Computing and Evaluating View-normalized Body Part Trajectories

Frédéric Jean [a,*] Robert Bergevin [a] Alexandra Branzan Albu [b]

[a] *Computer Vision and Systems Laboratory, Dept. of Electrical and Computer Engineering, Laval University, Québec, QC, Canada, G1K 7P4*

[b] *Laboratory for Applied Computer Vision Algorithms, Dept. of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada, V8W 3P6*

Abstract

This paper proposes an approach to compute and evaluate view-normalized body part trajectories of pedestrians from monocular video sequences. The proposed approach uses the 2D trajectories of both feet and of the head extracted from the tracked silhouettes. On that basis, it segments the walking trajectory into piecewise linear segments. Finally, a normalization process is applied to head and feet trajectories over each obtained straight walking segment. View normalization makes head and feet trajectories appear as if seen from a fronto-parallel viewpoint. The latter is assumed to be optimal for gait modeling and identification purposes. The proposed approach is fully automatic as it requires neither manual initialization nor camera calibration. An extensive experimental evaluation of the proposed approach confirms the validity of the normalization process.

Key words: body parts trajectories, view-invariance, normalization, gait

1 Introduction

A well-known medical study has shown that human gait is a complex motion that may be decomposed into twenty significant components [1]. It is believed that the complexity of the interactions between the various components encodes relevant information about the identity of the moving person. Recent

[*] Corresponding author. Tel.: +1 418-656-2131 ext. 4786; fax: +1 418-656-3594.
Email addresses: fjean@gel.ulaval.ca (Frédéric Jean), bergevin@gel.ulaval.ca (Robert Bergevin), aalbu@ece.uvic.ca (Alexandra Branzan Albu).

Accepted for publication in Image and Vision Computing *November 2008*

progress in computer-based analysis of gait has confirmed its potential as a biometric feature. First and foremost, gait analysis allows for person identification at a distance, which is difficult or even impossible with other biometric techniques such as retinal scanning, fingerprints, or face recognition.

Gait-based person identification represents a key element in the design of robust visual surveillance systems. To the best of our knowledge, gait-based identification has not yet been integrated in surveillance systems. Previous computer-based surveillance systems [2] have focused mostly on pedestrian detection [3] and low-level tracking of human subjects using basic 2D appearance models [4]. More sophisticated 3D part-based models may also be obtained but they require manual interventions for initialization [5]. Besides, stereo data is required for methods using 3D temporal motion models [6]. Finally, 3D models are computationally expensive and therefore difficult to use in real-time surveillance.

A key surveillance issue addressed by the proposed approach is the varying angle between the camera optical axis and the walking trajectory direction of an observed pedestrian. This phenomenon introduces a variation of the human motion captured with an uncalibrated camera. In practice, many gait modeling approaches are either only applicable to fronto-parallel viewpoints [7,8], or at least view-dependent [9]. Height and stride length are estimated in a view-invariant way in [10], but the method necessitates the camera to be calibrated with respect to the ground, which could be problematic in a realistic surveillance context. The method proposed in [11] uses the structure of articulated body part motion to recover the parameters of the projective transformation under which a subject is observed. The projective transformation is then used to generate a canonical fronto-parallel view. That method uses markers to obtain precise positions of the ankles, knees and hip, which are difficult to retrieve automatically from computer vision algorithms. Synthesis of a canonical side view from an arbitrary view is performed in [12] via two methods, namely perspective projection and optical flow-based structure-from-motion. However, the synthesis of a side view is only feasible from a limited number of initial views. The method in [13] involves a scaling process, for each known view, on silhouette parameters such as height and distance between head and pelvis. In [14], a method for tilt correction of silhouettes is proposed, but it requires camera calibration. Estimation of a person's 3D trajectory from a monocular calibrated camera is discussed in [15]. The 3D trajectories are used in order to recover the walking speed. Walking directions of people are computed in [16] using a camera with known focal length and the weak-perspective projection model. The walking direction is then used to recover view-invariant lengths at different parts of the silhouettes.

Other methods integrate the information from multiple views in order to recover a canonical view or to extract features that are view-invariant. This is the case of the method presented in [17] where the desired view of a moving object is reconstructed using multiple simultaneous views. The method proposed in [18] achieves view-invariance by learning gait parameters from mutiple views, and people identification is performed by providing only a single view. In [19], a bilinear model is fitted on mutiple views. View-invariant identification is achieved by decoupling the identity of the person and the viewpoint from which he is observed. A view transformation model is used in [14] in order to transform already observed gait features into the same walking direction as newly observed features. The view transformation model is learned from multiple views of walking subjects. Those methods are difficult to carry out in the context of a surveillance system where typically a single view is available. Moreover, a general learning phase could be hard to obtain.

In many realistic settings, for instance when pedestrians are observed in extended premises via a network of loosely-coupled nodes [20], an efficient and automatic modeling approach is required. Therefore, in order to address the constraints of a real-time surveillance system, we propose to use 2D trajectories of body parts for modeling gait. Our work is similar to [21] and [22] with respect to the fact that we use an implicit kinematic model for gait. Specifically, we extract spatiotemporal trajectories of body parts (head and feet) for modelling gait. Our method for trajectory generation improves upon previous work by solving the manual initialization issue in [21] and by extracting the spatiotemporal trajectories in real time from video data instead of obtaining them from a marker-based motion tracker [22]. Our main contribution consists in a novel technique for viewpoint normlization which is summarized below.

The trajectory of a body part (foot, head, etc.) is defined as a sequence of the successive 2D positions it takes in the frames of a video sequence. On a frame-by-frame basis, each body part is represented by one point respectively. Body part trajectories are assumed to contain sufficient information about the gait of a person for view-invariant modeling and identification; this assumption is based on early work by Johansson [23] and on related work on kinematic gait models [21] [22]. The walking trajectory, which is the path followed by a person on the floor, is not assumed to be a single straight line. Instead, it is assumed to be a polyline that is, a sequence of straight-line segments of variable orientations and lengths. View normalization consists in making body part trajectories appear as if seen from the same fronto-parallel viewpoint for all straight-line walking segments of the video sequence. The proposed approach to view normalization features automatic initialization, no camera calibration, as well as a low computational complexity.

This paper is an extension of the work presented in [24]. A new evaluation method is proposed to assess the performance of the normalization algorithm. An extensive evaluation is performed on more than 80 walking trajectories.

The remainder of the paper is organized as follows. Preprocessing is presented in Section 2. The proposed approach is detailed in Section 3. The evaluation method is dicussed in Section 4. Experimental results are presented in Section 5. Finally, Section 6 draws conclusions and outlines the main directions of future work.

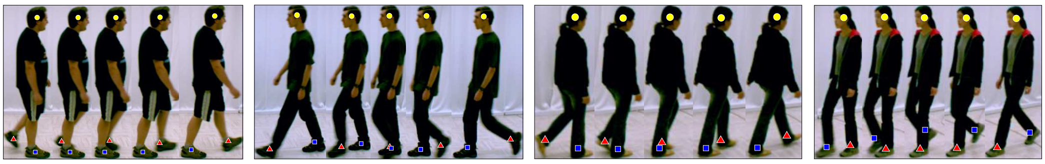
2 Preprocessing

Input data for the proposed approach consists of a set of 'raw' (i.e. view-dependent) body part trajectories. One should note that our normalization approach stays the same regardless on which (or how many) body parts are collected from the human walk. We present results for feet and head only, as we consider that the motion of these body parts encodes core, irreducible information about walking. However, our view normalization approach is compatible with any other algorithms that extract spatiotemporal trajectories of body parts, including optical motion capture technology.

For the purpose of this study, the 'raw' feet and head trajectories are generated via an algorithm customized for human walking. This algorithm is summarized below; its detailed presentation can be found in [25].

The head trajectory consists of the sequence of locations of the center of mass of the head extracted on a frame by frame basis. The generation of this trajectory is straightforward, as the head is always the highest part of the human silhouette and does not suffer any occlusions during walk. This is however not the case with feet in monocular sequences. Feet occlude themselves periodically in every viewpoint except one, where the subject walks along the optical axis of the camera (0 degrees). This self-occlusion needs to be addressed in order to obtain a correct feet correspondence (i.e. left-to-left and right-to-right) across every pair of adjacent frames in the sequence. The feet inter-frame correspondence algorithm handles all possible correspondence cases, as follows. Feet are first detected as regions in the image, and then each represented by one point on a frame by frame basis. In cases where legs are separable, these representative points are estimates of the centers of mass of the leg regions. First, inter-frame correspondence is initialized using an intuitive nearest point criterion. This criterion states that the 'right' foot in Frame i must be spatially closer to the 'right' foot than to the 'left' foot in Frame (i+1); a similar reasoning is applied for the 'left' foot correspondence. One should note that 'right' and 'left' have arbitrary meanings here, since the algorithm is not able to distinguish right from left in a monocular uncalibrated sequence.

Once correspondence is initialized,the tracking algorithm propagates this correspondence for every subsequent pair of adjacent frames. In doing so, it needs



(a) 90°
(b) 75°
(c) 60°
(d) 45°

Fig. 1. Body part tracking from different views

to evaluate whether self-occlusion is present or not.

In case self-occlusion is present, the legs' regions merge into one region. In this case, the legs' representative points are not retrievable as centers of mass and thus need to be estimated using motion information. One may note that, in human gait, feet self-occlusions have the interesting particularity that there is only one visible foot moving, while the other is grounded as support foot. Therefore, we retrieve the representative point of the moving foot using optical flow, while the point representing the stationary foot is the same as in the previous frame. Figure 1 is an excerpt of head and feet positions obtained for four pedestrians observed from different viewpoints. Squares and triangles represent feet positions while disks represent head positions. One can notice that correspondence is properly achieved by looking at symbols on the feet.

3 View Normalization of Body Parts Trajectories

An overview of the approach is presented first. A detailed discussion of the algorithms follows.

3.1 Overview

In realistic surveillance situations, pedestrians cannot be assumed to always follow a single straight line. Besides, their walking trajectory cannot be known in advance. The proposed approach deals with both difficulties by first estimating the walking trajectory using the original feet trajectories. The estimated walking trajectory is then "spatiotemporally" decomposed into piecewise linear segments. Original and normalized plane parameters are computed for each of those segments. Finally, body part trajectories of each segment are normalized using the computed homography between the corresponding planes.

Figure 2 presents a situation where a person follows a straight walking trajectory. In this case, the person is to occupy different positions with different postures while walking from the right-hand side at time t_1 to the left-hand side at time t_2. The walking trajectory angle with respect to the camera optical axis

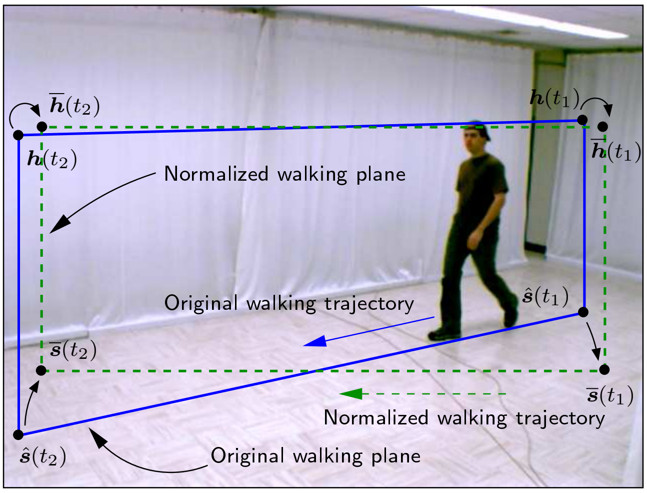

$\overline{\boldsymbol{h}}(t_2)$
$\boldsymbol{h}(t_1)$
$\overline{\boldsymbol{h}}(t_1)$
$\boldsymbol{h}(t_2)$
Normalized walking plane
Original walking trajectory
$\hat{\boldsymbol{s}}(t_1)$
$\overline{\boldsymbol{s}}(t_2)$
$\overline{\boldsymbol{s}}(t_1)$
Normalized walking trajectory
$\hat{\boldsymbol{s}}(t_2)$
Original walking plane

Fig. 2. View normalization

is about 45°. The "original walking plane" is formed by joining corresponding positions along the walking trajectory and head trajectory (the method used to estimate the walking trajectory is described below). This plane indicates how the pedestrian is positioned with respect to the camera over a given time slice. Here, he is far from the camera at right and closer to the camera at left. Due to perspective geometry effects, the plane edges are not parallel to the image frame.

A "normalized plane" is defined as having edges parallel to the image frame. The original plane is already normalized when the walking trajectory is fronto-parallel that is, when the straight walking trajectory is perpendicular to the optical axis of the camera. In other cases, such as the one illustrated in Figure 2, an original plane may be normalized using homography-based transformation. The homography matrix may be computed once a correspondence is established between the four corners (top and bottom positions) of the original plane ($\boldsymbol{h}(t_1)$, $\hat{\boldsymbol{s}}(t_1)$, $\boldsymbol{h}(t_2)$ and $\hat{\boldsymbol{s}}(t_2)$) and four corners of the normalized plane ($\overline{\boldsymbol{h}}(t_1)$, $\overline{\boldsymbol{s}}(t_1)$, $\overline{\boldsymbol{h}}(t_2)$ and $\overline{\boldsymbol{s}}(t_2)$) at time t_1 and t_2. The computed homography is applicable to all points in the plane, and in particular to the coordinates of the body parts between time t_1 and t_2 in order to transform them from the original plane to the normalized plane. Normalized body part trajectories appear as obtained from a fronto-parallel viewpoint. An important assumption behind the normalization is that the motion of each foot occurs in a plane parallel and close to the original walking plane. This assumption holds well when the distance to the camera is large compared to the size of the pedestrian.

3.2 Estimating the Walking Trajectory

The walking trajectory is computed using the feet trajectories obtained with preprocessing. On a frame-by-frame basis, the walking trajectory consists in one point defined by its 2D coordinates in the image plane. As shown in Figure 3, the estimated walking trajectory appears as a series of segments separated by local discontinuities caused by temporary feet self-occlusion. The main issue in estimating the walking trajectory is the selection of the representative point. The vertical projection $\hat{\boldsymbol{s}}$ of the silhouette mass center $\boldsymbol{s} = [s_x\ ,\ s_y]^{\mathrm{T}}$ on the line joining the feet positions on the floor provides a global estimate of the walking trajectory. The x-coordinate of $\hat{\boldsymbol{s}}$ is identical to s_x, while the y-coordinate is computed as follows. The position of each foot on the floor ($\hat{\boldsymbol{f}}^1$ and $\hat{\boldsymbol{f}}^2$) is obtained by vertically projecting its mass center ($\boldsymbol{f}^1$ and $\boldsymbol{f}^2$) to the lowest silhouette pixel at that x position: $\hat{\boldsymbol{f}}^1 = [f^1_x\ ,\ \mathcal{Y}(f^1_x)]^{\mathrm{T}}$, $\hat{\boldsymbol{f}}^2 = [f^2_x\ ,\ \mathcal{Y}(f^2_x)]^{\mathrm{T}}$. In these equations, $\mathcal{Y}(f^1_x)$ and $\mathcal{Y}(f^2_x)$ represent the lowest y pixel positions on the silhouette at the x position of the mass center of foot 1 and 2, respectively. The projected silhouette mass center is then computed as :

$$\hat{\boldsymbol{s}} = \left[s_x\ ,\ \frac{\hat{f}^2_y - \hat{f}^1_y}{\hat{f}^2_x - \hat{f}^1_x}(s_x - \hat{f}^1_x) + \hat{f}^1_y \right]^{\mathrm{T}} \tag{1}$$

The projected silhouette mass center provides a good sample on the walking trajectory as long as both feet are touching the floor. When one foot is moving, it could still be close to the floor. In that case, the resulting error is acceptable. However, when one foot is occluded, the moving foot is usually farther from the floor and a large error could be introduced. The tracking algorithm computing the original body part trajectories detects feet occlusions. This enables the mass center of the silhouette to be projected only when both feet are visible and close to the floor. As a result, piecewise-continuous walking trajectory samples are obtained as shown in Figure 3. The segmentation of the estimated walking trajectory into linear trajectory segments is the next step of the algorithm.

3.3 Segmenting the Walking Trajectory

For the purpose of clarity, the trajectory segmentation algorithm is described on a sequence with a single sudden walking direction change (Figure 4). However, our approach is able to handle multiple changes of direction. The original head and feet trajectories obtained by the tracking algorithm are shown in Figure 4(a). Figure 4(b) presents the piecewise-continuous walking trajectory samples obtained by projecting the silhouette's mass center on the floor. The

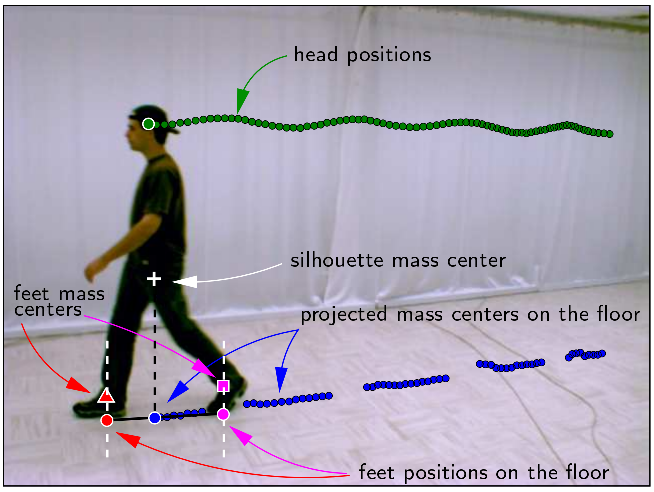

head positions
silhouette mass center
feet mass centers
projected mass centers on the floor
feet positions on the floor

Fig. 3. A typical walking trajectory

main idea behind the segmentation algorithm is to fit a straight-line segment to each "continuous" group of point samples followed by the estimation of junction points linking consecutive segments (see Figure 4(c)). The obtained samples are typically noisy which makes it difficult to compute junction points as intersections of fitted lines. The computed junction points are next considered as samples on a curve to be approximated by a polyline (open polygon). The number of straight-line segments in the polyline should match the number of straight walking segments in the pedestrian's trajectory under the reasonable assumption that the trajectory is piecewise linear. That is, significant corners have to be identified along the junction-sampled curve.

A group $\mathcal{G}^k$ of samples covers an interval $[t_b^k, t_e^k]$, where t_b^k is the first sample after an occlusion and t_e^k is the last sample before the next occlusion. $N_{\mathcal{G}}$ is the number of groups and $k = 1 \dots N_{\mathcal{G}}$. If a group $\mathcal{G}^k$ has less than three samples, then it is merged with the group $\mathcal{G}^{k+1}$. In the case where a change in the x direction within a group $\mathcal{G}^k$ is detected, the group is split into two groups at the sample where the change in x direction occurs (the walking direction changed horizontally in the image). In Figure 4(b), seven groups of samples are found. Most temporal instances where a pedestrian changes direction imply a feet occlusion event that is, one foot is temporarily occluded by the rest of the body. Hence, it is assumed that direction changes occur between groups of continuous samples. A junction point $\boldsymbol{j}^l$ is computed between each consecutive group of samples $\mathcal{G}^l$ and $\mathcal{G}^{l-1}$. There are $N_l = N_{\mathcal{G}} + 1$ junction points, with special cases $\boldsymbol{j}^1 = \hat{\boldsymbol{s}}(t_b^1)$, the first projected mass center of the first group, and $\boldsymbol{j}^{N_l} = \hat{\boldsymbol{s}}(t_e^{N_l-1})$, the last projected mass center of the last group. Intermediate junction points $\boldsymbol{j}^l$, $l = 2, \dots, N_l - 1$ are computed as follows (see Figure 5):

$$j_x^l = \left(\hat{s}_x(t_b^l) + \hat{s}_x(t_e^{l-1})\right)/2 \qquad (2)$$

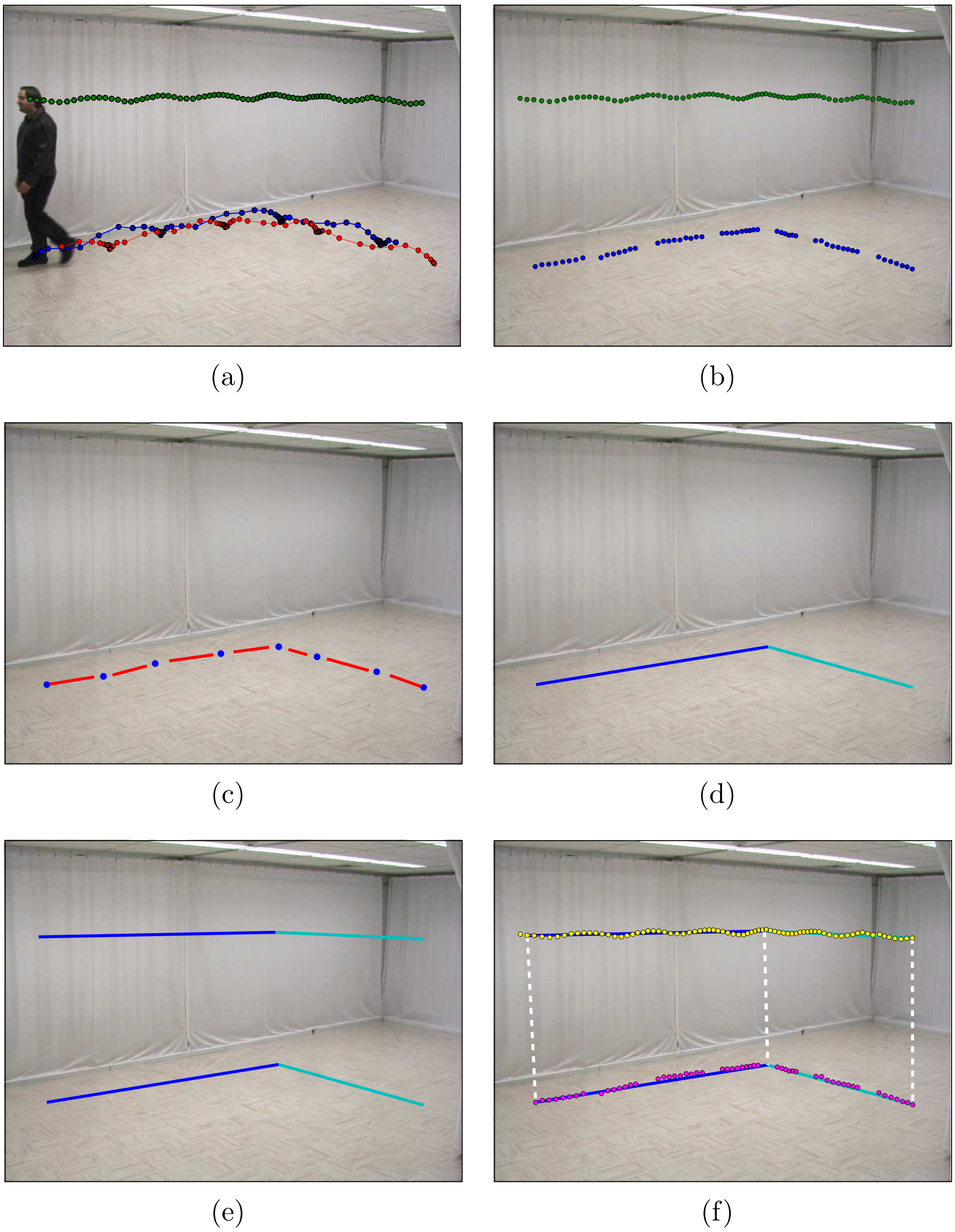

(a) (b) (c) (d) (e) (f)

Fig. 4. Walking trajectory segmentation

where j_x^l is the x coordinate of the junction point, $\hat{s}_x(t_b^l)$ is from the first point of $\mathcal{G}^l$ and $\hat{s}_x(t_e^{l-1})$ is from the last point of $\mathcal{G}^{l-1}$. As the y coordinates of the samples are noisier than the x coordinates, the computation of the y coordinate of the junction point is more involved. The fitted lines $\mathcal{L}^{l-1}$ and $\mathcal{L}^l$ are first used to extrapolate missing positions of projected mass center due to the feet occlusion event. The number of missing samples between the two groups is computed as:

$$\Delta t = t_b^l - t_e^{l-1} \tag{3}$$

To extrapolate the two lines, missed samples are first split between the two groups :

$$\Delta t^l = \lfloor (\Delta t - 1)/2 \rfloor \ , \ \ \Delta t^{l-1} = \lceil (\Delta t - 1)/2 \rceil . \tag{4}$$

The horizontal distance $\Delta x = \left| \hat{s}_x(t_b^l) - \hat{s}_x(t_e^{l-1}) \right|$ between point $\hat{\boldsymbol{s}}(t_e^{l-1})$ and point $\hat{\boldsymbol{s}}(t_b^l)$ is then split between the two groups according to the number of missing samples associated with each one:

$$\Delta x^l = \left(\Delta x \Delta t^l \right) / \Delta t \tag{5}$$

and

$$\Delta x^{l-1} = \left(\Delta x \Delta t^{l-1} \right) / \Delta t. \tag{6}$$

The junction y coordinate is then computed as :

$$j_y^l = \frac{\mathcal{L}_{\hat{s}}^l \left[\hat{s}_x(t_b^l - \Delta t^l) \right] + \mathcal{L}_{\hat{s}}^{l-1} \left[\hat{s}_x(t_e^{l-1} + \Delta t^{l-1}) \right]}{2} \tag{7}$$

where

$$\hat{s}_x(t_b^l - \Delta t^l) = \hat{s}_x(t_b^l) + a \Delta x^l, \tag{8}$$

$$\hat{s}_x(t_e^{l-1} + \Delta t^{l-1}) = \hat{s}_x(t_e^{l-1}) - a \Delta x^{l-1}, \tag{9}$$

and a is the sign of $\left\{ \hat{s}_x(t_e^{l-1}) - \hat{s}_x(t_b^l) \right\}$. $\mathcal{L}_{\hat{s}}^l [x]$ represents the y coordinate at x on line $\mathcal{L}_{\hat{s}}^l$. The y coordinate of the junction point is therefore at mid-distance from the two points extrapolated from the lines fitted to the groups $\mathcal{G}^l$ and $\mathcal{G}^{l-1}$. The x coordinate of the junction is the average of the last sample of $\mathcal{G}^{l-1}$ and the first sample of $\mathcal{G}^l$. Six computed intermediate junction points are displayed in Figure 4(c).

A classical iterative polyline fitting algorithm [26] is used next with ordered junction points acting as consecutive curve samples. The main steps of the algorithm are as follows:

(1) $\mathcal{I} \leftarrow \{1\}$
(2) $b = 1$, $e = N_l$
(3) Draw a line linking junction points $\boldsymbol{j}^b$ and $\boldsymbol{j}^e$.
(4) Compute the distances from junction points $\boldsymbol{j}^{b+1}$ to $\boldsymbol{j}^{e-1}$ to the line.

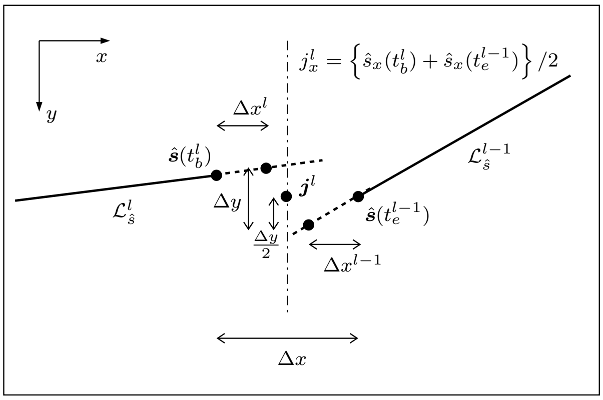

x
y
$j_x^l = \left\{\hat{s}_x(t_b^l) + \hat{s}_x(t_e^{l-1})\right\}/2$
Δx^l
$\hat{\boldsymbol{s}}(t_b^l)$
$\mathcal{L}_{\hat{s}}^{l-1}$
$\boldsymbol{j}^l$
Δy
$\mathcal{L}_{\hat{s}}^l$
$\hat{\boldsymbol{s}}(t_e^{l-1})$
$\frac{\Delta y}{2}$
Δx^{l-1}
Δx

Fig. 5. Computing junction points

(5) Determine the junction point $\boldsymbol{j}^d$ $(b < d < e)$ whose distance to the line is maximal.
(6) If that distance is above a predefined threshold T_d then $\mathcal{I} \leftarrow \mathcal{I} \cup \{d\}$ and repeat step 3 twice for $b = b$, $e = d$ and $b = d$, $e = e$.
(7) $\mathcal{I} \leftarrow \mathcal{I} \cup \{N_l\}$

Different results may be obtained according to the value selected for T_d. The purpose of the T_d threshold is to group togother trajectory segments that appear to have been performed in the same direction. However, one could set this threshold to 0 in order to get a trajectory segment for each pair of consecutive junction points. This would make the number of segment dependent only on the number of gait half-cycle, since at least one junction point is found for each gait half-cycle. A large value for T_d results in a small number of homographies but it may introduce distorsions in the normalized trajectories. In our experiments, a single value of $T_d = 8$ was selected empirically and used for all sequences.

The set $\mathcal{I} : \{i_1, i_2, \ldots, i_{N_\mathcal{I}}\}$ of $N_\mathcal{I}$ junction indices is obtained that correspond to positions where the walking trajectory is to be segmented. The walking trajectory is approximated by a polyline whose corners are the retained junction points. The latter are denoted as $\boldsymbol{j}_r^m = \boldsymbol{j}^{i_m}$, $m = 1, \ldots, N_m$ for $N_m = N_\mathcal{I}$. A frame number is associated with each retained junction point: $t_r^m = t_b^{i_m} - \Delta t^{i_m}$ for $m = 2, \ldots, N_m - 1$, and the special case $t_r^1 = t_b^1$ and $t_r^{N_m} = t_e^{N_l - 1}$. In Figure 4(d), the default threshold value produced three retained junction points whose indices are 1, 4, and 8.

Once the walking trajectory is segmented, the same frame indices are used to segment the head trajectory. Defining $\boldsymbol{h}(t)$ as the head position at time t, a straight-line segment $\mathcal{L}_h^m$ is fitted to the corresponding groups of head points. The obtained segments are then used to compute junction points $\boldsymbol{q}^m$ for the head trajectory :

$$q_x^m = \left\{ h_x(t_r^m) + h_x(t_r^m - 1) \right\} / 2, \tag{10}$$

$$q_y^m = \left\{ \boldsymbol{\mathcal{L}}_h^m \left[h_x(t_r^m) \right] + \boldsymbol{\mathcal{L}}_h^{m-1} \left[h_x(t_r^m - 1) \right] \right\} / 2, \tag{11}$$

where $\boldsymbol{\mathcal{L}}_h^m [x]$ represents the y coordinate at the x position on the line $\boldsymbol{\mathcal{L}}_h^m$. This is computed for $m = 2, \ldots, N_m - 1$ only, with special cases $\boldsymbol{q}^1 = \boldsymbol{h}(t_r^1)$ and $\boldsymbol{q}^{N_m} = \boldsymbol{h}(t_r^{N_m})$. As for the projected mass center, these junction points have a x coordinate at mid-distance from $h_x(t_r^m)$ and $h_x(t_r^m - 1)$, and a y coordinate at mid-distance from the two extrapolated y coordinates. The extrapolated y coordinates produce accurate junction points, assuming that the head trajectory is sinusoidal (the original y coordinates $h_y(t_r^m)$ and $h_y(t_r^m - 1)$ are not used). Finally, consecutive head junction points are linked to form a polyline approximating the head trajectory. Figure 4(e) presents the resulting approximated head trajectory. One can see in Figure 4(f) that the approximated trajectories fit well the original head and walking trajectories. Corresponding junction points are linked with dashed lines to show the two estimated original walking planes.

3.4 Computing Plane Parameters

The parameters of an original walking plane are computed using head junction points and two walking trajectory junction points. The different original walking planes are denoted Π^p for $p = 1, \ldots, N_p$, $N_p = N_m - 1$, and their four corners are defined as $\boldsymbol{\pi}_{\text{BB}}^p = \boldsymbol{j}_r^p$ (beginning bottom position), $\boldsymbol{\pi}_{\text{EB}}^p = \boldsymbol{j}_r^{p+1}$ (ending bottom position), $\boldsymbol{\pi}_{\text{BT}}^p = \boldsymbol{q}^p$ (beginning top position) and $\boldsymbol{\pi}_{\text{ET}}^p = \boldsymbol{q}^{p+1}$ (ending top position).

Once the four corners of the planes are known, planes are ready to be normalized. Figure 6 shows the first normalization step of a plane Π^p transformed into a plane $\hat{\Pi}^p$. Defining $d^p = \|\boldsymbol{\pi}_{\text{EB}}^p - \boldsymbol{\pi}_{\text{BB}}^p\|$ the length of the line segment in the approximated walking trajectory, the corners of the transformed plane are defined as :

$$\hat{\boldsymbol{\pi}}_{\text{BB}}^p = \left[\pi_{\text{MB},x}^p + uc^p d^p / 2 \ , \ \pi_{\text{MB},y}^p \right]^{\text{T}} \tag{12}$$

$$\hat{\boldsymbol{\pi}}_{\text{EB}}^p = \left[\pi_{\text{MB},x}^p - uc^p d^p / 2 \ , \ \pi_{\text{MB},y}^p \right]^{\text{T}} \tag{13}$$

$$\hat{\boldsymbol{\pi}}_{\text{BT}}^p = \left[\pi_{\text{MT},x}^p + uc^p d^p / 2 \ , \ \pi_{\text{MT},y}^p \right]^{\text{T}} \tag{14}$$

$$\hat{\boldsymbol{\pi}}_{\text{ET}}^p = \left[\pi_{\text{MT},x}^p - uc^p d^p / 2 \ , \ \pi_{\text{MT},y}^p \right]^{\text{T}} \tag{15}$$

where

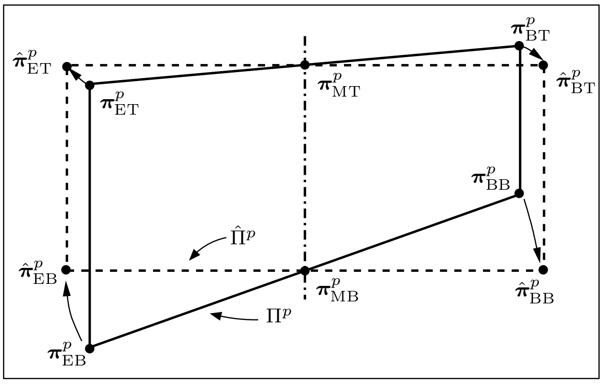

$\boldsymbol{\pi}^p_{\mathrm{BT}}$
$\hat{\boldsymbol{\pi}}^p_{\mathrm{ET}}$
$\boldsymbol{\pi}^p_{\mathrm{ET}}$
$\boldsymbol{\pi}^p_{\mathrm{MT}}$
$\hat{\boldsymbol{\pi}}^p_{\mathrm{BT}}$
$\boldsymbol{\pi}^p_{\mathrm{BB}}$
$\hat{\Pi}^p$
$\hat{\boldsymbol{\pi}}^p_{\mathrm{EB}}$
$\boldsymbol{\pi}^p_{\mathrm{MB}}$
$\hat{\boldsymbol{\pi}}^p_{\mathrm{BB}}$
Π^p
$\boldsymbol{\pi}^p_{\mathrm{EB}}$

Fig. 6. Plane normalization

$$\boldsymbol{\pi}^p_{\mathrm{MB}} = \left(\boldsymbol{\pi}^p_{\mathrm{EB}} + \boldsymbol{\pi}^p_{\mathrm{BB}}\right)/2 \ , \quad \boldsymbol{\pi}^p_{\mathrm{MT}} = \left(\boldsymbol{\pi}^p_{\mathrm{ET}} + \boldsymbol{\pi}^p_{\mathrm{BT}}\right)/2 \tag{16}$$

are the middle points of the top and bottom lines, respectively. Parameter c^p is the sign of $(\pi^p_{\mathrm{EB},x} - \pi^p_{\mathrm{BB},x})$ and u is a parameter taking value 1 or -1 which indicates the direction of the normalized body parts trajectories. Using $u = 1$, the trajectory will be from left to right, while for $u = -1$ the trajectory will be from right to left. Depending on the relative position of components $\pi^p_{\mathrm{EB},x}$ and $\pi^p_{\mathrm{BB},x}$, the relative x positions of the plane corners will be switched if $\pi^p_{\mathrm{EB},x} < \pi^p_{\mathrm{BB},x}$ and $u = 1$, or if $\pi^p_{\mathrm{EB},x} > \pi^p_{\mathrm{BB},x}$ and $u = -1$. This switching is necessary for body part trajectories normalization since it makes normalized trajectories appear as if the person had walked along a single direction, even when the walking trajectory includes a change in x direction. As shown in Figure 6, a new plane is formed by transforming the bottom and top borders of the original walking plane so they become parallel to the image frame. The bottom edge keeps its length while the top edge also has the same length. As a result, side edges become parallel to the image borders too.

Scaling and shifting is applied to the computed normalized planes in order to obtain adequate normalized body part trajectories. After scaling and shifting, all normalized planes have the same height as is the case for fronto-parallel views, and a width proportional to the elapsed time. They are also connected. The initial height H^p and the width W^p of each normalized plane $\hat{\Pi}^p$ are computed as follows:

$$H^p = \|\hat{\boldsymbol{\pi}}^p_{\mathrm{BT}} - \hat{\boldsymbol{\pi}}^p_{\mathrm{BB}}\| \ , \quad W^p = \|\hat{\boldsymbol{\pi}}^p_{\mathrm{EB}} - \hat{\boldsymbol{\pi}}^p_{\mathrm{BB}}\|. \tag{17}$$

A ratio R^p is then computed for each normalized plane. It indicates the relationship between the plane's width and its number of frames. This ratio is used to scale the width:

$$R^p = \frac{W^p}{(t_r^{p+1} - 1) - t_r^p} = \frac{W^p}{\Delta t^p} \tag{18}$$

where t_r^p represents the time associated with the junction point $\boldsymbol{j}_r^p$. Scaling uses a fixed beginning bottom corner $\hat{\pi}_{\mathrm{BB}}^p$. It was chosen to scale the height to H_{median} and use the maximum ratio of all normalized planes R_{max} :

$$\begin{aligned} H_{\mathrm{median}} &= \underset{p=1,\ldots,N_p}{\operatorname{median}} H^p, \qquad (19)\\ R_{\mathrm{max}} &= \max_{p=1,\ldots,N_p} R^p. \qquad (20) \end{aligned}$$

The obtained width of a normalized plane is:

$$W_{\mathrm{scaled}}^p = R_{\mathrm{max}} \Delta t^p. \tag{21}$$

Setting the width using the same R_{max} ratio for all planes implies that the walking velocity is assumed constant across all planes. The positions of the corners are finally computed using the new height and width ($\hat{\boldsymbol{\pi}}_{\mathrm{BB}}^p$ remains at the same position):

$$\begin{aligned} \hat{\boldsymbol{\pi}}_{\mathrm{EB}}^p &= \left[\hat{\boldsymbol{\pi}}_{\mathrm{BB},\,x}^p + uW_{\mathrm{scaled}}^p \,,\, \hat{\boldsymbol{\pi}}_{\mathrm{BB},\,y}^p\right]^{\mathrm{T}} \qquad (22)\\ \hat{\boldsymbol{\pi}}_{\mathrm{BT}}^p &= \left[\hat{\boldsymbol{\pi}}_{\mathrm{BB},\,x}^p \,,\, \hat{\boldsymbol{\pi}}_{\mathrm{BB},\,y}^p + H_{\mathrm{median}}\right]^{\mathrm{T}} \qquad (23)\\ \hat{\boldsymbol{\pi}}_{\mathrm{ET}}^p &= \left[\hat{\boldsymbol{\pi}}_{\mathrm{BB},\,x}^p + uW_{\mathrm{scaled}}^p \,,\, \hat{\boldsymbol{\pi}}_{\mathrm{BB},\,y}^p + H_{\mathrm{median}}\right]^{\mathrm{T}} \qquad (24) \end{aligned}$$

The last plane normalization step is the shifting of each plane such that the beginning corners of plane $\hat{\Pi}^{p+1}$ are at the same position as the ending corners of plane $\hat{\Pi}^p$. Such a shifting is necessary in order to obtain continuous spatiotemporal trajectories. If $\boldsymbol{z}^p = \hat{\boldsymbol{\pi}}_{\mathrm{BB}}^p - \overline{\boldsymbol{\pi}}_{\mathrm{EB}}^{p-1}$ represents the amount of shift between the beginning bottom corner $\hat{\boldsymbol{\pi}}_{\mathrm{BB}}^p$ and the new ending corner $\overline{\boldsymbol{\pi}}_{\mathrm{EB}}^{p-1}$, then the new corners of the normalized plane $\overline{\Pi}^p$ are:

$$\overline{\boldsymbol{\pi}}_{\mathcal{C}}^p = \hat{\boldsymbol{\pi}}_{\mathcal{C}}^p - z^p \tag{25}$$

for $\mathcal{C} : \{\mathrm{BB}\,,\, \mathrm{EB}\,,\, \mathrm{BT}\,,\, \mathrm{ET}\}$ and $p = 2, \ldots, N_p$. For the special case $p = 1$, $\overline{\boldsymbol{\pi}}_{\mathcal{C}}^1 = \hat{\boldsymbol{\pi}}_{\mathcal{C}}^1$.

Once the normalized planes are obtained, it is possible to compute a homography matrix $\boldsymbol{E}^p$ by constructing an 8-equation linear system using the correspondences between the corners of the original and normalized walking planes Π^p and $\overline{\Pi}^p$. The linear equation systems are solved using the Gauss-Jordan method.

The homography matrix associated with each normalized plane $\overline{\Pi}^p$ may be applied to body part trajectories in order to retrieve their normalized trajectories. If the position at time t of a body part $\mathcal{R}$ is defined as $\boldsymbol{b}^{\mathcal{R}}(t)$, then the normalized trajectory $\overline{\boldsymbol{b}}^{\mathcal{R}}(t)$ is computed as

$$\begin{bmatrix} \alpha \overline{\boldsymbol{b}}^{\mathcal{R}}(t) \\ \alpha \end{bmatrix} = \boldsymbol{E}^p \begin{bmatrix} \boldsymbol{b}^{\mathcal{R}}(t) \\ 1 \end{bmatrix}, \; t_r^p \leq t < t_r^{p+1} \tag{26}$$

where $p = 1, \ldots, N_p$ and α is a scale factor. The homography matrix $\boldsymbol{E}^p$ of the normalized plane $\overline{\Pi}^p$ is only used to normalize trajectory positions for $t_r^p \leq t < t_r^{p+1}$, since that time interval corresponds to the original walking plane Π^p.

4 Experimental Evaluation

The purpose of the normalization process is to obtain body part trajectories that appear to have been observed from a fronto-parallel viewpoint. The effectiveness of the normalization algorithms can be assessed by comparing the body part trajectories obtained from different views of the same walk to the reference view. The comparison is performed on both non-normalized (raw) and normalized body part trajectories in order to evaluate the improvement achieved with the normalization process.

The following section presents the trajectories comparison process and the measures that are used for the experimental evaluation of the proposed normalization algorithms.

4.1 Least-Squares Alignment of a Pair of Trajectories

In order to compare the body part trajectories from two different views, it is possible to find the best trajectories alignment in a least-squares sense between corresponding body parts. Considering a body part trajectory defined in the time slice common to both views, it is possible to use an optimal alignment algorithm such as the one described in [27]. This algorithm is used to obtain the optimal rotation and translation that best aligns two 3-D point lists of the same size in a least-squares sense. In the present case, this algorithm is used with two lists (trajectories) of 2-D points. If the lists are denoted $\boldsymbol{P} : \{\boldsymbol{p}_i\}$ and $\boldsymbol{Q} : \{\boldsymbol{q}_i\}$, $i = 1, \ldots, N$, the rotation $\boldsymbol{R}$ is defined as

$$\boldsymbol{R} = \boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^{\mathrm{T}}, \tag{27}$$

where $\boldsymbol{V}$ and $\boldsymbol{U}$ are matrices from the Singular Value Decomposition $\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\mathrm{T}}$ of the covariance matrix $\boldsymbol{C}$ defined as

$$\boldsymbol{C} = \sum_{i=1}^{N} (\boldsymbol{p}_i - \boldsymbol{\mu}_p)(\boldsymbol{q}_i - \boldsymbol{\mu}_q)^{\mathrm{T}}, \tag{28}$$

with $\boldsymbol{\mu}_p$ and $\boldsymbol{\mu}_q$ the means of the point lists $\boldsymbol{P}$ and $\boldsymbol{Q}$. If $\det(\boldsymbol{C}) \geq 0$, $\boldsymbol{S} = \mathrm{diag}(1, 1)$, the identity matrix. If $\det(\boldsymbol{C}) < 0$, $\boldsymbol{S} = \mathrm{diag}(1, -1)$, a matrix with ones on the diagonal except for the last element, which is equal to -1. The optimal translation vector $\boldsymbol{t}$ is given by

$$\boldsymbol{t} = \boldsymbol{\mu}_q - \boldsymbol{R}\boldsymbol{\mu}_p. \tag{29}$$

Once the matrix $\boldsymbol{R}$ and the translation vector $\boldsymbol{t}$ are determined, the best aligment of the point list $\boldsymbol{P}$ with respect to the point list $\boldsymbol{Q}$ is defined as a new point list $\boldsymbol{P}' : \{\boldsymbol{p}'_i\}$, where $\boldsymbol{p}'_i = \boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{t}$ and $i = 1, \ldots, N$.

4.2 Relative Root Mean Squared Distance

A distance between two aligned trajectories is computed in order to assess the quality of their aligment. A relative root mean square distance (RRMSD) is used in our evaluation:

$$\mathrm{RRMSD} = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{p}_i - \boldsymbol{q}_i\|^2}}{\mathrm{Diag}(\boldsymbol{P} \cup \boldsymbol{Q})} \tag{30}$$

where $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$ are points from the trajectories $\boldsymbol{P}$ and $\boldsymbol{Q}$, respectively. In order to be independent of the specific spatial scales of the normalized and the raw trajectories, $\mathrm{Diag}(\boldsymbol{P} \cup \boldsymbol{Q})$ is introduced in the calculation of RRMSD.

The diagonal length of the bounding box englobing both trajectories is chosen :

$$\mathrm{Diag}(\boldsymbol{P} \cup \boldsymbol{Q}) = \sqrt{\left(\max_j(r_{xj}) - \min_j(r_{xj})\right)^2 + \left(\max_j(r_{yj}) - \min_j(r_{yj})\right)^2} \quad (31)$$

where r_{xj} and r_{yj} are the x and y component of the point $\boldsymbol{r}_j = [r_{xj}, r_{yj}]^{\mathrm{T}}$, and $\boldsymbol{r}_j$ is a point from the union of the trajectories $\boldsymbol{P} \cup \boldsymbol{Q} : \{\boldsymbol{r}_j\}$, with $j = 1, \ldots, 2N$.

5 Experimental Results

Two experiments are presented in this section. The first experiment concerns subjects walking along a straight line while being observed simultaneously by four cameras (Section 5.1). Trajectory alignments are compared. The second experiment consists in one subject performing three walks in which the walking direction is changing (Section 5.2). For both experiments, the video sequences are preprocessed as explained in Section 2 in order to extract head and feet trajectories.

5.1 Trajectories Alignment Comparison

In this experiment, ten volunteers are asked to walk back and forth on a straight line in front of four roughly time-synchronized color cameras (30 frames per second, resolution of 640 × 480, synchronization of ±3 frames, indoor environment). Cameras are positioned such that their optical axis intercepts the walking trajectory at different pan angles : 90°, 75°, 60°, and 45°. The cameras have low tilt values (between 10 and 15 degrees) and zero roll. One should note that such a camera positioning scheme is consistent with most of the datasets used in related studies on gait analysis and recognition, which mostly use pan-type camera movements for varying the viewpoint. A detailed description of these datasets is available in [28]. The minimal distance between the subject and the cameras is approximately 2.5m, and the maximal distance is approximately 8.5m. Figure 7 shows the setup used for the acquisition process. This setup is appropriate to test the performances of the normalization algorithm since it provides four views of the same walk, including the fronto-parallel view (90°) to which other views are to be compared. A subject first appears at right of the image and then disappears at left (interval 1). He reappears one or two seconds later at left of the image and then disappears at right (interval 2). This provided four video sequences

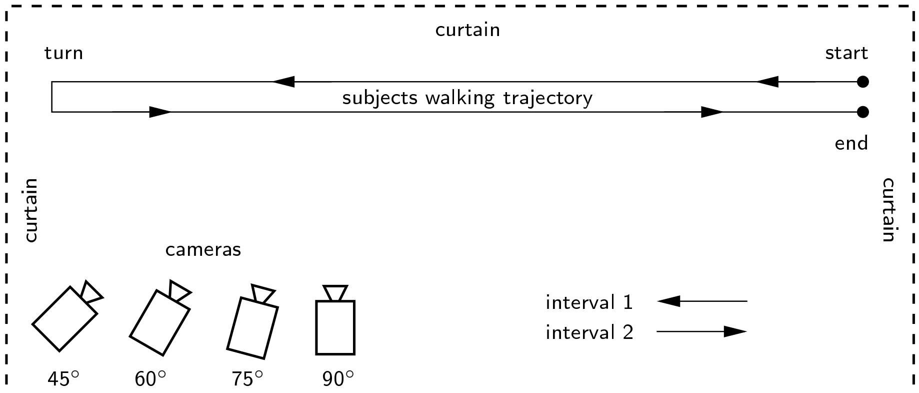


Fig. 7. Acquisition setup

(four views) for each subject. Preprocessings of the video sequences yields 80 head and feet trajectories (10 subjects × 4 views × 2 intervals). Depending on the view and the subject, each sequence interval contains from one to three visible gait cycles.

Normalized body part trajectories would offer a better basis than raw trajectories for modeling and comparing gaits as long as they appear as if observed from a fronto-parallel view. The performance of the normalization process, as providing this transformation to the fronto-parallel view, is evaluated.

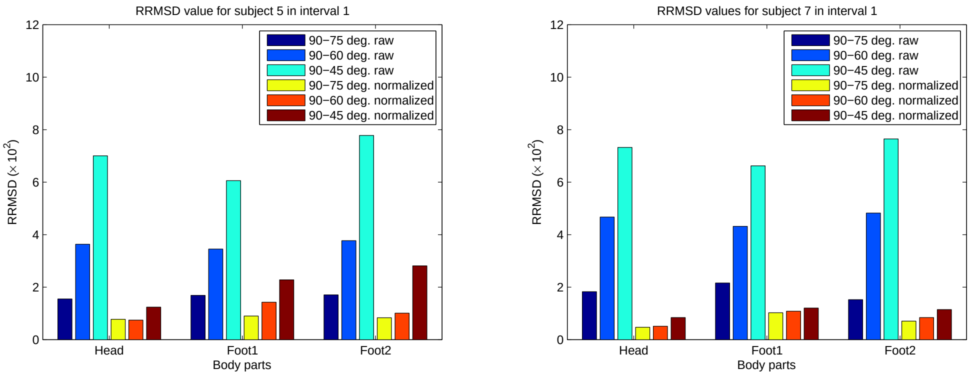
In order to properly evaluate the normalization process, time-synchronized cameras are needed in this experiment. Imperfect time-synchronization between cameras was refined interactively by selecting key frames in each view. Such time-synchronization is neither possible nor needed when gait models are built and compared from trajectories extracted from different cameras at different times. In this experiment, trajectories alignment between two synchronized views is performed for the time interval they share. That is, only the parts of the trajectories that are actually observed at the same time in both views is considered. Feet trajectories are manually labeled “left” or “right” so that a foot aligment between two different views is correctly performed. This could be automatically detected for a gait modelization algorithm that requires the knowledge of left and right foot. Normalized body part trajectories are scaled vertically and horizontally before being aligned so that they have the same normalized plane heights and R ratios (set to the values of one or the other trajectory). This is because plane heights and R ratios were set to the median plane heights and the maximum R ratio, which are not the same in different views (see Section 3.4). It is possible to do the vertical and horizontal scaling by assuming the same height and walking speed since it is known that the trajectories came from the same person. The same operation cannot be applied to raw trajectories since their planes may have different heights and widths that are independent of their duration.

Table 1 presents results of the body part trajectory aligments for three combi-

Subject	Interval	Type	90deg - 75deg			90deg - 60deg			90deg - 45deg		
			Head	Foot1	Foot2	Head	Foot1	Foot2	Head	Foot1	Foot2
1	1	raw	1.54	1.55	1.66	4.25	3.77	4.15	7.32	8.70	6.73
		norm	0.18	0.43	0.62	0.66	0.67	0.77	0.99	2.22	2.26
	2	raw	1.88	1.76	1.94	4.30	3.87	3.95	7.74	7.46	6.80
		norm	0.55	0.77	0.67	1.59	2.39	1.88	0.91	2.18	1.68
2	1	raw	1.34	1.28	1.57	4.07	3.86	4.05	7.21	5.77	8.58
		norm	0.80	1.12	1.09	0.44	1.39	1.29	1.21	2.23	1.60
	2	raw	1.79	1.65	1.74	4.72	3.33	5.56	7.72	6.33	8.42
		norm	0.52	1.03	0.95	0.95	1.86	1.80	1.10	1.62	2.81
3	1	raw	1.82	2.06	2.74	4.03	3.86	4.69	7.63	6.52	9.50
		norm	0.18	0.96	1.48	0.36	0.83	1.01	1.16	1.70	2.52
	2	raw	1.65	1.78	1.76	4.43	2.92	5.09	7.00	4.86	8.10
		norm	0.19	1.06	1.25	0.81	0.67	0.66	0.82	1.32	1.42
4	1	raw	1.80	1.52	2.37	4.01	3.37	4.87	7.90	5.62	10.25
		norm	0.16	0.54	0.59	0.78	1.42	1.03	0.85	2.01	2.37
	2	raw	1.60	1.13	1.63	4.48	3.16	5.00	7.04	5.23	7.88
		norm	0.33	0.56	0.84	0.25	1.21	0.70	0.64	1.39	1.74
5	1	raw	1.55	1.69	1.71	3.64	3.45	3.77	7.01	6.06	7.78
		norm	0.78	0.90	0.84	0.75	1.43	1.01	1.24	2.28	2.81
	2	raw	1.48	1.42	1.94	4.64	5.11	4.08	7.26	8.12	6.11
		norm	0.19	0.52	1.17	1.06	1.67	2.03	0.74	1.40	1.74
6	1	raw	1.89	2.18	1.65	4.83	4.34	4.95	7.70	6.94	8.10
		norm	0.37	1.19	0.67	1.30	1.42	1.34	1.48	1.80	1.91
	2	raw	1.92	2.06	1.51	4.30	4.48	3.37	8.08	7.59	8.34
		norm	0.26	0.50	0.66	0.70	1.03	1.16	0.90	0.97	1.87
7	1	raw	1.83	2.16	1.52	4.67	4.32	4.82	7.32	6.62	7.65
		norm	0.47	1.02	0.71	0.51	1.08	0.84	0.84	1.21	1.15
	2	raw	1.74	1.91	1.43	4.47	4.45	3.12	7.69	7.14	6.95
		norm	0.18	0.67	0.76	0.53	0.66	0.87	0.91	0.81	1.25
8	1	raw	1.67	1.83	1.45	4.48	4.07	4.62	7.19	6.53	7.54
		norm	0.45	0.74	0.51	0.59	1.06	1.33	0.65	1.92	1.97
	2	raw	1.68	1.40	1.56	4.24	3.56	3.50	7.69	5.72	9.03
		norm	0.13	0.69	0.53	0.27	0.84	1.17	0.32	1.22	1.89
9	1	raw	1.72	1.44	1.83	4.44	3.37	5.47	6.93	5.24	8.37
		norm	0.78	0.89	1.00	0.87	1.10	1.42	1.10	1.54	1.53
	2	raw	1.48	1.04	1.48	4.02	2.94	4.64	6.49	4.93	7.33
		norm	0.54	0.46	0.58	0.82	1.38	0.88	0.88	1.78	1.28
10	1	raw	1.54	1.47	1.80	4.28	3.16	4.99	7.30	6.56	7.53
		norm	0.75	1.13	1.19	1.12	1.56	2.02	1.02	1.72	1.87
	2	raw	2.05	2.02	2.06	5.18	5.67	4.33	8.02	8.41	6.94
		norm	0.21	0.52	0.49	0.74	0.99	1.45	1.04	1.02	1.35

Table 1
RRMSD values ($\times 10^2$)

nations of views : $90^\circ - 75^\circ$, $90^\circ - 60^\circ$, and $90^\circ - 45^\circ$. Alignment was performed for non-normalized (raw) and normalized (norm) body part trajectories in each view. RRMSD values of the resulting alignment are presented for each subject, interval and body part. Table 2 presents some alignment statistics, all subjects and intervals mixed. It is possible to see from those results that trajectories from 75°, 60°, and 45° views are closer to the trajectories from the fronto-parallel view (90°) after normalization. The importance of the tra-



(a) Subject 5, Interval 1

(b) Subject 7, Interval 1

Fig. 8. RRMSD values for the interval of two subjects

Statistic	Type	90deg - 75deg			90deg - 60deg			90deg - 45deg		
		Head	Foot1	Foot2	Head	Foot1	Foot2	Head	Foot1	Foot2
mean	raw	1.70	1.67	1.77	4.37	3.85	4.45	7.41	6.52	7.90
	norm	0.40	0.78	0.83	0.76	1.23	1.23	0.94	1.62	1.85
median	raw	1.70	1.67	1.68	4.37	3.82	4.63	7.32	6.52	7.83
	norm	0.35	0.75	0.73	0.74	1.16	1.16	0.91	1.66	1.81
std	raw	0.18	0.34	0.32	0.34	0.72	0.68	0.41	1.13	1.00
	norm	0.24	0.25	0.28	0.34	0.44	0.43	0.25	0.44	0.49
min	raw	1.34	1.04	1.43	3.64	2.92	3.12	6.49	4.86	6.11
	norm	0.13	0.43	0.49	0.25	0.66	0.66	0.32	0.81	1.15
max	raw	2.05	2.18	2.74	5.18	5.67	5.56	8.08	8.70	10.25
	norm	0.80	1.19	1.48	1.59	2.39	2.03	1.48	2.28	2.81

Table 2
Statistics on RRMSD ($\times 10^2$)

jectory normalization process is clearer as the difference in angle between the compared views is greater. This can be observed in Figure 8 where RRMSD values are plotted for subjects 5 and 7. In Figure 9 and 10, aligned trajectories of the same two subjects are shown. Trajectory aligment is more difficult when there is noise in the body part trajectories, as one may see in Figure 9 for the raw and normalized feet trajectories. This noise comes from the tracking algorithm whose performance may vary according to the viewpoint. The person's silhouette may indeed be noisier in some views. Moreover, feet occlusions become more difficult to handle as the view is departing from the fronto-parallel view. These factors explain why the observed RRMSD values increase as the angle between compared views is getting higher. The increase of RRMSD values for the normalized trajectories is however much smaller than the increase observed for raw trajectories.

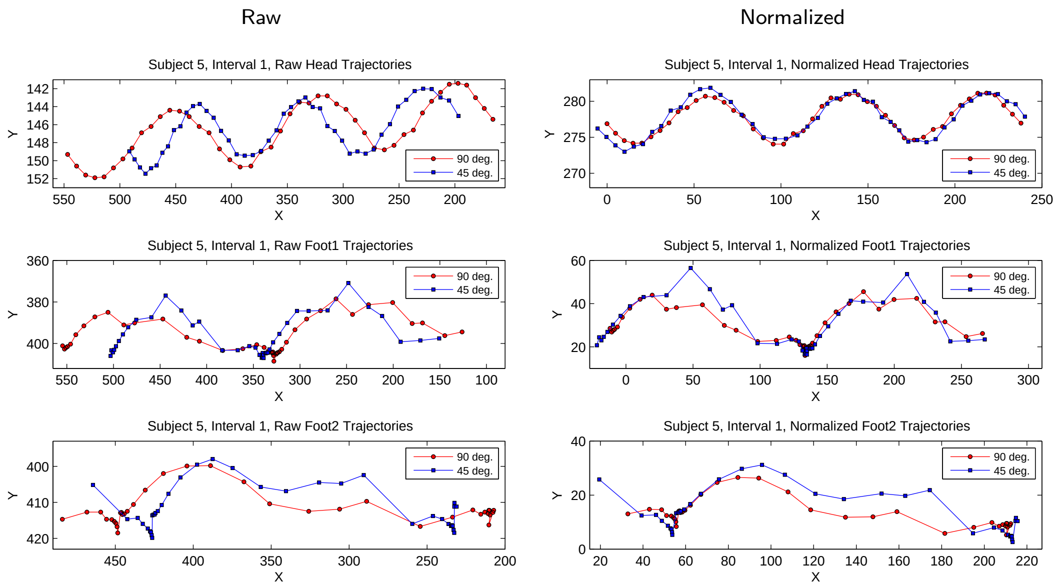


Fig. 9. Raw and normalized trajectory alignments of Subject 5, 90° – 45°

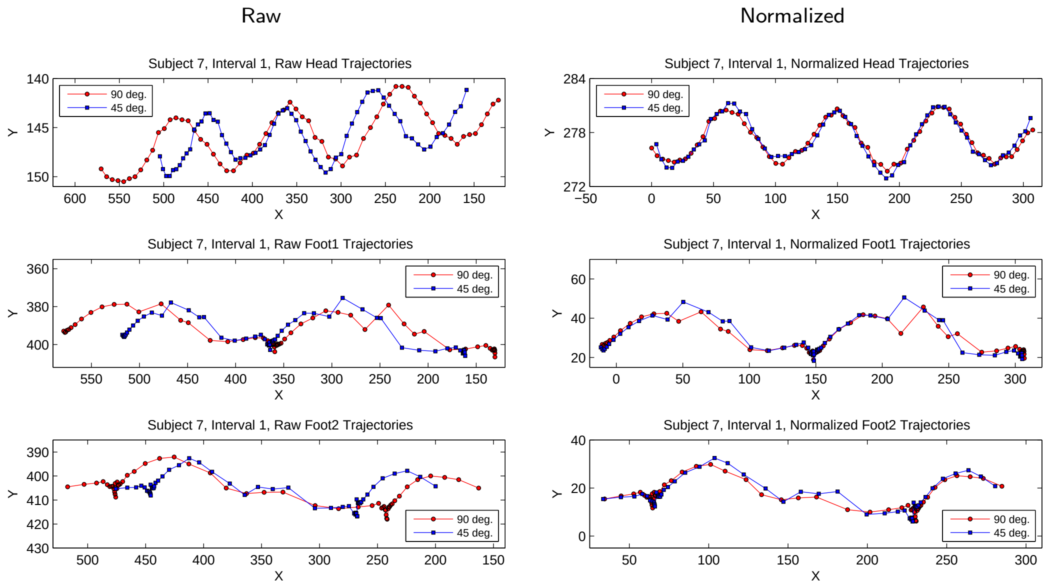
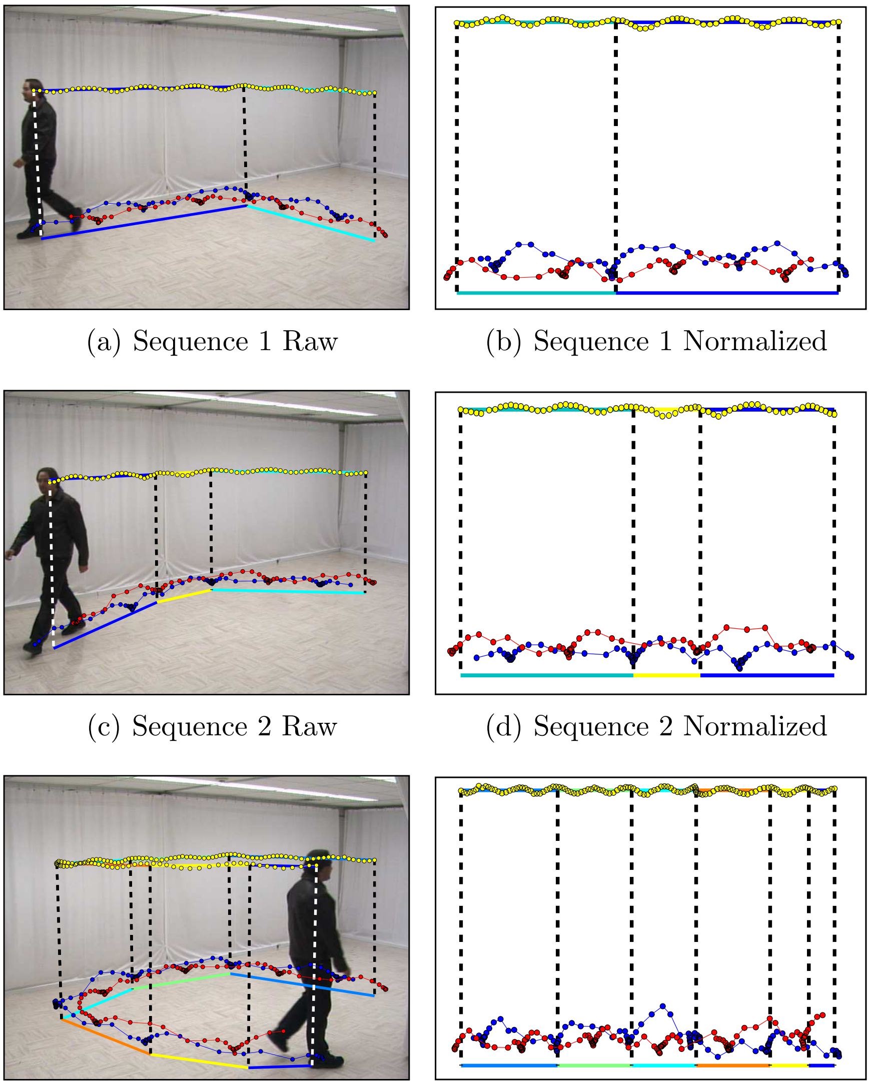


Fig. 10. Raw and normalized trajectory alignments of Subject 7, 90° – 45°

5.2 Changes in Walking Direction

In this experiment, three different walks of a single subject making smooth and/or sudden changes in his walking direction are processed. A single camera is used (15 frames per second, resolution of 640 × 480, indoor environment). This experiment permits a qualitative evaluation of the behaviour of the normalization algorithm when there are changes in the walking direction.

Figure 11 shows the result of view normalization on the three sequences. The



(a) Sequence 1 Raw (b) Sequence 1 Normalized

(c) Sequence 2 Raw (d) Sequence 2 Normalized

(e) Sequence 3 Raw (f) Sequence 3 Normalized

Fig. 11. Trajectory normalization for the second experiment.

changes in trajectory direction for each sequence are as follows:

(1) One sudden direction change
(2) One smooth direction change
(3) Two sudden direction changes, and one smooth direction change

The normalized trajectories look like trajectories obtained from a fronto-parallel viewpoint (side view), which is assumed optimal for gait modeling and identification [29]. In Figure 11(f), normalized feet and head trajectories are slightly deformed around the junction between the third and fourth planes. This is due to both the slowdown and the sudden change in walking direction. As a result, the feet locally violate the basic planar motion and constant speed assumptions. Fortunately, only a small part of the trajectory is affected since normal walk is mostly straight, apart from the occasional changes of direction.

One should note that the applicable range of views for the proposed approach is more limited by the preprocessing step (tracking) than by the view normalization algorithm itself. However, the proposed approach for view normalization is compatible with any other tracking algorithms that extract spatiotemporal trajectories of body parts, including optical motion capture technology that would not be affected at all by the viewpoint. Details follow below.

The proposed view normalization method is based on a homography, thus it works for all views except the one at 0 degrees, where the four points needed for the homography computation become collinear. This view corresponds to the subject walking towards (or away) from the camera along the optical axis of the camera. Views near 0 degrees are outside the applicable range, too, since they would result in an ill-conditioned homography matrix.

The tracking algorithm that was used in this paper to generate spatiotemporal trajectories has a more limited range of views. This is because tracking handles temporary feet self-occlusion for views where the duration of self-occlusion intervals is not longer than the duration of time intervals where both feet are visible. The approximate applicable range of views for tracking is $[0, 10] \bigcup [30, 90$ degrees$]$.

Experiments that provide the reader with information about the applicable range of views are shown in Figure 11 of the manuscript. In Figure 11(e) and 11(f) the subject walks on a quasi-circular path, which allows for observations on almost all views from 0 to 90 degrees.

5.3 Computational speed

In order to assess the compatibility of the proposed view normalization approach with real-time surveillance systems, the computational time necessary for processing normalized trajectories from video sequences was evaluated. The experiment involved 40 video sequences with statistics shown in Table 3; the content of these video sequences was previosuly described in 5.1. The frame resolution is 640×480 for all sequences. All experiments were performed on a computer with two *Dual Core AMD Opteron*^{TM} at 1.81 GHz.

The typical sequence of processes involved in generating view normalized trajectories consists in background subtraction followed by the generation of the raw trajectories (tracking) for a given time interval and then by the view normalization itself. In a surveillance context, real-time processing of video sequences means that background subtraction and generation of raw trajectories keep pace with video acquisition. Besides, normalization of trajectories,

	Length (frames)	Silhouette Size (pixels)
Mean	200	13696
Standard Deviation	62	4119
Median	179	14182
Mininum	119	3464
Maximum	352	23362

Table 3
Statistics on the number of frames and the size of the silhouettes for 40 sample video sequences with a total of 7993 frames.

gait modelling and gait matching need to be obtained fast enough, for instance in a matter of at most a few seconds, so that any problematic gait may be signaled appropriately. Timing data is provided here for tracking and view normalization algorithms. It shows that pedestrian tracking keeps largely pace with typical acquisition speeds (15 to 30 frames per second) and view normalization takes a fraction of a second. One should note that fast, real-time background subtraction algorithms such as optimizations of the Mixture of Gaussians [30] [31] can provide the silhouette input for tracking.

Since tracking is performed on a frame-by-frame basis, its computational speed is measured in miliseconds per frame. The data shown in Table 4 shows that the tracking approach used as preprocessing step for this paper is extremely fast (3.52 milliseconds/frame in average, that is 284 frames per second).

View normalization is performed on frame sequences which correspond to time intervals where the walking direction stays the same. Thus, the time is measured in seconds per sequence (with statistics on the length of the sequence given in Table 3). The average time needed to process a sequence of an average length of 200 frames is 0.2155 seconds, which would result in a processing speed of 1.0775 miliseconds per frame. Since the view normalization process is not performed on a frame-by-frame basis, we prefer reporting its computational performance in seconds per video sequence (See Table 5).

It is worth mentioning that, since view normalization needs the accumulation of information about all frames in a walking sequence performed along the same direction, background subtraction and preprocessing can be done in parallel with view normalization. One may conclude that the data shown in Tables 3, 4, and 5 supports the compatibility of the proposed method with the requirements of real-time surveillance systems.

	Time (ms/frame)
Mean	3.52
Standard Deviation	1.44
Median	3.41
Minimum	0.70
Maximum	21.69

Table 4
Performance of the tracking algorithm. Statistics were computed for frames of the 40 video sequences that were processed by the tracking algorithm (complete visible silhouette in the images). The tracking algorithm was implemented in *C++* using the *OpenCV* library (non-parallelized code).

	Time (sec/video sequence)
Mean	0.2155
Standard Deviation	0.0423
Median	0.2062
Minimum	0.1457
Maximum	0.3264

Table 5
Performance of the normalization algorithm. Times were computed for each of the 40 video sequences. The normalization algorithm was implemented in $MATLAB^{TM}$.

6 Conclusion

In this paper, an approach for normalizing body part trajectories was presented. The normalization process consists in the computation of a piecewise-straight walking trajectory, and a corresponding sequence of walking planes. A homography computation aligns the edges of each walking plane with the image edges. Each computed homography transforms the body part trajectories within the time interval of its corresponding walking plane.

The proposed approach is promising since it has direct applications to gait-based modeling and identification, which performs significantly better from a fronto-parallel (side) view. As validated experimentally, the normalized trajectories of head and feet from different views are well aligned with real fronto-parallel view trajectories. To the best of our knowledge, this is the first view-normalizing method proposed in the literature that is applicable to real-time gait-based identification in a surveillance context.

Ongoing work focuses on testing the proposed approach on trajectories of additional body parts (hands, knees etc.) involved in human walk. More tests are to be performed on trajectories with changes in walking direction. Gait-based identification will be perfomed by extracting gait characterisics from normalized body part trajectories.

Acknowledgements

This work is supported by Natural Sciences and Engineering Research Council of Canada, Fonds Québécois de la Recherche sur la Nature et les Technologies, and by Precarn inc.

References

[1] M. P. Murray, Gait as a total pattern of movement, American Journal of Physical Medicine 13 (1967) 290–332.

[2] R. Collins, A. Lipton, T. Kanade, Introduction to the special section on video surveillance, Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 745–746.

[3] R. Cutler, L. Davis, Robust real-time periodic motion detection, analysis, and applications, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 781–796.

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

[5] J. Han, B. Bhanu, Performance prediction for individual recognition by gait, Pattern Recognition Letters (2005) 615–624.

[6] R. Urtasun, P. Fua, 3d tracking for gait characterization and recognition, in: Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004, pp. 17–22.

[7] R. Zhang, C. Vogler, D. Metaxas, Human gait recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Vol. 1, 2004, pp. 18–25.

[8] J.-H. Yoo, M. S. Nixon, Markerless human gait analysis via image sequences, in: Proceedings of International Society of Biomechanics XIXth Congress, 2003.

[9] S. Yu, L. Wang, W. Hu, T. Tan, Gait analysis for human identification in frequency domain, in: Proceedings of the Third International Conference on Image and Graphics, 2004, pp. 282–285.

[10] C. BenAbdelkader, R. Cutler, L. Davis, View-invariant estimation of height and stride for gait recognition, in: Proceedings of the International ECCV Workshop on Biometric Authentication, Vol. LNCS 2359, 2002, pp. 155 – 167.

[11] N. M. Spencer, J. Carter, Towards pose invariant gait reconstruction, in: Proceedings of the IEEE International Conference on Image Processing, Vol. 3, 2005, pp. 261–264.

[12] A. Kale, A. K. R. Chowdhury, R. Chellappa, Towards a view invariant gait recognition algorithm, in: Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance, 2003, pp. 143–150.

[13] A. Y. Johnson, A. F. Bobick, A Multi-view Method for Gait Recognition Using Static Body Parameters, Vol. 2091 of LNCS, Springer, 2001, pp. 301–311.

[14] Y. Makihara, R. Sagawa, Y. Mukaigawa, T. Echigo, Y. Yagi, Adaptation to walking direction changes for gait identification, in: Proceedings of the 18th International Conference on Pattern Recognition, Vol. 2, 2006, pp. 96–99.

[15] M. Hild, Estimation of 3d motion trajectory and velocity from monocular image sequences in the context of human gait recognition, in: Proceedings of the 17th International Conference on Pattern Recognition, Vol. 4, 2004, pp. 231– 235.

[16] X. Han, J. Liu, L. Li, Z. Wang, Gait recognition considering directions of walking, in: Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems, 2006, pp. 1–5.

[17] A. Tyagi, J. W. Davis, M. Keck, Multiview fusion for canonical view generation based on homography constraints, in: Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks, ACM Press, Santa Barbara, California, USA, 2006, pp. 61–70.

[18] C.-S. Lee, A. Elgammal, Towards Scalable View-Invariant Gait Recognition: Multilinear Analysis for Gait, Vol. 3546 of LNCS, Springer, 2005, pp. 395–405.

[19] F. Cuzzolin, Using bilinear models for view-invariant action and identity recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 1701– 1708.

[20] A. B. Albu, D. Laurendeau, S. Comtois, D. Ouellet, P. Hébert, A. Zaccarin, M. Parizeau, R. Bergevin, X. Maldague, R. Drouin, S. Drouin, N. Martel-Brisson, F. Jean, H. Torresan, L. Gagnon, F. Laliberté, Monnet: Monitoring pedestrians with a network of loosely-coupled cameras, in: Proceedings of the IEEE International Conference on Pattern Recognition, Hong Kong, China, 2006.

[21] A. Bissacco, A. Chiuso, Y. Ma, S. Soatto, Recognition of human gaits, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001, pp. II:52–57.

[22] R. Tanawongsuwan, A. Bobick, Gait recognition from time-normalized joint-angle trajectories in the walking plane, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001, pp. II:726–731.

[23] G. Johansson, Visual perception of biological motion and a model for its analysis, Percept. Psychophys 14:2 (1973) 201–211.

[24] F. Jean, R. Bergevin, A. B. Albu, Computing view-normalized body parts trajectories, in: Proceedings of the Fourth Canadian Conference on Computer and Robot Vision, Montréal, Québec, Canada, 2007, pp. 89–96.

[25] F. Jean, R. Bergevin, A. B. Albu, Body tracking in human walk from monocular video sequences, in: Proceedings of the Second Canadian Conference on Computer and Robot Vision, Victoria, BC, Canada, 2005, pp. 144 – 151.

[26] R. O. Duda, P. E. Hart, Pattern Classification and Scene Analysis, New York: John Wiley & Sons, 1973.

[27] K. S. Arun, T. S. Huang, S. D. Blostein, Least-squares fitting of two 3-d point sets, IEEE Transactions on Pattern Analysis and Machine Intelligence 9 (1987) 698–700.

[28] S. Sarkar, P. Phillips, Z. Liu, I. Vega, P. Grother, K. Bowyer, The humanid gait challenge problem: Data sets, performance, and analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 162–177.

[29] A. Kale, A. K. R. Chowdhury, R. Chellappa, Gait-based Human Identification from a Monocular Video Sequence, 3rd Edition, World Scientific Publishing Company Pvt. Ltd., 2004.

[30] D. Lee, Effective gaussian mixture learning for video background substraction, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 827–832.

[31] Z. Zivkovic, Improved adaptive gausian mixture model for background subtraction, in: Proceedings of the IEEE International Conference on Pattern Recognition, Cambridge, UK, 2004.