Human Body Pose Recognition Using Spatio-Temporal Templates*

M. Dimitrijevic, V. Lepetit and P. Fua
Computer Vision Laboratory, EPFL, Switzerland

Abstract

We present a novel approach to detecting human silhouettes in monocular sequences that achieves very low rates of both false positives and negatives by combining shape and motion information. To this end, we use sequences of moving silhouettes built using motion capture data that we match against short image sequences.

We demonstrate the effectiveness of our technique using both indoor and outdoor images of people walking in front of cluttered backgrounds and acquired with a moving camera, which makes techniques such as background subtraction impractical.

1. Introduction

Approaches to recognizing 3–D human body postures from a single image have recently become increasingly popular [1, 3, 8, 11, 18]. While they do not suffer from many of the problems that affect more traditional recursive body tracking techniques, most of them have only been demonstrated in cases where clean body silhouettes can be extracted, for example using background subtraction, which is very restrictive. A key exception is the work reported in [6]. Combining a hierarchy of templates [13] and effectively using the chamfer distance has made the approach applicable to more challenging cases such as the one of a moving camera on a car. However, even then, the algorithm tends to produce many false positives, especially when the background is cluttered. As a result, in practice, it is used in conjunction with a stereo rig both to narrow the initial search area and to filter out false detections from the background [7, 8].

We improve upon this approach and achieve very low rates of both false positives and negatives by incorporating motion information into our templates. It lets us differentiate between actual people and static objects whose outlines roughly resemble those of a human, which are surprisingly numerous. As illustrated by Fig. 1, this is key to avoiding misdetections. This is of course a well known fact and optical flow methods have been proposed to detect moving humans [4]. However, accurately computing the flow on human limbs is notoriously difficult, especially if the background is not static. Our approach avoids this problem by relying on sequences of moving silhouettes.

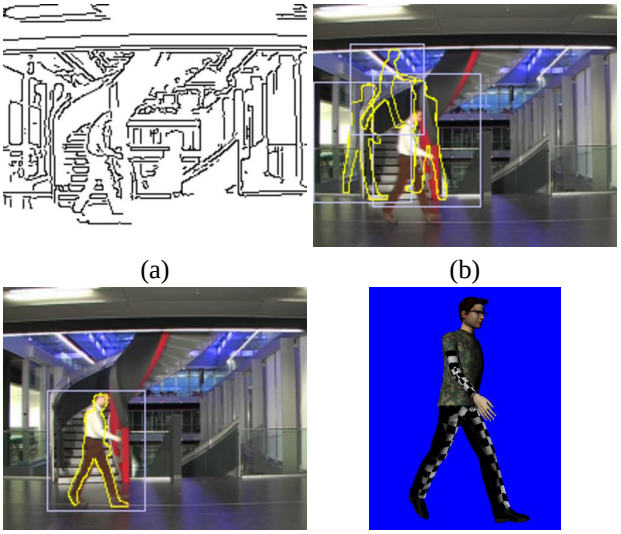


Figure 1: Detection against a cluttered background. (a) An edge image corresponding to one of the images of a sequence. (b) The first three best template matches obtained using single frame matching, which are wrong. (c) The best template match using the spatio-temporal templates we advocate. (d) The corresponding 3D pose.

More specifically, we focus on the part of the walking cycle where both feet are on the ground and use motion capture data to create sequences of 2–D silhouettes that we match against short image sequences. We chose this specific posture both because it is very characteristic and because it could easily be used to initialize a more traditional recursive tracking algorithm to recover the in-between body poses.

As shown in Fig. 2, we obtain good results even when the background is cluttered and background subtraction is impractical because the camera moves. Note that the subjects move closer or further so that their apparent scale changes and turn so that the angle from which they are seen also varies. In this example, no stereo data or information about the ground plane was required to eliminate false-positives. Our method retains its effectiveness indoors, outdoors, and

*This work was supported in part by the Swiss National Science Foundation.

Figure 2: 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. The sequence with several people is attached as a supplementary material.

under difficult lighting conditions. Furthermore, because the detected templates are projections of 3–D models, we can map them back to full 3–D poses.

Note that, even though we chose a specific motion 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 [18]. 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.

In the reminder of the paper we first briefly discuss earlier approaches. We then introduce our approach to body pose detection and present a number of results obtained in challenging conditions. Finally, we discuss possible extensions.

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 [19, 12] is to use classification to detect people in images, but it does not provide either a pose or a precise outline. Furthermore, such global approaches tend to be very occlusion sensitive.

Instead of detecting the body as a whole, a different tack is to look for individual body parts and then to try assembling them to retrieve the pose [14, 11, 10]. 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 [5]. 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, [15] uses multilayer perceptrons to map the silhouette represented by its moments to the 3–D pose. In [16] the mapping is performed using robust locally weighted regression over nearest neighbors that are efficiently retrieved using hash tables. In [3], 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 Machine. 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 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 [13] and extended in [7, 8]. 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 [7] or incorporating additional processing such as texture classification or stereo verification [8]. In the context of hand tracking, [17] also relies on the chamfer distance and a tree structure quite similar to the hierarchy of templates of [13] 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.

By contrast to these earlier approaches, 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 [2] for human action recognition, but only 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

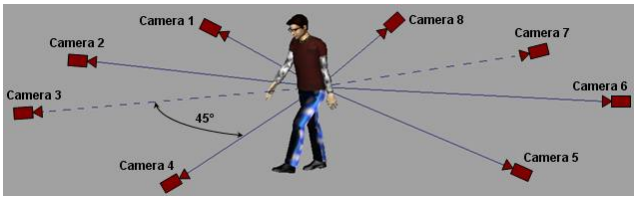
In this section, we describe how we introduce motion information into the silhouette matching process. This is done on the sole basis of the noisy and potentially incomplete silhouettes that can realistically be extracted from images of cluttered scenes acquired by a moving camera.

3.1 Creating the Templates

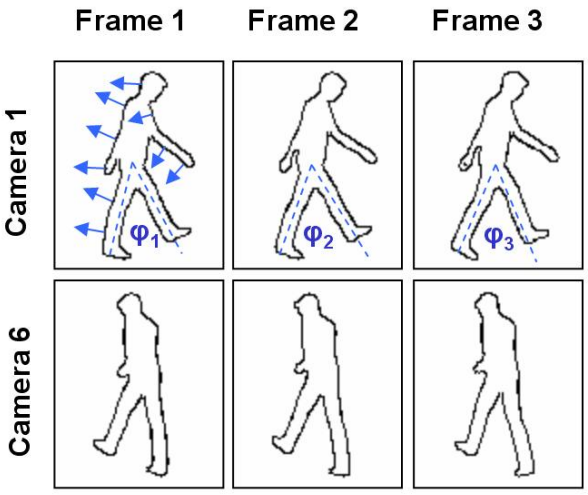
Here, we focus on the part of the walking cycle where both feet and legs are on the ground and the angle between them are the greatest, and use motion capture data and graphics software to create a database of templates.

We first 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 km/h to 7 km/h, by increments of 0.5 km/h. Also, we 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. 3(a). 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 searched pose from others. The rendered images are then used to create templates such as those depicted by Fig. 3(b). The rendered images are rescaled at seven different scales ranging from 52×64 to 92×113 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.

In practice we use 3 frame silhouette sequences. The top row of Fig. 3(b) corresponds to a profile view in which the φ_i represent the angles between the two legs. Here, we



(a)



(b)

(c)

Figure 3: Creating spatio-temporal templates: (a) Eight virtual cameras are placed around the model at every $45°$. (b) Each template corresponding to a single camera view consists of three silhouettes, extracted from three consecutive frames. 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 the outlines.

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. 3(c).

3.2 Single Silhouette Matching

As in previous approaches [13, 7], we rely on Chamfer distance, efficiently computed using the Distance Transform (DT) of the input image, to match silhouettes to individual input images. However, we have endeavored to increase its robustness.

The original formulation of the Chamfer distance is

$$d_{chamfer}(S, C) = \frac{1}{n} \sum_{s_i \in S} \min_{c_j \in C} \|s_i - c_j\| \qquad (1)$$

where S is the silhouette containing n points, and C is the set of contour points in the input image after Canny edge detection. Simply relying on the distance between edge produces a lot of false positives, especially in presence of clutter. We therefore also take into account the edge orientation by introducing a penalty term

$$p(s_i, c_j) = K * [\tan(\alpha_{s_i} - \beta_{c_j})]^2, \qquad (2)$$

where α_{s_i} and β_{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 algorithm for DT computation 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 $30°$, even if the distance between them is zero.

As discussed above, our template database contains different scale templates. To allow effective comparison between the chamfer distances for such templates, we explicitly introduce a scale factor k into Equation 1 to normalize the distance to the value that would be computed if the template has not been scaled. Finally we introduce the Tukey robust estimator [9] to reduce the effect of outliers or missing edges. We therefore take Chamfer distance to be

$$d_{chamfer}(S, C) = \frac{1}{n} \sum_{s_i \in S} \rho \left(\frac{1}{k} \|s_i - c(s_i)\| + p(s_i, c(s_i)) \right) \qquad (3)$$

where $c(s_i)$ is the closest contour point to point s_i.

3.3 Spatio-Temporal Template Matching

Instead of single silhouette matching, we match our templates made of several silhouettes against portions of the sequence. Let the input image sequence be the set of images $I_1, I_2, \ldots, I_t, \ldots, I_{t_{max}}$, where t represents the discretized time and t_{max} the time at which the last frame was acquired. Each template T created as explained in Section 3.1 is made of a sequence of silhouettes: $T = \{T_1, \ldots, T_i, \ldots, T_{N_S}\}$, where i is the index of the silhouette in the sequence and

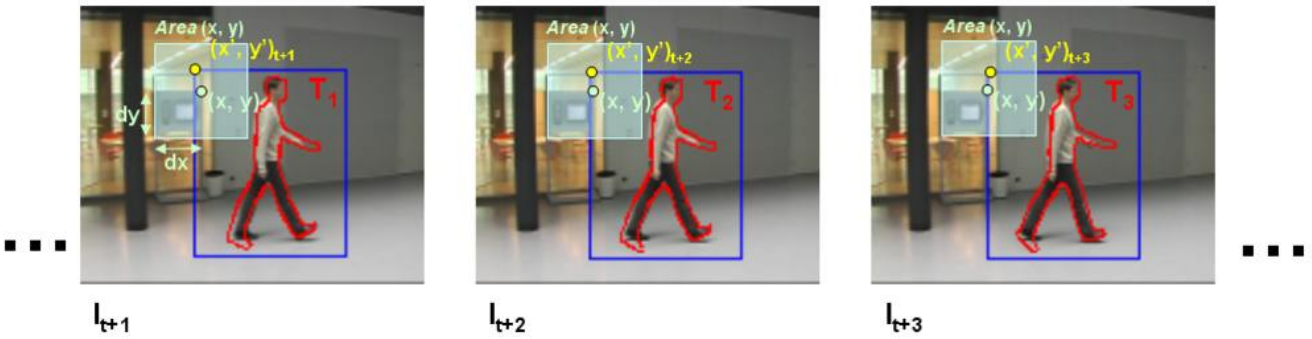


Figure 4: A spatio-temporal template matched against consecutive images of the sequence.

N_S the number of silhouettes in the templates. In our experiments, we took $N_S = 3$ but it could be higher. An example of template matched against a portion of an image sequence is presented Fig. 4. Note that, 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 of the templates. $Area(x, y)$ denotes an area centered on point (x, y) where $x - dx < x < x + dx$ and $y - dy < y < y + dy$, where dx and dy are proportional to the template scale.

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

$$D(T, x, y, I_{t+1} \dots I_{t+N_S}) = \frac{1}{N_S} \sum_{i=1}^{N_S} d_i(x, y, T), \qquad (4)$$

$$d_i(x, y, T) = \min_{\substack{(x',y') \in \\ Area(x,y)}} d_{chamfer}\left(T_i, I_{t+i}^{(x',y')}\right). \qquad (5)$$

This allows small variations on the locations of the successive silhouettes of a template. The templates can then be matched against the sequence by looking for local minima of the distance $D(.)$ when varying the template T, the location (x, y) over the images and t over time.

However, if we directly search for the best matches in a exhaustive way, we would get several responses for the same person around the correct location and time. To avoid that, we rely on the following strategy. 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 $\mathcal{L}$ of $\mathbf{m}_i$ vectors sorted according to their distances $D_\mathbf{m}$ as follows. For each $t = 1 \dots t_{max} - N_S$ we find the best match $\mathbf{m}$ according to $D_\mathbf{m}$ and insert it in the sorted list $\mathcal{L}$. We repeat this parsing of the sequence until the distance $D_\mathbf{m}$ falls above a given threshold θ_D excluding the matches already present in $\mathcal{L}$. θ_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 $\mathcal{L}$ only if it does not overlap either in space or time another match $\mathbf{m}'$ already in $\mathcal{L}$ with a smaller distance. More formally $\mathbf{m}$ is inserted if there is no match $\mathbf{m}' \in \mathcal{L}$ such as:

$$\begin{cases} D'_\mathbf{m} < D_\mathbf{m}, \\ (x_\mathbf{m}, y_\mathbf{m}) \in Area(x_{\mathbf{m}'}, y_{\mathbf{m}'}) \text{ and} \\ t_{\mathbf{m}'} - \delta t < t_\mathbf{m} < t_{\mathbf{m}'} + \delta t \end{cases}, \qquad (6)$$

where δt 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 $\mathcal{L}$ 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.

3.4 Implementation Details

In practice, a naive implementation of this method would be computationally very expensive. Therefore we propose an alternative way of finding the best matches. For each time step t, we search for the silhouettes T_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 $\mathcal{L}_t$ of detected templates for which the silhouette sequence starts at time t. By fusing the successive lists $\mathcal{L}_t$ while respecting the conditions given in (6), we retrieve the final list $\mathcal{L}$ described above.

Each list $\mathcal{L}_t$ is constructed as follows. For each silhouette T_i detected in image I_{t+i}, where i varies between 1 and N_S, we check if the other silhouettes T_j of the same template T have been detected around the location of T_i in the other images I_{t+j}. This search is performed efficiently using the lookup table. If all the successive silhouettes for a same template have been coherently detected, they are inserted as

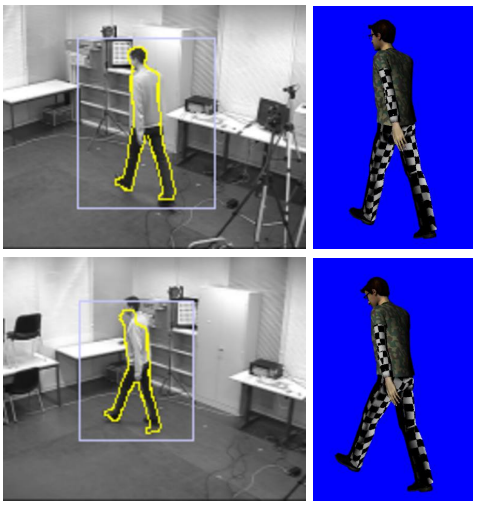


Figure 5: Our approach is robust to changes in the camera position. Here the camera is placed very high, and a satisfying pose is recovered even though there are no templates in the database for such camera view.

a single template in the list $\mathcal{L}_t$. The associated distance is simply the mean of the Chamfer distance of the successive silhouettes.

The silhouette detection involves matching all the silhouettes from the database against corresponding image region $I_t^{(x,y)}$. A naive implementation would be computationally very expensive as it would 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.

As a result, it takes a little under 0.06 seconds per spatio-temporal template per video frame on a 2.8 GHz PC. Since we use 432 such templates, it takes 25 seconds 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 Gavrila like template hierarchy, we do not see any theoretical obstacle to ultimately incorporating it into a practical real world application.

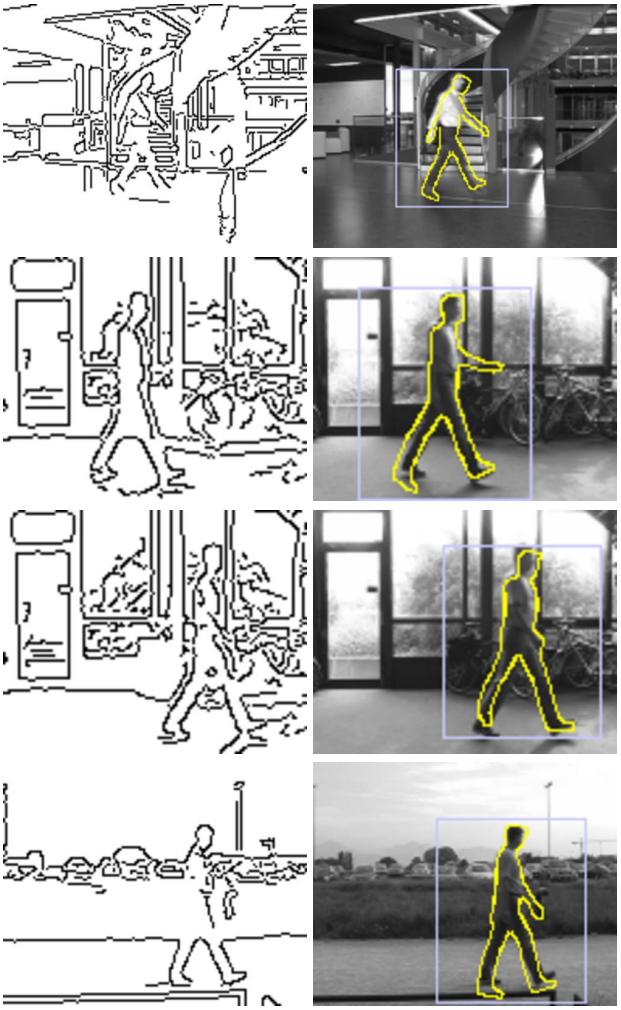


Figure 6: Top rows: Correct detections in spite of the cluttered background. Bottom row: Correct detection even though a substantial part of the silhouette is missing.

4 Results

We have already shown in Fig. 2 some of the results obtained from several image sequences with cluttered background. Note that the subjects move closer or further so that their apparent scale changes and turn so that the angle from which they are seen also varies. All the templates in our database are rendered from virtual cameras that are positioned at 1.20m from the ground level, so that optimal results can be expected when the camera is at that height. However, our algorithm is very robust with respect to camera position. Fig. 5 shows its good behaviour even when the camera is placed high above the head of the person.

In Fig. 6 we further demonstrate that the detections are correct even when the edge images are very noisy. Further-

Figure 7: 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. The 3–D pose corresponding to the right-most images of the first two rows are shown. The second sequence is supplied as a supplementary material.

more, we can map the detected templates back to full 3–D poses as shown in Fig. 5. Remember that our method is designed to detect people in a specific pose. As shown in the walking sequences of Fig. 7, that is exactly what it does. Note that the camera moves to follow the person.

Typical failure modes involve a detection location that is usually correct but an inaccurate orientation or scale, as shown in Fig. 8. To quantify this, we estimated the error rate on the two movies supplied with the paper as supplementary material. The first movie is depicted by the last three rows of Fig. 7. In the 590 frames we got 42 positives, 2 false negatives and no false positives. Among the positives there are 4 detections for which the scale is more than 10% incorrect, such as the one shown in Fig. 8(a). In the second movie, depicted by the second and third row of Fig. 2, there are multiple people. In this 445 frames sequence, our algorithm finds 37 positives, 6 false negatives and 2 false positives. Note that the two false positives do correspond to people but not in the searched posture, as shown in the first image of the third row of Fig. 2. There are no false positives in the background. Among the positives there are 5 detections for which the scale is more than 10% uncorrect, and 2 detections for which the orientation error is more than 45°, such as the one shown in Fig. 8(b). Like many other approaches, our algorithm has difficulties to disambiguate which leg is which in a key pose in strict side views. However, as soon as the view changes slightly, the correct pose is recognized.

In summary, our method detects people in the target posture with a very low error rate. The few false positives still correspond to people but at somewhat inaccurate scales or orientations. While this paper focuses on pure detection,

(a) (b)

Figure 8: Failure modes. (a) Wrong scale. (b) Wrong orientation.

it is therefore clear that the performance of our algorithm could be further increased by simple spatio-temporal filtering of several consecutive detections.

5 Conclusion

We have presented a method for human body pose detection that combines silhouette matching and motion information 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, we have already 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. Furthermore, since the detected templates are projections of 3-D models, mapping them back from 2-D to full 3-D poses is straightforward.

Our approach, even though tested on specific human motion, 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. Developing such tracker would be our long term task in the future.

References

[1] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. In *Conference on Computer Vision and Pattern Recognition*, 2004.

[2] A.A. Efros, A.C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *International Conference on Computer Vision*, pages 726–733, October 2003.

[3] A. Elgammal and C.S. Lee. Inferring 3D Body Pose from Silhouettes using Activity Manifold Learning. In *CVPR*, Washington, DC, June 2004.

[4] R. Fablet and M.J. Black. Automatic Detection and Tracking of Human Motion with a View-Based Representation. In *European Conference on Computer Vision*, May 2002.

[5] P. Felzenszwalb and D. Huttenlocher. Pictorial Structures for Object Recognition. *International Journal of Computer Vision*, 16(1), 2005.

[6] D. Gavrila, J. Giebel, and S.Munder. Vision-based pedestrian detection: the protector system. In *Intelligent Vehicles Symposium*, pages 13–18, 2004.

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

[8] J. Giebel, D.M. Gavrila, and C. Schnorr. A bayesian framework for multi-cue 3d object tracking. In *Proceedings of European Conference on Computer Vision*, 2004.

[9] P.J. Huber. *Robust Statistics*. Wiley, New York, 1981.

[10] Krystian Mikolajczyk, Cordelia Schmid, and Andrew Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *European Conference on Computer Vision*, volume I, pages 69–81, 2004.

[11] G. Mori, X. Ren, A.A. Efros, and J. Malik. Recovering Human Body Confi gurations: Combining Segmentation and Recognition. In *Conference on Computer Vision and Pattern Recognition*, Washington, DC, 2004.

[12] K. Okuma, A. Taleghani, N. de Freitas, J.J. Little, and D.G. Lowe. A boosted particle fi lter: multitarget detection and tracking. In *European Conference on Computer Vision*, Prague, Czech Republic, May 2004.

[13] C. F. Olson and D. P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Image Processing*, 6:103–113, January 1997.

[14] R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. In *European Conference on Computer Vision*, volume 4, pages 700–714, Copenhagen, Denmark, 2002.

[15] R. Rosales and S. Sclaroff. Infering Body Pose without Tracking Body Parts. In *Conference on Computer Vision and Pattern Recognition*, June 2000.

[16] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *International Conference on Computer Vision*, Nice, France, 2003.

[17] B. Stenger, A. Thayananthan, P.H.S. Torr, and R. Cipolla. Filtering Using a Tree-Based Estimator. In *International Conference on Computer Vision*, volume 2, pages 1063–1070, 2003.

[18] J. Sullivan and S. Carlsson. Recognizing and tracking human action. In *European Conference on Computer Vision*, 2002.

[19] P. Viola, M. Jones, and D.Snow. Detecting pedestrians using patterns of motion and appearance. In *International Conference on Computer Vision*, pages 734–741, 2003.