any current system's slowness today will likely be a non-issue in 5-10 years with the advent of faster CPUs and more cores. Anecdotally, at time of publication two years ago, the CPS system took 5 minutes and 15 seconds. Today, at roughly the same price and consumer availability, the system runs in 3 minutes.

Chapter 13

Future directions

1. When a distinguished but elderly scientist states that something is possible, he is almost certainly right. When he states that something is impossible, he is very probably wrong.
2. The only way of discovering the limits of the possible is to venture a little way past them into the impossible.
3. Any sufficiently advanced technology is indistinguishable from magic.

— Clarke's Three Laws, Arthur C. Clarke

In this chapter we suggest avenues for further research, both in pose estimation and other domains in computer vision and machine learning. Some of it is speculative. Our hope is that the work of this thesis is a building block for these new research directions in the immediate future.

13.1 Solving pose estimation

What would it take to declare pose estimation solved? Certainly with current accuracies—getting wrists right about half the time—we cannot claim that the state-of-the-art "works,"

from a layperson's perspective. Taking a functional standpoint, we deem to call pose estimation solved when it is ready to be featured in a consumer product in the wild, the way face detection is in cameras and the Kinect now in video games. We speculate that accurately localizing elbows and wrists 90% of the time in datasets like MoviePose would be sufficient for this level of broad use. However this would still leave enormous room for improvement—MoviePose and our current models do not consider handling multiple people and their interactions or reasoning about occlusion or very non-frontal non-upright poses.

As a side note, our current state-of-the-art in pose estimation may very well be ready to be used as a second-tier method in real-world applications, in the sense that the pose output is not relied upon, but is treated as a non-crucial but helpful noisy sensor. Wang et al. (2011b), for example, use it as a descriptor for action recognition. The same is done for image retrieval (Ferrari et al., 2009) and scene geometry estimation (Delaitre et al., 2012; Gupta et al., 2011).

13.1.1 Pushing current models to their limits: more data, more modes, more submodes and supermodes

How might we achieve higher accuracy on MoviePose? The superficially uninteresting but most promising way forward is using the same models, but with significantly more data. Our current research suggests that even fixing the current set of modes, test set accuracy has still not saturated as we increase the number of examples, as shown in Figure 12.1. In its current form, LLPS is comprised of 32 mode models, which cover the range of human pose in video quite well. The most common mode (arms at rest) has 800 training examples at its disposable; the least common modes (arms raised above head and hand held up to face) have less than 25 examples each. See Figure 13.1. Immediately we see there is room to flesh out some of the rarer modes and estimate them better.

As a back-of-the-envelope calculation, to collect enough data to train the least frequent mode with 500 examples, respecting the natural mode distribution collected from movies,

Figure 13.1: LLPS mode image centers and their population counts, populated by MoviePose.

we would need twenty times as much data. In addition, although 32 modes cover the variations in upper body pose well, we expect to get more accurate modeling by also considering modes of appearance, from clothing and body type. From intuition, it seems reasonable that each of our current 32 modes could be split into at least 3 submodes based on appearance—for example, one submode for men's arms, one for women's, and one for baggy clothing. This would bring up the tally of data needed to 60 times the current levels, for the least frequent modes to be modeled with 3 submodes, each with 500 training examples.

It is almost flippant to suggest simply using more data will solve our problems. First, obtaining and labeling this data is no small feat. Second, clever methods to train on such large datasets need to be designed. Naïvely using current training methods for 60 times more data would result in 13 days to train and would require 1.4TB of memory; just slightly out of the realm of feasibility for current standard servers.

A final problem is that scaling up to a larger number of classes always tends towards

more class confusion. This is a much more studied issue in large classification tasks such as the ImageNet retrieval challenge (Deng et al., 2010), in which 10,000 different classes are to be predicted. A common way to deal with class confusion at large scale is hierarchical classification, *e.g.* first predict non-animal from animal, then dog from other animals, then Corgi from Bernese Mountain Dog. The hierarchical decisions are easier to make and there are less of them than comparing all fine-level classes. This suggests an analogous approach to multimodal pose modeling: group the 32 modes recursively into 16, 8, 4 and 2 coarse supermodes. Cascaded prediction could also be effectively applied here.

In summary, to push the current LLPS model framework to its limits, we require at least an order of magnitude or more additional data, cleverer training algorithms (or patience), and a richer hierarchy of modes, some more specific, some more general than our current collection, based on both pose and appearance. It is our belief that such improvements to our model could enjoy incredible leaps in performance in the coming years. A preliminary study on pushing the state-of-the-art of multimodal models with increasing data and modes in other domains has reached similar conclusions (Zhu et al., 2012).

13.1.2 Putting people in their place

We believe pushing multimodal models to their limits in the coming years will bring a high degree of accuracy to frontal upper body pose estimation. Even so, these models leave much to be desired. Importantly, they are unable to reason about occlusions and multiple people.

Some past work models occlusion probabilistically, however, the basic reasoning essentially boils down to a threshold on the lack appearance evidence for a part (Wang and Mori, 2008). As we have shown throughout this thesis, the decision to declare an arm missing versus just being difficult to detect is extremely difficult with current models. In reasoning about multiple people, Eichner and Ferrari (2010) explore the combinatorially many possibilities of detected people in a scene, Kulesza and Taskar (2010)

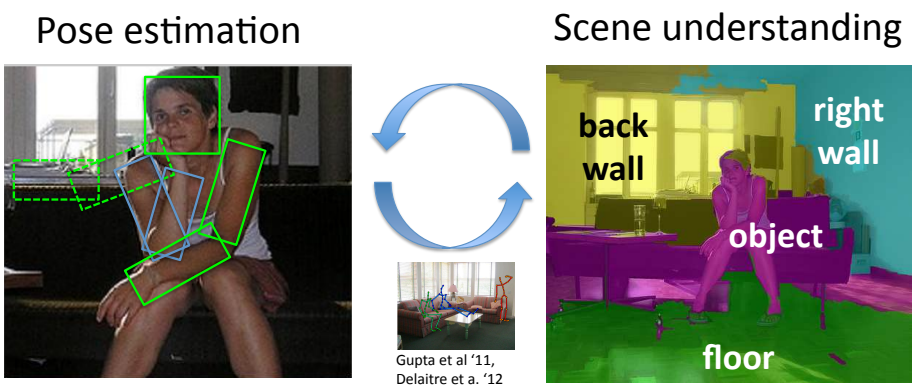


Figure 13.2: An illustration of how scene parsing and pose estimation can aid each other. Delaitre et al. (2012); Gupta et al. (2011) have shown that state-of-the art pose estimation helps infer scene geometry. We believe that scene parse information (such as that by Lee et al. (2009) run on the image on the right), can help pose estimation, *e.g.*, by indicating that the true arm location (blue rectangles) are more likely than the false positive in the background (dotted green rectangles).

provide a framework for sampling a high-quality, diverse set of poses in an image, and Andriluka and Sigal (2012) model interacting people by connecting them together in one tree-structured PS model.

In all the above models, we believe a key missing ingredient is the context around the person. Pose models to date only try to separate pose from background, and there is no attempt to understand or explain the background. We envision a model that attempts to put people in the scene around them, and label every pixel in the scene along with it, similar to standard scene parsing.

We believe modeling the scene and pose jointly is crucial to determining whether a part is occluded: it can consider as evidence whether there is an occluding foreground object, not simply declare occlusion whenever the part detector signal is weak. Similarly, multiple people can be explained and reasoned about explicitly.

Delaitre et al. (2012); Gupta et al. (2011) have already provided empirical evidence that current pose estimators can aid scene geometry and scene labeling inference. We believe

the other direction is also true: scene models can help pose estimators. A simple proof-of-concept is demonstrated in Figure 13.2. There is much interesting work to be done reconciling pose and scene models to agree, and what their lingua franca representation should be. Agreement algorithms such as those discussed in §6 may be quite viable here. As a starting point, Wang and Koller (2011) propose a simple model that coupled a figure-ground segmentation model with a PS model, with modest performance improvements.

13.2 Pose models on other problems

Pose estimation is a challenging problem with fundamental computational complexity issues. The methods developed in this thesis may well be applied to other important and challenging problems in computer science. We discussing a few promising ones here.

13.2.1 Bringing cascades to the masses

The application of structured prediction cascades has already been successful in a variety of domains besides pose estimation: optical character recognition, part-of-speech tagging (Weiss et al., 2012) and natural language dependency parsing (Rush and Petrov, 2012).

There are several structured problems in computer vision which contain similar computational bottlenecks, and as a result, models have restricted interaction terms. One such problem is **scene parsing**, in which each segment of an image must be labeled with one of a handful of labels. Because of the cyclic graph structure, only simple neighbor interactions are studied. Recently, Munoz et al. (2010) showed that a non-structured hierarchy of models which captured large part interactions proved to be very effective. We expect that applying a cascade to this problem would allow us to keep both the nice properties of a structured model as well as the benefits of data-dependency in large cliques of segments in an image.

Similar trends can be seen in the **optical flow** literature, which poses the correspondence problem between a pair of images as a structured prediction problem with a data-independent pairwise smoothness term. Good results are currently achieved Liu et al. (2011), who use a coarse-to-fine approach to speed up the matching. Cascades would provide a principled way to do this, as well as incorporate data-dependent smoothing terms—for example it could depend on local texture or permit sharp changes at strong boundaries.

Image retrieval systems typically follow a paradigm of (1) obtain a short list of possible candidates for retrieval using cheap and coarse nearest neighbor search via bag-of-words descriptors, and (2) perform more computationally intensive matching on the short list (Wang et al., 2011a). This is a heuristic two step cascade. We envision a progression of cascade models that operates on histograms of bags of words, then histograms of word bigrams, then trigrams, etc. With larger n-grams we hope to use richer word frequency and spatial consistency cues to get better results. The finer cascade progression and learned accuracy-efficiency trade-off we hope would also yield better performance.

Scaling up to many various applications, one is compelled to think of **automatically building cascade architectures**. Designing structured prediction cascade applications thus far has been accomplished manually based on domain expertise—determining how the state space should be refined, and at what stage of the cascade should refinements and additional modeling power and complexity be introduced. In fact, the CPS model cascade architecture was designed by intuition and cross-validation in a greedy stage-wise manner. It's unlikely that it has the optimal speed-vs-accuracy tradeoffs compared to the introduction of features earlier or later in a cascade, deeper cascades, or different features. The same question exists for the original Viola and Jones (2002) classifier.

One promising approach towards automatic cascade architecture design would be to provide an algorithm with a pool of cascade refinement elements (from features, or state space transformations), each with measurable efficiency, accuracy and computation behavior. It would be the job of the algorithm to choose a sequence of elements to build an

effective cascade from scratch. For example, in an LPPS-type model, we could give the algorithm a pool of modes that span the continuum of coarseness to fineness. The algorithm could decide how many modes to use, what their granularity is, and how to arrange them so that at test time inference is efficient and accurate.

This has been studied for binary cascades, for example by Lefakis and Fleuret (2010), which jointly learns all levels of the cascade to achieve better performance than the hand-crafted Viola and Jones (2002) detector. These same issues are also of critical importance in applications such as medical diagnosis, where the introduction of new features comes at a considerable cost (from additional diagnostic tests or adding more participants in a clinical study). In statistics this is known as sequential hypothesis testing, which has been applied to efficiently determining image similarity by Pele and Werman (2008). Deng et al. (2011) greedily infer a "label tree" for efficient classification of a large number of classes. Lastly, Van Roy (1998) learns a decision process model via reinforcement learning.

13.2.2 Solving graph problems with tree agreement

In §6 we propose a simple approximation to minimizing a general pairwise energy function with cyclic dependencies—tree decomposition with single variable agreement; combining beliefs through max-marginals. In this problem, we showed that it was a cheap and effective alternative to dual decomposition (Komodakis et al., 2007).

The raises the question of whether the technique can be applied as well to other hard graph minimization problems. These include problems such as scene parsing, stereo matching, graph matching, image denoising, optical flow, and protein design. For problems that are relatively easily solved, state-of-the-art minimization techniques might be overkill, and single variable agreement may be just fine. For a toy demonstration of this, see Figure 13.3. Methods such as Batra and Kohli (2011); Jojic et al. (2010); Sontag (2010) are iterative processes with approximation guarantees but are considerably slower.

On problems that are difficult, it may also be the case that our single variable agreement may work just well as more sophisticated costly techniques—in other words, they all

may be far from the global optimum. It remains an open but important question to characterize when single-variable agreement may work well empirically and in theory, and what approximation guarantees it might admit.

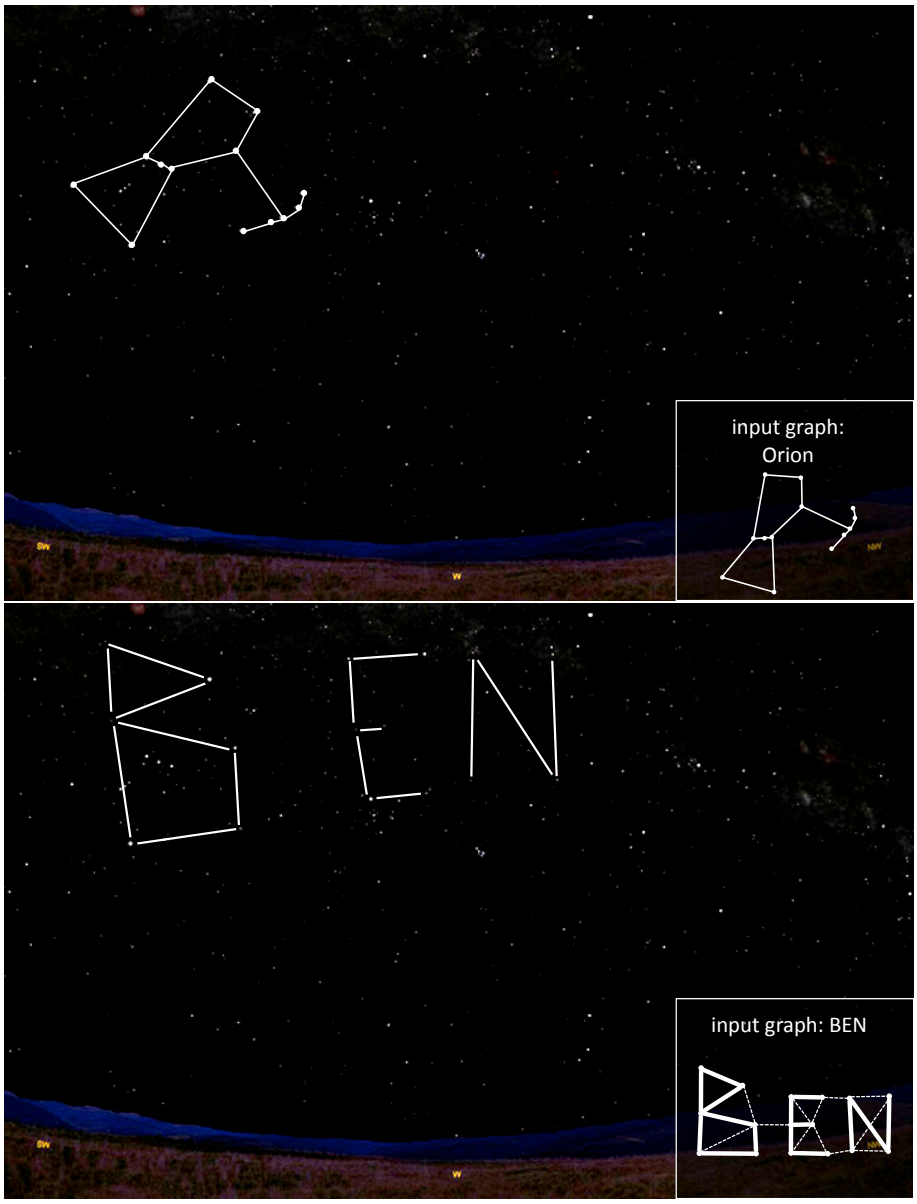


Figure 13.3: A demonstration of a toy application of tree agreement: a constellation finder. Given a loopy graph that defines a constellation's geometric structure, our tree decomposition plus variable agreement method (§6) can match the graph in the night sky (state space is the set of star points). Here we decompose the cyclic graph into a collection of spanning trees, one rooted at each node and obtained automatically via depth-first search. This method works both for searching known exact geometries like the Orion constellation, as well as for discovering new, user-designed constellations. Note that any single spanning tree of these constellation models fails to represent all the important geometric connections, and results in false positive matches in practice.

Chapter 14

Conclusion

This thesis proposes advancements in 2D human pose estimation models to improve upon state-of-the-art performance. Specifically, previously proposed pictorial structures models are handicapped by the needs of efficient inference tricks; forced to express simple pairwise cues that are only a function of parts' spatial relationships. Further, the spatial relationship structure had to form a tree graph.

We push past the inference barrier in several ways, allowing us to include richer image-dependent interactions. First, we proposed Cascaded Pictorial Structures (**CPS**), a sequence of structured models that efficiently prune the state space of possible poses down to a manageable number. This allows us to perform efficient exact inference without restrictions. We exploit this by incorporating a variety of rich features from complementary sources, improving upon state-of-the-art PS approaches in single frame pose estimation.

We then extend this approach to handle pose estimation in video. Maintaining a rich set of variable interactions in video creates a cyclic network, which is known to require inference inference exponential in the number of frames of video. We maintain tractability through the use of a cascade step as in CPS, and an approximate inference method which decomposes the cyclic structure of interactions into an ensemble of tree graphs, which capture all the interactions of the cyclic network, with redundancies. With this Ensemble of Stretchable Models (**ESM**), approximate inference is only linear in the number of frames

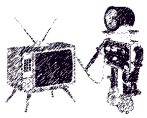
of video. Furthermore, to handle fine-grained articulation and foreshortening effects often present in real video clips, we use a joint-based representation of pose, as opposed to a limb-based representation, hence our model is "stretchable".

Finally, we explore a complementary approach to the line of research that motivated CPS and ESM. These methods focused on novel computational techniques that allowed us to add more and more features and rich interactions into our pose models, in the hope that more features would lead to better generalizability and performance accuracy. Alternatively, in Local Linear Pictorial Structures (**LLPS**), we focus instead explicitly on the nonlinear, multimodal nature of the problem, *not* by introducing more and more features, but instead modeling each local neighborhood with it's own PS model.

We show empirically that our models are state-of-the-art on competitive public datasets, verifying the worthiness of our modeling innovations: (1) cascades of structured models (2) ensemble of tree models (3) joint-based representations (4) local linear modeling. All of these ideas are valuable contributions to the field of pose estimation, with potential to help in other domains involving structured problems as well.

Human pose estimation in the wild in its most general setting is still far from a solved problem, although we have made significant advances through the course of this research. Moving forward, we expect further advances to be made with (1) larger datasets, (2) the computational capabilities to scale current approaches to an order of magnitude more data, and (3) reconciling estimates of pose within the larger context of scene understanding.

Future improvements in pose accuracy seem promising and the dream of understanding human pose for a variety of applications is increasingly compelling given the advancements in robotics and the pervasiveness of cameras in our lives. In conclusion, the future of pose estimation is bright.



Appendix A

Qualitative results

Here we include the output of our various systems on the datasets Buffy, Pascal and Video-Pose datasets discussed in §IV.

A.1 CPS results

- CPS results on Buffy Figure A.1
- CPS results on Buffy Figure A.2
- CPS results on Buffy Figure A.3
- CPS results on Buffy Figure A.4
- CPS results on Pascal Figure A.5
- CPS results on Pascal Figure A.6
- CPS results on Pascal Figure A.7
- CPS results on Pascal Figure A.8
- CPS results on Pascal Figure A.9

A.2 ESM results

Video pose estimation is best viewed in video:

- ESM on VideoPose:
  http://www.youtube.com/watch?v=vnOh8_D3RhQ
- Yang and Ramanan (2011) on VideoPose:
  http://www.youtube.com/watch?v=6wNksPXoAxI
- Eichner and Ferrari (2009) on VideoPose:
  http://www.youtube.com/watch?v=xW6rwYOywTc

A.3 LLPS results

A sampling of results of LLPS are shown in Figure A.10, Figure A.11 and Figure A.12. The 32 modes are displayed as average image cluster center thumbnails. Red 'X's denote modes filtered by the cascade step in less than 0.02 seconds. The remaining models are run, and the final predicted mode is highlighted with the green rectangle. The predicted pose is overlaid as limbs for the upper and lower arm on the test image.

Figure A.1: CPS results on Buffy #1

Figure A.2: CPS results on Buffy #2

Figure A.3: CPS results on Buffy #3

Figure A.4: CPS results on Buffy #4

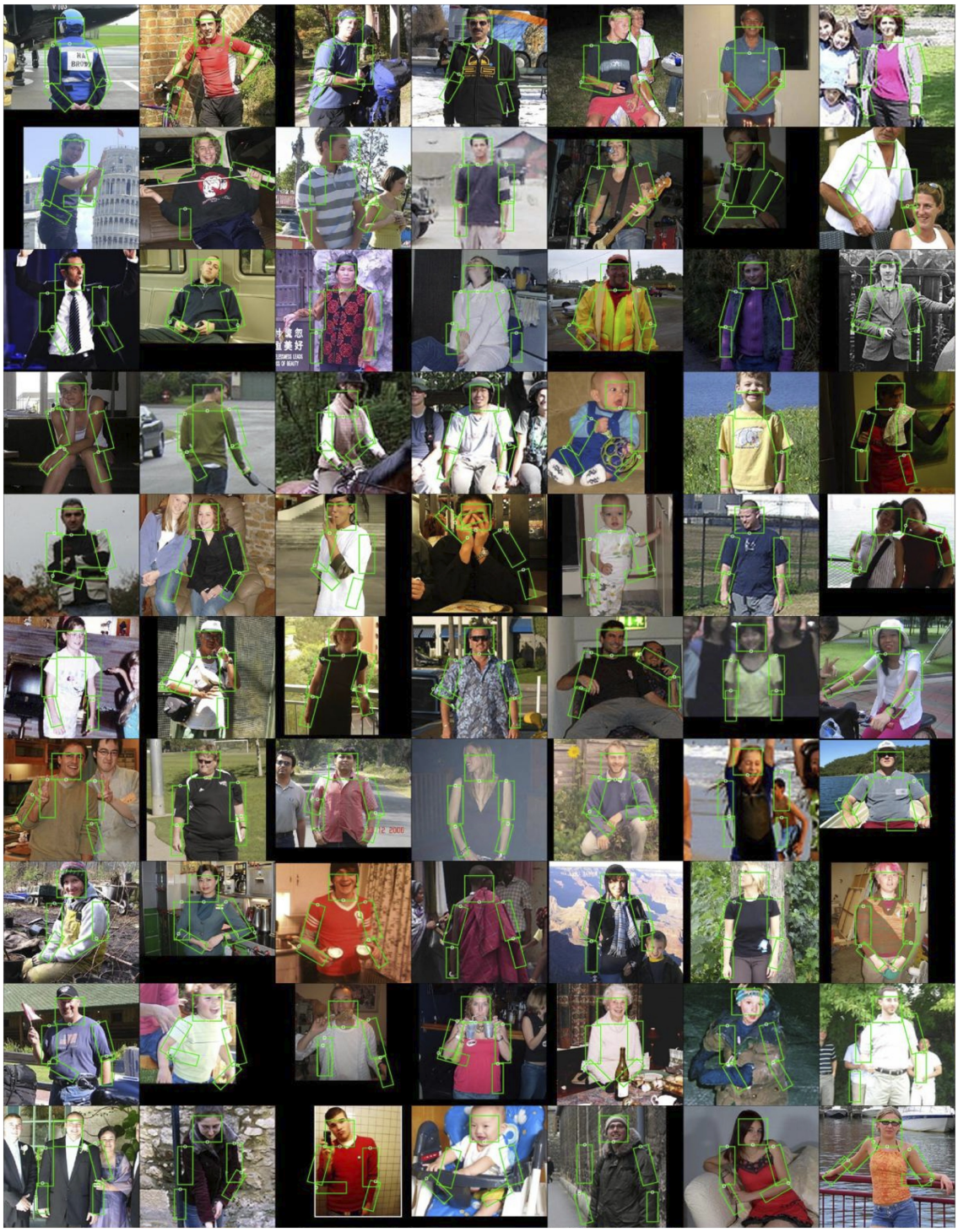


Figure A.5: CPS results on Pascal #1

Figure A.6: CPS results on Pascal #2

Figure A.7: CPS results on Pascal #3

Figure A.8: CPS results on Pascal #4

Figure A.9: CPS results on Pascal #5

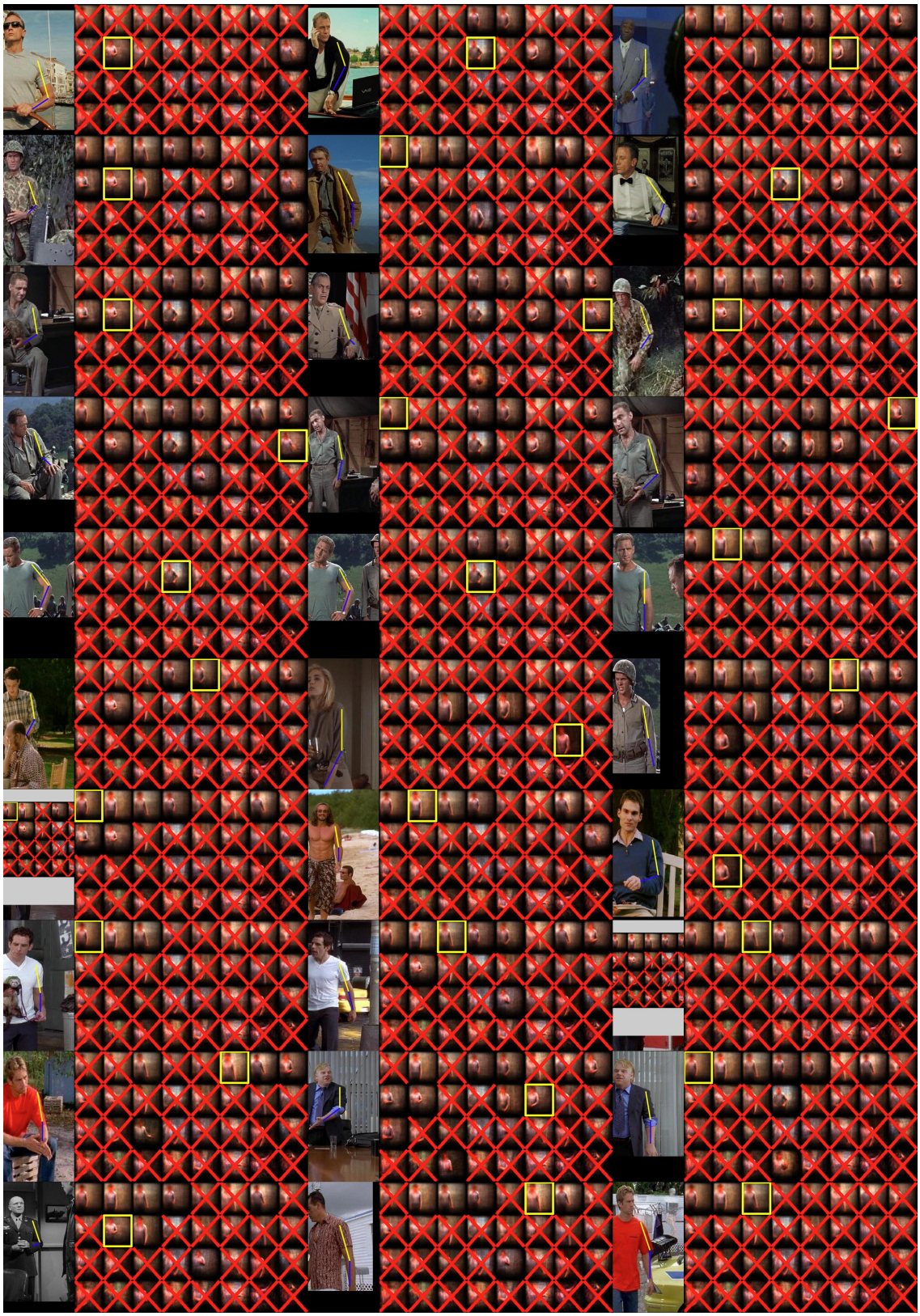


Figure A.10: LLPS results on MoviePose #1

Figure A.11: LLPS results on MoviePose #2

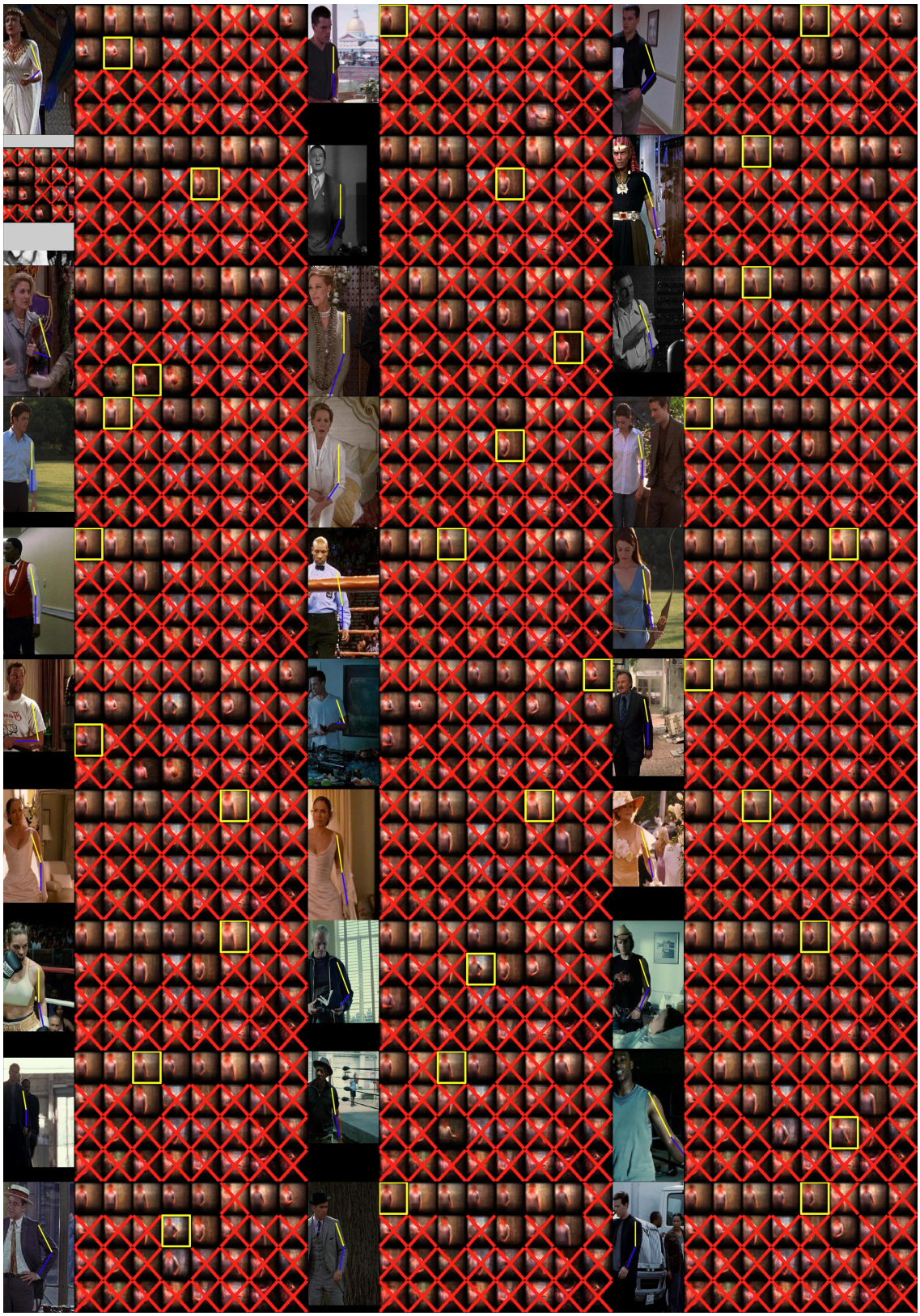


Figure A.12: LLPS results on MoviePose #3

Appendix B

Max-marginal computation

The algorithm to compute max-marginals $s_x^\star(y_i)$ for all nodes $i \in \mathcal{V}$ and all possibilities $y_i \in \mathcal{Y}_i$ is shown in Algorithm 2. The algorithm works by passing forward messages $\overrightarrow{m}_i$ in the first for-loop and backward messages $\overleftarrow{m}_i$ in the second for-loop. The backward (similarly forward) messages consist of the invariant "best possible score achieved considering only nodes topologically downstream (similarly upstream) plus the the current node". The summation of the best possible downstream scores and best possible upstream scores yields the best possible score attainable for each possibility at the current node: $m_i^\star = \overleftarrow{m}_i + \overrightarrow{m}_i - \phi_i$, where we subtract ϕ_i to prevent overcounting it. The quantity $m_i^\star$ is a vector of max-marginal scores for all possibilities y_i in node i. In the last step, we simply make explicit how $m_i^\star$ satisfies our definition of max-marginal by indexing the vector of scores via the $[\cdot]$ vector element indexing operator.

The best scoring sequence $y^\star = \arg\max_{y \in \mathcal{Y}} s(x, y)$ and the corresponding score $s(x, y^\star)$ can be obtained via the max-marginals: It must be the case that $m_i^\star[y_i^\star]$ is the largest value in $m_i^\star$, otherwise $y^\star$ would not be the highest scoring sequence—the highest scoring sequence would use the alternate $(\arg\max m_i^\star) \neq y_i^\star$ for node i.

One can also obtain witness statistics of the form $\#[y^\star(y_i)]$ denoting the number of times y_i is used as a participant (a witness) in all max-marginal sequences in the network. It is always used at least once, for its own max-marginal sequence, but may be used in

other sequences in which it is unconstrained but still chosen. These witness statistics can be computed in the same way that messages $\overrightarrow{m}_i$ and $\overleftarrow{m}_i$ are propagated, but now by accumulating histograms of argmax indices forward and backward through the graph.

For a real implementation in MATLAB, please refer to
https://github.com/bensapp/Stretchable-Models-for-Motion-Parsing/blob/master/cps/ps_model_max_inference_2clique_sparse_edges.m

---

Algorithm 2: Max-sum message passing to solve

$$s_x^\star(y_i) = \max_{y'} s(x, y') = \max_{y'} \sum_{i \in \mathcal{V}} \phi_i + \sum_{ij \in \mathcal{E}} \phi_{ij},$$

subject to: $y'_i = y_i$

Input: Factors $\{\phi_i\}, \{\phi_{ij}\}$. Tree graph $G = (\mathcal{V}, \mathcal{E})$ with (arbitrary) root node index r and topological ordering π, where $\pi_n = r$.

Output: $s_x^\star(y_i) = \arg\max_y s(x, y)$

for $i = \pi_1, \pi_2, \ldots, \pi_n$ **do**
- $\overrightarrow{m}_i = \phi_i + \sum_{j \in \text{kids}(i)} m_{j \to i}$
- **if** $i == r$ **then**
  - break
- $p = \text{parent}_\pi(i)$
- $m_{i \to p} = \max_{y_i} \phi_{ip} + \overrightarrow{m}_i$

for $i = \pi_n, \pi_{n-1}, \ldots, \pi_1$ **do**
- **if** $i == r$ **then**
  - $\overleftarrow{m}_i = \phi_i$
- **else**
  - $\overleftarrow{m}_i = \phi_i + m_{\text{parent}_\pi(i) \to i}$
- **for** $j \in kids(i)$ **do**
  - $m_{i \to j} = \max_{y_i} \phi_{ij} + \overleftarrow{m}_i$
- $m_i^\star = \overleftarrow{m}_i + \overrightarrow{m}_i - \phi_i$

$s_x^\star(y_i) \triangleq m_i^\star[y_i], \ \forall y_i \in \mathcal{Y}_i$

Bibliography

M. Andriluka and L. Sigal. Human context: Modeling human-human interactions for monocular 3d pose estimation. In *Proc. AMDO*, 2012. 136

M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Proc. CVPR*, 2008. 49

M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proc. CVPR*, 2009. 9, 39, 43, 52, 54, 60, 64, 65, 66, 68, 76, 81, 91, 113, 120, 121, 125, 128

M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. In *Proc. CVPR*, 2010. 5

A.O. Balan and M.J. Black. An adaptive appearance model approach for model-based articulated object tracking. In *Proc. CVPR*, 2006. 64

D. Batra and P. Kohli. Making the right moves: Guiding alpha-expansion using local primal-dual gaps. In *Proc. CVPR*, 2011. 139

S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. *NIPS*, 2001. 91

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, 1999. 71

Irving Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 1987. 95

T.O. Binford. Visual perception by computer. In *Proc. Systems and Control*, 1971. 42

C.M. Bishop. *Pattern recognition and machine learning*. Springer New York, 2006. 25, 32, 33

L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *Proc. ICCV*, 2009. URL http://www.eecs.berkeley.edu/~lbourdev/poselets. 39, 108

Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 2001. 113

C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proc. CVPR*, 1998. 49, 64

P. Buehler, M. Everingham, DP Huttenlocher, and A. Zisserman. Long term arm and hand tracking for continuous sign language TV broadcasts. In *Proc. BMVC*, 2008. 64

P. Buehler, A. Zisserman, and M. Everingham. Learning sign language by watching tv (using weakly aligned subtitles). In *Proc. CVPR*, 2009. 111

J. Carreira, F. Li, and C. Sminchisescu. Object Recognition by Sequential Figure-Ground Ranking. *IJCV*, 2011. 95

X. Carreras, M. Collins, and T. Koo. Tag, dynamic programming, and the perceptron for efficient, feature-rich parsing. In *Proc. CoNLL*, 2008. 55

M. Collins. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proc. EMNLP*, 2002. 33

T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *Proc. CVPR*, 2005. 24, 95

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005a. 92, 95

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005b. 5

V. Delaitre, D. Fouhey, I. Laptev, J. Sivic, A. Gupta, and A. Efros. Scene semantics from long-term observation of people. In *Proc. ECCV*, 2012. 133, 136

J. Deng, A. Berg, K. Li, and L. Fei-Fei. What does classifying more than 10,000 image categories tell us? In *Proc. ECCV*, 2010. 135

J. Deng, S. Satheesh, A. Berg, and L. Fei-Fei. Fast and balanced: Efficient label tree learning for large scale object recognition. In *NIPS*, 2011. 139

K. Duan, D. Batra, and D. Crandall. A multi-layer composite model for human pose estimation. In *Proc. BMVC*, 2012. 79, 80, 86

M. Eichner and V. Ferrari. Better appearance models for pictorial structures. In *Proc. BMVC*, 2009. 39, 54, 64, 66, 76, 81, 91, 93, 107, 112, 113, 125, 126, 146

M. Eichner and V. Ferrari. We are family: Joint pose estimation of multiple persons. *Proc. ECCV*, 2010. 135

M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari. Articulated human pose estimation and search in (almost) unconstrained still images. Technical report, ETH Zurich, D-ITET, BIWI, 2010. 43, 112, 120, 121, 122

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results, 2009. 2, 107, 108

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 95

R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. LIBLINEAR: A library for large linear classification. *JMLR*, 2008. 87, 97, 104

P. Felzenszwalb and D. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell Computing and Information Science, 2006. 38

P. Felzenszwalb and D.P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 2005. 8, 34, 37, 39, 40, 42, 52, 54, 64, 68, 89, 101

P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proc. CVPR*, 2008. 92

P. Felzenszwalb, R. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4, 2011. URL http://people.cs.uchicago.edu/pff/latent-release4/. 88, 95, 103, 126

V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Proc. CVPR*, 2008. 43, 47, 52, 64, 65, 66, 68, 91, 93, 106, 107, 112, 125, 126

V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Pose search: retrieving people using their pose. In *Proc. CVPR*, 2009. 1, 123, 133

M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 1973. 11, 34, 36, 42, 52

G. Fleuret and D. Geman. Coarse-to-Fine Face Detection. *IJCV*, 2001. 53, 55

D. Forsyth and M. Fleck. Body plans. In *Proc. CVPR*, 1997. 46

J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The annals of statistics*, 2000. 100

J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*. Springer Series in Statistics, 2001. 25, 33, 94

A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *Proc. ICCV*, 2007. 81

S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-class segmentation with relative location prior. *IJCV*, 2008. 102

A. Gupta, S. Satkin, A. Efros, and M. Hebert. From 3d scene geometry to human workspace. In *Proc. CVPR*, 2011. 1, 133, 136

P. Heckbert. Color image quantization for frame buffer display. *Computer Graphics*, 1982. 94

S. Ioffe and D. Forsyth. Human tracking with mixtures of trees. In *Proc. ICCV*, 2001. 45, 47

C. Ionescu, Fuxin Li, and C. Sminchisescu. Latent structured models for human pose estimation. In *Proc. ICCV*, 2011. 50

M. Isard and A. Blake. Condensation conditional density propagation for visual tracking. *IJCV*, 1998. 48, 65

T. Jaakkola and M. Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 2000. 47

E.T. Jaynes. Information theory and statistical mechanics. *Statistical Physics*, 1963. 25

S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *Proc. CVPR*, 2011. 45, 78, 79, 80, 86, 126

V. Jojic, S. Gould, and D. Koller. Accelerated dual decomposition for MAP inference. In *Proc. ICML*, 2010. 139

S. Ju, M. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated motion. In *Automatic Face and Gesture Recognition*, 1996. 39, 64, 68

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009. 13, 15, 21, 25, 28, 44, 47, 88

N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *Proc. ICCV*, 2007. 48, 71, 72, 139

A. Kulesza and B. Taskar. Structured determinantal point processes. In *NIPS*, 2010. 135

S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *Proc. ICCV*, 2003. 26

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*, 2001. 26

X. Lan and D.P. Huttenlocher. Beyond trees: Common-factor models for 2d human pose recovery. In *Proc. ICCV*, 2005. 64

D.C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *Proc. CVPR*, 2009. 136

M. Lee and R. Nevatia. Human pose tracking using multi-level structured models. *Proc. ECCV*, 2006. 66

L. Lefakis and F. Fleuret. Joint cascade optimization using a product of boosted classifiers. *NIPS*, 2010. 139

C. Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009. 96, 102

C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 2011. 138

T. Malisiewicz, A. Gupta, and A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *Proc. ICCV*, 2011. 79, 81, 86

K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. *Proc. ECCV*, 2004. 64

G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *Proc. ECCV*, 2002. 49

G. Mori, X. Ren, A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *Proc. CVPR*, 2004. 46, 95

D. Munoz, J. A. D. Bagnell, and M. Hebert. Stacked hierarchical labeling. In *Proc. ECCV*, 2010. 137

S. Nowozin, C. Rother, S. Bagon, T. Sharp, B. Yao, and P. Kohli. Decision tree fields. In *Proc. ICCV*, 2011. 115

D. McAllester P. Felzenszwalb, R. Girshick. Cascade Object Detection with Deformable Part Models. In *Proc. CVPR*, 2010. 55

O. Pele and M. Werman. Robust real time pattern matching using bayesian sequential hypothesis testing. *PAMI*, 2008. 139

S. Petrov. *Coarse-to-Fine Natural Language Processing*. PhD thesis, University of California at Bekeley, 2009. 55

D. Ramanan. Learning to parse images of articulated bodies. In *NIPS*, 2006. 43, 64

D. Ramanan and C. Sminchisescu. Training deformable models for localization. In *Proc. CVPR*, 2006. 52, 54, 76, 81, 91, 93, 113, 125

D. Ramanan, D. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. In *Proc. CVPR*, 2005. 49, 64, 65, 66, 93

X. Ren and J. Malik. Tracking as repeated figure/ground segmentation. In *Proc. CVPR*, 2007. 48, 64, 66, 102

G. Rogez, J. Rihan, S. Ramalingam, C. Orrite, and P.H.S. Torr. Randomized trees for human pose detection. In *Proc. CVPR*, 2008. 66

R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. In *Proc. ECCV*, 2002. 64

A. Rosti and M. Gales. Rao-blackwellised gibbs sampling for switching linear dynamical systems. In *Proc. ICASSP*, 2004. 85

S. Roweis and Z. Ghahramani. A unifying review of linear gaussian models. *NIPS*, 1999. 48

A. Rush and S. Petrov. Vine pruning for efficient multi-pass dependency parsing. In *Proc. NAACL*, 2012. URL http://www.aclweb.org/anthology/N12-1054. 137

B. Sapp and B. Taskar. Local linear pictorial structures (in submission). In —, 2012. 16

B. Sapp, C. Jordan, and B. Taskar. Adaptive pose priors for pictorial structures. In *Proc. CVPR*, 2010a. 39, 85

B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *Proc. ECCV*, 2010b. 16, 39, 65, 68, 81

B. Sapp, D. Weiss, and B. Taskar. Sidestepping intractable inference with structured ensemble cascades. In *NIPS*, 2010c. 16, 64, 65, 66, 67, 73, 123

B. Sapp, D. Weiss, and B. Taskar. Parsing human motion with stretchable models. In *Proc. CVPR*, 2011. 16, 39, 81, 82

T. Sebastian, P. Klein, and B. Kimia. Recognition of shapes by editing their shock graphs. *PAMI*, 2004. 95

G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Computer Vision*, 2003. 49

S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *Proc. ICML*, 2007. 33

J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proc. CVPR*, 2011. 3

L. Sigal and M.J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *Proc. CVPR*, 2004. 48

L. Sigal, S. Bhatia, S. Roth, M.J. Black, and M. Isard. Tracking loose-limbed people. In *Proc. CVPR*, 2004a. 47, 66

L. Sigal, M. Isard, B.H. Sigelman, and M.J. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In *NIPS*, 2004b. 64, 66

V. Singh and R. Nevatia. Action recognition in cluttered dynamic scenes using pose-specific part models. In *Proc. ICCV*, 2011. 48

C. Sminchisescu and B. Triggs. Estimating articulated human motion with covariance scaled sampling. *IJRR*, 2003. 64, 66

D. Sontag. *Approximate Inference in Graphical Models using LP Relaxations*. PhD thesis, Massachusetts Institute of Technology, 2010. 139

P. Srinivasan and J. Shi. Bottom-up recognition and parsing of the human body. In *Proc. ICCV*, 2007. 46, 95

M. Sun and S. Savarese. Articulated part-based model for joint object detection and pose estimation. In *Proc. ICCV*, 2011. 45, 79, 80

M. Sun, M. Telaprolu, H. Lee, and S. Savarese. An efficient branch-and-bound algorithm for optimal human pose estimation. In *Proc. CVPR*, 2012. 44

C.J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Proc. CVIU*, 2000. 39

G.W. Taylor, R. Fergus, G. Williams, I. Spiro, and C. Bregler. Pose-sensitive embedding by nonlinear nca regression. *NIPS*, 2010. 50

Y. Tian, C.L. Zitnick, and S.G. Narasimhan. Exploring the spatial hierarchy of mixture models for human pose estimation. In *Proc. ECCV*, 2012. 79, 80, 86

A. Toshev, B. Taskar, and K. Daniilidis. Object detection via boundary structure segmentation. In *Proc. CVPR*, 2010. 95

K. Toyama and A. Blake. Probabilistic tracking with exemplars in a metric space. *IJCV*, 2002. 49

D. Tran and D. Forsyth. Improved Human Parsing with a Full Relational Model. In *Proc. ECCV*, 2010. 44, 76, 81, 91, 126

K.N. Tran, I.A. Kakadiaris, and S.K. Shah. Modeling motion of body parts for action recognition. In *Proc. BMVC*, 2011. 1

R. Urtasun and T. Darrell. Local Probabilistic Regression for Activity-Independent Human Pose Inference. In *Proc. CVPR*, 2008. 66

L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 1979. ISSN 0304-3975. doi: 10.1016/0304-3975(79)90044-6. URL http://www.sciencedirect.com/science/article/pii/0304397579900446. 28

B. Van Roy. *Learning and value function approximation in complex decision processes*. PhD thesis, Massachusetts Institute of Technology, 1998. 139

P. Viola and M. Jones. Robust real-time object detection. *IJCV*, 2002. 53, 55, 95, 100, 138, 139

H. Wang and D. Koller. Multi-level inference by relaxed dual decomposition for human pose segmentation. In *Proc. CVPR*, 2011. 137

J. Wang, D. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *PAMI*, 2007. 66

X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T.X. Han. Contextual weighting for vocabulary tree based image retrieval. In *Proc. ICCV*, 2011a. 138

Y. Wang and G. Mori. Multiple tree models for occlusion and spatial constraints in human pose estimation. In *Proc. ECCV*, 2008. 44, 78, 79, 80, 135

Y. Wang, D. Tran, and Z. Liao. Learning hierarchical poselets for human parsing. In *Proc. CVPR*, 2011b. 45, 79, 80, 108, 111, 126, 133

K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, 2006. 81

D. Weiss and B. Taskar. Structured prediction cascades. In *Proc. AISTATS*, 2010. 16, 55, 56

D. Weiss, B. Sapp, and B. Taskar. Structured prediction cascades (under review). In *JMLR*, 2012. 16, 137

Y. Yang and D. Ramanan. Articulated pose estimation using flexible mixtures of parts. In *Proc. CVPR*, 2011. 45, 78, 79, 80, 82, 86, 88, 91, 113, 116, 120, 121, 126, 128, 130, 131, 146

B. Yao and L. Fei-Fei. Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. *PAMI*, 2012. ISSN 0162-8828. 1

H. Zhang, A. C. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *Proc. CVPR*, 2006. 81

Q. Zhu, L. Wang, Y. Wu, and J. Shi. Contour context selection for object detection: A set-to-set contour matching approach. *Proc. ECCV*, 2008. 95

X. Zhu and D. Ramanan. Face detection, pose estimation and landmark localization in the wild. In *Proc. CVPR*, 2012. 45, 78, 79, 80

X. Zhu, C. Vondrick, D. Ramanan, and C.C. Fowlkes. Do we need more training data or better models for object detection? In *Proc. BMVC*, 2012. 126, 135