Recognition of Human Poses Using Pictorial Structures

Jevgenijs Ivanovs

Supervisors: **Prof. Dr. Dariu M. Gavrila**
Dr. Ad J. Feelders

Universiteit Utrecht

Submitted for the degree of
Master in Applied Computing Science

Department of Information and Computing Sciences
Faculty of Science
Universiteit Utrecht
The Netherlands

May 2007

Recognition of Human Poses Using Pictorial Structures

Jevgenijs Ivanovs

Submitted for the degree of Master in Applied Computing Science
May 2007

Abstract

Human pose recognition is an important problem in the computer vision domain. Pose recognition can be used as a basis for surveillance, human computer interaction and motion analysis systems, provided recognition is done reliably and efficiently. A 2D approach with explicit shape model is used in this work. More specifically, we consider a probabilistic model called pictorial structure, which can be also viewed as a Bayesian network. The model defines a posterior probability distribution over human poses given an input image. For any given image we sample full body configurations from the posterior distribution in an efficient way. Sampled configurations are then ordered according to their posterior probabilities. The final scoring of body poses and the best candidate selection is a challenging problem of its own and is not considered in this work.

We extended and improved the pictorial structure model in a few ways. The underlying body part appearance model received much attention in this work. The pictorial structure model was analyzed as a model for human pose recognition and as a classifier between person presence and non-presence. Experimental results suggest that the pictorial structure model is a good alternative for human pose recognition in real-world images.

Declaration

The work in this thesis is based on research carried out at the Intelligent Systems Lab Amsterdam, Faculty of Science, University of Amsterdam, the Netherlands. No part of this thesis has been submitted elsewhere for any other degree or qualification and it all my own work unless referenced to the contrary in the text.

Acknowledgements

I would like to express my profound gratitude to Wojtek Zajdel for his guidance, useful comments and insightful discussions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction and Motivation

Human pose recognition is an important task in the computer vision domain. It can form a basis for many important applications. Possible application areas include surveillance, human computer interaction, automatic video annotation, motion analysis and many others.

For a concrete example consider crime prevention and aggression detection system. There is a huge number of surveillance cameras monitoring people. The cameras are installed on the corners of buildings, at train stations, parking lots, airports, just to name a few. For example, there are at least 500,000 surveillance cameras in London as of 2005. Surveillance cameras produce enormous amount of data, therefore, it is important to have an autonomous system able to analyse these data. Such a system for a single camera may output aggression level and draw somebody's attention at the moment when this level exceeds some specified threshold. Human pose recognition algorithm, provided it is reliable and efficient, can be employed as a basic component of the above described system. Estimated human poses may be used to initialize a human tracking algorithm or they may be directly fed into an aggression classifier together with other cues.

The human body is complex and estimation of human poses involves estimation of high number of parameters. There are large differences in body dimensions between persons. Moreover, loose clothing, difference in appearance, huge number of possible poses, environment and illumination changes and many other factors contribute to the complexity of the human pose recognition task.

1.1.1 Work Summary

In present work we consider single real-world images coming from a static camera. We focus on human pose recognition from these images. No assumptions are made about possible poses. A 2D model based approach to human pose recognition is used in this work. More specifically, we consider a known probabilistic model called pictorial structure, which can be also viewed as a Bayesian network.

In this work the pictorial structure model for human pose recognition was extended and improved in a few ways. The underlying body part appearance model is an important component of a pictorial structure. We devote much time for developing and testing these models. Pictorial structure parameters are learned from training data in a rigorous way. Finally, we analyze the model both as a model for human pose recognition and as a classifier between person presence and non-presence.

In this paragraph we very briefly describe how a new input image is processed using a pictorial structure. The parameterized model defines posterior probability distribution over human poses given the image. A number of poses is sampled from the posterior distribution in an efficient way. The final scoring of body poses is a challenging problem of its own. We approach it in a very simplistic way and report our findings based on the list of candidate poses ordered according to their posterior probability.

1.2 Previous Work

In this section we briefly review state of the art in the domain of human pose recognition. A detailed account of existing methods of the analysis of human movement can be found in [10]. Many criteria can be used to classify previous work. We list some of these

- number of people: single person versus multiple people;

- environment: indoor versus outdoor environment;
- type of input data: single image, video sequence or video sequences from a number of cameras;
- appearance descriptors: descriptors based on background subtraction, descriptors based on edges, Haar-like features, etc.
- estimation method;
- type of the model used.

In the following paragraph we consider the last criterion.

The approaches to human pose estimation can be split in three groups: 2D model approaches, 3D model approaches and approaches without explicit model [10]. The approaches from the first group use explicit a priori knowledge of how the human body looks in 2D. The 2D models are usually stick figures, wrapped around with ribbons or 'blobs'. Systems of this kind often assume knowledge of the viewpoint under which people are observed. A 3D model approach is about 3D pose estimation from 2D points. It is mainly used for tracking, and often assumes that data from multiple cameras are available. In case of 3D model one can take advantage of the knowledge of kinematic and shape properties of the human body. Moreover, one can predict such events as occlusion and collision. In case of model based approaches the pose recognition problem is usually solved by an analysis-by-synthesis approach, where one seeks for the minimum of distance function between image features and features synthetically generated from the model. Model-free approaches to pose recognition are based on examples. A collection of representative examples is stored in a database. For each input image, a similarity search is performed. The main problem of example-based approach is a need of a huge number of examples to cover large variability of human poses.

We describe some 2D approaches in more detail. Many approaches use so called bottom-up strategy. At the first step candidate body parts are detected, then a search over possible assemblies of candidate parts is performed. There is usually a huge number of assemblies to be considered, therefore either a small subset of candidate parts is considered, either greedy type search strategies are used. The bottom-up methods often output a partial configuration. In [23] authors model body parts with a pair of parallel line segments. Approximate integer quadratic programming is used to find the best assembly of parts. In [19] authors split an image into segments and assume that body parts often pop out as a single segment. They classify the segments using a set of cues. Finally, authors evaluate partial body configurations consisting of three parts. In [17] authors consider only three parts from the start: head, head-shoulders region and legs (as a single part). They aim at building strong and robust part detectors. Assemblies of parts are evaluated in a probabilistic way. A very similar approach is used in [29].

Human pose recognition can also be approached in a top-down fashion, where coarse-to-fine strategy is employed. This strategy is often used for 3D models in the context of tracking. For example see [11], where an initial human pose prediction is based on joint angle acceleration. Then pose is adjusted in a hierarchical way starting from torso/head. There are also 2D model based approaches, which employ top-down startegy. In [15] authors work with blobs of foreground pixels. First, authors analyze if the blob represents one person or multiple people. In case of a single person an attempt to detect a type of pose is made. Next, a set of vertices on a contour of the blob is defined. Finally, authors label these vertices taking different strategies for different types of poses.

The pictorial structure model integrates both top-down as well as bottom-up reasoning mechanisms. Similar to bottom-up methods we can first process a given image and find responses of single parts placed at different locations in the image. We, however, do not threshold these responses. No lists of candidate parts are made. Similar to top-down methods we assume that body parts are salient when considered in the context of full body configuration. In other words configuration of parts is the most distinctive feature. Therefore, we aim at discovering the full configuration as a single thing.

The pictorial structure model was introduced more than 30 years ago by Fischler and Elschlager [9] as a model of a human face. The pictorial structures, however, were not broadly applied to recognition problems due to computational difficulty of matching them to images. The model gained some popularity with the paper [6], where an efficient matching algorithm based on fast computation of generalized distance transforms was presented. Another contribution of [6] is a probabilistic formulation of the pictorial structure matching problem. Probabilistic approach was further developed in [8]. An interesting extension of tree models is presented in [16], where a common factor model is integrated into the pictorial structure to introduce additional correlations. Some work has been also done to improve appearance models of parts. An interesting example is [25], where authors use SVMs as appearance models of parts. In [30] authors construct a very complex human body model. Inference is done by a direct sampling from the posterior, using Sequential Monte Carlo simulation enhanced with annealing.

Chapter 2

Pictorial Structures

2.1 Generic Model

Pictorial structure is a collection of parts of an object arranged in a deformable configuration. Each part of the model encodes local visual properties of the object, and the deformable configuration is characterized by spring-like connections between certain pairs of parts. The appearance model of each part is given by a function, which measures how much a location in an image looks like the corresponding part. The best match of such a model given image is found by minimizing energy function that measures both a match cost for each part and a deformation cost for each pair of connected parts. See figure 2.1 for an example of two pictorial structure models matched against images.

The appearance model for each part can be fairly generic. This is because we do not recognize parts independently, but together with other parts of the object. This is different from most methods which use part-based representations, where in an initial phase parts are recognized individually, and in the next phase they are assembled into groups to form objects. Individual recognition of parts requires quite complex part models, whereas if we rely on their configuration as a distinguishing feature we can use quite simple appearance models for each part. Finally, we note that pictorial structure framework is independent from the scheme used to model appearance of body parts.

As mentioned above, a deformable configuration of parts is represented by con-

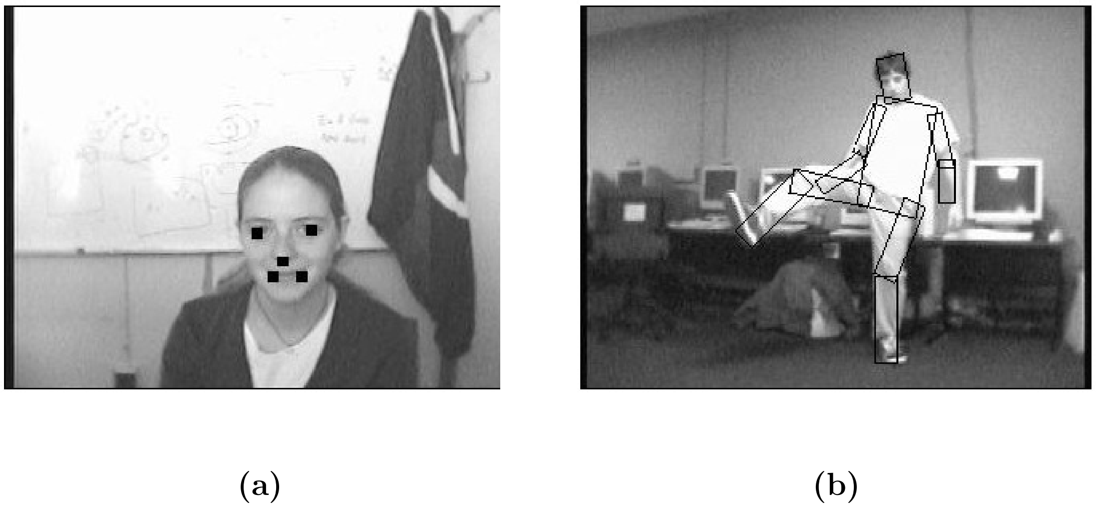

Figure 2.1: Examples of two pictorial structure models matched against images. (a) detection of a face; (b) detection of a human body. Adapted from [8].

nections between pairs of parts. A connection between two parts indicates a relation between locations of these parts. This relation can be again very generic. It can be something very simple like a constraint on the distance between parts or, as in case of an articulated object, it can specify a distribution of joint angles and distances between joint points. Since the appearance models of parts and the relationships between parts can be fairly generic, pictorial structures provide a flexible framework for object recognition problems [8].

Before we give a mathematical definition of a model and a matching problem, we make a short note about notation we use. We use capital letters to denote such mathematical objects as sets, graphs and random variables. In a probabilistic context we use small letters (e.g. x) to denote a realization, that is a particular value taken on by a random variable (X). Bold symbols are used to stress that an object under consideration is a vector.

A natural way to express pictorial structure is in terms of an undirected graph $G = (V, E)$, where vertices $V = \{v_0, \ldots, v_{n-1}\}$ correspond to the n parts and there is an edge $(v_i, v_j) \in E$ for each pair of connected parts v_i and v_j. An instance of an object is given by a configuration $\mathbf{l} = (l_0, \ldots, l_{n-1})$, where each l_i specifies the location of part v_i. The location of each body part can simply be the position of the part in an image. More complex parameterizations are also possible. For example, for the person model it makes sense to use not only position of a limb in the image, but also its orientation and scale.

The problem of matching a pictorial structure model to an image can be defined as an energy minimization problem. Let $m_i(l_i)$ be a function measuring the degree of mismatch, when part v_i is placed at location l_i in the image. For a given pair of connected parts let $d_{ij}(l_i, l_j)$ be a function measuring the degree of deformation of the spring-like connection between parts v_i and v_j, when they are placed at locations l_i and l_j in the image respectively. Now the optimal match of the model to the image is naturally defined as

$$\mathbf{l}^* = \arg\min_{\mathbf{l}} \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right), \tag{2.1}$$

which is a configuration minimizing the sum of the match costs m_i and sum of deformation costs d_{ij} for connected pairs of parts. Generally, the deformation costs depend only on relative positions of connected parts, making the model invariant to certain global transformations.

2.2 Probabilistic Framework

In this section we will consider a pictorial structure in a probabilistic framework, where the energy minimization problem defined in the previous section becomes a maximum a posteriori problem (MAP). Energy minimization or MAP formulation characterizes only the optimal solution. It might be useful to consider several good matches and subsequently select the final configuration using temporal information, symmetry in clothing or some other criterion. As we will see further, the above statement can be explained by the fact that our model is just a rough approximation of reality, and thus the configuration with the highest posterior probability may not be the correct one. The probabilistic formulation provides a natural way of finding several good matches of a model to an image. We can achieve this by sampling object configurations from their posterior probability distribution given the observed image. Moreover, the probabilistic framework can be used to learn parameters of the model.

A standard way of approaching object recognition in a probabilistic setting is as follows. Let θ be a set of parameters of a model, I denote an image, and a random vector $\mathbf{L} = (L_0, L_1, , \ldots, L_{n-1})$ denote a configuration of the object, that is a location for each part. The distribution $p(I|\mathbf{L}, \theta)$ is the probability of observing a particular image given the location of the object. The distribution $p(\mathbf{L}|\theta)$ measures the prior probability of a particular configuration. Finally, the posterior distribution, $p(\mathbf{L}|I, \theta)$ specifies the probability that the object is at a particular location given the observed image. Using Bayes' rule the posterior can be written as

$$p(\mathbf{L}|I, \theta) \propto p(\mathbf{L}|\theta)p(I|\mathbf{L}, \theta), \tag{2.2}$$

because θ is fixed (obtained in a learning phase). At this point we stop writing θ in the conditioning set and assume it is implicitly present there. The vector of model parameters θ will be made explicit in the discussion of learning model parameters.

A common difficulty in the Bayesian approach is to determine a prior distribution $p(\mathbf{L})$. Often the prior is taken to be uniform, that is prior knowledge is completely rejected. However, only with an informative prior we get a truly Bayesian approach [18]. For pictorial structures, the prior over configurations of the object encodes information about relative locations of the parts. In case of a human model, it specifies a distribution of human poses. The prior is an important component in the pictorial structure model, thus our approach is truly Bayesian.

In order to make inference feasible, we need to further decompose $p(\mathbf{L}|I)$, making some assumptions on the way. First, let us consider $p(I|\mathbf{L})$, the probability of seeing an image given an object configuration. Suppose that given the configuration $\mathbf{L} = \mathbf{l}$ we can split the image I into the set of subimages $I_B, I_0, \ldots, I_{n-1}$, so that a subimage $I_i, i = 0, \ldots, n-1$ contains only the object part v_i and pixels just around it; I_B is the background. For illustration see figure 2.2. Moreover, we assume that what we see in one subimage is independent from the location and content of the others, given model parameters θ. By doing so we disregard some information, e.g. symmetry in clothing. The following decomposition, however, is essential for our approach.

Having made the assumption we can write

$$p(I|\mathbf{L}) = p(I_B, I_0, \ldots, I_{n-1}|\mathbf{L}) = p(I_B|\mathbf{L})p(I_0|\mathbf{L})\ldots p(I_{n-1}|\mathbf{L}) =$$
$$p(I_B)p(I_0|L_0)\ldots p(I_{n-1}|L_{n-1}) \propto p(I_0|L_0)\ldots p(I_{n-1}|L_{n-1}),$$

where $p(I_B)$ is taken to be uniform. Thus we obtain

$$p(I|\mathbf{L}) \propto \prod_{i=0}^{n-1} p(I_i|L_i). \tag{2.3}$$

From the above discussion, we see that this approximation is good if parts do not overlap and are not occluded. This, however, is rarely true. As a consequence we may get a multi modal posterior, where the correct configuration may correspond to some local maximum. This is one of the reasons to favor sampling with subsequent processing and not to rely on the MAP solution.

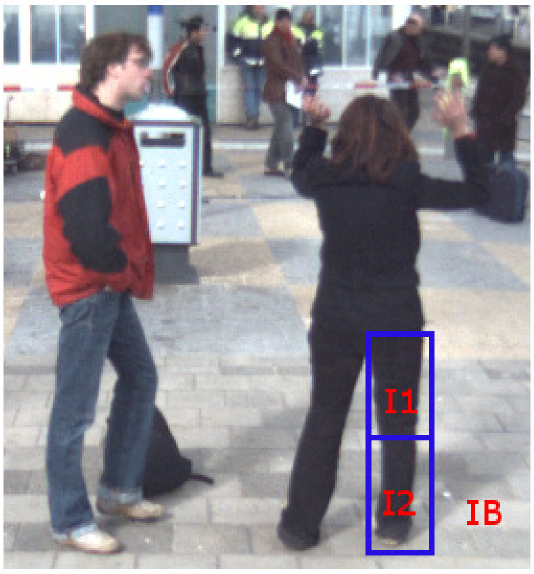

Figure 2.2: Splitting an image given a configuration $\mathbf{l} = (l_1, l_2)$ and model parameters θ.

Consider the prior distribution over object configurations, $p(\mathbf{L})$. We take our undirected graph G (the one which specifies a pictorial structure) to be an independence graph of a graphical model representing the prior distribution $p(L_0, \ldots, L_{n-1})$. For an introduction to graphical models and Bayesian networks see [28] and [20]. Efficient inference algorithms, such as belief propagation, exist if the independence graph G is a tree. One can use these inference algorithms to compute the best match (MAP) as well as to sample from the posterior. Restricting the independence graph to a tree may seem natural. For example, we can take a tree corresponding to a skeletal structure of an articulated object. A tree model, however, introduces strong assumptions about independences between (body) parts in the pictorial structure. Thus it misses important information about the articulated object such as coordination of limbs, which is necessary for balance. For simplicity in the remaining part of the thesis we assume that the independence graph is a tree.

We find it more convenient to work with a *directed* independence graph $G = (A, V)$. As before $V = \{v_0, \ldots, v_{n-1}\}$ is a set of vertices, where a vertex v_i corresponds to a location l_i of a part. Set $A = \{(v_i \to v_j)\}$ is the set of arcs, where v_i is a parent node(part) and v_j is a child node. We pick v_0 to be a root of a directed tree. The directed tree G represents now a Bayesian network, where the joint probability distribution factorises as

$$p(\mathbf{L}) = p(L_0) \prod_{(v_i \to v_j) \in A} p(L_j | L_i). \tag{2.4}$$

In the previous paragraphs we obtained a factorial decomposition (2.3) of $p(I|\mathbf{L})$, the probability of seeing an image given that the object is at some configuration, and a decomposition (2.4) of $p(\mathbf{L})$, the prior probability that the object will assume a particular configuration. These can be substituted into (2.2) yielding

$$p(\mathbf{L}|I) \propto p(L_0) \prod_{(v_i \to v_j) \in A} p(L_j | L_i) \prod_{v_i \in V} p(I_i | L_i). \tag{2.5}$$

Next we will show a relation between the MAP solution and the optimal solution of the energy minimization problem (2.1). For this assume that the priors $p(L_i)$ are uniform distributions. Then we obtain from equation (2.5)

$$p(\mathbf{L}|I) \propto \prod_{(v_i, v_j) \in E} p(L_j, L_i) \prod_{v_i \in V} p(I_i | L_i). \tag{2.6}$$

Suppose now that we want to find the MAP solution to the above problem. So we need to find a configuration of body parts $\mathbf{l}^{**}$ maximizing $p(\mathbf{l}|I)$, which is the same as minimizing $-\log p(\mathbf{l}|I)$. Using equation (2.6) we see that we need to minimize $\sum_{v_i \in V} -\log p(I_i|l_i) + \sum_{(v_i,v_j)\in E} -\log p(l_i, l_j)$. Now let $m_i(l_i) = -\log p(I_i|l_i)$ be a match cost measuring how well part v_i matches the image data at location l_i, and $d_{ij}(l_i, l_j) = -\log p(l_i, l_j)$ be a deformation cost measuring how well the locations of parts v_i and v_j agree with a prior model. Notice now that the MAP solution $\mathbf{l}^{**}$ exactly corresponds to the optimal solution $\mathbf{l}^{*}$ obtained by solving the energy minimization problem (2.1). Thus the probabilistic framework allows us to formulate the optimization problem from the previous section. Moreover, it allows us to apply approximate inference techniques, such as sampling.

2.3 Human Body Model

An obvious way to model a human body on a coarse level is to consider a puppet consisting of $n = 10$ body parts: head, torso, upper and lower arms, upper and lower legs. Further in the text we will use abbreviations: H for Head, T for Torso, RLL for Right Lower Leg and so on. Our tree structured Bayesian network has 10 vertices and 9 arcs. For the time being we chose torso to be a root of the tree, as illustrated in figure 2.3.

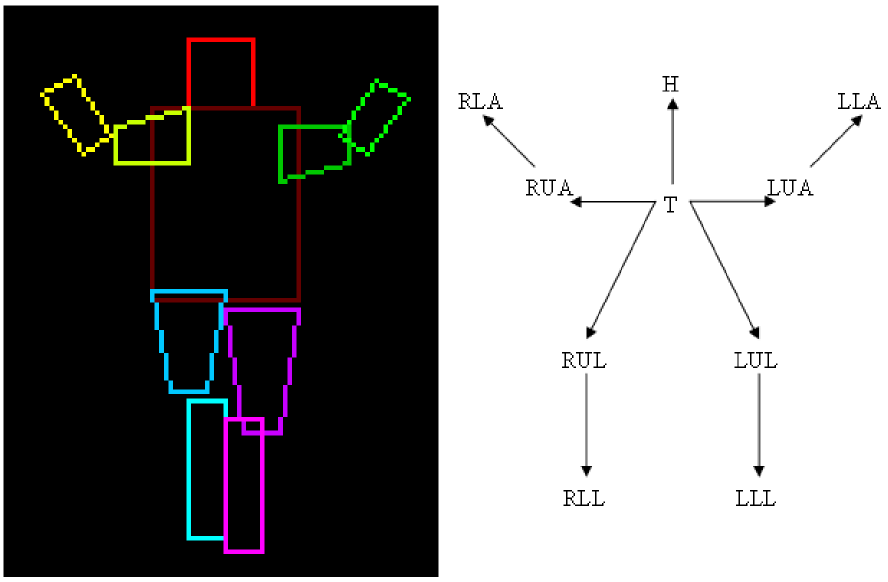

Figure 2.3: Human body model and the underlying Bayesian network.

2.3.1 Parts of a Human Body

We assume that the image of an object is generated by means of a perspective projection. Human body parts are more or less cylindrical, so their projections can be approximated with rectangles. In order to make a better fit of a model to a possible human projection, we represent upper arms and legs with trapezoids. For the moment suppose that we fix geometrical shapes of all the limbs by hand. Later, in the model learning phase, we will adjust shapes to training data.

Assume that the scale of a human in an image is known. This is a reasonable assumption if our recognition algorithm is to be applied to a region of interest produced by a detection or tracking algorithm. Given size of the region and knowing that a person is in upright position we can infer the scale of the human. Now we can resize the region of interest to some standard size, so that our model fits a human projection contained in the region.

It is convenient to consider a local coordinate system for every body part. According to our above discussion we have specifications of all trapezoids representing body parts. Thus we can choose 4 points $(x_0, y_0), \ldots, (x_3, y_3)$ in the local coordinate system of every part so that these points define the corresponding trapezoid. Note that these points defining the shape of a body part are fixed or learned beforehand. We can think of them as living in a vector θ of model parameters, which is implicitly present in the conditioning set.

Next we need to describe a location of a body part in an image, where location means full specification of the geometrical figure representing the body part in an image coordinate system. Note that the image is a properly scaled region of interest of the original image. In order to define a trapezoid representing the body part in the image coordinate system, we specify x and y coordinates of the origin of the part's local coordinate system and orientation o, which can be viewed as a rotation angle. Finally, notice that the width of a rectangle representing the body part comes from a diameter of a cylinder and is fixed, while the length of the rectangle may vary due to foreshortening (consider an arm pointing towards the observer). The same also holds for a trapezoid. Thus we introduce a parameter specifying the amount of foreshortening. Therefore, the location of a part is parameterized by (x, y, o, s), where (x, y) is the position of the origin of the part's coordinate system (anchor point) in the image coordinates; o specifies the orientation and s is the amount of foreshortening (scale) of the part. For illustration see figure 2.4.

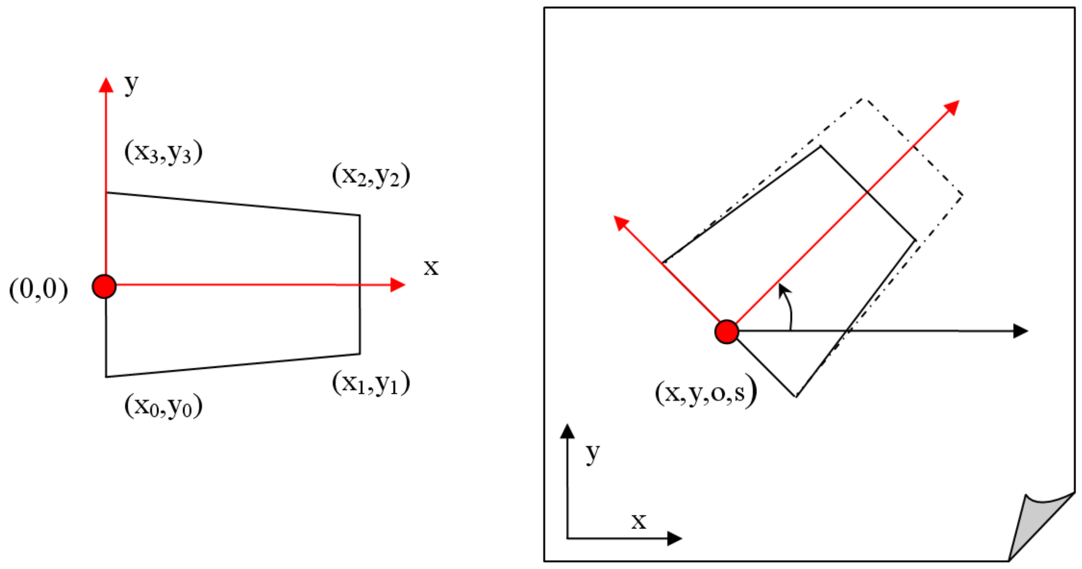

Figure 2.4: Body part's model in its own coordinate system; the part positioned in the image plane according to the location specified by (x, y, o, s).

In the scope of this thesis we restrict our attention to a discrete parameter case. Thus all the components of a location of a part are to be discretized in some way. For example, it is usual to discretize orientation into 32 values: $0, \frac{1}{16}\pi, \ldots, \frac{31}{16}\pi$. Even though image coordinates are discrete, we are not obliged to use them in our parameterization. We can superimpose a coarser grid, getting smaller domain from one side; from the other side it will be possible to attach an anchor of the limb to the points of the grid only. Thus the proper discretization is a tradeoff between efficiency and accuracy. Finally, let us denote a set of possible values of the location of the body part by $\mathcal{L}$. Suppose that x, y, o, s take their values in sets $\mathcal{X}, \mathcal{Y}, \mathcal{O}, \mathcal{S}$ respectively. We also assume that these sets are the same for all body parts. Now we have that $|\mathcal{L}| = |\mathcal{X}||\mathcal{Y}||\mathcal{O}||\mathcal{S}|$. It is important to notice that in practice $|\mathcal{L}|$ is very large, because it is a product of sizes of the domains of all the variables forming the location vector.

Transformation from the Part's Coordinate System to the Image Coordinate System

Given a location $l = (x, y, o, s)$ of a body part (do not confuse with a configuration of all body parts $\mathbf{l}$) and a point $p = (p_x, p_y)$ in part's local coordinate system we would like to obtain coordinates of this point $p' = (p'_x, p'_y)$ in an image plane. That is we need a transformation $p \mapsto \phi_l(p) := p'$. This transformation is required for various tasks:

- draw the body part in the image plane given part's location l (we compute vertices of a trapezoid in the image plane $\phi_l((x_i, y_i))$, for $i = 0, 1, 2, 3$ and connect them with lines);
- compute some dissimilarity score between the image and the part positioned at the location l (see chapter 3 on detailed discussion of part's appearance model);
- compute coordinates of a joint in the image plane (see section 2.3.3 on the prior distribution over human poses).

One can verify that transformation $\phi_l(p)$ as was defined above has the following form

$$\begin{aligned} p'_x &= \phi_l(p)_x = x + \cos(o)p_x s - \sin(o)p_y \\ p'_y &= \phi_l(p)_y = y + \sin(o)p_x s + \cos(o)p_y, \\ &\text{where } l = (x, y, o, s) \end{aligned} \tag{2.7}$$

We assume that foreshortening of a limb is done along the x axis of the local part's coordinate system.

2.3.2 Appearance Model of Parts

The pictorial structure model does not put any restrictions on a set of possible appearance models of parts of the modeled object. On one hand an appearance model of a part can be fairly generic. As we have pointed out before, this is the case, because we do not attempt to find body parts separately, but we match the whole human model at once assuming that the actual configuration is discriminative. Therefore, it is possible to use simplistic appearance models boosting execution speed of the inference algorithm. On the other hand a very simple appearance model would provide a relatively sparse cue, which requires dense sampling. Moreover, notice that a match against background clutter commonly should score less than a match against well standing out body part. Otherwise, every body part of our model and thus the whole model will match to clutter very well.

For an overview of possible appearance models we provide two examples. One example would be a simple predictor based on the number of foreground pixels belonging to a part and the number of foreground pixels belonging to some border area around the part [8], so that in ideal case we expect first number to be relatively big and second number to be relatively small. More complex models are also possible. An example of modeling appearance of body parts using support vector machines (SVM) can be found in [25].

In terms of our probabilistic framework we need a way to compute $p(I_i|l_i)$, an image probability given a location of a body part. A common approach is to compute some representative score or a set of cues. Then we assume that this score encompasses all the information, which is needed to decide upon the image probability(*), thus $p(I_i|l_i) \propto f(s)$. Moreover, we can view $f(s)$ as a function defining the probability distribution of the score s. That is we assume that the image probability is proportional to a probability of the corresponding score

$$p(I_i|l_i) \propto f(s) = p(s). \tag{2.8}$$

To see this consider the following equation. $p(I_i|l_i) = p(I_i, s|l_i) = p(I_i|s, l_i)p(s|l_i) \propto p(s|l_i) = p(s)$, where the score s is viewed as a function of the image given the location. The image probability given the score $p(I_i|s, l_i)$ is uniform, because of the assumption (*). Finally, we assume that the score is independent of the location. Next, we need to specify an appropriate distribution for s. For this we may choose some parametric family of distributions, e.g. normal or log-normal families. Parameters of the distribution then may be learned from training data. More detailed discussion about appearance models as well as experimental results for those can be found in the following chapters.

2.3.3 Prior Distribution of Human Poses

A tree-structured Bayesian network is chosen to model a prior probability of human poses. This provides us with factorization (2.4), which consists of $p(L_0)$ and $p(L_j|L_i), (v_i \to v_j) \in A$. The prior of the root (torso) $p(L_0)$ can be taken uniform or we can assume it is proportional to $p(O_0)$, where O_0 denotes orientation of the torso. Then we can learn $p(O_0)$ from data. This may provide us with information that observed people, for example, are mostly in upright pose (more precisely: torso is upright).

It is, however, more important to define appropriate conditionals $p(L_j|L_i)$, which model deformation potentials of two connected body parts v_i and v_j. For an articulated object, like a human, pairs of parts are connected by flexible joints. The location of the joint is specified by a point (x_{ij}, y_{ij}) in the coordinate system of the part v_i and a point (x_{ji}, y_{ji}) in the coordinate system of the part v_j.

Recall that a location of a body part is given by a position of the anchor point, orientation and foreshortening or scale parameter. We have noticed that it is very convenient to make the anchor point (origin of a local coordinate system) of a child part to coincide with the joint of this part connecting it to the parent part, which implies $(x_{ji}, y_{ji}) = (0, 0)$. The advantages of this will be discussed in the following. We have a tree model, therefore, every node except the root has a unique parent, and thus the locations of anchor points are well defined. For the root we can choose any point to be an anchor. A pair of connected parts is illustrated in figure 2.5.

The human body is flexible. A person can stretch, bend, twist, etc. Moreover, different people have different build and wear different clothes. This implies that our model should be flexible enough to be able to fit to various images of people. Thus we define probability distributions on relative locations of joints for every two connected parts, which assigns to any configuration of these two body parts some score representing how likely the configuration is. Notice that we can control rigidness of the pictorial structure via a choice of distributions. Spread distribution

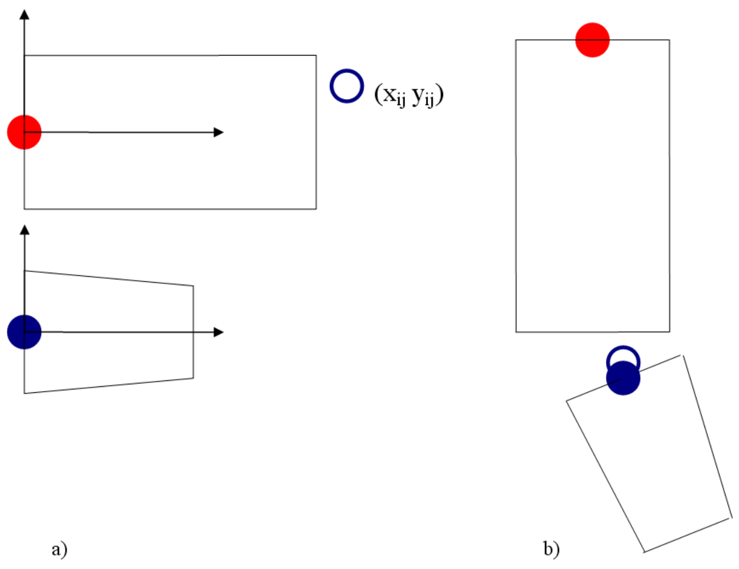

Figure 2.5: a) two parts v_i and v_j of a human model in their own coordinate systems; b) a configuration of these parts.

would make our model more flexible, whereas peaked distribution would make it more rigid.

Location of a child part's joint is given by $l_j = (x_j, y_j, o_j, s_j)$, because of the way we parameterized locations of body parts (the anchor point coincides with the joint point). Similarly we need a location of parent part's joint $l_i^{(j)} = (x_i^{(j)}, y_i^{(j)}, o_i^{(j)}, s_i^{(j)}) = (x_i^{(j)}, y_i^{(j)}, o_i, s_i)$. Notice that we have included orientation and scale into the set of joint parameters. So the location of the joint fully specifies the location of the body part.

Now we write

$$\begin{aligned} p(L_j|L_i) &= p(L_j|L_i^{(j)}) \propto p(L_j - L_i^{(j)}) = p(X_j - X_i^{(j)}, Y_j - Y_i^{(j)}, O_j - O_i^{(j)}, S_j - S_i^{(j)}) \\ &= p(X_j - X_i^{(j)})p(Y_j - Y_i^{(j)})p(O_j - O_i^{(j)})p(S_j - S_i^{(j)}), \end{aligned} \tag{2.9}$$

assuming that horizontal, vertical, angular and scale distances are independent.

In this paragraph we quickly review distributional assumptions made in [8], which is a classical paper on pictorial structures. Distributions of the distances is a serious issue in the above paper. Horizontal, vertical and scale distances are assumed to be normally distributed. An angular distance is assumed to come from von Mises distribution [13]. It may be thought of as the circular version of the normal distribution, since it describes the distribution of a random variate with period 2π. Next, authors specify von Mises distribution over angular distances in terms of a Gaussian over two independent variables. Thus they finally obtain a multivariate normal distribution over five variables with a diagonal covariance matrix. This procedure is needed to represent relative distance (deformation cost) between locations of two body parts, $d_{ij}(l_i, l_j) = -\log p(l_j - l_i^{(j)})$ from equation (2.1), as a Mahalonobis distance between transformed locations. Then an efficient algorithm for computation of a distance transform of a function from [7] can be used. This in turn allows to solve the energy minimization problem (2.1) efficiently.

We observed that the normality assumption is superfluous in the case, where one considers to sample from posterior only. We can directly use, for example, von Mises distribution without any increase in running time complexity. Moreover, even non-parametric estimates, e.g. a histogram, can be used. There is, however, a dependency between the form of the distribution and efficiency. We will discuss this in more detail later. Finally, we note that restriction to sampling from posterior is not a serious limitation. Firstly, the MAP solution can be approximated by taking a configuration from the sample with the highest posterior probability. Secondly, the MAP solution is quite unreliable taking into account that our model is just a rough approximation of reality. It becomes even more unreliable, when one decides to use simplistic appearance models. Therefore, when concentrating on sampling from the posterior, we actually do not lose much. This, however, allows us to drop unnecessary assumptions and consequently simplify and make the sampling algorithm more efficient (it is not required any more to work in five-dimensional space as in [8]).

Some authors do not assume independence between distances. For example in [25] a distribution over deformation cost $d_{ij}(L_i, L_j)$ is learned using a SVM. This, however, implies a very serious slowdown of the inference algorithm.

Finally, let us specify a transformation $T_{ij}(l_i) := l_i^{(j)}$ from a location l_i of a body part v_i to a location $l_i^{(j)}$ of a joint of the part v_i corresponding to a body part v_j. For illustration see figure 2.6.

$$T_{ij}(l_i) = l_i^{(j)} = \begin{pmatrix} x_i^{(j)} \\ y_i^{(j)} \\ o_i^{(j)} \\ s_i^{(j)} \end{pmatrix} = \begin{pmatrix} \phi_{l_i}((x_{ij}, y_{ij}))_x \\ \phi_{l_i}((x_{ij}, y_{ij}))_y \\ o_i \\ s_i \end{pmatrix}, \tag{2.10}$$

where (x_{ij}, y_{ij}) is the joint point in part's coordinate system and ϕ_{l_i} is a transformation from part's coordinates to image coordinates given that the part's location is l_i (see equation (2.7)).

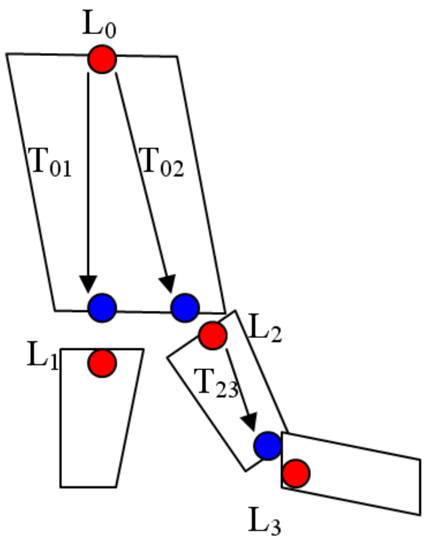

Figure 2.6: Illustration of transformations from body part locations to locations of joints.

2.4 Sampling from the Posterior

So far we have specified a pictorial structure model of a human pose in a probabilistic framework. Let as before $\mathcal{L}$ be a finite set of possible discrete locations for all body parts. This means that a configuration $\mathbf{L}$ of the human pose (a vector of the locations of all parts) takes its values in $\mathcal{L}^n$. Assume that for the given state space $\mathcal{L}^n$ we have obtained model parameters θ, which define distributions over relative joint locations and parts' appearance models. Suppose now that we are given an image I. More precisely I is a properly scaled region of interest of an input image. Now our pictorial structure model defines a decomposition (2.5) of the posterior probability $p(\mathbf{L}|I)$ of a configuration $\mathbf{L}$, given image I and model parameters θ (implicit). Our goal is to sample configurations of poses efficiently from the posterior. The sampling algorithm presented in this section uses dynamic programming techniques and can be seen as an extension of belief propagation algorithm for probabilistic inference.

2.4.1 Probabilistic Inference

Recall that our pictorial structure model is a Bayesian network with independence graph G. It is known that for arbitrary Bayesian network inference is an NP-complete problem [5]. However, when the independence graph G is a tree, polynomial time algorithms exist. One of the most popular such algorithms is belief propagation.

As we saw in section 2.2 the posterior probability of a human pose configuration is given by

$$p(\mathbf{L}|I) \propto p(L_0) \prod_{(v_i \to v_j) \in A} p(L_j|L_i) \prod_{v_i \in V} p(I_i|L_i),$$

where $v_0 \in V$ is the root of the tree. Let $Ch(i)$ be a set of children of vertex v_i. The algorithm works by first computing $p(L_0|I)$. Then we sample a location of the root from this distribution. Next, given $L_0 = l_0$ we sample a location l_c for each child $v_c \in Ch(0)$ of the root from $p(L_c|l_0, I)$. We continue in the same manner until we have sampled a location for each part.

The marginal distribution of the root location is

$$p(l_0|I) \propto \sum_{l_1 \in \mathcal{L}} \cdots \sum_{l_{n-1} \in \mathcal{L}} \left(p(l_0) \prod_{(v_i \to v_j) \in A} p(l_j|l_i) \prod_{v_i \in V} p(I_i|l_i) \right). \tag{2.11}$$

Computing distribution $p(l_0|I)$ in this way would take exponential time. But since graph G is a tree we can rewrite the above expression as follows.

$$p(l_0|I) \propto p(l_0)p(I_0|l_0) \prod_{v_c \in Ch(0)} S_c(l_0), \tag{2.12}$$

where $S_j(l_i)$ expresses an effect of marginalization over child body part v_j and its descendants. So that

$$S_j(l_i) \propto \sum_{l_j} \left(p(l_j|l_i)p(I_j|l_j) \prod_{v_c \in Ch(j)} S_c(l_j) \right) \qquad (v_i \to v_j) \in A. \tag{2.13}$$

In the terminology of belief propagation algorithm $S_j(l_i)$ is nothing else, but a diagnostic parameter or a message received by the node v_i from its child v_j. Similarly to equation (2.12), we have for $(v_i \to v_j) \in A$

$$p(l_j|l_i, I) = p(l_j|l_i, \mathbf{l}_{\text{cut(i,j)}}, I) \propto p(l_j|l_i)p(I_j|l_j) \prod_{v_c \in Ch(j)} S_c(l_j), \tag{2.14}$$

where cut(i,j) is a set of indices of those vertices, which are cut from v_j when v_i is observed. By a property of a Bayesian network this means that L_j is independent from $\{L_k | k \in \text{cut(i,j)}\}$ given L_i. This in turn means that we can put these variables in the conditioning set. Finally, we see that when marginalizing we need to sum over descendants of v_j only.

The recursive functions in displays (2.12),(2.13) and (2.14) define a dynamic programming algorithm for computing distribution functions $p(L_0|I)$ and $p(L_j|l_i, I)$, $(v_i \to v_j) \in A$ up to normalizing constants. First, we compute S functions for the leaves of the tree. Then we trace back the tree and compute S functions for the vertices, all children of which already have been processed. Finally, the posteriors can be computed in the trivial way. One can view this as inference in a tree-structured Bayesian network using belief propagation algorithm. Note that the trivial way to compute each S function has running time complexity $O(|\mathcal{L}|^2)$. We already know that $|\mathcal{L}|$ is huge in practice, therefore the trivial way is unpractical. In section 2.4.3 we will discuss an efficient algorithm, though approximate, to compute S functions in linear time.

2.4.2 Sampling

After S functions have been computed, we are ready to start sampling. Suppose we need a sample of poses of size N. After computing $p(L_0|I)$ we sample a root location N times. For each root location l_0 we compute distribution functions $p(L_j|l_0)$, $v_j \in Ch(0)$, sample locations of parts $v_j \in Ch(0)$ and continue with children of children and so forth until we get full pose configuration corresponding to l_0.

Computation of root distribution $p(L_0|I)$ takes linear time when S functions are precomputed. Sampling from $p(L_0|I)$ can be done in $O(|\mathcal{L}| + N \log |\mathcal{L}|)$ time. For completeness we give a succinct description of the algorithm to sample from multinomial distribution. First, we index locations of the root in some order. That is we introduce a bijective map $\psi : l_0 \to \{0, 1, \dots, |\mathcal{L}| - 1\}$. Next, we compute cumulative distribution function $F(\psi(l_0))$. Then we do N times the following. Generate a random $c \in [0, 1]$ from the uniform distribution defined on $[0, 1]$ and perform binary search of an index i satisfying $\begin{cases} F(i-1) < c \le F(i), & i > 0 \\ F(i) \le c, & i = 0 \end{cases}$. Finally, we set $l_0 = \psi^{-1}(i)$.

Usually conditional probabilities $p(L_j|l_i), (v_i \to v_j) \in A$ are non-zero only on some small subset $\mathcal{L}^{>0} \subset \mathcal{L}$, because joints of two connected parts v_i and v_j can't be far apart. Assume the size of such subset is bounded from above by h.

$$h = \max\{|\{l_j \in \mathcal{L}; p(l_j|l_i) > 0\}|; (v_i \to v_j) \in A, l_i \in \mathcal{L}\}$$

Note that $p(l_j|l_i) = 0$ implies $p(l_j|l_i, I) = 0$ and, therefore, when sampling l_j given l_i we only need to consider the subset $\mathcal{L}^{>0}$. Recall that $p(l_j|l_i) \propto p(x_j - x_i^{(j)})p(y_j - y_i^{(j)})p(o_j - o_i^{(j)})p(s_j - s_i^{(j)})$ (independence assumption for distances). Therefore, the subspace $\mathcal{L}^{>0}$ is a (hyper) rectangle and thus it can be easily determined knowing support (non-zero probability) intervals of distributions of distances. To be precise, we note that $\mathcal{L}^{>0}$ generally is a union of rectangles. But since these rectangles should be close together, we take $\mathcal{L}^{>0}$ to be a convex hull of these rectangles. For illustration see figure 2.7.

From the above discussion it is clear that full configuration sampling given precomputed S functions is $O(|\mathcal{L}| + N \log |\mathcal{L}| + Nnh)$ fast, where N is the sample size,

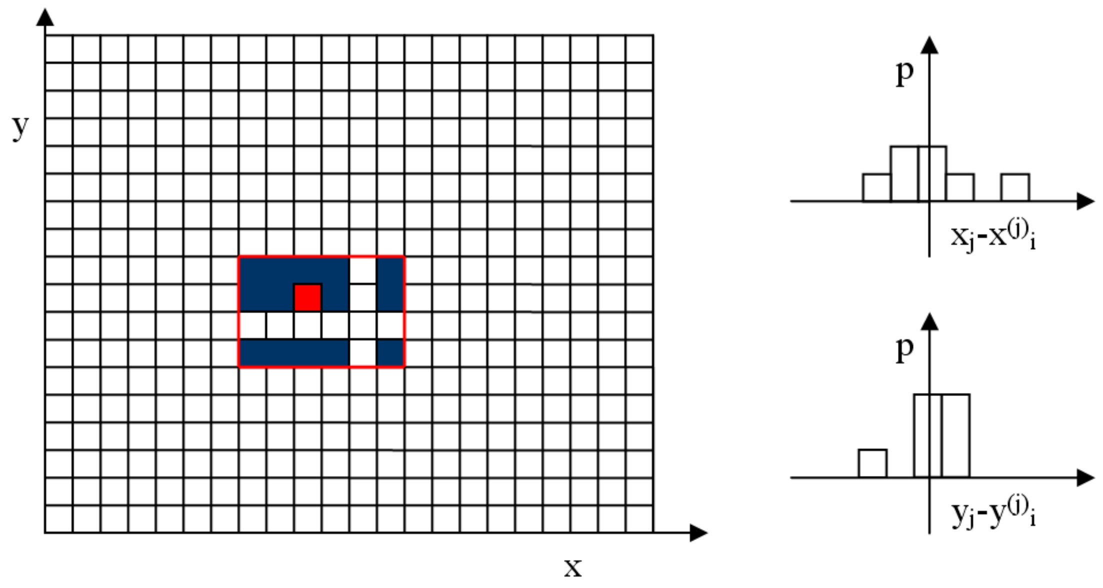

Figure 2.7: Hypothetic state space $\mathcal{L}$, location of parent's joint $l_i^{(j)}$ in red, support subspace $\mathcal{L}^{>0}$ denoted by red rectangle and corresponding distributions over distances.

n is the number of parts, $\mathcal{L}$ is the state space (same for every part) and h is the upper bound on the size of non-zero probability sets. In practice Nnh term dominates the others. Thus peaked distributions of distances make sampling faster, especially when N is large. On the other hand the model becomes more rigid and may not fit well to some image. Finally, we note that our choice of the origins of local coordinate systems for every part is very beneficial here. Recall that it was placed in a joint corresponding to part's parent. If this was not the case $(l_j^{(i)} \neq l_j, (v_i \rightarrow v_j) \in A)$ we would have two options. One option is to work with expanded non-zero probability region (maximal distance between joints is much smaller than between parent part's joint and child part's centroid for example). Another option is to define non-zero probability space for part's joint, then go through it and apply reverse transform T_{ji}^{-1} to each joint location $l_j^{(i)}$ in order to get l_j. Neither of these approaches is as simple and efficient as our approach.

2.4.3 Efficient Computation of S Functions

We start from rewriting equations (2.12), (2.13), (2.14) in convenient form. Notice that as soon as we fix or sample location l_i it is not convenient to work with this parametrization of the part any more. We rather need locations of joints corresponding to children, than the location of joint corresponding to part's parent. So we write using equation (2.9)

$$p(l_0|I) \propto p(l_0)p(I_0|l_0) \prod_{v_c \in Ch(0)} S_{0c}(l_0^{(c)}) \tag{2.15}$$

$$p(l_j|l_i^{(j)}, I) \propto p(l_j - l_i^{(j)})p(I_j|l_j) \prod_{v_c \in Ch(j)} S_{jc}(l_j^{(c)}) \qquad (v_i \to v_j) \in A \tag{2.16}$$

$$S_{ij}(l_i^{(j)}) \propto \sum_{l_j} \left(p(l_j - l_i^{(j)})p(I_j|l_j) \prod_{v_c \in Ch(j)} S_{jc}(l_j^{(c)}) \right) \qquad (v_i \to v_j) \in A, \tag{2.17}$$

where S_{ij} is a function of a location of part v_i's joint corresponding to child part v_j.

Now suppose that we want to compute function S_{ij}. This means that functions $S_{jc}, c \in Ch(j)$ have already been computed. Denote

$$\alpha(l_j) := p(I_j|l_j) \prod_{v_c \in Ch(j)} S_{jc}(l_j^{(c)}), \tag{2.18}$$

then

$$S_{ij}(l_i^{(j)}) \propto \sum_{l_j} \Big(\alpha(l_j)p(l_j - l_i^{(j)}) \Big) = (\alpha \oplus p)(l_i^{(j)}), \tag{2.19}$$

where $\oplus$ denotes discrete convolution. We see that computation of function S_{ij} can be split in two steps. First, we compute $\alpha(l_j)$ for every $l_j \in \mathcal{L}$ and then we compute discrete convolution of α and p.

Let us consider now the discrete convolution problem. The assumption of independence of distances (2.9) comes in handy again.

$$S_{ij}(l_i^{(j)}) \propto$$
$$\sum_{x_j} p(x_j - x_i^{(j)}) \left(\sum_{y_j} (p(y_j - y_i^{(j)}) \left(\sum_{o_j} (p(o_j - o_i^{(j)}) \left(\sum_{s_j} p(s_j - s_i^{(j)}) \alpha((x_j, y_j, o_j, s_j)) \right) \right) \right) \right) \tag{2.20}$$

We see that original convolution in four-dimensional space can be obtained by a repeated one-dimensional convolution.

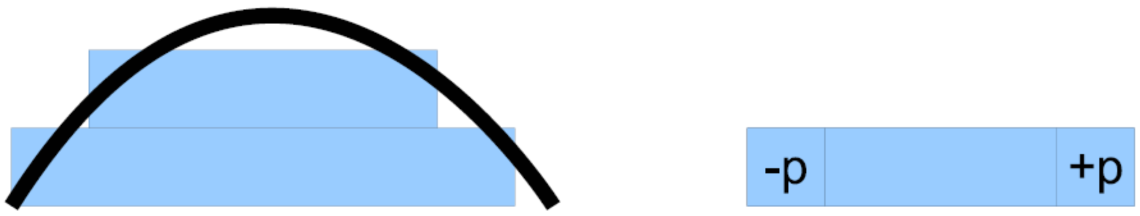

Figure 2.8: Approximation of a distribution by a pyramid of uniform distributions; efficient computation of moving average.

Fast Convolution

Suppose that we are given two functions α and p, then the discrete convolution is defined as follows

$$\alpha \oplus p(n) = \sum_m \alpha(m)p(m-n). \tag{2.21}$$

Assume for the moment that p is a Gaussian distribution function. Then the convolution is just a Gaussian blur (filter), which is very popular in image processing and computer vision domains. Linear time approximation algorithms exist for computation of a filtered image [27].

The idea is to approximate a Gaussian by a pyramid of uniform distributions. Convolution with a uniform distribution (moving average) can be done very efficiently. Given the result at the previous step we only need to add one term and to subtract one term. For illustration see 2.8.

This is a general idea and will work with other distributions as well. For example, convolution with a Von Mises distribution should be of similar complexity as Gaussian convolution. Note that running time complexity depends only on the number of levels in a pyramid, not on the length of those. On one hand fast convolution presented here is only an approximate technique. We can control running time/accuracy by the choice of the appropriate number of levels in the pyramid. On the other hand our distributions of distances are learned from data or fixed by hand. We have a very limited number of training examples and thus it is unnecessary to approximate distributions very accurately.

Finally, if our distance distributions are peaked (the non-zero probability interval is short), then a straightforward algorithm for computation of the S functions will be quite efficient. The straightforward algorithm should also be preferred if we define distributions as histograms. In that case the assumption about independence of distances is not required any more.

2.4.4 On the Importance of Parameterization of Parts

As we have already seen before, parameterization of body parts by the location of the joint corresponding to part's parent has quite a few advantages. We discuss this issue in more details in the following.

- A smaller state space $\mathcal{L}$ of possible locations of parts is required. In our case we only require that joints of a person in an image are inside chosen state space. For the common parameterization when we describe position of parts by the location of centroids, the state space should be extended to include those centroids, if we hope to find the correct pose. For illustration see figure 2.9. The bounding box defining the state space in practice should be obtained automatically. For example we may use an output (an ellipse denoting current location of a person) from a tracking algorithm, a region from a detector or a blob of foreground pixels obtained by background subtraction. These regions may not include the whole person and thus should be extended in some way. We see that this extension of the bounding box may be done to a smaller degree for our parameterization.

- The sampling algorithm becomes more efficient, easier to understand and implement. This is a consequence of a few factors.
  - The reverse transform $T_{ij}^{-1}(l_i^{(j)}) = l_i$ is not needed anymore.
  - Given joint location of the parent, we can easily find non-zero probability subspace $\mathcal{L}^{>0}$ for the body part location (which is just a joint location, corresponding to the parent).
  - Computation of the $p(l_j|l_i)$ and S functions becomes more transparent and require a smaller number of transformations.

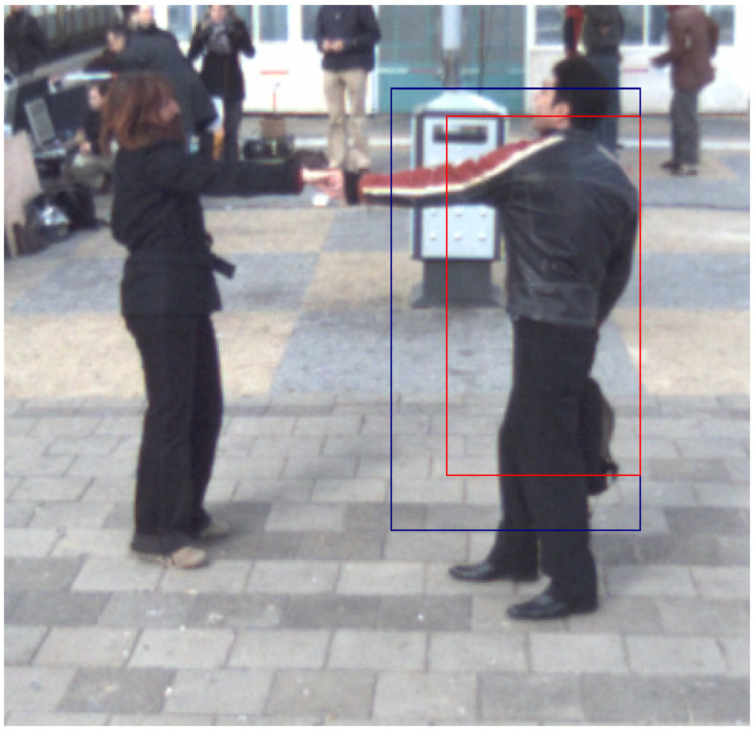

Figure 2.9: The minimal bounding boxes required to find correct pose for our parameterization (red) and common one (blue).

For completeness we would like to mention here another source of reduction of a state space. As we have seen before, the assumption that relative distances must be Gaussian can be relaxed. Thus, for example, we can directly work with Von Mises distribution over joint angles without need to go to five dimensional space.

Chapter 3

Modeling Appearance of Parts

In this chapter we propose a few approaches to model appearance of body parts. Our goal according to the probabilistic framework is to model the image probability $p(I_i|l_i)$ given a location of a body part i. Obviously, this task links well with the task of detecting individual body parts. A lot of research have been done in this domain, and several methods for body part detection have been proposed.

Usually one goes through all possible object locations and for each location decides if the object is present in an image at the given location. In order to make this process faster some authors use so called zooming techniques. First, the image is parsed on a coarse scale to detect regions of interest. Then the scale is gradually refined and only the regions of interest are considered. For an example see [12], where authors simultaneously use coarse-to-fine approach over the shape hierarchy and over the transformation parameters. Our task, however, as it was defined, assumes that the object location is given. We will return to image scanning issue later, when discussing optimization possibilities.

Most of the approaches to body part detection can be split in two steps. First, we apply some filter or a cascade of filters to an image to prune away irrelevant information and obtain a succinct discriminative image representation. This representation can be viewed as a set of cues. Next, we feed these cues into some classifier, which decides if given image region contains an object or not. For example, in [17] authors apply different filters to compute some orientation features of an image region. Then they group them and feed into cascade of weak classifiers. Another approach is presented in [24], where probabilistic region templates are used as object models. First, the dissimilarity score between the appearance of foreground and background of a transformed probabilistic region is obtained (this too can be seen as a filter response). Then distribution over this score is learned from data.

As we have noted several times before, the structure model does not put any restrictions on the appearance model. Moreover, the appearance model of parts can be fairly generic. This is because we do not attempt to find body parts separately, but we match the whole human model at once assuming that the actual configuration of body parts is discriminative. Having efficiency issue in mind, we restrict our attention to simple models only. For a given location of a body part we compute some score s and set $p(I_i|l_i) \propto p(s)$, where distribution over s is to be learned from training data.

In the following sections we define a few different scoring methods. We also present a model of a body part with variable width and efficient scoring algorithm for this model. Finally, some optimization remarks are given.

3.1 Edge Map and Chamfer Distance

Matching geometric primitives to an edge map of an image is a popular method for object detection. The edge map is commonly obtained using Canny edge detector [4]. For illustration see figure 3.1. A common measure of a shape match to an edge map is a chamfer distance [3]. There are many successful applications in computer vision domain, which rely on a chamfer distance as a dissimilarity measure between a shape template and an image. One of the examples is [12], where a method for efficient shape-based object detection based on distance transforms is presented.

The distance transform is an operator normally only applied to binary images (edge maps). The result of the transform is a gray-level image that looks similar to the input image except that the gray-level intensity of points are changed to show the distance to the closest edge from each point. Of particular interest is a chamfer transform, which approximates global distances in the image by propagating local distances in a raster scan fashion. Local distances are given by a mask, e.g. 3×3

mask may look as follows $\begin{array}{|c|c|c|}\hline 3 & 2 & 3 \\ \hline 2 & 0 & 2 \\ \hline 3 & 2 & 3 \\ \hline \end{array}$. It should be noted that the chamfer distance transform can be computed fast (in $O(N)$ time, where N is the number of pixels in the image). The algorithm performs a series of local operations while scanning the image twice. See figure 3.1 for an example of a distance transform image.

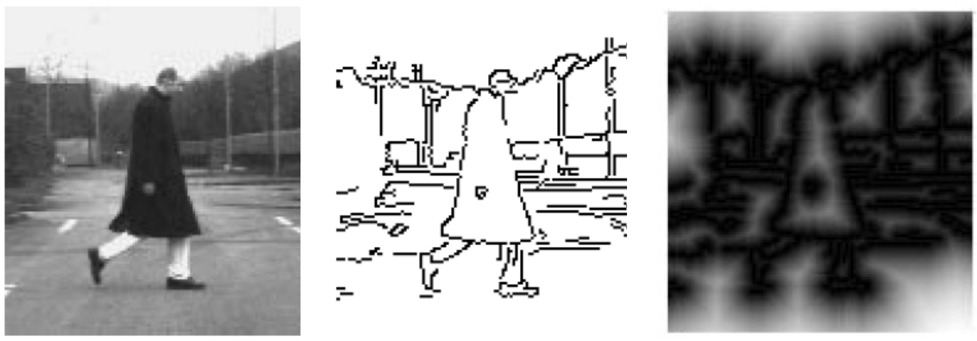

Figure 3.1: Original image, edge image, distance transform image. Adapted from [12].

Once a chamfer transform image has been computed, we can calculate chamfer distance of a shape matched to the image at some location in a very simple and efficient way. We only need to superimpose the shape template over the image (the shape template of a body part is positioned, rotated and scaled according to the given location of the part) and to sum distance values under the template pixels. Next we divide this sum by a number of template points, obtaining the average chamfer distance over all template points. To put this mathematically, suppose that T is a set of template points, $d_I(t)$ is a chamfer distance between template point t and image I (approximate distance from t to the closest image edge). Then the average chamfer distance is given by

$$d(I,l) = d(I,T) = \frac{1}{|T|}\sum_{t\in T} d_I(t), \tag{3.1}$$

where set T is determined by the shape and the location l of a body part. Note that $d(I,l)$ is nonnegative. Small values of distance $d(I,l)$ indicate good match of the template to the image.

Distance transform is calculated only once and is actually the input image. Whenever we need to compute the image probability given a location of a body part $p(I_i|l_i)$, we transform and position the shape of the body part in the image plane according to the location l_i (see section 2.3.1) and calculate dissimilarity score $d(I_i, l_i)$ according to equation (3.1). Next, we obtain the image probability according to equation (2.8), where distribution of score is to be specified or learned from data.

The advantage of matching a template to a distance transform image rather than to the edge image is that the resulting dissimilarity measure will be smoother as a function of location parameters. This enables us to use fairly coarse discretization and fairly generic models of body parts (rectangles and trapezoids). To see this imagine that (1) the shape positioned at l fits the image perfectly, though our chosen discretization allows for only the neighbor values of l and (2) the template rectangle is slightly wider than the rectangle in the image representing a body part. It is easy to see that matching to the distance transform image would give low dissimilarity score, but matching to the edge image would give high dissimilarity in the presented examples.

There is, however, a problem. Consider two cases. In the first case we try to match a vertical line to the image containing similar vertical line, but shifted slightly aside. In the second case we match our vertical line to a clutter consisting of horizontal and diagonal lines. The chamfer distance, as it was defined above, may even be larger in the first case and thus cluttered image would be preferred. To solve this problem we use the same idea as in [12]. Assume we have already discretized orientation component of a part's location vector. For each orientation (we do not distinguish between α and $\pi + \alpha$ here) we construct a separate edge map by selecting only those pixels from the full edge map, which are properly oriented. Orientation of a pixel is obtained from a gradient image, by taking the orientation of a perpendicular to the corresponding gradient. Thus for a high contrast line in the image we should ideally get this line in the edge map, associated with proper orientation. This line, however, should not appear in an edge map with quite different orientation. Obviously, we allow some degree of freedom and select also pixels with orientations slightly different from the edge map's orientation. See figure 3.2 for illustration.

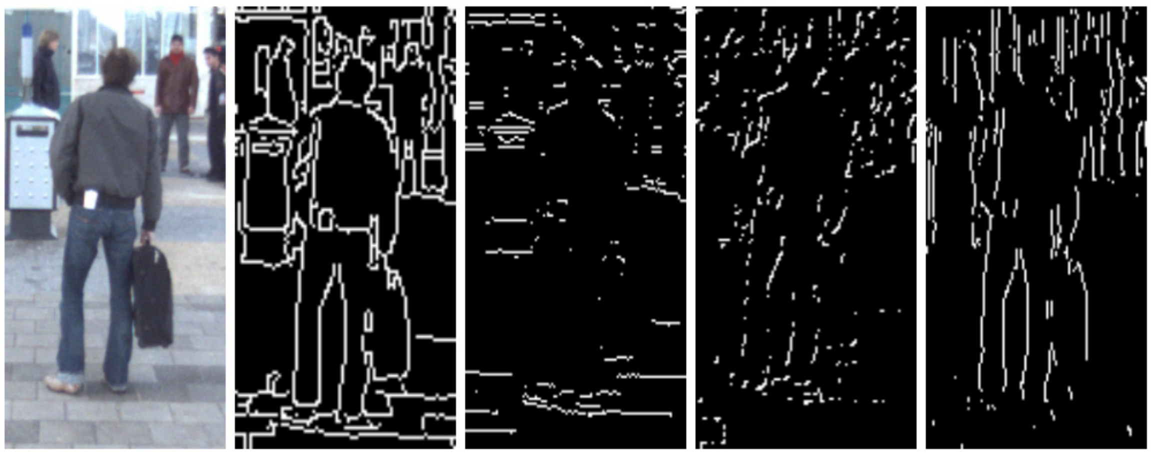

Figure 3.2: Image, edge map and some examples of oriented edge maps.

Before running the sampling algorithm we construct oriented edge maps and apply chamfer transform to each of them. We use only long sides of rectangles representing human limbs, when computing chamfer distance. For these lines we chose the edge map associated with the orientation of the limb. Here we assume that trapezoids, modeling upper limbs are long enough, so that orientation of its sides is quite similar to limb's orientation. Finally, we note that for head and torso we use all four lines. In this case two different oriented edge maps are to be selected.

3.2 Color Dissimilarity Between Foreground and Background

After performing several experiments with chamfer distance based score, we found out that the algorithm is quite strongly distracted by cluttered background, even if we use oriented edge maps (see figure 5.3 (a)). We noticed that sometimes it is hard to guess the correct pose of a human if we only see an edge map. Making thresholds higher results in fewer edges in the background, however, we simultaneously loose some important human body contour segments. At the same time people in the color images are well distinguishable. Therefore, we decided to experiment with a description that takes account of color or texture.

One might envisage learning a model that describes the wide variation of appearances of differently clothed people. This, however, would require a very complex model with many parameters, which in turn needs a prohibitively large amount of training data. In our case, when we have just a few people present in the training data, learning such a model would be prone to overfitting errors. Instead we will exploit dissimilarity between the appearances of human body parts and background regions of these parts. These appearances would be dissimilar unless the part is completely camouflaged. The approach is not new and can be considered as a variation of the one used in [24].

For every body part we define a background region, which covers a limited area around the part. Similarly to chamfer distance case, we only consider background regions along long sides of a rectangle modeling the body part. This seems to be a good choice for limbs. For simplicity we define background regions for all body parts in the same way. For illustration see figure 3.3. The background regions of parts

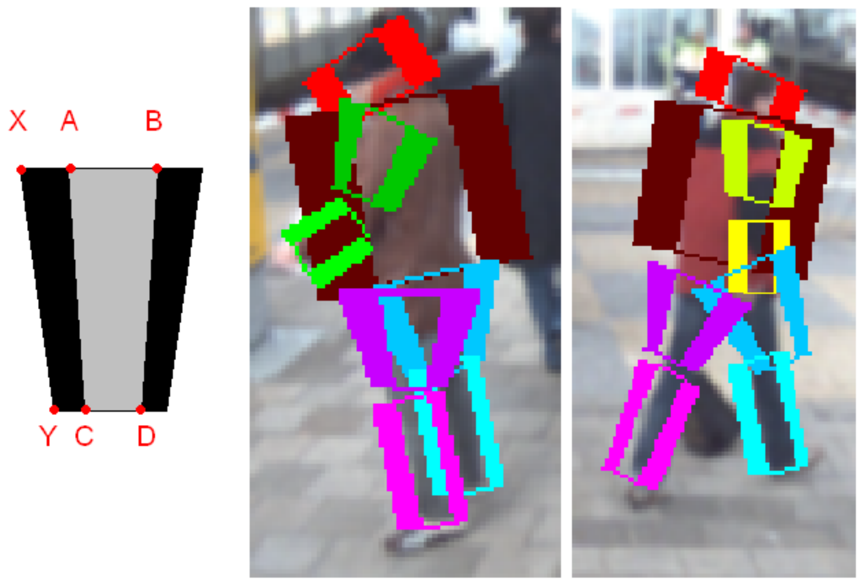

Figure 3.3: A limb in gray and its background region in black; examples of parts drawn according to ground truth and their background regions.

are obtained in the following way. Suppose A, B, D, C are vertices of a trapezoid modeling the part. We assume that pairs of points A, B and C, D define two parallel sides of the trapezoid. We put $X = A+(A-B)/2$ and similarly $Y = C+(C-D)/2$. The polygon with vertices X, A, C, Y constitutes one part of the background region. The symmetrical part is constructed in the same way. Finally, we note that the foreground and the background regions are of the similar size.

So far we have defined foreground and background regions of a part. There is still a question left. How do we find all pixels belonging to a region? First, note that we can restrict our attention to polygons only (we consider each polygon forming the background region separately). Next, we determine a bounding box of a polygon and test each point inside this bounding box if it is also inside the polygon. This is a well-known point-in-polygon (PIP) problem in computational geometry. We use ray casting algorithm to solve it efficiently [14].

Next step is to specify appearance dissimilarity measure. We restrict our attention to color only. Two histograms of color values are computed; one for foreground region, another for background region. We use k bins for each color component (R,G,B), so there are k^3 bins in each histogram. Now pairwise appearance dissimilarity can be viewed as a dissimilarity of two histograms. A popular measure of similarity between two discrete distributions is the Bhattacharyya coefficient [2]. Consider two histograms $p = \{p^{(u)}\}_{u=1\ldots m}$ and $q = \{q^{(u)}\}_{u=1\ldots m}$, then the coefficient is defined as

$$\rho(p, q) = \sum_{u=1}^{m} \sqrt{p^{(u)} q^{(u)}}. \tag{3.2}$$

As dissimilarity score between two color histograms p and q we use

$$\sigma(p, q) = 1 - \rho(p, q). \tag{3.3}$$

This score takes values between 0 and 1. The bigger is σ, the more dissimilar the distributions are. It attains 1 when common support set of these distributions is empty, which indicates that background and foreground appearances are very dissimilar. Later, we will use $\sigma(p, q)$ as a basis for appearance model.

Finally, we note that there are few issues to consider when computing color dissimilarity score. First, we can use different resolutions of an input image, which determines number of data points for which histograms are constructed. This in turn may influence our choice of the number of bins in the histogram. Note that high resolution and large number of bins make computation of color dissimilarity slow. Secondly, it seems to be beneficial to use Gaussian smoothing of the input image.

3.3 Background Subtraction Based Score

Whenever possible background subtraction provides a powerful cue to object detection. The intended use of our algorithm is to recognize people in images taken by a static camera. This makes it feasible to employ background subtraction. We use a very simple background subtraction algorithm. We pick a background image, which does not contain people in the foreground plane (area close to camera). There are, however, people present in the background. Now for every input image the corresponding background mask is constructed as follows. First, size of the background image is adjusted to the size of the input image. Then we go through all pixels in the input image. For each pixel we compute a distances between it's color and color of corresponding background pixel. We choose the distance to be a maximum of absolute differences over color channels: $\mathrm{dist}([R_I G_I B_I], [R_B G_B B_B]) = \max\{|R_I - R_B|, |G_I - G_B|, |B_I - B_B|\}$, where $[R_I G_I B_I]$ represents color of a pixel from the input image and $[R_B G_B B_B]$ represents color of the corresponding pixel in the background image. Finally, we assign the pixel to the background if computed distance is less than some threshold. For an example of background masks consult 3.4.

There are a few options for defining image probability $p(I_i|l_i)$ when using a background mask. The simplest way is to compute proportion of background pixels s inside the rectangle modeling body part v_i. Assuming that we know a distribution over this dissimilarity score s, we put $p(I_i|l_i) \propto p(s)$. The above defined model has a serious drawback: a body part will match to any sufficiently large foreground region equally well. For example, torso may be represented with a big blob of foreground pixels, then any limb having any orientation will match well to this blob. This light model based on background mask, however, may be used in combination

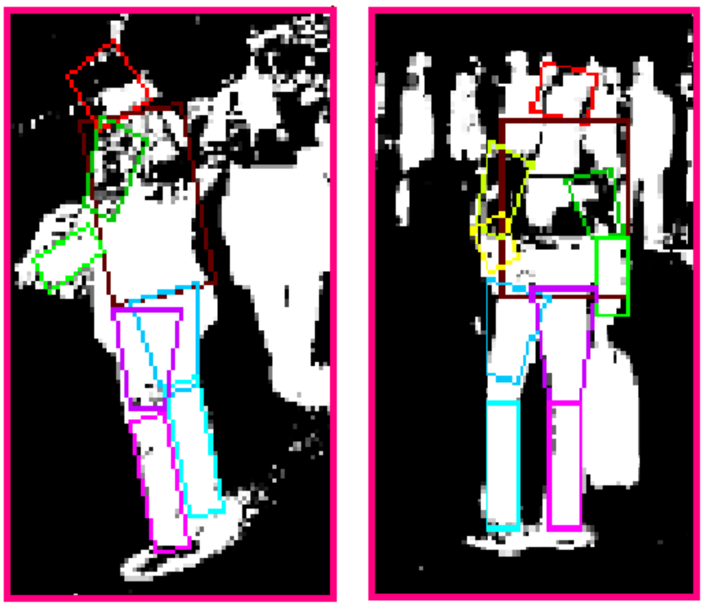

Figure 3.4: Background masks and configurations of parts specified by the ground truth.

with some other appearance model to prune away high responses of matches against background clutter.

The idea from the previous section, where we compute dissimilarity between part's appearance and appearance of its background, is suitable in the current setting as well. An area around a part should contain little number of foreground pixels in the ideal case and thus should be dissimilar from the inside region. We specify the background region of the part in the same way as in the previous section. There is, however, a difference from appearance dissimilarity case. Now we know how appearances (expressed by the number of foreground pixels) of part and its background region should look like. Thus we define the image probability in a different way. Let us denote the number of foreground pixels belonging to the part (the background region) by $N_0^F(N_1^F)$. We use $N_0(N_1)$ to denote the area of the part (the background region). These areas are constants for a fixed pictorial structure model. Similarly to [8] we put

$$p(I_i|l_i) = f(N_0^F, N_1^F) = q_0^{N_0^F}(1-q_0)^{(N_0-N_0^F)} q_1^{N_1^F}(1-q_1)^{(N_1-N_1^F)}. \qquad (3.4)$$

The above expression can be viewed in the following way. Consider pixels in the region of interest as independent random variables taking values in the set $\{\text{foreground}, \text{background}\}$. For each of these pixels we know if it belongs to the part or to the part's background region. Finally, we assume that each pixel is Bernoulli distributed with parameter q_0 for inside pixels and q_1 for outside pixels. The appearance parameters q_0 and q_1 are to be learned from training data.

3.4 Combination of Appearance Models

We have defined a few appearance models. Every model has its strong and weak sides. It is, therefore, beneficial to combine these models. Recall that appearance models are based on some kind of score. For the ease of combination we assume that these scores are independent random variables. Similarly to equation (2.8) we get

$$p(I_i|l_i) \propto p(s_i^1, \ldots, s_i^R) = \prod_{r=1}^{R} p(s_i^r), \tag{3.5}$$

where R is the number of combined appearance models and s_i^r is an image score given part's location and corresponding to model r. The factorial model of (3.5) is also grounded on our belief that an image of a well standing out body part should have a high probability under every model. For example, if for some image and a given part's location the probability of a corresponding chamfer distance is high, but the probability of a background subtraction based score is small, then the image probability should be small too. It is likely that we are considering a match against background clutter. We will test different combinations of appearance models in the experiments chapter.

3.5 Optimization Issues

Efficiency of obtaining image probability given a location of a body part is crucial for our algorithm. Recall that state space $\mathcal{L}$ of possible locations of parts is huge. It can be seen from equation (2.17) that computation of each S-function requires to compute image probability at every possible location of a child part. According to our algorithm we compute function $\alpha(l_j)$ for each $l_j \in \mathcal{L}$, see equation (2.18). Note that there are $n-1$ S-functions each of which corresponds to some arc in the tree used as Bayesian network structure. Moreover, computation of posterior distributions for each body part as given by formulas (2.16) and (2.17) requires repeated computation of image probability.

It is very beneficial to precompute image probability for every part and every possible location of it. Moreover, when this job is done in batch further optimization is possible. Notice that here we return to image scanning issue touched upon in the beginning of this chapter.

Computation of image probability using any of the appearance models discussed above needs some information about pixels in a region of interest. Consider chamfer distance based model. In this case we need a chamfer distance value for each pixel of the lines belonging to the trapezoid modeling a part. In the case of appearance dissimilarity based model we need the color value for each pixel from background and foreground regions of the part. It is not very trivial to compute these sets of pixels. Computation involves repeated application of transformation from part's local coordinate system to image coordinate system, line drawing and ray casting algorithms. Notice that we can compute these sets of pixels of interest just once for every orientation and scale of the part v_i (put $x_i = 0$ and $y_i = 0$). For any other location $l_i = (x_i, y_i, o_i, s_i) \in \mathcal{L}$ with the same orientation and scale we only need to shift all the pixels by $(\lambda x_i, \lambda y_i)$, where λ is discretization step.

Another source of optimization is to represent a few body parts by the same appearance model. For example, we may use the same rectangle for both lower legs, or we even can use the same rectangle for all limbs. This approach saves a lot of time when we have a few pictorial structure models. We give two examples when one might consider to use several pictorial structures. First, we may have different pictorial structure models for people seen in different perspectives. Note that torso width may vary a lot depending on the camera angle. Secondly, we may try to handle occlusions by sampling not only from a full model, but also from a set of partial human body models. In both of presented cases it is computationally beneficial to use the same appearance model for similar parts in all pictorial structures.

It is also possible to speed up precomputation of image probabilities by using zooming techniques mentioned at the beginning of the chapter. It is important to note here that we should assign zero probabilities to locations of $\mathcal{L}$ with care. It might happen that some part is occluded in an image or it is almost indistinguishable, but all other parts match extremely well to their correct locations. In this case we may reject the full configuration, because one part had zero contribution to the total product.

3.6 Variable Width Templates

The main disadvantage of all appearance models discussed above is their rigidity. Even if we consider just one person, his/her torso width may change a lot if we see him/her from a different view angle. Our model, however, should fit to any person in a variety of poses and views. Additionally people can wear loosely fitting clothes making their appearance even more different.

The above issues suggest that our rigid template usually would not get score close to ideal, when matched against real images. This would make posterior density flat, which in turn requires dense sampling. The latter implies inefficiency and hard final pose selection problem.

In our opinion, non rigid templates may improve considerably compared to rigid templates. One way to relax rigidity is to introduce another location component, responsible for changes in width of a part. This, however, would make the state space even larger, making our sampling algorithm inefficient.

There is another, efficient way to tackle this problem. Note that limb's width does not have any influence on the positions of limb's joints. This question is more involved if we consider torso. As we have noted above, torso appearance varies a lot depending on a view. The problem is not only with shape. Positions of joints in the local coordinate system of torso change dramatically. If we consider a person sideways then his shoulder joint is in the middle of torso. On the other hand, a person with frontal view has shoulder joint very far away from the middle(symmetry) line. This implies that distributions specified on differences between locations of shoulder joints should be very relaxed. Thus we may vary torso width without changing positions of shoulder and hip joints in torso coordinate system.

The above observations allow us to consider part's width in isolation from other parts. We introduce variable width template and later discuss a method for efficient image probability computation. We begin by noticing that the appearance model based on chamfer distance is much lighter than the one based on color dissimilarity. Moreover, it has comparable performance as we will see in the experiments section. Thus we define variable width model as an extension of the model based on chamfer distance.

For each template we define two sets of points L and R. Points in the set L (R) are located to the left (right) of symmetry line. For illustration see figure 3.5. Next, we specify two sets of line segments L' and R'. These are line segments centered at the points from corresponding sets L and R and having orientations within some δ from the part's orientation. Length of every line segment is equal to the length of the template. When template is matched against an image we only consider one line from the set L' and one line from the set R', which have the lowest chamfer distance:

$$d(I,T) = (min\{d(I,l)|l \in L'\} + min\{d(I,r)|r \in R'\})/2, \tag{3.6}$$

where template T and lines in $L' \bigcup R'$ are transformed according to location of the part in image coordinate system. In case the part under consideration is torso we may include chamfer distances of horizontal line segment into above formula. This line segments are fixed, thus we may get four disconnected lines instead of a polygon. This, however, does not have a significant influence on the final chamfer distance.

Note that torso template minimizing chamfer distance may be shifted to one side of the former symmetry line. This in our opinion is not a shortcoming. When observed from a side, human body is not symmetric. For a plump person it might be a good idea to shift the torso template in the direction, where person is looking. It is also possible to perform minimization in some constrained way. For example, we may require that chosen line segments are centered at the pair of symmetric points.

Variable width template placed at any image location will get a better score comparing to the rigid template. It, however, should discriminate better between

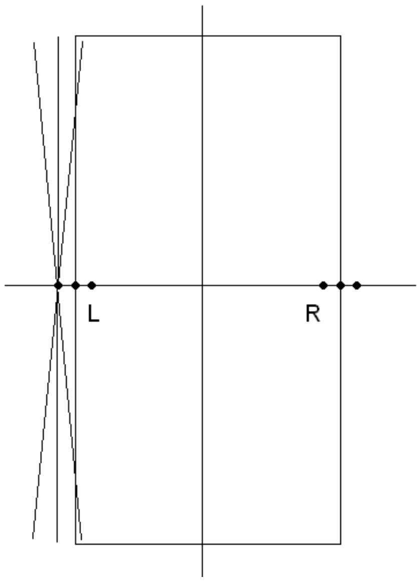

Figure 3.5: Variable width template.

body parts and background clutter. Variable width template will adjust itself to a part present in an image. Another advantage is as follows. In order to make sampling faster we may use coarse location discretization. At the same time the variable width template may be used to correct appearance score.

An important question now is how to obtain a chamfer distance of a variable width template to an image efficiently. This can be done as follows. First, we determine all possible lengths of line segments needed. Here we consider all body parts together with all scaling factors. Next for every orientation and every line segment length we do the following. For every pixel in the image we obtain a chamfer distance of the line segment with proper orientation and length centered at this pixel. This can be seen as a convolution of chamfer image with uniform distribution (moving average). We have already seen that uniform convolution can be obtained efficiently. Finally, when computing chamfer distance of the template, we only need to minimize over two small sets. Here we used precomputation strategy. One particular line segment is present in many templates, but its chamfer distance is computed only once.

The above appearance model may be combined with background subtraction based model. In this case we propose to use the light version of background subtraction based model, where we only compute average number of foreground pixels inside the former rigid template corresponding to a part. This is done to prune away high responses from background clutter.

Chapter 4

Learning Model Parameters

Performance of our sampling algorithm is very much dependent on a chosen parameterization of our model. The model should be tuned and adjusted to training data. There are a few important issues to consider here. The first is acquisition of training data. It is essential to have a representative sample for learning model parameters. Secondly, the ground truth labeling of parts has to be obtained. Finally, learning of model parameters itself should be designed and implemented.

4.1 Data Acquisition

Our human pose recognition algorithm was meant to be a part of an aggression detection system. Training data from a set of fixed cameras at a train station was collected within CASSANDRA project. This data consists of a number of video clips showing a few professional actors in different activities: standing, walking, running, carrying bags, using ticket machine, hugging, fighting, etc. The background is usually cluttered. There are many people present in the background. Sometimes trains pass by. Clips were taken under different lightning conditions. We stress here that we have a realistic setup. A sample video frame is depicted in figure 4.1.

For our purposes we selected clips from one particular camera. Then we randomly selected 200 frames from these clips.

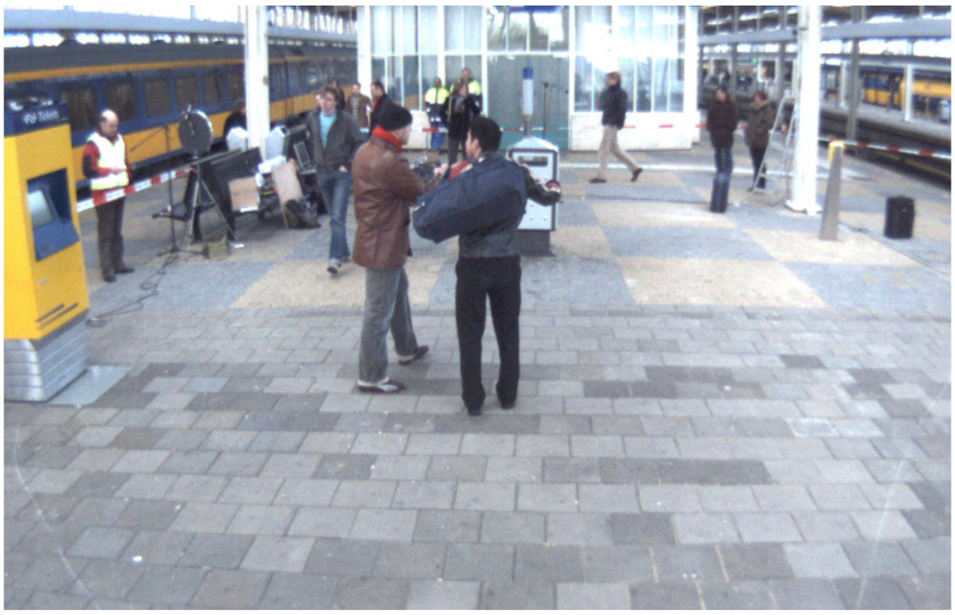

Figure 4.1: Sample input image from CASSANDRA test set.

4.2 Data Labeling

Data labeling can be done relatively fast if we need only to specify some points of interest in an image. At each step we consider just one person in the image. We choose to use the following points of interest: top-left and bottom-right corners of a bounding box, head top, neck, navel, shoulders, elbows, palms, hips, knees and feet. In addition we also require to specify a facing direction of the person.

For the labeling purpose we have developed a simple tool. Using this tool we load an image and select our points of interest in some predefined order by clicking on them. Then we choose a facing direction of a person and save all this information. Some parts may be occluded. In this case one or more points of interest corresponding to this part are occluded too. Thus during the labeling process it is enough to specify that a point is occluded. Later occlusion of parts may be inferred from this information. For an illustration of image labeling see figure 4.2.

A bounding box specified at labeling stage is a temporary measure taken to simplify matters. The bounding box is used to restrict the set $\mathcal{L}$ of possible locations of body parts. Moreover, we infer the proper scale of a person in an image from the height of the bounding box. Later we will describe an algorithm for automatic

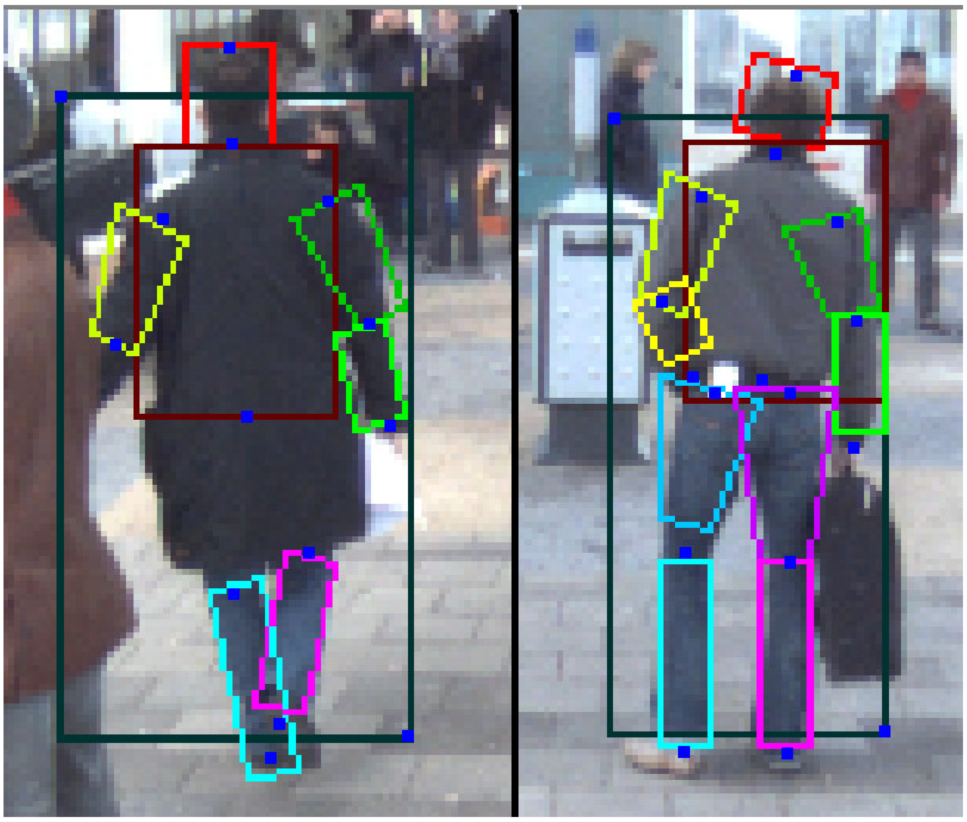

Figure 4.2: Two sample images with pose specifying points (in blue) and fitted polygons modeling body parts.

detection of the bounding box and scale of a person.

Raw information containing coordinates of points specifying a pose need to be converted into location vector $\mathbf{l} = (l_0, \ldots, l_{n-1})$. Note that the resulting location vector depends on chosen discretization and trapezoids modeling parts (we can not determine the degree of foreshortening of a limb in an image without knowing the length of the corresponding trapezoid). Appearance models of parts as well as distributions over relative joint locations depend on these trapezoids too. In order to simplify the learning procedure, we do not include the trapezoids modeling parts in the pictorial structure parameter set θ, but assume that they are fixed beforehand. First, we use a heuristic to adjust the trapezoids to data. Only after part models are fixed we consider learning of model parameters θ.

4.2.1 Adjusting Models of Parts to Training Data

In the chapter on appearance models we noted that it is very beneficial from computational point of view to use the same model for a few parts, e.g. right and left lower legs. In the following we consider the whole class of parts having the same model as a single entity. For every part (class of parts having the same model) we specify a trapezoid modeling it by hand. Next we compute the lengths of the part at every training image, where the part is not occluded. Note that our chosen set of interest points allows us to do so. For example, to compute the length of a lower leg we need to find the distance between corresponding knee and foot points. Torso length is given by the distance between the neck and the navel. From the given set of lengths we need to obtain the length of the model of the part. Note that in some images parts are foreshortened. Probably the best way to tackle the problem is to use clustering algorithm, where the number of clusters is taken the same as the number of possible scales. Next one can define the model length and a set of possible model scales based on means of the clusters. We, however, take a simpler approach. First, we compute the average a over all lengths. Then we choose only those lengths, which are bigger than λa for some chosen constant λ. We assume that others are foreshortened. Finally, we make the part's model length equal to the average taken over the selected set of lengths.

After a model of a part has been fixed, we can obtain part's scale at a particular image. We take the scale that minimizes the difference between the length of the foreshortened model and the length of the corresponding part in the image. Orientation and position of a part are obtained in a trivial way given discretization of parameter space. In the above way we obtain locations of all non-occluded parts in all training images. See figure 4.2 for an example of a body pose specified by the ground truth.

It is also important to define positions of joints in local part's coordinate system properly. Proper positions would result in more peaked distributions over relative joint locations making our algorithm both more efficient and more accurate. For upper arms and legs it is wise to assume that a joint connecting a part to its child part is located in between two lower trapezoid vertices. Thus it should ideally coincide with elbow or knee point specified during labeling phase. Positions of torso joints are inferred from data. Note that we already know locations of torso in every image. Thus we can use a inverse transform $\phi_l^{-1}(p)$ to obtain a joint position in torso coordinate system for particular image (p is the joint position in image coordinates). For the definition of $\phi_l(p)$ see (2.7). Finally, we take the average of the above values over all images as the position of the joint in local torso coordinate system.

4.3 Learning Model Parameters

In previous sections we discussed data acquisition, ground truth specification and model adjustment issues. As a result we have got a set of example images $\{I^1, \ldots, I^m\}$ and a set of corresponding human poses or configurations of parts $\{\mathbf{l}^1, \ldots, \mathbf{l}^m\}$. The objective of this section is to obtain estimates for the model parameters $\theta = (\mathbf{a}, \mathbf{d})$, where $\mathbf{a} = \{a_0, \ldots, a_{n-1}\}$ is a set of appearance parameters for each part $v_i, i = 0, \ldots, n-1$ and $\mathbf{d} = \{d_{ij} | (v_i \to v_j) \in A\}$ is a set of parameterizations of distributions over relative locations of connected joints. The maximum likelihood (ML) estimator of θ is the value θ^* which maximizes

$$p(I^1, \ldots, I^m, \mathbf{l}^1, \ldots, \mathbf{l}^m | \theta) = \prod_{k=1}^{m} p(I^k, \mathbf{l}^k | \theta,) \tag{4.1}$$

where we assume that each example was generated independently. Since $p(I, \mathbf{l}|\theta) \propto p(I|\mathbf{l}, \theta) p(\mathbf{l}|\theta) = p(I|\mathbf{l}, \mathbf{a}) p(\mathbf{l}|\mathbf{d})$, the ML estimator is

$$\theta^* = (\mathbf{a}^*, \mathbf{d}^*) = \arg\max_{\theta} \prod_{k=1}^{m} p(I^k|\mathbf{l}^k, \mathbf{a}) \prod_{k=1}^{m} p(\mathbf{l}^k|\mathbf{d}). \tag{4.2}$$

The first term in this equation depends only on the appearance parameters of the parts, whereas the second term depends only on the parameterizations of distributions over distances between connected joints. Thus we can solve for appearance and connection parameters independently.

4.3.1 Learning Parameters of Appearance Models

From equation (4.2) we see that

$$\mathbf{a}^* = \arg\max_{\mathbf{a}} \prod_{k=1}^{m} p(I^k|\mathbf{l}^k, \mathbf{a}). \tag{4.3}$$

The probability of seeing image I^k given the configuration $\mathbf{l}^k$ of parts is given by equation (2.3). After plugging it into (4.3) we obtain

$$\mathbf{a}^* = \arg\max_{\mathbf{a}} \prod_{k=1}^{m} \prod_{i=0}^{n-1} p(I_i^k | \mathbf{l}_i^k, a_i) = \arg\max_{\mathbf{a}} \prod_{i=0}^{n-1} \prod_{k=1}^{m} p(I_i^k | \mathbf{l}_i^k, a_i). \qquad (4.4)$$

Now we see that to find $\mathbf{a}^*$ we can independently solve for each a_i^*. Moreover,

$$a_i^* = \arg\max_{a_i} \prod_{k=1}^{m} p(I_i^k | \mathbf{l}_i^k, a_i) \qquad (4.5)$$

is exactly the ML estimate of the appearance parameters for the part v_i.

There are images, where part v_i is occluded. We assume that occlusion information is implicit in location variable L_i. We take image distribution $p(I_i|l_i, a_i)$ given that the part v_i is occluded to be uniform. Thus, when computing ML estimate of a_i we may drop from consideration those images, where v_i is occluded. It is an interesting question for future research to rigorously model the posterior distribution over human poses $p(\mathbf{L}|I, \theta)$, when occlusion information is explicitly present in the configuration $\mathbf{L}$. The above equations are generic, in the sense that they apply to any of the appearance models.

Now we consider obtaining ML estimates for models that are combinations of simple appearance models. We rewrite equation (3.5) in the following way

$$p(I_i|l_i, a_i) \propto p(s_i^1, \ldots, s_i^R | a_i^1, \ldots, a_i^R) = \prod_{r=1}^{R} p(s_i^r | a_i^r), \qquad (4.6)$$

where R is the number of combined appearance models, s_i^r is an image score given part's location and corresponding to model r, a_i^r is a set of parameters of the r-th model. Once again we notice that the ML estimate can be decomposed into independent ML estimates (for reference see equation (4.4)). More precisely, the ML estimate can be written as $a_i^* = ((a_i^1)^*, \ldots, (a_i^R)^*)$. Thus it is enough to consider each appearance model separately. In our experiments we will consider only the first three models and some of their combinations, leaving variable width templates for future work.

Note that all appearance models that we consider are based on some score, which can be interpreted as a dissimilarity measure. We have few observations available, thus we assume that for every appearance model the distribution of dissimilarity score comes from some parametric family of distributions. Our task is to obtain ML estimates of the parameters of these distributions. The main question now is what parametric family of distributions to use.

Distribution of Dissimilarities

Implicit in almost all data analysis tasks is some statement about the manner in which observations vary from its fitted value. This important issue is studied in [21] and [22], where a variety of distributional assumptions for dissimilarity are considered, with the log-normal distribution being favored for most situations. The log-normal distribution is the probability distribution of a random variable whose logarithm is normally distributed. A random variable X has log-normally distribution if $\log(X)$ is normally distributed. The probability density function of the log-normal distribution parameterized with μ and σ is as follows (for illustration see figure 4.3)

$$f(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}x\sigma}\exp(-\frac{(\ln x - \mu)^2}{2\sigma^2}).$$

The choice of a log-normal distribution can be motivated by the following facts

- dissimilarities d are defined naturally on the positive real line (the log-normal distribution attaches positive probability to all positive values of d);
- the log-normal distribution has considerable positive skewness, which corresponds to an increase in spread of responses with location. That is there are more possible outcomes, which correspond to a bigger dissimilarity value.
- Finally, the log-normal assumption can be thought of as arising from a transformation of the response to an interval scale through the use of logarithms, and then assuming normality [22].

We will see in the experiments section, that the log-normal distribution fits our training data very well. ML estimates of the log-normal distribution parameters are

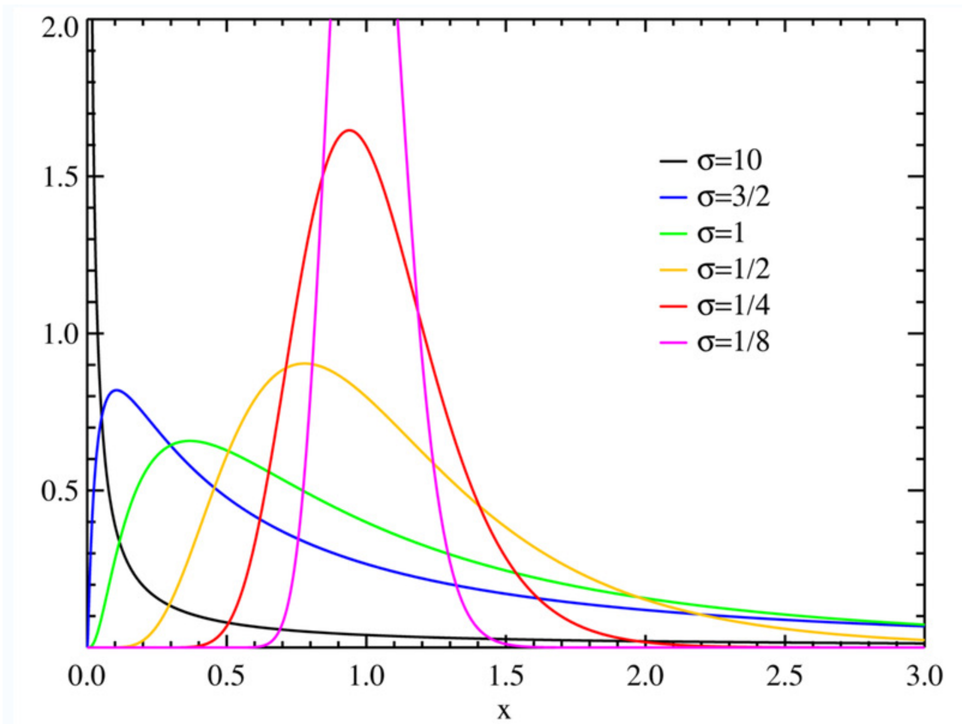

Figure 4.3: Probability density function of a log-normal distribution with fixed $\mu = 0$ and different σ. Adapted from [1].

given by

$$\mu^* = \sum_{i=1}^{N} \ln x_i / N \tag{4.7}$$

$$\sigma^* = \sqrt{\sum_{i=1}^{N} (\ln x_i - \mu^*)^2 / N}, \tag{4.8}$$

where N is the number of observations.

So far we have fully specified the procedure to obtain the ML estimates for models based on dissimilarity score. For completeness we consider the background subtraction based appearance model given by equation (3.4). The right hand side is a product of Bernoulli distribution functions. The ML estimates of parameters q_0 and q_1 are given by

$$q_b^* = \frac{\sum_{i=1}^{N} (N_b^F)_i}{\sum_{i=1}^{N} (N_b)_i}, \qquad b = 0, 1. \tag{4.9}$$

Data Correction

Our training data contain errors. Consider first the appearance model based on chamfer distance. Suppose that l_i is the location of a body part v_i in a given image as specified by the ground truth. Usually it is possible to make the dissimilarity score smaller by a slight change of the location of the body part. Let d' be the smallest dissimilarity score among all scores attained by locations from a small neighborhood of l_i. We assume that the location l'_i corresponding to d' still can be considered correct for the given image. We would like to use d' instead of the score d corresponding to l_i, as a training example. In order, to correct for these errors in training data we assume that there exists a constant $0 < \lambda \leq 1$ such that $d' = \lambda d$ for all training examples. This amounts to downscaling along data axis. ML estimates corresponding to the corrected data set are given by

$$(\mu')^* = \mu^* + \ln \lambda \tag{4.10}$$

$$(\sigma')^* = \sigma^*. \tag{4.11}$$

Note that the above procedure reduces both mean and variance of fitted log-normal distribution.

Similarly to the previous paragraph we can correct observed dissimilarities of color histograms. Recall that these dissimilarities take values in $[0, 1]$ and that the bigger is dissimilarity the better is match. Thus we would like to downscale data towards value 1. Therefore, we assume that $d' = 1 - \lambda(1 - d)$. In this case we recompute corrected ML estimates from data. Similar ideas might be used for background subtraction based models. We discuss our choice of an appropriate λ for each appearance model in the experiments section.

4.3.2 Learning Distributions over Relative Locations of Joints

In this section we consider distributions over relative locations of joints of connected parts. Similarly to learning appearance model parameters we can decompose estimation of different distributions into separate tasks. Combining equations (4.2) and (2.4) we get

$$d^*_{ij} = \arg\max_{d_{ij}} \prod_{k=1}^{m} p(l^k_j | l^k_i, d_{ij}) \tag{4.12}$$

for each pair (i, j), such that $(v_i \rightarrow v_j) \in A$. Furthermore, exploiting the assumption of independence between different types of distances given by equation (2.9) we see that distributions over different types of distances can be again learned separately.

In our experiments we use very simple non-parametric estimates of distributions over distances, namely histograms obtained from training data. These are exactly the ML estimates of probabilities comprising distributions of interest. Our data set, however, is very small. We may get a zero mass for some distance values even though these distances may occur in future images. Moreover, we can specify constraints on distances between joints of connected parts quite reliably. Thus we decide to incorporate available prior knowledge into final estimates of distributions. Prior distributions are obtained as follows. First, we specify constraints. Then we assume for simplicity that the distribution is uniform over values, which satisfy the above specified constraints.

We combine available data and prior knowledge in the following way:

$$\hat{p} = \lambda p_0 + (1 - \lambda) p^* \qquad 0 \leq \lambda \leq 1, \tag{4.13}$$

where p^* is a ML estimate of a probability of distance between joints taking on a particular value and p_0 is a prior value for the given probability. We discuss our choice of mixing coefficient λ in the experiments section.

Note that it is possible to obtain λ in a data driven way. For example, one may consider the above defined prior probabilities as a model under the null hypothesis H_0. Next we may test this hypothesis. A common statistic for testing multinomial distribution is Pearson statistic [26]. This statistic converges in distribution to a χ^2_{k-1} distribution if H_0 is true, where k is number of possible outcomes of a random variable under consideration. We may use the asymptotic p-value of our statistical test as a measure of fit of the prior model. The bigger is p-value the bigger weight we would like to use for the prior model. It seems sensible to set λ equal to calculated p-value. This interesting idea needs further elaboration. We leave evaluation of presented data driven procedure to obtain mixing coefficient λ

for future work.

The form of learned distributions over relative locations of joints has a serious influence on execution time. Consult section 2.4 on sampling and in particular subsection 2.4.3 on fast convolution for more details. Note that it should be beneficial to put some constraints on the distributions of distances. We might require that each distribution should be decomposable in a weighted sum of uniform distributions, where number of this uniform distributions do not exceed some fixed small constant. The above constraint would ensure that sampling is done fast enough. We leave the problem of constrained estimation of distributions for further research.

View-specific Models

In current work we consider a tree structured model. Thus we disregard many probably important correlations. Using the same pictorial structure model for every possible human pose would result in flat distributions over joint distances. Consider a shoulder joint for example. If person stands sideways than shoulder is located on the middle line of torso. If, however, we have a frontal view of a person then shoulder is located quite far away from the middle line. Thus if we use the same model for all views, we get quite a spread out distribution over x and y distances. Similarly distributions over angular differences will become tighter if we use different models for different views or different sets of poses. It is very important to have tight distributions over distances in view of tree structure assumption. We, therefore, decide to introduce three pictorial structures for modeling people oriented to the right, to the left and those having frontal or rear views. We use the same appearance models inside the groups of upper legs, lower legs, upper arms, lower arms and heads present in all three pictorial structures. Two separate torso models are used: one for frontal view, another for sideways views.

Sampling is organized as follows. First, we randomly choose a pictorial structure model, where probability of each model is learned from training data. Next, we sample configurations from the chosen model. Because of running time efficiency, we first do selection of models. Next we count number of samples, which are to be drawn from a distribution represented by a particular model, and only then we apply our sampling algorithm.

Chapter 5

Experiments

5.1 Building the Model

We start our experiments with choosing appropriate discretization of the space of possible locations of body parts. Next, we test different appearance models of the parts. As a result we pick one appearance model, which will be used in all the following experiments. We finish construction of our pictorial structure model with a discussion of the learning procedure for distributions over relative locations of joints.

5.1.1 Discretization

Performance of our sampling algorithm depends on the chosen discretization of the space of possible locations of body parts. This choice is a trade-off between running time and accuracy. After a series of preliminary experiments we choose the following. We scale the region of interest to make a person depicted there approximately 130 pixels high. We use only every fourth pixel both in x and y direction as a possible position of part's anchor point. Angles are discretized into 32 values: $\frac{2\pi}{32}i, i = 0, \ldots, 31$. We use only 3 scales: $1.0, 0.8, 0.6$. These choices make the average size of the state space of a part's location close to 30000, if the specified bounding box around a person in the image is tight. The state space increases if the bounding box is obtained automatically.

5.1.2 Choosing Appearance Model

Choice of appropriate appearance models of parts is crucial for performance of our recognition algorithm. We had to carry out experiments with a few appearance models and different their parameterizations before we achieved satisfactory performance. We believe that one can both increase the accuracy and decrease the running time by making a better choice of appearance models and their parameterizations. One possibility is to consider variable width templates together with optimization techniques discussed in the chapter on appearance models.

We consider the following appearance models:

a. model based on chamfer distance

b. model based on color dissimilarity between foreground and background

c. model based on background subtraction

d. model (a) combined with light version of background subtraction based model (c')

e. model (b) combined with (c')

Models (a), (b) and (c') are based on a dissimilarity score. Following our discussion of learning appearance model parameters we fit log-normal distribution to a set of observed dissimilarities for every body part under each of the models (a), (b) and (c') (recall that some parts like lower legs are considered equivalent). Histograms of observed dissimilarities together with fitted log-normal distributions for torso and lower leg are illustrated in figure 5.1. For reference we also collect dissimilarities for these body parts placed at randomly chosen locations.

Notice that log-normal distribution fits data quite well for all the models except the light version of the background subtraction based model (c') (see figure 5.1, c' right). The problem with the last model is that some body parts, e.g. lower legs, are very well separable in many cases using background subtraction. Thus we get many dissimilarities close to 0, which make ML estimates of log-normal distribution parameters to over-stress importance of very small dissimilarities. In other words we get a distribution with a very high peak close to 0. Our intention,

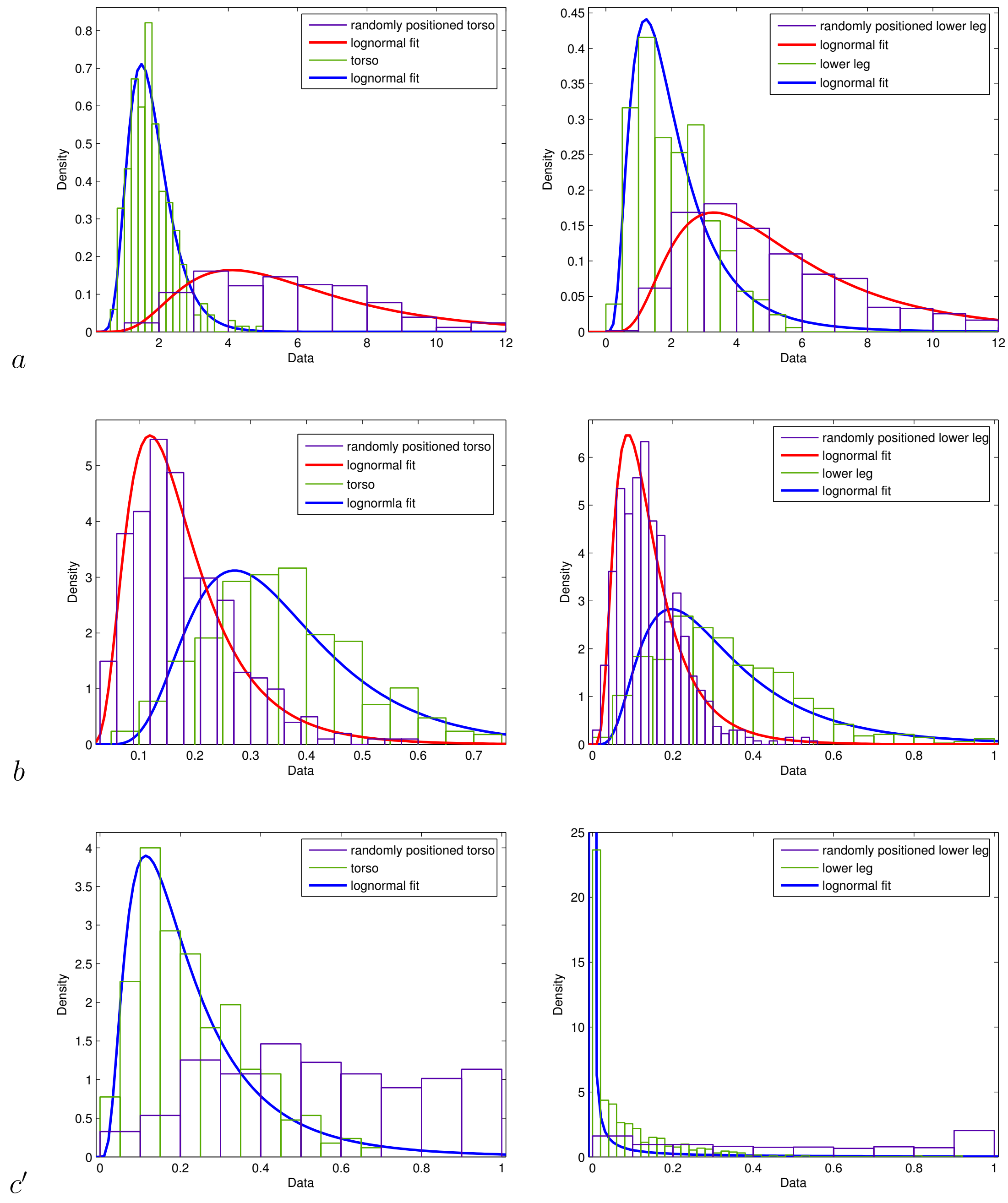

Figure 5.1: Observed dissimilarities and dissimilarities of randomly positioned parts together with fitted log-normal distributions corresponding to models (a), (b) and (c').

however, is to use model (c') for guidance, to reduce high responses from matches to background clutter. We would like to keep chamfer distance or color dissimilarity as a basic, most influential score. Therefore, we do the following. We combine the dissimilarity scores under model (c') from all the body parts. Then we fit a normal distribution with $\mu = 0$ and free parameter σ to this combined dataset. In this case we should not get a highly peaked distribution. This can be seen in figure 5.2.

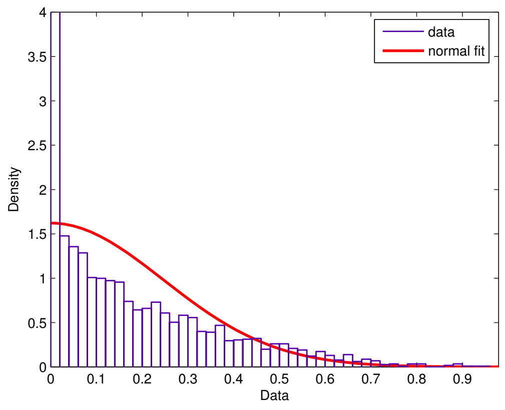

Figure 5.2: Normal distribution with $\mu = 0$ fitted to dissimilarity scores under model (c') for all parts together.

Finally we note that we are using the correction parameter $\lambda = 0.5$ for all models except (c') (above figures display data before corrections). See section 4.3.1 for a discussion of data correction issue. This correction is rather strong. It is necessary to make distributions more peaked, otherwise we would require dense and thus inefficient sampling, which would also make post-sampling selection procedure difficult. This strong correction can be explained by the facts that we are using quite coarse discretization and that our training data contain many observations of actually occluded body parts.

In order to choose the ultimate appearance model we visually inspect both the results of body pose sampling algorithm as well as responses of the individual appearance models. In the latter case we depict a number of body part locations with the highest probabilities under different appearance models. For an example see figure 5.3.

One may notice in figure 5.3 that the chamfer score based model (a) is very much distracted by background clutter. Background clutter has much smaller influence on the color dissimilarity based model (b). This is so, because, under model (b) we

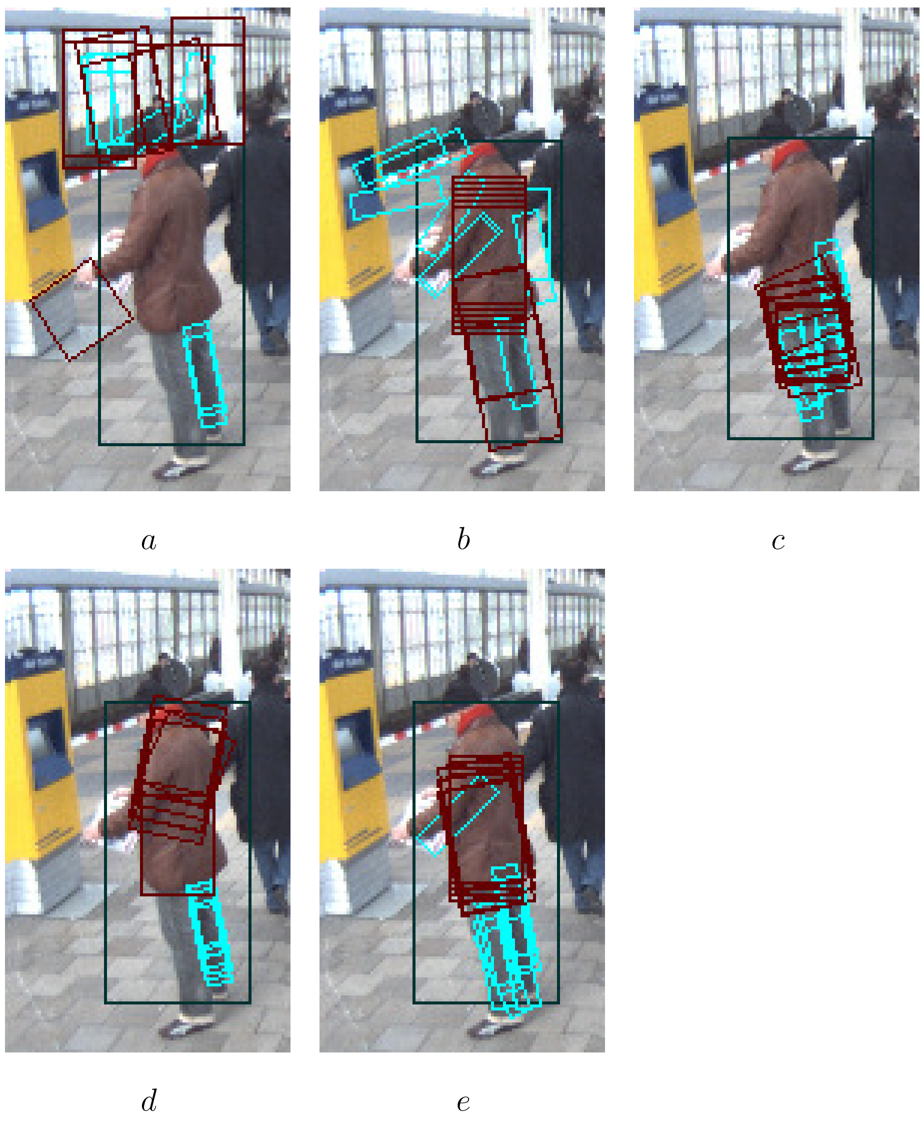

Figure 5.3: 10 locations of torso and 10 locations of lower leg with highest probabilities under appearance models a,b,c,d and e.

average responses of a bigger number of pixels than under model (a). Thus many edges from background clutter will have almost no influence on model (b). We also notice that model (c) is easily deceived when some body parts are close together. This is, however, a very common situation for upper arm and torso as well as for upper legs. Models (a) and (b) cope with this situation better. For example, it is often the case that edge between legs is extracted even though legs are placed close together. We notice that combined models (d) and (e) are much better than simple models (a), (b) and (c). We found out that there is almost no difference in performance of models (d) and (e), except that the first of them has smaller execution time. Therefore, as our ultimate appearance model we choose model (d), which is a combination of chamfer distance and the light version of background subtraction based models.

5.1.3 Distributions over Distances between Joint Locations

Distributions over distances between joint locations are estimated according to our discussion in the chapter on learning model parameters. First, we specify constraints on possible distances and construct corresponding prior distributions. Then we obtain distributions from training data by histogramming distances. Finally, we mix these two using coefficient λ. Note that small changes of the value of the mixing coefficient λ have almost no effect on the resulting distributions and thus on the performance of sampling algorithm. We fix $\lambda = 0.25$, which gives acceptable estimates of distribution under consideration. See figure 5.4 for an example of learned distributions. Preliminary experiments with full body configuration sampling using the above specified λ showed that our choice is acceptable.

5.2 Experiments with the Complete Model

5.2.1 Human Pose Estimation

In this section we evaluate our pictorial structure algorithm in terms of human pose estimation.

The Setup

First we explain the setup of the experiments. According to the discussions in section 4.3.2 we use three view-specific pictorial structure models. Corresponding appearance models and parameterizations are chosen as stated above.

In order, to simplify evaluation we consider only one person per input image. We restrict our attention to some region of interest of the given image. Other people may be interacting with the person under consideration, may stand close to, behind

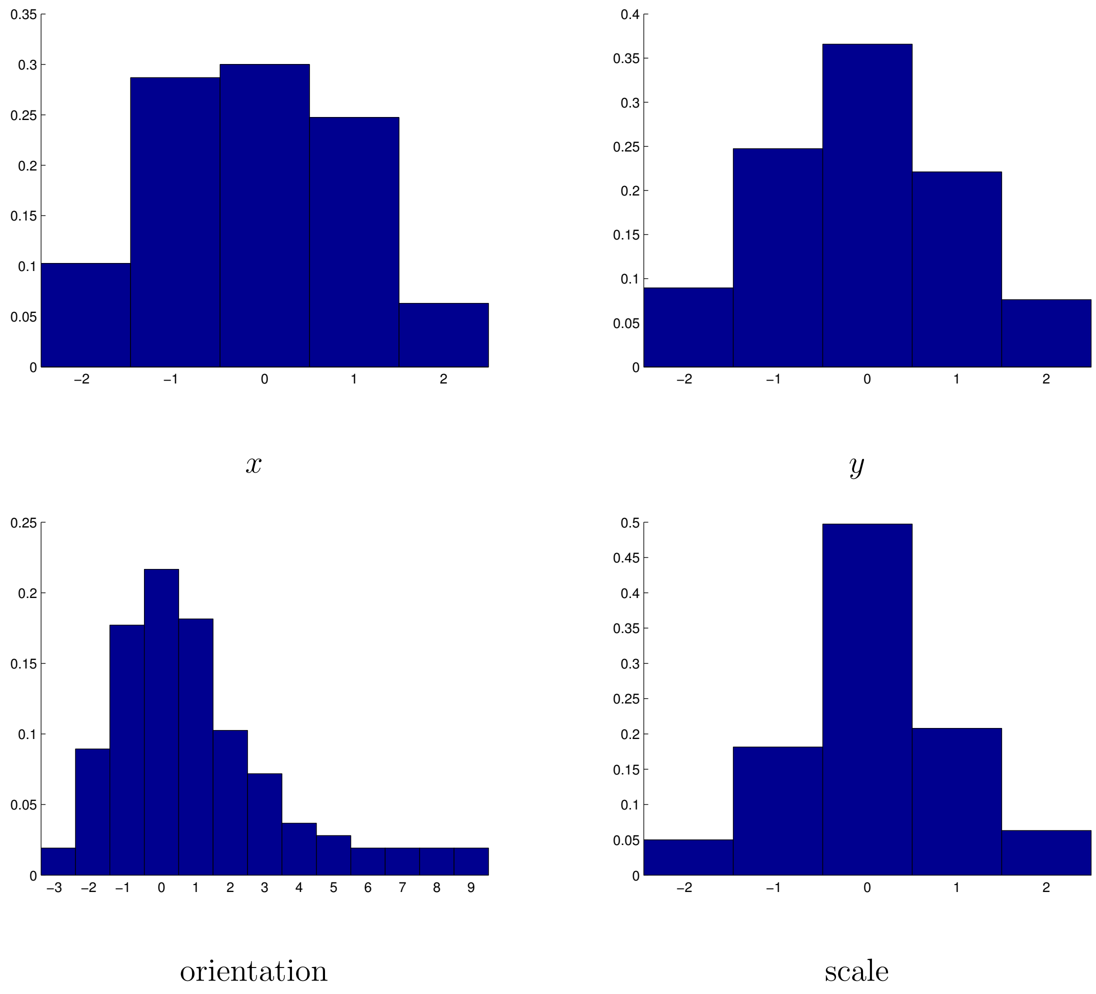

Figure 5.4: Learned distributions over distances for the joint between torso and right upper leg. The pictorial structure models a person going right.

or in front of this person. Thus our region of interest will inevitably contain other people. Usually those will be partially visible. We define the region of interest in terms of a bounding box containing all joints of the person under consideration.

There are different ways of how the bounding box can be obtained. For example, we may use an output of a tracking algorithm or detection system. For our case, we develop a simplistic person detector, which operates on a background subtracted image. Using this detector we obtain a number of bounding boxes for each image. Next, we select (if there exists) a bounding box corresponding to our person of interest. And only then we apply our pose recognition algorithm.

It is a very difficult task to evaluate a system consisting of a detector and a pose recognition component as a whole. Moreover, the first simplistic component may introduce many errors and bias the reader against the pose estimation component. Therefore, we prune false detections before applying pose recognition algorithm. Our ground truth data of human poses contain bounding boxes specified during the labeling stage. The only our intention is to substitute these hand specified bounding boxes with automatically detected ones. This is needed to show that the algorithm does not rely on a very precise specification of a bounding box.

After we run a simplistic human detector and prune away false detections, we are left with 269 images of people. For each of these images we have automatically detected bounding box, containing all joints of a person of interest, and the ground truth specification of the pose, which will be used to evaluate the output of our algorithm.

Next we define a measure, specifying how good is any given pose hypothesis. We say that a body part is found correctly if x, y and orientation components of the location describing vector are at most two discretization units away from the corresponding ground truth values. Let us give a mathematical formulation. Consider i-th body part v_i, which hypothetical position is given by $l_i = (x_i, y_i, o_i, s_i)$. Suppose that part's true position is $\hat{l}_i = (\hat{x}_i, \hat{y}_i, \hat{o}_i, \hat{s}_i)$. Now we say that the part is correctly detected if

$$|x_i - \hat{x}_i| \leq 2 \;\; \& \;\; |y_i - \hat{y}_i| \leq 2 \;\; \& \;\; |o_i - \hat{o}_i| \leq 2 \tag{5.1}$$

The total measure of pose dissimilarity we define as a number of incorrectly found body parts. We ignore position of those parts, which are occluded according to the ground truth.

Note, that even though the above constraints may seem quite relaxed, the configuration of all body parts satisfying the above constraints will tend to assume a pose very close to the correct one. In other words, the configuration of parts specifies additional constraints. From the other side, if we make the constraints tighter, we may strongly underestimate our algorithm. For example, it is very often the case that some joint point can be placed in many different locations (not just one) still defining the correct pose. Moreover, our ground truth is not ideal. It is sometimes quite difficult to guess where the actual joint is, because of partial occlusions of body parts and loose clothing.

To illustrate our measure of pose fit we provide some examples. In figure 5.5 in the first row we depict the only 3 pose hypotheses out of 300 samples for the given image, which are fully correct according to our measure. In the second row one can see ground truth labeling and two selected hypotheses with 1 mistake in each (erroneously found body parts are filled). In our opinion these should be valid pose hypotheses. Our measure, however, is too restrictive in this case.

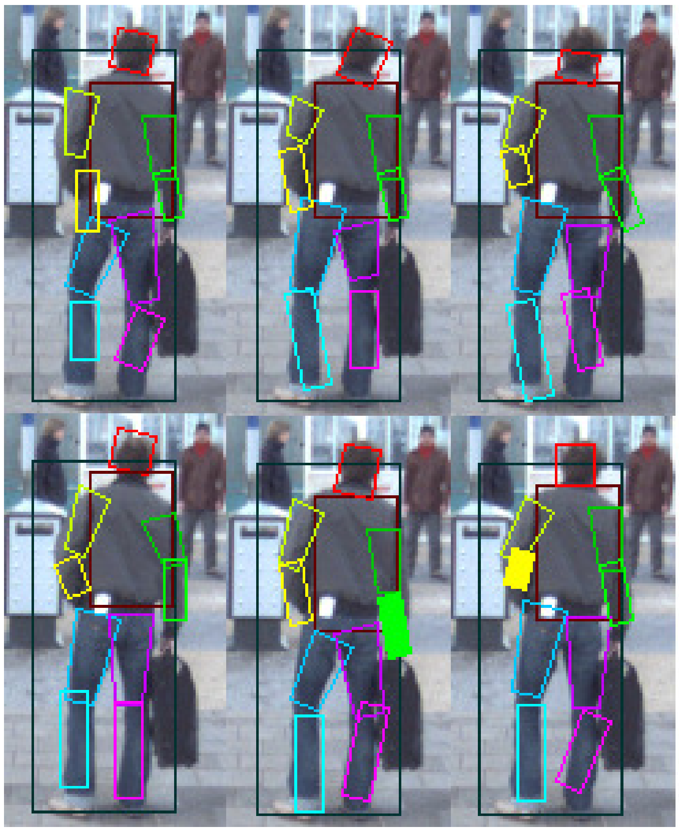
Figure 5.5: Illustration of pose quality measure. First row: the only 3 fully correct poses out of 300 samples; second row: ground truth labeling and 2 selected poses with 1 mistake (filled).

Results

During our experiments we use 10-fold cross validation strategy. For each given image we sample 300 poses from the posterior. Next, we sort these samples in decreasing order of posterior probability. For each sample we also compute the measure of fit to ground truth as was described above.

Our pictorial structure models rely on quite strong assumptions. For example, prior distribution of poses is given by a tree model. Moreover, we use simplistic appearance models for parts. This means that we introduce modeling errors. Therefore, posterior probability defined by the pictorial structure may not be very reliable. We believe that our models are good for preselection of candidate poses, that is for sampling. After we obtain a number of candidates, another richer model is to be used for evaluation and reranking of these candidates. Note, that from a very huge set of possible poses we select a very small set of candidates. Thus any algorithm, even inefficient one, can be used to make the final choice.

Accurate evaluation and reranking of samples is a separate and probably difficult task. We leave it for future work. At present we use posterior probability to rank our samples, even though we argued that it is unreliable. In the following figure 5.6 we depict percentage of images (y axis), for which a sample with given quality (x axis) was present among the k samples with highest posterior probability. The case when $k = 1$ approximates the quality of a MAP solution. On the other hand, the case when $k = 300$ would give the accuracy of our pose estimation algorithm, if we were given ideal hypothesis selection algorithm.

According to our expectations we see that the MAP solution seldom is a good pose hypothesis. Only in 10% of images it gives the correct pose and in 30% we get at most one mistake. There is, however, a correct sample among 300 samples in 65% of cases and a sample with at most one mistake in 90% of cases.

In the next figure 5.7 we swap the dimensions. From this figure we see that it is on average enough to consider about 70 first samples, if we aim at finding correct pose in 50% of images.

Finally, we depict some more statistics. We consider position or rank of the first sample, which has some given quality (maximal number of incorrectly found body parts) for all test images. Then we compute average, standard deviation and the median of the ranks. Average and standard deviation are very sensitive to outliers, thus we prune away 10% of ranks having the largest values. Moreover, we only

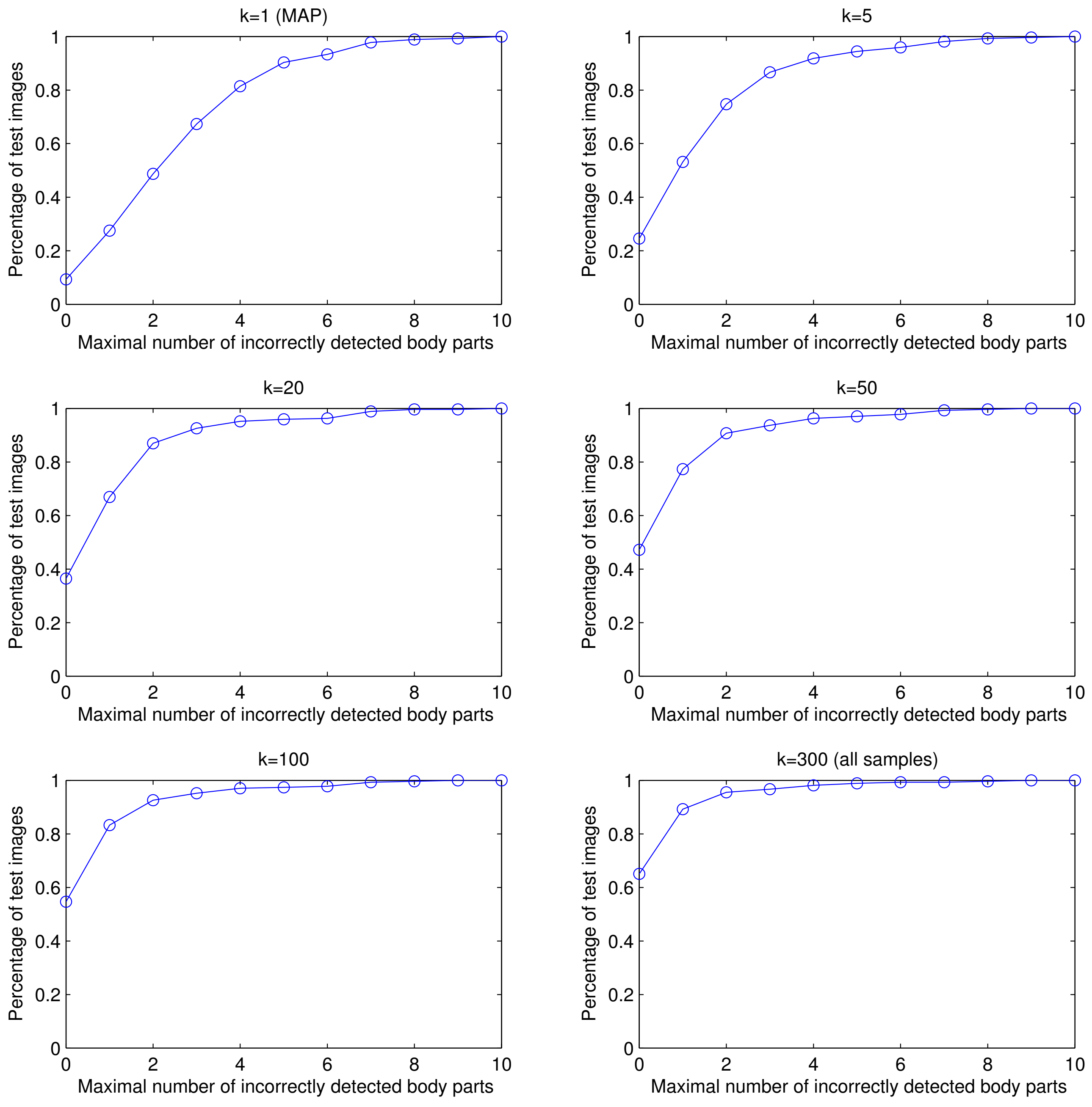

Figure 5.6: Percentage of test images (y), for which a sample with at most x incorrectly found body parts is present among k samples with the highest posterior.

consider the cases, where a sample with given quality was found among all the samples. The median provides more insight. We neither prune outliers, neither

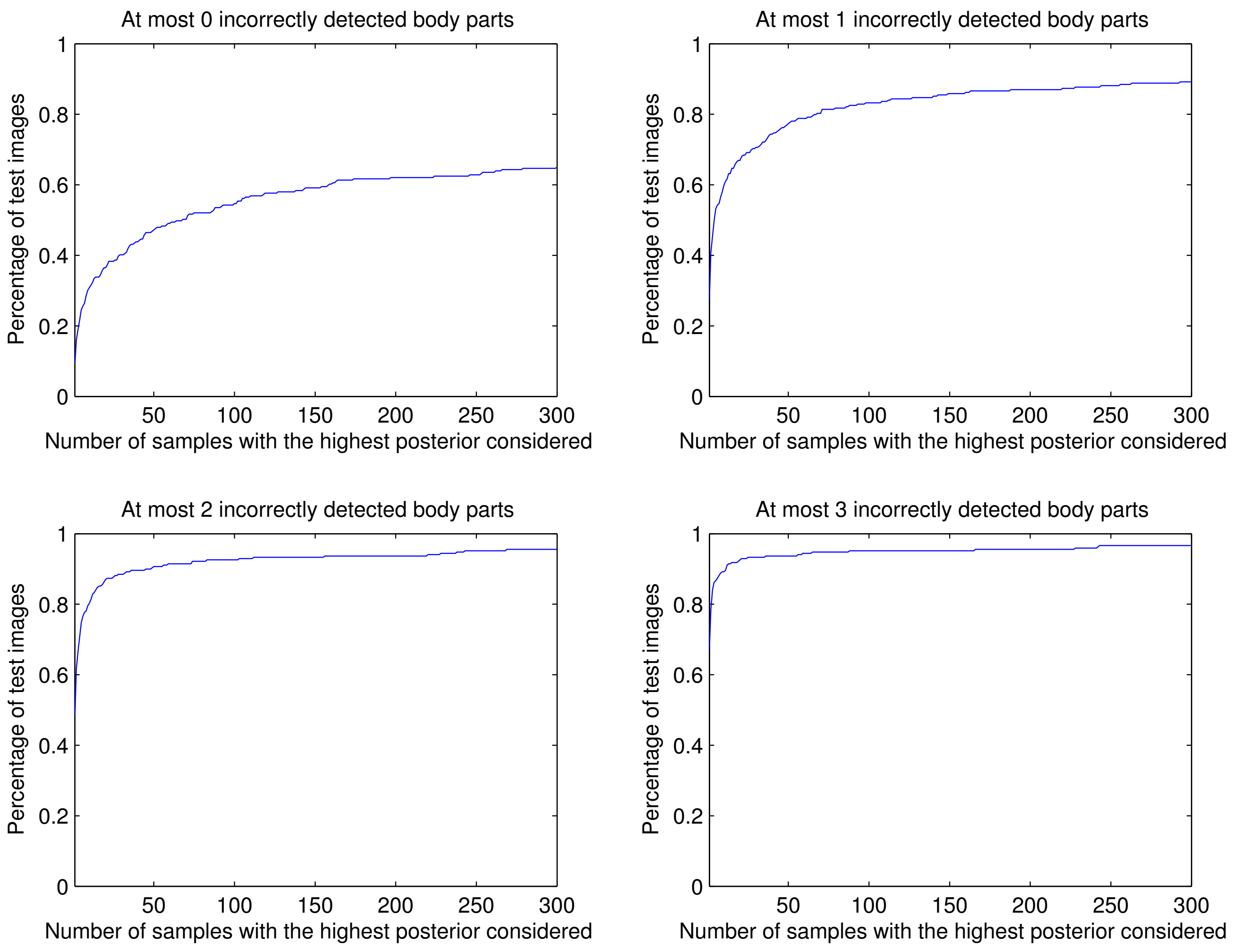

Figure 5.7: Percentage of test images as a function of the number of samples with the highest posterior, where a sample with given quality was found among samples considered.

drop cases, where rank is undefined. Instead we assume the rank is ∞ in this case. It is interesting to note that it is enough to consider 5, respectively 2 first samples with the highest posterior, if we aim at finding a pose with at most 1, respectively 2 mistakes, in 50% of images.

In order to provide the reader with more qualitative insight into results, we present some examples of good (figure 5.9) and bad (figure 5.10) pose hypotheses. From the examples of bad hypotheses one may see the common problems, such as cluttered background, bad fit of rectangles to parts they model and occlusions. When

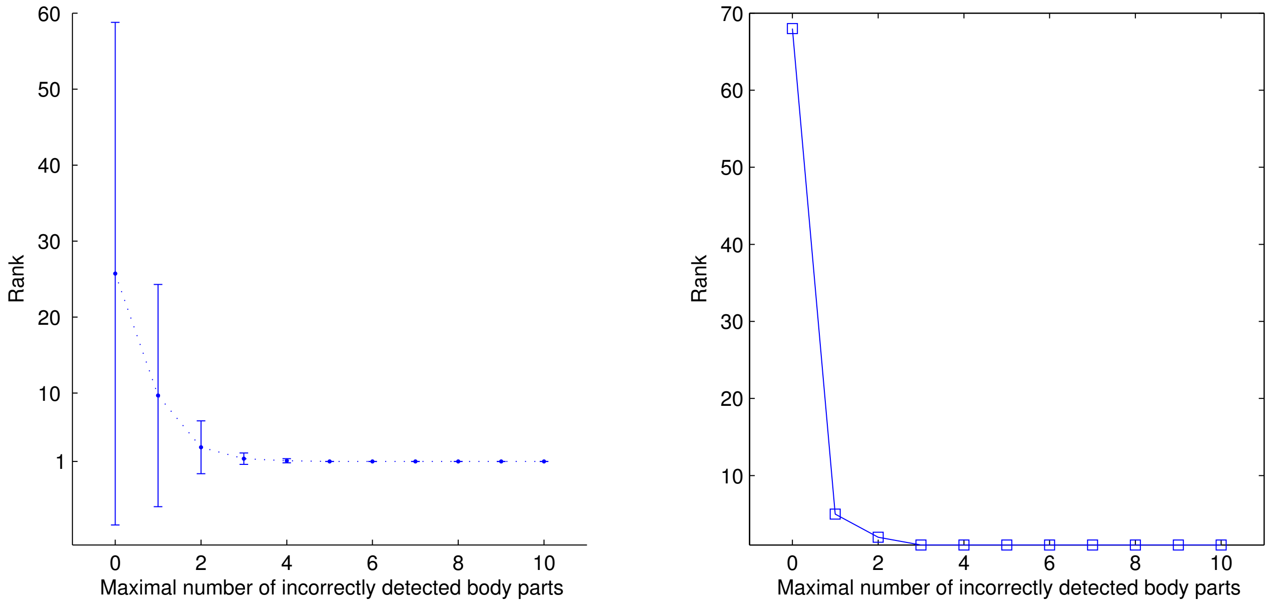

Figure 5.8: Statistics over ranks of the first sample with some quality in the list of samples ordered by their posterior. Left: averages and standard deviations (shown by vertical bars); Right: medians.

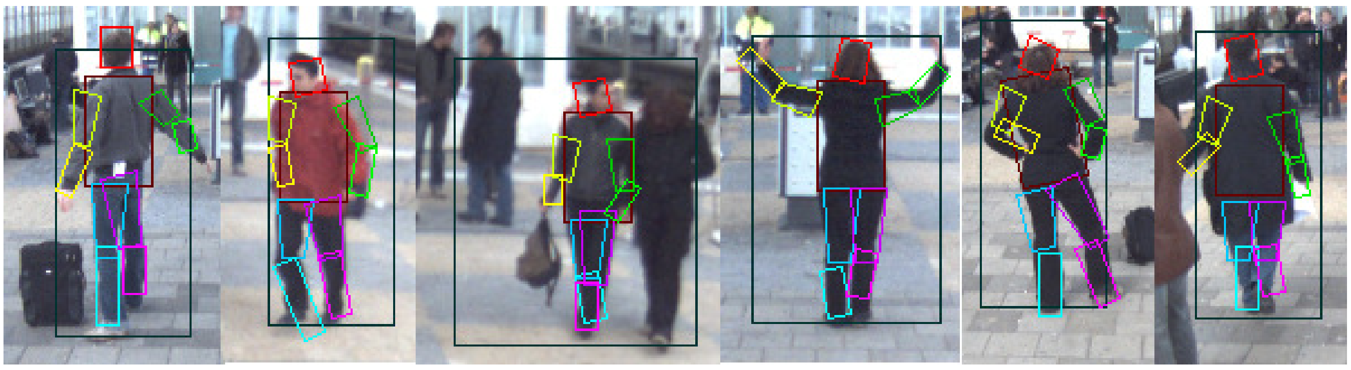

Figure 5.9: Examples of correct samples of human poses.

a person wears loosely fitting clothes or some of his body parts are close together, so that appearance models of separate parts score low, we may observe samples, where parts are positioned against background. As a remedy one may try to improve our very simple background subtraction and edge segmentation algorithms, try to use different appearance models. We also believe that variable width templates will help us to solve this problem to some extent.

Finally, we note that our pose recognition algorithm requires some kind of post-

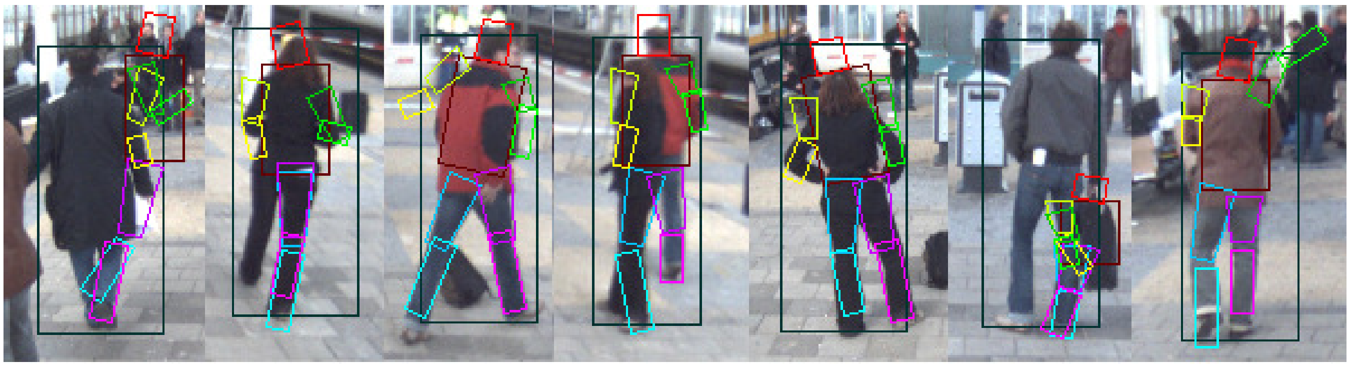

Figure 5.10: Examples of incorrect samples of human poses.

processing system, which would allow us to prune away erroneously positioned parts. This system, for example, may be based on color and pattern similarities of different parts or may exploit constraints imposed by the balance.

5.2.2 Classification between Presence and Non-presence of a Human

In this section we explore classification power of pictorial structures. Suppose we apply some human detection algorithm and obtain a set of regions of interest. Now we would like to use the pictorial structures to test if those regions indeed contain a person. One also may consider splitting the whole image into boxes, possibly overlapping, and then testing each of those. In this way we may get a person detector. So our task is to discriminate between regions, which contain a person, and regions, which does not.

Classification is done in the following way. For a given image we obtain 300 samples and select the one with the highest posterior. This is just an approximation of the MAP solution. Then we determine presence of a person by thresholding the posterior value. Note that actually we compute not a posterior probability for a pose, but its scaled value. The scaling coefficient depends both on the view-specific model, which we use, and on the given image region. We believe that our scaled posterior is even more descriptive than posterior itself. For example, our score does not depend on the size of region of interest.

We use the same set of images as before. The set of scores corresponding to boxes containing a human we take from the above experiment. The set of scores corresponding to empty boxes we generate in the following way. First we learn our pictorial structure models from all the examples available. Then for each image from our set of images we randomly shift the bounding box along the x axis. Next, we sample 300 poses given this shifted bounding box and record the highest posterior value.

Figure 5.11 depicts MAP scores using a logarithmic scale. After plotting these scores, we saw a very good separation between person and non person cases. Some non person scores, though, were very high. We decided to check the images. We found out that in most of the images, where we get high posterior value, we happened to shift the bounding box to another person having similar scale. In figure 5.12 we depict receiver operating characteristic (ROC) curves for two cases: before and after manual reclassification of MAP scores.

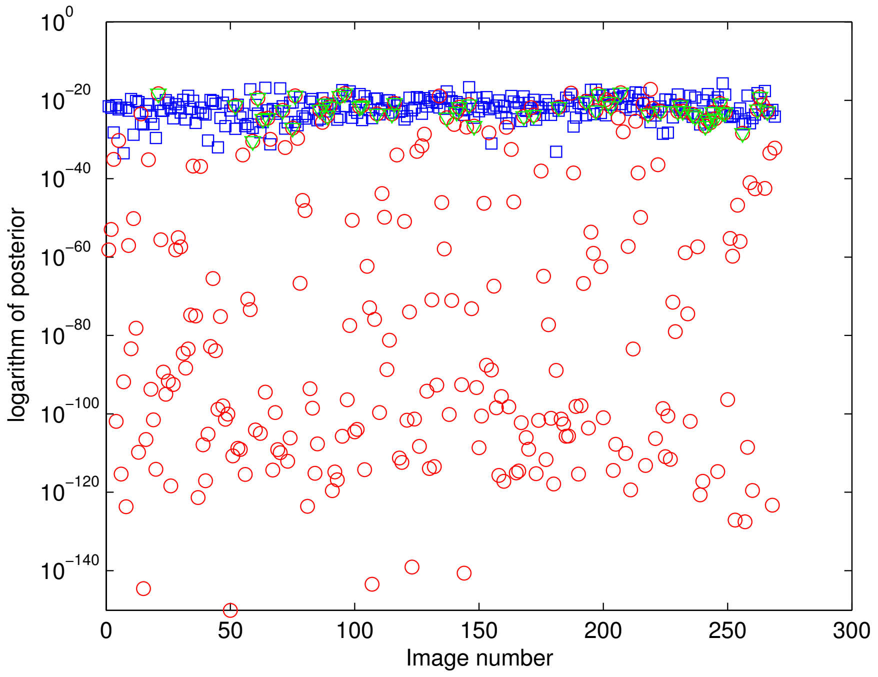

Figure 5.11: Approximate MAP values for bounding boxes with human (blue squares) and without human (red circles) being present. Cases, where bounding boxes were assumed empty, but manual checking showed human presence are denoted by green triangles.

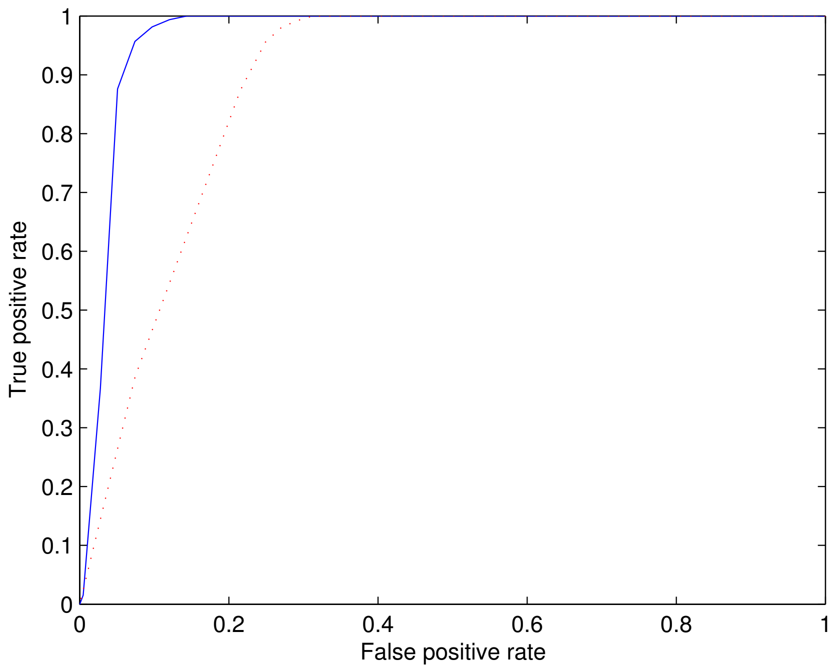

Figure 5.12: ROC curves: before manual reclassification (red dotted line) and after manual reclassification (blue solid line).

Finally, we show in figure 5.13 some MAP poses for shifted bounding boxes. First case was reclassified as a positive example after manual checking. We also note that some boxes, which remained as negative examples, contain parts of a human body or even a full person, but in quite different scale. These examples usually tend to return rather high posterior score.

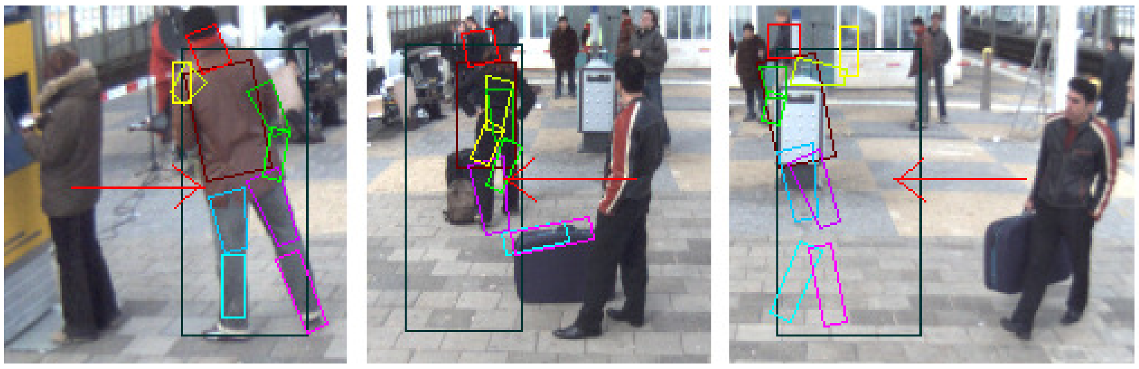

Figure 5.13: Examples of MAP poses for 'empty' bounding boxes.

5.2.3 No Bounding Box Setup

In this section we assume that bounding boxes are not available. We do not use any human detectors, but apply our sampling algorithm to the whole image. In this setup we encounter the following problems:

- increased running time
- attraction of cluttered background
- undetermined scale of a person
- presence of multiple people

Depending on the scale of a person running time in our experiments increased 8-30 times compared to the setup, when bounding boxes were specified. We provide more information about running times in the next section.

Recall that our test images have cluttered background, therefore, many samples will come from cluttered upper part of the image. Such poses are 'flying in the air' and are not possible. Knowing ground plane we constrain knee joints to be below some fixed line. This simple trick solves the problem of 'flying people' and improves quality of the set of samples very much.

The scale of a person in the image is not known anymore. There also may be multiple people present in the image. Thus we run our algorithm multiples times on the same image. Each time choosing another scale from the set of possible scales. We choose this set to be $\{160, 200, 240, 280, 320\}$, where each number denotes hight of a person in pixels. Figures 5.14 and 5.16 depict two test images. Figures 5.15 and 5.17 show MAP poses for 5 chosen scales for each of test images.

It is interesting to note that MAP samples correspond to a person, unless there is no person of similar scale present in the image. We expected such behavior after we have seen in the previous section that pictorial structures are very good at classification.

We notice that (scaled) posterior values increase, when scale of person increases. Thus it is unwise to merge samples corresponding to different scales and sort them

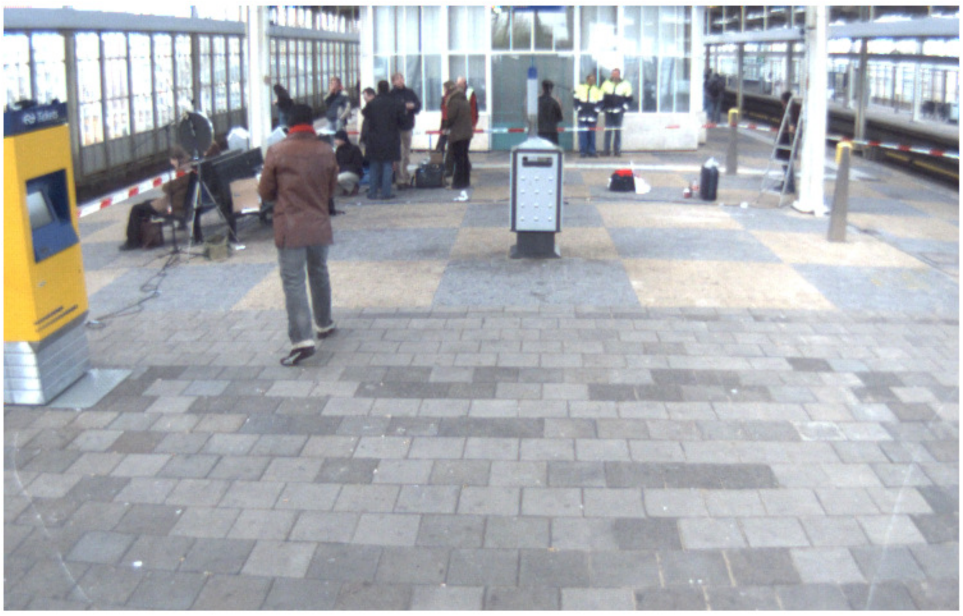

Figure 5.14: Test image nr. 1.

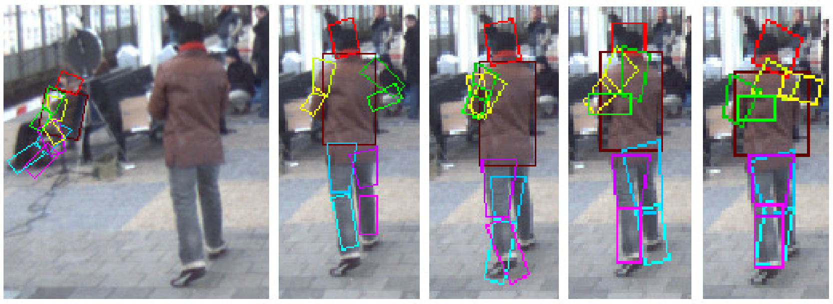

Figure 5.15: MAP poses for 5 chosen scales corresponding to the test image nr. 1.

according to posterior values. Therefore, each list of samples should be analyzed separately, unless we find a measure of fit, which is independent of person scale.

The second test image shows a major problem. If there are many people present in the image, then most of the samples will come from the most salient of these people. Even though 4-th scale suits to the person looking to the left and does not suit to the person going right, we still get the MAP sample from the second person. This person with legs apart is more salient than others.

To stress the above point we perform the following simple experiment using the image depicted in figure 5.16. We choose the scale corresponding to the person

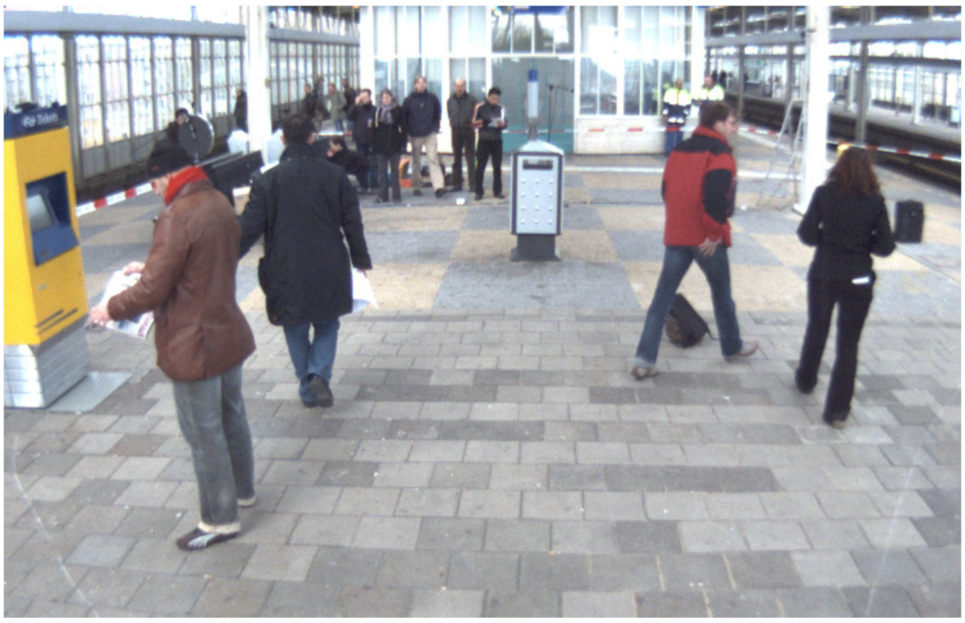

Figure 5.16: Test image nr. 2.

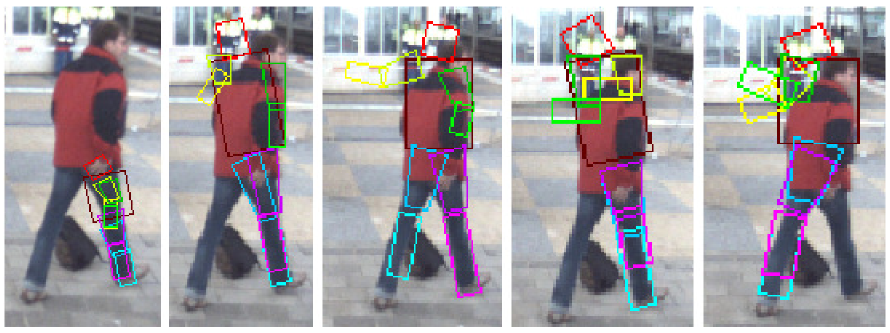

Figure 5.17: MAP poses for 5 chosen scales corresponding to the test image nr. 2.

looking left. Next we sample 30 poses and depict all of those in figure 5.18.

We see that no samples come from the intended person. Instead the samples correspond to two other more salient people (legs apart), even though, they are of smaller size. We see a few ways to solving this problem. One way is to split the image before running the algorithm, so that every part fully contains just one person. Another way is to run the algorithm, determine the best hypothesis and make the region of the best hypothesis to provide smaller appearance score. We, however, think that using a human detector to obtain bounding boxes is the most promising solution.

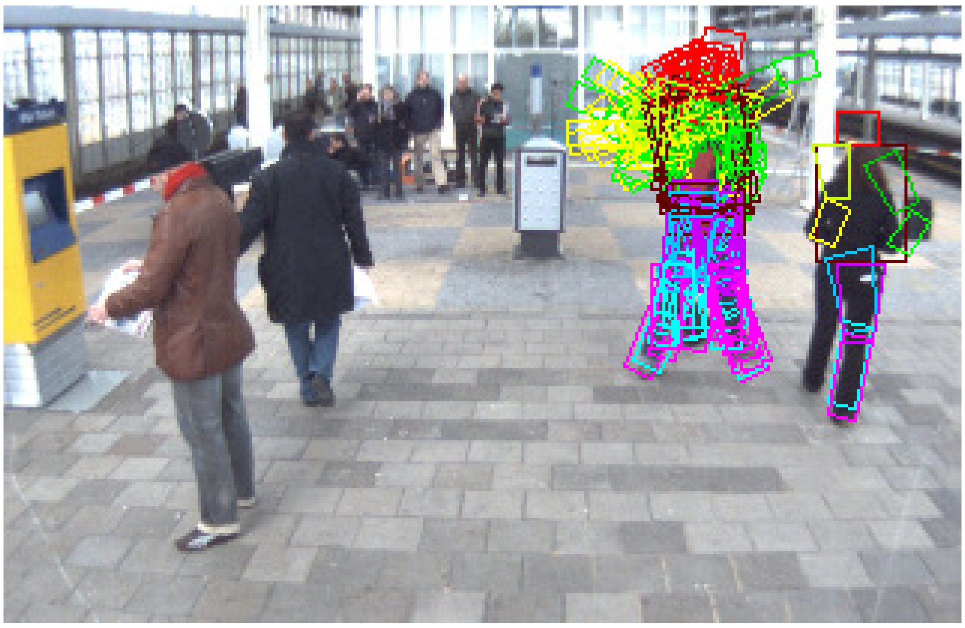

Figure 5.18: All 30 samples corresponding to the scale of the person looking left.

5.2.4 Running Time

Our intention was to build real-time pose recognition algorithm. Therefore, we have chosen a coarse discretization of the state space and very simple appearance models. Even having made these choices, it is rather difficult to achieve real-time performance. In the following table we report running times for different sizes of the state space. As usually by the state space we mean a set of possible locations for each body part separately. We used 2Ghz CPU for the experiments.

Table 5.1: Running times

State space size	Comp. of appearance probabilities (sec)	Sampling (sec)	Total (sec)
46080	6	6	12
350208	54	39	93
459360	71	54	125
630240	97	72	169
901824	139	100	239
1415424	215	152	367

The first row of the table corresponds to an average automatically detected bounding box. Next five rows correspond to the whole image of size 752×480 and 5 scales we used in the above experiment. Each scale is just the assumed person hight in pixels. The image is scaled so that the assumed hight of a person becomes about 130 pixels. Thus the bigger is scale, the smaller image becomes, and the faster we can process it.

In figure 5.19 we plot running time as a function of size of the state space. We see that running time grows linearly, which is according to our expectations, see section 2.4.

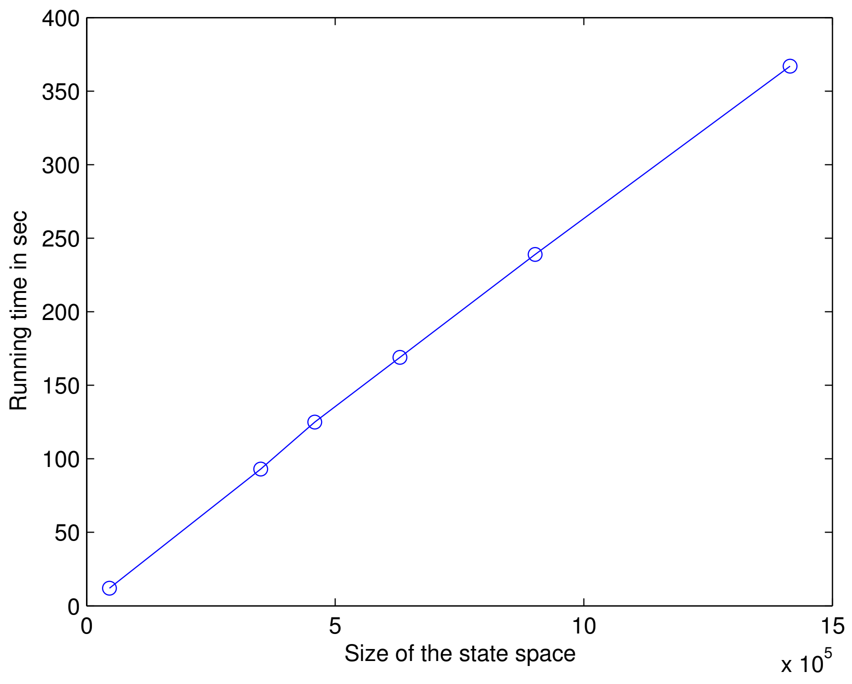

Figure 5.19: Running time.

Finally, we note that there is large space for optimization. Zooming techniques and fast convolution can be used to speed up computation of appearance probabilities of parts. See section 3.5 for more details. Sampling can be done more efficiently by constraining distributions of differences of joint locations to have some nice form, see section 2.4.3 for details.

Chapter 6

Conclusions

6.1 Contributions and Innovations

Our innovations in the domain of pictorial structures can be split in a few groups. Firstly, these are theoretical results and improvements of the pictorial structure model. We considered new advantageous parameterization of body parts, where the origin of part's local coordinate system is placed in the joint point corresponding to the parent part. We also showed that normality assumption concerning the distributions over distances between locations of joints, can be dropped. We defined so called S-functions on a transformed state space, instead of the space of locations. The above modifications allowed us to reduce the space of possible locations of parts, to make the algorithm simpler and more efficient. We made it possible (by releasing normality assumption) to improve accuracy via specification of distributions, which better fit to data.

Other contributions came from the need to cope with real world data in (close to) real time. We considered a rather big set of simple appearance models and their combinations. We defined probability of an image region given a location of a body part in a unified probabilistic way for all appearance models. The log-normal family was taken as a family of distributions over dissimilarities. We approached learning of model parameters in a rigorous way. Mixing and correction coefficients were used to improve accuracy. We introduced a procedure of adjusting sizes of rectangles modeling parts according to training data. Moreover, a variable width template was defined, which we believe can improve accuracy and keep running time the same. We also used three view-specific models to improve the results.

We split our algorithm in two parts: precomputation of appearance probabilities and inference-sampling procedure. This allows for significant optimization. Some of optimization techniques were implemented, some were only outlined and remain for future work.

Finally, we performed different experiments to select the appearance model and the way it is parameterized. We considered pictorial structures as a model for pose recognition and as a classifier between presence and non-presence of a person. Results were reported using a number of different charts and figures.

6.2 Conclusions and Future Work

Pictorial structure is a powerful probabilistic model. One, however, should make many simplifying assumptions in order to make computation feasible. First of all we assume that appearances of parts are uncorrelated. By doing so we disregard some sources of important information, for example, symmetry in clothing. Secondly, we work with a tree structured model. This does not take into account many important correlations. Moreover, one is made to use very simple appearance models, if he/she wants to achieve close to real execution time.

The above mentioned assumptions introduce modeling errors. This makes the MAP solution a rather weak candidate. We have also shown this in our experiments. Thus restricting ourselves to sampling from posterior is a well-taken approach. In this case, though, we get another problem of selecting the best hypothesis. We have shown that it is enough to draw a small number of samples to obtain good pose estimate. Therefore, we may say that pictorial structures allow to reduce a huge set of possible poses to a small subset.

The above conclusion is true for images, where just one person is fully present. Thus a splitting of image or human detection algorithm is required as a preceding component. After samples have been drawn, it would be beneficial to analyze these or selected number of samples and to prune away erroneously positioned body parts.

We may also try to detect those missing parts afterwards. We believe that the need of many other preprocessing and supplementary algorithms is one of the main barriers on the way to using pictorial structures for pose recognition.

We see two main areas for improvement of our algorithm. Firstly, one may try to use a more complex model than tree structured one. This would, however, lead to other problems: long execution time and need of large training data set, in order to get reliable parameter estimates. Secondly, one may try to improve appearance models. Appearance models may be improved by incorporating responses from face or limb detectors. We believe this is a more promising approach, at least in the case of real world cluttered scenes.

Finally, we would like to note that a set of samples may not only be searched for the best one, but also used to build a new hypothesis. We believe that the following approach is worth investigation. We first cluster lower limbs and select instances minimizing in-cluster distances for each cluster. Next for each such instance we take a corresponding upper limb from the sample of the given lower limb instance. Now we consider all possible configurations of the above selected full limbs and leave only feasible ones. In the end we search for head and torso.

Bibliography

[1] Wikipedia, the free encyclopedia.

[2] F. Aherene, N. Thacker, and P. Rockett. The Bhattacharyya metric as an absolute similarity measure for frequency coded data. *Kybernetika*, 32(4):1–7, 1997.

[3] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *International Joint Conference on Artificial Intelligence*, pages 659–663, 1977.

[4] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–714, 1986.

[5] G. F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2-3):393–405, 1990.

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient matching of pictorial structures. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 66–73, 2000.

[7] P. F. Felzenszwalb and D. P. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell Computing and Information Science, 2004.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

[9] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92, 1973.

[10] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

[11] D. M. Gavrila and L. S. Davis. 3D model-based tracking of humans in action: A multi-view approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–80, 1996.

[12] D. M. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. In *IEEE International Conference on Computer Vision*, pages 87–93, 1999.

[13] E. J. Gumbel, J. A. Greenwood, and D. Durand. The circular normal distribution: Theory and tables. *Journal of the American Statistical Association*, 48:131–152, 1953.

[14] E. Haines. *Graphics Gems IV.* Academic Press, 1994.

[15] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[16] X. Lan and D. P. Huttenlocher. Beyond trees: Common-factor models for 2D human pose recovery. In *IEEE International Conference on Computer Vision*, pages 470–477, 2005.

[17] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *European Conference on Computer Vision*, pages 69–81, 2004.

[18] T. P. Minka. Pathologies of orthodox statistics.

[19] G. Mori, X. Ren, A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 824–831, 2004.

[20] J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, 1988.

[21] J. O. Ramsay. Maximum likelihood estimation in multidimensional scaling. *Psychometrika*, 42(2):241–266, 1977.

[22] J. O. Ramsay. The joint analysis of direct ratings, pairwise preferences, and dissimilarities. *Psychometrika*, 45(2):149–165, 1980.

[23] X. Ren, A. C. Berg, and J. Malik. Recovering human body configurations using pairwise constraints between parts. In *IEEE International Conference on Computer Vision*, pages 824–831, 2005.

[24] T. J. Roberts, S. J. McKenna, and I. W. Ricketts. Human pose estimation using learnt probabilistic region similarities and partial configurations. In *European Conference on Computer Vision*, pages 291–303, 2004.

[25] R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. In *European Conference on Computer Vision*, pages 700–714, 2002.

[26] A. W. van der Vaart. *Asymptotic Statistics.* Cambridge University Press, 1998.

[27] W. M. Wells. Efficient synthesis of Gaussian filters by cascaded uniform filters. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 8(2):234–239, 1986.

[28] J. Whittaker. *Graphical models in applied multivariate statistics.* John Willey and Sons Ltd, 1990.

[29] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 2007.

[30] J. Zhang, R. Collins, and Y. Liu. Bayesian body localization using mixture of nonlinear shape models. In *IEEE International Conference on Computer Vision*, pages 725–732, 2005.