Tracking People and Recognizing their Activities

Hauptseminar
Videobasierte Erkennung und Analyse menschlicher Aktionen

Benjamin Kormann

Betreuer: Dipl.-Inform. Jan Bandouch
Abgabetermin: 06. Juni 2006


## Contents

1 Introduction 1

2 Background 3
   2.1 Hidden Markov Models 4
   2.2 Why Markovian Models fail 5
   2.3 Constant Appearance Models 6

3 Model-based Tracking 7
   3.1 Pictorial Structure 7
   3.2 Model Building 9
   3.3 Model Detection 14
   3.4 Evaluation 16

4 Summary 19

Bibliography 20
Abstract

This paper analyzes and presents the dissertation of Deva Kannan Ramanan - *Tracking People and Recognizing their Activities*. It is being demonstrated how the automatic people tracker operates focusing on each stage of the system. The localization and tracking of people is hard for many reasons. Persons tend to take in atypical poses in images, which complicates the identification process. In many approaches people are treated as blobs, which can be enough for tracking, but not for understanding their activities. The dissertation proposes a way people can be tracked by detecting a previously assembled appearance model derived from all human body parts - needed for further activity-interpretation - in each frame. Finally all steps of the system are evaluated by experimental results and further improvements are mentioned.
Chapter 1

Introduction

*To interact with computers we still have to learn their languages; they are not smart enough to learn ours.*
- Stephen Pinker

Computer systems have been getting more powerful over the last decades. Applications seem to become better and bigger and so do their requirements, because more memory, higher CPU frequency and less data access time is needed to make them work adequately.

Contemporaneously the human brain is also emerging more and more as a natural matter of fact. It does not increase human’s performance like computer’s do, but the brain of a human being is still much more powerful than any existing cluster of high performance machines.

Simple object recognition is handled by our brains within small parts of a second, whereas a computer consumes lots of processing power. The first difficulty, which has been proven as one is the issue of transforming human object recognition into a language a computer can understand. In other words, it is fairly hard to derive a computational equivalent to a simple process, which is achieved by a human brain with a winking time frame.

There is no all-mighty algorithm which solves all of these problems. The problems a computer is faced with are obvious. Objects in general in a special environment have a changing background, which makes it hard to distinguish between actual object boundaries and background informations given by the image. Illumination though makes it sometimes hard for humans to recognize objects, because depending on the incoming angle of the light, its resulting shadow and daytime changing passive light, objects change their appearance. A major change in perception occurs when looking onto a certain object from different perspectives. A well-shaped ball becomes an egg, a cube becomes a cuboid and so forth.
All these issues are summed up when it comes to the presence detection of people in an image without knowing if there are any. Depending on the application of the vision system a different level of people understanding is needed, i.e. a University main hall observing vision system dedicated to analyze a lecture room’s utilization is not interested in what a person holds in his hand, whereas a system that is run for tennis shot analysis must determine the person’s limbs position, his pose while waiting and finally returning the ball to the opponent. The recognition of activities (e.g. waving, window cleaning) is hardly understood. Even for humans it is a fairly tough task to figure out a person’s activity by simply looking at an image. For a correct automatic activity interpreter a classifier is needed that takes a person’s limbs position as input and returns the corresponding activity as output.

The dissertation of D. Ramanan addresses fundamental methods in order to detect and track people in video sequences over time and finally the system tries to automatically correlate the poses with an activity, which will not be part of this paper.
Chapter 2

Background

*Knowledge is power.*
- *Sir Francis Bacon*

Tracking objects, persons respectively and interpreting their activity are challenging tasks in computer vision. A stable system must meet a long list of requirements in order to operate correctly and efficiently.

A simple technique for segmentation in video sequences is the use of motion. Motion is used by humans and animals to extrude interesting objects from unwanted background information. One of the simplest approaches is to compare two taken images $f(x,y,t_i)$ and $f(x,y,t_j)$ [RCG02]. One option is to compute the difference between a reference image containing only stationary information and each image, where moving objects are expected. The difference image of images taken at $t_i$ and $t_j$ is

$$d_{ij}(x,y) = \begin{cases} 
1 & |f(x,y,t_i) - f(x,y,t_j)| > T \\
0 & \text{otherwise}
\end{cases}$$

(2.1)

where $T$ is a predefined threshold. This technique reaches its limits real soon, because it outputs noisy images and can only segment an image into two parts (foreground, background). Additionally the threshold is a sensitive parameter. In some applications an adaptive threshold technique can be applied to compensate illumination changes across an image, but this paper does not go into greater details of this strategy.

This section describes the Hidden Markov Models and explains why the use of them will fail in this case. Furthermore it will be told what to use data association and a constant appearance model for.
2.1 Hidden Markov Models

Markov Models provide a well-founded methodology for reasoning about temporal events. Hidden Markov Models (HMM) consist of unknown internal parameters that need to be guessed by observable facts. The hidden states are poses $X_t$ and the observed parameters can be retrieved by images $I_{mt}$. Standard Markov assumptions applied to poses and images lead to

$$Pr(X_{1:T}|I_{m1:T}) = \prod_{t} Pr(X_t|X_{t-1})Pr(I_{mt}|X_t)$$  \hspace{1cm} (2.2)$$

Now tracking refers to the inference over the probability model, where one usually looks for the maximum a posteriori (MAP) sequence of poses, which can be computed with

$$\hat{X}_{1:T} = \arg\max Pr(X_{1:T}|I_{m1:T}) = \arg\max \prod_{t} Pr(X_t|X_{t-1})Pr(I_{mt}|X_t)$$  \hspace{1cm} (2.3)$$

2.1.1 Inference Algorithm

The domain of $X_t$ can not be discretized because of its complexity and therefore many approaches use inference by variants of kalman or particle filtering [BI98], [JDR00]. Given the current frame $t$ with the identified pose, it is feasible to predict the pose in frame $t+1$ where the actual pose can be determined and used to adapt the model for the next prediction.

2.1.2 Motion Model

Human motion is hard to predict. Researchers in the graphics community have been trying to get a generic model of human movement. D. Ramanan uses the simple constant velocity model for his automatic people tracker.

2.1.3 Image Likelihood

It must be general enough being able to detect limbs in all different kinds of poses, but on the flipside it should not be specific, so that it is not disturbed by any clutter. If an arm is occluded by an arm-like pole, the system tends to drift and gradually loses track of the human body part. Tracking by detecting is the better approach, because it searches for the highest likelihood in each frame and so the system gets reinitialized after every frame, which prevents drifts from occurring.
2.1.4 Initialization

Many systems need a manual initialization, which means that a special domain knowledge is needed in order to adapt the system to the requirements accordingly. Different pose variations can be covered by bottom-up modelling, i.e. considering each limb as a segment, but a likelihood is needed for each human body part. It is a general matter of fact, that background can contain elements similar to e.g. arms or legs.

2.1.5 Data Association

Why is data association worth being taken into account? A likelihood model could directly derive a segmented image and filter out persons. $Pr(Im_t | X_t)$ knows a person’s properties like Lola’s red hair, hence if there are no red pixels in an image, Lola can not be present. Unfortunately this information is not present a priori, but must be generated dynamically while processing the video. Obviously adding an appearance model $A_t$ to a pose $X_t$ does not meet the requirements of an automatic initializing system, because the appearance is unknown.

![Figure 2.1: Inference applied with a standard torso tracker without any constant appearance constraint. Observing the tracker over $T$ frames shows that the system drifts away, such as $A_1$ and $A_T$ have absolutely no similarity anymore.](image)

2.2 Why Markovian Models fail

Markovian models can be used for a template-matching torso tracker, but they fail under particular circumstances and therefore they can not be taken into account for a reliable automatic people tracker. The hidden state is defined as

$$X^i = \begin{bmatrix} P^i \\ A^i \end{bmatrix}$$ (2.4)
where \( P^i \) denotes the torso’s position and \( A^i \) represents a cloud of pixels for the appearance. The observations are the images \( I_{mt} \) from the video. The likelihood model returns high rates for pixel positions \( P_1 \), where the image patch \( I_m(P_1) \) looks like the appearance template \( A_i \). Figure 2.1 shows the performed inference applied to a sample of the Winter Olympics in 1998. The track of the torso drifts away step by step starting from frame 1 to \( T \), all corresponding appearance models are shown. The reason it drifts away is, that there is no constraint saying that the limb’s appearance \( A_T \) must roughly be the same like \( A_1 \) and so the small errors done with each step sum up. The torso image patch at each frame \( T \) is modeled as a sample from a Gaussian centered at an underlying torso patch \( C \). Two samples from the same Gaussian can be far away from each other, so that the variance in a constant model is \( \text{var}_c(A_t|A_1) = 2 \), whereas in a Markov model the variance increases linearly \( \text{var}_m(A_t|A_1) = T - 1 \). Because of the linear behavior of HMM, they will lose track of any object in longer sequences.

### 2.3 Constant Appearance Models

The constant appearance model augments the image likelihood

\[
Pr(I_{mt}|P_t, C) \propto e^{-||I_{mt}(P_t) - C||^2}
\]

and takes care, that each found representative of a body part looks more or less like the canonical appearance \( C \) in general. Figure 2.2-a) shows, a searched image patch, which looks like the canonical appearance \( C \) of the torso. Template matching algorithms use a predefined template with a constant appearance model.

![Figure 2.2: The pictorial structure shows a tracked torso over time like in Figure 2.1. Here, all found image patches must have a common appearance, which the local image patch likelihood takes into account.](image-url)
Chapter 3

Model-based Tracking

I want all my senses engaged.
Let me absorb the world’s variety and uniqueness.
- Maya Angelou

A system that automatically tracks people and understands their activities, assuming there are any, from a video sequence as input needs a certain level of understanding, which depends on the application of the system. Before people can be tracked in a video and their activities understood, the correct number and position of all visible and necessary people must be determined. Unlike earlier people tracking approaches D. Ramanan’s automatic people tracker does not treat humans as blobs, which fulfills the requirements of quite a few applications though. D. Ramanan considers a higher detailed person model needed, because gestures represent an important part in communication. Therefore the person model is not just a high density scatter cloud, but a person is being divided up into his limbs (torso, arms, legs, ...).

Figure 3.1 visualizes the system’s mode of operation. The generic person model is used to build an instance-specific model wrapping a person’s appearance. The tracking process is done by detecting the model in each frame.

The next section describes the pictorial structure used to represent the model of a human body.

3.1 Pictorial Structure

The pictorial structure [FH05] in Figure 3.2 is the basic representation of the used person model. The standard model for all $T$ frames is
The notation convention is

- superscripts denote body parts $i \in \{\text{torso}, \text{arm}, \ldots\}$
- subscripts denote frames $t \in \{1, \ldots, T\}$
- $\pi(i)$ denotes the parent-function of limb $i$
- the vector $P^i_t$ holds the position $(x, y)$ and the orientation $\theta$ of part $i$ at time $t$

The Equation 3.1 is a product over all limbs $i$ and frames $t$, where $Pr(P^i_t|P^i_{t-1})$ stands for the motion model, which tries to estimate the position of limb $i$ in frame $t$ knowing its position in frame $t-1$. $Pr(P^i_t|P^{\pi(i)}_{t-1})$ represents the standard geometric likelihood, which models the probability that limb $i$ in frame $t$ belongs to the parent limb $\pi(i)$, e.g. $Pr(P^\text{arm}_t|P^\text{torso}_{t-1})$. $Pr(Im_t(P^i_t)|P^i_t; C^i)$ means the local image patch likelihood, which takes care that the local image patch $Im_t(P^i_t)$ looks like the model of limb $i$. Tracking now refers to inference over the probability model, in other words finding a valid sequence of limbs in a video sequence that follows the model.

Retrieving a customizable and generic motion model [LKP02], [JLP02], [AF02] is an ongoing area of research. A rule that always applies is the easier, the better and therefore the bounded velocity comes into operation for the motion model on the right hand side of Equation 3.1

$$\Psi(P^i_t, P^i_{t-1}) = Pr(P^i_t|P^i_{t-1})$$

$$\propto T(||P^i_t - P^i_{t-1}|| < v_{max})$$
CHAPTER 3. MODEL-BASED TRACKING

Figure 3.2: This pictorial structure models the organization of all limbs \( Pr(P^i_t|P^j_t) \) hierarchically and the local appearance \( Pr(Im_t(P^i_t)|P^i_t, C^i) \) denoted by the leaf-nodes. b) shows the entire torso-arm structure tracked over a sequence of \( T \) frames.

The human body movement modelling uses potentials of

\[
\Psi(P^i_t, P^j_t) = Pr(P^i_t|P^j_t) \quad \propto \quad I(D(P^i_t, P^j_t) < d_{\text{max}})
\]

Finally the local image patch is modelled with a Gaussian centered at template \( C^i \)

\[
\Psi_t(P^i_t, C^i) = Pr(Im_t(P^i_t)|P^i_t, C^i) \quad \propto \quad \exp^{-||Im_t(P^i_t) - C^i||^2}
\]

In the Equations 3.3 and 3.5 \( I \) is the standard identity function. The operator \( D \) taking two positions as input variables determines the distance between two puppet rectangles \( i \) and \( j \) [FH05]. As described in section 2.1, typically the MAP in the pictorial structure (e.g. Figure 3.2-(b)) must be found to track a person over time in a video.

3.2 Model Building

The system is designed to automatically generate a model from a video without any manual initialization. The model of a human body (a puppet of rectangles) needs to be derived to guarantee a proper object tracking process. Since tracking is done by model detection in each frame, the model building step becomes an essential part in the entire system.

The fundamental assumption is that, "coherence in appearance is a stronger cue to body configuration than dynamics because body segments may move very fast but it takes time to change clothes" [Ram05]. The model being derived is an appearance model
telling what a person looks like, e.g. red hair. In the following sections, it will be described how the bottom-up and top-down approaches operate.

### 3.2.1 Bottom-Up

The bottom-up process looks for all possible limbs $i \in \{1, \ldots, N\}$ in each frame $t \in \{1, \ldots, T\}$ until the appearance of the template $C_i$ is learned. This approach works in three stages:

1. Detect human body parts
2. Cluster returned image patches
3. Purge unnecessary clusters

The generic person model consists out of 9 different limbs as seen in Figure 3.2-a)

#### 3.2.1.1 Detect human body parts

The body parts appear as rectangles on images. A rectangle detector using a Haar-like bar template is shown in Figure 3.3. An original image convolved with that particular template of a light bar adjacent to dark background pixels returns values, where only those exceeding a predefined threshold are of interest. One can expect some false-positives and missing results from that rectangle detector. The linearity of the convolution can be exploited and a left-bar template and right-bar template can be applied to the same image. Taking the minimum response of these two convolved images leads to a more robust detector.

![Diagram](image)

**Figure 3.3:** The minimum edge response of two bar-template applied images. In this case light pixels are emphasized, with little modifications dark pixels can be supported.
3.2.1.2 Cluster returned image patches

The drawback in the first place of the detector introduced in section 3.2.1.1 is, that many segment-like objects are returned. Another issue is, that the number of people shown in the image is unknown to the system, even if it was known, the system would not be able to distinguish between pixels representing a person and non-person pixels a priori.

The next step in the bottom-up approach is to cluster all returned image patches of the first $K$ frames used for model building. Clustering is the process of partitioning a data set into subsets (cluster) so that all elements in a cluster share common settings. The only difficulty is the uncertainty of existing clusters and given there are natural clusters in this data set, the quantity is still unknown, like in many other applications. If the Hopkins-Index [AKJ88] $h_i$ of the analysed data is $h_i > 0.5$, it is very likely that clusters exist. An exact number of clusters can be estimated with e.g. the partition’s coefficient [Bez81].

Having that in mind, it is obvious that the k-means algorithm fails. The mean-shift [CM02] is a non-parametric technique for the analysis of multimodal feature space. Each segment gets a 512 dimensional feature vector (3 color axes, $8^3 = 512$). Across all $T$ frames all identified segments represent a point in this feature space. A valid segment is a spatial scatter cloud represented by points, where at most one was retrieved from each frame. The mean-shift algorithm now determines the mean value of all these feature points within a hypersphere of radius $r_h$, receneters the hypersphere and repeats until a stop criterion (i.e. convergence) is reached. The algorithm terminates with a certain number of clusters and their corresponding centers.

At this point it is very likely that the clustered data set contains false-positive segments. As a final step in the model building pipeline, a clean-up process zeroes out unnecessary clusters.

Figure 3.4: These four images a)-d) sketch the bottom-up model building process for a torso. a) shows the result of the rectangle detector with some false-positives and one missing torso. b) shows all identified candidate torsos including background segments. c) shows 12 clusters, where all but the first three get pruned and shown in d).
3.2.1.3 Purge unnecessary clusters

$Pr(P_t^i|P_{t-1}^i)$ from the standard model on the RHS of Equation 3.1 stands for the motion model in that system. The preceding step created clusters, containing image-segments, that might be moving over time in the video. Each cluster must now be checked with the bounded velocity motion model whether it qualifies for or not. An appearance model is fitted to each cluster, usually a Gaussian, with its mean value at the cluster’s mean and standard deviation computed from the cluster. The candidates obtain their likelihood under the Gaussian appearance model. Dynamic programming returns a set of segments of constant velocity lying near the cluster center. All those sequences that never move are purged. Figure 3.4 shows the entire bottom-up approach with its final result in image d).

3.2.1.4 Approximate Inference

Each remaining cluster in section 3.2.1.3 is considered as a unique person. In order to build a complete instance-specific person model, the other 8 missing limbs need to be detected. The standard geometric likelihood $Pr(P_t^i|P_{t}^{\pi(i)})$ is used to find all near torso segments as possible arm candidates, that run smoothly like the torso across all $T$ frames. A complete pictorial structure for a 9 segment puppet, see Figure 3.2-a), can only be achieved by the use of a high quality segment detector. The lower arm, lower leg and torso detector work best. All remaining limbs are hard to distinguish between background pixels, because their appearance is either too small, fuzzy or occluded.

Tracking refers to finding a valid sequence of a torso in each frame, that fulfills the similarity measurement with the constant appearance model and moves smoothly over time. All other limbs have the same requirements. In order to complete the instance-specific person model, all identified limbs are checked with the geometric likelihood model to see which limbs belong to one person. Figure 3.2-b) shows an arm-torso assembly, where the image appearance is masked out for simplicity.

3.2.2 Top-Down

The bottom-up approach described in the previous section operates well, if and only if all parts of the human body can be absolutely reliably detected. Additionally the overall detection mechanism needs a certain amount of frames in order to build a working model for each existing person in a video.

Another way of getting a model of appearance is by detecting the entire person in one step, also known as a top-down approach. This is difficult, because people can take in awkward poses, which ends up in occluded limbs depending on the position of the capturing camera. In other words, the part appearance $C^i$ is unknown a priori. However
the system is not built to detect persons, rather to learn appearance models and detect those in every following frame.

Nevertheless the model must be retrieved by a person in a particular pose. Figure 3.5 shows a few possible poses. The temporal pictorial structure from Equation 3.1 limited to a single time frame can be fed best with a lateral walking pose. That body configuration is easy to detect, because both legs and at least one arm are visible to the observer. The 9 part generic person model becomes an 8 part person model. The detector will therefore fire as soon as it has detected a lateral walking pose, but can keep track even if the pose changes to weird and unknown poses, because a model of appearance was learned. The top-down process works in two stages

1. Pose detection

2. Discriminate appearance model

Figure 3.5: These six images show the difficulty and variety of arbitrary poses in a sport event. Image a) is a perfect example for self occlusion, which does not exclude occlusion by extrinsic objects. The pose in b) is application-specific and might not be found in daily life or other camera observed areas. Motion blur occurs exactly when objects move quicker than the camera shuts. In the non-distinctive pose it is fairly hard to separate the torso and the two legs, whereas in the bottom middle the actual point of interest is hard to define. The lateral walking pose in the bottom right stands for a frequently occurring pose.

3.2.2.1 Pose Detection

With the help of a stylized person detector generated by the restricted pictorial structure from Equation 3.1, a person can be detected in lateral walking poses within one frame. Figure 3.6 illustrates the top-down approach. Since we know that one arm is hidden in lateral walking, there are only 8 parts left to look for. Additional modifications must be made in order to perform correctly.
• The geometric likelihood $Pr(P^i|P^{\pi(i)})$ must be adapted for torso and upper legs combo. The angle $\theta$ between the torso and the two upper legs span a restricted angle between 15 and 45 degrees, because of the shape of an upside-down turned V in a laterally walking position.

• The local image patch likelihood $Pr(Im(P^i)|P^i, C^i)$ will be evaluated with a chamfer template [ATC03]. "Given an image, the chamfer cost of an edge template is the average distance between each edge in the template and the closest edge in the image" [Ram05]. Edge-pixels are quantized in 12 directions and the cost is computed for each direction and cumulated afterwards. By applying rotated representatives of the template, all rectangle instances can be covered.

The response of chamfer costs gets non-maximum suppressed, so that only values above a predefined threshold are kept. Detected arm candidates in horizontal or vertical positions are declined, because one can expect many of those as background information, which leads to false-positive results.

Figure 3.6: Top-Down approach at a glance. The detector gets one single frame from "Run Lola Run" as input. Limbs are expected to be positioned for a walking left person, but the system also searches the space with a mirrored model. After a score segmentation was found, the false detections get purged by classifying pixels into limb and non-limb pixels, assuming a quasi constant color for each limb.

### 3.2.2.2 Discriminating Appearance Model

The previously introduced person detector segments the image in person pixels and non-person pixels. Now a quadratic logistic regression classifier is trained with the pixels of the limbs, declaring pixels inside as positive and outside as negative. Such an appearance model denotes a quadratic surface in RGB space, which splits it into limb and non-limb pixels. Section 3.4 demonstrates the model's generalization abilities.

### 3.3 Model Detection

Whether the models have been learned by bottom-up or top-down, either way the model detection can get started with an appearance of each part $C^i$. As introduced in section 3.2.1.2 the local image likelihood is determined with the use of an RGB histogram and distance measurement within a hypersphere. The classifier’s quality is given by the
accumulated amount of misplaced pixels. $C^i$ is good enough to detect a person with a single frame pictorial structure in of course each frame.

An automatic people tracker always suffers from occlusion in certain situations reflected in images, see Figure 3.5-a). Humans move more or less unpredictable and so it’s not preventible to capture them in atypical poses. Figure 3.7 shows superimposed samples from the posterior of $Pr(P^{1:N}|Im, C^{1:N})$ for a single arm, single leg pictorial structure. Surprisingly the map shows two disjoint legs belonging to different modes in the posterior, which can be reasoned by the uncertainty in the matching procedure. The mean-shift [CM02] algorithm iteratively finds the modes indicated by the highest posterior value above a threshold. The mean-shift spatially smooths the posterior function $Pr(P^{1:N}|Im, C^{1:N})$ in proportion to its bandwidth, which leads to a sub-pixel accuracy.

The bottom-up approach takes care of multiple people instances in a video by concerning multiple cluster centers and pruning those containing segments that do not move. The top-down approach returns multiple walking-pose candidates out of which the correct number of different people being present must be derived. By fitting a Gaussian for each limb mask, the returned vector can again be clustered in RGB space to obtain all people models using mean-shift. Clusters containing similar appearing people, positive and negative examples are used to train the regression for each human body part appearance. In a video with many equally looking persons, e.g. military parade the posterior will return many modes of disjunctive soldiers. The procedure of the previous paragraph retrieves all unique detections. It fairly happens that one single appearance model of a part $C^i$ appears multiple times in a single frame. The system first assigns $C^i$ to the optimum-fitting local image part and searches all remaining non-overlapping instances for the one that qualifies the best, which gets $C^i$ assigned. This greedy procedure repeats from the beginning, which works perfect for images with well seperated people in it.

![Figure 3.7](image)

Figure 3.7: This image shows 1000 samples superimposed from the posterior for a single arm and single leg pictorial structure. The uncertainty implies the presence of two arms and two legs, because the e.g. two different arms correspond to two different modes, derived by the mean-shift procedure. The bandwidth-proportional smoothing enables to find the foreshortened arm.
### 3.4 Evaluation

In this section the two model building strategies are compared from a performance and field of application prospective. It is important to keep in mind, that tracking does not work with person detection, but model detection and so the performance highly relies upon the quality of the persons template derived by either one strategy.

#### 3.4.1 Bottom-Up

Table 3.1 shows the performance of the bottom-up model builder applied to four different sequences. In the first two video sequences 75/100 and 150/288 frames were used for clustering, whereas for the other two 200/380 and 150/300 were reserved for clustering. A correct localization is reached, if the majority of the pixels was properly detected. As expected it is very complicated to detect small parts like arm, especially in low contrast images or simply because of the small appearance of those limbs.

The system itself does not need any manual initialization at all, only a threshold for the segmentation operator and a bandwidth for the mean-shift must be defined in the first place. But parameters like that could probably be estimated by adding further intelligence to the system. Nevertheless the system is able to track people without a strong motion model, because it detects the learned appearance model and does not look for people in the image anymore.

Another remarkable feature of the model detection technique is, that the system recovers automatically from errors and occlusion. Figure 3.8 shows a sample of finally three persons running in a weave fashion, while the system recovers from occlusion and tracking errors automatically.

So far it is unknown how many frames are needed to achieve an adequate instance-specific model in a video sequence. It turns out, that no matter if only the first $K$ frames were used for clustering or all $T$ frames in the video-clip, the results in both ways are equivalent concerning their accuracy. This insight implies that the implemented model is fairly generalizing the person’s appearance.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Torso</th>
<th>Arm</th>
<th>Leg</th>
</tr>
</thead>
<tbody>
<tr>
<td>J. Jacks</td>
<td>94.6</td>
<td>87.4</td>
<td>91.8</td>
</tr>
<tr>
<td>Walk</td>
<td>99.5</td>
<td>84.3</td>
<td>68.2</td>
</tr>
<tr>
<td>Street Pass</td>
<td>91.2</td>
<td>57.4</td>
<td>38.7</td>
</tr>
<tr>
<td>Weave Run</td>
<td>90.3</td>
<td>21.2</td>
<td>61.8</td>
</tr>
</tbody>
</table>

Table 3.1: % of frames correctly localized
Figure 3.8: These six sample images illustrate the detection, tracking and recovery processes on a video sequence with people walking in a weave-fashion. Since the system works by detecting a previously learned appearance model, the recovery from occlusion or even from error works well (two right images).

3.4.2 Top-Down

The top-down approach is faster than bottom-up, because it operates on a single frame, whereas bottom-up needs the first $K$ frames in order to cluster segment candidates and derive the actual moving objects (persons). This strategy can be divided up into two stages, firstly the stylized pose detector to generate a segmented appearance model, which can then be tracked by detection. It is unusual hard to tell when the lateral walking pose detector fails, because it searches the frame only for a lateral walking position to the left or right. Anyhow, the detector simply fails, if a person in lateral walking pose is present and the system does not fire.

The model building in a top-down fashion was tested on a few different sequences. To make an automatic initialization possible, all videos were scaled in such a way, that appearing people are approximately of the same size.

Table 3.2 shows the performance of correctly localized limbs with the appearance model built by the stylized pose detector. It turns out again, that the torso detection works best for both methods in all sequences. A correct localization of arms in park scenes is hard, because arms are small, often occluded, move fast and reflect low contrast. On the flipside the results on the commercial video sequences are amazing keeping in mind the complexity of the background.

In general it can be said, that top-down works better, because the detector only fires if people in the image appear in a particular predefined lateral walking pose. Out of this simple to detect pose, an appearance model can be built easily. On the other side it can happen, that people never get registered by the system, because they never take in a lateral walking pose.

A drawback in both methods and a general detection problem are brightness changes
over time caused by moving objects casting a cloud or simply by a *moving sun*. A possible way to improve the detector and keep a more robust tracking procedure can be achieved by adding an illumination variable to the constant appearance model $C^i$. The brightness changes can be taken into account in all detection processes and backpropagation could be applied if different illumination is noticed.
Chapter 4

Summary

*I hope you become comfortable with the use of logic without being deceived into concluding that logic will inevitably lead you to the correct conclusion.*

- Neil Armstrong

This final chapter will sum up all arguments, balances the pros and cons and concludes with a personal evaluation of people tracker in general.

The requirements, like automatic initialization, track independent of activity, track multiple people, track through occlusion, be computationally efficient firstly assigned to the system’s requirements list were mainly considered and implemented. In fact a manual initialization is not needed. In spite of everything there is still the need of a manual adjustment of thresholds and the bandwidth for the mean-shift algorithm and in some cases a proper scaling of the videos is needed, because objects (persons) are not searched over all scales. The well-known issues in people recognition still exist, talking about small and quickly moving limbs, can not be solved by 100 % by this work.

Independent of the model building strategy, the fundamental assumption that “coherence in appearance is a stronger cue to body configuration than dynamics” is a felicitious step on the way to a reliable people tracker. The system seems to be working in real-time real well and can adapt even in videos containing strong lightning changes.

Today’s people or object tracker in general are able to analyze many situations automatically. Those systems offer chances for an enrichment for mankind. On the flipside it can take away one’s freedom and anonymity. From my point of view such a system observing a company’s rooms endangers everyone’s public appearance and produces glassy citizens. One should not aspire to build systems spying on each and everyone’s behavior.
Bibliography


