Sensor specific distributions for improved tracking of people

Rik Bellens, Sidharta Gautama, Johan D'Haeyer

Telin, University of Ghent, St.Pietersnieuwstraat 41, B-9000 Ghent, Belgium

ABSTRACT

In this paper, we examine sensor specific distributions of local image operators (edge and line detectors), which describe the appearance of people in video sequences. The distributions are used to describe a probabilistic articulated motion model to track the gestures of a person in terms of arms and body movement. The distributions are based on work of Sidenbladh where general distributions are examined, collected over images found on the internet. In our work, we focus on the statistics of one sensor, in our case a standard webcam, and examine the influence of image noise and scale. We show that although the general shape of the distributions published by Sidenbladh are found, important anomalies occur which are due to image noise and reduced resolution. Taking into account the effects of noise and blurring on the scale space response of edge and line detectors improves the overall performance of the model. The original distributions introduced a bias towards small sharp boundaries over large blurred boundaries. In the case of arms and legs which often appear blurred in the image, this bias is unwanted. Incorporating our modifications in the distributions removes the bias and makes the tracking more robust.

Keywords: sensor modelling, image statistics, tracking, particle filter

1. INTRODUCTION

Tracking humans is not an easy task. A system is needed that is general enough to capture all the variations in human appearance, but at the same time is specific enough to be able to distinguish between humans and other objects with similar structures.

Many methods have been proposed to track people (for surveys see[1, 2]). The complexity of the system depends on the desired level of detail in which the pose and movement of the human is described and of the a priori made assumptions about the appearance of people and background. For example, in surveillance applications, we might merely be interested in whether or not a person is present, in human-machine interfaces the machine should be able to understand the gestures of the person and in virtual reality applications a complete three dimensional description of shape, pose and movement is needed. Most of the present solutions constraint the environment by making some assumptions about the appearance of people and background. For example, they might assume a static background, people with special clothes, no moving objects other than humans, etc.... The system we shall adapt is based on an article by Sidenbladh.[3, 4] This system results in a three dimensional description of the pose of a person in every frame and is applicable in an unconstrained environment.

This system consists of two parts: the temporal model and the appearance model. The task of the temporal model is to predict in a probabilistic manner a new pose of the human model based on a prediction of the old pose. The task of the appearance model is to calculate a match metric between a predicted pose and an image frame. In a formal way, the match metric is defined as the probability that a person in the given pose is present in the current frame and is calculated by comparing the actual image information, in our case edge and ridge responses, with the expected image information when a person in the given pose would be present. To model the expected image information we learn the distributions of the filter responses, both on and off people. These two parts are used in a Bayesian framework for tracking people. We shall use the particle filter[5, 6] to solve the tracking.

Sidenbladh[3] estimated general distributions based on (high quality) training images found on the internet and taken by different cameras. In this work we examine the statistics of one camera. By doing this, we are able to model sensor specific noise and blur, which leads to better recognition of the correct state. Effects of spatial

Rik Bellens: E-mail: rik.bellens@telin.ugent.be

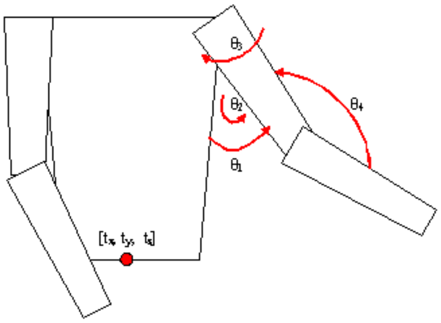


Figure 1. the torso and arms of the 3D human model

correlation of image information will be reduced by means of a correction exponent. This dependency correction will lead to more robust tracking.

In the next section we explain the different parts of our tracking system. In section 3 sensor specific distributions will be examined, the differences with the distributions of Sidenbladh will be explained and the advantages of the sensor specific distributions will be demonstrated. Section 4 will present some results of tracking experiments and show the advantages of the dependency correction. In section 5 our conclusions and some ideas for future work will be formulated.

2. TRACKING FRAMEWORK

A pose of the person in the video will be described by a state of the human model. We shall use a 3D articulated model of the human body in which every limb is represented by a cone. The individual limbs are connected with joints. The angles of these joints, together with the global translation and rotation of the model, are the parameters of the human model ϕ_t. In this manner we can represent a human model with 25 parameters. Figure 1 shows the upper part of the human model.

In a tracking context we want to assign to each frame of a video sequence a state of the human model. For the first frame of a sequence many states are possible, for the following frames only those states which are close to the estimated state of the last frame, are likely. As a consequence, the initialisation problem is much more complicated than the tracking problem. Therefore, we shall manually initialise the system with the correct state of the first frame. The tracking is performed by the particle filter or condensation algorithm.[5,6] Herein, state space distributions are represented by weighted particles, in such a way that in areas with high probability many particles, with high weights, occur and in areas with low probability few particles, with low weights, occur. Every particle represents a possible state of the human model. The condensation algorithm consists of three phases. Initially, the particles represent the posterior distribution $p(\phi_t|\overrightarrow{F_t})$ of the pose of the person in the frame at time t, given all the image information until time t. In the first phase, the state of every particle is propagated in time. This means that based on the old state and based on a model of human motion, i.e. the temporal model, a prediction of the state at time $t+1$ is made. The particles now represent the prior distribution $p(\phi_{t+1}|\overrightarrow{F_t})$ of the pose of the person in frame $t+1$, given all the image information until time t (equation 1). In the second phase, every particle is weighted with the match metric, this is the probability of observing the current frame given the pose of the person $p(F_{t+1}|\phi_{t+1})$. Next, the particles are normalized to one. The particles now represent the posterior distribution of time $t+1$ (equation 2). The maximum of this distribution is the estimation of the pose at time $t+1$. In the last phase, a Monte Carlo resampling is performed, which results in a set of particles with equal weights, still representing the same posterior distribution. Because of this, unlikely particles will disappear and likely particles will multiply, which makes it possible to test more states in the likely areas in the next iteration.

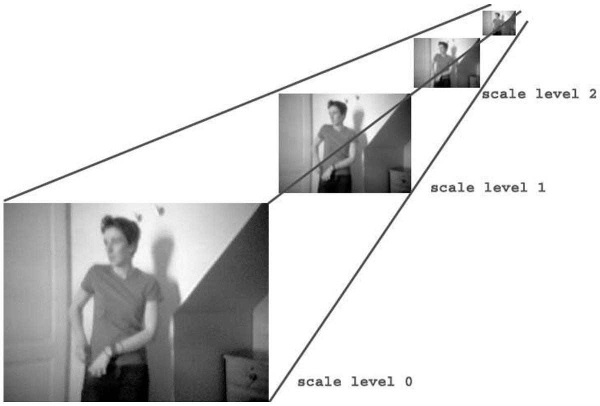


Figure 2. scale pyramid

$$p(\phi_{t+1}|\overrightarrow{F_t}) = \int p(\phi_{t+1}|\phi_t)p(\phi_t|\overrightarrow{F_t})d\phi_t \qquad (1)$$

$$p(\phi_{t+1}|\overrightarrow{F_{t+1}}) = \kappa p(F_{t+1}|\phi_{t+1})p(\phi_{t+1}|\overrightarrow{F_t}) \qquad (2)$$

2.1. Appearance Model

The task of the appearance model is to calculate a match metric between a state of the human model and the current frame. In our case, this is done by comparing the location of the edges and ridges in the frame with the location of the edges and ridges of the limbs in the human model. The human model is three dimensional, the image is two dimensional. Therefore, we will first have to project a state of the human model onto the two dimensional frame.[4]

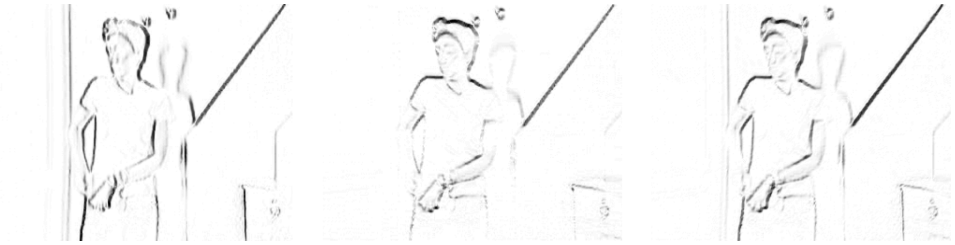
Secondly, we will have to extract the information about edges and ridges from the image. This information will be investigated at different scales. Therefore, a scale pyramid of the image will be created. The original frame is the image at scale level 0. An image at scale level i is created by filtering the image at scale level $i-1$ with a Gaussian kernel and subsampling it (Figure 2). We shall use up to 4 levels (0-3) for the edge cue and up to 6 levels (0-5) for the ridge cue.

In every point, which belongs to the edge of a limb according to the state of the human model, we shall calculate the edge response for the different scale levels. This is done by applying the first derivative filters in vertical (figure 3(a)) and horizontal (figure 3(b)) direction and combining these responses for an arbitrary direction θ (figure 3(c))as in equation 3. For points located between pixel locations the edge responses are calculated by linear interpolation of the edge responses of the four surrounding pixel locations.

$$f_e = \sin\theta f_x - \cos\theta f_y \qquad (3)$$

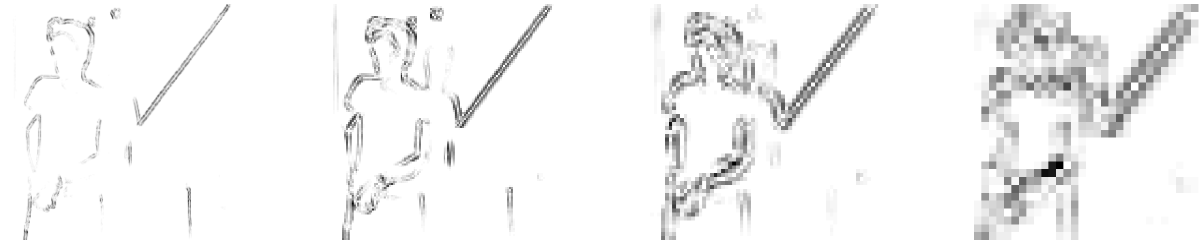
Analogously, in every point, which belongs to the axis of a limb according to the state of the human model, we shall calculate the ridge response. The ridge response is calculated by first applying the second derivative filters in horizontal, vertical and diagonal direction. These filter responses are then combined to give the second derivative in the direction of a limb in in the direction perpendicular to it. The difference in magnitude between these two values gives an indication of the linearity of the structure (equation 4).

$$\begin{aligned} f_r &= |\sin^2\theta f_{xx} + \cos^2\theta f_{yy} - 2\sin\theta\cos\theta f_{xy}| \\ &\quad -|\cos^2\theta f_{xx} + \sin^2\theta f_{yy} + 2\sin\theta\cos\theta f_{xy}| \end{aligned} \qquad (4)$$



(a) vertical direction (b) horizontal direction (c) left under arm direction

Figure 3. edge responses in vertical direction (a), horizontal direction (b) and in the direction of the left under arm (c)



(a) scale level 0 (b) scale level 1 (c) scale level 2 (d) scale level 3

Figure 4. ridge responses in the direction of the left under arm for scale levels 0-3

A ridge of a limb will best show on a particular scale depending on the width w of the limb. Therefore, we shall only use the ridge information of one scale level. A formula for the optimal scale level σ was established by Sidenbladh[3] (equation 5). Figure 4 shows the ridge responses in the direction of the left under arm for different image scales. As can be seen, scale level 3 gives highest responses on the left under arm.

$$\sum_{i=1}^{\sigma} 4^{i-1} = -24 + 4.45w \tag{5}$$

Once the edge and ridge responses of a point are known, we would like to determine the probability that this point actually belongs to the edge or ridge of a limb (p_{on}) and the probability that this point belongs to the background (p_{off}). This can be done by looking up the observed filter responses in learned distributions of fore- and background. Sidenbladh observed that better results - this means more deviation between the distributions for foreground and background - can be accomplished when applying local contrast normalization prior to calculating filter responses.[3]

In the system used by Sidenbladh,[3] information from different points, different levels, different cues and different limbs are fused in a naive Bayesian manner. This means that independence is assumed between different information sources. Equation 6 shows the calculation of the match metric. Herein, X_l^i is the set of locations belonging to the edge or ridge of limb l and f_i^σ is the filter response of cue i at level σ.

$$p(F_t|\phi_t) = \kappa \prod_{l \in \text{limbs}} \prod_{i \in \text{cues}} \prod_{\sigma \in \text{scales}} \prod_{x \in X_l^i} \frac{p_{on}(f_i^\sigma(x))}{p_{off}(f_i^\sigma(x))} \tag{6}$$

Since pixel intensities in natural images are highly correlated for neighbouring pixels, the former equation will not result in a correct estimate of the probability. Because of the high correlation, the state space distributions calculated with the naive Bayesian method, are very peaked. As a result, when resampling the states in the particle filter, almost all the states will disappear and only a few (mostly only one) will survive. As was shown in[7] the number of particles needed for tracking an object, and thus the time needed for performing the tracking, increases polynomial with the decrease of the survival rate α of the states (equation 7). Therefore, it is very important not to overestimate the peaks of the state space distribution. Sidenbladh tried to reduce this problem by randomly sampling a number of points on the foreground, thus limiting the set X_l^i. However, to eliminate the influence of the correlation, one has to choose very little sample points, which results in less stable estimates. We shall artificially flatten the distribution and thus increase the survival rate, by using a correction exponent δ when calculating the match metrics ($W = W_{naive}^{\delta}$). Good values for δ will depend on the number of correlated information points that are fused. We choose δ equal to $\frac{2}{N}$, where N is the number of information points used for every limb.

$$N \geq D_{min}/\alpha^d \tag{7}$$

2.2. Temporal Model

The Temporal Model is used to propagate a state of the human model in time. This is done in a probabilistic manner in order to simulate the probability that a human is standing in a specific pose given the pose(s) he was in in the previous frame(s). Different kind of Temporal Models have been proposed. Typical characteristics of human motion, for example the cyclic character of many human actions,[8] can be exploited. In[9] a temporal model is proposed, which is based on a database of motion examples. We use an adaption of the smooth motion model used by Sidenbladh[4] which is very simple model, and for which little training is necessary.

The smooth motion model assumes all motion to be linear with Gaussian errors. In the smooth motion model a state of the human body consists of a number of position parameters and an equally number of velocity parameters. Every parameter will be propagated independently. A position parameter is propagated by adding the corresponding velocity parameter and a random Gaussian distributed value with mean 0 to the old value of the position parameter. Values for the variance were empirically determined by Sidenbladh.[4] Originally the velocity parameter is propagated by adding white Gaussian noise to the old velocity parameters. These new velocity parameters will not influence the pose of the human model in the current frame. As a result only the position parameters are scored by the Appearance Model and a state with erroneous velocity parameters can get a high match measure. Therefore, we have adapted this system to calculate the velocity parameters based on the difference of the old and the new position parameters. The reduce system noise we let the velocity parameters change slowly by using a weighted mean of this difference and of the old velocity parameters. Weights around 0.5 give good results.

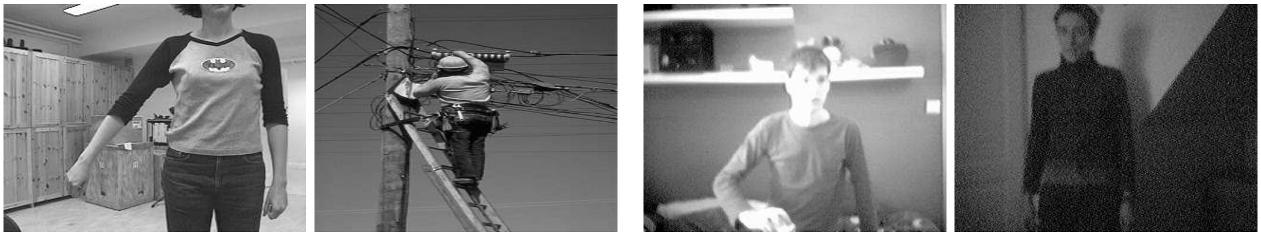
3. LEARNING SENSOR SPECIFIC FILTER DISTRIBUTIONS

The training images used by Sidenbladh[3] were collected from the internet, and were taken by different, but high quality cameras. This general training set, as well as the annotations of limb edges, can be found on the internet*. In our work we use a sensor specific training set with images of one camera only, namely a standard webcam. These images are typically noisy and blurry. Example images of the two training sets are shown in figure 5.

Our training set consists roughly out of one hundred images. Every image in the training set was manually annotated with the edges of the limbs. To learn the distributions of edge responses, we randomly sampled points on the edges of the limbs and calculated the edge responses in those points. We then constructed the distributions by dividing the edge responses in bins and counting the number of responses in each bin. This was done for different limbs and different scales separately. For the background distributions we randomly sampled points on the background and calculated the edge response for random directions. Analogously, we randomly sampled points on the axis of the limbs to learn the ridge distributions.

Figure 7 shows the logarithm of the distributions of the edge responses on the background and on the lower arm for different scales based on the webcam training set. The distributions for the background show a maximum at 0, whereas those for the foreground show two extra sub maxima at 1 and −1. As expected high edge responses (in absolute value) are more likely to belong to a point on the edge of a limb than to a point on the background.

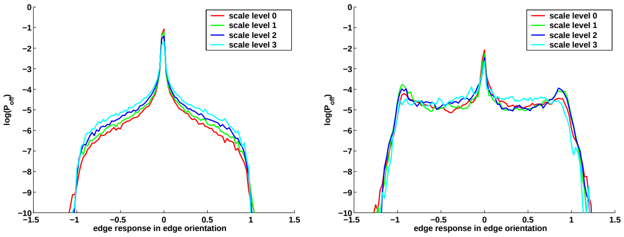
*http://www.nada.kth.se/˜ hedvig/data.html



(a) general

(b) webcam

Figure 5. examples of general (a) and webcam (b) training images



(a) background

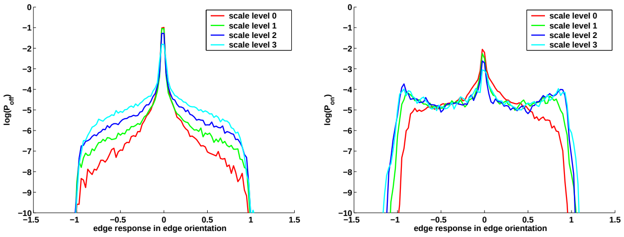
(b) lower arm

Figure 6. Logarithm of the edge distributions for the background (a) and lower arm (b) based on the training set of Sidenbladh

The difference between the distributions of the background and those of the foreground make it possible to distinguish between points on the foreground and points on the background.

For comparison, figure 6 shows the same distributions based on the training set used by Sidenbladh. As can be seen, the distributions of the webcam are slimmer than those of Sidenbladh. Lower edge responses are observed. Additionally, where the distributions found by Sidenbladh are more or less equal on different image scales, the webcam distributions are clearly different on different image scales.

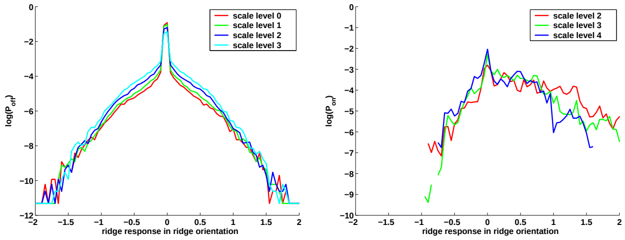
Figure 9 shows the logarithm of the distributions of the ridge responses on the background and on the lower arm for different scales based on the webcam training set. Figure 8 shows the same distributions based on the Sidenbladh training set. On the background distributions we see a peak at 0 and symmetric distributions around this peak. On the foreground as well, we see a peak at 0, but on the negative side the probability decreases very fast, while on the positive side the probability decrease much slower. Indeed, when looking on the axis of a limb we expect low gradients in the direction of the limb and high gradients in the direction perpendicular to the limb (see equation 4). These foreground distributions, however, are very noisy. This is due to the fact that ridges are



(a) background

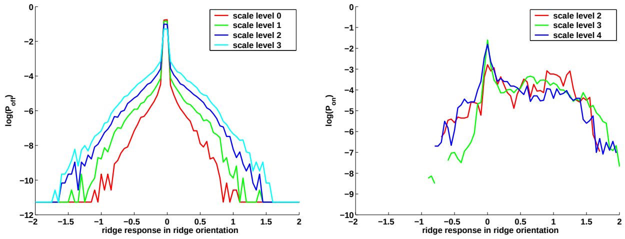
(b) lower arm

Figure 7. Logarithm of the edge distributions for the background (a) and lower arm (b) based on the webcam training set



(a) background (b) lower arm

Figure 8. Logarithm of the ridge distributions for the background (a) and lower arm (b) based on the training set of Sidenbladh



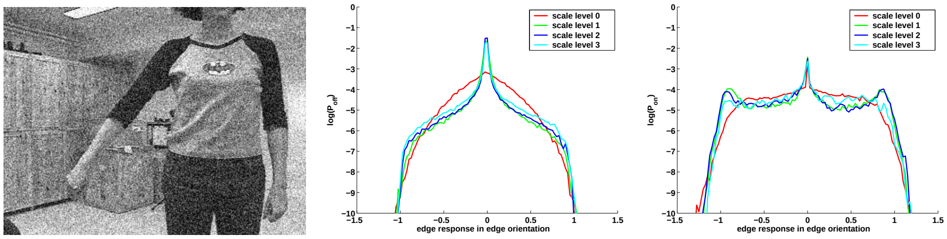
(a) background (b) lower arm

Figure 9. Logarithm of the ridge distributions for the background (a) and lower arm (b) based on the webcam training set

only investigated at an optimal scale level and as a result, the distributions are estimated based on less data. Unlike the distributions found by Sidenbladh, our distributions differ clearly for different scale levels, at least for the background distributions. Because of the noisy estimates of the foreground distributions, whether or not these distributions are equal for different scale levels can not unquestionable be determined. We assume that they will stay more or less equal, so that we can estimate distributions over all scales and thus over more data. This will result in more stable distributions.

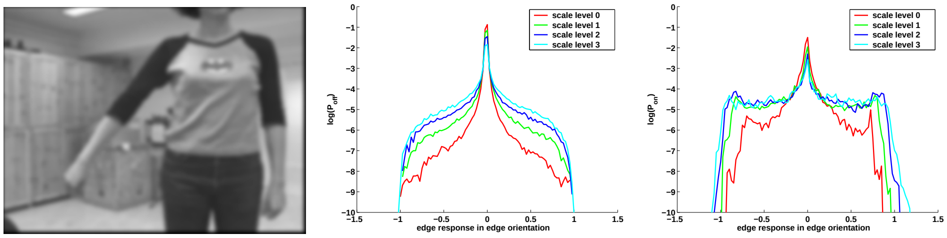
Ruderman[10, 11] showed that the statistics of natural images are equal for different scale levels. Nevertheless, our distributions show clear differences between scales. We show that this contradiction is the result of the blurry and noisy images. First, we examine the effect of noise on the distributions. For this purpose, we add white Gaussian noise to the training set of Sidenbladh and learn the distributions for this new training set. The resulting distributions for edge responses are shown in figure 10. Analogous results are found for ridge responses. Especially the distribution at scale level 0 changes quite a lot. The peak at zero decreases, and more high responses occur. The effect on the higher scale levels is negligible. This is logical because filtering with a Gaussian kernel has the property to reduce the noise. Ruderman[10, 11] as well, observed that noise increases the energy of high spatial frequencies - edge responses at scale level 0 can be seen as high spatial frequencies -, while lower spatial frequencies - a Gaussian filter is a low pass filter - are not affected. Since in the webcam training set, low edge responses are more frequent and high edge responses are less frequent than in the Sidenbladh training set, noise alone can not explain the anomalies in the webcam distributions.

To examine the influence of blur on the distributions, we filter the training images of Sidenbladh repeatedly with a mean filter. The resulting distributions for edge responses are shown in figure 11. Analogous results are found for ridge responses. Although the influence decreases for higher scale levels, blur in contrast to noise affects all scale levels. The peak around zero increases and the higher edge responses become less likely. Because of the subsampling performed when generating the scale pyramid, the images are sharpened and the influence on the distributions of the higher scale levels decreases. Ruderman[10, 11] as well, observed that blur decreases the energy of high spatial frequencies and that the influence decreases for lower spatial frequencies.



(a) training image (b) background (c) lower arm

Figure 10. Logarithm of the edge distributions for the background (b) and lower arm (c) based on the training set of Sidenbladh with white Gaussian noise added (a)



(a) training image (b) background (c) lower arm

Figure 11. Logarithm of the edge distributions for the background (b) and lower arm (c) based on the training set of Sidenbladh after repeatedly filtering with a mean value filter (a)

When calculating the match metric between a state and a frame, the ratio of the probability that a certain point belongs to the foreground to the probability that that point belongs to the background is used (equation 6). These ratios for the webcam dataset and the Sidenbladh dataset are compared in figure 12 for the edge responses. The Sidenbladh distributions assign rather low scores to edge responses between -0.75 en 0.75. Outside this interval the ratio increases fast to peaks at -1.2 and 1.2 after which it decreases again. The webcam distributions at scale level 0 on the other hand, already assign relative high scores to edge responses just above and below 0, where the ratio increases slowly to relative low peaks at -0.75 and 0.75. Edge responses beyond these peaks are assigned very low values. For higher scale levels the webcam distributions approaches the Sidenbladh distributions.

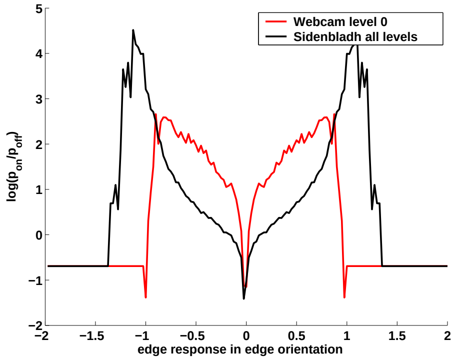


Figure 12. Logarithm of the ratio of lower arm and background distributions for edge responses. In red: scale level 0 of the webcam distributions, in black: Sidenbladh distributions over all scale levels

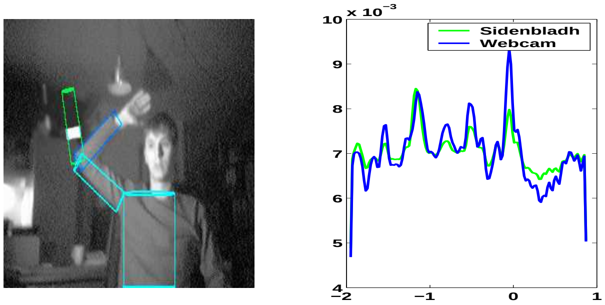


Figure 13. Left: the optimal state according to the Sidenbladh distributions (green) and the webcam distributions (blue). Right: the match metric for different states in function of the fault on the elbow angle

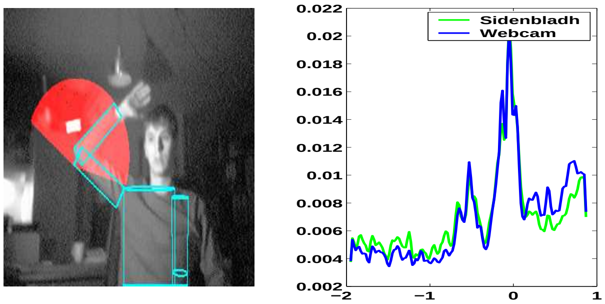


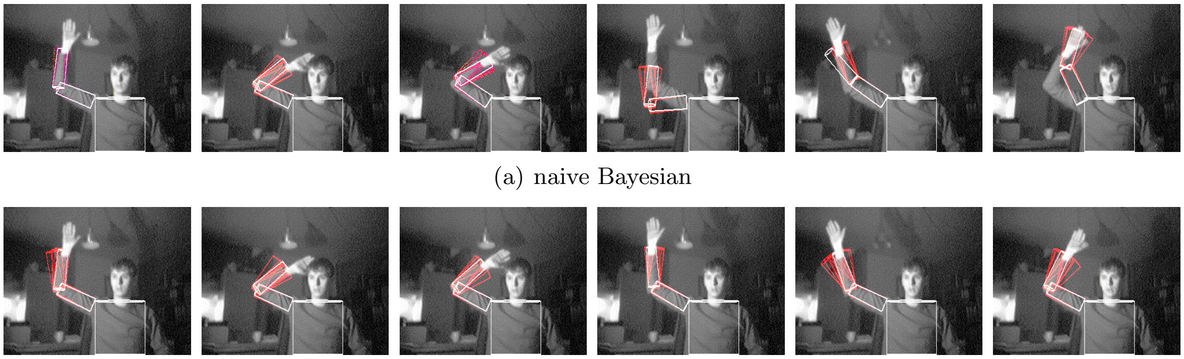
Figure 14. Left: the optimal state according to the Sidenbladh distributions (green) and the webcam distributions (blue). Right: the match metric for different states in function of the fault on the elbow angle

Because of the shape of the ratios, the distributions of Sidenbladh will especially respond to sharp edges and ridges, while the webcam distributions will respond to less sharp edges and ridges. To examine the influence of the training set on the match metric we conduct the following experiment. The match metrics between an image and a number of states of the human model are calculated (only the edge cue is used), both with the Sidenbladh distributions and with the webcam distributions. These states all differ in only one parameter, the elbow angle. As can be seen in figure 13, the Sidenbladh distributions are easily mislead by a small structure which contains sharp edges. Even when the edges of a projected state coincide for only a small distance with the edges of this structure, with the Sidenbladh distributions, this state will be assigned a high match metric because of the dominant character of high edge responses. With the webcam distributions, high edge responses are less dominant, which results in a higher match metric for a state of which the edges coincide over a longer distant with blurry image edges.

Ridges are investigated at only one scale depending on the width of the limbs. The optimal scale is almost always 3 or higher, where there is little difference between the general distributions of Sidenbladh and the sensor specific distributions of the webcam. When repeating the previous experiment for the ridge cue, both the general and the sensor specific distributions give more or less the same result (figure 14).

4. TRACKING EXPERIMENTS

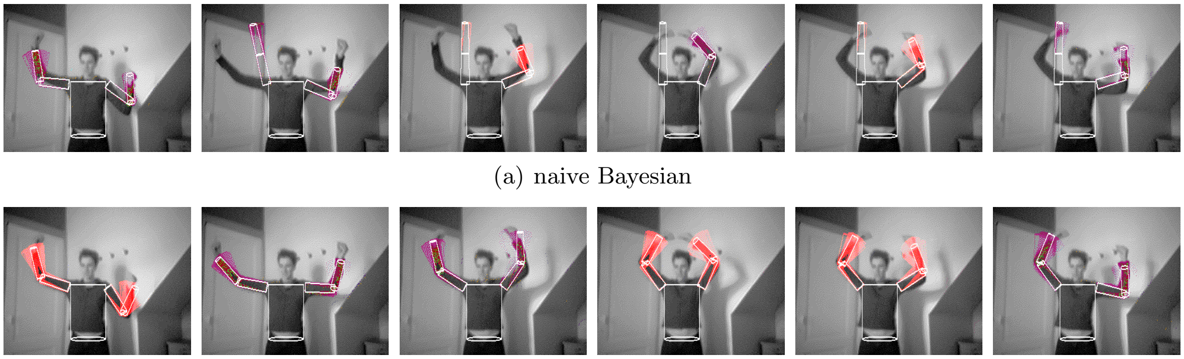
In our first tracking experiment, we try to follow a waving arm using the edge and ridge cue. Only the arm of the person moves and it only moves parallel to the image plane. As a result, we only need to estimate two parameters: one of the three shoulder angles and the elbow angle. the values of the other parameters are kept fixed. This simplifies the experiment a lot. We tried tracking with naive Bayesian fusion and with dependency correction and used thereby 10 particles. The results are shown in figure 15. Although the estimation for certain frames is not very good, when using dependency correction, we are able to recover from this. When using the



(a) naive Bayesian

(b) dependency correction

Figure 15. frames 130, 160, 190, 220, 250 and 280 of a tracking experiment of a waving arm (estimating 2 parameters) using 10 particles



(a) naive Bayesian

(b) dependency correction

Figure 16. frames 130, 160, 190, 220, 250 and 280 of a tracking experiment of a waving arm (estimating 2 parameters) using 10 particles

naive Bayesian method, once the estimation is no longer correct, the system can not recover. The dependency correction results in more robust tracking.

In the second tracking experiment, we try to track two arms, consisting of one of the tree shoulder angles and 1 elbow angle each, which gives a total of four parameters to estimate. One hundred particles were used. The results are shown in figure 16. When using the naive Bayesian way, the system looses track very fast. With dependency correction, the estimates are not always very precise, but the system is able to keep track. When trying to estimate even more parameters the number of needed particles increases dramatically.

5. CONCLUSIONS AND FUTURE WORK

Our system is able to track parts of a body in a robust manner. When tracking a full body two problems occur: one, the correct state cannot always be distinguished and two, the number of needed particles and thus the time needed, increases exponentially with the number of parameters d of the human model (see equation 7). Solutions for the first problem can be found in using more image information. This extra information can be extracted by using more cues, like texture, (skin) colour, motion, ... Modelling the spatial correlations more explicitly, might improve recognition of typical structures of clothes and skin.

Two strategies can be followed for trying to solve the second problem. One, by using a more advanced temporal model, which is more specific to human motion, better prior predictions of a new state can be made and less energy is lost in exploring areas which will not lead to the correct state. Two, one might try to lower the number of needed particles by using a variation of the particle filter. Several variations have been proposed in the literature: annealed particle filter,[12] partitioned sampling,[7] .... The possibilities and usability of these variations need to be further studied.

REFERENCES

1. D. Gavrila, "The visual analysis of human movement: A survey," *Computer Vision and Image Understanding* **73**(1), pp. 82–98, 1999.
2. T. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding* **81**(3), pp. 231–268, 2001.
3. H. Sidenbladh and M. Black, "Learning the statistics of people in images and video," *International Journal of Computer Vision* **54**(1-3), pp. 183–209, 2003.
4. H. Sidenbladh, *Probabilistic Tracking and Reconstruction of 3D Human Motion in Monocular Video Sequences.* PhD thesis, Universitet Stockholms, 2001.
5. N. Gordon, "A novel approach to nonlinear/non-gaussian bayesian state estimation," *IEE Proceedings on Radar, Sonar and Navigation* **140**(2), pp. 107–113, 1993.
6. M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *International Journal of Computer Vision* **29**(1), pp. 5–28, 1998.
7. J. MacCormick and M. Isard, "Partitioned sampling, articulated objects, and interface-quality hand-tracking," *European Conference on Computer Vision, ECCV* **2**, pp. 3–19, 2000.
8. D. Ormoneit, H. Sidenbladh, M. Black, and T. Hastie, "Learning and tracking cyclic human motion," in *Advances in Neural Information Processing Systems 13*, pp. 894–900, 2001.
9. H. Sidenbladh, M. Black, and L. Sigal, "Implicit probabilistic models of human motion for synthesis and tracking," in *Proc. 7th European Conference on Computer Vision*, **1**, pp. 784–800, (Copenhagen, Denmark), 2002.
10. D. Ruderman, "The statistics of natural images," *Network: Computation in Neural Systems* **5**(4), pp. 517–548, 1994.
11. D. Ruderman, "Origins of scaling in natural images," *Vision Research* **37**(23), pp. 3385–3395, 1997.
12. J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," *IEEE Conference on Computer Vision and Pattern Recognition, CVPR* **2**, pp. 126–133, 2000.