Body Tracking in Human Walk from Monocular Video Sequences

Frederic Jean, Robert Bergevin and Alexandra Branzan Albu
Computer Vision and Systems Laboratory (CVSL),
Department of Electrical and Computer Engineering,
Laval University, Sainte-Foy (Quebec), Canada, G1K 7P4.
E-mail : {fjean, bergevin, branzan}@gel.ulaval.ca

Abstract

This paper proposes a method to automatically track human body parts in the context of gait modelisation and recognition. The proposed approach is based on a five points human model (head, hands, and feet) where the points are detected and tracked independently. Tracking is fully automatic (no manual initialization of the five points) since it will be used in a real-time surveillance system. Feet are detected in each frame by first finding the space between the legs in the human silhouette. The issue of feet self-occlusion is handled using optical flow and motion correspondence. Skin color segmentation is used to find hands in each frame and tracking is achieved by using a bounding box overlap algorithm. The head is defined as the mass center of a region of the upper silhouette.

1. Introduction

Recently, a lot of research has been done in the field of intelligent surveillance systems [5, 12]. The main purpose of these systems is to track humans in uncontrolled indoor and outdoor environments, and eventually to describe what they are doing in order to detect abnormal and/or dangerous behaviors. The work described in this paper is part of the effort for developing one such system, which is named MONNET (Monitoring of Extended Premises: Tracking Pedestrians Using a Network of Loosely Coupled Nodes) [3, 8]. This system for indoor surveillance is constituted of several nodes, each consisting of two uncalibrated cameras (visible-infrared) linked to a computer. Nodes can exchange different human models in order to track and recognize people.

Different biometric methods can be used to efficiently recognize humans [7]. One of these methods aims at gait modeling and recognition. It can be used alone or in conjunction with other non-intrusive recognition methods. General human movement analysis methods can be classified into three categories [4]: 2D model-free methods, 2D model-based methods, and 3D methods. In the context of the MONNET system, which uses uncalibrated cameras, only 2D methods are relevant.

The 2D model-free methods are using low-level data (ex. silhouettes, optical flow, edge map) to describe human gait. These methods also embed in the gait model the shape and appearance of people, which is not desirable here since it introduces unnecessary photometric information in the gait model. Model-based methods describe body part trajectories, which are more representative of the human gait dynamics, and they avoid introducing spurious features in the model. However, body parts must be found and tracked in order to modelize the gait. This paper presents a method to dynamically track body parts of walking humans in order to develop a gait modeling and recognition module for the MONNET system.

Model initialization represents a central issue in 2D model-based human tracking. Methods that need manual initialization have focused on tracking features efficiently in each frame of video sequences, but need the user to specify these features in the first frame. That is the case in [9], where a chain structure of constrained patches are used as the human model and patch motion is estimated in each frame using an optical flow based method. Each patch location must be defined by the user in the first frame by specifying its four corners.

Some of the entirely automatic methods are using a frame-by-frame model fitting technique. In [11], a fifteen point stick-model is progressively fitted on a human silhouette using a distance transform. The fitting is done independently on each frame which often leads to incoherent skeleton sequences, especially in self-occlusion cases where there is no information available on some body part locations. A five-link biped locomotion model [13] can be fitted using some silhouette region for the shape model calculation, and color and inner silhouette regions for tracking. This method yields robust results only when the human subject is walking at 90° with respect to the camera axis.

Proceedings of the Second Canadian Conference on Computer and Robot Vision (CRV'05)
0-7695-2319-6/05 $ 20.00 **IEEE**

Methods that are similar to the one proposed in this paper use silhouette regions to find the location of body parts. In [1], the feet locations are defined as the local minima of the silhouette signature, which can yield poor performances in case of noisy background subtraction. The head is represented by the upper edge of the silhouette bounding box, which is also very sensitive to background subtraction noise. In [2], the bounding box silhouette is separated into three main sections (head, pelvis and feet) and the body part locations are described by the mass center of the foreground pixels in those sections. The feet region is further separated into two foot regions and the feet locations are the farthest foreground pixel positions from the head in the foot regions. The methods in [1, 2] do not take into consideration hand motion, feet self-occlusion and body parts correspondence. The method proposed in this paper will try to address these issues.

The rest of the paper is organized as follows. Section 2 contains a description of our approach. The experimental results are discussed in section 3. Section 4 contains the conclusions as well as the future work directions.

2. Approach

The proposed approach is based on a five points human model (head, hands, and feet) where the points are detected and tracked independently. The approach is characterized by a fully automatic initialization algorithm (no manual initialization of the five points) based on the only assumption that people walk upright in the video sequences. The input data is represented by color sequences from an uncalibrated monocular camera. The sequences in the experimental database contain a unique human subject walking in a typical indoor environment. The next sections present the pre-processing prior to body parts tracking, followed by the tracking algorithms which are adapted for movement characteristics of each body part.

2.1. Pre-processing

Initially, a simple background subtraction method is applied to each frame of the video sequence in order to find the silhouette of the person. The background subtraction method uses mean and standard deviation of each pixel computed from a subsection of the video sequence where the background is static and no walking human subject is present. Once the silhouette is detected, its bounding box is split into three regions according to fixed height proportions: head region, hands region and legs region. These fixed proportions were found experimentally and are working for all test sequences. Each of these regions will be used respectively for head, hands and feet tracking. Figure 1 shows the background subtraction and the silhouette bounding box splitting process.

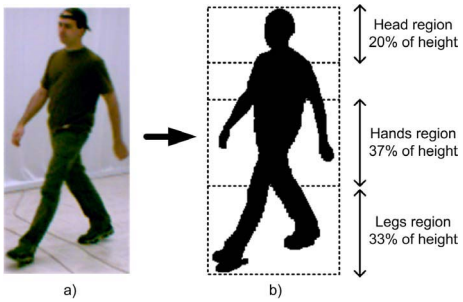

Figure 1. a) Original input color image. b) Resulting background subtraction and silhouette bounding box splitting.

2.2. Feet Tracking

A flowchart of the feet tracking method is presented in Figure 2. Input data for this method consist of the legs region belonging to the binary silhouette and the output is the position and correspondence of the feet. Separation of the legs is achieved for each frame, followed by localisation of the feet if the two legs can be separated. In that case, feet correspondence will be established if the correspondence is initialized, or correspondence is initialized if the legs were successfully separated in the previous frame. If the two legs cannot be separated in the current frame (occlusion case), then the correspondence is established with optical flow only if the correspondence was initialized. In other cases, there is simply no correspondence.

Separating legs can be efficiently achieved by scanning each line of the legs region and searching for foreground-background-foreground (FBF) patterns. The background in those patterns can potentially represent the space between the 2 legs for a given line, which can be further used to separate them. To find that space, it is possible to identify consecutive background pixels intervals in a line that are delimited by foreground pixels at both ends. If foreground pixels are represented by 1 and background pixels by 0 and $I(i,j)$ represents a pixel value at line i and column j for $i \in [1,N], j \in [1,M]$ (N and M are the height and width of the legs region respectively), the set S_i of starting and ending positions of intervals in the line i can be written as :

$$S_i : \Big\{ (b_i, e_i) \mid I(i, b_i - 1) - I(i, b_i) = 1 \wedge I(i, e_i) - I(i, e_i - 1) = -1 \wedge \forall j \in [b_i, e_i], I(i,j) = 0 \Big\} \quad (1)$$

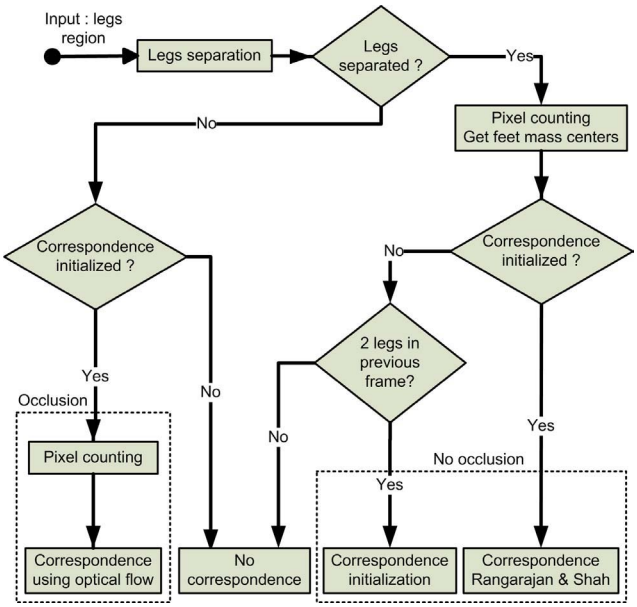

Figure 2. Overview of the feet tracking algorithm applied at each frame.

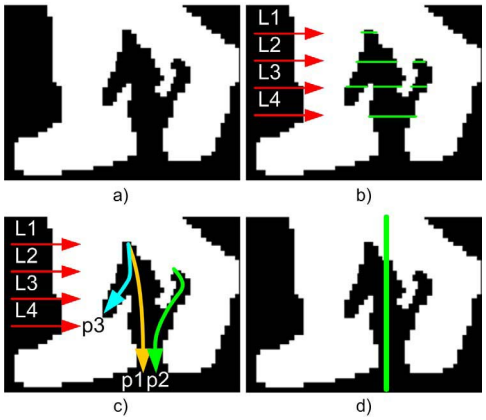

Figure 3. Example for separating the legs. a) An input noisy legs region. b) Lines are scanned to find intervals. An example of four line scans is shown, with lines named $L1$ to $L4$, and intervals are drawn. c) Path continuation. In $L1$, the only existing path is $P1$. In $L2$, there is path $P1$ and a new path $P2$ is created. In $L3$, $P1$ and $P2$ still exist, but there are two intervals that overlap with the previous interval of $P1$, that is why a new path $P3$ is created. In $L4$, only $P1$ and $P2$ still exist. d) Position for splitting legs region is found (longest path is $P1$).

where b_i and e_i are the beginning and ending position of an interval respectively. The first condition imposes that an interval begins with a foreground-background variation, while the second one imposes that an interval ends with a background-foreground variation. Finally, the last condition defines an interval as a consecutive set of zero pixels. Denoting K_i as the number of intervals found for line i, a specific interval can be defined as s_i^k with b_i^k and e_i^k the starting and ending position, $k \in [1, K_i]$. See Figure 3b for an example of lines scan and intervals.

Once all intervals are found for a line i, overlaps between these intervals and the ones of the previous line $(i-1)$ are searched in order to indentify vertically connected intervals. These vertically connected intervals are grouped together to form a path, which is defined as a set of intervals. When all lines have been scanned, the longest path will then be considered as the space between the two legs and a position will be computed from it to separate the legs region in two. A simple algorithm to find paths is shown in Figure 4. In this algorithm, a set P_i of continuing paths in line i is defined, and each continuing path is denoted p_i^l, $l \in [1, L_i]$, where L_i is the number of paths in P_i. An interval s_i^l is the interval that belongs to path l in line i. Each interval s_i^k of current line i that has an overlap with an interval s_{i-1}^l that belong to path p_{i-1}^l is included in that path. The resulting path is then included in set P_i since this path is continuing in line i. If two intervals overlap with an interval of a path of the previous line, this path is simply duplicated and the resulting paths are included in set P_i. A flag is used to indicate for each interval s_i^k if it has belonged to previous path(s). If it has not, then a new path is created with this interval. Examples of paths are shown in Figure 3c.

For $i = 1, 2, \ldots, N$

1. $P_i \leftarrow \{\}$
2. For $k = 1, 2, \ldots, K_i$
   (a) $S_{\texttt{flag=0}}$
   (b) For $l = 1, 2, \ldots, L_{i-1}$
      i. If s_i^k overlaps with s_{i-1}^l
         A. $P_i \leftarrow P_i \cup \{p_{i-1}^l \cup \{s_i^k\}\}$
         B. $S_{\texttt{flag=1}}$
   (c) If $S_{\texttt{flag=0}}$
      i. $P_i \leftarrow P_i \cup \{\{s_i^k\}\}$

Figure 4. Algorithm for paths detection.

The next step is to find the longest path p_f :

$$p_f = \underset{p_i^l,\, i\in[1,N],\, l\in[1,L_i]}{\arg\max} \left(\left|p_i^l\right|\right) \qquad (2)$$

where $\left|p_i^l\right|$ is the number of intervals in the path p_i^l (length). This path is accepted as representing the space between the

two legs only if it has a length that is greater than a fixed percentage (20%) of the height (N) of the legs region. If the path does not respect this condition, then legs are not separated and this is defined as an occlusion case. When the condition is respected, the path can be used to find a position where the legs region bounding box can be separated vertically to provide two bounding boxes that represent two leg regions. The minimum and maximum position b_{min} and e_{max} among intervals belonging to the path p_f must first be found :

$$\begin{aligned} b_{\text{min}} &= \min_{s_f\,:\,[b_f,e_f]\,,\,s_f \in p_f} (b_f) \\ e_{\text{max}} &= \max_{s_f\,:\,[b_f,e_f]\,,\,s_f \in p_f} (e_f) \end{aligned} \tag{3}$$

where b_f and e_f are beginning and ending positions of an interval of p_f. It is now possible to find from the path p_f the position of a vertical line separating the legs (see Figure 3d). This line represents a successful trade off between minimizing the number of contained foreground pixels and maximizing the distance of the vertical line position with respect to the two feet regions. For a position $j \in [b_{\text{min}}, e_{\text{max}}]$ and a given (b, e) starting/ending position pair, a function $V(j, b, e)$ is computed as follows :

$$V(j,b,e) = \begin{cases} j-b & \text{if } |j-b| < |e-j| \\ e-j & \text{otherwise} \end{cases} \tag{4}$$

This function computes a score for a given position j. If the vertical line position is outside the range $[b, e]$, the value of the function is negative (bad score) which means that the vertical line position contains a foreground pixel. However, if the position is in the range $[b, e]$, a positive value is obtained and the vertical line position does not intercept a foreground pixel (the highest score is obtained at position $(b + e)/2$). This function is used to compute a score for a given vertical line position j for all intervals belonging to the path p_f. These score are summed to obtain a general score for each j position, and the position j_{max} is the one that produces the highest score :

$$j_{\text{max}} = \underset{j \,\in\, [b_{\text{min}}, e_{\text{max}}]}{\arg\max} \left(\sum_{\forall\ [b_f,e_f] \,\in\, p_f} V(j, b_f, e_f) \right) \tag{5}$$

Finally, the legs region bounding box is split in two bounding boxes at position j_{max} (see Figure 3d).

Feet correspondence is computed when the legs region has been separated into two regions (no self occlusion). It uses each foot mass center from the current and previous frame(s). To find the foot in each leg region, a pixel counting algorithm is used to select pixels that will be used to compute a mass center. It scans lines of each foot region bottom-up and cumulates the number of foreground pixels it encounters. When this number is greater than 25% of the initial legs region area, all counted pixels are aggregated into a foot region and the region center of mass is computed. Figure 5 shows a typical result of this algorithm.

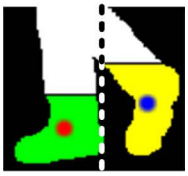

Figure 5. Aggregation of foreground pixels of each leg region (from bottom) until 25% of the number of foreground pixels in the legs region is reached. The mass center of the aggregated pixels is then computed to define a foot position.

This method is more robust than fixed region methods (as in [2]) because it relies on regions that are dynamically defined by the legs separating algorithm. It also finds a foot no matter what is its height in the legs region.

If in the current frame t foot mass centers are defined as f_a^t and f_b^t, and feet correspondence has not been initialized yet, the correspondence is initialized with a nearest point algorithm. The initialization can only be carried out if there were 2 feet detected in previous frame[1] $(t-1)$, where those 2 feet positions are defined as f_1^{t-1} and f_2^{t-1}. The nearest point algorithm simply does a natural feet correspondence as follows :

$$(a,b) = \begin{cases} (1,2) & \text{if } d_{a1} < d_{a2} \wedge d_{a1} < d_{b1} \\ (2,1) & \text{if } d_{a2} < d_{a1} \wedge d_{a2} < d_{b2} \\ (1,2) & \text{if } d_{b1} < d_{b2} \wedge d_{b1} < d_{a1} \\ (2,1) & \text{if } d_{b2} < d_{b1} \wedge d_{b2} < d_{a2} \end{cases} \tag{6}$$

where d_{a1} is the mass center distance of foot f_a^t and f_1^{t-1} and so forth. In the case where an initial correspondence cannot be established (f_1^{t-1} and f_2^{t-1} are at equal distance from f_a^t and f_b^t), the initialization is simply delayed to the next frame to avoid a bad foot correspondence initialization.

In the case where the correspondence is already established in frame t, the motion correspondence algorithm developed by Rangarajan & Shah [10] is used. This algorithm basically builds a correspondence between moving feature points assuming that points follow a smooth trajectory and have a small spatial displacement between each

1 Feet must have been found with the pixel counting algorithm on the two feet regions obtained from the legs separating method (i.e. no occlusion in the previous frame).

frame (which is true in the present case). This algorithm provides robustness to the proposed approach since the feet correspondence can be preserved in case of a sudden noisy human silhouette (ex. a leg part disappears from one frame to the next). However, it requires that the correspondence is correctly initialized in the two previous frames. This is why the nearest point algorithm is used to initialize correspondence as soon as there are two consecutive frames with feet detected only with the separating legs and counting pixels algorithms. Since in those cases the feet are far enough from each other and they do not move rapidly, the initial correspondence can be established correctly with the nearest point algorithm.

Self-occlusion occurs when two feet were detected in the previous frame but they could not be separated in the current one. The first step that is performed is to define a feet region with the help of the same counting pixel algorithm described previously, using two times the percentage used for the case without occlusion (because the foreground pixels in the legs region contain two feet).

Feet self-occlusions that occur when a human is walking have the interesting particularity that there is only one visible foot moving. Based on experimental observations, the speed of the moving foot is at its maximum in the self-occlusion case whereas the other foot is not moving at all. With respect to the above remarks, it is possible to determine which foot was moving in the previous frame $(t-1)$ by comparing the difference between each foot x position from the frame $(t-1)$ and the frame $(t-2)$:

$$m = \begin{cases} 1 & \text{if } \left|f_{1x}^{t-1} - f_{1x}^{t-2}\right| > \left|f_{2x}^{t-1} - f_{2x}^{t-2}\right| \\ 2 & \text{otherwise} \end{cases} \quad (7)$$

where m represents the moving foot (foot 1 or foot 2). The position (mass center) of the stationary foot is simply considered to be the same as in the previous frame. To determine the position of the moving foot, optical flow is computed using the method developed by Horn & Schunck [6]. This method uses data from the current and previous grey frame only in the feet region, and therefore it is very fast. The absolute value of the optical flow is computed at each pixel of the foot region and only the pixel positions that exceed an absolute value of 10 are kept to form a mask (this value has been determined experimentally and works for all test sequences). Thereafter, morphological dilatations are applied to the resulting mask and a logical `and` operator is applied between this mask and the original feet region silhouette. This leads to another mask which represents what has moved only in the feet region silhouette. The counting pixel algorithm is finally used on the mask with the same percentage as without occlusion (resulting pixels belong only to the moving foot). The mass center of the aggregated pixels is computed and associated with the moving foot. In the cases where there are not enough pixels positions that exceed the predefined absolute value, the moving foot positions is approximated by a linear extrapolation from the two previous moving foot positions. It is important to note that the foot correspondence is implicit in the occlusion case, since the stationary and the moving foot are tracked differently.

2.3. Hands Tracking

Finding hands in each frame of a video sequence is more difficult than finding feet since hands are often in self-occlusion with the body (except for the frontal view, which is not very meaningful for gait modelling). It is not possible to rely only on the human silhouette because of the fact that hands detach from one side of the silhouette for only a few frames and do not always reappear and detach from the other side of the silhouette. One simple way of finding the hands in this case is to use skin color detection in the hands region (see Figure 1). Using this method, it is possible to detect the hands most of the time (with the exception of the frames where one of the hands is hidden by the body).

Hands skin color detection is done using the HSV color space in the hands region. Pixels in this region that belong to skin color category are found using predefined thresholds for H and S. Thereafter, too small skin regions are removed and morphological dilatations followed by morphological erosions are applied on the remaining skin regions. The two largest skin regions are kept - if they exist - and the second largest region is further considered only if its size is at least a given percentage (15%) of the largest region size. Finally, a hand position is found using the lowest focus of the ellipse that has the same second moment as the skin region. Considering the lowest focus position leads to better results than considering the lowest skin pixel position, with respect to smoothness of the trajectory. If the lowest focus does not fall on the silhouette, then the nearest silhouette point position with respect to the focus is used (see Figure 6).

Hand tracking is achieved by using a bounding box overlapping algorithm. This kind of algorithm is used since it is not possible to determine a priori if the two hands will eventually be visible again or only one of them (or even none of them). Therefore, it is not possible to initialize a motion correspondence algorithm like the one used for the feet.

A hands correspondence must also be initialized and computed. First, a bounding box h^t is computed for each hand pixel region that exists in the current frame t. For each detected hand in the previous frame $(t-1)$, an estimated bounding box h_{est}^t is computed for the current frame and it is compared with the hand(s) bounding box(es) of the current frame. The estimated bounding box is computed in a

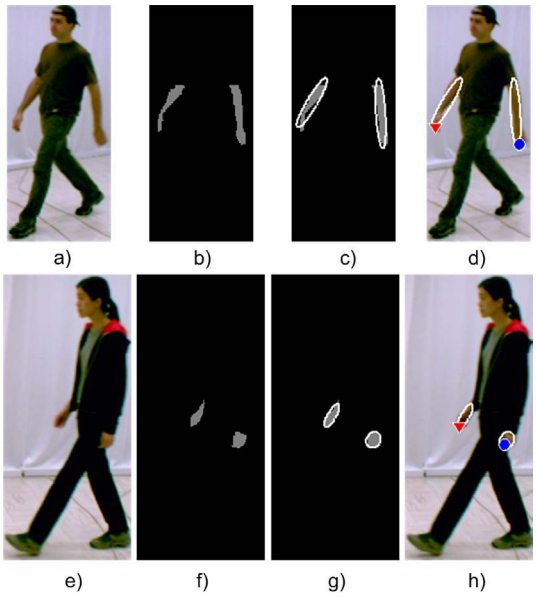

Figure 6. Finding hands using skin color. a) and e) Original color frames. b) and f) Results of skin color segmentation, the two largest skin regions en conserved. c) and g) Ellipses that have the same second moment as each skin region are found. d) and h) Hands position are defined as the lower foci of ellipses.

different way depending on if the correspondence initialization was carried out or not in the previous frame. If it was not (see Figure 7a), the estimated bounding box is h^{t-1} with the x position moved by the difference of x positions of the human silhouette mass centers between frames t and $(t-1)$. This method moves the x position considering the global motion of the walking person. In the case where the correspondence was established (see Figure 7b), the bounding boxes of frame $(t-1)$ and $(t-2)$ are used to estimate a bounding box in the current frame. Its width and height are computed as a linear extrapolation of width and height of the two previous bounding boxes. For example, if a bounding box in frame $(t-2)$ has width 5 and height 10 and a bounding box in frame $(t-1)$ has width 4 and height 12, then the estimated bounding box will have a width 3 and a height 14. The position of the estimated bounding box is also linearly extrapolated from the corner of the two previous bounding boxes with the smallest variation in position. For instance, if the corner with the smallest position variation moves from point $(1,5)$ to point $(3,4)$, then its new position will be $(5,3)$ and the bounding box is repositioned according to the new position of that corner. Correspondence algorithms are based on the fact that the hands have small displacements and small changes in appearance (hand skin area that can be seen) between consecutive frames.

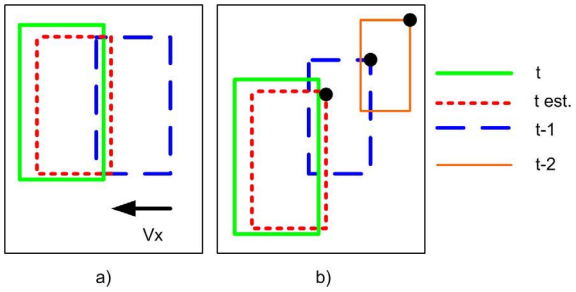

Figure 7. Estimation of a bounding box in the current frame. a) Correspondence initialization is not carried out. The estimated bounding box is simply the one at frame $(t-1)$ with x displacement V_x corresponding to displacement of the silhouette mass center. b) Correspondence initialization has already been carried out. The estimated bounding box applies the growth rate in width and height of the bounding boxes from frame $(t-1)$ and $(t-2)$. The position is defined by the corner that has the smallest displacement (black dots). In a) and b), the actual bounding box found in frame t is shown (plain line) to present the kind of overlap that occurs with the estimated bounding box.

As said previously, the hands correspondence is achieved by comparing their current bounding boxes with estimated bounding boxes. The way the current bounding box overlaps with an estimated bounding box determines if a correspondence is established or not between hands. The first condition for a correspondence between these bounding boxes is that they must have an area ratio larger than a predefined value (33%). The ratio is computed between the smallest bounding box area and largest one. This condition will not be met if the bounding boxes do not have a similar area. The second condition that must be met is that the two bounding boxes must have at least a certain area in common (33%). The ratio between the area of the common region and the area of the smallest bounding box is a good measure to verify that condition. If the ratio is 0, that means there is no overlap between the bounding boxes, and if the ratio is 1, that means that the smallest bounding box is totally included in the largest one. These two conditions must be met to establish a hand correspondence.

2.4. Head Representation

The head position can be represented in two ways : the position of the very top pixel of the silhouette, or a mass

center of a region. The second approach has been retained since it is less sensible to background subtraction noise than the first one. To find the region of pixels for which the mass center will be computed, the same counting pixel algorithm presented previously is applied. Using a percentage (66%) of pixels present in the head region, the counting pixel algorithm is used top-down on the head region. The mass center of the retained pixels is then computed and used as the head position. The head is not tracked since it has the same displacement as the whole body, which is not the case for the hands and the feet. An example can be seen in Figure 8.

Figure 8. Finding head and computing its mass center.

3. Experimental results

A special acquisition setup was designed in order to test the proposed approach. As shown in Figure 9, four cameras are used in order to obtain different views of a person walking on a straight line trajectory. The 4 views are captured at the same time so the camera sees exactly the same movement made by the person. Video sequences of ten walkers were acquired, which resulted in forty test sequences with an average of 500 frames per sequence. All algorithm parameters were the same for all human subjects and all views.

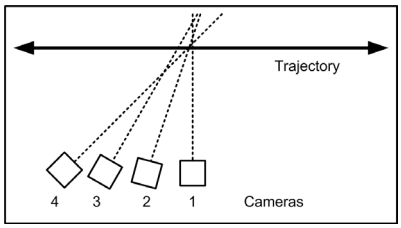

Figure 9. Setup for the acquisition of walking people from different views. The optical axis of camera 1 intersects the walking trajectory at 90°, camera 2 at 75°, camera 3 at 60° and camera 4 at 45°.

Results obtained with the proposed approach will be compared with results from another approach [11] which uses a skeleton model fitted on a human silhouette on each frame of video sequences (see section 1 for details). This comparative approach represents the class of algorithms that use only spatial information to track human body parts (and do not use spatiotemporal information as in the proposed method). The aim of this comparison is to show that the description of human movement is a complex issue that cannot be well handled using only spatial information on a frame-by-frame basis.

Figures 10, 11 and 12 show algorithms comparison for walking people viewed from different angles. The first row represents results of the proposed approach (named P) and the second row represents results of the static skeleton approach (named S), and the frames[2] order is from left to right. There are two different symbols for the hands and the feet but none of them is associated with a specific body-centered part (left or right). In the case of the proposed approach, the symbol associated with a given hand or foot remains the same over each sequence. That is, no false correspondence of hands or feet occurred with the proposed approach within the whole forty video sequences. This is a very significant improvement over the static skeleton algorithm which produced a large number of false correspondences.

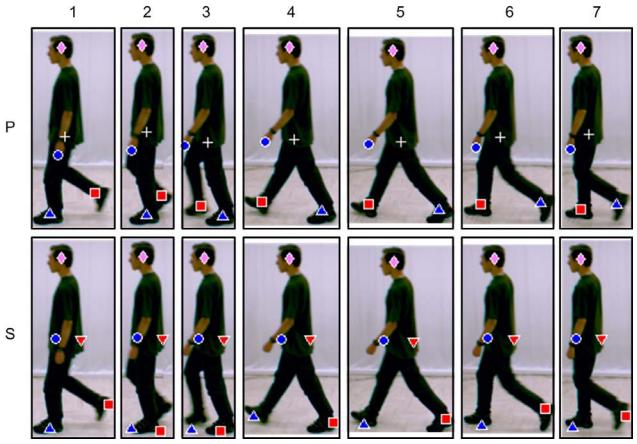

Figure 10. Walking sequence from camera 1.

One can see in the figures how well the proposed feet tracking algorithm works. In fact, using motion correspondence and optical flow (temporal information) leads to more realistic feet trajectories, which is a great advantage for modeling the human gait precisely. For the head, positions are nearly the same.

2 These frames are distanced in time by two frames in order to show interesting events without taking too much space.

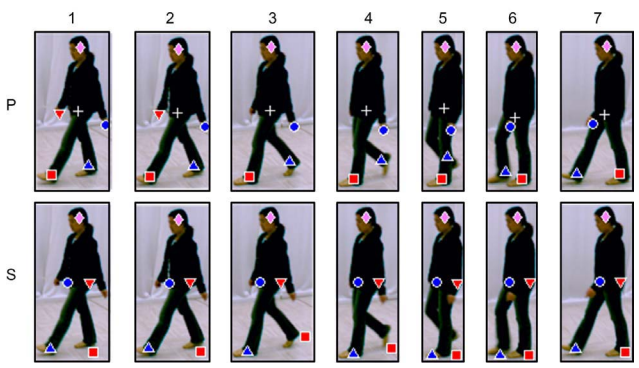

Figure 11. Walking sequence from camera 3.

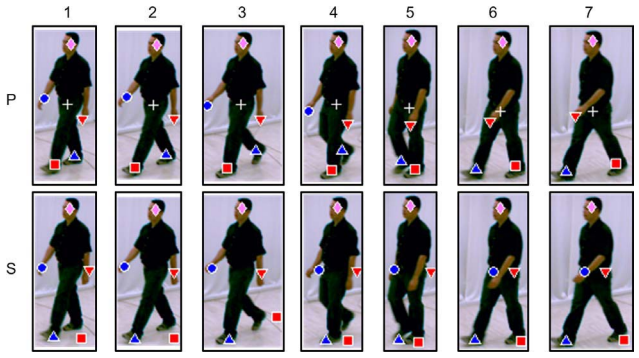

Figure 12. Walking sequence from camera 4.

One disadvantage of the proposed approach is the fact that the position of the two hands is not always given. However, in the cases where they are found, the tracking and the correspondence is established correctly. It is possible to see one case in figure 12, frame 1 to 3, where the skeleton algorithms produce a better right hand position than the proposed approach. Using the skin color information does not always allow an accurate detection of the hands skin region, which sometimes leads to erroneous positions.

4. Conclusion and Future Works

An entirely automatic method for dynamically tracking the head, hands and feet of a walking person has been presented. Feet are tracked using a legs separating algorithm and motion correspondence algorithms. Hands are detected using skin color and are tracked using a bounding box overlap algorithm. Finally, the head is represented by the mass center of a region of the upper silhouette part. The robust tracking of the five points model proposed in this works will be useful in the context of human gait modelization and recognition.

Future work will be focused on refining hands detection and tracking algorithms. Normalization of the body parts trajectories is also envisaged in order to modelize the human gait in a view-independent manner.

Acknowledgements: This work is supported by Precarn inc. through the MONNET project.

References

[1] C. BenAbdelkader, R. Cutler, and L. S. Davis. View-invariant estimation of height and stride for gait recognition. In *Proc. of the International ECCV Workshop on Biometric Authentication*, pages 155–167, 2002.

[2] A. F. Bobick and A. Y. Johnson. Gait recognition using static activity-specific parameters. In *Proc. of CVPR Conference*, 2001.

[3] R. Drouin, D. Laurendeau, and A. Branzan-Albu. Monnet : Simulation of the inter-node communication strategy for a network of loosely-coupled nodes in a surveillance application. In *14th Canadian Conference on Intelligent Systems*, volume 1, page 23, June 6-8 2004.

[4] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding: CVIU*, 73(1):82–98, 1999.

[5] I. Haritaoglu, D. Harwood, and L. S. Davis. W4 : Real-time surveillance of people and their activities. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(8):809 – 830, 2000.

[6] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[7] A. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *IEEE Trans. on Circuits and Systems for Video Technology*, 14(1):4 – 20, 2004.

[8] F. Jean, S. Quirion, R. Bergevin, and A. Branzan-Albu. Skeleton-based segmentation and recognition of human activities from video sequences. In *14th Annual Canadian Conference on Intelligent Systems*, volume 1, page 46, June 6-8 2004.

[9] S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated image motion. In *Proc. of the 2nd International Conference on Automatic Face and Gesture Recognition*, pages 38–44, 1996.

[10] K. Rangarajan and M. Shah. Establishing motion correspondence. *CVGIP: Image Understanding*, 54(1):56 – 73, 1991.

[11] J. Vignola, J.-F. Lalonde, and R. Bergevin. Progressive human skeleton fitting. In *Proc. of the 16th Conference on Vision Interface*, pages 35–42, 2003.

[12] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder : Real-time tracking of the human body. In *Proc. of SPIE Conference on Integration Issues in Large Commercial Media Delivery Systems*, 1995.

[13] R. Zhang, C. Vogler, and D. Metaxas. Human gait recognition. *IEEE Workshop on Articulated and Nonrigid Motion*, 2004.