Available online at www.sciencedirect.com
ScienceDirect
Computer Vision and Image Understanding 104 (2006) 127–139
Computer Vision and Image Understanding
www.elsevier.com/locate/cviu

# Human body pose detection using Bayesian spatio-temporal templates ☆

M. Dimitrijevic *, V. Lepetit, P. Fua

*Computer Vision Laboratory, EPFL, Switzerland*

Received 31 January 2006; accepted 25 July 2006
Available online 14 September 2006

**Abstract**

We present a template-based approach to detecting human silhouettes in a specific walking pose. Our templates consist of short sequences of 2D silhouettes obtained from motion capture data. This lets us incorporate motion information into them and helps distinguish actual people who move in a predictable way from static objects whose outlines roughly resemble those of humans. Moreover, during the training phase we use statistical learning techniques to estimate and store the relevance of the different silhouette parts to the recognition task. At run-time, we use it to convert Chamfer distance to meaningful probability estimates. The templates can handle six different camera views, excluding the frontal and back view, as well as different scales. We demonstrate the effectiveness of our technique using both indoor and outdoor sequences of people walking in front of cluttered backgrounds and acquired with a moving camera, which makes techniques such as background subtraction impractical.


*Keywords:* Body pose detection; Spatio-temporal templates; Template matching; Chamfer matching

## 1. Introduction

Methods for recognizing 3-D human body poses in individual frames have become increasingly popular [1–6] because they are indispensable to achieve full automation in tracking. When occlusions are to be expected and background subtraction is not an option, for example because the camera is moving, Chamfer-based methods [7,8] are among the most robust ones.

However, as can be seen in Fig. 1b, Chamfer-based matching can easily produce many false positives and some false negatives, especially when the background is cluttered. In this work on detecting human silhouettes in specific walking pose, we achieve much lower error rates by introducing sophisticated templates that let us effectively take into account additional information.

- *Motion information.* We use motion capture data to create sequences of 2-D silhouettes that we match against short image sequences. The resulting *spatio-temporal templates* let us differentiate between actual people who move in a predictable way and static objects whose outlines roughly resemble those of humans, which are surprisingly numerous. Motion is well known to be important and optical flow methods have been proposed to exploit it [9]. However, accurately computing the flow on human limbs is notoriously difficult, especially if the background is not static. Replacing flow computation by matching against a set of moving silhouettes effectively addresses this issue.
- *Statistical relevance.* During a training phase, we use statistical learning techniques to estimate and store the relevance of the different silhouette parts to the recognition task. At run-time, we use it to convert Chamfer distances into meaningful probability estimates. For example, for walking motions, this accounts for the fact that feet and shoulders provide much more discriminant information than the trunk.

☆ This work was supported in part by the Swiss National Science Foundation and in part by the EU VisionTrain Project.
* Corresponding author.
*E-mail address:* miodrag.dimitrijevic@epfl.ch (M. Dimitrijevic).

1077-3142/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2006.07.007

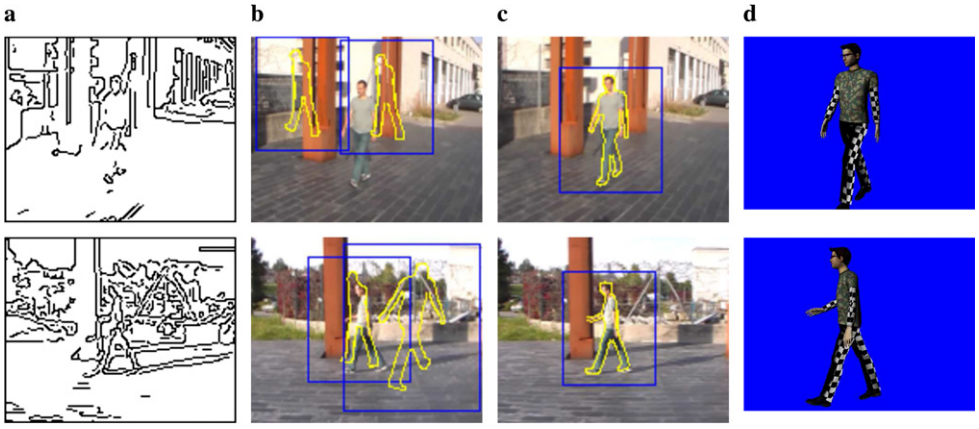


Fig. 1. Detection against a cluttered background. (a) Edge images corresponding to two images of a sequence acquired using a moving camera. (b) The two best matches obtained using single frame matching in each image, which are all wrong. (c) The best match using the spatio-temporal templates we advocate. (d) The corresponding 3D pose.

In this work, all the templates represent the specific part on the walking cycle where the feet are on the ground and the angle between the legs is greatest. We chose this pose because it is the one for which the temporal information is most informative. The angle between the legs first increases and then decreases. The templates cover six different camera views, excluding frontal and back view, as well as different scales.

As shown in Fig. 1c, we obtain much improved results even when the background is cluttered and background subtraction is impractical because the camera moves. Furthermore, because the detected templates are projections of 3-D models, we can map them back to full 3-D poses. Our method retains its effectiveness indoors, outdoors, and under difficult lighting conditions. It can also handle subjects who move closer or further so that their apparent scale changes and turn so that the angle from which they are seen also varies. In Fig. 2, we superpose ROC curves when using ordinary templates versus our spatio-temporal templates with or without taking statistical relevance into account. These graphs clearly show that including motion information and accounting for statistical relevance both contribute significantly to improving the level of performance.

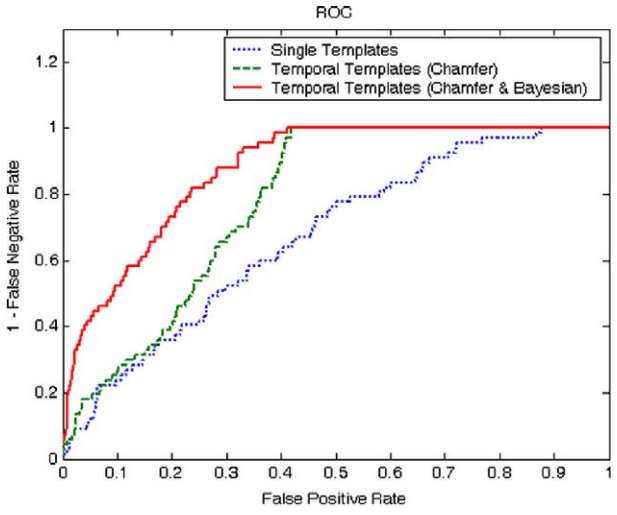


Fig. 2. ROC curves for three different techniques. Ordinary template matching (dotted curve), temporal template matching without (dashed curve) and with (full line curve) taking statistical relevance into account. The curves are computed for 2647 frames of the sequence acquired with a moving camera depicted by Fig. 1. The ordering of the curves from top to bottom quantifies the improvements that our technique brings about.

Note that, even though we chose a specific pose to test it, our approach is generic and could be applied to any other actions that all people perform in roughly similar ways but with substantial individual variations. For example, there also are characteristic postures for somebody sitting on a chair or climbing stairs. In the area of sports, we could use a small number of templates to represent the consecutive postures of a tennis player hitting the ball with a forehand, a backhand, or a serve, as is done in [5]. We could similarly handle the transition between the upswing and the downswing for a golfer. In short, characteristic postures are common in human motion and, therefore, worth finding. The only requirement for applying our method is that a representative motion database can be built.

## 2. Related work

Until recently, most approaches to capturing human 3-D motion from video relied on recursive frame-to-frame pose estimation. While effective in some cases, these techniques usually require manual initialization and re-initialization if the tracking fails. As a result, there is now increasing interest for techniques that can detect a 3-D body pose from individual frames of a monocular video sequence.

One approach [10–12] is to use classification-based techniques to decide whether or not image-windows contain a person. Such global approaches tend to be very occlusion sensitive and bag-of-features approaches have proved more effective at detecting pedestrians in crowded scenes [13]. They involve combining the local information provided by specific image patches with global clues provided by image silhouettes. The Chamfer distance is used for this

purpose but these techniques are not designed to handle viewpoint changes or recover 3-D pose and the best results are obtained for people seen from profile.

A different tack is to look for individual body parts and then to try assembling them to retrieve the pose [14,4,15,16]. This can be done by minimizing an appropriate criterion, for example using an A* algorithm. This has the potential to retrieve human bodies under arbitrary poses and in the presence of occlusions. Furthermore, it can be done in a computationally effective way using pictorial structures [17]. However, it can easily become confused because there are many limb-like objects in real world images.

Another class of approaches relies on techniques such as background subtraction to produce silhouettes that can then be analyzed. Several methods learn during an offline stage a mapping between the visual input space formed by the silhouettes and the 3-D pose space from examples collected manually or created using graphics software. For example, [18] uses multilayer perceptrons to map the silhouette represented by its moments to the 3-D pose. In [19] the mapping is performed using robust locally weighted regression over nearest neighbors that are efficiently retrieved using hash tables. In [2], it is done indirectly via manifolds embedded in low dimensional spaces, where each manifold corresponds to the subset of silhouettes for walking motion seen from a particular viewpoint. Local Linear Embedding is used to map the manifolds to both the silhouettes and the 3-D pose. In [1], the mapping between the couple formed by an extracted silhouette and a predicted pose to the corresponding 3-D pose is established using Relevant Vector Machines. More recently, it has been shown that combining a powerful statistical representation of the articulated body contours with silhouette and color information yields excellent performance [6]. While these works introduce powerful tools to associate 3-D poses to detected silhouettes, they tend to be of limited practical use because they require relatively clean silhouettes or foreground masks that are not always easy to obtain.

A more robust way to match global silhouettes against image contours is to use both a hierarchy of templates and the Chamfer distance, an approach originally introduced in [7] and extended in [20,3]. This produces excellent results when applied to difficult outdoor images. However, it seems to have a relatively high false detection rate. Reducing this rate involves either introducing *a priori* assumptions about where people can be [20] or incorporating additional processing such as texture classification or stereo verification [3]. In the context of hand tracking, [21] also relies on the Chamfer matching and a tree structure quite similar to the hierarchy of templates of [7] for efficiency. In this case, the false positives and negatives problem is avoided by assuming that one and only one hand is present in the image. Bayesian tracking is combined with detection to disambiguate the hand pose. [22] presents a method for estimating 3D hand pose that uses Chamfer distance and probabilistic line matching. It gives good results for cluttered images but, similarly to [21], requires a bounding box of roughly the right location and size placed around the hand.

Our method, which also relies on global silhouettes matching, includes an original way to take motion into account to avoid false positives. Such information was also exploited in [23] for human action recognition, but under the assumption that preprocessed and centered subimages of the people are available. In our case, we directly use the full images as input.

## 3. Approach

In this section, we first describe how we introduce motion information into the silhouette matching process, and then show how statistical relevance can be added. This is done on the sole basis of the noisy and potentially incomplete silhouettes that can realistically be extracted from images of cluttered scenes acquired by a moving camera.

### 3.1. Creating the templates

As shown in Fig. 3, we focus on the part of the walking cycle where both feet are on the ground and the angle between legs is greatest. We use motion capture data from several people walking at different speeds to create a database of templates such as the ones of Fig. 4. They consist of several silhouettes corresponding to consecutive time steps and are represented as sets of oriented pixels. The silhouettes are rendered from six different camera views and at seven different scales.

More specifically, we used a Vicon$^{tm}$ optical motion capture system and a treadmill to capture 8 people, 5 men and 3 women, walking at nine different speeds ranging from 3 to 7 km/h, by increments of 0.5 km/h. We also built a virtual character that can perform the captured motions, and rendered images at a rate of 25 frames per second as seen from the virtual cameras depicted by Fig. 4a. Note that Camera 3 (frontal view) and Camera 7 (back view) are not used, since these views give images of the model in which it is very difficult to distinguish the target pose from others. The rendered images are then used to create templates such as those depicted by Fig. 4b. The rendered images are rescaled at seven different scales ranging from $52 \times 64$ to $92 \times 113$ pixels, so that an image at one scale is 10% larger than the image one scale below. From each one of the rendered images, we extract the silhouette of the model. Each template is made of a short sequence of silhouettes that includes a key frame, that is the frame representing the specific walking pose and which is always taken to be the middle frame in the sequence. The silhouettes are represented as sets of oriented pixels that can be efficiently matched against image sequences, as will be discussed in Section 3.2.1.

In practice, we use 3 frame silhouette sequences. The top row of Fig. 4b corresponds to a profile view in which the $\varphi_i$

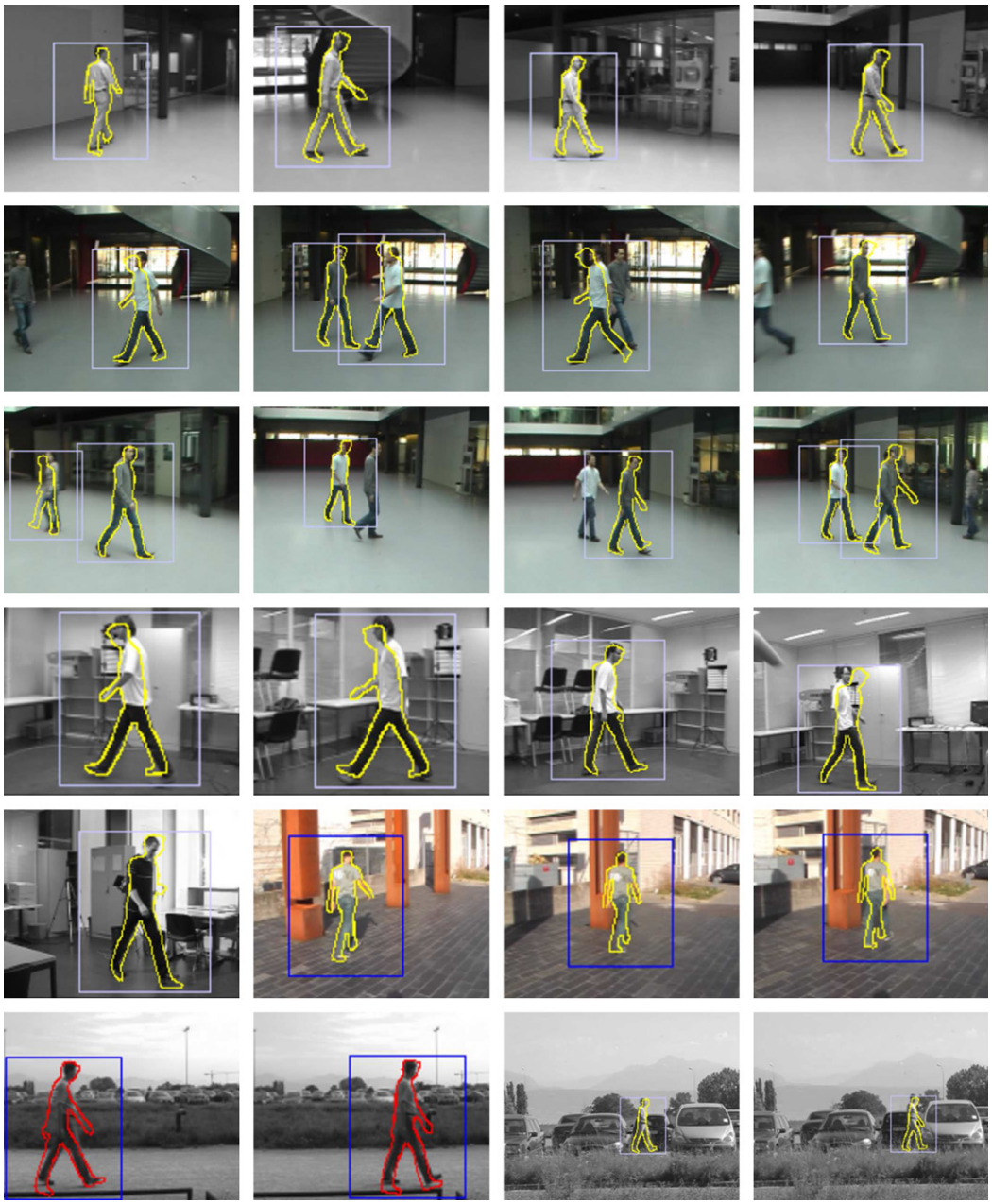

Fig. 3. Detected silhouettes in several indoor and outdoor sequences acquired by a moving camera. Since we search for a specific posture—the one where both legs are on the ground and the angle between them is greatest—the fact that the algorithm does not respond to some of the people in the second and third image of the third row is correct. In that sense, the detection on the left of the first image in the third row is one of the rare false positives it produces.

represent the angles between the two legs. Here, we have $\varphi_2 > \varphi_1$ and $\varphi_2 > \varphi_3$. The bottom row represents the same motion but seen from a different angle. To highlight the differences between the three silhouettes, we superpose the three profile ones in Fig. 4c.

### 3.2. Template matching

We first discuss robust template matching in individual images. We then outline our procedure for matching the whole spatio-temporal template to a short sequence.

#### 3.2.1. Single silhouette matching

We match the silhouettes to individual input images using Chamfer distance, which we compute using the Distance Transform (DT) of Canny edge images. We write the distance between $S$, a silhouette containing $n$ points, and $C$, a set of edge points detected in the image, as

$$d_{\text{chamfer}}(S, C) = \frac{1}{n} \sum_{s_i \in S} \delta(s_i, C), \tag{1}$$

where $\delta(s_i, C)$ can be taken to be the distance between $s_i$ and the closest point on $C$.

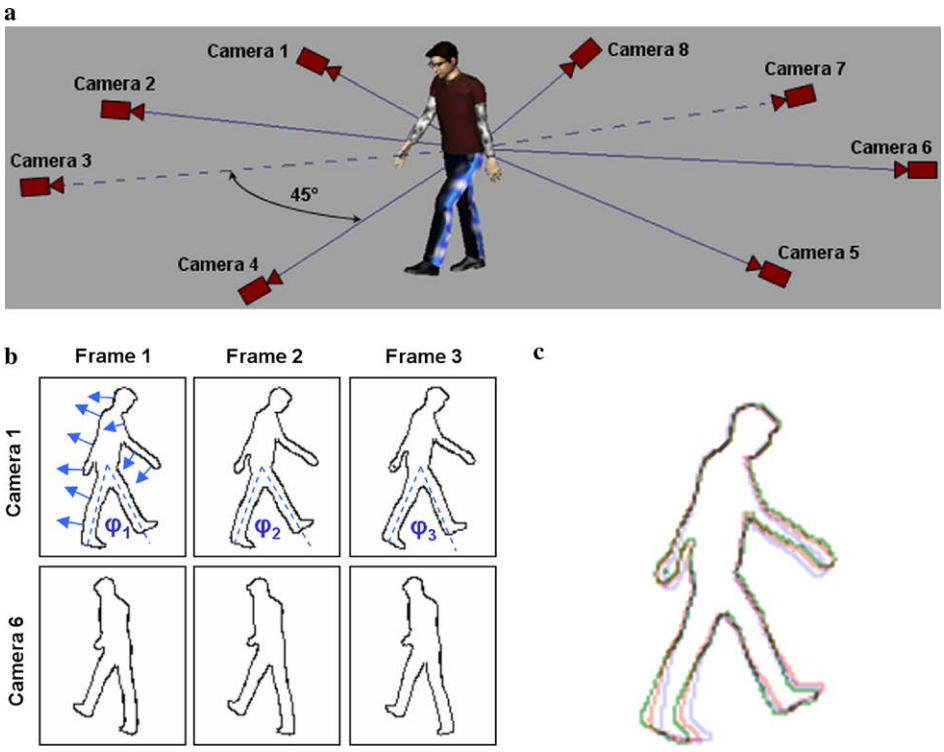


Fig. 4. Creating spatio-temporal templates. (a) Eight virtual cameras are placed around the model. (b) A template corresponding to a particular view consists of several silhouettes computed at three consecutive instants. The small blue arrows in image Camera 1/Frame 1 represent edge orientations used for matching silhouettes for some of the contour pixels. (c) The three silhouettes of a template are superposed to highlight the differences between outlines.

In the presence of clutter, defining $d_{\text{chamfer}}$ in this manner lacks robustness. To increase it, we take edge orientation into account by introducing a penalty term

$$h(s_i, c_j) = K(\tan(\alpha_{s_i} - \beta_{c_j}))^2,$$

where $\alpha_{s_i}$ and $\beta_{c_j}$ are the edge orientation, respectively, at the silhouette point $s_i$ and at the contour point $c_j$, and $K$ is a weight that defines the slope of the penalty function. The DT algorithm is modified so that each location in the DT image also contains the edge orientation of the closest edge pixel. In practice, we use $K = 20$, which is enough to completely eliminate the influence of the pixels that have the edge orientation difference greater than 30°.

Since our database contains templates at different scales, to allow effective comparison between Chamfer distances, we also introduce a scale factor $k$. Its role is to normalize the distance to the value that would be computed, had the template not been scaled.

We therefore take $\delta(s_i, C)$ in Eq. (1) to be

$$\delta(s_i, C) = \rho\left(\frac{1}{k}\|s_i - c(s_i)\| + h(s_i, c(s_i))\right), \tag{2}$$

where $c(s_i)$ is the closest contour point to point $s_i$, and $\rho(.)$ is the Tukey robust estimator [24] that further reduces the effect of outliers or missing edges.

### 3.2.2. Spatio-temporal template matching

Instead of single silhouette matching, we match our multi-silhouette templates against portions of the input sequence, as depicted by Fig. 5. Let $I_1, I_2, \ldots, I_t, \ldots, I_{t_{\max}}$ be an image sequence, where $t$ represents the discretized time and $t_{\max}$ the time at which the last frame was acquired. Each template $T = \{S_1, \ldots, S_i, \ldots, S_{N_S}\}$ includes a sequence of silhouettes $S_i$, where $i$ is the silhouette index and $N_S$ the number of silhouettes in the templates. In our experiments, we take $N_S = 3$ but it could be higher.

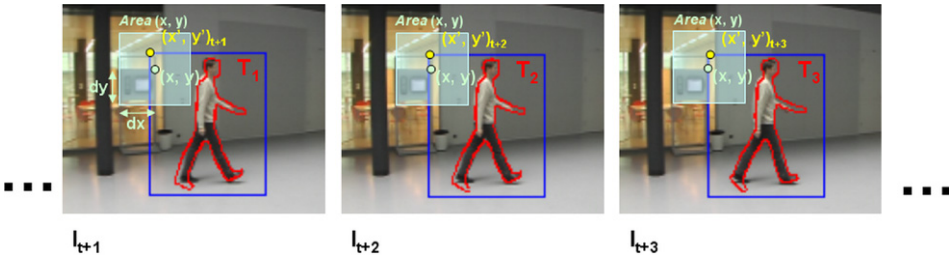


Fig. 5. A spatio-temporal template matched against consecutive images of the sequence.

For the sake of simplicity, the template scale is not explicitly represented in the following.

Let $I_t^{(x,y)}$ be the rectangular patch of image $I_t$ whose upper-left corner is positioned at $(x,y)$ and that is of the same size as the templates. $\mathrm{Area}(x,y)$ denotes an area centered on point $(x,y)$ such that $\forall(x',y') \in \mathrm{Area}(x,y)$, $x - \mathrm{d}x < x' < x + \mathrm{d}x$ and $y - \mathrm{d}y < y' < y + \mathrm{d}y$, where $\mathrm{d}x$ and $\mathrm{d}y$ are proportional to the template scale.

Using these notations, we take the distance $D$ between $N_S$ consecutive input images $I_{t+1} \ldots I_{t+N_S}$ and a template $T$ located at pixel $(x,y)$ to be

$$D(T,x,y,I_{t+1} \ldots I_{t+N_S}) = \frac{1}{N_S} \sum_{i=1}^{N_S} \mathrm{d}_i(x,y,T), \tag{3}$$

$$d_i(x,y,T) = \min_{(x',y') \in Area(x,y)} d_{\mathrm{chamfer}}(S_i, C(I_{t+i}^{(x',y')})), \tag{4}$$

where $C(I_{t+i}^{(x',y')})$ is the set of the edge points detected in the $I_{t+i}^{(x',y')}$.

This allows small variations on the locations of the successive silhouettes of a template. That is important because we make no assumptions about camera motion, which can be completely arbitrary. The templates can then be matched against the sequence by looking for local minima of the distance $D(.)$ when varying the template $T$, the image location $(x,y)$, and time $t$.

### 3.2.3. Avoiding multiple matches

An exhaustive search for the best matches over the whole sequence would produce several responses for the same person around the correct location and time. We avoid this as follows.

Let the match between an input sequence of $N_S$ frames and a template $T$ be the vector $\mathbf{m} = [T_\mathbf{m}, t_\mathbf{m}, x_\mathbf{m}, y_\mathbf{m}, D_\mathbf{m}]^T$. We build the sorted list $\mathscr{L}$ of $\mathbf{m}_i$ vectors sorted according to their distances $D_\mathbf{m}$ as follows. For each $t = 1 \ldots t_{\max} - N_S$ we find the best match $\mathbf{m}$ according to $D_\mathbf{m}$ and insert it in the sorted list $\mathscr{L}$. We repeat this parsing of the sequence until the distance $D_\mathbf{m}$ falls above a given threshold $\theta_D$ excluding the matches already present in $\mathscr{L}$. $\theta_D$ can be dynamically chosen as discussed below. This gives us a single match per person because a match $\mathbf{m}$ is inserted into the list $\mathscr{L}$ only if it does not overlap either in space or time another match $\mathbf{m}'$ already in $\mathscr{L}$ with a smaller distance. More formally $\mathbf{m}$ is inserted if there is no match $\mathbf{m}' \in \mathscr{L}$ such that

$$\begin{cases} D'_\mathbf{m} < D_\mathbf{m}, \\ (x_\mathbf{m}, y_\mathbf{m}) \in \mathrm{Area}(x_{\mathbf{m}'}, y_{\mathbf{m}'}) \text{ and}, \\ t_{\mathbf{m}'} - \delta t < t_\mathbf{m} < t_{\mathbf{m}'} + \delta t \end{cases} \tag{5}$$

where $\delta t$ is a constant that defines a frame range within which multiple detections in the same area are not allowed. Finally, we end up with the sorted list of matches $\mathscr{L}$ for the whole input sequence. Assuming the best match to be correct, it is possible to dynamically set the threshold to $\theta_D = K_D D_{\mathbf{m}_1}$ where $K_D$ is the same scalar value for all results shown in this paper.

### 3.2.4. Implementation issues

To avoid an exhaustive search for each silhouette in each frame, we implement our algorithm as follows. At each time step $t$, we search for the silhouettes $S_i$ of each template $T$ in the image $I_{t+i}$, $1 < i < N_S$. We also build a lookup table for a fast access to the silhouettes detected in an image around a given location. As before, to avoid multiple responses for the same person, we reject detections that overlap with better ones.

From these silhouette detections, we will build the list $\mathscr{L}_t$ of detected templates for which the silhouette sequence starts at time $t$. By fusing the successive lists $\mathscr{L}_t$ while respecting the conditions given in (5), we retrieve the final list $\mathscr{L}$ introduced in Section 3.2.3.

To build $\mathscr{L}_t$, we consider each silhouette $S_i$ detected in image $I_{t+i}$, where $i$ ranges between 1 and $N_S$, and check if the other silhouettes $S_j$ of the same template $T$ have been detected around the location of $S_i$ in the other images $I_{t+j}$. Using a lookup table lets us perform this search effectively. If all the successive silhouettes that compose spatio-temporal template are coherently detected, it is inserted into $\mathscr{L}_t$. The associated distance is simply the mean of the Chamfer distance of the successive silhouettes.

Since this involves matching all the silhouettes from the database against corresponding image region $I_t^{(x,y)}$, this could still be very slow as it might require $w_T \times h_T \times N_S \times N_T$ operations for Chamfer score computation, where $w_T$ is the silhouette width, $h_T$ is the silhouette height, $N_S$ is the number of silhouettes per template and $N_T$ is the number of templates. To decrease this complexity, we precompute a list of edge pixels that belong to at least one database silhouette. This list lets us reduce the number of accesses to the Chamfer map to less than $w_T \times h_T$ because only the pixels from the list are accessed. At the same time, the required number of operations is reduced by a factor $K \simeq 0.07$, which is the ratio of edge pixels to the template size. In practice, this roughly corresponds to a speed-up by a factor of 15.

All the techniques introduced above are exact and do not decrease the performance in any way. As a result, it takes a little under 0.06 s per spatio-temporal template per video frame on a 2.8 GHz PC. Since we use 432 such templates, it takes 25 s to process a frame. This is admittedly not particularly fast but adequate to demonstrate feasibility, which is our goal. Furthermore, since the current technique could be significantly speeded up by using a template hierarchy [7,20], we do not see any theoretical obstacle to ultimately incorporating it into a practical real world application.

## 3.3. Statistical relevance

Using templates as described above gives the same importance to all silhouette pixels. Intuitively, this is wrong because some body parts are clearly more informative than others. In our specific case, since we seek the pose where the legs are furthest apart, they

are clearly more important than the trunk whose silhouette is mostly made of two parallel vertical lines, which match many background shapes such as doors or windows. To solve this problem, we learn the relevance of different silhouette portions in a Bayesian framework. More specifically, we divide the templates into the patches depicted by Fig. 6a. We then learn the relevance of these patches using training sequences other than those we use to test our algorithm as follows.

Let $H_{I,T}$ denote the fact that a given image portion $I$ contains a human silhouette in a specific pose $T$. Maximizing $P(H_{I,T} = \text{yes}\,|\,I)$ is equivalent to minimizing the ratio of the likelihood terms $P(I\,|\,H_{I,T} = \text{no})$ and $P(I\,|\,H_{I,T} = \text{yes})$. We therefore replace the distance $d_{\text{chamfer}}$ in Eq. (4) by $d_{\text{bayes-chamfer}}$, the logarithm of this ratio:

$$d_{\text{bayes-chamfer}}(T,I) = \log\frac{P(I|H_{I,T} = \text{no})}{P(I|H_{I,T} = \text{yes})}.$$

We define these terms as

$$P(I|H_{I,T} = \text{yes}) = \left[\prod_{s\in S} P(\delta(s,C)|H_{I,T} = \text{yes})\right]^{\frac{1}{n}},$$

$$P(I|H_{I,T} = \text{no}) = \left[\prod_{s\in S} P(\delta(s,C)|H_{I,T} = \text{no})\right]^{\frac{1}{n}},$$

where $\delta(s,C)$ is the distance of Eq. (2), $C$ is the set of contour points extracted from $I$, and the distributions $P(\delta(s,C)|H_{I,T} = \text{yes})$ and $P(\delta(s,C)|H_{I,T} = \text{no})$ are learned from training examples. The $1/n$ power is for normalization over the number of the contour pixels $n$.

To handle the fact that not all the body parts have the same relevance, instead of learning one distribution for the whole image $I$, we divide $I$ into several patches and we learn different distributions, one for each patch, depending on which of the patches of Fig. 6 the point $s$ falls into. We therefore write

$$P(I|H_{I,T} = \text{yes}) = \left[\prod_{\mathbf{p}\in\mathbf{P}}\prod_{s\in S\cap\mathbf{p}} P_{\mathbf{p}}(\delta(s,C)|H_{I,T} = \text{yes})\right]^{\frac{1}{n}},$$

where $\mathbf{P}$ is the set of patches $\mathbf{p}$ depicted by Fig. 6, and $P_{\mathbf{p}}(\delta(s,C)|H_{I,T} = \text{yes})$ is the distribution learned for patch $\mathbf{p}$. We do the same to estimate $P(I|H_{I,T} = \text{no})$. To learn the $P_{\mathbf{p}}$ distributions, we ran the algorithm described in the previous subsection on several sequences of about 5000 frames. The detections, among which there are false positives, were manually classified into correct and incorrect ones. This training set is then used for representing the distributions as histograms, one for each patch and for each view. $d_{\text{bayes-chamfer}}(S,C)$ therefore becomes

$$d_{\text{bayes-chamfer}}(T,I) = \frac{1}{n}\sum_{\mathbf{p}\in\mathbf{P}}\sum_{s\in S\cap\mathbf{p}} \log\frac{\text{Hist}_{\mathbf{p},\text{no}}[\delta(s,C)]}{\text{Hist}_{\mathbf{p},\text{yes}}[\delta(s,C)]}, \qquad (6)$$

where $\text{Hist}_{\mathbf{p},\text{no}}$ and $\text{Hist}_{\mathbf{p},\text{yes}}$ are the histograms representing the corresponding distributions. Fig. 6b depicts the ratio between $\text{Hist}_{\mathbf{p},\text{no}}[\delta(s,C)]$ and $\text{Hist}_{\mathbf{p},\text{yes}}[\delta(s,C)]$ for a side camera view. Note that, not all patches have the same influence on $d_{\text{bayes-chamfer}}$. The patches that are far away from the contour have flat histograms, when none of the contour pixels of any of the templates belongs to them, or histograms that are highly non-uniform, when only few contour pixels of some of the templates belong to them. These patches are irrelevant to the final score since they contain a negligible number of pixels with respect to the overall number of pixels. Among the more central patches, those with monotonously increasing histograms are the most relevant ones: as expected, they mostly lie on the leg region. For these patches, image contours close to the template silhouette do not increase very much the final distance, while contours far from the silhouettes are strongly penalized. Using $d_{\text{bayes-chamfer}}$ instead of $d_{\text{chamfer}}$ therefore gives more importance to such relevant parts of the silhouettes, and thus yield better matching results as shown in Figs. 2, 7 and 12.

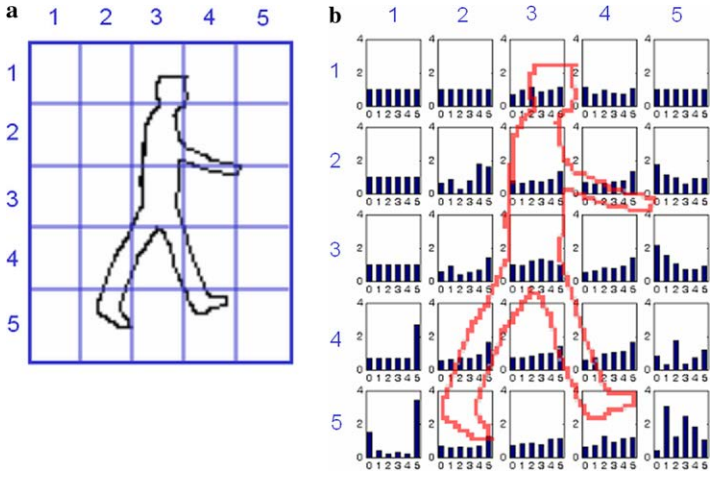


Fig. 6. (a) The templates are divided into 25 square patches $\mathbf{p}$. (b) The ratios $P_{\mathbf{p}}(\delta(s,C) = d|H_{I,T} = \text{no})/P_{\mathbf{p}}(\delta(s,C) = d|H_{I,T} = \text{yes})$ for each patch $\mathbf{p}$ and integer values for $d$. One of the template contours is superimposed in order to distinguish relevant patches.

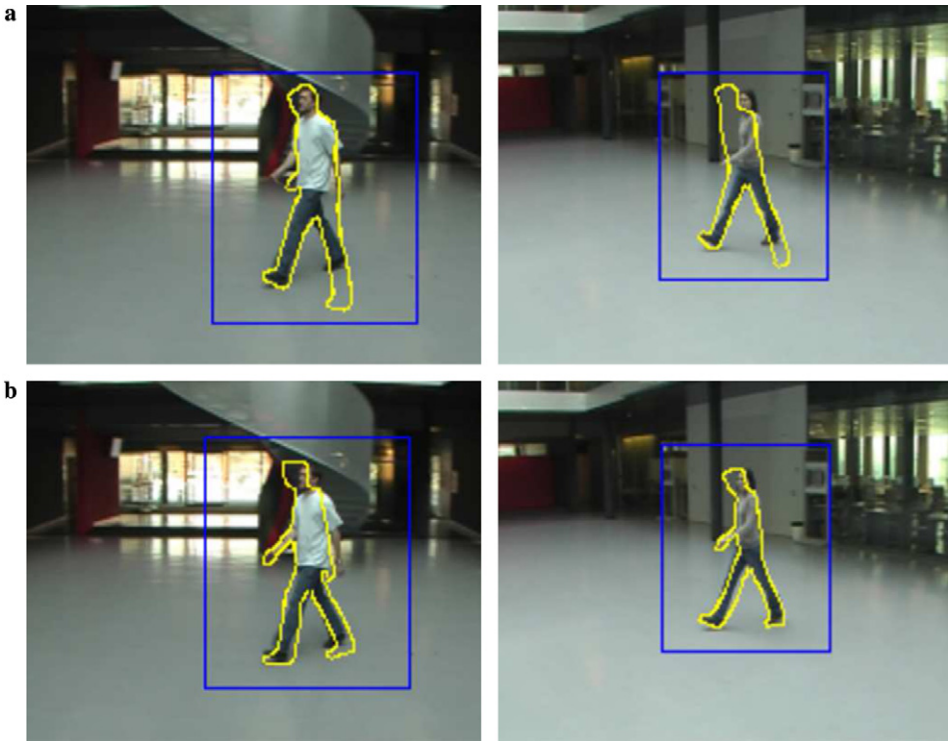


Fig. 7. Taking statistical relevance into account. (a) Best matches using the spatio-temporal templates and giving all silhouette pixels the same weight. (b) Best matches once statistical relevance is incorporated. Both scale and orientation are now correct.

## 4. Results

We have already shown in Fig. 3 some of the results obtained from several image sequences with cluttered backgrounds. Note that the people move closer or further so that their apparent scale changes. They also turn so that the angle from which they are seen varies as well.

### 4.1. Qualitative analysis

Our method is designed to detect people in a specific pose. As shown in the walking sequences of Fig. 8, that is exactly what it does. Note that the camera moves to follow the person.

In Fig. 9, we further demonstrate that the detections are correct even when the edge data is very poor. All the templates in our database are rendered from virtual cameras that are positioned at 1.20 m above ground level, so that optimal results can be expected when the real camera is at that height. However, our algorithm is very robust with respect to camera position. Fig. 10 shows that we can still recover full 3-D poses even when the camera is placed high above the person's head or very close to the ground level. The quantitative analysis is shown in Fig. 13. Our approach shows certain robustness to different clothing and shape appearance, as shown in Fig. 11, although the templates are created using one single virtual character.

False negatives occur only in frames where human silhouettes are so poor as to be unrecognizable, as in the case of Fig. 15a and b. When this happens, they can be accompanied by false positives because the Chamfer distance scores become unreliable and cannot be depended upon to eliminate dubious candidates. Another kind of failure depicted by Fig. 15c and d involves correctly detecting a person but with an inaccurate orientation or scale. The latter is slightly more frequent than the former but, as will be shown in the quantitative analysis below, both remain very rare.

### 4.2. Quantitative analysis

We provide quantitative performance-analysis in the form of the ROC curves of Figs. 2 and 12 , which correspond to the 2647-frame sequence of Fig. 1 and the 456-frame sequence depicted by the second and the third row of Fig. 3, respectively. Computing the required false-positive and false-negative error rates is not trivial because, for a 2647-frame sequence, the theoretical number of possible matches is larger than $15 \times 10^9$. We address this problem by first running our algorithm, retaining all matches that are within an extremely tolerant Chamfer distance threshold, and manually labeling these matches as good or bad. In our experience, this is a meaningful measure because the threshold can be easily chosen high-enough so as not to miss any of the true positives. Note that the labeling is done only for validation purposes and the test sequences are completely disjoint from the training ones.

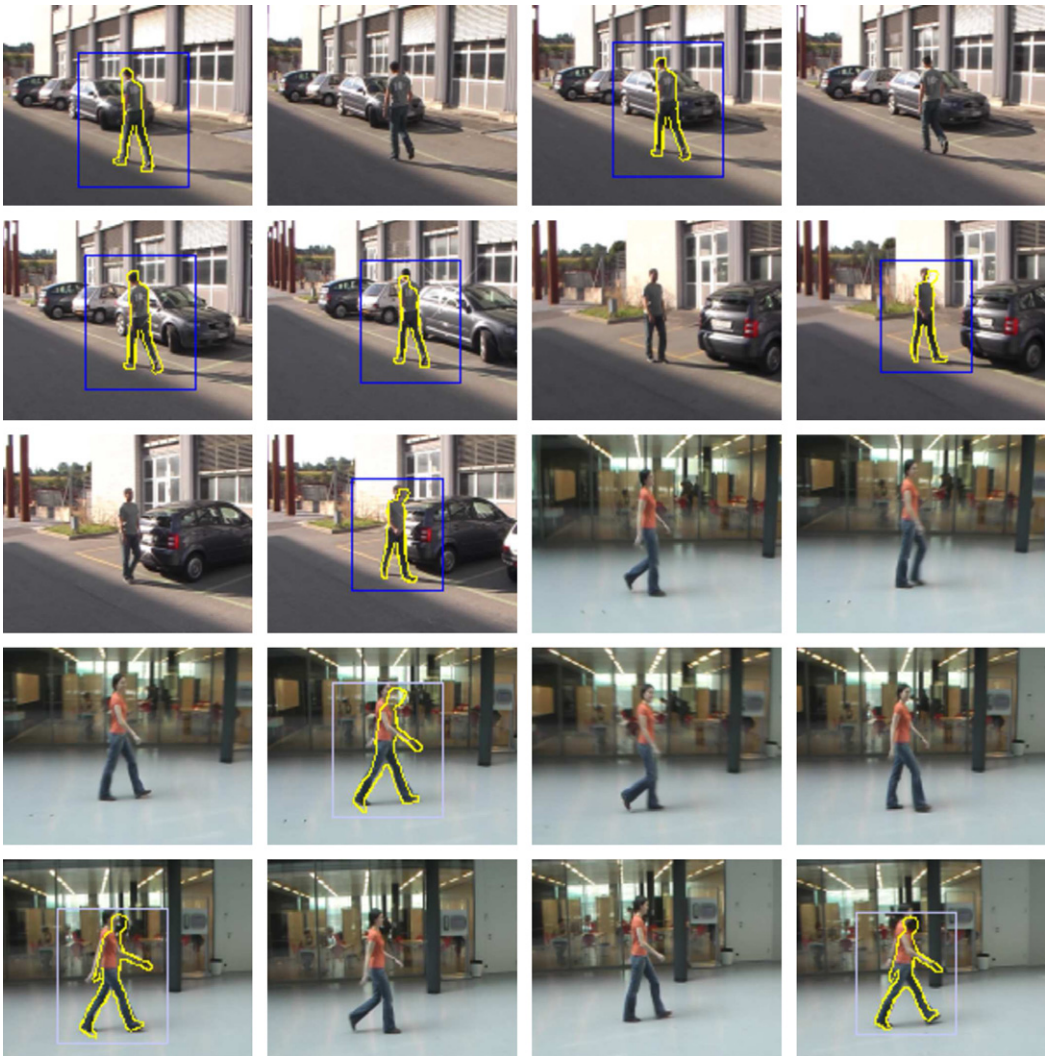

Fig. 8. Frames from two different sequences in which our algorithm finds only humans in the correct key pose. Note that the camera is moving to follow the person.

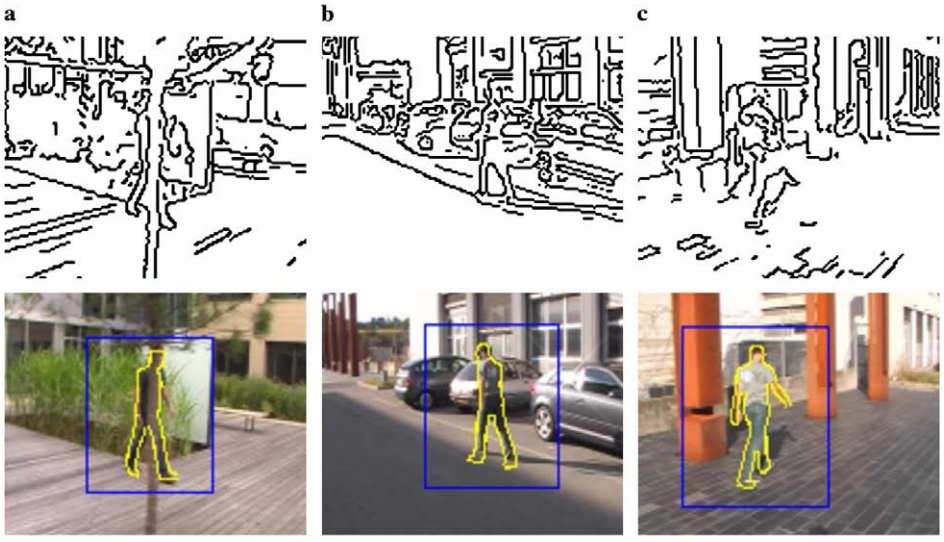


Fig. 9. Robustness to clutter and occlusions. We show the edge image in the first row and the corresponding detection in the second one. (a) Partial occlusion by a tree. (b) Cluttered background. (c) A substantial fraction of the silhouette edges are missing.

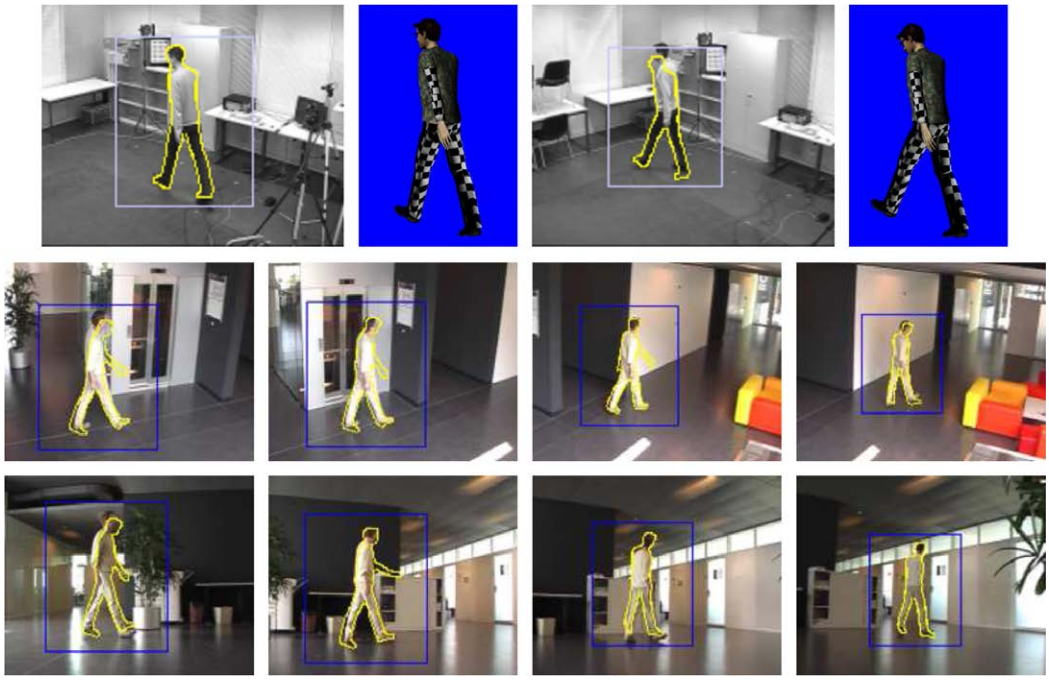

Fig. 10. Robustness to changes in camera height. First two rows: the 3-D pose is correctly recovered even though the actual camera is much further above ground level than the virtual cameras used to create the templates. Third row: correctly detected silhouettes when the camera is placed only 30 cm above the ground level.

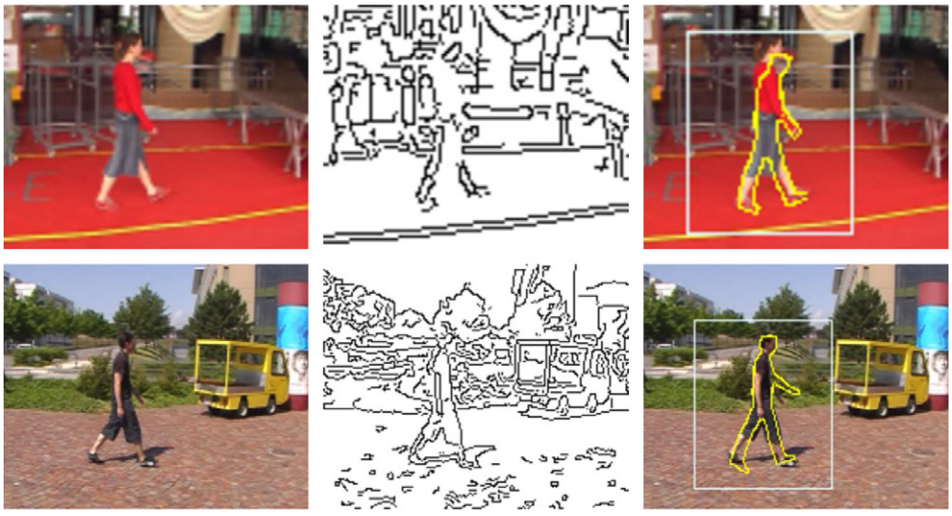

Fig. 11. Robustness to changes in clothing. First row: the subject is detected even though she wears a shirt. Second row: correctly detected person wearing knee-length pants.

We take the *False Positive Rate* to be the ratio between the number of *false positives* and overall number of *bad* matches. Similarly, $1 - false\ negative\ rate$ is computed as the ratio between the number of *true positives* and the overall number of *good* matches. For both sequences, in Figs. 2 and 12, we show three ROC curves computed by matching against ordinary templates, spatio-temporal templates without taking statistical relevance into account, and the same spatio-temporal templates with statistical relevance taken into consideration. Since the three curves are clearly above one another, this demonstrates the clear improvement that using both statistical relevance and motion information brings about.

### 4.3. Number of silhouettes

Motion information is incorporated into the templates by using several silhouettes from consecutive frames. The specific pose in which the legs are furthest apart is the one for which the temporal information is most informative because the angle between them first increases and then decreases. However, to detect this at least three silhouettes

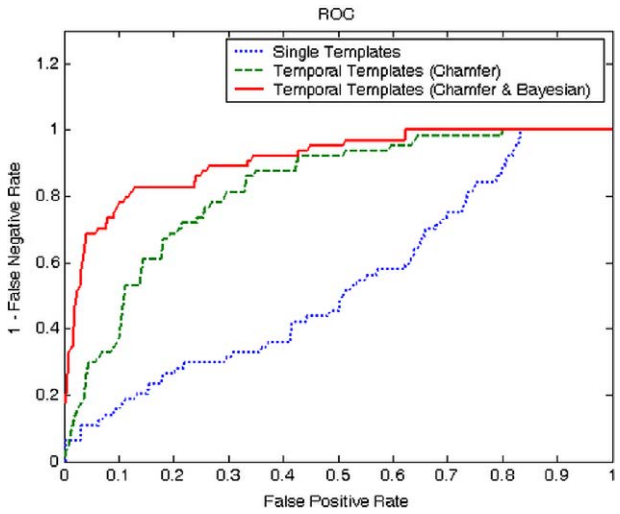


Fig. 12. ROC curves for the 456-frame sequence of the second and the third row of Fig. 3. The bottom curve depicts the results obtained using simple templates. Using the spatio-temporal templates yields a very marked improvement as the corresponding curve is much higher. The results are further improved by incorporating statistical relevance, which yields the topmost curve.

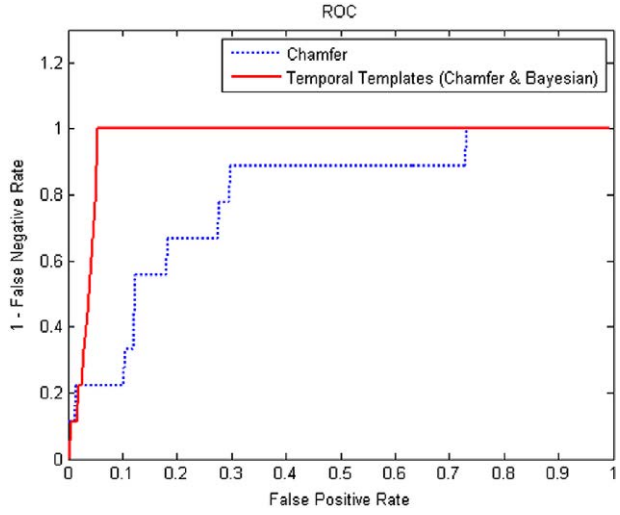


Fig. 13. ROC curves for the 260-frame sequence of the second row of Fig. 10 where the actual camera is placed much further above ground level. The bottom curve depicts the results obtained using simple templates while the top one corresponds to the use of bayesian spatio-temporal templates.

are needed, which is therefore the minimum number we have used in our experiments. Quantitative analysis is presented by ROC curves (Fig. 14) that are created on the image sequence depicted by the second and the third row of Fig. 3. While the use of three silhouettes per template brigs considerable performance improvement, further increasing the number of silhouettes does not significantly improve the results. Moreover, the use of five instead of three silhouettes increases the computation time for about 30%. Therefore, we conclude that the use of three silhouette templates is the most reasonable.

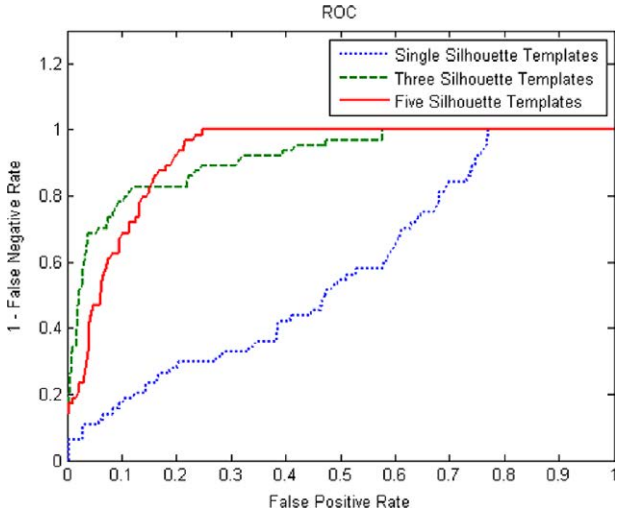


Fig. 14. ROC curves for different number of template silhouettes: ordinary template matching (dotted curve), bayesian temporal template matching with the templates consisting of three silhouettes (dashed curve) and five silhouettes (full line curve). The curves are computed for the 456-frame sequence of the second and the third row of Fig. 3.

### 4.4. Summary

Our method detects people in the target posture with a very low error rate. The few false positives mostly correspond to people but at somewhat inaccurate scales or orientations. While this paper focuses on pure detection, it is therefore clear that the performance of our algorithm could be further increased by linking the various detections to create plausible trajectories, using an algorithm such as dynamic programming.

## 5. Conclusion

We have presented a method for human body pose detection based on template matching. The templates are rendered from a virtual character that performs real human motions captured by a motion capture system. They can be used for detecting a characteristic human body pose at different scales and six different camera views, excluding the frontal and back view.

Our method combines silhouette matching, motion information, and statistical relevance estimates in an original way. This is important because human motion is very different from other kinds of motions and can be effectively used to reduce the false positive and negative detection rate. As a result, even though we do not use color or texture, we have been able to demonstrate very good results for indoor and outdoor sequences for which background subtraction is impossible, under difficult lighting conditions, different camera viewpoints and apparent scale changes. Obviously, in future work, these additional clues should also be taken into account to increase performance. Furthermore, since the detected templates are projections

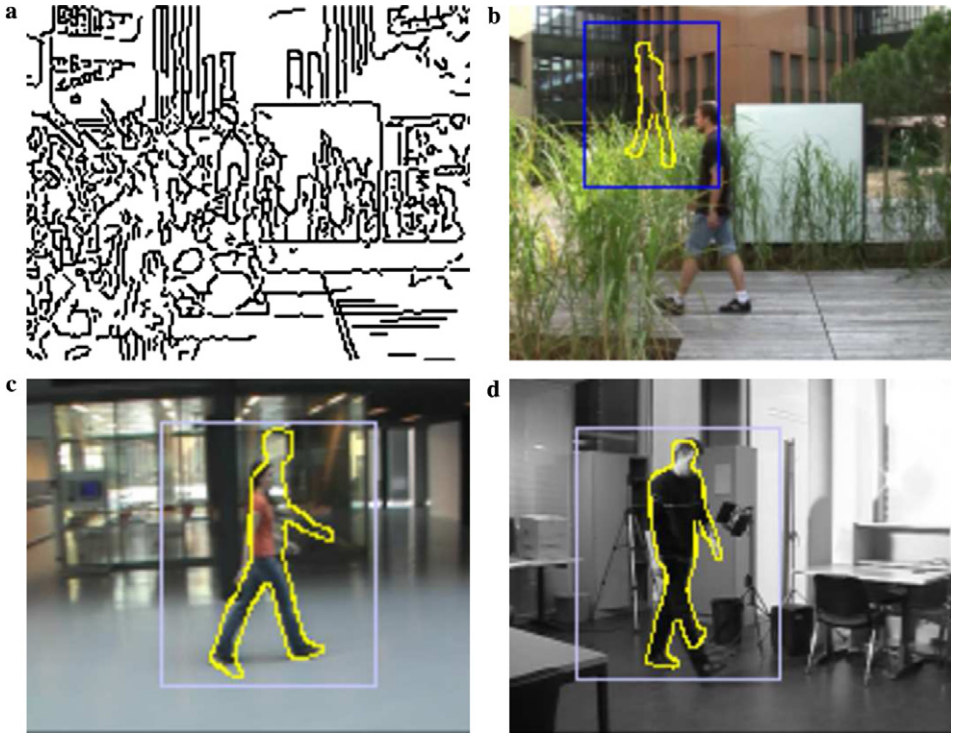


Fig. 15. Failure modes. (a and b) The subject is partially hidden by the foreground bush, which completely prevents recovery of edges on the legs. This results in both a false negative and a false positive. (c) Wrong scale. (d) Wrong orientation.

of 3-D models, mapping them back from 2-D to full 3-D poses is straightforward.

Our approach, even though tested on a specific human pose, is generic and could be applied for any other actions that all people perform in roughly similar ways but with substantial individual variations. The only requirement is that a representative motion database can be built.

This method, with its accurate 3-D pose detections, is a key step towards robust full 3-D body pose tracking algorithms that can initialize and re-initialize themselves in difficult real world conditions where techniques such as background subtraction are impractical.

## Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.cviu.2006.07.007.

## References

[1] A. Agarwal, B. Triggs, 3d human pose from silhouettes by relevance vector regression, in: Conference on Computer Vision and Pattern Recognition, 2004.
[2] A. Elgammal, C. Lee, Inferring 3D body pose from silhouettes using activity manifold learning, in: CVPR, Washington, DC, 2004.
[3] J. Giebel, D. Gavrila, C. Schnorr, A bayesian framework for multi-cue 3d object tracking, in: Proceedings of European Conference on Computer Vision, 2004.
[4] G. Mori, X. Ren, A. Efros, J. Malik, Recovering human body configurations: combining segmentation and recognition, in: Conference on Computer Vision and Pattern Recognition, Washington, DC, 2004.
[5] J. Sullivan, S. Carlsson, Recognizing and tracking human action, in: European Conference on Computer Vision, 2002.
[6] J. Zhang, R. Collins, Y. Liu, Bayesian body localization using mixture of nonlinear shape models, in: International Conference on Computer Vision, Beijing, China, 2005.
[7] C.F. Olson, D.P. Huttenlocher, Automatic target recognition by matching oriented edge pixels, IEEE Transactions on Image Processing 6 (1997) 103–113.
[8] D. Gavrila, J. Giebel, S. Munder, Vision-based pedestrian detection: the protector system, in: Intelligent Vehicles Symposium, 2004, pp. 13–18.
[9] R. Fablet, M. Black, Automatic detection and tracking of human motion with a view-based representation, in: European Conference on Computer Vision, 2002.
[10] P. Viola, M. Jones, D. Snow, Detecting pedestrians using patterns of motion and appearance, in: International Conference on Computer Vision, 2003, pp. 734–741.
[11] K. Okuma, A. Taleghani, N. de Freitas, J. Little, D. Lowe, A Boosted Particle Filter: Multitarget Detection and Tracking, in: ECCV, Prague, Czech Republic, 2004.
[12] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Conference on Computer Vision and Pattern Recognition, 2005.
[13] B. Leibe, E. Seemann, B. Schiele, Pedestrian detection in crowded scenes, in: Conference on Computer Vision and Pattern Recognition, vol. 1, San Diego, CA, 2005.
[14] R. Ronfard, C. Schmid, B. Triggs, Learning to parse pictures of people, in: European Conference on Computer Vision, vol. 4, Copenhagen, Denmark, 2002, pp. 700–714.
[15] K. Mikolajczyk, C. Schmid, A. Zisserman, Human detection based on a probabilistic assembly of robust part detectors, in: European Conference on Computer Vision, vol. I, 2004, pp. 69–81.
[16] T.J. Roberts, S.J. McKenna, I.W. Ricketts, Human pose estimation using learnt probabilistic region similarities and partial configura-

tionsEuropean Conference on Computer Vision, vol. 4, Prague, Czech Republic, 2004, pp. 291–304.
[17] P. Felzenszwalb, D. Huttenlocher, Pictorial structures for object recognition, International Journal of Computer Vision 61 (1) (2005).
[18] R. Rosales, S. Sclaroff, Infering body pose without tracking body parts, in: Conference on Computer Vision and Pattern Recognition, 2000.
[19] G. Shakhnarovich, P. Viola, T. Darrell, Fast pose estimation with parameter-sensitive hashing, in: International Conference on Computer Vision, Nice, France, 2003.
[20] D. Gavrila, V. Philomin, Real-time object detection for "smart" vehicles, in: International Conference on Computer Vision, 1999, pp. 87–93.
[21] B. Stenger, A. Thayananthan, P. Torr, R. Cipolla, Filtering using a tree-based estimator, in: International Conference on Computer Vision, vol. 2, 2003, pp. 1063–1070.
[22] V. Athitsos, S. Sclaroff, Estimating 3d hand pose from a cluttered image, in: Conference on Computer Vision and Pattern Recognition, Madison, WI, 2003, pp. 432–439.
[23] A. Efros, A. Berg, G. Mori, J. Malik, Recognizing action at a distance, in: International Conference on Computer Vision, 2003, pp. 726–733.
[24] P. Huber, Robust Statistics, Wiley, New York, 1981.