Information and Computing Sciences
Faculty of Science, Utrecht University

Master of Science Thesis

Pose Estimation in Video

Author:
Elena Alexandra Ursu
Student number:
ICA-3730468

Supervisors:
Dr. Robby Tan
Dr. Ir. Nico van der Aa

Utrecht, August 2013

My acknowledgements go to Nuffic – the Netherlands organisation for international cooperation in higher education – for awarding me the Huygens scholarship which allowed me to pursue my master's abroad. It proved to be both an academic and a character building experience and I am grateful for the chance I had to study in this beautiful country.

I would like to thank my supervisors for their outstanding dedication towards their students, which was an important factor that helped me deliver this thesis. To Dr. Nico van der Aa for his valuable help with revising this document and for always acknowledging my achievements, and to Dr. Robby Tan for consistently trying to stimulate my independent thinking.

I have numerous friends, both from Utrecht and from home, to whom I would like to thank for their encouragement, for the adventures, for their programming expertise and for teaching me how to ride a bicycle at an adult age. I would like to mention Dana Ivan and Mariela Ramirez Goio – your wisdom and friendship were my survival guide through these years.

Finally, I express my gratitude to my family for always encouraging and caring for me, and to my partner for patiently supporting me from afar.

Abstract

Human pose estimation in video has numerous applications, such as human activity analysis, automatic surveillance, human-computer interaction and markerless motion capture. It is challenging because of the kinematic structure of the human body and the variety of possible human poses, the endless appearance options caused by clothing and, finally, due to background clutter that can look like parts in the human body and confuse the system.

Current methods in human pose estimation either focus on specific situations, such as pedestrians or laboratory controlled motions, or sacrifice accuracy in favour of coping with videos containing any type of human activity. What we will show in this thesis is an improved system built upon the method of [Ramanan et al., 2007], which models a person's body configuration as a puppet of rectangles. The system first analyses all the frames from a video to find a specific pose from which it learns the appearance of the person to be tracked. Then it processes the video to detect the person in any possible pose.

We analysed the robustness of the original method by comparing pose estimations with labelled ground truth. We challenged the authors' claim that one set of parameters can fit multiple videos, which remains an open issue. Then, we extended the original method by including temporal information using two different types of motion models, which improved the tracking results. According to our qualitative evaluation of side-by-side tracking sequences, the new extensions resulted in more stable and accurate detections throughout time and are able to solve some challenging situations which arise when the motion is fast or body parts resemble each other. We found that the system performs poorly when detecting arms, due to their size, which remains the main problem to be solved in future work.

Contents

1 Introduction

Human pose estimation in video is a promising topic that can enable numerous applications, such as human activity analysis, automatic surveillance, human-computer interaction and markerless motion capture [Brubaker et al., 2010, Sigal]. It is a challenging task, mainly because of the kinematic structure of the human body and the variety of possible human poses, the endless appearance options caused by clothing and, finally, due to background clutter that can look like parts in the human body and confuse the system. Other challenges include and are not limited to: the variety of human motions (from pedestrian walking to sports activities), the number of people in the scene (which can be crowded with either a large number of distinctly looking people, or with a sports team wearing the same clothes), camera motion, point of view and quality of the recorded material. Solving these issues can be done either by knowing the nature of a particular type of input beforehand, or by working on improving the robustness of the system with respect to the variety of inputs.

Given all the above stated challenges in the broad problem of human pose estimation in video, we address a particular type of associated applications - automatic sport analysis from video. Analysing videos is a low cost, nonintrusive procedure. Another advantage is that numerous recordings of sports activities exist and can be thus interpreted. Research in the field of sports analysis from video (such as [Li et al., 2010, 2006, Han et al., 2005, Ekin et al., 2003, Zhang et al., 2012, Ghanem et al., 2012]) provided systems that can automatically detect and classify athlete's actions, which is useful for video indexing and retrieval or for summarization of sports matches. Other uses include performance improvement and coach assistance based on kinematic measurements, real life action comparisons with actions in video, or strategy revealing and understanding based on player formations and trajectory patterns.

Different sports and types of video data face different challenges. For example, automatic analysis of large court sports, such as football or American football, needs to cope with camera movements (pan, tilt, zoom), blurred or low-resolution capture of distinct players due to the far-away view of the camera and appearance similarity between players belonging to the same team [Zhang et al., 2012, Ghanem et al., 2012]. In individual sports, like diving and athletic jumping [Li et al., 2010, 2006] the background clutter and dynamics pose a challenge on the person segmentation. Regarding the types of video data, two examples of associated challenges are to discriminate between commercial sequences and actual sports sequences in a television broadcast [Han et al., 2005] and to identify cuts and transitions in the video material [Ekin et al., 2003].

Within the scope of human pose estimation and with the above applications and their associated challenges in mind, we aim to estimate the human body pose from videos of individual athletes. Our interest is to determine the best extent to which we can accurately approximate the 2D body configuration, given a video sequence from a single colour camera. We choose as input videos of gymnasts performing at the balance beam, in front of both cluttered and uncluttered background. The desired output is the body pose detection at every frame, consisting of the exact localisation of major body parts.

Because we aim at finding precise pose detections in videos that have been recorded in the past, we do not restrict ourselves to real-time methods. The basic employed method is *tracking by model-building and detection*, which means that the system first learns a model of the people to be tracked in the video sequence, then uses this model to detect them frame by frame. Hence, the system consists of two parts: the model building module and the detection module.

Possible applications of precise pose estimation in individual sports such as gymnastics include and are not limited to: athlete's time progress analysis or performance comparison with other athletes, visual cues highlighting on replays or markerless motion capture for sports motion databases. Once the goal of estimating major body parts is achieved, one could go further to more complex models of the human body, which could result in a system that automatically evaluates athlete's performances in formal competitions.

For this purpose, we start from the excellent work of [Ramanan et al., 2007], which employs *pictorial structures* as a successful technique for the task of estimating the human pose as a configuration of body parts. We investigate how their system reacts to different inputs, in order to investigate whether the method is applicable to material coming from different video archives. Then we extended it to improve the tracking quality. We plan to do so by exploring the temporal component, or how the detection in the current frame can help the detection in the next frame.

The layout of this thesis report is as follows. We continue this chapter by giving an overview of the practical application, including constraints and implementation details. The Related work chapter gives a short survey of relevant papers in the field and the motivation for our choice of the main reference papers. The Pictorial structures chapter explains the statistical framework which constitutes the skeleton of the tracking system. The Model building chapter explains the theory behind [Ramanan et al., 2007]'s stylised pictorial structure which enables us to learn the appearance of the people in the video. The Detection chapter explains [Ramanan et al., 2007]'s single frame pictorial structure algorithm, which is an approach to detect people from still images, that we will deploy as part of the general tracking solution. The Detection chapter also introduces our contributions: two variants of expanding the graphical model behind the pictorial structure algorithm to include information from the previous frame as well, under inference techniques and implementation-wise constraints. Chapter Experimentation shows our studies on the system robustness and on the newly implemented temporal graphical model variants. Finally, we conclude with the Conclusions and future work based on our findings.

System description and constraints

The application takes in a video sequence of a person and outputs a video containing the person's pose detection depicted with solid line coloured rectangles corresponding to each body part. The system works in two phases:

1. Model building - processes each frame in search for a *stylised pose*, then learns the appearance parameters for a person. This module is explained in Chapter 4.

2. Detection - uses the learnt appearance parameters to detect *general poses* in each frame. This module is explained in Chapter 5.

The constraints of our implementation are:

1. The system learns the appearance model and detects the pose for a single person in the video.
2. Both modules of the system require that the scale of the person to be detected is known and that it remains approximately constant throughout the video.
3. The body parts can easily be approximated by rectangles. The system will not detect people wearing skirts, dresses or loose clothes.

The chosen videos of gymnasts performing the balance beam conveniently cope with these constraints. First, the videos contain only one gymnast at the time. Next, the gymnast movement is constrained mostly on the horizontal axis, so that the scale remains approximately constant. Last, the lean body constitution and the tightfitting sports suit allows for good approximations of individual body parts with rectangles.

System implementation

Our implementation of the human tracking system is based on [Ramanan et al., 2005]'s Matlab implementation made available at `http://www.ics.uci.edu/~dramanan/papers/pose/index.html`. Our implementation is written in C++ and makes use of the OpenCV 2.4.3 library [Bradski and Kaehler, 2008] for computer vision functions and ALGLIB [alg, 2013] for algebraic functions. The design is object oriented, each of the two modules being written as separate classes. Besides the video sequence, the system also requires two settings files (one for each module), containing information about the person's size and various thresholds that we will explain in the dedicated chapters.

2 Related work

[Sigal] defines human pose estimation as “the process of estimating the configuration of the body (pose) from a single, typically monocular, image”. Estimation of human poses over time is a different problem and can be referred to as human motion analysis [Poppe, 2007]. A great deal of research exists on human pose estimation in *still images*. This research usually focuses on learning sophisticated deformation models and appearance dependencies from labelled datasets. These models are then applied to still images (sometimes using advanced limb detectors). The person(s) in these images is not necessarily someone who has been seen in the training data and might appear in any pose or wear any type of clothes.

Pose estimation in *video* can be 2D or 3D. The latter can be aided by multiple camera footage and motion databases. On the other hand, 2D pose estimation in video only requires a single camera captured video as input. We wish to focus on this type of input. In this chapter we approach human pose estimation in video from the tracking perspective and select a number of papers that deal specifically with 2D pose estimation in video for comparison.

[Forsyth et al., 2006] identify the scale as the most important variable of the human tracking problem. They differentiate between the following three levels of the scale of the people in video frames:

1. **Coarse scale**. People occupy small patches in the frame, which allows only for global assumptions of a person’s position, but not about the positions of individual body parts. Examples of applications that use such videos as input are applications that analyse patterns of activities in crowds or in large open spaces ([Stauffer and Grimson, 2000]).
2. **Medium scale**. People can be regarded as blobs with motion fields. The task of tracking the entire body as a single object is called human tracking or detection [Poppe, 2007]. Possible applications that deal with this type of input videos might concern situations such as a TV broadcast of a team sport or surveillance cameras overseeing a subway station, a traffic intersection, etc. ([Breitenstein et al., 2011])
3. **Fine scale**. All body parts are visible and the application aims at finding the body configuration or the body pose. This task can come under different names: kinematic tracking [Forsyth et al., 2006] or human motion analysis [Poppe, 2007].

We focus on the last case, where the people are clearly visible in the video sequences and their scale allows for individual identification of distinct body parts. The most successful approach so far towards statistically modelling the human body as a collection of parts are the *pictorial structures* [Fischler and Elschlager, 1973]. We discuss these in Chapter 3.

Pose representation Various representations of the body parts are possible. For example, [Niebles et al., 2008, 2010] aim at finding volumes (or contours) of the human body in an application that does not require any assumption about the appearance or number of people in the videos and is targeted at real world data, such as low-resolution videos from the internet, depicting a large range of human motions. [Huo and Hendriks, 2012] focus on the upper body (motivated by human-computer interaction applications) and show their tracking results as rectangles for the torso and the head and as lines (skeleton representation) for the arms. [Ramanan, 2007, Andriluka et al., 2008] detect full body humans and show the poses as (line) skeletons drawn on the colour frames. [Ramanan et al., 2007, 2005] represent people as *puppets of rectangles*, where a rectangle is matched to the image evidence corresponding to each major body part (torso, upper and lower legs and arms and the head).

Video characteristics Aside from the representation, we also bring the character of the videos into discussion. Researchers tend to study specific situations and solve problems that arise in each case. [Andriluka et al., 2008] focus their efforts on long-term occlusion in the case of pedestrians. [Niebles et al., 2008, 2010] sacrifice per-pixel accuracy and build a fully automatic system that requires no manual initialisation and no a priori knowledge about the number of people in the scene or about their appearance and works with arbitrary videos. [Huo and Hendriks, 2012] focus on estimating the positions of occluded parts and test their system on laboratory captured upper body motions that are designed for games interaction.

Unlike the previous, [Ramanan et al., 2007, 2005] build a more general framework, which allows for different inputs in terms of activity types, number of people in the video, source of the video and type of setting, such as sports activities, outdoor activities, movie scenes and people walking in a park. While covering all these situations, the tracking representation (puppet of rectangles) remains highly detailed, aiming at identifying individual body parts.

Constraints Tracking systems require a degree of knowledge about the people to be found. This knowledge is referred to as a *model* and can be either *generic* or *specific*. A generic model will describe the human body and will be applicable to any person. Such models will usually contain information about the shape and the configuration of the human body. The configuration of the human body does not only encompass the hierarchy of parts, but also the range of possible relative positions, which depend on the type of actions that the person performs in the video. The wider the range of actions that the system needs to cover, the less restrictive the model will be. For this reason, for the generic model to succeed, it will need stronger knowledge about some characteristics of the people in the video, which will either restrict the applicability of the system with respect to the type of input, or will make the system require some sort of manual intervention or initialisation.

For example, [Andriluka et al., 2008] develop an expressive kinematic limb model based on the characteristics of the walking cycle for pedestrians. Also, [Hogg, 1983] describes very specific positional, movement and posture constraints for walking. [Niebles et al., 2008, 2010] do not require knowledge about the type of motion or about the person, but only deliver an approximate volume of the person's position and configuration in the video. [Huo and Hendriks, 2012] require a controlled laboratory setup, where the cameras are synchronized, their position is known and the subjects are asked to perform a specific pose to initialise the system. Such restrictions, however, make it possible for the system to accurately deal with inter-person and self occlusion and show the position of the occluded body parts. [Ramanan et al., 2007] learn the colour appearance models of the people (specific model) in the video offline (in the first stage), then use these models to detect each person at limb-level accuracy (in the second stage). This procedure is also called tracking by model-building and detection. There is no restriction on the type of action performed in the video and the body configuration restrictions are designed as general focusing mainly on making sure that the pairs of body parts that are normally connected in the human body also have close positions in the video frames.

Our employed method We chose to use [Ramanan et al., 2007]'s work as a starting point, because their method was tested on videos from various sources, containing a wide range of activities and showed promising detection results at body part level accuracy. The full body puppet of rectangles representation is suitable for our goal of precisely tracking all the major body parts. The fact that the method was tested on videos from unconstrainted sources shows potential for our goal of being able to process diverse archives of sports videos.

3 Pictorial structures

3.1 Pictorial structures in static images

A pictorial structure [Fischler and Elschlager, 1973] is an object representation consisting of a collection of parts connected in a deformable configuration. The parts encode how well the image patch matches the visual data according to a visual model, while the connections describe the agreement between the relative parts positions and the deformable model. [Felzenszwalb and Huttenlocher, 2005] define the statistical framework for matching a *static* image to a pictorial structure through the following concepts:

$u = \{u_1, \ldots, u_N\}$	appearance model parameters, where N is the number of parts
$c = \{c_{ij} \vert (v_i, v_j) \in E\}$	the deformable model between connected edges in set E
$\theta = (u, E, c)$	set of object model parameters
I	the image
$P^{1:N}$	the object configuration
$P\left(I \vert P^{1:N}, \theta\right)$	the likelihood of seeing an image given the object's configuration
$P\left(P^{1:N} \vert \theta\right)$	the prior probability that the object is in a particular configuration
$P\left(P^{1:N} \vert I, \theta\right)$	the posterior distribution of the object configuration given the model θ and the image I.

The problems that can be solved within this statistical framework are the following:

1. *MAP estimation*, which finds the configuration $P^{1:N}$ with maximum posterior probability,
2. *Sampling from the posterior*, which finds several good matches of the object model to the image, instead of only the best one and accounts for imprecise models (for example, due the large variety of deformations in the human body),
3. *Model estimation*, which learns the model θ from training data using maximum likelihood estimation.

[Felzenszwalb and Huttenlocher, 2005] demonstrate that for articulated models (such as models of the human body), where the constraints between parts are relatively loose and allow for overlapping, generating multiple samples from the posterior distribution gives a good estimate of the object configuration. [Ramanan et al., 2007] feed these samples into a mode finding procedure to obtain the final body configuration.

Using Bayes' rule, the posterior $P\left(P^{1:N}|I,\theta\right)$ is given by:

$$P\left(P^{1:N}|I,\theta\right) = \frac{P\left(I|P^{1:N},\theta\right)P(P^{1:N}|\theta)}{P(I|\theta)} \tag{3.1.1}$$

and results in the following form for the posterior distribution of the object configuration:

$$P(P^{1:N}|I,\theta) \propto P\left(I|P^{1:N},\theta\right)P\left(P^{1:N}|\theta\right). \tag{3.1.2}$$

Assuming conditional independence between the parts appearance model, the *likelihood* of seeing an image given the object's position becomes:

$$P\left(I|P^{1:N},\theta\right) = P\left(I|P^{1:N},u\right) \propto \prod_{i=1}^{N} P(I|P^i,u^i). \tag{3.1.3}$$

This assumes that the body parts occupy different patches of the image, meaning they do not overlap. [Felzenszwalb and Huttenlocher, 2005] solve this model impreciseness by sampling from the posterior distribution, which gives several possible matches instead of just the best one, and finally selecting one sample.

The *prior* distribution is described by a tree-structured Markov random field with vertices V and edge set E as:

$$P\left(P^{1:N}|\theta\right) = \frac{\prod_{(v_i,v_j)\in E} P\left(P^i,P^j|c_{i,j}\right)}{\prod_{v_i\in V} P\left(P^i|\theta\right)^{deg_{v_i}-1}}, \tag{3.1.4}$$

where deg_{v_i} is the number of parts connected to part i, $P\left(P^i|\theta\right)$ models the absolute position of part i, $P(P^i,P^j|c_{i,j})$ models the relative configuration of parts i and j and $c_{i,j}$ are the deformable model parameters for the connection between parts i and j. As there is no need to model a preferred absolute position, $P\left(P^i|\theta\right)$ can be set to 1. Therefore, the previous equation becomes:

$$P\left(P^{1:N}|\theta\right) = \prod_{(v_i,v_j)\in E} P(P^i,P^j|c_{i,j}). \tag{3.1.5}$$

For the remaining of this report, we will use [Ramanan et al., 2007]'s notation, where C is the appearance model, the image likelihood is $P\left(I|P^i,C^i\right)$, and each part's configuration depends only on the parent body part, such that the prior is written as $P(P^i|P^{\pi(i)})$, where $\pi(i)$ is the parent of part i. With this convention and the likelihood detailed in equation (3.1.3) and the prior detailed in equation (3.1.5), equation (3.1.2) becomes:

$$P(P^{1:N},I|C^{1:N}) \propto \left(\prod_{i=1}^{N} P\left(I|P^i,C^i\right)P(P^i|P^{\pi(i)})\right). \tag{3.1.6}$$

3.2 Temporal pictorial structures

[Ramanan et al., 2007] approach the tracking in video problem using a Hidden Markov Model (HMM), where the hidden states are the poses to be estimated and the observations are the video frames. Figure 3.1 shows the pictorial structure graphical model for a full body pose at frame t. Figure 3.2 shows the temporal graphical model, in which we only selected two connected body parts for clarity. The arrows between connected body parts represent the relative configuration probability $P(P_t^i|P_t^{\pi(i)})$, the arrows between the body parts and the image observations represent the image likelihood $P\left(I_t|P_t^i,C^i\right)$, while

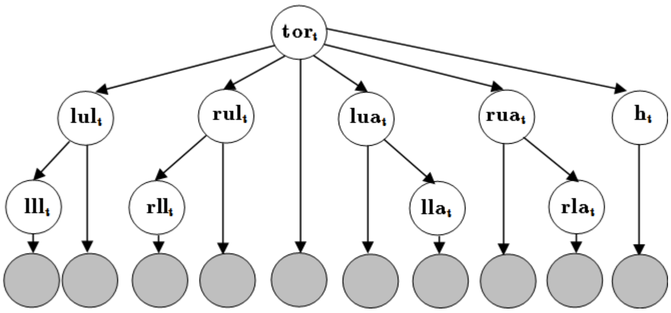


Figure 3.1: Full body pictorial structure graphical model for frame t. Body part nodes are depicted in white: **tor**so, **h**ead, **l**eft **u**pper **l**eg, **r**ight **u**pper **l**eg, **l**eft **u**pper **a**rm, **r**ight **u**pper **a**rm, **l**eft **l**ower **l**eg, **r**ight **l**ower **l**eg, **l**eft **l**ower **a**rm, **r**ight **l**ower **a**rm. Gray nodes represent the **image observations** $I(P^i)$.

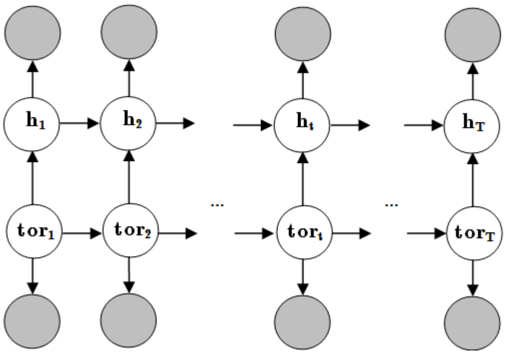


Figure 3.2: Temporal graphical model for a sequence of T frames. Horizontal arrows represent the motion model.

the arrows between same body part nodes at consecutive frames represent the *motion model* $P(P_t^i|P_{t-1}^i)$.

The posterior distribution for the temporal pictorial structure also includes the motion model prior:

$$P(P_{1:T}^{1:N}, I_{1:T}|C^{1:N}) \propto \prod_{t=1}^{T}\prod_{i=1}^{N} P(P_t^i|P_{t-1}^i)P\left(I|P^i, C^i\right) P(P^i|P^{\pi(i)}). \tag{3.2.1}$$

As inference on the full graphical model is difficult due to loops and the large state space, it is convenient to ignore loops and pass local messages. The message passing procedure is explained in the sampling from posterior algorithm (see Section 3.3).

In this thesis, we show the simplified (single frame) graphical model implemented by [Ramanan et al., 2007], and extend it to two more complex variants, including temporal information.

Image likelihoods

We practically calculate the image likelihood as a function of a *singleton potential*:

$$P(I|P^i, C^i) \propto \Phi_i(x_i). \tag{3.2.2}$$

The singleton potential $\Phi_i(x_i)$ represents a score for the match between an image patch (or a candidate x_i) and a part template and is usually the result of a convolution between a template and a feature image. [Ramanan et al., 2007] define templates and features for each of the modules of the tracking system:

- In the model building module, rectangular filters are used to detect body parts on the edges of the input frame (described in Section 4.1);
- In the detection module, 2D Gaussian filters of rectangular shape are used to detect body parts on areas of the input frame that resemble the previously learnt colour appearances (described in Section 5.1).

Kinematic constraints

The prior which describes the conformity with the deformable model is calculated depending on a *pairwise potential*:

$$P(P^i|P^{\pi(i)}) \propto \Psi_{i,\pi(i)}\left(x_{i,\pi(i)}\right). \tag{3.2.3}$$

In the above equation we use $x_{i,\pi(i)}$ for the pair of body part candidate and parent candidate $\left(x_i, x_{\pi(i)}\right)$. We will further use the term *kinematic constraints* when referring to the deformable model. These constraints can be seen as falling into one of the following categories:

1. Hard constraints - which *validate* $(P(P^i|P^{\pi(i)}) = 1)$ or *discard* $(P(P^i|P^{\pi(i)}) = 0)$ a body part, usually based on the distance between body parts;
2. Soft constraints - which set a *preference* for specific configurations, by setting the value for potential $\Psi_{i,\pi(i)}\left(x_{i,\pi(i)}\right)$ according to a function based on one or both the distance and angle between body parts.

The relative position between body parts plays a key role in calculating the pairwise potential $\Psi_{i,\pi(i)}\left(x_{i,\pi(i)}\right)$. This position is calculated between hinge points of the body parts, as shown in the figure below.

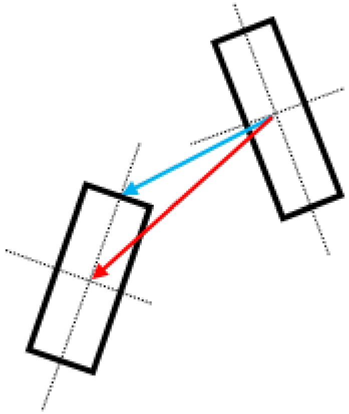


Figure 3.3: Relative positions between different hinge points: blue - centre to top, red - centre to centre.

Motion models

We calculate the probability that represents the motion model as a function of a potential depending on the coordinates of a body part in two consecutive frames:

$$P(P^i_t|P^i_{t-1}) \propto \Psi_{i,prev\_i}\left(x_{i,prev\_i}\right). \tag{3.2.4}$$

In the equation above, we use $x_{i,prev\_i}$ for the pair of body part candidate in the current frame and the same body part in the previous frame $\left(x_i, x_{prev\_i}\right)$.

We describe two possibilities for the motion model: the bounded velocity model [Ramanan et al., 2007] and the Gaussian noise [Brubaker et al., 2010]. Due to the nature of these two models, which consist of normalized values between 0 and 1, the relationship between the probability and the potential in equation (3.2.4) becomes the identity function.

[Ramanan et al., 2007] use a bounded velocity motion model. This means that if a body part in the current frame is within a certain distance away from its position in the previous frame, then it is assigned the highest probability $P(P_t^i|P_{t-1}^i) = 1$, otherwise it is assigned the lowest probability $P(P_t^i|P_{t-1}^i) = 0$. The distance is calculated between the centres of the two body part detections as follows:

$$P(P_t^i|P_{t-1}^i) = \begin{cases} 1, & \|(x_t^i, y_t^i) - (x_{t-1}^i, y_{t-1}^i)\| \leq v_{max} \\ 0, & \|(x_t^i, y_t^i) - (x_{t-1}^i, y_{t-1}^i)\| > v_{max} \end{cases} \tag{3.2.5}$$

The issues that arise from this motion model are the following:

- v_{max} should be activity specific, which implies that we either know the type of motion a priori, or we set a general value in turn for a lower accuracy;
- the camera should be still, so that v_{max} only describes the person's motion and does not need to compensate for camera movement as well;
- v_{max} can take a general value for all body parts, or individual values for different body parts, requiring fine tuning for the latter option.

We also look at another motion model, where noise is added to the previous pose, in order to determine the current one [Brubaker et al., 2010]:

$$P_t^i = P_{t-1}^i + \eta. \tag{3.2.6}$$

Here, η is a *process noise* that can be modelled with a Gaussian $\eta \sim \mathcal{N}(0, \Sigma)$. The motion model prior results in:

$$P(P_t^i|P_{t-1}^i) = \mathcal{N}(P_t^i; P_{t-1}^i, \Sigma), \tag{3.2.7}$$

where $\mathcal{N}(P_t^i; P_{t-1}^i, \Sigma)$ is the Gaussian distribution function centred at the previous part pose and covariance Σ, evaluated at the current part pose. Aside from the centre coordinates, we can also model the variation of the body part orientation θ as process noise. The above equation becomes:

$$P(P_t^i|P_{t-1}^i) = \mathcal{N}(x_{t-1}^i; x_t^i, \sigma_x)\mathcal{N}(y_{t-1}^i; x_t^i, \sigma_y)\mathcal{M}(\theta_{t-1}^i; \theta_t^i, k), \tag{3.2.8}$$

where $\mathcal{M}(\theta_{t-1}^i; \theta_t^i, k)$ is the Von Mises or circular normal distribution [Gumbel et al., 1953],

$$\mathcal{M}(\theta_{t-1}; \theta_t, k) \propto e^{k\cos(\theta_t - \theta_{t-1})}. \tag{3.2.9}$$

The same issue as for the bounded velocity motion model arises in this case: tuning the parameters σ_x, σ_y and k.

In comparison, the Gaussian noise motion model gives a preference over the candidates and does not simply accept or reject candidates, as the bounded velocity motion model does.

The effect of the motion model is that it smoothes the tracking result and eliminates the candidates which are too far from the previous detection. But, the constraint on this model is that the previous detection should always be correct. When this fails, it is possible that the body track will deviate from the correct position, in time.

3.3 Sampling from the posterior distribution of body configurations

Given that the structure of the human body can be modelled as a tree (see Figure 3.1), it is convenient to use belief propagation [Yedidia et al., 2003] to calculate the posterior at each node. The root (torso) posterior will be proportional to the product of the local evidence $P(I|P^r, C^r)$ and all the messages coming from the root's children:

$$P(P^r, I|C^r) \propto P(I|P^r, C^r) \prod_{j \in C(r)} m_{j,r}(x_r), \tag{3.3.1}$$

where the *messages* $m_{j,i}(x_i)$ are calculated according to the *message update rule*:

$$m_{j,i}(x_i) \propto \sum_{x_j} \left(P\left(I|P^j, C^j\right) P\left(P^j|P^{\pi(j)}\right) \prod_{k \in C(j)} m_{k,j}(x_j) \right). \tag{3.3.2}$$

The messages are calculated starting at the leaves, then propagate through the nodes to the root. Once the root posterior is known, a sample is obtained (see the sampling procedure below) from this distribution. With the sampled root (denoted as $\pi(i)\_idx$), the children samples will be obtained from the following posterior distribution:

$$P\left(P^i, I|P^{\pi(i)\_idx}, C^i\right) \propto P\left(I|P^{\pi(i)\_idx}, C^{\pi(i)\_idx}\right) P\left(P^i|P^{\pi(i)\_idx}\right) \prod_{j \in C(i)} m_{j,i}(x_i). \tag{3.3.3}$$

This procedure is continued until the leaf nodes. It is necessary to calculate the messages only once, then the sampling can be done multiple times. We detail this algorithm for each employed tree model, respectively, in the following sections.

Sampling procedure
We explain the employed sampling procedure in the following.

Probability density function A *probability density function* (PDF), also called density of a continuous random variable, is a function that describes the relative likelihood for this random variable to take on a given value.

A PDF satisfies $P(x \in B) = \int_B P(x)dx$, and the normalization condition $P(-\infty < x < \infty) = \int_{-\infty}^{\infty} P(x)dx = 1$ [Weisstein, a].

Distribution function The cumulative distribution function (CDF) describes the probability that a variable X takes on a value less than or equal to a number x. The relation with a continuous PDF is the following:

$$D(x) = P(X \leq x) = \int_{-\infty}^{x} P(\xi)d\xi, \tag{3.3.4}$$

namely, the PDF, when exists, is the derivative of the CDF. The relation with a discrete probability $P(x)$ is the following:

$$D(x) = P(X \leq x) = \sum_{X \leq x} P(x). \tag{3.3.5}$$

We wish to generate numbers distributed as $P(x)$. For this, we use a random number generator that gives uniformly distributed values $y \in (0, 1)$. The x variable distributed as $P(x)$ is obtained by inverting the CDF, as $x = D^{-1}(y)$ [Weisstein, b]. Figure 3.4 shows the CDF graph for a normal distribution and the corresponding variable x to a randomly generated value y.

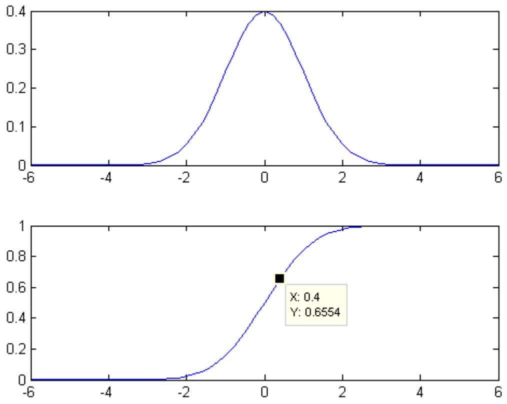


FIGURE 3.4: CDF of the normal Gaussian distribution

4 Model building

In this chapter, we explain the model building module, which is responsible with finding a stylised pose in a video sequence, then choosing the frame that contains the best pose and learning the body parts colour appearance model parameters from that particular frame. The module pipeline is shown in Figure 4.1. We break down the components of this pipeline and explain them in their dedicated sections from this chapter.

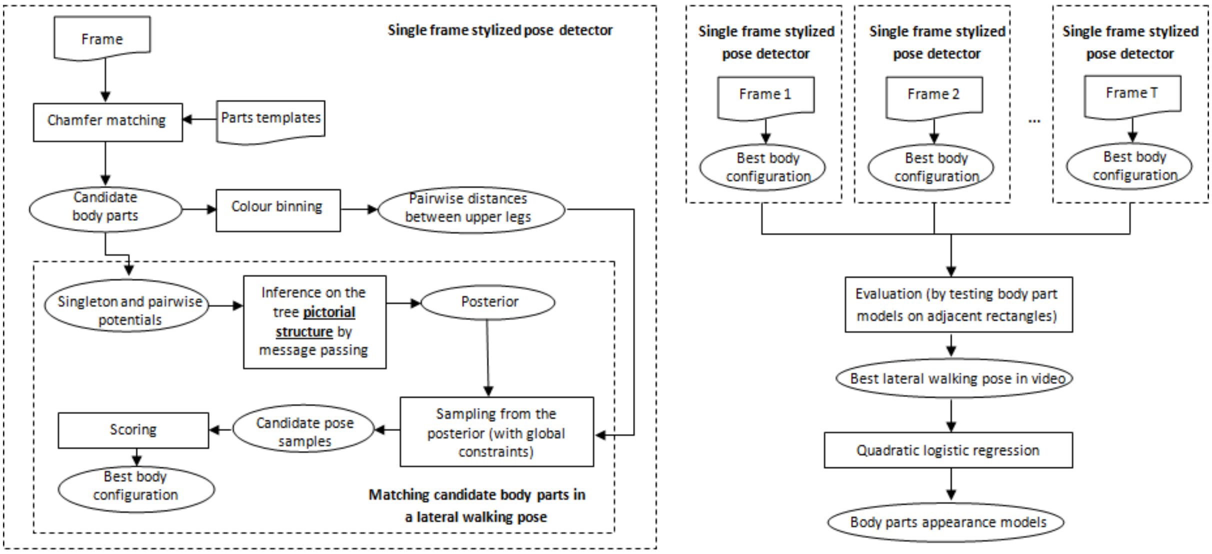

Single frame stylized pose detector
Frame
Chamfer matching
Parts templates
Candidate body parts
Colour binning
Pairwise distances between upper legs
Singleton and pairwise potentials
Inference on the tree pictorial structure by message passing
Posterior
Sampling from the posterior (with global constraints)
Candidate pose samples
Scoring
Best body configuration
Matching candidate body parts in a lateral walking pose
Single frame stylized pose detector
Frame 1
Best body configuration
Single frame stylized pose detector
Frame 2
Best body configuration
...
Single frame stylized pose detector
Frame T
Best body configuration
Evaluation (by testing body part models on adjacent rectangles)
Best lateral walking pose in video
Quadratic logistic regression
Body parts appearance models

FIGURE 4.1: Model building module pipeline.

4.1 Searching for body parts using generic templates

A reasonable assumption for the shape of the human body is that body parts are cylindrical. Then, their projection on an image can be approximated by rectangles [Felzenszwalb and Huttenlocher, 2005]. A rectangle is parameterised by its width, length, x and y centre coordinates and orientation θ. A main assumption of [Ramanan et al., 2007] is the fact that the scale of the person in the video is known. Knowing the width and length for each body part, the centre coordinates and orientation remain to be determined. The authors do this by chamfer matching (see Section 4.1.2) a body part template with the input frame.

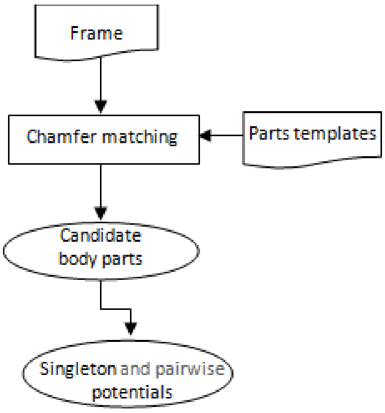


FIGURE 4.2: Model building pipeline: Searching for body part candidates.

This section corresponds to the pipeline components shown in Figure 4.2. The input to this module consists of the colour frame and the body part templates. The output of this module represents the body part candidates x_i and their associated singleton potentials $\Phi_i(x_i)$.

In the following we discuss the basis for finding patterns in images, then we elaborate on chamfer matching and on additional techniques.

4.1.1 Rectangular filters as part templates

Linear filtering is a strategy to find different image patterns. Linear filtering consists of replacing a pixel in an image with a weighted sum of the surrounding pixel values using a selected set of weights [Forsyth and Ponce, 2002]. This operation is:

1. *shift invariant* - meaning the output depends on the pattern in the image neighbourhood, rather than its position;
2. *linear* - meaning that the output of several images summation is the same as the summation of the individual outputs.

The process is also referred to as a *convolution*, represented by the following operation:

$$I^{'}(u,v) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(u-i, v-j)H(i,j) = I \star H(u,v)\,, \tag{4.1.1}$$

where I is the original image, $I^{'}$ is the filtered image, H is the *kernel* of the filter and (u,v) are pixel coordinates [Sanchez, 2011].

Filters can be used to find simple patterns in an image, because they give higher responses in the areas that look like the filters [Forsyth and Ponce, 2002]. [Ramanan et al., 2007] use rectangular templates, with known width and length for each body part. These filters are applied to the edge image of an input frame. In this form, part templates are colour invariant and only model the shape of the body part. Figure 4.6(c) shows a rectangular template for an upper leg rotated with 45°.

4.1.2 Chamfer matching

Matching is the problem of determining the similarity between predicted features (in our case, the body part templates) and image features (such as edges). *Chamfer matching* [Barrow et al., 1977] compares the shape of two collections of curves, by calculating a measure of similarity called *chamfer distance*. The chamfer distance between two sets of feature points is the mean of distances between each point in the template and the nearest image point. For two point sets $\mathcal{U} = \{u_i\}_{i=1}^{n}$ and $\mathcal{V} = \{v_j\}_{j=1}^{m}$, the formula for the chamfer distance is the following [Thayananthan et al., 2003]:

$$d_{cham}(\mathcal{U},\mathcal{V}) = \frac{1}{n} \sum_{u_i \in \mathcal{U}} \min_{v_j \in \mathcal{V}} \|u_i - v_j\| \tag{4.1.2}$$

To calculate this distance, we first introduce the *distance transform* of an image, then illustrate how this helps calculating the chamfer distance.

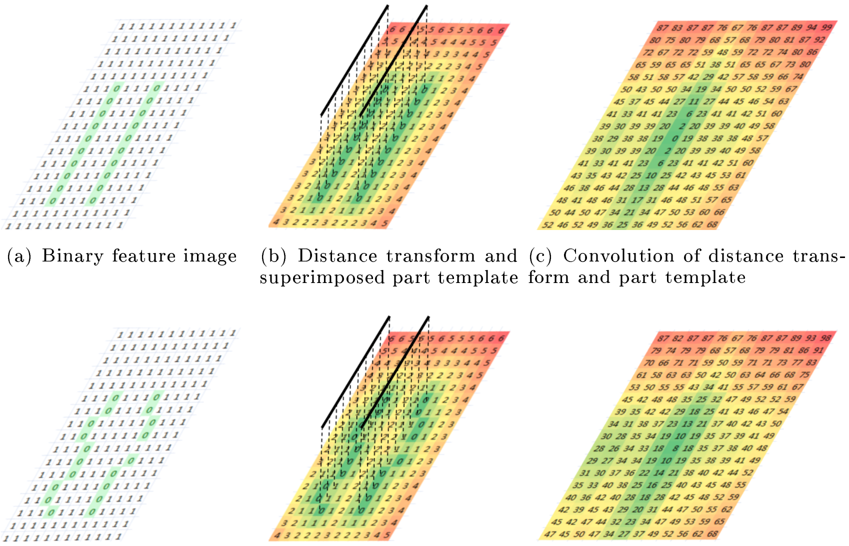
Distance transform

The distance transform of an image is an output image in which every pixel is labelled with its distance to the closest zero pixel in the original image [Bradski and Kaehler, 2008]. The zero pixels in the original image denote features, such as edges. Distance transforms are calculated using masks where each pixel is labelled with the distance between a pixel at that position and the centre of the mask.

Figures 4.3(a) and 4.3(d) show two binary feature images. These images could be the result of an edge detection process on a colour frame. Figures 4.3(b) and 4.3(e) were obtained by using the OpenCV function cvDistTransform with the Euclidean distance as a metric.

Chamfer distance

To calculate the chamfer distance, we superimposed a bar template (like the rectangle template used to detect body parts), depicted by the black lines in Figures 4.3(b) and 4.3(e). The chamfer distance is simply calculated by averaging all the pixel values underneath the template points. This is done for every pixel by translating the template above the feature image.



(a) Binary feature image

(b) Distance transform and superimposed part template

(c) Convolution of distance transform and part template

(d) Binary feature image

(e) Distance transform and superimposed part template

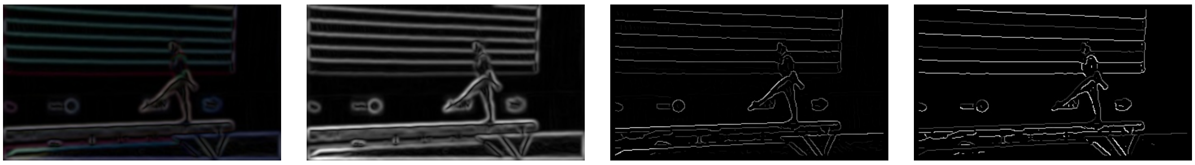
(f) Convolution of distance transform and part template

FIGURE 4.3: Chamfer matching

We can see that in Figure 4.3(c), where the image looks exactly like the template, the values in the centre of the image patch are lower than in Figure 4.3(f). This result can be obtained by convolving the distance transform with the template. Figures 4.3(c) and 4.3(f) resulted by using the OpenCV function cvFilter2D. The lowest value (in dark green) will give the score and the position of the centre of the patch that looks like the template. In Figure 4.3(c) we obtained a score of 0 for perfect matching, while in Figure 4.3(d) we illustrated some degenerate edges and obtained a score of 8 in the centre of the convolution result shown in Figure 4.3(f).

Edges and orientation cues

We saw that a feature image is needed for chamfer matching. Given that the search is performed to find shapes that look like the body part templates, the feature image will actually represent the edge detection of a video frame. To obtain the edges, first the x and y derivatives of the image are computed, by convolving the image with elongated Gaussian filters [Martin, 2003]. Then the gradient magnitude of the image (Figure 4.4(a)) is computed as the square root of the sum of squared derivatives. These operations are performed per separate colour channels. The maximum gradient magnitude is selected from the colour channels (shown in Figure 4.4(b)) and further processed by non-maximum suppression (explained next). By imposing a limit on the gradient magnitude level (zeroing the pixels under a certain threshold), we obtain the strong edges (Figure 4.4(c)).



(a) Gradient magnitude of RGB image

(b) Maximum gradient magnitude across channels

(c) Non-maximum suppression result

(d) Edge image labelled with orientation bins

FIGURE 4.4: Edge detection

To explore orientation cues, [Ramanan et al., 2007] label each edge pixel with the bin in which the gradient orientation falls (out of 24 bins). We show the labelled edge image in Figure 4.4(d), where the 24 orientations have been scaled to display a grayscale image. Then, only those edge pixels whose orientations either fall perpendicular to the rectangular template, either neighbour these orientations to the left and to the right, are selected.

An example is illustrated in Figure 4.5. For a template rotated 60° counterclockwise, with the bin numbering starting near the positive x axis in counterclockwise order, the selected edges will correspond to the following bins: 4, 5, 6 (for orientations between 45° and 90°) and the opposite bins, 16, 17, 18 (for orientations between 225° and 270°).

The distance transform (Figure 4.6(b)) of these edges (Figure 4.6(a)) is finally convolved with the rotated template (Figure 4.6(c)) to give the chamfer score (Figure 4.6(d)).

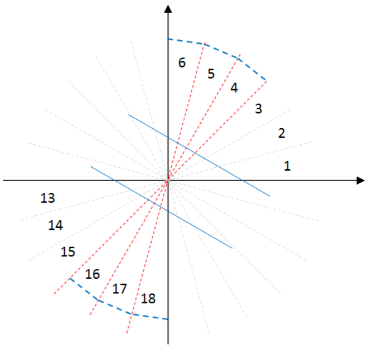
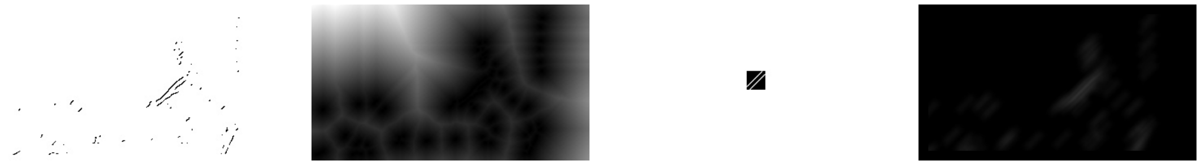


FIGURE 4.5: Orientations perpendicular on the template

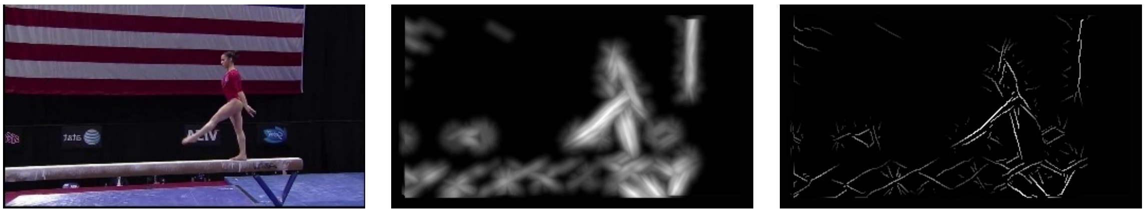


(a) Edges of interest orientations (b) Distance transform (c) Part template (d) Chamfer distance as convolution result

Figure 4.6: Chamfer matching

Non-maximum suppression

Figure 4.7(b) shows the resulting chamfer distance image obtained by selecting the best chamfer scores over individual edge orientations and their corresponding rotated templates. Sharpening this response is a useful technique that allows for thinner, more precise lines and eliminates noise. By imposing a threshold on these thin lines, it is possible to determine the position of the candidate body parts.



(a) Original image (b) Chamfer distance for an upper leg (c) Non-maximum suppression result

Figure 4.7: Non-maximum suppressed chamfer distance

Non-maximum suppression is commonly used to sharpen such responses, making them more appropriate to threshold [Kitchen and Rosenfeld, 1982]. The idea of non-maximum suppression of gradients is to search for local maxima in the gradient strength image (in our case, the chamfer distance image, see Figure 4.7(b)) in the gradient direction [Jepson, 2011].

We implement the non-maximum suppression algorithm as provided by [Martin, 2003]. We use a similar figure to [Jepson, 2011] to illustrate the method - see Figure 4.8. The algorithm steps are the following:

1. Determine in which of the eight areas does the edge normal (as depicted by the dashed green arrow) at the interest pixel (as depicted by a green dot) fall into.
2. Consider the original image. If the two neighbouring pixels in that area (as depicted by black dots on dashed blue lines) exist (they do not fall outside the image), calculate the angle tangent.
3. Interpolate the values of the pixels, based on the distance (the purple line in Figure 4.8) between them as given by the tangent.
4. Compare the interest pixel value with the interpolated value. If smaller, then suppress (make equal to zero) the pixel value in the result image.

Figure 4.7(c) shows the result of the non-maximum suppression on the chamfer distance image. This result gives clearer indications of the candidate upper legs positions. As the maximum values across orientations have been retained in the chamfer distance image (Figure 4.7(b)), as well as the corresponding orientations, only the body part position remains to be determined. By imposing a threshold for the chamfer distance, only the most likely positions are retained.

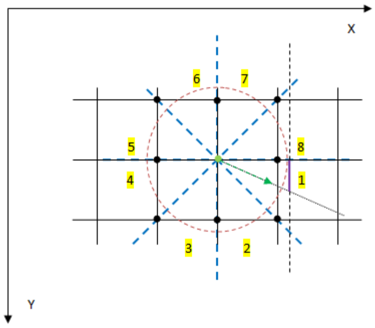


FIGURE 4.8: Non-maximum suppresion in gradient direction

Finding the body part candidates

The above procedures can be put in order in the following algorithm that finds candidates for a specific body part:

1. Build rectangle template of known width and length.
2. For each possible orientation of the body part,
   - 2.1 Rotate template.
   - 2.2 Select edges with perpendicular (and neighboring) orientations.
   - 2.3 Calculate distance transform of selected edges.
   - 2.4 Convolve distance transform with rotated template to obtain chamfer cost image.
3. For each pixel in the image, choose the best chamfer score across orientations.
4. Threshold chamfer scores and select those pixels locations that have better scores than the threshold.

Steps 3 and 4 give the orientation and the centre position of the body part detection. With the known width and length, the body part detection is now fully parameterised and scored. The corresponding chamfer score will represent the singleton potentials, $\Phi_i(x_i)$.

Figure 4.9 shows all upper leg candidates obtained with the above procedure.

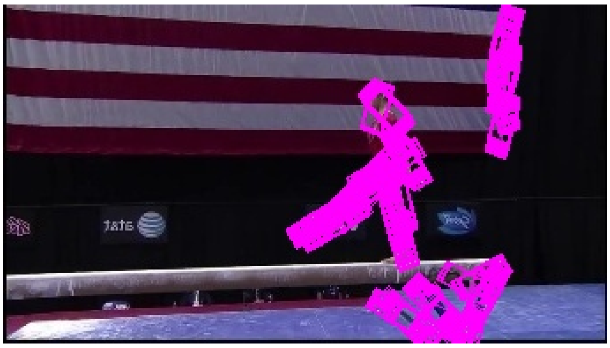


FIGURE 4.9: Sample image with candidate upper legs represented by purple rectangles

The discussed procedure can find thousands of body part candidates in an image. Imposing a threshold on the chamfer score does lower the number of candidates, but it is also convenient to set a maximum number of candidates. If there are more than a fixed number of candidates, [Ramanan et al., 2007] sample with replacement (see the sampling procedure in Section 3.3) an exact number of samples from the discrete distribution given by the chamfer scores of the candidate body parts.

4.2 Stylised pictorial structure

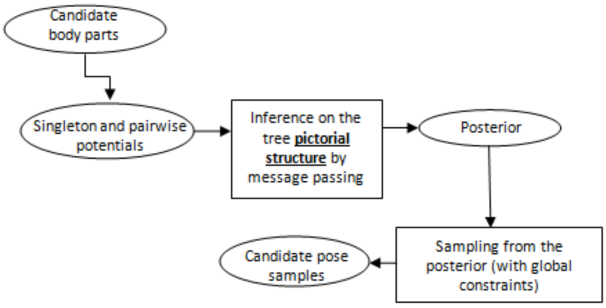

Candidate body parts
Singleton and pairwise potentials
Inference on the tree pictorial structure by message passing
Posterior
Sampling from the posterior (with global constraints)
Candidate pose samples

FIGURE 4.10: Model building pipeline: Matching a stylised pictorial structure.

Once the body part candidates have been found, using the procedure described in Section 4.1, we need to find those candidates which form a typical pose, as defined by kinematic constraints (hence, *stylised*). This is done by performing the inference (described in Section 3.3) on the pictorial structure tree model, where the nodes are represented by the candidates. Once the belief (or posterior) for each node has been obtained through message passing, we sample from the posterior to obtain several matches for the body configurations. This section refers to the model building module pipeline components in Figure 4.10.

Finding a typical pose in a video sequence is called *opportunistic detection*. The system will only recognize this pose and will use it to learn the colour appearance model parameters. The appearance model will be used by the detection module to find general poses in every frame.

The chosen stylised pose is a *lateral walking pose*, which has the following properties: it is usually encountered in a video of a person, it offers a clear and distinctive perspective on each of the body parts (little self-occlusion) and it is relatively easy to detect due to the scissor pattern in the legs. Figure 4.11 shows the tree model of the lateral walking pose, with both legs and one arm only. The acronyms for the body parts have been explained in Figure 3.1.

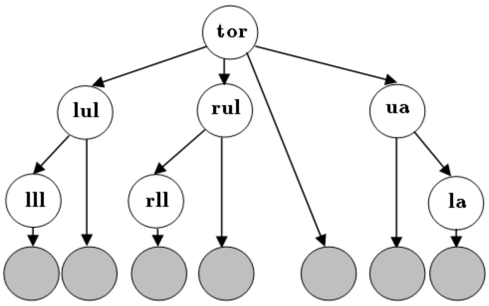


FIGURE 4.11: Tree model of a stylised pictorial structure.

In the single frame pictorial structure formula $P(P^{1:N}, I|C^{1:N}) \propto \prod_{i=1}^{N} P\left(I|P^i, C^i\right) P(P^i|P^{\pi(i)})$, the value for the image likelihood $P(I|P^i, C^i)$ will be:

$$P(I|P^i, C^i) = \frac{e^{-\Phi_i(x_i)/k}}{\sum_{x_i} e^{-\Phi_i(x_i)/k}}, \tag{4.2.1}$$

where x_i is a candidate obtained with the procedure described in Section 4.1, $\Phi_i(x_i)$ is a function of the candidate's chamfer score and k is a scaling factor used to smooth the likelihood. This procedure is related to *annealing* and ensures that different samples will be obtained in the sampling from posterior distributions procedure [Felzenszwalb and Huttenlocher, 2005].

The image likelihood expresses the match between a rectangle of a given size, representing the shape template of a body part, and a patch in the edges image corresponding to an input frame. It contains no colour information about the person we want to detect. This weakness is compensated by the prior term $P(P^i|P^{\pi(i)})$, which models *strict* kinematic constraints of the lateral walking pose such as:

1. the torso must be vertical such that the head is above it,
2. the bent in the elbow must be close to 180° (we chose this constraint due to the nature of walking poses in gymnastics, the source of our test videos),
3. the legs must be below the torso,
4. lower end points of the two lower legs must be away from each other.

Figure 4.12 shows a sampled upper leg and all lower leg candidates which satisfy the kinematic constraints with respect to the sampled upper leg as the parent body part.

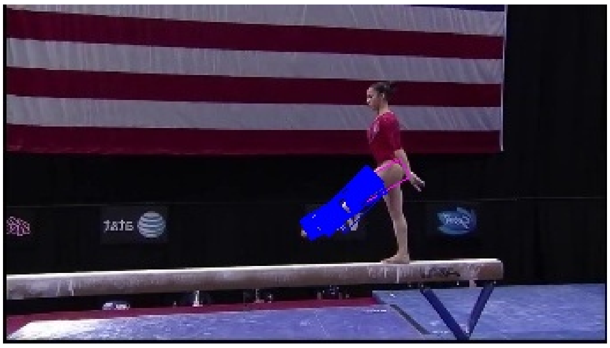


FIGURE 4.12: Valid lower legs (blue rectangles) under kinematic constraints for a sampled upper leg (purple rectangle).

[Ramanan et al., 2007] also imposed global appearance constraints for the legs, which means that they compare the colour histograms of the candidate legs and verify if they have a low dissimilarity. We found that this feature (shown as the pipeline component in Figure 4.13) does not always add value to the lateral walking pose detection and often chose to disable it.

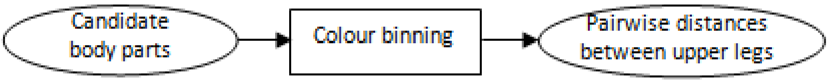
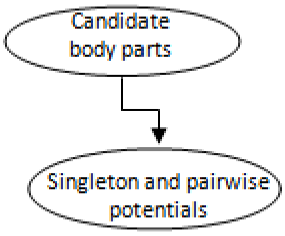


FIGURE 4.13: Module building pipeline: General appearance constraints.

Next, we describe the algorithm steps for finding a lateral walking pose in a static frame.



Model building pipeline: Algorithm steps 1 and 2 (finding candidates and calculating potentials).

Algorithm

1. Find body part candidates based on shape:
   1.1 Calculate singleton potentials $\Phi_i(x_i)$ for $i \in \{tor, la, ua, ll, ul\}$.

1.2 Retain candidates for which $\Phi_i\left(x_i\right) \geq chamfer\_thresh_i$ for $i \in \{tor, la, ua, ll, ul\}$.

2. Calculate pairwise (kinematic) potentials $\Psi_{i,\pi(i)}\left(x_{i,\pi(i)}\right)$ where $i \in \{tor, la, ua, ll, ul\}$ and $\pi(i)$ is the parent of part i.
3. Potentials to image likelihood and deformable model probabilities:

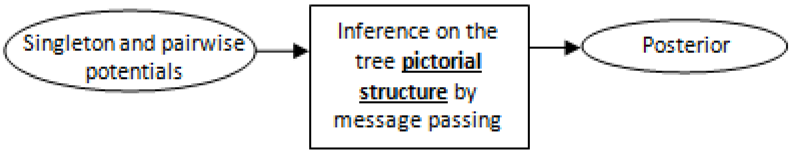
$$P(I|P^{la}, C^{la}) \leftarrow \frac{e^{-\Phi_{la}\left(x_{la}\right)/k}}{\sum_{x_{la}} e^{-\Phi_{la}\left(x_{la}\right)/k}} \qquad P(P^{la}|P^{ua}) \leftarrow \frac{e^{-\Psi_{la,ua}\left(x_{la,ua}\right)/k}}{\sum_{x_{la},x_{ua}} e^{-\Psi_{la,ua}\left(x_{la,ua}\right)/k}}$$

$$P(I|P^{ua}, C^{ua}) \leftarrow \frac{e^{-\Phi_{ua}\left(x_{ua}\right)/k}}{\sum_{x_{ua}} e^{-\Phi_{ua}\left(x_{ua}\right)/k}} \qquad P(P^{ua}|P^{tor}) \leftarrow \frac{e^{-\Psi_{ua,tor}\left(x_{ua,tor}\right)/k}}{\sum_{x_{ua},x_{tor}} e^{-\Psi_{ua,tor}\left(x_{ua,tor}\right)/k}}$$

$$P(I|P^{ll}, C^{ll}) \leftarrow \frac{e^{-\Phi_{ll}\left(x_{ll}\right)/k}}{\sum_{x_{ll}} e^{-\Phi_{ll}\left(x_{ll}\right)/k}} \qquad P(P^{ll}|P^{ul}) \leftarrow \frac{e^{-\Psi_{ll,ul}\left(x_{ll,ul}\right)/k}}{\sum_{x_{ll},x_{ul}} e^{-\Psi_{ll,ul}\left(x_{ll,ul}\right)/k}}$$

$$P(I|P^{ul}, C^{ul}) \leftarrow \frac{e^{-\Phi_{ul}\left(x_{ul}\right)/k}}{\sum_{x_{ul}} e^{-\Phi_{ul}\left(x_{ul}\right)/k}} \qquad P(P^{ul}|P^{tor}) \leftarrow \frac{e^{-\Psi_{ul,tor}\left(x_{ul,tor}\right)/k}}{\sum_{x_{ul},x_{tor}} e^{-\Psi_{ul,tor}\left(x_{ul,tor}\right)/k}}$$

$$P(I|P^{tor}, C^{tor}) \leftarrow \frac{e^{-\Phi_{tor}\left(x_{tor}\right)/k}}{\sum_{x_{tor}} e^{-\Phi_{tor}\left(x_{tor}\right)/k}}$$


Singleton and pairwise potentials
Inference on the tree pictorial structure by message passing
Posterior

Model building pipeline: Algorithm steps 4, 5 and 6 (tree inference).

4. Calculate messages from lower body parts to upper body parts:

$$m_{la,ua}\left(x_{ua}\right) \leftarrow \sum_{x_{la}} P(I|P^{la}, C^{la})P(P^{la}|P^{ua}) \qquad m_{la,ua}\left(x_{ua}\right) \leftarrow \frac{m_{la,ua}\left(x_{ua}\right)}{\sum_{x_{ua}} m_{la,ua}\left(x_{ua}\right)}$$

$$m_{ll,ul}\left(x_{ul}\right) \leftarrow \sum_{x_{ll}} P(I|P^{ll}, C^{ll})P(P^{ll}|P^{ul}) \qquad m_{ll,ul}\left(x_{ul}\right) \leftarrow \frac{m_{ll,ul}\left(x_{ul}\right)}{\sum_{x_{ul}} m_{ll,ul}\left(x_{ul}\right)}$$

5. Separate left upper legs and right upper legs, based on orientation, and calculate messages from upper body parts to torso:

$$m_{ua,tor}\left(x_{tor}\right) \leftarrow \sum_{x_{ua}} P(I|P^{ua}, C^{ua})P(P^{ua}|P^{tor})m_{la,ua}\left(x_{ua}\right) \qquad m_{ua,tor}\left(x_{tor}\right) \leftarrow \frac{m_{ua,tor}\left(x_{tor}\right)}{\sum_{x_{tor}} m_{ua,tor}\left(x_{tor}\right)}$$

$$m_{lul,tor}\left(x_{tor}\right) \leftarrow \sum_{x_{lul}} P(I|P^{lul}, C^{lul})P(P^{lul}|P^{tor})m_{ll,lul}\left(x_{lul}\right) \qquad m_{lul,tor}\left(x_{lul}\right) \leftarrow \frac{m_{lul,tor}\left(x_{tor}\right)}{\sum_{x_{tor}} m_{lul,tor}\left(x_{tor}\right)}$$

$$m_{rul,tor}\left(x_{tor}\right) \leftarrow \sum_{x_{rul}} P(I|P^{rul}, C^{rul})P(P^{rul}|P^{tor})m_{ll,rul}\left(x_{rul}\right) \qquad m_{rul,tor}\left(x_{rul}\right) \leftarrow \frac{m_{rul,tor}\left(x_{tor}\right)}{\sum_{x_{tor}} m_{rul,tor}\left(x_{tor}\right)}$$

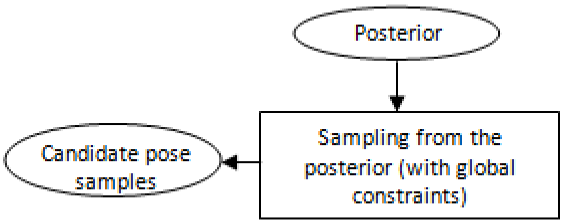
6. Calculate torso (root) posterior:

$$b_{tor}(x_{tor}) \leftarrow m_{ua,tor}\left(x_{tor}\right) m_{lul,tor}\left(x_{tor}\right) m_{rul,tor}\left(x_{tor}\right) \qquad b_{tor}\left(x_{tor}\right) \leftarrow \frac{b_{tor}(x_{tor})}{\sum_{x_{tor}} b_{tor}(x_{tor})}$$

7. Sampling:

- sample tor_idx from $b_{tor}\left(x_{tor}\right)$
- sample lul_idx from $P(I|P^{lul}, C^{lul})P(P^{lul}|P^{tor\_idx})m_{ll,lul}\left(x_{lul}\right)$
- sample lll_idx from $P(I|P^{lll}, C^{lll})P(P^{ll}|P^{lul\_idx})$ where x_{ll} on same side as x_{lul_idx}

- sample rul_idx from $P(I|P^{rul}, C^{rul})P(P^{rul}|P^{tor\_idx})m_{ll,rul}(x_{rul})$ where x_{rul} similar in appearance with x_{lul_idx} and far from x_{lul_idx}
- sample rll_idx from $P(I|P^{rll}, C^{rll})P(P^{ll}|P^{rul\_idx})$ where x_{ll} on same side as x_{rul_idx} and similar in appearance with x_{lll_idx}
- sample ua_idx from $P(I|P^{ua}, C^{ua})P(P^{ua}|P^{tor\_idx})m_{la,ua}(x_{ua})$
- sample la_idx from $P(I|P^{la}, C^{la})P(P^{la}|P^{ua\_idx})$



Model building pipeline: Algorithm step 7 (sampling).

Step 7 is performed 2000 times. This leads to 2000 different lateral walking pose configurations obtained in a single frame. The final walking pose in this frame is selected as the pose with the highest score, where the score is calculated as the sum of the image likelihoods of all body parts (model building module pipeline component shown in Figure 4.14).

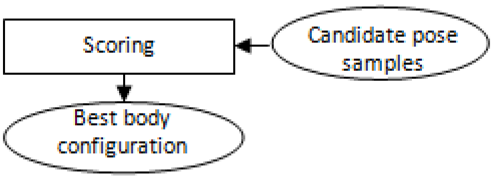


FIGURE 4.14: Model building pipeline: Choosing the best sampled configuration in a frame.

A result example of the above algorithm on an input frame of a gymnast on a beam is shown in Figure 4.15.

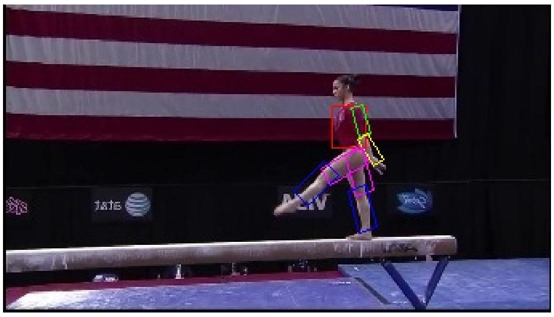
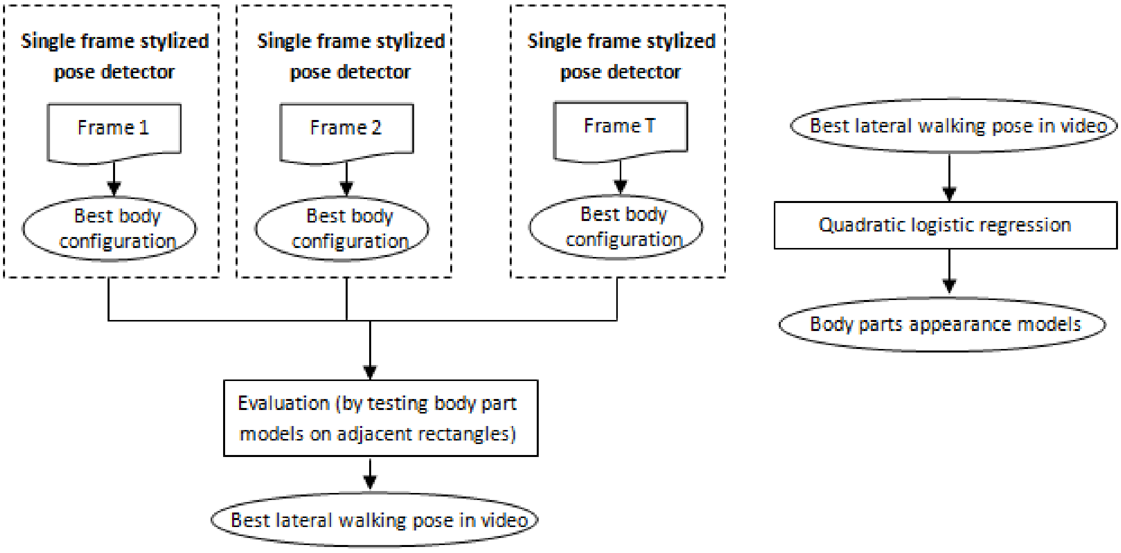


FIGURE 4.15: Lateral walking pose detection

4.3 Classification and Evaluation

When a lateral walking pose is found in a frame, the system first evaluates the correctness of the pose detection. Once the entire video has been processed to find all the lateral walking poses, their respective scores are compared to obtain the best detection (pipeline component shown in Figure 4.16(a)). We explain the evaluation in Section 4.3.3. The best lateral walking pose detection is chosen to further learn the appearance model of the distinct body parts (pipeline component shown in Figure 4.16(b)).


Single frame stylized pose detector
Frame 1
Best body configuration
Single frame stylized pose detector
Frame 2
Best body configuration
Single frame stylized pose detector
Frame T
Best body configuration
Evaluation (by testing body part models on adjacent rectangles)
Best lateral walking pose in video
Best lateral walking pose in video
Quadratic logistic regression
Body parts appearance models

(a) Model building pipeline: Walking pose evaluation in different frames.

(b) Model building pipeline: Learning appearance model parameters.

Figure 4.16

We first lay the foundation of the classification method in Section 4.3.1 and Section 4.3.2. Then we explain the pose evaluation procedure in Section 4.3.3.

4.3.1 Linear methods for classification

Given a number of variables that describe an object, classification addresses the problem of placing the object in a class and estimating the probability that the object is part of that class [Feelders and Veltkamp, 2012].

In the following, we will use the concepts and notations of [Hastie et al., 2001] to derive the theoretical background for linear methods for classifications and linear logistic regression. The set of variables are called *inputs* which determine the *outputs*, also called *responses*. The outputs can belong to different classes denoted as $\mathcal{G}$. In our context, the inputs are the colour values of a pixel and the classes are: body part pixel and non-body part pixel, for a specific body part.

The main assumption of linear regression is that the regression function $E(Y|X)$ is linear in the inputs. Given a vector of inputs $X = (X_1, X_2, \ldots, X_p)$, the linear regression model that predicts the output Y

has the form:

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j\beta_j \tag{4.3.1}$$

where β_0 is called *intercept* or *bias* and β_j's are model coefficients and need to be learnt through classification. If the vector X includes the constant 1 and the intercept is included in β, equation (4.3.1) can be rewritten as

$$f(X) = X^T\beta \tag{4.3.2}$$

We wish to model posterior probabilities $P(G = k|X = x)$, that is, given that we know the colour of a pixel, what is the probability that the pixel is part of a specific body segment. Linear methods require that some monotone transformation of this probability is linear, so that the decision boundaries which divide the input space in regions are linear.

A common model for the posterior probabilities is the *logistic function* as $logit^{-1}(\alpha) = \frac{1}{1+e^{(-\alpha)}} = \frac{e^{\alpha}}{1+e^{\alpha}}$. The graph for this function is drawn in Figure 4.17.

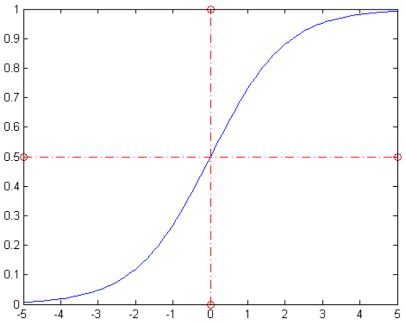


FIGURE 4.17: Standard logistic function

For the two class case, the probabilities are formulated as follows:

$$\begin{aligned} P(G = 1|X = x) &= \frac{e^{(\beta_0+\beta^T x)}}{1 + e^{(\beta_0+\beta^T x)}} \\ P(G = 2|X = x) &= \frac{1}{1 + e^{(\beta_0+\beta^T x)}} \end{aligned} \tag{4.3.3}$$

The monotone transformation is the *logit* transformation $logit\,(p) = \log\,[\frac{p}{1-p}]$. The logistic function is the inverse of the logit transformation. The ratio $\frac{P(G=1|X=x)}{1-P(G=1|x=x)}$ is called *odds*.

Equations (4.3.3) result in the *log-odds*:

$$\log\frac{P(G = 1|X = x)}{P(G = 2|X = x)} = \beta_0 + \beta^T x \tag{4.3.4}$$

The decision boundary is obtained by making equation (4.3.4) equal to 0 which describes the hyperplane $\{x|\beta_0 + \beta^T x\}$. Linear logistic regression results in linear log-odds and will be discussed in Section 4.3.2. Now, considering that $\mathcal{G}$ has K classes, the responses will be represented by K indicator variables $Y_k, k = 1, \ldots, K$, with: $Y_k = 1$ if $G = k$ and $Y_k = 0$ elsewise.

The indicator response matrix $\mathbf{Y}$ is a $N \times K$ matrix, where N is the number of available input data and each row is constituted by the vectors $Y = (Y_1, \dots, Y_K)$. In the following we will consider $K = 2$ classes, body part pixel and non-body part pixel.

4.3.2 Logistic regression

The model for two class logistic regression has the same form as equation (4.3.4). Logistic regression ensures that the posterior probabilities of the K classes sum to one and remain under the unit value. [Hastie et al., 2001] demonstrate this on the general case of K classes.

Considering N observations, their log-likelihood is:

$$l(\beta) = \sum_{i=1}^{N} \log p_{g_i}(x_i; \beta) \tag{4.3.5}$$

where $p_k(x_i; \beta) = P(G = k|X = x_i; \beta)$.

For our two classes, we can encode $g_i = 1$ with $y_i = 1$ and $g_i = 2$ with $y_i = 0$. Therefore, the response vectors will contain a 1 for those pixels which belong to the body part and a 0 for all the other pixels in the image. The log-likelihood $\log p_{g_i}(x; \beta)$ takes one of two values:

$$\log p_{g_i}(x; \beta) = \begin{cases} \log p(x; \beta), & y_i = 1 \\ 1 - \log p(x; \beta), & y_i = 0 \end{cases} \tag{4.3.6}$$

Using the 0/1 for y_i values as a selector, we can rewrite equation (4.3.5) as such:

$$l(\beta)) = \sum_{i=1}^{N} \{y_i \log p(x_i; \beta) + (1 - y_i) \log(1 - p(x_i; \beta))\} \tag{4.3.7}$$

$$= \sum_{i=1}^{N} \{y_i \log p(x_i; \beta) + \log(1 - p(x_i; \beta)) - y_i \log(1 - p(x_i; \beta))\} \tag{4.3.8}$$

$$= \sum_{i=1}^{N} \left\{ y_i \log \frac{p(x_i; \beta)}{1 - p(x_i; \beta)} + \log(1 - p(x_i; \beta)) \right\} \tag{4.3.9}$$

From equation (4.3.4), we see that $\log \frac{p(x_i;\beta)}{1-p(x_i;\beta)} = \beta^T x_i$. From equation (4.3.3), we see that $1 - p(x_i; \beta) = \frac{1}{1+e^{\beta^T x_i}}$. Replacing these into (4.3.9) leads to:

$$l(\beta) = \sum_{i=1}^{N} \left\{ y_i \beta^T x_i - \log\left(1 + e^{\beta^T x_i}\right) \right\}, \tag{4.3.10}$$

where β contains the intercept and vectors x_i contain the corresponding constant value 1.

Knowing that $\frac{d}{dx} \log_b u(x) = \frac{1}{x \ln b} u'(x)$, the derivation of the previous equation results in the following:

$$\frac{\partial l(\beta)}{\partial \beta} = \sum_{i=1}^{N} \left(x_i y_i - \frac{1}{(1 + e^{\beta^T x_i}) \ln e} e^{\beta^T x_i} x_i \right) = \sum_{i=1}^{N} x_i \left(y_i - \frac{e^{\beta^T x_i}}{1 + e^{\beta^T x_i}} \right) \tag{4.3.11}$$

With (4.3.3) and setting the derivative to 0, in order to maximize (4.3.9), (4.3.11) becomes:

$$\frac{\partial l(\beta)}{\partial \beta} = \sum_{i=1}^{N} x_i(y_i - p(x_i;\beta)) = 0 \tag{4.3.12}$$

Newton-Raphson algorithm

Equations (4.3.12) are called *score* equations and are solved using the Newton-Raphson algorithm [Weisstein, c]. This algorithm is an iterative method to find the root of a function. Considering x_0 an initial approximation of the root, $x_1 = x_0 + \epsilon_0$ is taken as a better approximation of the root. Using the Taylor expansion of $f(x_1)$, the offset ϵ_0 is $\epsilon_0 = -\frac{f(x_0)}{f'(x_0)}$. With the calculated value of x_1, the process can be repeated until convergence.

In our terms, we can say that x_1 is β_{new}, x_0 is β_{old} and $f(x)$ is $\frac{\partial l(\beta)}{\partial \beta}$. Then,

$$\beta^{new} = \beta^{old} - \left(\frac{\partial^2 l(\beta)}{\partial\beta\partial\beta^T}\right)^{-1} \frac{\partial l(\beta)}{\partial(\beta)} \tag{4.3.13}$$

This update is repeated until either the log-likelihood increases, or, if the log-likelihood decreases, the step size is halved.

In matrix notation:

$\mathbf{y}$	vector of y_i values
$\mathbf{X}$	$N \times (p+1)$ matrix of x_i values
$\mathbf{p}$	vector of fitted probabilities $p(x_i;\beta^{old})$
$\mathbf{W}$	$N \times N$ diagonal matrix of weights $p(x_i;\beta^{old})(1-p(x_i;\beta^{old}))$

Then, the first derivative of the likelihood w.r.t. β is $\frac{\partial l(\beta)}{\partial(\beta)} = \mathbf{X}^T(\mathbf{y}-\mathbf{p})$, and the second derivative of the likelihood w.r.t. β, or the Hessian, is $\frac{\partial^2 l(\beta)}{\partial\beta\partial\beta^T} = -\mathbf{X}^T\mathbf{W}\mathbf{X}$. Practically, in the implementation, we invert the Hessian matrix using [alg, 2013].

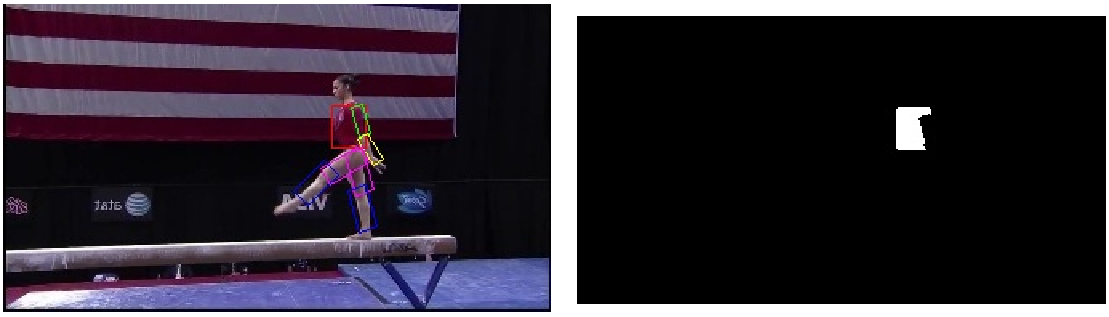
Input data

[Ramanan et al., 2005] used a quadratic logistic regression model to learn the appearance of distinct body parts. This model solves the issue of testing on the same data used for training, which in case of nearest neighbors classifiers gives zero errors. The input vectors x_i, in this case, are composed of:

- R, G, B values of foreground and background pixels
- basis expansions R^2, G^2, B^2, RB, GB, RG

Foreground pixels are those pixels contained within the rectangle that represents the detection of a body part. Background pixels are all the other pixels in the image. Figure 4.18 shows the mask for the torso pixels where the occluding arm has been removed. The pixels under this masks will be used as positive samples, while all other pixels in the image will be used as negative samples.

For each upper leg and lower leg respectively, the pairwise body parts are ignored, in order to remove false negatives. This is founded by the assumption that the legs should be similar in appearance. Unlike the original method where pixels from both legs were used as positives, we only use the left side leg pixels to learn the appearance, for computation speed reasons.



(a) Lateral walking pose detection

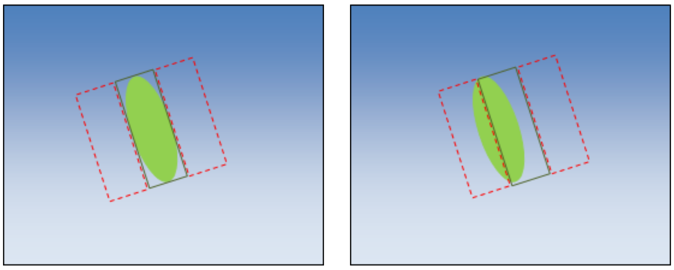
(b) Torso pixels mask

FIGURE 4.18: Torso pixels masks with removed occluding arm for appearance learning

4.3.3 Walking pose evaluation in different frames

In this section we refer to the model building module pipeline components shown in Figure 4.16(a). The body part detection (pixels within the green rectangles in Figure 4.19) are used as positives and the flanking rectangles (pixels within the dashed red rectangles in Figure 4.19) of said detection are used as negatives to build a logistic regression appearance model of the respective body part. The model is then tested on the same pixels and the misclassified pixels are counted, accounting for the score of a detection. A misclassified pixel is said to be a pixel whose classification result differs from the actual status (inside or outside the body part detection).

The classification result is given by the computed log-likelihood of the pixel being a body part pixel, which can be computed with known appearance model β using equations (4.3.3) and (4.3.7). A likelihood above 0.5 means that the pixel was classified as body part pixel, otherwise the pixel was classified as background pixel (see the graph of the standard logistic function in Figure 4.17).



(a) Correct body part detection

(b) Wrong body part detection

FIGURE 4.19: Body part detections and flanking rectangles used to evaluate detection correctness

We explain this reasoning by illustrating two cases in Figure 4.19: the case when the detection is correct and the case when the detection is slightly deviated from the correct position. In case 4.19(a), the pixels within the flanking rectangles will have a completely different appearance than the pixels within the detection. In case 4.19(b), the pixels within the flanking rectangles will look like some of the pixels within the detection (as the incorrect detection includes both body part pixels and non-body part pixels in the illustrated case). In the latter case, the learnt model will classify pixels from the flanking rectangles as body part pixels, increasing the number of misclassified pixels, therefore accurately indicating that the detection is incorrect.

Finally, the summation of the misclassified pixels for each body part gives the total score for a lateral walking pose detection. The appearance models are learnt from the frame that contains the lateral walking pose detection with the highest score.

5 Detection

The goal of this module is to find general human poses by using the learnt body part colour appearance models, as well as the initially known person scale. The pipeline of the detection module is given in Figure 5.1. The candidates are matched in a pictorial structure, then inference is performed on the tree model, to obtain the posterior. Several configurations are sampled from the posterior, which determine the distribution of poses. The last step is to find the modes of this distribution which represent the final detection.

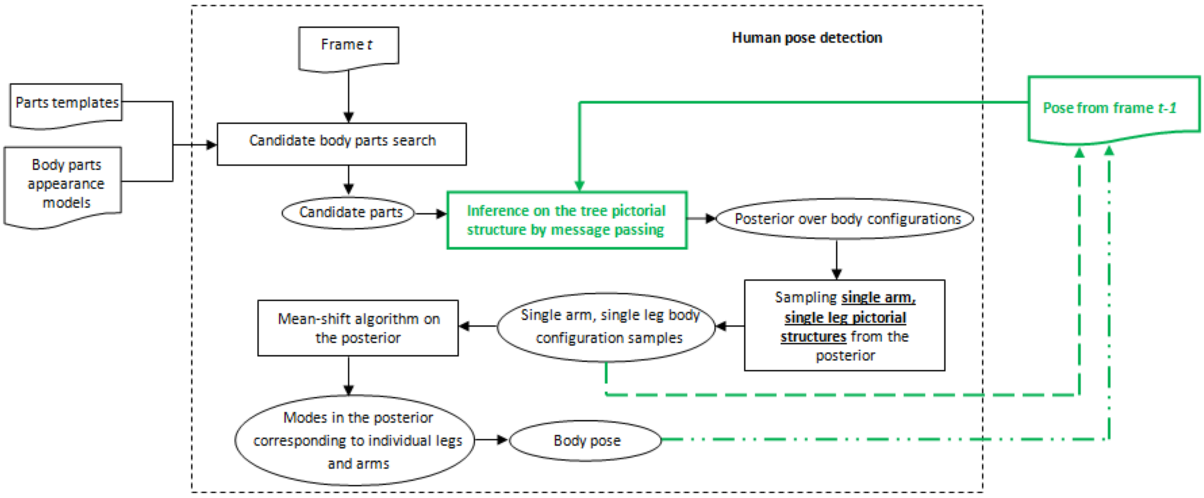

Frame *t*
Human pose detection
Parts templates
Body parts appearance models
Candidate body parts search
Pose from frame *t-1*
Candidate parts
Inference on the tree pictorial structure by message passing
Posterior over body configurations
Sampling single arm, single leg pictorial structures from the posterior
Mean-shift algorithm on the posterior
Single arm, single leg body configuration samples
Modes in the posterior corresponding to individual legs and arms
Body pose

FIGURE 5.1: Detection module pipeline.

The original method, that of the single frame pictorial structure, is contained within the black dashed rectangle. We try to improve this method by suggesting two ways of including the previous frame information through the motion model prior:

1. We consider the previous body part detection (obtained with the modes finding procedure) as P_{t-1}^i and calculate the motion model prior with respect to this previous body part as $P(P_t^i|P_{t-1}^i)$. This feature is shown with a dot and dash green arrow in Figure 5.1.
2. We consider all the previously sampled body part candidates $x_{i_{t-1}}$, and calculate the motion model prior with respect to these candidates as $\sum_{x_{i_{t-1}}} \left[P(P_t^i|P_{t-1}^i)P(I_{t-1}|P_{t-1}^i, C^i)\right]$. This feature is shown with a dashed green arrow in Figure 5.1.

Finding a stylised pose, as described in Section 4.2, is possible due to rigorously imposed kinematic constraints, general poses require more relaxed body part connections, to account for variety. Colour and shape appearance models combined give a more reliable image likelihood compared to shape templates alone. This compensates for the soft kinematic constraints. We also added a motion model to aid detection accuracy.

5.1 Searching for body parts using appearance models

In this section we explain how rectangular templates and the learnt colour appearance models can be used to find candidate body parts in input frames containing any possible human pose. With learnt colour appearance model parameters β_i, the indicator response vector for body part i becomes:

$$Y_i = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{pmatrix} \begin{pmatrix} \beta_{i1} \\ \vdots \\ \beta_{i9} \\ 1 \end{pmatrix} \tag{5.1.1}$$

where:

x_j^T colour values $\begin{pmatrix} R & G & B & R^2 & G^2 & B^2 & RB & GB & RG & 1 \end{pmatrix}$, at pixel j

N number of pixels in the image

Y_i binary N sized indicator response vector

i body part.

The binary image in Figure 5.2(f) showing pixels assigned to a specific body part is obtained by reshaping the values from the output vector Y according to input frame width and height. Knowing which areas in the image look like body part of interest, the next step is to match the rectangular body part template to these areas. Matching will give the body part centre coordinates and orientation with the highest score, or image likelihood. The procedure is conceptually similar to the one described in Section 4.1, namely using rectangular filters as part templates. The difference is that the search is now more localized.

Similar to [Ramanan et al., 2007], we use a Gaussian centered at the part template to model the local image patch. This is a two-dimensional Gaussian function calculated as:

$$f(x,y) = Ae^{-\left[\frac{x-x_0}{2\sigma_x^2}+\frac{y-y_0}{2\sigma_y^2}\right]}, \text{for } x \in [0, width] \text{ and } y \in [0, length] \tag{5.1.2}$$

where:

A is the amplitude, set at $A = length \cdot width$

σ_x, σ_y are the x and y spreads, set at $\sigma_x = 0.2 \cdot width^2$ and $\sigma_y = 0.2 \cdot length^2$

x_0, y_0 are the means, $x_0 = {}^{width}/_2$ and $y_0 = {}^{length}/_2$.

For A, σ_x and σ_y we use the same values as the authors.

The following two filters are used: an interior filter obtained by flanking the two-dimensional Gaussian function in equation (5.1.2) with two equally sized black rectangles (Figure 5.2(b)) and an exterior filter composed of a black rectangle flanked by two white rectangles of the body part's width and length (Figure

5.2(g)). Let these two filters be H_{i1} and H_{i2} (where i denotes the body part), and I_i the input image obtained from the indicator response vector Y_i. Then, the singleton potential for body part i, calculated at body part i candidate x_i, is obtained from the following equation:

$$\Phi_i\left(x_i(x,y,\theta)\right) = e^{-\frac{\left[(1-I_i)\star H_{i1}(\theta)\right](x,y)+\left[I_i\star H_{i2}(\theta)\right](x,y)}{length_i\cdot width_i\cdot k}}, \tag{5.1.3}$$

where $H_{i1}(\theta)$ and $H_{i2}(\theta)$ are filters rotated with angle θ. Figures 5.2(c) and 5.2(h) show convolutions $(1 - I_i) \star H_{i1}(135°)$ and $I_i \star H_{i2}(135°)$, respectively.

The body part candidate x_i is parameterised by its centre coordinates (x, y), known width and length and orientation θ. Possible orientations θ are $\theta = a \cdot 15°$, for $a = 1, \ldots, 12$.

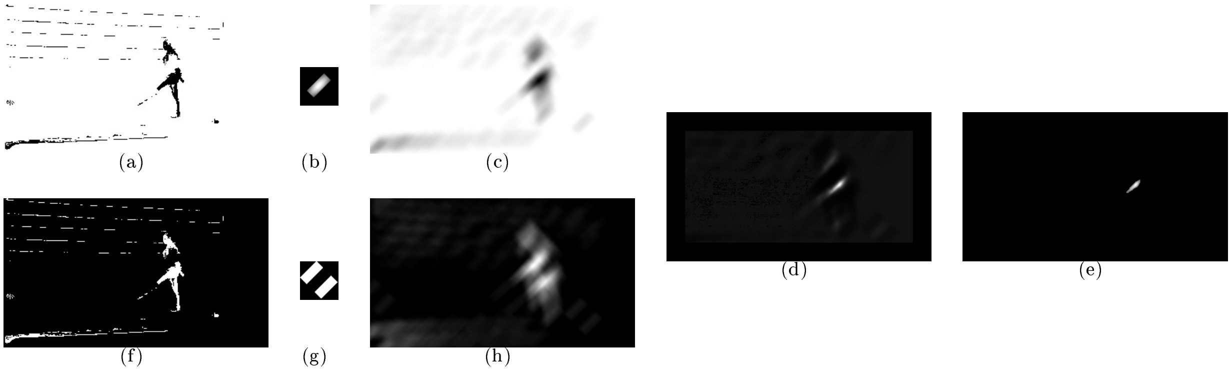

(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 5.2: Body part detection with known appearance and shape model.
(a),(f) - assigned body part pixels. (b), (g) - interior and exterior filters. (c), (h) - convolutions
(d) - cost as exponential of the sum of (c) and (h). (e) - thresholded cost.

The valid candidates x_i are the candidates for which the singleton potential $\Phi_i(x_i)$ is larger than an imposed threshold. Figure 5.2(d) shows the result of equation (5.1.3) for $\theta = 135°$. Figure 5.2(e) shows the thresholded result. The non-black pixel values represent the values for $\Phi_i\left(x_i(135°)\right)$, while their coordinates represent the center coordinates (x, y) for candidates x_i oriented at 135°. This procedure is performed for all possible orientations, mentioned above.

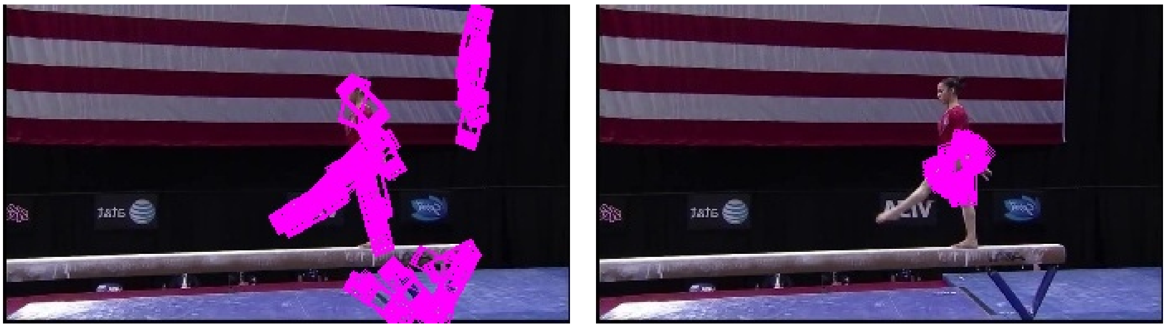
In Figure 5.3 we compare the results obtained when looking for upper legs using only the shape model (rectangular template) and using both the shape model and the colour appearance model. Figure 5.3(b) shows more accurately localized upper legs, relative to the person's position.

For the head, which is represented as a square instead of a rectangle and parameterised with a single orientation $\theta = 0°$, the above algorithm is slightly different. The Gaussian centred at the part template f, the interior filter f_1, and exterior filter f_2, are given by the following equations:

$$f(x,y) = A_1 e^{-\left[\frac{x-x_0}{2\sigma x_1^2}+\frac{y-Y_0}{2\sigma y_1^2}\right]} + A_2 e^{-\left[\frac{x-x_0}{2\sigma x_2^2}+\frac{y-y_0}{2\sigma y_2^2}\right]} \tag{5.1.4}$$

$$f_1(x,y) = \begin{cases} f(x,y), & f(x,y) > 0 \\ 0, & f(x,y) \leq 0 \end{cases} \tag{5.1.5}$$

$$f_2(x,y) = \begin{cases} -f(x,y), & f(x,y) \leq 0 \\ 0, & f(x,y) > 0 \end{cases} \tag{5.1.6}$$



(a) Result with shape model

(b) Result with shape and appearance model

FIGURE 5.3: Sample image with candidate upper legs represented by purple rectangles

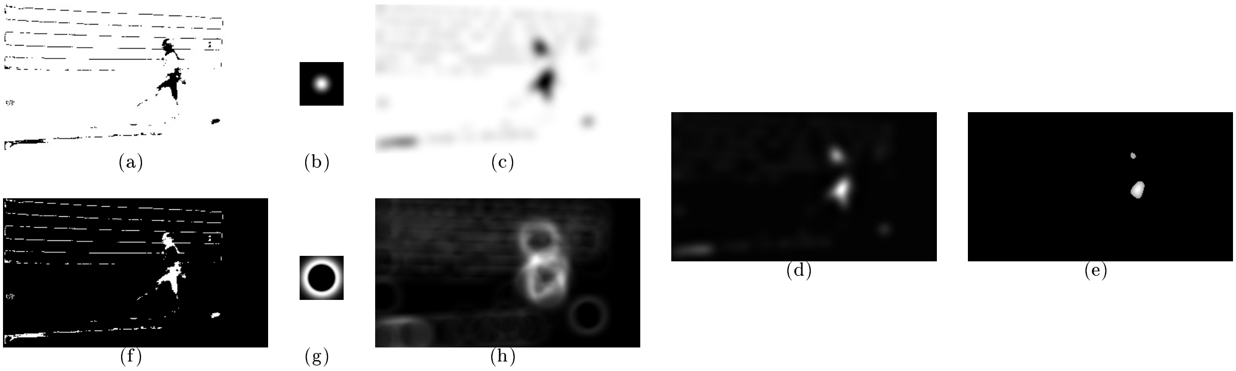
for $x, y \in [0, width]$, where:

A_1, A_2 are amplitudes, set at $A_1 = 1$ and$A_2 = 0.5$

$\sigma_{x1}, \sigma_{y1}, \sigma_{x2}, \sigma_{y2}$ are the x and y spreads, set at $\sigma_{x1} = \sigma_{y1} = 0.25 \cdot width^2$ and $\sigma_{x2} = \sigma_{y2} = 0.5 \cdot width^2$

x_0, y_0 are the means, $x_0 = y_0 = {}^{(4 \cdot width + 1)}/_{2}$.

Filters f_1 and f_2 are shown in Figures 5.4(b) and 5.4(g).



(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 5.4: Head detection with known appearance and shape model
(a),(f) - assigned head pixels. (b), (g) - interior and exterior filters. (c), (h) - convolutions
(d) - cost as exponential of the sum of (c) and (h). (e) - thresholded cost.

The following equation gives the singleton potential $\Phi_{head}(x_{head})$:

$$\Phi_{head}\left(x_{head}(x, y)\right) = e^{-\frac{((1-I_{head})\star f_1)(x,y)+(I_{head}\star f_2)(x,y)}{k}}. \tag{5.1.7}$$

Figures 5.4(c) and 5.4(h) show convolutions $(1 - I_{head}) \star f_1$ and $I_{head} \star f_2$, respectively.

As for the other body part types, the valid candidates x_{head} are the candidates for which the singleton potential $\Phi_{head}(x_{head})$ is larger than an imposed threshold. Figure 5.4(d) shows the result of equation (5.1.7). Figure 5.4(e) shows the thresholded result. The non-black pixel values represent the values for $\Phi_{head}(x_{head})$, while their coordinates represent the center coordinates (x, y). We show the result obtained via this procedure in Figure 5.5.

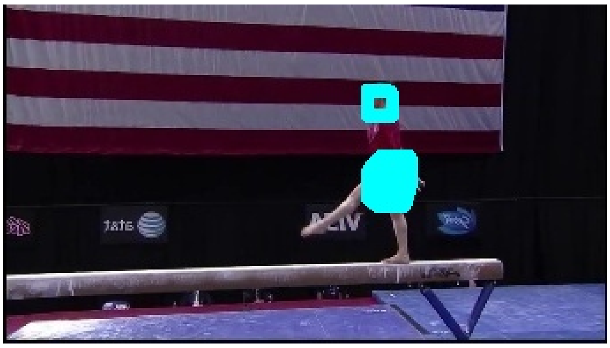


FIGURE 5.5: Sample image with candidate heads (turquoise squares) obtained with shape and appearance model

5.2 General pictorial structure

To deal with overlap between body parts, the general pictorial structure is modelled as a *single arm, single leg pictorial structure*. By sampling from the posterior of body configurations, we will find such poses that will either encompass only one leg/arm at the time. This means that, for example, for a fully visible person, some single arm, single leg configuration samples will contain the right arm, other samples will contain the left arm. In this case, the mode finding procedure will determine two modes which correspond to the two distinct arms or legs (see Section 5.3).

In the single frame pictorial structure formula $P(P^{1:N}, I|C^{1:N}) \propto \prod_{i=1}^{N} P\left(I|P^i, C^i\right) P(P^i|P^{\pi(i)})$, the value for the image likelihood $P(I|P^i, C^i)$ will be:

$$P(I|P^i, C^i) = \frac{e^{-\Phi_i(x_i)/k}}{\sum_{x_i} e^{-\Phi_i(x_i)/k}}, \tag{5.2.1}$$

where x_i is a candidate obtained with the procedure described in Section 5.1, $\Phi_i(x_i)$ is calculated with one of the formulas (5.1.3) or (5.1.7) and k is a scaling factor used to smooth the likelihood.

In comparison with the stylised pictorial structure case, where the image likelihood was a function of the matching score between a shape template and an edge image patch, now the image likelihood also carries information about the match between colour appearances of the learnt template and the image patch, respectively. This allows for lesser strict kinematic constraints, which need to be *general* enough in order cover the entire range of human poses. Except for the hard constraints, which impose that body parts are connected, some softer constraints express a preference for certain relative positions, like:

1. the upper arm/leg position should preferably be far away from the torso,
2. shoulder and wrist should not overlap,
3. the lower leg position should preferably be far away from the upper leg.

5.2.1 Single frame pictorial structure

Figure 5.6 shows the graphical model for a single frame, single arm, single leg pictorial structure. The inference on this model is done according to the procedure explained in Section 3.3. This algorithm finds single arm, single leg body configurations using the information found in the current frame only (without a motion model prior). [Ramanan et al., 2007] obtained successful detection results using this algorithm and by smoothing the track. Smoothing was done by adding the previous frame and the next frame samples to the current set of samples, which they provided as input to the modes finding procedure. Sampling was done 1000 times in each frame.

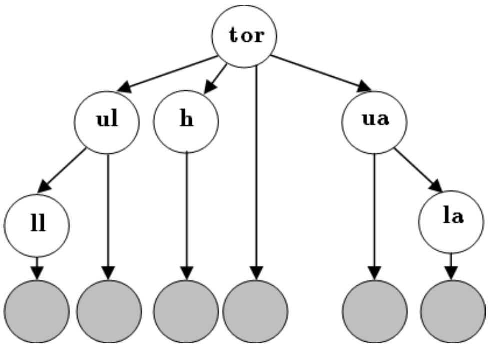


FIGURE 5.6: Tree model of a single frame pictorial structure

Algorithm

1. Find body part candidates based on shape and colour appearance:
   1.1 Calculate singleton potentials $\Phi_i\left(x_i\right)$ for $i \in \{tor, la, ua, ll, ul, h\}$.
   1.2 Retain candidates for which $\Phi_i\left(x_i\right) \geq thresh_i$ for $i \in \{tor, la, ua, ll, ul, h\}$.
2. Calculate pairwise (kinematic) potentials $\Psi_{i,\pi(i)}\left(x_{i,\pi(i)}\right)$ where $i \in \{tor, la, ua, ll, ul, h\}$ and $\pi(i)$ is the parent of part i.
3. Potentials to probabilities:

$$P(I|P^{la}, C^{la}) \leftarrow \frac{e^{-\Phi_{la}\left(x_{la}\right)/k}}{\sum_{x_{la}} e^{-\Phi_{la}\left(x_{la}\right)/k}} \qquad P(P^{la}|P^{ua}) \leftarrow \frac{e^{-\Psi_{la,ua}\left(x_{la,ua}\right)/k}}{\sum_{x_{la},x_{ua}} e^{-\Psi_{la,ua}\left(x_{la,ua}\right)/k}}$$

$$P(I|P^{ua}, C^{ua}) \leftarrow \frac{e^{-\Phi_{ua}\left(x_{ua}\right)/k}}{\sum_{x_{ua}} e^{-\Phi_{ua}\left(x_{ua}\right)/k}} \qquad P(P^{ua}|P^{tor}) \leftarrow \frac{e^{-\Psi_{ua,tor}\left(x_{ua,tor}\right)/k}}{\sum_{x_{ua},x_{tor}} e^{-\Psi_{ua,tor}\left(x_{ua,tor}\right)/k}}$$

$$P(I|P^{ll}, C^{ll}) \leftarrow \frac{e^{-\Phi_{ll}\left(x_{ll}\right)/k}}{\sum_{x_{ll}} e^{-\Phi_{ll}\left(x_{ll}\right)/k}} \qquad P(P^{ll}|P^{ul}) \leftarrow \frac{e^{-\Psi_{ll,ul}\left(x_{ll,ul}\right)/k}}{\sum_{x_{ll},x_{ul}} e^{-\Psi_{ll,ul}\left(x_{ll,ul}\right)/k}}$$

$$P(I|P^{ul}, C^{ul}) \leftarrow \frac{e^{-\Phi_{ul}\left(x_{ul}\right)/k}}{\sum_{x_{ul}} e^{-\Phi_{ul}\left(x_{ul}\right)/k}} \qquad P(P^{ul}|P^{tor}) \leftarrow \frac{e^{-\Psi_{ul,tor}\left(x_{ul,tor}\right)/k}}{\sum_{x_{ul},x_{tor}} e^{-\Psi_{ul,tor}\left(x_{ul,tor}\right)/k}}$$

$$P(I|P^{h}, C^{h}) \leftarrow \frac{e^{-\Phi_{h}\left(x_{h}\right)/k}}{\sum_{x_{h}} e^{-\Phi_{h}\left(x_{h}\right)/k}} \qquad P(P^{h}|P^{tor}) \leftarrow \frac{e^{-\Psi_{h,tor}\left(x_{h,tor}\right)/k}}{\sum_{x_{h},x_{tor}} e^{-\Psi_{h,tor}\left(x_{h,tor}\right)/k}}$$

$$P(I|P^{tor}, C^{tor}) \leftarrow \frac{e^{-\Phi_{tor}\left(x_{tor}\right)/k}}{\sum_{x_{tor}} e^{-\Phi_{tor}\left(x_{tor}\right)/k}}$$

4. Calculate messages from lower body parts to upper body parts:

$$m_{la,ua}\left(x_{ua}\right) \leftarrow \sum_{x_{la}} P(I|P^{la}, C^{la})P(P^{la}|P^{ua}) \qquad m_{la,ua}\left(x_{ua}\right) \leftarrow \frac{m_{la,ua}\left(x_{ua}\right)}{\sum_{x_{ua}} m_{la,ua}\left(x_{ua}\right)}$$

$$m_{ll,ul}\left(x_{ul}\right) \leftarrow \sum_{x_{ll}} P(I|P^{ll}, C^{ll})P(P^{ll}|P^{ul}) \qquad m_{ll,ul}\left(x_{ul}\right) \leftarrow \frac{m_{ll,ul}\left(x_{ul}\right)}{\sum_{x_{ul}} m_{ll,ul}\left(x_{ul}\right)}$$

5. Calculate messages from upper body parts and head to torso:

$$m_{ua,tor}\left(x_{tor}\right) \leftarrow \sum_{x_{ua}} P(I|P^{ua},C^{ua})P(P^{ua}|P^{tor})m_{la,ua}\left(x_{ua}\right) \qquad m_{ua,tor}\left(x_{tor}\right) \leftarrow \frac{m_{ua,tor}\left(x_{tor}\right)}{\sum_{x_{tor}} m_{ua,tor}\left(x_{tor}\right)}$$

$$m_{ul,tor}\left(x_{tor}\right) \leftarrow \sum_{x_{ul}} P(I|P^{ul},C^{ul})P(P^{ul}|P^{tor})m_{ll,ul}\left(x_{ul}\right) \qquad m_{ul,tor}\left(x_{ul}\right) \leftarrow \frac{m_{ul,tor}\left(x_{tor}\right)}{\sum_{x_{tor}} m_{ul,tor}\left(x_{tor}\right)}$$

$$m_{h,tor}\left(x_{tor}\right) \leftarrow \sum_{x_{h}} P(I|P^{h},C^{h})P(P^{h}|P^{tor}) \qquad m_{h,tor}\left(x_{tor}\right) \leftarrow \frac{m_{h,tor}\left(x_{tor}\right)}{\sum_{x_{tor}} m_{h,tor}\left(x_{tor}\right)}$$

6. Calculate torso (root) posterior as mixture or messages:

$$[b_{tor}(x_{tor})]_1 \leftarrow m_{ul,tor}\left(x_{tor}\right) m_{h,tor}\left(x_{tor}\right) \qquad [b_{tor}(x_{tor})]_1 \leftarrow \frac{[b_{tor}(x_{tor})]_1}{\sum_{x_{tor}} [b_{tor}(x_{tor})]_1}$$

$$[b_{tor}(x_{tor})]_2 \leftarrow m_{ul,tor}\left(x_{tor}\right) m_{h,tor}\left(x_{tor}\right) m_{ua,tor}\left(x_{tor}\right) \qquad [b_{tor}(x_{tor})]_2 \leftarrow \frac{[b_{tor}(x_{tor})]_2}{\sum_{x_{tor}} [b_{tor}(x_{tor})]_2}$$

$$b_{tor}\left(x_{tor}\right) \leftarrow P(I|P^{tor},C^{tor})\{[b_{tor}(x_{tor})]_1 + [b_{tor}(x_{tor})]_2\}$$

$$b_{tor}\left(x_{tor}\right) \leftarrow \frac{b_{tor}(x_{tor})}{\sum_{x_{tor}} b_{tor}(x_{tor})}$$

7. Sampling:
   - sample tor_idx from $b_{tor}\left(x_{tor}\right)$
   - sample ul_idx from $P(I|P^{ul},C^{ul})P(P^{ul}|P^{tor\_idx})m_{ll,ul}\left(x_{ul}\right)$
   - sample ll_idx from $P(I|P^{ll},C^{ll})P(P^{ll}|P^{ul\_idx}))$
   - sample ua_idx from $P(I|P^{ua},C^{ua})P(P^{ua}|P^{tor\_idx})m_{la,ua}\left(x_{ua}\right)$
   - sample la_idx from $P(I|P^{la},C^{la})P(P^{la}|P^{ua\_idx})$
   - sample h_idx from $P(I|P^{h},C^{h})P(P^{h}|P^{tor\_idx})$

5.2.2 Two frame pictorial structure

We aim to improve the detection result obtained with the single frame pictorial structure algorithm (without smoothing). Therefore, we introduce two variants of temporal graphical models, which include the detection from the previous frame.

Algorithm 1

First, we add only the previous part detection (obtained from the mode finding procedure) as parents to the current body part nodes, as shown in Figure 5.7. This allows us to introduce the motion model prior in the posterior from which a body part is sampled as:

$$P\left(P_t^i, I_t|P_t^{\pi(i)\_idx}, P_{t-1}^i, C^i\right) \propto P\left(I_t|P_t^{\pi(i)\_idx}, C^{\pi(i)\_idx}\right) P\left(P_t^i|P_t^{\pi(i)\_idx}\right) P\left(P_t^i|P_{t-1}^i\right) \prod_{j\in C(i)} m_{j,i}\left(x_i\right), \tag{5.2.2}$$

where P_{t-1}^i is the body part detection from the previous frame.

The algorithm steps are the following:

1. Find body part candidates based on shape and colour appearance:
   1.1. Calculate singleton potentials $\Phi_i\left(x_i\right)$ for $i \in \{tor, la, ua, ll, ul, h\}$.
   1.2. Retain candidates for which $\Phi_i\left(x_i\right) \geq thresh_i$ for $i \in \{tor, la, ua, ll, ul, h\}$.

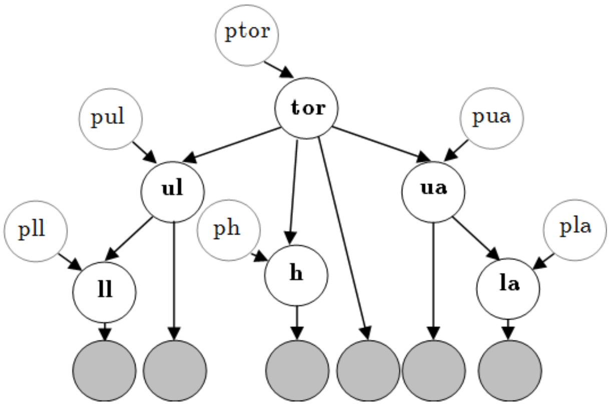


Figure 5.7: Two frame pictorial structure with previous parts detections as parents

2. Calculate pairwise potentials:

    2.1. Kinematic constraints $\Psi_{i,\pi(i)}\left(x_{i,\pi(i)}\right)$, where $i \in \{tor, la, ua, ll, ul, h\}$ and $\pi(i)$ is the parent of part i.

    2.2. Motion model $P(P_t^i|P_{t-1}^i) = \Psi_{i,prev\_i}\left(x_{i,prev\_i}\right)$, where $i \in \{tor, la, ua, ll, ul, h\}$ and $prev\_i$ is the body part detection in the previous frame.

3.-5. Similar to single frame pictorial structure algorithm.

6. Calculate torso (root) posterior as mixture or messages:

$$[b_{tor}(x_{tor})]_1 \leftarrow m_{ul,tor}\left(x_{tor}\right) m_{h,tor}\left(x_{tor}\right) \qquad [b_{tor}(x_{tor})]_1 \leftarrow \frac{[b_{tor}(x_{tor})]_1}{\sum_{x_{tor}} [b_{tor}(x_{tor})]_1}$$

$$[b_{tor}(x_{tor})]_2 \leftarrow m_{ul,tor}\left(x_{tor}\right) m_{h,tor}\left(x_{tor}\right) m_{ua,tor}\left(x_{tor}\right) \qquad [b_{tor}(x_{tor})]_2 \leftarrow \frac{[b_{tor}(x_{tor})]_2}{\sum_{x_{tor}} [b_{tor}(x_{tor})]_2}$$

$$b_{tor}\left(x_{tor}\right) \leftarrow P(I_t|P_t^{tor}, C^{tor})P(P_t^{tor}|P_{t-1}^{tor})\{[b_{tor}(x_{tor})]_1 + [b_{tor}(x_{tor})]_2\}$$

$$b_{tor}\left(x_{tor}\right) \leftarrow \frac{b_{tor}(x_{tor})}{\sum_{x_{tor}} b_{tor}(x_{tor})}$$

7. Sampling:

- sample tor_idx from $b_{tor}\left(x_{tor}\right)$
- sample ul_idx from $P(I_t|P_t^{ul}, C^{ul})P(P_t^{ul}|P_t^{tor\_idx})P(P_t^{ul}|P_{t-1}^{ul})m_{ll,ul}\left(x_{ul}\right)$
- sample ll_idx from $P(I_t|P_t^{ll}, C^{ll})P(P_t^{ll}|P_t^{ul\_idx})P(P_t^{ll}|P_{t-1}^{ll})$
- sample ua_idx from $P(I_t|P_t^{ua}, C^{ua})P(P_t^{ua}|P_t^{tor\_idx})P(P_t^{ua}|P_{t-1}^{ua})m_{la,ua}\left(x_{ua}\right)$
- sample la_idx from $P(I_t|P_t^{la}, C^{la})P(P_t^{la}|P_t^{ua\_idx})P(P_t^{la}|P_{t-1}^{la})$
- sample h_idx from $P(I_t|P_t^{h}, C^{h})P(P_t^{h}|P_t^{tor\_idx})P(P_t^{h}|P_{t-1}^{h})$

Algorithm 2

Second, we add all the previously sampled body part candidates as parents to the current body part nodes, as shown in Figure 5.8. This allows us to calculate the posterior from which a body part is sampled as:

$$P\left(P_t^i, I_t | P_t^{\pi(i)\_idx}, P_{t-1}^i, C^i\right) \propto P\left(I_t | P_t^{\pi(i)\_idx}, C^{\pi(i)\_idx}\right) P\left(P_t^i | P_t^{\pi(i)\_idx}\right) \sum_{x_{i_{t-1}}} \left[P(P_t^i|P_{t-1}^i) P(I_{t-1}|P_{t-1}^i, C^i)\right] \prod_{j \in C(i)} m_{j,i}\left(x_i\right), \tag{5.2.3}$$

where we have averaged over all previously sampled candidates $x_{i_{t-1}}$, to obtain the motion model prior. This difference reflects in Steps 6 and 7 of Algorithm 2, compared to Steps 6 and 7 of Algorithm 1.

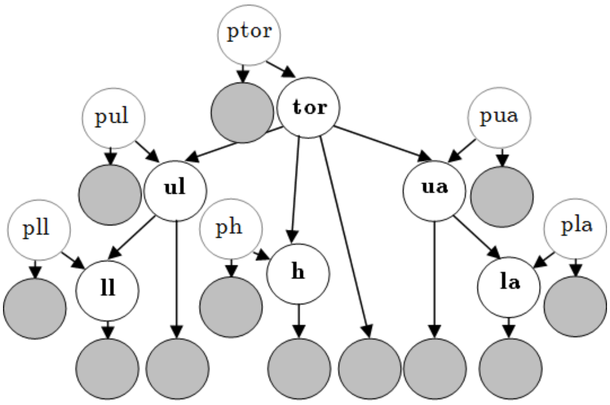


FIGURE 5.8: Two frame pictorial structure with previous parts candidates as parents

The algorithm steps are the following:

1.-5. Similar to Algorithm 1.

6. Calculate torso (root) posterior as mixture or messages:

$$[b_{tor}(x_{tor})]_1 \leftarrow m_{ul,tor}\left(x_{tor}\right) m_{h,tor}\left(x_{tor}\right) \qquad [b_{tor}(x_{tor})]_1 \leftarrow \frac{[b_{tor}(x_{tor})]_1}{\sum_{x_{tor}} [b_{tor}(x_{tor})]_1}$$

$$[b_{tor}(x_{tor})]_2 \leftarrow m_{ul,tor}\left(x_{tor}\right) m_{h,tor}\left(x_{tor}\right) m_{ua,tor}\left(x_{tor}\right) \qquad [b_{tor}(x_{tor})]_2 \leftarrow \frac{[b_{tor}(x_{tor})]_2}{\sum_{x_{tor}} [b_{tor}(x_{tor})]_2}$$

$$b_{tor}\left(x_{tor}\right) \leftarrow P(I_t|P_t^{tor}, C^{tor}) \sum_{x_{tor_{t-1}}} \left[P(P_t^{tor}|P_{t-1}^{tor}) P(I_{t-1}|P_{t-1}^{tor}, C^{tor})\right] \{[b_{tor}(x_{tor})]_1 + [b_{tor}(x_{tor})]_2\}$$

$$b_{tor}\left(x_{tor}\right) \leftarrow \frac{b_{tor}(x_{tor})}{\sum_{x_{tor}} b_{tor}(x_{tor})}$$

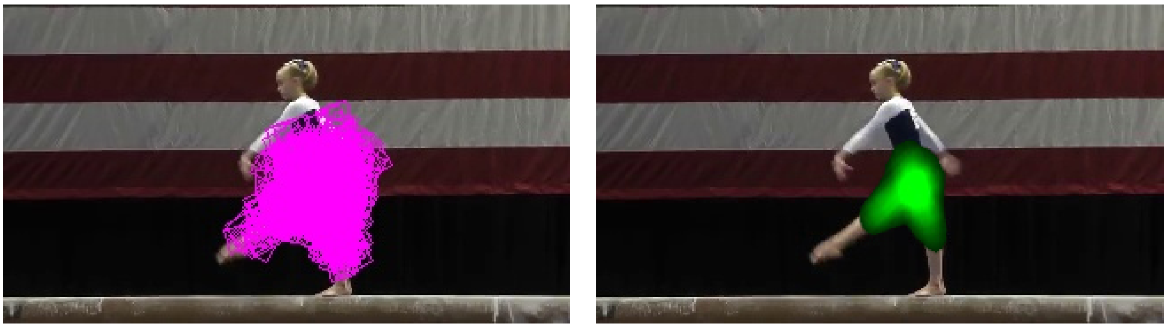
7. Sampling:

- sample tor_idx from $b_{tor}\left(x_{tor}\right)$
- sample ul_idx from $P(I_t|P_t^{ul}, C^{ul}) P(P_t^{ul}|P_t^{tor\_idx}) \sum_{x_{ul_{t-1}}} \left[P(P_t^{ul}|P_{t-1}^{ul}) P(I_{t-1}|P_{t-1}^{ul}, C^{ul})\right] m_{ll,ul}\left(x_{ul}\right)$
- sample ll_idx from $P(I_t|P_t^{ll}, C^{ll}) P(P_t^{ll}|P_t^{ul\_idx}) \sum_{x_{ll_{t-1}}} \left[P(P_t^{ll}|P_{t-1}^{ll}) P(I_{t-1}|P_{t-1}^{ll}, C^{ll})\right]$
- sample ua_idx from $P(I_t|P_t^{ua}, C^{ua}) P(P_t^{ua}|P_t^{tor\_idx}) \sum_{x_{ua_{t-1}}} \left[P(P_t^{ua}|P_{t-1}^{ua}) P(I_{t-1}|P_{t-1}^{ua}, C^{ua})\right] m_{la,ua}\left(x_{ua}\right)$
- sample la_idx from $P(I_t|P_t^{la}, C^{la}) P(P_t^{la}|P_t^{ua\_idx}) \sum_{x_{la_{t-1}}} \left[P(P_t^{la}|P_{t-1}^{la}) P(I_{t-1}|P_{t-1}^{la}, C^{la})\right]$
- sample h_idx from $P(I_t|P_t^{h}, C^{h}) P(P_t^{h}|P_t^{tor\_idx}) \sum_{x_{h_{t-1}}} \left[P(P_t^{h}|P_{t-1}^{h}) P(I_{t-1}|P_{t-1}^{h}, C^{h})\right]$

5.3 Modes

The 1000 samples obtained for each body part in the single arm, single leg pictorial structure constitute the feature space for that specific body part. Figure 5.9(a) shows an example of 1000 sampled upper legs drawn on the input frame. Figure 5.9(b) is an alternative representation to show the density distribution obtained from these samples.

The significant features, or the two upper leg detections that we need to find in this case, will be at the dense regions (or clusters) of the feature space. Finding these clusters can be done using a density estimation-based nonparametric clustering approach. The main idea is that the feature space is regarded as the empirical probability density function (p.d.f.) of the represented parameter, such that dense regions correspond to the local maxima of the p.d.f., or the *modes* of the unknown density.



(a)

(b)

FIGURE 5.9: (a) 1000 upper legs sampled from the posterior, depicted as purple rectangles. (b) Probability density function obtained from the 1000 upper leg samples.

5.3.1 Mean shift algorithm

The employed mode detection procedure by [Ramanan et al., 2007] is the mean shift algorithm [Comaniciu and Meer, 2002]. We derive this algorithm by first introducing *kernel estimation* as a nonparametric method for probability density estimation. For an arbitrary set of n data points $\{x_i\}_{i=1,\dots,n}$ in the d-dimensional space $\mathbb{R}^d$, the multivariate *kernel density estimator* with kernel $K(x)$ and windows radius (or bandwidth) h, in the point x is defined as:

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right). \tag{5.3.1}$$

The *profile* of a kernel K is a function $k : [0, \infty) \to \mathbb{R}$ such that $K(x) = c_{k,d}k\left(\|x\|^2\right)$, where $c_{k,d}$ is a strictly positive normalisation constant that makes $K(x)$ integrate to 1. Using the profile notation, the density estimator in equation (5.3.1) can be written as:

$$\hat{f}_{h,K}(x) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^{n} k\left(\left\|\frac{x - x_i}{h}\right\|^2\right). \tag{5.3.2}$$

Assuming that the derivative of the kernel profile k exists for all $x \in [0, \infty)$, except for a finite set of points, the profile g can be defined as $g(x) = -k'(x)$. A kernel G can now be defined as $G(x) = c_{g,d}g\left(\|x\|^2\right)$.

The modes are located at the zeros of the gradient $\nabla f(x) = 0$. The estimate of the density gradient can be defined as the gradient of kernel density estimate:

$$\hat{\nabla} f_{h,K}(x) \equiv \nabla \hat{f}_{h,K}(x) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (x - x_i) k' \left(\left\| \frac{x - x_i}{h} \right\|^2 \right) \tag{5.3.3}$$

$$= \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (x_i - x) g \left(\left\| \frac{x - x_i}{h} \right\|^2 \right) \tag{5.3.4}$$

$$= \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} g \left(\left\| \frac{x - x_i}{h} \right\|^2 \right) \right] \left[\frac{\sum_{i=1}^{n} x_i g \left(\left\| \frac{x-x_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g \left(\left\| \frac{x-x_i}{h} \right\|^2 \right)} - x \right]. \tag{5.3.5}$$

The first term in equation (5.3.5) can be written in relation to the density estimate computed with kernel G as:

$$\hat{f}_{h,G}(x) = \frac{c_{g,d} h^2}{2c_{k,d}} \cdot \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} g \left(\left\| \frac{x - x_i}{h} \right\|^2 \right) \right]. \tag{5.3.6}$$

The last term in equation (5.3.5) is the sample mean shift vector,

$$m_{h,G}(x) \equiv \left[\frac{\sum_{i=1}^{n} x_i g \left(\left\| \frac{x-x_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g \left(\left\| \frac{x-x_i}{h} \right\|^2 \right)} - x \right], \tag{5.3.7}$$

or the difference between the weighted mean, where the weights are given by kernel G, and the centre of the kernel window x. It can be seen that the maximum value for $\hat{f}_{h,K}$ will be attained when $m_{h,G}(x) = 0$.

Replacing (5.3.6) and (5.3.7) in (5.3.5) results in:

$$\hat{\nabla} f_{h,K}(x) = \hat{f}_{h,G}(x) \frac{2c_{k,d}}{c_{g,d} h^2} m_{h,G}(x) \Rightarrow m_{h,G}(x) = \frac{c_{g,d} h^2}{2c_{k,d}} \cdot \frac{\hat{\nabla} f_{h,K}(x)}{\hat{f}_{h,G}(x)}, \tag{5.3.8}$$

which shows that the mean shift vector computed with kernel G is proportional to the normalized density gradient estimate obtained with kernel K, meaning that the mean shift vector points towards the maximum increase in the density. The mean shift vector can lead to the stationary point of the estimated density, which represents the mode.

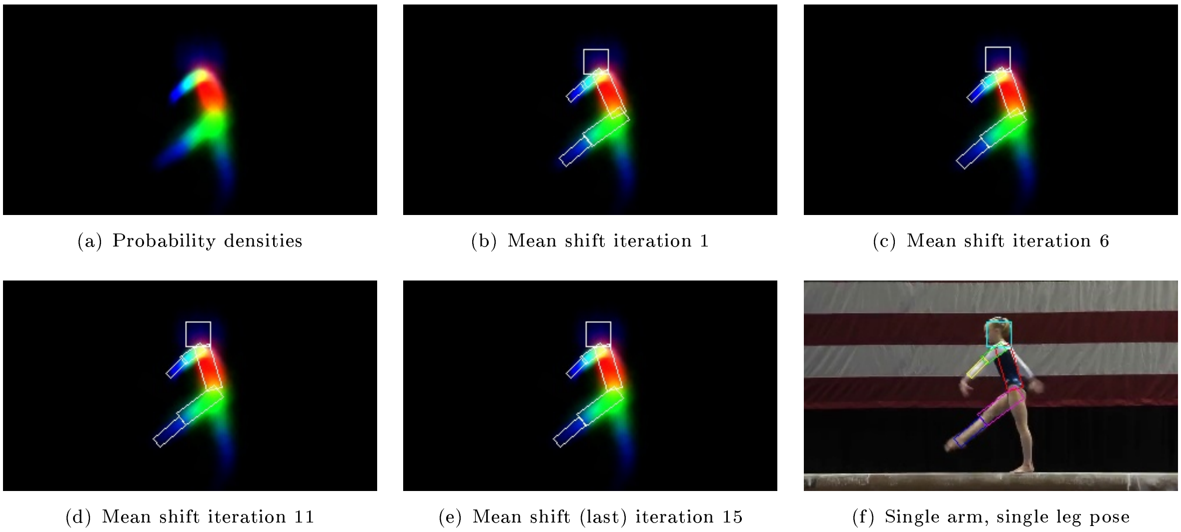
The mean shift procedure thus consists of the following steps, repeated until convergence:

1. Fix a (kernel) window $G(x)$ around every data point.
2. Calculate the mean shift vector $m_{h,G}(x)$ within that window.
3. Translate the (kernel) window $G(x)$ by $m_{h,G}(x)$.

The convergence of this algorithm is demonstrated in [Comaniciu and Meer, 2002].

Implementation The feature space employed by [Ramanan et al., 2007] consists of the 2D end points coordinates for each sampled body part. The chosen kernel, G, is the uniform kernel, the metric is the Euclidean distance and the chosen bandwidth h is the torso length (which is the largest body part length). The starting point x is the feature vector which tessellates the entire feature space by kernels of the given bandwidth.

Figure 5.10(a) shows the different body part distributions obtained from the 1000 samples, which are colour coded as follows: red for the torso, green for the upper arms and upper legs, blue for the lower arms, lower legs and the head. Finding the single arm, single leg pose means finding the first mode from each of these distributions. This is achieved in a relatively small number of iterations, as can be seen in Figure 5.10.



(a) Probability densities

(b) Mean shift iteration 1

(c) Mean shift iteration 6

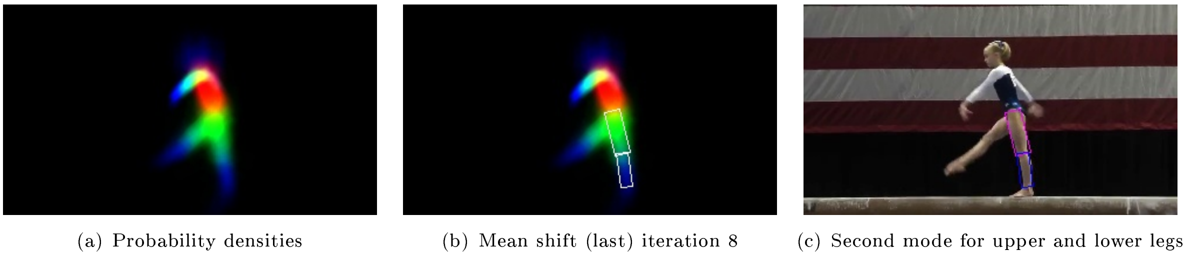
(d) Mean shift iteration 11

(e) Mean shift (last) iteration 15

(f) Single arm, single leg pose

Figure 5.10: Mean shift on posterior distributions to find single arm, single leg pose modes.

The second mode on the arms and legs distributions will fall on the second limb. The search for the second mode is done on samples that do not belong to the first mode. [Ramanan et al., 2007] impose that a certain fraction of the total number of samples lay next to this second mode. This is motivated by the fact that the number of remaining samples might be low, thus forming a weak distribution. A mode found on this distribution would not be as prominent as the first one. Figure 5.11 shows the result of the mean shift procedure on the distribution obtained from the leg samples which do not belong to the first modes (for the first upper leg and lower leg, respectively).



(a) Probability densities

(b) Mean shift (last) iteration 8

(c) Second mode for upper and lower legs

Figure 5.11: Mean shift on posterior distributions to find the second mode for upper and lower legs

6 Experimentation

6.1 Method analysis

6.1.1 Size influence on candidate detection

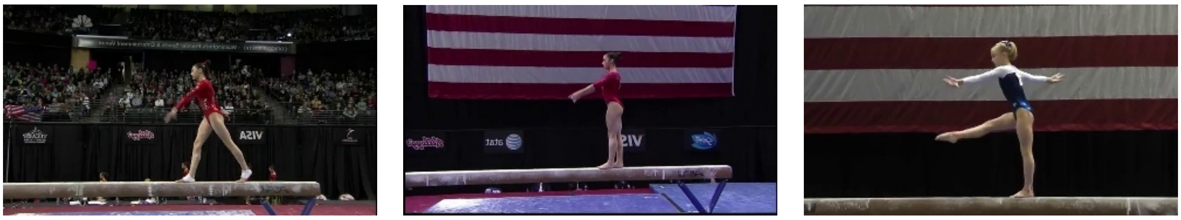
We study the relationship between a body part's size and the quality of the search results in the detection module of the tracking system (body part size and appearance parameters known, as described in Section 5.1). We hypothesize that the larger a body part is, the more reliable the detection.

Experiment setup

We test this hypothesis by searching for a person in three different sequences consisting of ten frames each. To obtain the ground truth, we first inspect the detection results obtained with the two frame pictorial structure algorithm 1. If the system outputs visually correct final body pose detections (in which the rectangles cover the actual body parts), then we consider these as the ground truth. This automatic step is useful to save time in the process of labelling the ground truth data. In case the body pose detection is not entirely correct, meaning it has wrong or undetected limbs, we manually label those parts (by providing the body part parameters: centre coordinates (x, y) and orientation θ, which we approximate visually).

In our evaluation, a candidate is considered to be *correct* if its centre deviates less than 40% of the body part diagonal length from the ground truth centre coordinates. We chose this measure because we found that the candidates that abide it manage to cover (a significant part of) the actual body parts.

A sample image from each of the three different sequences is provided in Figure 6.1.



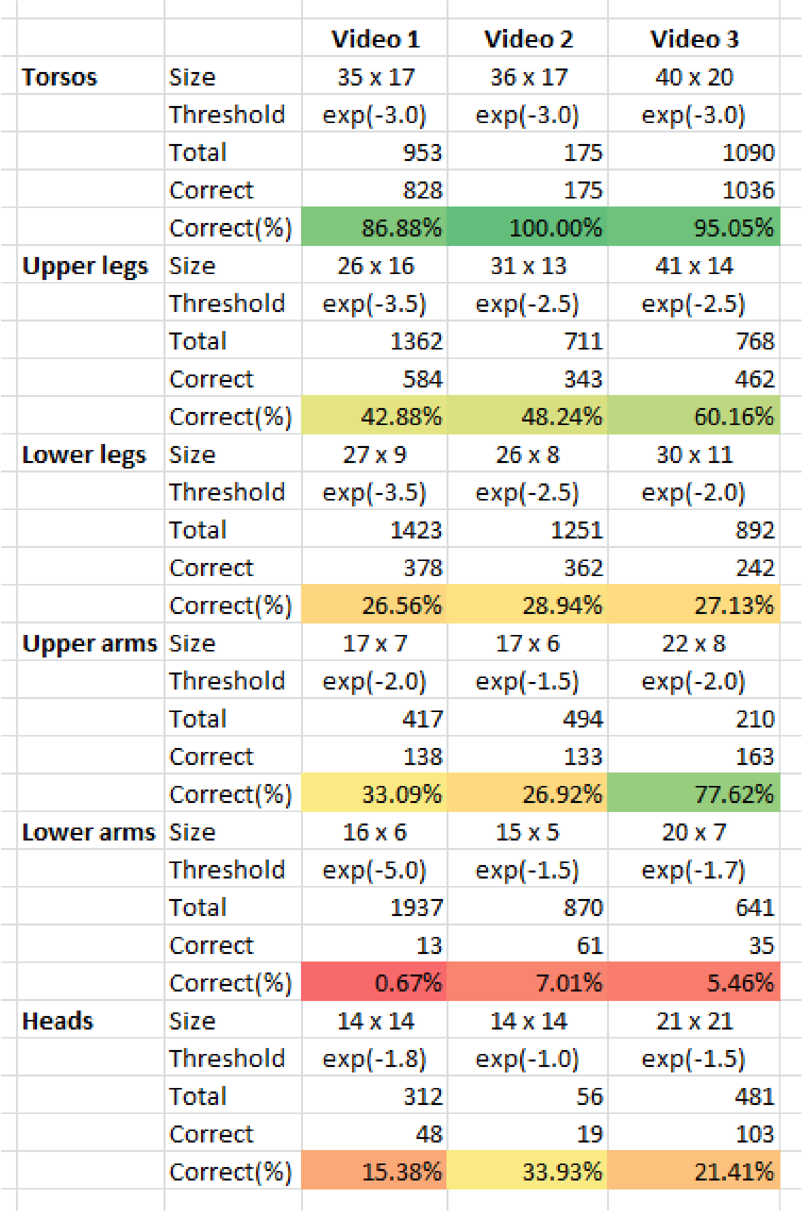
(a) Video 1

(b) Video 2

(c) Video 3

Figure 6.1: Sample image from each test sequences

Table 6.1: Search precision for candidate body parts using appearance and shape models



		Video 1	Video 2	Video 3
Torsos	Size	35 x 17	36 x 17	40 x 20
	Threshold	exp(-3.0)	exp(-3.0)	exp(-3.0)
	Total	953	175	1090
	Correct	828	175	1036
	Correct(%)	86.88%	100.00%	95.05%
Upper legs	Size	26 x 16	31 x 13	41 x 14
	Threshold	exp(-3.5)	exp(-2.5)	exp(-2.5)
	Total	1362	711	768
	Correct	584	343	462
	Correct(%)	42.88%	48.24%	60.16%
Lower legs	Size	27 x 9	26 x 8	30 x 11
	Threshold	exp(-3.5)	exp(-2.5)	exp(-2.0)
	Total	1423	1251	892
	Correct	378	362	242
	Correct(%)	26.56%	28.94%	27.13%
Upper arms	Size	17 x 7	17 x 6	22 x 8
	Threshold	exp(-2.0)	exp(-1.5)	exp(-2.0)
	Total	417	494	210
	Correct	138	133	163
	Correct(%)	33.09%	26.92%	77.62%
Lower arms	Size	16 x 6	15 x 5	20 x 7
	Threshold	exp(-5.0)	exp(-1.5)	exp(-1.7)
	Total	1937	870	641
	Correct	13	61	35
	Correct(%)	0.67%	7.01%	5.46%
Heads	Size	14 x 14	14 x 14	21 x 21
	Threshold	exp(-1.8)	exp(-1.0)	exp(-1.5)
	Total	312	56	481
	Correct	48	19	103
	Correct(%)	15.38%	33.93%	21.41%

Results Table 6.1 shows the results of our experimentation, averaged over the ten frames. For each video and each body part in the single arm, single leg pictorial structure, we list the known size, the singleton potential $\Phi_i(x_i)$ threshold for a candidate to be considered valid, the total number of retrieved candidates (which scored better than the threshold), the number of correctly retrieved candidates (as explained above) and the search precision. We colour code the precision with green for high values, yellow for in-between values and red for small values.

Torso and legs We find high precision values for the torso, which means that the method is guaranteed to find a person in a frame, if the person exists and its torso colour appearance model was learnt. We notice that, for each video, the precision decreases with the size of the body part: from torso to upper legs and lower legs, respectively. This result is in agreement with our hypothesis.

Arms The upper arms tend to have a search precision as high as the lower legs, with the exception of the upper arms in video 3. We assume this is due to the fact that the upper arms in video 3 are white and the torso is blue and no upper arm candidates are detected on the torso. We observe that a significant number of incorrect upper arm candidates in the other videos tend to lie on the torso, because of the colour appearance.

For each video, the precision of the upper arm search is higher than the precision of the lower arm search, which is in agreement with our hypothesis.

Because the arm detection seems to be more difficult than the other body parts detection, we also look at how the degree of similarity between the arms and the other body parts affects the search precision. We list the following properties of our input sequences:

1. In video 1, upper and lower arms are similar in appearance to the torso.
2. In video 2, the upper arm is similar in appearance to the torso and the lower arm is similar in appearance to the legs and head (skin colour).
3. In video 3, the upper and lower arms are similar in appearance to each other but different from the other body parts.
4. In videos 1 and 2 one arm is clearly visible and the second arm is fully occluded at all times. In video 3, both arms are clearly visible at all times.
5. The arms are far away from the rest of the body, such that the arm candidates laying on other areas of the body are not taken into account as false positives.

Table 6.1 only contains candidates that were found on the body. Those candidates found in the background (only in the case of the second video) were removed, so that we focus our results on the arm colour - body colour relationship.

For upper arms in videos 1 and 2, where the sleeve has the same colour as the suit, we find close precision percentages. For upper arms in video 3, where the sleeve has a different colour than the suit, we obtain high precision.

For the lower arms in videos 2 and 3, we chose the $\Phi_{lower\_arm}(x_{lower\_arm})$ thresholds in order to obtain a correct final full body pose. For the lower arms in video 1, which proved to be the most challenging to find, we lowered the $\Phi_{lower\_arm}(x_{lower\_arm})$ threshold to obtain some correct candidates. However, the strong (high score) candidates were found on the body, instead of on the arms, and due to the fact that they also lie within the kinematic constraints, they were included in the final body pose.

Head For each video, we found heads with a better precision than that of the lower arms. While the lengths of the head and lower arms are close, the difference between the widths seems to make the difference in the overall quality of the detection.

Full pose detection We observed that, to obtain a successful full body pose detection, the score of the correct detections is more important than their number with respect to the total number of retrieved candidates. For video 1 we were unable to find more than one correct full body pose, due to the low scores of the correct lower arm candidates. In videos 2 and 3, even though the precision for lower arms was small, we found the correct body pose in all tested frames, due to the high score of the few correct candidates. All body poses from the three sequences contained correct torsos and legs.

Conclusion

We found that the results were in agreement with our hypothesis. However, as the size of the body part becomes smaller, we consider that the degree to which the part is similar to other parts in the body is also a factor that influences the search precision. We illustrated this case for the lower and upper arms.

For challenging situations, like video 3, where the lower arm was covered in a long sleeve of the same colour as the suit, we recommend raising the score threshold for the lower arms, such that incorrect candidates that lay elsewhere on the body, but have good scores, will not be encountered in the final pose.

6.1.2 Robustness

In this section, we analyse the robustness with respect to input parameters of the model building module, responsible for finding an accurate lateral walking pose. We wish to verify if we can find one set of parameters to fit our input videos, similarly to [Ramanan et al., 2007]. As the correctness of the lateral walking pose is crucial for the success of the detection module, we assume that chamfer score thresholds in detecting candidate body parts need to be accurately set for different videos.

In this experiment, we refer to the body part candidate search using rectangular filters, described in Section 4.1.

Experiment setup

We choose one frame from three distinct videos, with the following properties:

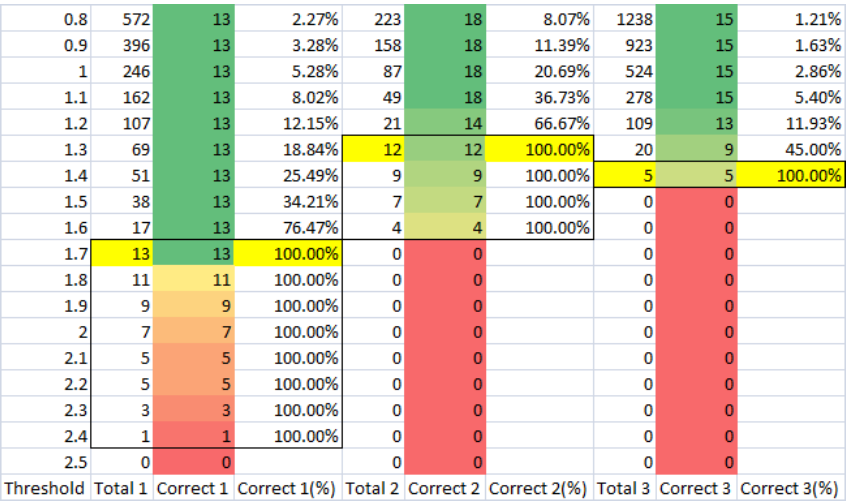
- the image depicts a lateral walking pose
- our system is able to clearly detect the lateral walking pose

We use the second property to determine the ground truth. For each of the three frames, we determine by trial and error a set of parameters that results in a successful full body pose detection. Then, for each different frame, we vary the chamfer score thresholds in the same way. In our implementation, we use three thresholds: one for the torso, one for the legs (same for upper/lower legs) and one for the arms (same for upper/lower arms).

For each value of the chamfer score thresholds and their corresponding body parts, we count the number of *correct* detections, as well as the total number of detections. We consider a detection to be correct if its centre deviates less than 10% of the body part diagonal length from the ground truth. We restrict this error to only 10% because is it essential that the appearance models are learnt from correct body part detections.

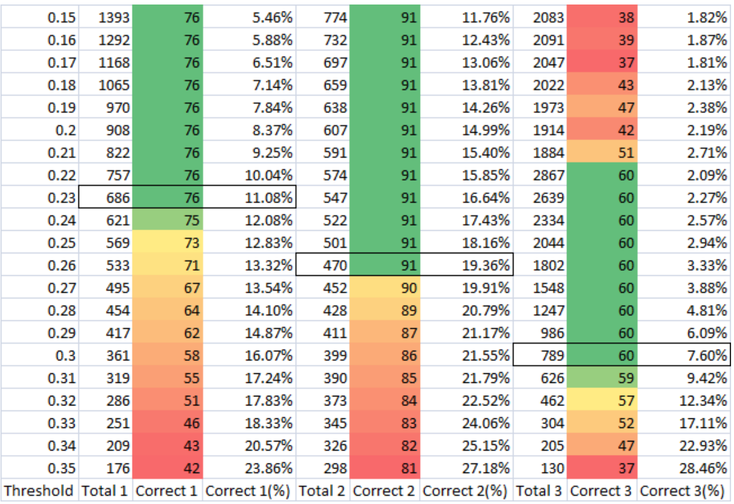
Results For each tested frame, we list in Table 6.2 the following: the known size, the singleton potential $\Phi_i(x_i)$ threshold for a candidate to be considered valid, the total number of retrieved candidates (which

TABLE 6.2: Search precision for candidate body parts using rectangular templates



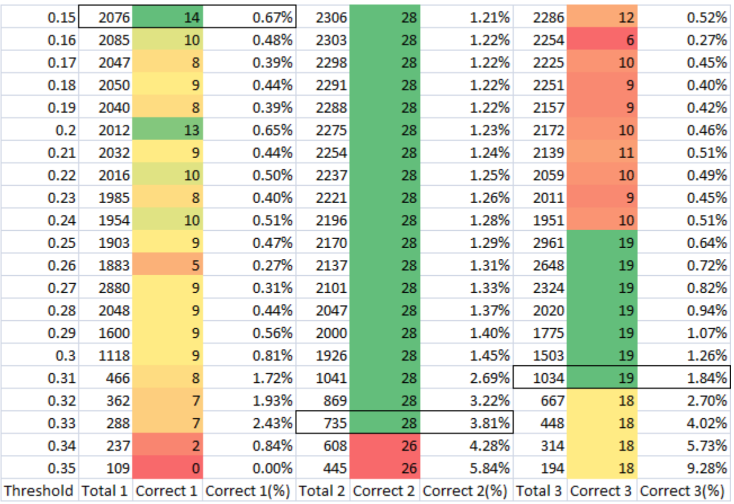
0.8	572	13	2.27%	223	18	8.07%	1238	15	1.21%
0.9	396	13	3.28%	158	18	11.39%	923	15	1.63%
1	246	13	5.28%	87	18	20.69%	524	15	2.86%
1.1	162	13	8.02%	49	18	36.73%	278	15	5.40%
1.2	107	13	12.15%	21	14	66.67%	109	13	11.93%
1.3	69	13	18.84%	12	12	100.00%	20	9	45.00%
1.4	51	13	25.49%	9	9	100.00%	5	5	100.00%
1.5	38	13	34.21%	7	7	100.00%	0	0	
1.6	17	13	76.47%	4	4	100.00%	0	0	
1.7	13	13	100.00%	0	0		0	0	
1.8	11	11	100.00%	0	0		0	0	
1.9	9	9	100.00%	0	0		0	0	
2	7	7	100.00%	0	0		0	0	
2.1	5	5	100.00%	0	0		0	0	
2.2	5	5	100.00%	0	0		0	0	
2.3	3	3	100.00%	0	0		0	0	
2.4	1	1	100.00%	0	0		0	0	
2.5	0	0		0	0		0	0	
Threshold	Total 1	Correct 1	Correct 1(%)	Total 2	Correct 2	Correct 2(%)	Total 3	Correct 3	Correct 3(%)

(a) Search results for torso



0.15	1393	76	5.46%	774	91	11.76%	2083	38	1.82%
0.16	1292	76	5.88%	732	91	12.43%	2091	39	1.87%
0.17	1168	76	6.51%	697	91	13.06%	2047	37	1.81%
0.18	1065	76	7.14%	659	91	13.81%	2022	43	2.13%
0.19	970	76	7.84%	638	91	14.26%	1973	47	2.38%
0.2	908	76	8.37%	607	91	14.99%	1914	42	2.19%
0.21	822	76	9.25%	591	91	15.40%	1884	51	2.71%
0.22	757	76	10.04%	574	91	15.85%	2867	60	2.09%
0.23	686	76	11.08%	547	91	16.64%	2639	60	2.27%
0.24	621	75	12.08%	522	91	17.43%	2334	60	2.57%
0.25	569	73	12.83%	501	91	18.16%	2044	60	2.94%
0.26	533	71	13.32%	470	91	19.36%	1802	60	3.33%
0.27	495	67	13.54%	452	90	19.91%	1548	60	3.88%
0.28	454	64	14.10%	428	89	20.79%	1247	60	4.81%
0.29	417	62	14.87%	411	87	21.17%	986	60	6.09%
0.3	361	58	16.07%	399	86	21.55%	789	60	7.60%
0.31	319	55	17.24%	390	85	21.79%	626	59	9.42%
0.32	286	51	17.83%	373	84	22.52%	462	57	12.34%
0.33	251	46	18.33%	345	83	24.06%	304	52	17.11%
0.34	209	43	20.57%	326	82	25.15%	205	47	22.93%
0.35	176	42	23.86%	298	81	27.18%	130	37	28.46%
Threshold	Total 1	Correct 1	Correct 1(%)	Total 2	Correct 2	Correct 2(%)	Total 3	Correct 3	Correct 3(%)

(b) Search results for legs



0.15	2076	14	0.67%	2306	28	1.21%	2286	12	0.52%
0.16	2085	10	0.48%	2303	28	1.22%	2254	6	0.27%
0.17	2047	8	0.39%	2298	28	1.22%	2225	10	0.45%
0.18	2050	9	0.44%	2291	28	1.22%	2251	9	0.40%
0.19	2040	8	0.39%	2288	28	1.22%	2157	9	0.42%
0.2	2012	13	0.65%	2275	28	1.23%	2172	10	0.46%
0.21	2032	9	0.44%	2254	28	1.24%	2139	11	0.51%
0.22	2016	10	0.50%	2237	28	1.25%	2059	10	0.49%
0.23	1985	8	0.40%	2221	28	1.26%	2011	9	0.45%
0.24	1954	10	0.51%	2196	28	1.28%	1951	10	0.51%
0.25	1903	9	0.47%	2170	28	1.29%	2961	19	0.64%
0.26	1883	5	0.27%	2137	28	1.31%	2648	19	0.72%
0.27	2880	9	0.31%	2101	28	1.33%	2324	19	0.82%
0.28	2048	9	0.44%	2047	28	1.37%	2020	19	0.94%
0.29	1600	9	0.56%	2000	28	1.40%	1775	19	1.07%
0.3	1118	9	0.81%	1926	28	1.45%	1503	19	1.26%
0.31	466	8	1.72%	1041	28	2.69%	1034	19	1.84%
0.32	362	7	1.93%	869	28	3.22%	667	18	2.70%
0.33	288	7	2.43%	735	28	3.81%	448	18	4.02%
0.34	237	2	0.84%	608	26	4.28%	314	18	5.73%
0.35	109	0	0.00%	445	26	5.84%	194	18	9.28%
Threshold	Total 1	Correct 1	Correct 1(%)	Total 2	Correct 2	Correct 2(%)	Total 3	Correct 3	Correct 3(%)

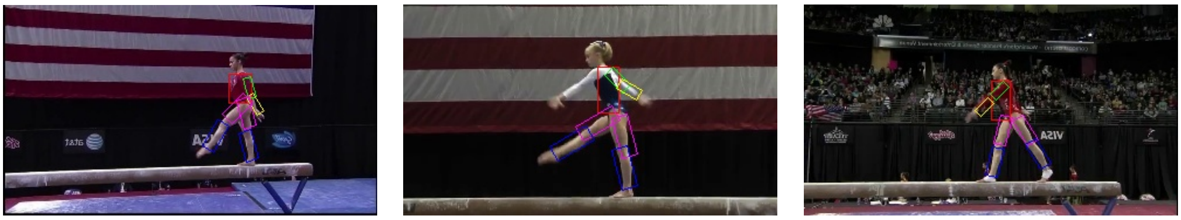
(c) Search results for arms

scored better than the threshold), the number of correctly retrieved candidates (as explained before) and the search precision.

Torso Table 6.2 shows that 100% detection precision for the torso is possible. However, this does occur at different chamfer score thresholds, for the three different frames.

Legs and arms For the legs and arms we look at the maximum number of correct candidates (highlighted in green). We see that the green stripes do not coincide for the three distinct frames.

Full body pose For the final detection, we wish to include as many correct torsos as possible and the maximum number of correct legs and arms. As the message passing from the tree inference algorithm is quadratic in the number of candidates, we also wish to have a low total number of candidates. From the three tables in Table 6.2, we select those values that meet the above conditions. The values are bordered and highlighted (for the torso) in the table. Figure 6.3 shows that these parameters lead to good lateral walking pose detections.



(d) Video 1 (e) Video 2 (f) Video 3

Figure 6.3: Lateral walking poses obtained with experimentally determined chamfer score tresholds.

Conclusion

We showed that, unlike [Ramanan et al., 2007]'s results, one set of values for the three chamfer score thresholds does not suit all input videos. We also provided some guidelines for choosing these values - leaving from a trial and error set of parameters, it is possible to refine then by choosing those chamfer score thresholds that results in 100% precision for the torso and in the highest precision corresponding to the maximum possible number of correct candidates. Finding a unique set of thresholds could be possible for videos which share a set of common features, but could come at a cost on computation time, if the total number of candidates includes a high ratio of incorrect to correct candidates.

6.2 Contributions

6.2.1 Two frame pictorial structure

Motion model implementation

In Section 5.2.2 we described in detail two algorithms in which we introduced the previous body part detection (Algorithm 1) or the previously sampled body part candidates (Algorithm 2) as nodes in the graphical model.

Calculating the motion model prior for the head and the torso is straightforward. But, because we are implementing [Ramanan et al., 2007]'s single arm, single leg pictorial structure, we have to deal with left/right ambiguity in the arms and legs. In other words, for a current leg or arm candidate, we cannot know for sure which is the corresponding detection from the previous frame. So, for the arms and legs, our implementation of the motion model for Algorithm 1 follows these rules:

1. If no previously detected arms or legs, $\Psi_{i,prev\_i}\left(x_{i,prev\_i}\right) = 1$.
2. If one arm or one leg was detected, consider that part as x_{prev_i} and calculate $\Psi_{i,prev\_i}\left(x_{i,prev\_i}\right)$ according to the chosen motion model.
3. If two arms or two legs were detected, calculate $\Psi_{i,prev\_i}\left(x_{i,prev\_i}\right)$ for both x_{prev_i} and choose the highest value.

Step 1 ensures that the current detection is not compromised if the previous detection failed, and makes the method equivalent to the single frame pictorial structure algorithm described in Section 5.2.1. Step 3 tries to solve the left/right ambiguity by choosing as previous node the detection or candidate that is closest to the current candidate.

Step 2 covers the case where one arm or one leg is occluded. However, if in the current frame the second limb becomes visible, it might not validate the motion model, as it is calculated in relationship with the other limb that was visible in the previous frame. When using the bounded velocity motion model, we deal with this shortcoming by not setting the motion model prior to zero in equation (3.2.5), which would discard the candidates entirely. Instead, we set the motion model prior at a small number, such that the posterior is different from zero and the candidates could still be sampled in the sampling from the posterior phase. For the Gaussian noise motion model this is not necessary as we do not obtain absolute zero values for the prior.

For Algorithm 2, where the difference is that instead of the previous body part detection (obtained through the modes finding procedure over the distribution obtained from the 1000 samples), we marginalize over all previously sampled candidates. This means we calculate $P(P_t^i|P_{t-1}^i) = \Psi_{i,prev\_i}\left(x_{i,prev\_i}\right)$ for all corresponding previously sampled candidates, and obtain the motion model prior by averaging with the formula $\sum_{x_{i_{t-1}}}\left[P(P_t^i|P_{t-1}^i)P(I_{t-1}|P_{t-1}^i, C^i)\right]$.

Hypothesis For both algorithms shown in Section 5.2.2 and reffered to as Method 1 and Method 2 below, we expect that both the tracking quality and smoothness would improve, compared to the single frame pictorial structure algorithm.

Experiment setup

We run our two algorithms and the single frame pictorial structure algorithm on different videos and determine by trial and error the input parameters that give satisfactory results. The common input parameters that we need to adjust for all methods are:

- appearance thresholds per body part (see Section 5.1),
- second mode thresholds for arms and legs (see Section 5.3).

To this set of parameters we add v_{max} for the bounded velocity motion model. Due to the already large dimension of the parameter set, we use the same v_{max} for all body parts.

We use the bounded velocity motion model for Methods 1 and 2. For each comparison of two methods, we show results obtained with the same set of input parameters.

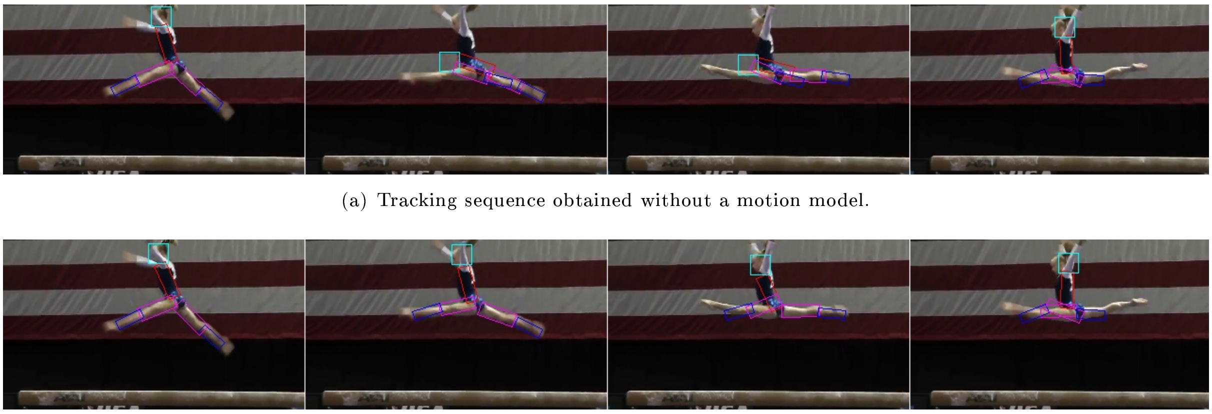
We choose to evaluate our results by comparing sequences of images that contain the pose detections. We compare Method 1 and Method 2 to the single frame pictorial structure algorithm, in turns. We consider this form of evaluation suitable to our purpose, because it allows us to identify the situations in which our method exceeds or fails, and it also gives the reader the chance to observe the differences between the detections obtained with the distinct methods. We choose relevant sequences which show both the advantages and disadvantages of each method.

For each sequence obtained with the motion model prior using Method 1, we manually labelled the initial frame. For each sequence obtained with the motion model prior using Method 2, the candidates from the initial frame were obtained using the single frame pictorial structure algorithm. The initialised frames are not shown in our result figures.

Method 1: adding the previous body part detection as a parent node to the current body part node

Results

Video 1 We first show our results on a clear and simple setting: the background is not cluttered, the body parts are distinctly coloured and the person occupies a large space in the image (torso area to frame area ratio is 1:80).

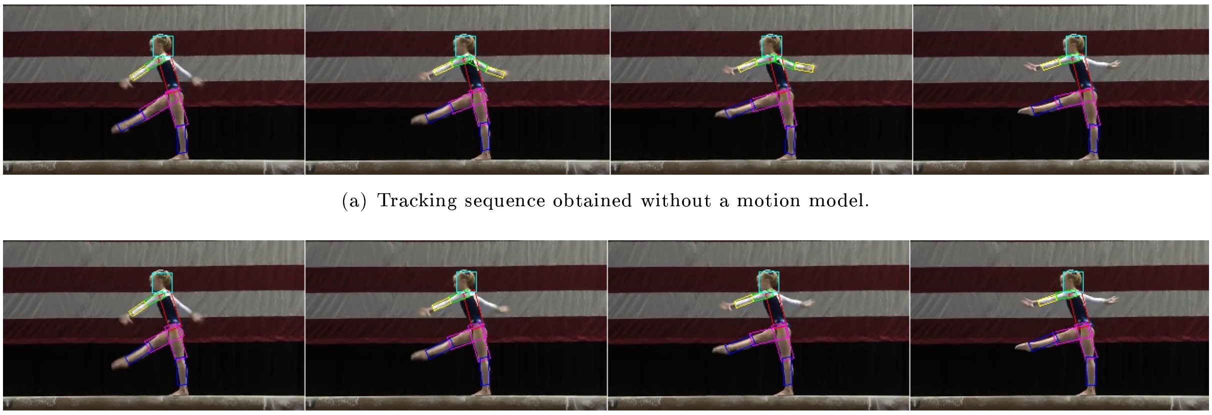


(a) Tracking sequence obtained without a motion model.

(b) Tracking sequence obtained with a motion model.

Figure 6.4

Figure 6.4 shows a challenging sequence: the movement is fast, blurred and the head is partially occluded by the arms. We found that, with the same set of input parameters for both methods, the motion model helps to maintain the torso and head track (in the middle two frames), unlike the single frame pictorial structure algorithm.



(a) Tracking sequence obtained without a motion model.

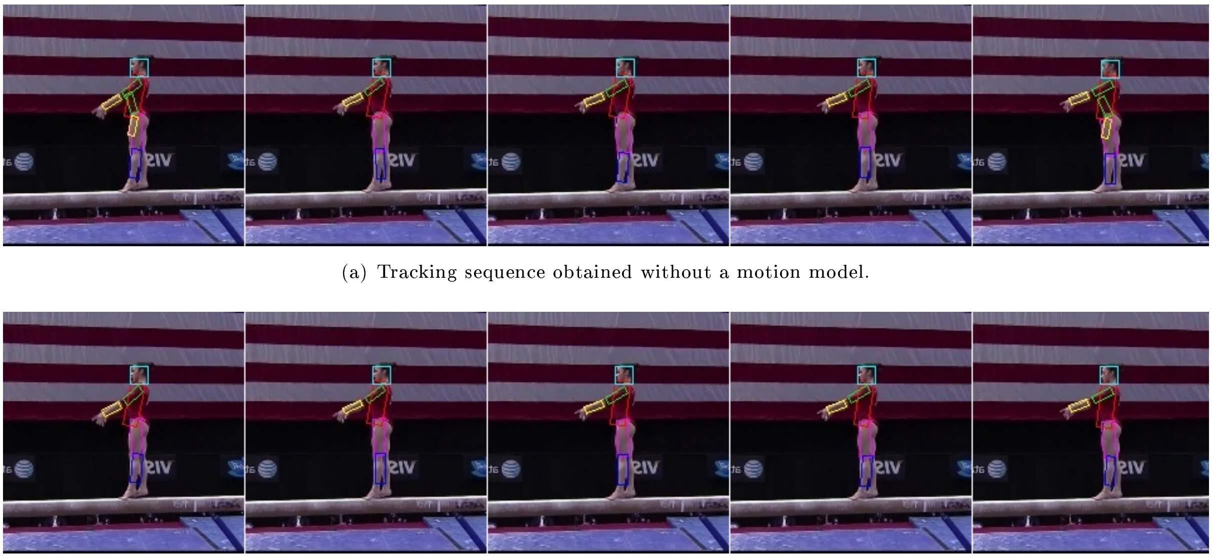
(b) Tracking sequence obtained with a motion model.

Figure 6.5

Figure 6.5 shows that both tracks are equally precise, due to the fact that the person is clearly visible such that the appearance model performs well. While the single pictorial structure algorithm finds both arms in the middle two frames, the motion model fails to do so, due to the shortcoming of Step 2 in the motion model implementation. In other words, the *right* arm in the sequence shown in Figure 6.5(b) is not detected because the motion model prior is calculated with respect to the *left* arm instead, which was previously detected.

Video 2 Second, we show our results on a more challenging setting: the person is smaller compared to the frame size (torso area to frame area ratio is 1:107) and the upper arms are coloured similarly as the torso.

The motion model helps to prevent wrong arm detection which occurs due to appearance model similarity with the rest of the body (upper arm with torso and lower arm with the leg), as can be seen in the first and last frames in Figure 6.6.



(a) Tracking sequence obtained without a motion model.

(b) Tracking sequence obtained with a motion model.

Figure 6.6

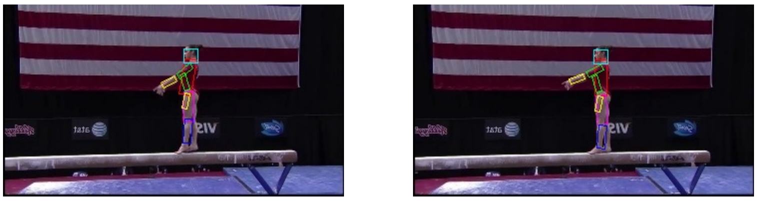
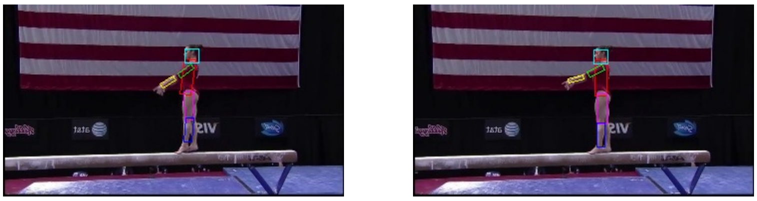
First and last frame in sequence 6.6(a)	Appearance thresholds
	$thresh_{tor} = exp(-2.5)$ $\boldsymbol{thresh_{ua} = exp(-1.5)}$ $\boldsymbol{thresh_{la} = exp(-1.0)}$ $thresh_{ul} = exp(-2.0)$ $thresh_{ll} = exp(-2.0)$ $thresh_{h} = exp(-1.2)$
	$thresh_{tor} = exp(-2.5)$ $\boldsymbol{thresh_{ua} = exp(-1.0)}$ $\boldsymbol{thresh_{la} = exp(-0.5)}$ $thresh_{ul} = exp(-2.0)$ $thresh_{ll} = exp(-2.0)$ $thresh_{h} = exp(-1.2)$

TABLE 6.2: Example of parameter adjustment to obtain correct detections with the single frame pictorial structure algorithm.

We also show the middle three frames with correct body pose detections, in Figure 6.6(a), to illustrate the following often encountered situation: even within a short sequence of consecutive frames (five frames in this case), a single set of input parameters does not always produce the correct body pose detection, which makes the robustness of the method with respect to input parameters again debatable.

We illustrate the method's sensitivity to input parameters in Table 6.2. The first row contains the first and last frame in Figure 6.6(a), together with the appearance thresholds used to obtain this sequence. The second row shows an example of how these parameters can be adjusted in order to obtain the correct detections. The motion model prior automatically removes the necessity for these manual adjustments (as illustrated in Figure 6.6(b) obtained with a single set of appearance thresholds).

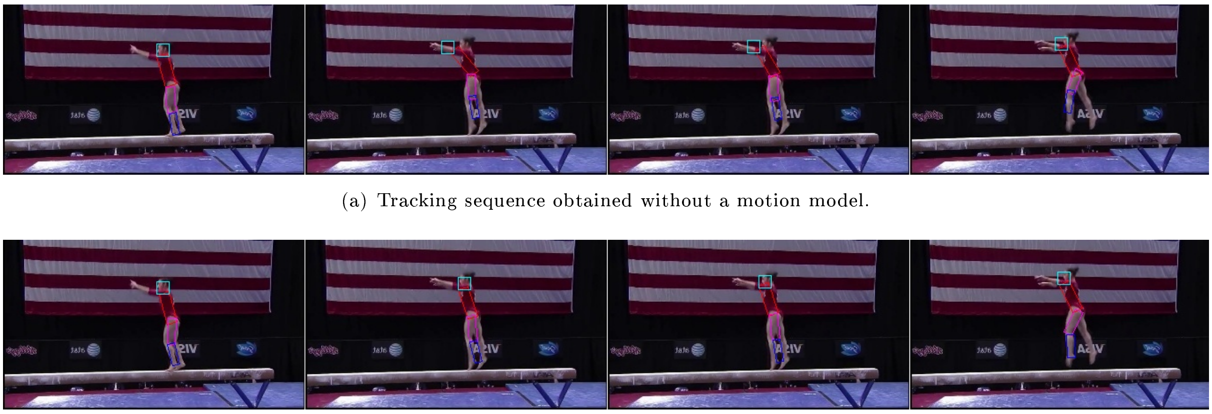
We also found in our experiments that the motion model might propagate wrong detections, if they strongly resemble the learnt appearance model. For both this situation and the one seen in Figure 6.6(a), the solution would be to enforce stronger appearance similarity thresholds (as exemplified in the second row of Table 6.2).

Figure 6.7 shows a situation similar to Figure 6.4: fast motion with a partly occluded head. As above, the motion model successfully manages to maintain the head track (in the middle frames).

Video 3 Figure 6.8 shows our results on a short sequence from a third video. In Figure 6.8(a) we notice that the left upper legs tends to drift from the correct position, also influencing the position of the left lower leg (due to kinematic constraints). This occurs due to differences in the colour appearance between the current frames and the frame from which the appearance model was learnt (the frame where the lateral walking pose was detected). The motion model prior compensates for this inaccuracy and maintains the upper leg track consistent from one frame to the other, as shown in Figure 6.8(b).

Conclusion

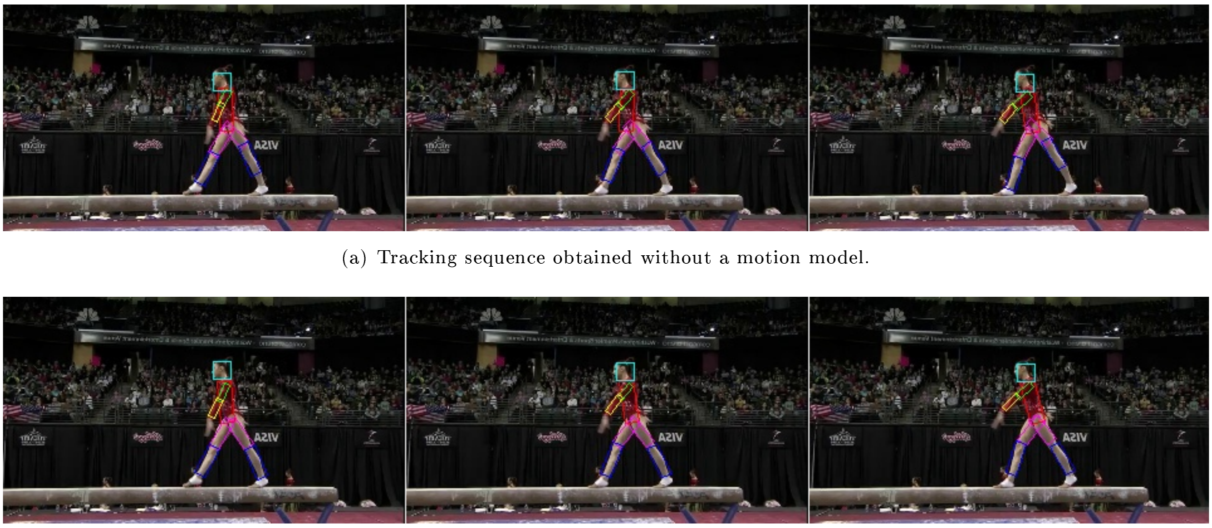
Our results confirmed the hypothesis in very specific situations, enumerated below. We consider that indeed the appearance model can perform well in clear scenarios, where the person moves at a steady pace and the body parts are clearly defined, both in shape and in colour.



(a) Tracking sequence obtained without a motion model.

(b) Tracking sequence obtained with a motion model.

Figure 6.7



(a) Tracking sequence obtained without a motion model.

(b) Tracking sequence obtained with a motion model.

Figure 6.8

Additionally, we showed that the motion model can improve tracking in more challenging situations, like the following:

- partially occluded head;
- fast, blurred motion;
- limbs detected on other parts of the body due to the similarity in colour appearance (for example, arms on torso);
- drifting body parts due to differences between learnt colour appearance model from the stylized pictorial structure frame and the colour appearance in the current frame.

The main weakness of the motion model, as currently implemented, is that it needs to be readjusted, depending on the movement. For example, for (moderate pace) equilibrium exercises performed at the balance beam, lower values for the bounded velocity parameter v_{max} resulted in more accurate detections, while for the swift movements (like the jumps shown in some of the sequences above), larger values for

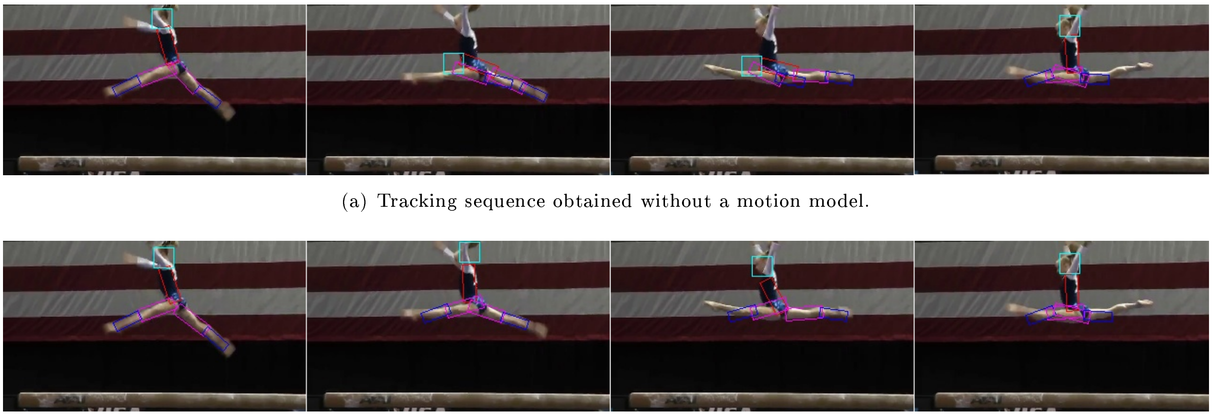
v_{max} were required. As such, the bounded velocity motion model requires precise tuning for the v_{max} parameter value.

Method 2: adding the previously sampled body part candidates as a parent node to the current body part node

We use the same video sequences as before, to demonstrate our results of Method 2 compared to the single frame pictorial structure algorithm. Second to this, we also make comments, where relevant, about the results obtained with Method 2 compared to Method 1, with no intention to extensively compare these two methods.

Results

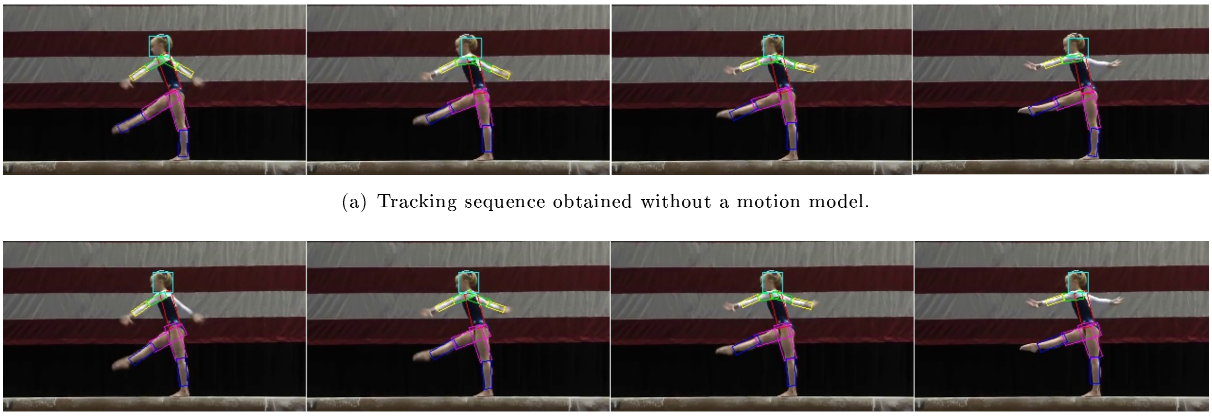
Video 1 Figure 6.9 shows the challenging scenario for Video 1 that we described before (see Video 1 in the results for Method 1). Just as Method 1, our current method manages to maintain the head track stable (and with it, the rest of the body), unlike the single frame pictorial structure algorithm.



(a) Tracking sequence obtained without a motion model.

(b) Tracking sequence obtained with a motion model.

Figure 6.9

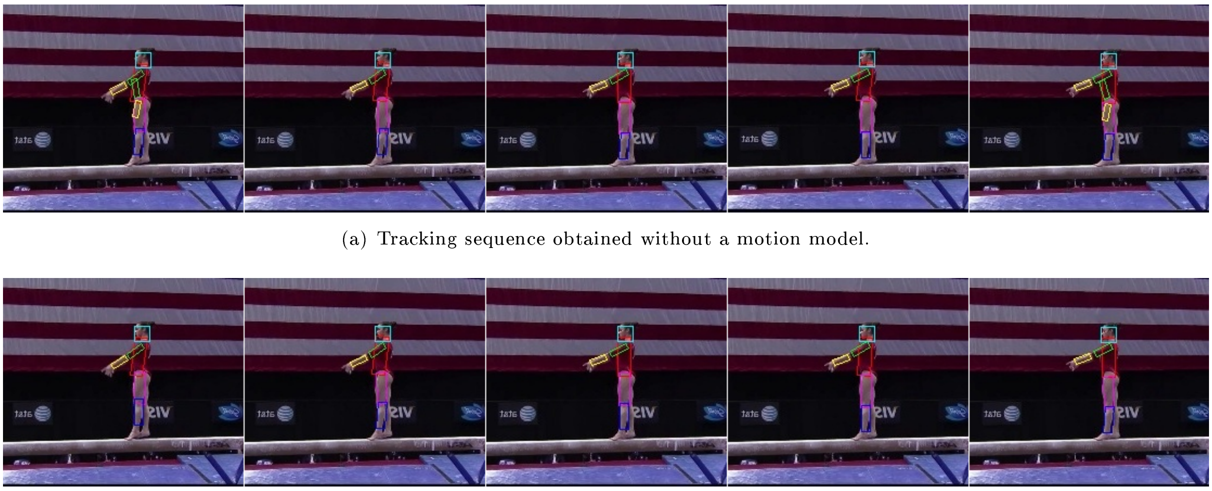


(a) Tracking sequence obtained without a motion model.

(b) Tracking sequence obtained with a motion model.

Figure 6.10

The quality of the two tracks in Figure 6.10 is similar, with the observation that the motion model prior maintains a more stable head track. Figure 6.10(b) shows that the algorithm is able to recover the right arm in the second frame. This is due to the fact that, even though the system sampled a low number of right arms which did not suffice to constitute a second mode (or a second limb detection) in the first frame, the right arm samples did weight in the arm motion model prior $\left(\sum_{x_{i_{t-1}}}\left[P(P_t^i|P_{t-1}^i)P(I_{t-1}|P_{t-1}^i, C^i)\right]\right)$ for the next frame with a high probability, which led to the right arm detection in the second frame. We consider this an advantage of Method 2 over Method 1 (comparing the track in Figure 6.10(b) with the one in Figure 6.5(b)).



(a) Tracking sequence obtained without a motion model.

(b) Tracking sequence obtained with a motion model.

Figure 6.11

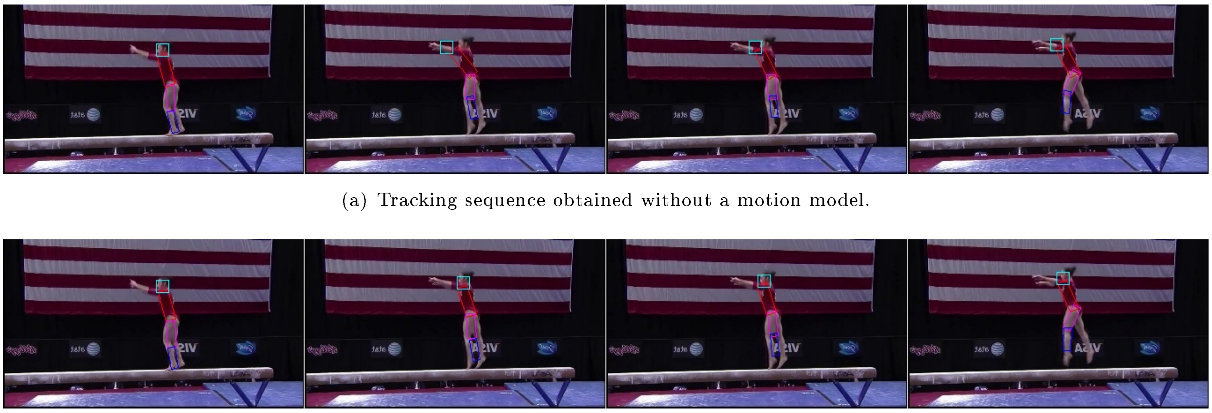
Video 2 For the second video (of a gymnast wearing a uniformly coloured suit, with elbow-length sleeves of the same colour as the torso), Figure 6.11(b) shows a similar result to Figure 6.6(b): the motion model maintains only the correct arm detection, while the single frame pictorial structure algorithm (Figure 6.11(a)) finds a wrong upper arm on the torso and a wrong lower arm on the leg in the first and last frames. This occurs due to the colour appearance similarity between the upper arm and the torso (clothing colour) and the lower arm and the leg (skin colour). The motion model prevents the wrong arm detection, as there is no corresponding arm in a similar position in the previous frame.

The second sequence from Video 2 (in Figure 6.12) also shows that, unlike the single frame pictorial structure, Method 2 is able to maintain a stable head track, in a sequence where the head is partially occluded. The incorrect positioning of the head also leads to a displacement of the torso (due to kinematic constraints), which makes the track from one frame to another even more inconsistent.

Video 3 Unlike in the previous two examples, the results obtained in the third video (Figure 6.13) show that Method 2 adds no value, compared to the single frame pictorial structure algorithm. The legs drift in an equal manner, towards the areas that comply with the learnt appearance model.

Conclusion

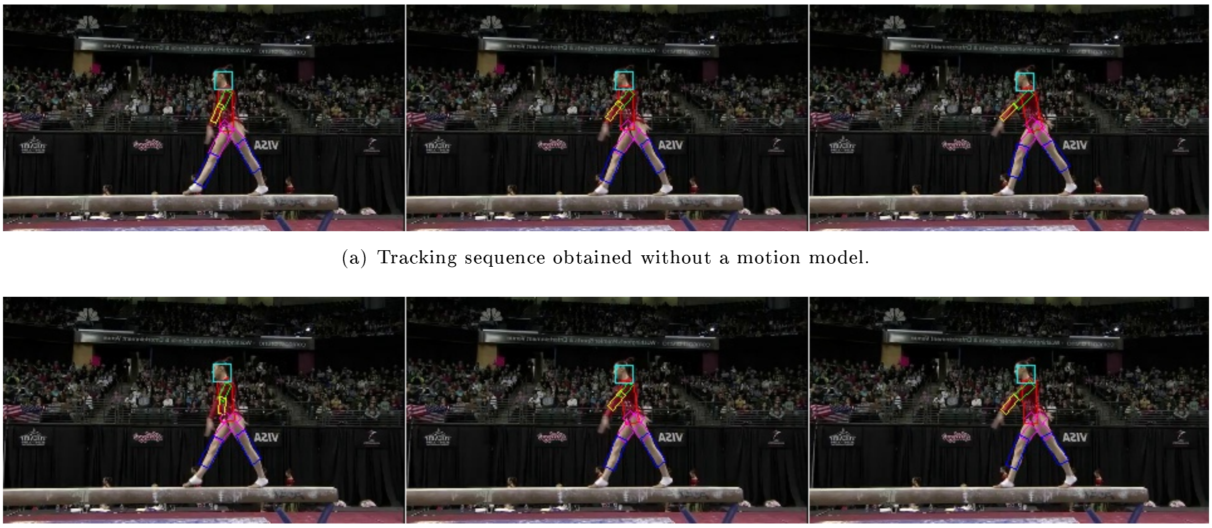
Most of our conclusions for Method 1 also stand for Method 2. The results confirmed our hypothesis for



(a) Tracking sequence obtained without a motion model.

(b) Tracking sequence obtained with a motion model.

Figure 6.12



(a) Tracking sequence obtained without a motion model.

(b) Tracking sequence obtained with a motion model.

Figure 6.13

Method 2 in very specific situations, but in less examples than for Method 1. We consider that Method 2 can improve the detection results, as compared to the single frame pictorial structure, in the following cases:

- partially occluded head;
- fast, blurred motion;
- limbs detected on other parts of the body due to the similarity in colour appearance (for example, arms on torso).

6.2.2 Motion models: bounded velocity and Gaussian noise

We analyse the tracking results obtained when implementing the motion model prior $P(P_t^i|P_{t-1}^i)$ according to either the bounded velocity motion model (see equation 3.2.5) or the Gaussian noise motion model (see equation 3.2.8). The differences between these two motion models are:

1. The bounded velocity motion model discards candidates that are positioned outside the imposed distance with respect to the detection from the previous frame and assigns the same probability for the candidates positioned within the distance limit. The Gaussian noise motion model expresses a preference for those candidates which are positioned closer to the detection from the previous frame. Closer candidates will be assigned a higher probability, while farther candidates will be assigned a lower probability.
2. The bounded velocity motion model considers the distance between the centre coordinates of the previous detection and the current candidate. Aside from the centre coordinates, the Gaussian noise motion model also includes the von Mises distribution that takes into account the angle variation.

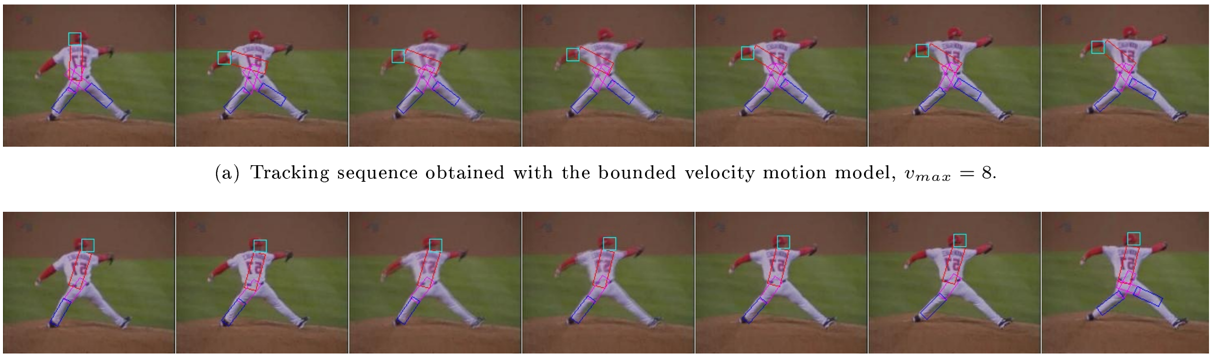
Note: In our implementation of the bounded velocity motion model we do not fully discard the candidates that lie outside the distance limit. Instead of assigning $P(P_t^i|P_{t-1}^i) = 0$, we assign a very small number, std :: numeric_limits<double>::epsilon(), for the motion model prior. This is motivated by the fact that the value for v_{max} needs to be determined experimentally because there is no a priori knowledge about the motion in the video. We determine this value by trial and error. This might lead to one of the following situations: either the v_{max} value is inadequate (too low) for the entire video sequence, which might lead to *all* candidates being discarded with respect to the previous detection, or the v_{max} value is appropriate for a part of the sequence but not for other parts, where the motion could suddenly accelerate. Assigning a very small motion model prior for candidates outside the v_{max} limit solves both these cases at the same time, as follows: for the first case, the two frame pictorial structure algorithm becomes equivalent to the single frame pictorial structure algorithm, while for the second case, the two frame algorithm applies for those video parts where the bounded velocity distinguishes plausible from implausible candidates, and becomes equivalent to the single frame pictorial structure algorithm for those video parts where it would normally discard all candidates.

Hypothesis We expect the Gaussian noise motion model to successfully remove those candidates which have a sudden shift in orientation with respect to the previous frame detection, but whose centre position is close to the previous detection, as imposed by v_{max}.

Experiment setup

For this comparison, we use the baseball pitch sequence that [Ramanan et al., 2007] have also reported detailed results on. As stated in our hypothesis, we try to address the case where candidates with implausible orientation, but plausible centre coordinates, negatively influence the final body part detection, resulting in sudden orientation shifts. We explain later in this section why this situation occurs in the baseball sequence and not in the gymnasts videos that we used to exemplify our results before.

We use Algorithm 1 of the two frame pictorial structure algorithm. We set the bounded velocity motion model at $v_{max} = 8$, because it results in good detections for the sequences of frames where there are no premises (discussed below) for sudden shifts in orientation. For the sequences where this occurs (relevant for our experiment and illustrated in the comparative figures), we find this parameter value suitable for comparison between the two motion models. For the Gaussian noise motion model, we varied σ_x and σ_y between 7 and 14 simultaneously and we chose between 1 and 4 as values for k. For these values we have obtained similar results relative to the problem that we are studying (sudden shifts in orientation). For each sequence that we show in the Results section, we mention the correspondent parameter values



(a) Tracking sequence obtained with the bounded velocity motion model, $v_{max} = 8$.

(b) Tracking sequence obtained with the Gaussian noise motion model, $(\sigma_x, \sigma_y, k) = (8, 8, 4)$.

Figure 6.14

in the figure description. All other parameters (body part sizes, appearance thresholds, second mode thresholds) are the same for both cases.

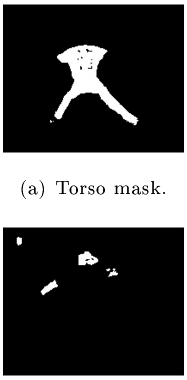
Results

The second frame in Figure 6.14(a) shows that the bounded velocity motion model allows for the torso detection to rotate to the left, although in the first frame the torso detection is approximately vertical.

Finding candidate torsos in that configuration is possible due to clothes deformation (the stretched arms elongate the blouse) and the appearance similarity of the upper arms and the torso, which makes the torso-like coloured image patch (the white mask in Figure 6.15(a)) fairly wide relative to the length, thus allowing for torso candidates to rotate in place.

The pictorial structure match in this case is also facilitated by the fact that the head and the lower sleeve have a similar appearance (both coloured in red). Figure 6.15(b)) shows the mask that covers the pixels which have the colour appearance of the head in the second frame from Figure 6.14.

Throughout the sequence shown in Figure 6.14(a), the bounded velocity motion model propagates the wrong detection of the head on the lower arm connected to the rotated torso detection. Figure 6.14(b) shows that the Gaussian noise motion model correctly maintains the orientation of the torso and successfully tracks the head and the torso throughout the same sequence.



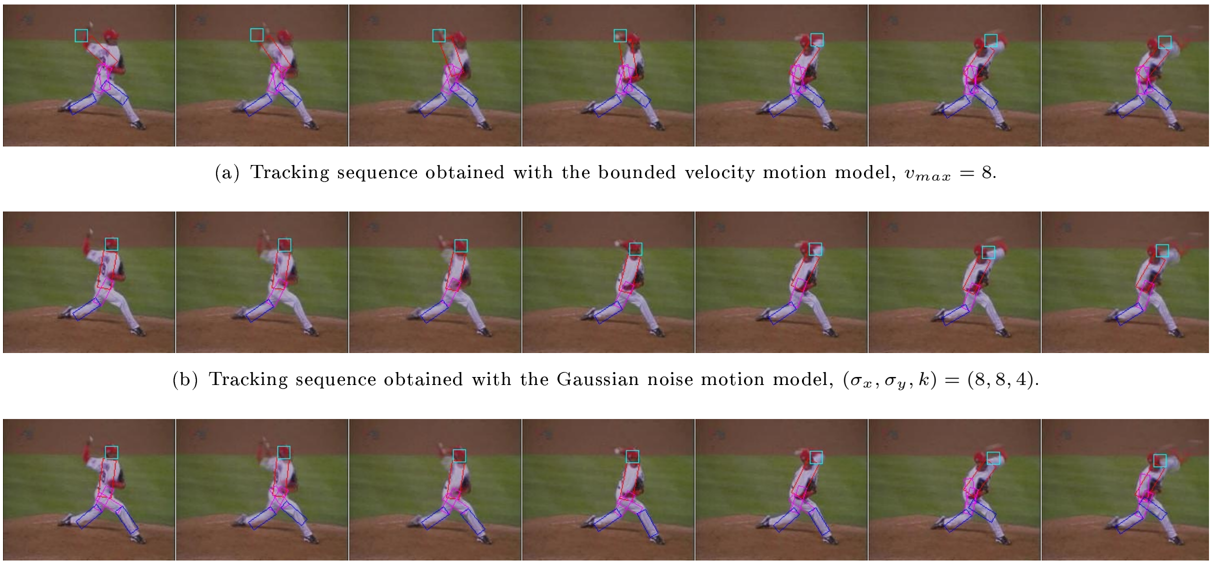
(a) Torso mask.

(b) Head mask.

Figure 6.15: Pixels that match head and torso appearance models (second frame in Figure 6.14).

Throughout the sequence shown in Figure 6.14(a), the bounded velocity motion model propagates the wrong detection of the head on the lower arm connected to the rotated torso detection. Figure 6.14(b) shows that the Gaussian noise motion model correctly maintains the orientation of the torso and successfully tracks the head and the torso throughout the same sequence.

Figure 6.16 is a continuation of the sequence shown in Figure 6.14 (ten frames later). It shows that both motion models perform the same in terms of consistently propagating the detections from the previous frames. However, due to the faulty detection in the initial frame in Figure 6.16(a), the system tracks the lower arm instead of the head. It is able to recover the torso configuration in the fifth frame, but



(a) Tracking sequence obtained with the bounded velocity motion model, $v_{max} = 8$.

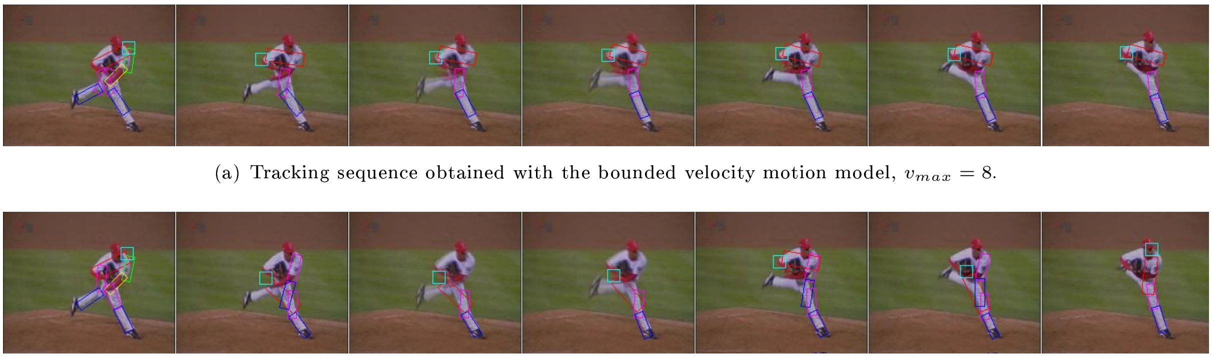
(b) Tracking sequence obtained with the Gaussian noise motion model, $(\sigma_x, \sigma_y, k) = (8, 8, 4)$.

(c) Tracking sequence obtained with the Gaussian noise motion model, $(\sigma_x, \sigma_y, k) = (8, 8, 1)$.

FIGURE 6.16

continues to track the arm instead of the head. The head detection is positioned between the lower arm and the actual head in the last three frames.

Figures 6.16(b) and 6.16(c) show that, with the Gaussian noise motion model and a correct initial frame, the detection is more stable. The head is correctly tracked and the orientation of the torso changes smoothly according to the actual motion. Figure 6.16(c) shows an overall better detection for the legs, due to the more permissive value for parameter k. Both sequences were initialised with the same detection, where both the legs were present, but only the sequence in Figure 6.16(c) perpetuates both leg tracks. This indicates that while one set of parameter values (like the one for Figure 6.16(b)) gives good results for some body parts (the head and the torso, in this case), a different set of parameter values could improve results for other body parts (in this case, the legs).



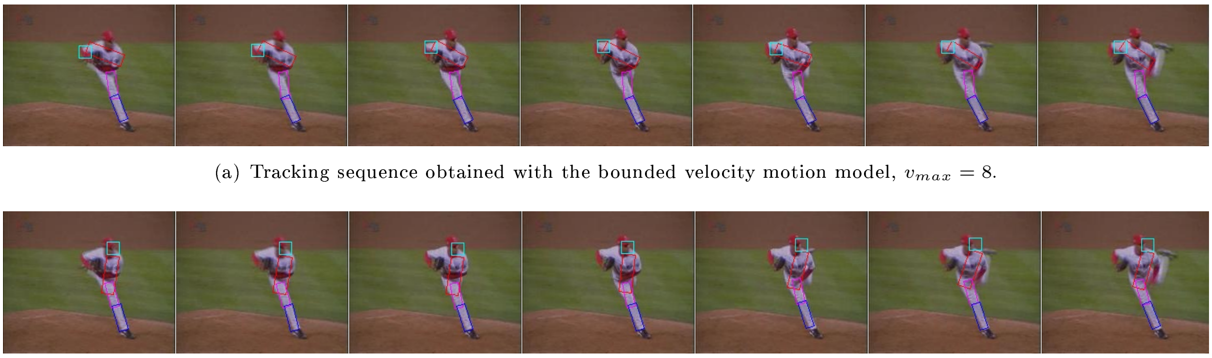
(a) Tracking sequence obtained with the bounded velocity motion model, $v_{max} = 8$.

(b) Tracking sequence obtained with the Gaussian noise motion model, $(\sigma_x, \sigma_y, k) = (8, 8, 1)$.

FIGURE 6.17

Figure 6.17 shows a challenging sequence that destabilises both tracks. In this case, the white torso is occluded by a black baseball glove. The detection in the sequence obtained with the Gaussian noise motion model is confused for the first six frames, but it manages to recover in the last frame (in Figure

6.17(b)). The bounded velocity motion model is more stable (although mostly incorrect) - it maintains the (partly incorrect) torso track and the (correct) leg track (in Figure 6.17(a)).



(a) Tracking sequence obtained with the bounded velocity motion model, $v_{max} = 8$.

(b) Tracking sequence obtained with the Gaussian noise motion model, $(\sigma_x, \sigma_y, k) = (8, 8, 1)$.

Figure 6.18

Figure 6.18 is the immediate continuation of the sequence in Figure 6.17. It shows that, due to the recovery from the last frame in Figure 6.17(b), the tracking algorithm using the Gaussian noise motion model successfully propagates the torso and the head (in Figure 6.18(b)).

Figures 6.17 and 6.18 show that the Gaussian noise motion model can recover more reliably from a challenging situation where the torso is occluded by the glove. We focused our observations on the torso and the head detections because we believe that inaccuracies in these body parts' detections lead to the most disturbing effects on a viewer's perception of the detected body pose.

Conclusion

The results confirm our hypothesis for the given set of parameters and the selected sequences. We showed an example video of situations where shifts in orientation of the candidates can occur due to the deformation of the clothes and the similarity in appearance of the connected body parts (upper arms and the torso). For this case, the Gaussian noise motion model succeeds in reliably maintaining the correct detection.

We identified the clothes deformation as the issue that can be addressed using the Gaussian noise motion model. We could not find dramatic shifts in torso candidates orientation in the gymnasts videos, where the torso's shape is constant because the suit is tightfitting. For this reason, we do not consider that the Gaussian noise motion model would add value to videos where people wear tightfitting clothes and where the provided body part sizes apply throughout the sequence. For this case we recommend the bounded velocity motion model.

7 | Conclusions and future work

We tackled the problem of estimating the pose of an individual in video, by tracking the major body parts in every frame. We used pictorial structures to represent the human body as a puppet of rectangles. We coped with the frequent challenges which occured in our test input videos of gymnasts performing at the balance beam, such as fast motion, resembling body parts and illumination changes, by extending [Ramanan et al., 2007]'s single frame pictorial structure graphical model. We added nodes containing temporal information from the previous frame, through connections in the form of motion models.

We implemented two algorithms which calculated the motion model prior in two ways: by considering that the previous node is represented by the previous body part detection (mode) and by considering that the previous node is represented by all previously sampled body part candidates. Both methods showed advantages compared to the single frame pictorial structure algorithm, in terms of accurate detection of the body parts, especially that of the arms and the head.

We also found that incorporating the body part orientation, aside from the centre position, in the motion model can help cope with situations where body part candidates are incorrectly localised due to the mismatch between the part templates and the image patch belonging to the actual part being deformed by the clothes stretch.

We analysed the original method in terms of robustness and performance with respect to the type of input videos and we found that the original method is highly dependent on the set of input parameters, which do not translate across different videos. We also showed situations where one set of parameters can lead to different results within the same video and found that the motion model can eliminate the necessity to manually adjust these parameters when processing a single video.

Motion model prior We proposed two methods to include the previous pose detection in the graphical model of the pictorial structure. These methods are simplifications and particularisations of the complete temporal model that we partly showed in Figure 3.2. In the first method we considered that the current pose detection depends solely on the pose detection from the previous frame. In the second method we considered that the current pose depends on all previously sampled (single arm, single leg) body configurations. We found that both methods can improve the single frame pictorial structure algorithm in specific situations such as fast motion or similarly coloured body parts. We put forward the first method as being an elegant solution to integrate the temporal dimension into the tracking system, as it is both efficient in terms of computation time and it also provides reliable results.

We also compared two variants to calculate the motion model prior: the bounded velocity motion model, that [Ramanan et al., 2007] suggested as the simplest solution, and the Gaussian noise motion model which has been used in other research papers. We only identified one situation (refer to Section 6.2.2) where the bounded velocity motion model did not perform as well as the Gaussian noise motion model. Thus we conclude that the bounded velocity motion model remains a reliable solution.

Input parameters The original method requires a number of input parameters that must be hand-picked by the user, as follows:

- 15 parameters for the model building module: 6 body part widths and lengths and 3 chamfer score thresholds (for the torso, legs and arms),
- 18 parameters for the detection module: 6 body part widths and lengths, 6 appearance similarity thresholds (for each body part) and 2 second mode thresholds (one for the arms, one for the legs),

to which we added:

- 1 parameter for the bounded velocity motion model (same for all body parts),
- 3 parameters for the Gaussian noise motion model (covariances σ_x and σ_y and parameter k for the von Mises distribution, same for all body parts).

We observed that the performance of the system can change dramatically with the adjustment of these parameters. The model building module finds a fixed scale stylised pose, thus it requires exact sizes for the body parts. We obtained these by visually inspecting the video and measuring the body parts in one of the frames containing a lateral walking pose.

The detection module performance highly depends on the appearance similarity thresholds. We showed that one set of such thresholds does not translate across different videos. We also found in our experiments that one set of thresholds per video does not give optimum results either. The thresholds need fine tuning in order to obtain the best results at frame level (we intervened manually when we noticed that the pose detection became inaccurate). We made an attempt to eliminate this shortcoming by making the system more precise by implementing a **motion model**. We showed that, in specific situations, the motion model prior can compensate for inaccuracies in the image likelihood (matching an image patch to the body part appearance model).

The entire system, however, remains highly dependent on the input parameters. In order not to increase the complexity of the parameter adjusting process even more, we opted to provide the same motion model parameter values for all body parts. We obtained satisfactory results, but there is still room for improvement that we suggest in Future work.

Appearance and deformable models We explained the complementarity between the appearance model and the kinematic constraints. When using generic appearance models (via general features like edges), it is crucial that the deformable model is precisely defined (such as for the stylised pose). When detecting general poses, which require a relaxed set of kinematic constraints, it is crucial that the appearance model is precisely defined (using features such as shape and colour, in the detection module). We consider that the image likelihood calculated with the colour appearance model gives the strongest indication about the person's position and configuration and that further refinements of the kinematic model are not necessary.

Stylised pose We found that rectangular templates are a simple and powerful tool to detect body parts and we successfully detected lateral walking poses in our videos. However, from our experiments, we found that the current implementation imposes very strict limitations on the input video. In other words, a pose that a human would perceive as a lateral walking pose might not always be detected by the computer, as the implementation requires the following: the torso must be perfectly vertical, the head must be exactly above the torso, the elbow must display an imposed angle and the overlap between legs must be at a minimum. Also, one main assumption of learning appearance models from a lateral pose is that the person's clothes are symmetrical in colour. For this reason we were not able to use videos of gymnasts whose sport costumes were unevenly coloured.

Prior knowledge and system limitations We found that a great deal of research was dedicated to tracking pedestrians or upright people. Other papers focused on laboratory controlled settings or sacrificed accuracy for a large applicability in terms of video input. We also conclude that to achieve the (body part level) detailed results and wide applicability (with respect to the video input) from [Ramanan et al., 2007]'s method, the system requires extensive knowledge about the video. This knowledge can be gained through observation (person scale as defined by body part sizes), or through experimentation (chamfer score thresholds, appearance similarity thresholds, second mode threshold). Our solution of implementing a motion model can be seen as a compromise, because we attempt to minimize the manual intervention on the previously listed parameters, but we also introduce a supplementary set of parameters for the motion model.

Future work We list here a series of possible improvements and open issues for future work.

1. Illumination invariance
   We found that the results notably suffer from illumination changes. Therefore, we believe that finding other illumination invariant features to model the appearance of the person would significantly improve the detection results.
2. Chamfer matching
   We found that upper and lower arm detection is difficult in the model building module because of their reduced size. At the moment, chamfer matching is used to find body part candidates, using edges as features. Scientists proposed improvements to the classic chamfer matching algorithm, such as [Borgefors, 1988], which could be a starting point to research further on this detection procedure.
3. Motion model parameters
   With the scope to keep the amount of parameters low, we resumed to only one set of motion model parameters for all body parts. Without increasing the number of parameters, it would be interesting to determine if there exist some general dependencies between motions of different body parts (regardless of the activity performed by the person in the video), such that only one set of parameters would need to be provided by the user (for the torso, for example) and the rest would be expressed as functions of this set.

4. Implementation limitations

   The following two features have been solved in previous research and could be included in our implementation, to expand the system's functionality: tracking more than one person at the same time and tracking people at various scales.

5. Ground truth comparison

   All our experiments could offer more conclusive results if they were compared with ground truth. This would require a measure of similarity between body part detections (rectangles) and a dataset where the same body configuration (torso, head, upper legs, lower legs, upper arms, lower arms) would be labelled per body part.

6. Temporal smoothing

   [Ramanan et al., 2007] implemented temporal smoothing by inputting the body configuration samples from the previous frame, the current frame and the next frame into the modes finding procedure to obtain the current pose. It would be interesting to compare our results using the two frame pictorial structure algorithms to the results obtained with the aid of temporal smoothing.

7. Torso template

   Aside from the set of input parameters that we discussed, the model building module requires a torso template which consists of the torso and head edges labelled with the orientation. We obtain these by looking at a frame containing a lateral walking pose and by running the edge detection on it, then by selecting that area that corresponds to the torso and the head. We speculate that it is possible to find the torso using a rectangular template (as for the other body parts). Using a rectangular template instead of the toilsome torso template would be a step further to make the system fully automatic.

Bibliography

D. Ramanan, D. A. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):65–81, 2007.

M. A. Brubaker, L. Sigal, and D. J. Fleet. Video-based people tracking. In *Handbook of Ambient Intelligence and Smart Environments*, pages 57–87. Springer, 2010.

L. Sigal. Human pose estimation. URL `http://cs.brown.edu/~ls/Publications/SigalEncyclopediaCVdraft.pdf`.

H. Li, J. Tang, S. Wu, Y. Zhang, and S. Lin. Automatic detection and analysis of player action in moving background sports video sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(3):351–364, March 2010. ISSN 1051-8215.

H. Li, S. Wu, S. Ba, S. Lin, and Y. Zhang. Automatic Detection and Recognition of Athlete Actions in Diving Video. In *Advances in Multimedia Modeling*, volume 4352, chapter 8, pages 73–82. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

J. Han, D. Farin, P.H.N. de With, and W. Lao. Automatic tracking method for sports video analysis, 2005.

A. Ekin, A. M. Tekalp, and R. Mehrotra. Automatic soccer video analysis and summarization. *Image Processing, IEEE Transactions on*, 12(7):796–807, July 2003.

T. Zhang, B. Ghanem, and N. Ahuja. Robust multi-object tracking via cross-domain contextual information for sports video analysis. In *ICASSP*, pages 985–988. IEEE, 2012.

B. Ghanem, T. Zhang, and N. Ahuja. Robust multi-object tracking via cross-domain contextual information for sports video analysis. In *ICASSP*. IEEE, 2012.

D. Ramanan, D. A. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego*, volume 1, pages 271–278, 2005. URL `http://www.ics.uci.edu/~dramanan/papers/pose/index.html`.

G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library.* O'Reilly Media, 2008.

ALGLIB, 2013. URL `http://www.alglib.net/`.

R. Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108(1-2):4–18, October 2007.

D. A. Forsyth, O. Arikan, and D. Ramanan. Computational studies of human motion: Part 1, tracking and motion synthesis. In *Foundations and Trends in Computer Graphics and Vision*, page 2006. Now Publishers Inc, 2006.

C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, August 2000.

M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1820–1833, September 2011.

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

J. C. Niebles, B. Han, A. Ferencz, and L. Fei-Fei. Extracting moving people from internet videos. In *Proceedings of the 10th European Conference on Computer Vision: Part IV*, ECCV ’08, pages 527–540, Berlin, Heidelberg, 2008. Springer-Verlag.

J. C. Niebles, B. Han, and L. Fei-Fei. Efficient extraction of human motion volumes by tracking. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, June 2010.

F. Huo and E. A. Hendriks. Multiple people tracking and pose estimation with occlusion estimation. *Computer Vision and Image Understanding*, 116(5):634 – 647, 2012.

D. Ramanan. Learning to parse images of articulated bodies. In *Advances in Neural Information Processing Systems 19*, pages 1129–1136. MIT Press, Cambridge, MA, 2007.

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

D. Hogg. Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5 – 20, 1983.

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

E. J. Gumbel, J. Arthur Greenwood, and David Durand. The Circular Normal Distribution: Theory and Tables. *Journal of the American Statistical Association*, 48(261), 1953.

J. S. Yedidia, W. T. Freeman, and Y. Weiss. Exploring artificial intelligence in the new millennium. chapter Understanding belief propagation and its generalizations, pages 239–269. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

E. W. Weisstein. Probability density function, a. URL `http://mathworld.wolfram.com/ProbabilityDensityFunction.html`.

E. W. Weisstein. Distribution function, b. URL `http://mathworld.wolfram.com/DistributionFunction.html`.

D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach.* Prentice Hall Professional Technical Reference, 2002. ISBN 0130851981.

H. Sanchez. Image Kernels and Convolution (Linear Filtering), September 2011. URL `http://demonstrations.wolfram.com/ImageKernelsAndConvolutionLinearFiltering/`.

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 *Proceedings of the 5th international joint conference on Artificial intelligence - Volume 2*, pages 659–663, San Francisco, CA, USA, 1977. Morgan Kaufmann Publishers Inc.

A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla. Shape Context and Chamfer Matching in Cluttered Scenes. In *Proceedings of the 2003 IEEE computer society conference on Computer vision and pattern recognition*, pages 127–133, Washington, DC, USA, 2003. IEEE Computer Society.

D. R. Martin. Benchmark and Boundary Detection Code of the Berkeley Segmentation Dataset and Benchmark. Computer Vision Group, University of California, Berkeley, 2003. URL `http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/code/`.

L. Kitchen and A. Rosenfeld. Non-maximum suppression of gradient magnitudes makes them easier to threshold. *Pattern Recognition Letters*, 1(2):93 – 94, 1982.

A. Jepson. Introduction to Image Understanding. Department of Computer Science, University of Toronto, 2011. URL `http://www.cs.toronto.edu/~jepson/csc420/notes/edgeDetection.pdf`.

A. Feelders and R. Veltkamp. Pattern Recognition. Information and Computing Sciences, Faculty of Science, Utrecht University, 2012. URL `http://www.cs.uu.nl/docs/vakken/mpr/index.php`.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction: with 200 full-color illustrations.* New York: Springer-Verlag, 2001.

E. W. Weisstein. Newton's Method – from Wolfram MathWorld, c. URL `http://mathworld.wolfram.com/NewtonsMethod.html`.

D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):849–865, November 1988.