SIViP (2011) 5:415–430
DOI 10.1007/s11760-011-0245-5

ORIGINAL PAPER

Occlusion detection and gait silhouette reconstruction from degraded scenes

Aditi Roy · Shamik Sural · Jayanta Mukherjee · Gerhard Rigoll

Received: 31 July 2010 / Revised: 14 December 2010 / Accepted: 17 February 2011 / Published online: 29 July 2011

Abstract Gait, which is defined as the style of walking of a person, has been recognized as a potential biometric feature for identifying human beings. The fundamental nature of gait biometric of being unconstrained and captured often without a subject's knowledge or co-operation has motivated many researchers over the last one decade. However, all of the approaches found in the literature assume that there is little or no occlusion present at the time of capturing gait images, both during training and during testing and deployment. We look into this challenging problem of gait recognition in the presence of occlusion. A novel approach is proposed, which first detects the presence of occlusion and accordingly extracts clean and unclean gait cycles from the whole input sequence. In the second step, occluded silhouette frames are reconstructed using Balanced Gaussian Process Dynamical Model (BGPDM). We evaluated our approach on a new data set TUM-IITKGP featuring inter-object occlusion. Algorithms have also been tested on CMU's Mobo data set by introducing synthetic occlusion of different degrees. The proposed approach shows promising result on both the data sets.

Keywords Gait · Occlusion · Pose detection · Dynamic programming · BGPDM · Silhouette reconstruction

A. Roy (✉) · S. Sural
School of Information Technology,
Indian Institute of Technology Kharagpur,
Kharagpur, West Bengal 721302, India
e-mail: aditi.roy@sit.iitkgp.ernet.in

S. Sural
e-mail: shamik@cse.iitkgp.ernet.in

J. Mukherjee
Department of CSE, Indian Institute of Technology Kharagpur,
Kharagpur, West Bengal 721302, India
e-mail: jay@cse.iitkgp.ernet.in

G. Rigoll
Institute for Human Machine Communication,
Technical University of Munich, Munich, Germany
e-mail: rigoll@tum.de

1 Introduction

Biometrics-based human identification is becoming increasingly important in visual surveillance systems, tracking, monitoring, forensics, etc., since they provide reliable and efficient means of identity verification. Gait-based human recognition is one of the topics of active interest in biometric research since it provides many unique advantages such as non-contact, non-invasive, and perceivable at a distance compared with the traditional biometric features such as face, iris, palm print, and finger print. Gait recognition refers to verifying and/or identifying persons using their walking style. Although gait analysis for human identification is not yet as mature as fingerprint, iris, or face, it can still be a useful tool. For example, in a bank robbery case in Denmark, the Court found gait analysis from video to be a valuable evidence [1]. Often, in cases of robbery, the perpetrator hides his face using mask and puts on gloves so that no face image or fingerprint is captured, but security cameras can record his gait.

Gait of a person is a periodic activity where each gait cycle covers two strides: the right foot forward and the left foot forward as shown in Fig. 1. Like other biometric-based systems, gait recognition also follows three stages of operation. At first, videos of walking subjects are captured. In the second step, gait sequences are analyzed for detecting the gait cycles. Then, relevant features are extracted from each cycle usually from the shape and dynamics of each stride. Shape

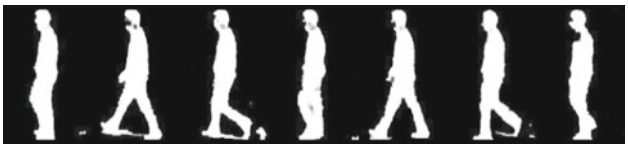

Fig. 1 The key stances of a gait cycle

means physical build of a person seen in different gait phases, and dynamics means motion dynamics of the person in a gait cycle. The shape and the dynamics of a gait cycle for a person together form the gait biometric feature for that person [22]. In the last stage, the extracted feature set is compared against a template set maintained in the database. Generally, multiple gait cycles are used in order to make the system robust against small variations that can occur in individual cycles.

Almost all of the available gait data sets consider that only a single person is moving in the field of view of the camera [14,17,33,34]. Thus, the gait cycles obtained from those videos are clean. All the gait recognition techniques proposed in the literature have been developed considering such clean gait cycles. However, in real-life applications, more than one person could be present in the field of view of the camera and almost invariably they occlude one another. Occlusion can occur due to other factors as well, like the presence of beams, pillars, and other non-living objects. Since the gait video sequence is captured without the subject's active participation or co-operation and in unconstrained environment, this type of situation is more likely to happen. An example sequence of video frames where the first person on the right in the first frame is occluded during his gait cycle is shown in Fig. 2. Thus, unlike the example shown in Fig. 1, it may not be possible to extract clean gait cycles for this subject. None of the methods available in the literature is able to recognize the subject from such a video sequence using gait features even if the classifier was originally trained for the same individual.

For human recognition using gait in the presence of occlusion, it has to be first determined how many clean gait cycles could be captured in the video. A gait cycle is represented by a series of key poses. If all the key poses are present in a gait cycle, the gait cycle is considered to be clean. Given a gait sequence, key pose estimation is done to determine which of the following situations has occurred:

- The gait sequence contains multiple clean gait cycles.
- It has only one clean gait cycle.
- There is no clean gait cycle.

In the first case where multiple clean gait cycles are present, classification can be carried out in the usual process by extracting suitable gait features from the available clean gait cycles. In the second case, recognition can be done in two different ways. First, the single clean cycle can be used for recognition. But, in this case, the recognition accuracy degrades due to the vulnerability of gait features to small variations in individual cycles. As an alternative, occluded silhouettes of partial gait cycles can be reconstructed to get multiple clean gait cycles. Then, usual gait recognition process can be followed. In the third situation, occluded silhouettes have to be reconstructed to get at least one clean gait cycle.

To determine which of the above three situations had occurred, the input silhouette sequence is first partitioned into subsequences of one gait cycle length. Then, each of these subsequences is checked to determine whether any of the poses is occluded. If occlusion is present, then that subsequence is considered as unclean. But determining gait cycles correctly in the input sequence in the presence of multiple degraded silhouettes is not possible using the methods proposed in Sundaresan et al. [12], Sarkar et al. [17]. Hence, an alternative approach has to be devised, which can simultaneously detect key poses, occluded poses as well as the gait cycles. In this paper, we propose a novel method to classify an input sequence of silhouette frames to the most probable key poses. Since a sequence of key poses makes up one gait cycle, classification of the input silhouette sequence as clean or unclean can be done depending on the output of this step, namely how many of the key poses could be identified.

After detecting which of the silhouettes is degraded by occlusion, the next step is to reconstruct them. Here, we apply Gaussian Process Dynamic Model (GPDM) [28], which is

Fig. 2 A sequence of frames where a subject is occluded by dynamic objects (dynamic occlusion)

a latent variable model used for nonlinear time series analysis, to reconstruct the occluded silhouettes.

The rest of this paper is organized as follows. In Sect. 2, we describe the existing approaches for gait recognition. Section 3 describes the overall approach in detail. Section 4 introduces a dynamic programming-based key pose estimation and occlusion detection method. We describe the silhouette reconstruction approach in Sect. 5. Then, in Sect. 6, we present detailed results and finally conclude in Sect. 7.

2 Related work

Gait recognition approaches are mainly classified into two types, namely model-based approaches and motion-based or holistic approaches. Both of these approaches follow the usual framework of biometric-based human recognition, i.e., feature extraction, feature correspondence, and high level processing. The difference is in the way feature correspondence is done.

Model-based approaches

Model-based approaches generally model the human body or its motion from input gait sequences. Then, the model is matched in every frame of a gait sequence by measuring the parameters such as trajectories, limb lengths, and angular speeds. Cunado et al. [2] and Yam et al. [4] first extracted leg motion and then computed gait signature by Fourier analysis. Activity-specific static body parameters are used by Johnson and Bobick [3] without directly analyzing gait dynamics. Jain et al. [5] proposed a fuzzy approach, where they used a bio-mechanical model for identification. In [6], Zhang et al. introduced a novel approach by employing a five-link biped locomotion human model. But, here, the recognition rate is significantly limited by the distance of the subject from the camera.

In contrast to the above, Lu et al. [7] proposed a full-body layered deformable model using manually labeled silhouettes. The model is defined for the fronto-parallel gait with 22 parameters describing human body part shapes (widths and lengths) and dynamics (positions and orientations). While other model-based approaches mainly focus on lower limbs, this approach is based on full-body model utilizing the dynamics of upper limbs, shoulders, and head as well. Another manually labeled silhouette-based approach is the one proposed in Huang and Boulgouris [8]. This approach fuses several discriminative features extracted from manually labeled silhouettes, i.e., the area, the gravity center, and the orientation of each body component. Although they reported promising results, these approaches are restricted by the use of manually labeled silhouettes. While model-based methods are generally view and scale invariant, the use of such methods is still limited due to current imperfect vision techniques (e.g., tracking and localizing human body accurately in 2D or 3D space has long been a challenging and unsolved problem), requirement of good quality silhouettes, and high computational cost.

Motion-based or holistic approaches

Most of the current approaches are motion based, which directly use the silhouettes of gait sequences for feature extraction without developing any model. These approaches are further categorized into two classes, namely *state-space methods* and *spatiotemporal methods*.

In *state-space methods*, gait dynamics is assumed to be composed of a sequence of static gait poses. Temporal variation of observations with respect to these static poses is used for recognition [9,10].

Spatiotemporal methods characterize the spatiotemporal distribution of gait dynamics. The earliest approach in this category is by Niyogi and Adelson [13]. In their approach, recognition is done using spatiotemporal gait patterns obtained from curve-fitted 'snake'. Later, Little and Boyd used frequency and phase features from optical flow information of gait [14] and obtained better recognition result on a small database of six subjects. This method is susceptible to noise. BenAbdelkader et al. [15] used image self-similarity plots of a moving person to recognize gait. For recognition and classification, they use principal component analysis (PCA) and K-nearest neighbor method, respectively. Their approach is sensitive to walking speed, clothing, lighting, etc. Vega and Sarkar [16] described a gait recognition method, which exploits the non-stationarity in the distribution of feature relationships. Sarkar et al. [17] proposed a baseline algorithm based on spatiotemporal silhouette correlation. This approach is quite often used as a reference for comparing different gait recognition methods.

An automatic gait recognition method using spatiotemporal symmetry was introduced in Hayfron-Acquah et al. [18]. Generalized symmetry operator is used for extracting the features. Lee et al. propose a method to divide the silhouette of a walking person into regions to facilitate the recognition task [19]. The features used for recognition are view and appearance based. As a result, with a change of appearance of the subject, recognition accuracy decreases. Han and Bhanu [23] proposed a new gait feature named as Gait Energy Image (GEI). In GEI, the spatiotemporal information is represented in a single 2D gait template. GEI reflects major shapes of silhouettes and their changes over the gait cycle. They reported promising result using this new feature representation. Later, a number of variations of this basic GEI feature have been proposed, namely enhanced GEI [24], frame difference energy image (FDEI) [25], and active energy image [26]. In [20], a detailed comparison was done for several

motion-based techniques. Advantages and disadvantages of different features are also identified.

Lu and Zhang [21] proposed a method using multiple features and view fusion based on Genetic Fuzzy Support Vector Machine (GFSVM). They show that the recognition performance using a fusion of multiple features and multiple views is better than single feature– or single view–based methods. Another multiple feature–based approach was proposed in Chen et al. [11]. Factorial HMM (FHMM) is employed here as a feature level fusion scheme for fusing different gait features, which is then compared with a Parallel HMM (PHMM)- based decision level fusion scheme.

It can be observed from the above discussion that all of the existing approaches tried to improve the recognition accuracy irrespective of walking environment, viewing angle, walking surface, walking style, carrying condition, etc. But all of them ignore the important aspect of occlusion that is quite common in real-world scenarios. This issue needs to be addressed in order to do gait recognition in practical real-life situations. Our current work addresses this challenging problem of occlusion handling in the context of gait recognition.

3 Overall algorithm description

Figure 3 shows the block diagram of the proposed approach, and Algorithm 1 describes the steps followed.

Algorithm 1 Complete Algorithm

Step 1: ***Key Pose Estimation***
Input: Training silhouettes $(I_i, i = 1, \ldots, I_M)$, No. of eigen vectors $(\mathcal{K})$, No. of key poses (K)
Output: Key poses in eigen space$(P_1, \ldots, P_K)$
Step 1.1: Project silhouettes (I_i) into eigen space and obtain $\mathcal{K}$ eigensilhouette $(u_i, i = 1, \ldots, \mathcal{K})$
Step 1.2: Compute weight vectors $(\Omega_i, i = 1, \ldots, M)$
Step 1.3: Apply K-means clustering on eigen images (Ω_i) to get K key poses representing a gait cycle $(P_1, \ldots, P_K)$

Step 2: ***Occlusion Detection***
Input: Test silhouettes $(TI_1, \ldots, TI_T)$, Key poses $(P_1, \ldots, P_K)$
Output: Clean and Unclean gait cycles
Step 2.1: Apply eigen-space projection on training silhouettes $(TI_i)$to get the weight vectors (Ω_i)
Step 2.2: Compute match scores among all training images and key poses $P_1, \ldots, P_K$
Step 2.3: Apply graph-based path searching to find out the most likely poses of the test silhouette sequence
Step 2.4: Analyze the result and find out which and how many gait cycles are degraded by occlusion

Step 3: ***Missing Silhouette Reconstruction***
Input: Training silhouettes $(I_1, \ldots, I_N)$, test silhouettes $(TI_1, \ldots, TI_T)$ where $TI_i - TI_j$ are missing
Output: Reconstructed silhouettes $(RI_i - RI_j)$
Step 3.1: Training/ Learning: Train GPDM with normal walking sequences $I_1, \ldots, I_N$
Step 3.2: Deployment: Use trained GPDM to predict the missing silhouettes $TI_i - TI_j$ as $RI_i - RI_j$

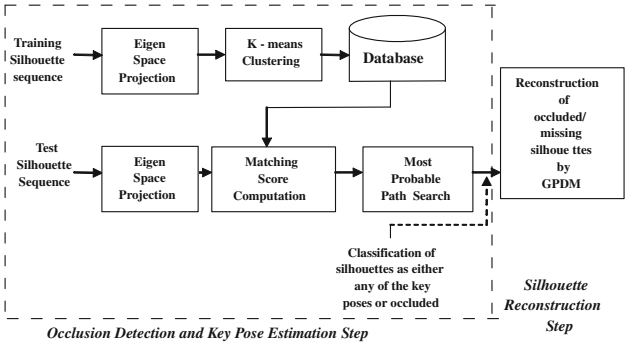

Fig. 3 Block diagram of the proposed approach

For classifying the input silhouettes into key poses, it is required to define the key poses first. Since there is no standard way of determining the number of key poses and their characteristics, we apply unsupervised learning, specifically constrained K-means clustering, to choose the key poses. Instead of directly applying K-meaning clustering, PCA is first applied on training silhouette images to map them in the eigen space. Since PCA finds a small set of orthogonal vectors ($\mathcal{K}$) that captures the observed total variance in a better way than the original feature space, clustering in eigen space results in better clusters (see steps 1.1 and 1.2 in Algorithm 1). Once the weight vectors of each silhouette in the eigen space is obtained, K-means clustering is applied on these weight vectors (see step 1.3 in Algorithm 1). The clusters thus formed represent different key pose classes, and the mean vectors are used to represent the key poses.

Next stage is occlusion detection. Given the input test sequence $(TI_1, \ldots, TI_T)$, each silhouette is first linearly projected into the eigen space to get the weight vectors (see step 2.1 in Algorithm 1). Euclidean distance is used to compute a match score between the observed weight vector of a silhouette and each of the key poses (see step 2.2 in Algorithm 1). If there are K key poses and T frames in a sequence, an $[K \times T]$ array of match scores is obtained. From these match scores, the input silhouette can be classified directly by considering the best- matched key pose. But this oversimplified method does not consider the temporal context of key pose sequence. Thus, only shape-based classification can potentially give misleading result. So, we use a state transition model (see Fig. 5), which shows the allowable transitions between key poses that are associated with each state in the model. Based on this model, a directed graph is constructed, where vertices are the key poses and edges represent allowable transitions. Classification of an input sequence to a sequence of known key poses is formulated as a most probable path search problem which is solved using dynamic programming (see step 2.3 in Algorithm 1). Each silhouette in the input sequence is labeled into one of the key poses. Substantially degraded silhouette frames are classified as occluded. Thus, the output

Mapped Sequence: S1--S2--S2--S3--S3--S3--S4--S4--S4--S5--S5--S6--S6--S7--S8--S8--S8--S9--S10--S10--S11--S11--S12--S12--S12--S13--S13--S14--S14--S15--S15--S0--S0--S0--S0--S0--S0--S0--S0--S3--S3--S4--S4--S4--S5--S5--S6--S6--S7--S7--S8--S8--S9--S0--S0--S0--S0--S0--S0--S0--S0--S0--S13--S14--S14--S14--S15--S15--S16--S16--S1--S1--S2--S2--S2--S3--S3--S4--S4--S4--S5--S5--S5--S6--S6--S7--S7--S8--S8--S9--S9--S10--S11--S11--S12--S12--S13--S13--S13--S14--S14--S15--S15--S15--S16--S16

GC 1: S1 --S2--S2--S3--S3--S3--S3--S4--S4--S4--S5--S5--S6--S6--S7--S8--S8--S8--S9--S10--S10--S11--S11--S12--S12--S12--S13--S13--S14--S14--S15--S15--**

GC 2: ** --S3--S3--S4--S4--S4--S5--S5--S6--S6--S7--S7--S8--S8--S9--**--S13--S14--S14--S14--S15--S15--S16--S16

GC 3: S1 --S1--S2--S2--S2--S3--S3--S4--S4--S4--S5--S5--S5--S6--S6--S7--S7--S8--S8--S9--S9--S10--S11--S11--S12--S12--S13--S13--S13--S14--S14--S15--S15--S15--S16--S16

Fig. 4 Output of the pose estimation step. Mapped Sequence shows class of each frame of the input sequence. Index labels 'S1' to 'S16' denote one of the sixteen key poses, and index label 'S0' denotes occluded pose. From this mapped sequence, three extracted subsequences are shown as GC 1, GC 2, and GC 3. Subsequence GC 1 and GC 2 are unclean and GC 3 is clean. *Asterisk* indicates presence of occluded frame(s)

at this stage is a sequence of key pose labels representing the most probable class of each silhouette frame in the input sequence as shown in Fig. 4. By analyzing these class labels, the subsequence of frames corresponding to a gait cycle can be extracted. Since a sequence of frames containing all the key poses constitute a clean gait cycle, if any of the frames in the sequence is identified as occluded, then the complete sequence is labeled as unclean. After checking whether any of the frames in a subsequence is classified as occluded or not, it can also be determined whether the subsequence is clean or not (see step 2.4 in Algorithm 1). Thus, we can indirectly identify the gait cycles in the input sequence without applying any of the methods used in Sundaresan et al. [12], Sarkar et al. [17].

In the final stage, reconstruction of occluded silhouettes is done using GPDM. First, the GPDM is trained to learn the model parameters (see step 3.1 in Algorithm 1). This learned model is then used for reconstructing the missing silhouettes (see step 3.2 in Algorithm 1). The reconstructed silhouettes are used for reconstructing the unclean gait cycles, so that they can now be used for feature extraction and subsequent gait recognition using any of the existing methods described in Sect. 2.

4 Key pose estimation and occlusion detection

As discussed before, the first step in occlusion handling is to determine whether any of the gait cycles present in the input sequence is degraded by occlusion. In this section, we describe the method of key pose estimation and occluded frame detection.

4.1 Eigen-space projection

As discussed above, the first step is to apply eigen-space projection to find the principal components or the eigenvectors of the silhouette image set. Since these eigenvectors have a silhouette- like appearance, we call them eigensilhouettes. Every silhouette image in the training set can be represented as a weighted linear combination of these basis eigensilhouettes. The number of eigensilhouettes we obtain is equal to the number of silhouette images in the training set. Since some of these eigensilhouettes are more important in encoding the variation in silhouette images, we select only the K most significant eigensilhouettes.

Let there be M training silhouette images $I_1, I_2, \ldots, I_M$ of size $S = W \times H$, where W = width and H = height of a silhouette image frame. We represent each image $I_i \in I$ as a column vector Γ_i of size $S \times 1$, where I represents the training set. We find the mean silhouette vector Ψ as follows:

$$\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i \tag{1}$$

Next, we compute the normalized silhouette image vector Φ_i by subtracting the mean silhouette vector Γ_i from each training silhouette vector.

$$\Phi_i = \Gamma_i - \Psi \tag{2}$$

Thus, only the distinguishing features from each silhouette are considered. We then find the covariance matrix C as follows:

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T = \frac{1}{M}\sum_{n=1}^{M}(\Gamma_n - \Psi)(\Gamma_n - \Psi)^T = AA^T \tag{3}$$

where $A = [\Phi_1, \Phi_2, \ldots, \Phi_M]$. Since the size of C is $S \times S$, computing eigenvectors u_i from this covariance matrix is intractable for typical image sizes [35]. This problem can be solved by first computing the eigenvectors of much smaller A^TA matrix of size $M \times M$ and taking linear combinations of the silhouette images Φ_i. For example, in TUM-IITKGP dataset, the number of images used for training, $M \approx 2{,}000$ of size $S = 88 \times 128$ ($M \ll S$). By solving A^TA, M eigenvectors ($v_i, i = 1, 2, \ldots, M$) are obtained, each of dimension $M \times 1$. Now from matrix properties, we compute eigenvectors ($u_i, i = 1, 2, \ldots, M$) of the covariance matrix $C = AA^T$ as $u_i = Av_i$ [35]. Since the dimension of A is $S \times M$, the dimension of u_i becomes $S \times 1$. u_i is normalized such that $\|u_i\| = 1$. Thus, M eigenvectors of C are obtained.

Since the number of training data M is very large (depends on the dataset size, e.g., TUM-IITKGP dataset $M \approx 2{,}000$), the number of eigenvectors is still large. Further, all of them

do not contain significant information. To select $\mathcal{K}$ most significant eigenvectors, we sort the eigenvalues in decreasing order and select $\mathcal{K}$ number of eigenvectors that account for variance more than 90%.

Once eigenvectors are computed, we find the weight vectors, also known as silhouette space image, as follows:

$$\Omega_i = u^T \Phi_i, \quad i = 1, 2, \ldots, M \tag{4}$$

where $u = [u_1, u_2, \ldots, u_{\mathcal{K}}]$, $\mathcal{K} \leq M$, and silhouette space image $\Omega_i = [w_1, w_2, \ldots, w_{\mathcal{K}}]^T$.

Now, each silhouette in the training set (mean subtracted), Φ_i, can be represented as a linear combination of these eigensilhouettes u_i as follows

$$\Phi_i = \sum_{j=1}^{\mathcal{K}} w_j u_j \tag{5}$$

4.2 K-means clustering

After finding the most relevant eigensilhouettes for the training images, the weight vector of each silhouette is computed by projecting it into the eigen space and these vectors are stored in the database. The next step is to determine the key poses present in a gait cycle. We apply constrained K-means clustering, an unsupervised learning technique, such that each cluster represents a key pose class. The inherent sequential nature of the key poses in a gait cycle makes the clusters formed by K-means clustering temporally adjacent. Since the key poses are defined in a way to represent the asymmetry of an entire gait cycle, separate key poses or clusters are formed during left foot forward position and right foot forward position.

Let us assume that feature vectors of a gait cycle of the nth subject is given by $\mathcal{O}^n = \omega_1^n, \omega_2^n, \ldots, \omega_p^n$, where $p =$ number of frames in a gait cycle. We initialize the clusters by equally partitioning each gait cycle into K segments. The jth frame is assigned to cluster $i = 1 + \lfloor (\frac{j*K}{p}) \rfloor$, where $i \in K$. Thus, all the frames in the ith segment of each gait cycle of all the subjects are grouped under the ith cluster. Let the initial set of clusters be $\mathcal{S}^0 = S_1^0, S_2^0, \ldots, S_K^0$, and the corresponding centroids are $\mathcal{P}^0 = P_1^0, P_2^0, \ldots, P_K^0$, each of which represents a key pose. Then, constrained K-means clustering is applied for iteratively refining the clusters. We apply the constraints to maintain the sequential nature of gait poses. The constraints are as follows:

- The only allowable transitions are from the ith cluster to $(i-1)$th or ith or $(i+1)$th clusters.
- After performing cluster assignment by taking the first constraint into account, check the transition order of each frame. If it is not ordered properly, then reassign those frames such that the previous frame's cluster is lower or equal to the current frames's cluster.
- Ensure that every cluster has at least one frame from each gait cycle of each subject.

After initialization, the algorithm proceeds by alternating between the following two steps:

Update step: Calculate the centroid of the cluster.

$$P_i^{(t)} = \frac{1}{|S_i^{(t)}|} \sum_{\omega_j \in S_i^{(t)}} \omega_j \tag{6}$$

Assignment step: Reassign each frame to the allowable cluster with the closest mean.

$$S_i^{(t+1)} = \left\{ \omega_j : \left\| \omega_j - P_i^{(t)} \right\| \leq \left\| \omega_j - P_j^{(t)} \right\| \right. \\ \left. \text{for } j = i-1 \text{ or } i+1 \right\} \tag{7}$$

The algorithm terminates when the assignments no longer change.

4.3 Match score computation

Let the mean weight vectors corresponding to the key poses be $(P_1, P_2, \ldots, P_K)$. Given an unknown probe silhouette Γ, we first normalize it as $\Phi = \Gamma - \Psi$. Then, this normalized silhouette is projected onto the eigen space and the weight vector is determined as follows:

$$\Omega = u^T \Phi \tag{8}$$

After the feature vector (weight vector) for the probe silhouette is computed, the match scores of the probe silhouette to all of the key pose weight vectors $(P_1, \ldots, P_K)$ are determined. To do this, we use simple Euclidean distance measure $(D(P_i - \Omega))$. If $D(P_i, \Omega) < \Theta$, where Θ is a threshold chosen empirically, then the probe image can be matched to one of the key poses. If, however, $D(P_i\Omega) > \Theta$, then the probe does not belong to any of the key poses. This situation occurs when a silhouette is degraded due to occlusion. Thus, $D(P_i, \Omega) > \Theta$ indicates the presence of occlusion in the corresponding silhouette image. To choose the threshold, we consider a large set of random silhouette images (both occluded and not occluded) and calculate the distance values for silhouette images in the database and also for this random set. The threshold Θ is set accordingly.

Since we require similarity score, we compute $S(P_i, \Omega) = 1 - D(P_i - \Omega)$ for $i = 1, 2, \ldots, K$. When $D(P_i, \Omega) > \Theta \forall i$, $S(Occluded, \Omega) = 1$ and $S(P_i, \Omega) = 0$. Conversely, when $D(P_i, \Omega) < \Theta \forall i$, $S(Occluded, \Omega) = 0$ and $S(P_i, \Omega) = 1 - D(P_i - \Omega)$. All the similarity values are then normalized.

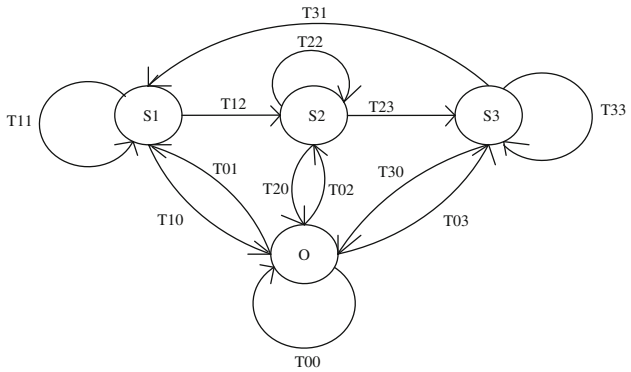

Fig. 5 Proposed state transition diagram considering three states (S1–S3) corresponding to three key poses (P1–P3) and one occluded pose state (O)

4.4 Graph-based path searching

A gait cycle is modeled as a chain of estimated key poses. An example state transition diagram considering only three key poses in a gait cycle is shown in Fig. 5. Each state in this graph model corresponds to one key pose. This state transition model provides contextual and temporal constraints where the links specify the temporal order of the key poses. The special feature of this model is inclusion of an occluded state. Like each key pose associated with a state, we associate an occluded pose to the occluded state. Since occlusion can occur randomly at any frame in a sequence, transition to the occluded state can occur from any key pose state. Similarly, when occlusion is over, the next key pose can be any pose depending on the duration of occlusion. Thus, we can determine when occlusion occurs and in how many frames.

Now, suppose there are N states in the state transition model and T silhouette frames in the input sequence. It is required to determine the key pose of each silhouette frame. In the preceding subsection, an $N \times T$ array of matching scores is obtained. To find out the key pose of the ith silhouette frame, one straightforward approach would be to assign the best-matched key pose as the key pose of the ith frame. However, this oversimplified approach does not consider the following factors which may lead to false detection.

- Silhouettes can be easily distorted by a bad foreground segmentation, and thus the matching score may be misleading.
- Even if silhouettes are clean, different key poses may generate similar silhouettes (like left foot forward position and right foot forward position).

Thus, decision based only on individual matching scores is unreliable. To robustly recognize key poses from unreliable individual observations, we take advantage of the temporal constraints imposed by the state transition model and formulate the key pose finding problem as the most likely path finding problem in a directed graph, where each key pose is a node and the edges are the allowable transitions among them.

4.4.1 Directed graph construction

We construct a directed graph from the state transition model proposed in the previous subsection. Each node of the graph represents a state while each edge represents allowable state transition from the current silhouette frame to the next silhouette frame. The graph is constructed as follows.

Let, the input sequence of frames be $\mathbf{F} = \mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_T$ and the possible set of states in the ith frame be $\mathcal{S}^i = S_1^i, \ldots, S_N^i$, where S_1 to (S_{N-1}) states represent key poses $\mathcal{P}^i = P_1^i, P_2^i, \ldots, P_{N-1}^i$ and S_N represents occluded pose. The set of vertices V of the graph $\mathbb{G}_\mathbf{F}$ corresponds to the key pose states and occluded state, $\mathcal{S}^i = S_1^i, S_2^i, \ldots, S_N^i$ for $i = 1$ to T. An edge $e \in E$ of the graph $\mathbb{G}_\mathbf{F}$ links vertices in frame $\mathcal{F}_i$ with vertices in frame $\mathcal{F}_i + 1$. Thus, edge $e_{kp}^i : S_k^i \rightarrow S_p^{i+1}$ is added to the graph $\mathbb{G}_\mathbf{F}$ only if transition from S_k to S_p is allowable by the state transition diagram. The graph thus constructed is a directed acyclic graph. Figure 6 shows an example of a graph constructed from the example state transition model shown in Fig. 5 for five consecutive frames.

4.4.2 Most likely pose sequence search

Once the graph is constructed, we need to find out the most probable key pose assignment for each silhouette frame in the input sequence. As mentioned before, since we want to consider the temporal context during key pose assignment, the graph path search technique is adopted. Thus, the most likely sequence of key poses for a sequence of frames will be the most probable path (the path having maximum weight) belonging to the set of all admissible paths in the directed graph, which can be formulated as a dynamic programming problem [30].

The example shown in Fig. 6 is used to illustrate how dynamic programming is employed in our approach. The figure shows the graph constructed with three key pose states and one occluded state for a five frame sequence. The goal is to find a path from the first frame to the last frame having maximum path weight. At each time step, we compute three values for each state: the matching score (the first value shown in each node) between state j in the graph and input frame $\mathcal{F}_i$, the best score (the second value shown in each node) along a path up to node (t_{ij}) and the previous element on this path. The matching score actually represents to what extent the silhouette of the current input frame matches the key pose corresponding to state j. The procedure for computing this value is described in detail in Sect. 4.3. At frame

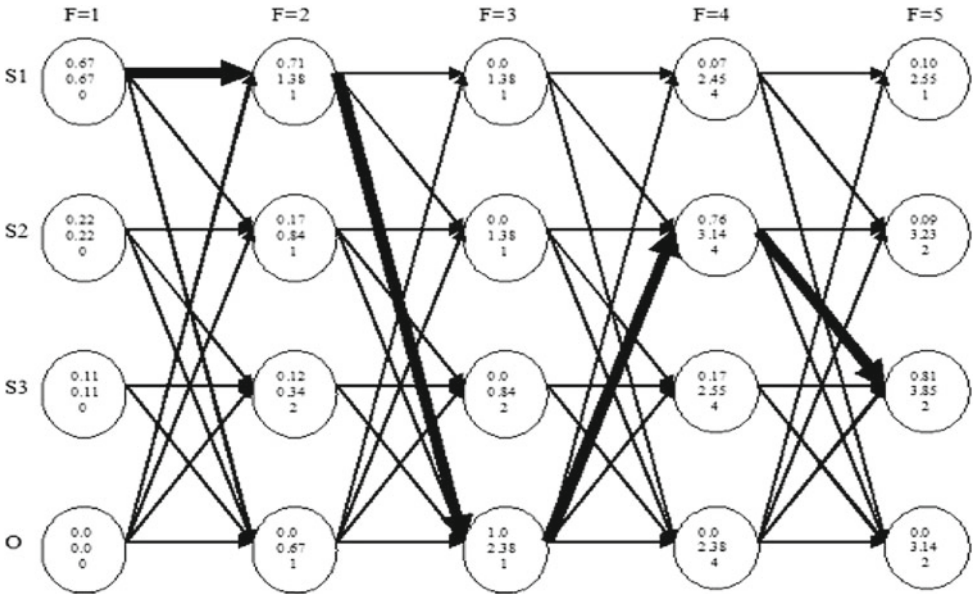

Fig. 6 Directed acyclic graph constructed for three key pose states (S1–S3) and one occluded state (O) over five frames. The *bold* edges show the most probable path found by dynamic programming. The pose assignment obtained for each frame is: S1–S1–O–S2–S3(1–1–4–2–3)

number $\mathcal{F}_i$, node t_{ij} searches at every possible previous node that links to the current node in the graph and chooses the one with the maximum path score. The path score of the current node t_{ij} is updated accordingly, and the selected previous node is recorded (the third value shown in each node). When the last frame is reached, the node with the maximum path score is selected and then backtracking starting from this node is done to get the most probable path (shown in bold). The complete algorithm for finding most probable path is described in Algorithm 2. The complexity of the algorithm for a fully ergodic graph (graph constructed from all possible transitions between states in the state transition diagram) is $O(N^2T)$, where N is the number of states and T is the number of frames. In our case, since the average in-degree of each node is small, the overall complexity reduces to $O(NT)$.

5 Reconstruction of occluded silhouettes

For occluded silhouette estimation/reconstruction, we apply a learning-based approach to model the silhouette observations and their dynamics using GPDM. In [27], Pullen et al. reported that human motions have certain cooperative relationships especially when people do some specific movements like walking, swimming, etc. This relationship can be used to reduce the high observation space dimensionality while performing human motion analysis. So, human motion can be modeled in a low-dimensional latent space. Since human motion is non-linear, basic dimensionality reduction methods such as PCA are inadequate to describe it. Thus, these methods are not suitable for building low-dimensional walking models [28]. Gaussian Process Latent Variable

Algorithm 2 Key Pose Detection Algorithm

Input:

T = total number of frames in a sequence
F = a sequence of frames $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_T$ where $\mathcal{F}_1$ is the starting frame and $\mathcal{F}_T$ is the ending frame
Construct the graph $\mathbb{G}_{\mathbf{F}}(V, E)$
N = number of nodes in each frame (same as the number of states in the state transition model)
$t_{ij} = j^{th}$ state in frame $\mathcal{F}_i$
$E(t_{ij}, t_{kl})$ = edge joining the node t_{ij} with t_{kl}
$MatchScore(t_{ij})$ = probability of being $\mathcal{F}_i$ in j^{th} state
$BestScore(t_{ij})$ = weight of the most probable path up to $\mathcal{F}_i$ that accounts for first i frames and ends in state j
$PrevNode(t_{ij})$ = keeps track of the state that maximizes $BestScore(t_{ij})$ in previous frame $\mathcal{F}_{i-1}$

Output: $MaximumWeightedPath$ from $\mathcal{F}_1$ to $\mathcal{F}_T$, and $BestState_i$ for $i = 1, \ldots, T$

Initialization:

$BestScore(t_{1j}) = MatchScore(t_{1j})$,
$BestScore(t_{ij}) = 0 \ , \ \forall i > 1, j$
$PrevNode(t_{ij}) = 0, \ \forall i, j$

Iteration:

For every frame $\mathcal{F}_i$ and for each j^{th} state in $\mathcal{F}_i$, such that $1 \leq i \leq T$
Compute:
$BestScore(t_{ij}) = max(BestScore(t_{kl}) + MatchScore(t_{ij}))$
$\forall$ nodes in the previous frame $\mathcal{F}_k$ such that $k = i - 1$
$PrevNode(t_{ij}) = argmax(BestScore(t_{kl}) + MatchScore(t_{ij}))$

Termination:

$MaximumWeightedPath = max(BestScore(t_{Ti}))$
$\forall$ nodes i in the last frame $\mathcal{F}_T$
$BestState_T = argmax(BestScore(t_{Ti}))$

Path Backtracking:

$BestState_i = PrevNode(t_{(i+1)(BeatState_{i+1})})$,
$i = T - 1, T - 2, \ldots, 1$

Models (GPLVM) [29] and later extended approaches likes Gaussian Process Dynamic Models (GPDM) [28,31] can learn a non-linear mapping between the observation space and the latent space, and they also provide an inverse mapping. But GPLVM is a static model that represents data independently without considering their temporal continuity. GPDM is specially designed to handle the chronological relations between successive data points, in our case a silhouette sequence. It can also learn dynamical model from missing data and produce estimates of them. This motivates us to use GPDM to predict the missing silhouettes. The silhouette frames that are detected as occluded are treated as missing.

5.1 Gaussian process dynamical model

Gaussian Process Dynamic Model (GPDM) [28] is a powerful latent variable model that can be applied for probabilistically modeling high-dimensional nonlinear time series data. It consists of two nonlinear mappings, namely a continuous mapping from the high-dimensional observation silhouette image space to the low-dimensional latent space and a dynamical mapping in the latent space. The model is obtained by marginalizing out the parameters of the two mappings in closed form by using Gaussian process priors and optimizing the latent coordinates of training data. Suppose $\{I_1, \ldots, I_t, \ldots, I_N\}$ denotes observation silhouette data set and I_t represents a particular silhouette image at time t, $I_t \in R^D$ where D is the size of the silhouette image. $\{L_1, \ldots, L_t, \ldots, L_N\}$ is the set of corresponding data points in the latent space, L_t represents the d-dimensional latent coordinate of the silhouette image at time t, $L_t \in R^d$. The first-order Markov dynamics and the latent space mapping can be expressed as:

$$L_t = \sum_i a_i \phi_i(L_{t-1}) + n_{L,t} \tag{9}$$

$$I_t = \sum_j b_j \varphi_j(L_t) + n_{I,t} \tag{10}$$

where weights $A = [a_1, a_2, \ldots]$, $B = [b_1, b_2, \ldots]$, ϕ_i and φ_i are basis functions, and $n_{L,t}, n_{I,t}$ are zero-mean, isotropic, white Gaussian noise processes. The model parameters A and B are marginalized out in GPDM through model averaging. Using an isotropic Gaussian prior on each b_j, we can marginalize over B in closed form to yield a multivariate Gaussian data likelihood [28]:

$$p(\mathcal{I} \mid \mathcal{L}, \overline{\beta}) = \frac{|W|^n}{\sqrt{(2\pi)^{ND}|K_{\mathcal{I}}|^D}} \times \exp\left(-\frac{1}{2} tr\left(K_{\mathcal{I}}^{-1} \mathcal{I} W^2 \mathcal{I}^T\right)\right) \tag{11}$$

where $\mathcal{I} = [I_1, \ldots, I_N]^T$ is a matrix of training silhouette images, $\mathcal{L} = [L_1, \ldots, L_N]^T$ is corresponding matrix of latent positions, $K_|$ is a kernel matrix, and $\overline{\beta} = \{\beta_1, \beta_2, \ldots, W\}$ are the hyperparameters of the kernel. $W \equiv diag(w_1, \ldots, w_D)$ is a scaling matrix which captures different variances in the different data dimensions. The kernel matrix elements are defined by a kernel function $(K_|)_{i,j} = K_|(L_i, L_j)$. To get the latent mapping of the training silhouette images, $\mathcal{L} \rightarrow \mathcal{I}$, radial basis function (RBF) is used.

$$k_{\mathcal{I}}(L, L') = \beta_1 \exp\left(-\frac{\beta_2}{2}\|L - L'\|^2\right) + \beta_3^{-1}\delta_{L,L'} \tag{12}$$

where hyperparameter β_1 is the output scale of the kernel function, β_2 is the inverse width of the RBF, and β_3^{-1} is the variance of the additive noise $n_{L,t}$.

The dynamic mapping for latent coordinate $\mathcal{L}$ is similar to the latent space mapping. The joint probability density over the latent coordinates can be represented by:

$$p(\mathcal{L} \mid \overline{\alpha}) = p(L_1) \frac{1}{\sqrt{(2\pi)^{(N-1)d}|K_{\mathcal{L}}|^d}} \times \exp\left(-\frac{1}{2} tr\left(K_{\mathcal{L}}^{-1} \mathcal{L}_{\text{out}} \mathcal{L}_{\text{out}}^T\right)\right) \tag{13}$$

where $\mathcal{L}_{\text{out}} = [L_2, \ldots, L_N]^T$ denotes the latent coordinates of the input silhouette sequence except the first frame, $K_{\mathcal{L}}$ is the $(N-1) \times (N-1)$ kernel matrix constructed from $\mathcal{L}_{\text{in}} = [L_1, \ldots, L_{N-1}]$, and L_1 is given an isotropic Gaussian prior. $\overline{\alpha}$ is a vector of kernel hyperparameters. The dynamics is modeled using the following "Linear + RBF" kernel:

$$k(L, L') = \alpha_1 \exp\left(-\frac{\alpha_2}{2}\|L - L'\|^2\right) + \alpha_3 L^T L' + \alpha_4^{-1}\delta_{L,L'} \tag{14}$$

where hyperparameter α_1 is the output scale, α_2 is the inverse width of the RBF, and α_3 is the output scale of the linear term. α_4^{-1} represents the variance of the noise term $n_{L,t}$. The linear term is useful for approximately linear human motion.

5.2 Training

Training the GPDM from input silhouette data $\mathcal{I} = [I_1, \ldots, I_N]^T$ entails estimating their latent positions and the kernel hyperparameters. To avoid overfitting, prior distributions are placed on hyperparameters ($p(\bar{\alpha}) \propto \prod_i \alpha_i^{-1}$, $p(\bar{\beta}) \propto \prod_i \beta_i^{-1}$). Then, GPDM posterior for training silhouette sequences is obtained through a latent space mapping, a dynamic mapping and prior distributions [28]:

$$p(\mathcal{L}, \bar{\alpha}, \bar{\beta} \mid \mathcal{I}) \propto p(\mathcal{I} \mid \mathcal{L}, \overline{\beta}) p(\mathcal{L} \mid \overline{\alpha}) p(\bar{\alpha}) p(\bar{\beta}) \tag{15}$$

The latent positions and hyperparameters are computed by minimizing the following negative log posterior:

$$\begin{aligned}\ell &= -\ln p(\mathcal{L}, \overline{\alpha}, \overline{\beta} \mid \mathcal{J}) \\ &= \frac{d}{2}\ln|K_{\mathcal{L}}| + \frac{1}{2}tr\left(K_{\mathcal{L}}^{-1}\mathcal{L}_{\text{out}}\mathcal{L}_{\text{out}}^{T}\right) \\ &\quad - N\ln|W| + \frac{D}{2}\ln|K_{\mathcal{J}}| + \frac{1}{2}tr\left(K_{\mathcal{J}}^{-1}\mathcal{J}W^{2}\mathcal{J}^{T}\right) \\ &\quad + \sum_{j}\ln a_j + \sum_{j}\ln\beta_j + C \end{aligned} \tag{16}$$

where C is a constant. We use Balanced GPDM (BGPDM) proposed in Urtasun et al. [32] to improve the smoothness of data in the latent space. BGPDM slightly alters the objective function used during training. The dimension differences in the silhouette pose and latent space are discounted by raising the dynamics density function in (13) to the ratio of their dimensions, i.e., $\lambda = D/d$. Thus, the first two terms in (14) become

$$\lambda\left(\frac{d}{2}\ln|K_{\mathcal{L}}| + \frac{1}{2}tr\left(K_{\mathcal{L}}^{-1}\mathcal{L}_{\text{out}}\mathcal{L}_{\text{out}}^{T}\right)\right) \tag{17}$$

5.3 Inference of missing data

Once the model is learnt, it is used to predict the occluded/missing silhouettes of input test sequences. Given the trained model, $\Gamma = \{\mathcal{J}, \mathcal{L}, \bar{\alpha}, \bar{\beta}, W\}$, the conditional density of the new sequence $\mathcal{J}^*$ and its corresponding latent coordinates $\mathcal{L}^*$ are as follows [28],

$$\begin{aligned} p(\mathcal{J}^*, \mathcal{L}^* | \Gamma) &= p(\mathcal{J}^* | \mathcal{L}^*, \Gamma) p(\mathcal{L}^* | \Gamma) \\ &\propto p(\mathcal{J}, \mathcal{J}^* | \mathcal{L}, \mathcal{L}^*, \bar{\beta}, W) p(\mathcal{L}, \mathcal{L}^* | \bar{\alpha}) \end{aligned} \tag{18}$$

For missing silhouette frames, we set $I^* = \mu_{\mathcal{J}}(Y^*)$. Then, the missing frames are predicted by optimizing (16). Once the occluded silhouette frames are reconstructed, unclean gait cycles are cleaned. These reconstructed clean gait cycles can then be used by any existing algorithm for gait recognition. Thus, the challenges caused due to the presence of occlusions are handled.

6 Experimental results

In this section, we demonstrate the performance of the proposed approach on different gait data sets. Our aim is to evaluate how well the method classifies each frame to the key pose classes and also detect the presence of occlusion in varied situations such as variation in the size of data set, walking surface, walking speed, and carrying condition. After detecting occluded silhouettes, we evaluate the silhouette reconstruction accuracy of the proposed GPDM- based approach. Since none of the existing data sets address occlusion, we use a new data set (TUM-IITKGP), developed as a collaborative research work between Technical University of Munich, Germany, and Indian Institute of Technology, Kharagpur, India, for evaluating the performance of the proposed algorithms in real occlusion situations. In addition to this, we also use the existing MoBo data set from CMU [33]. On this data set, we introduce different degree of occlusion synthetically and evaluate the performance. A description of the new data set and the detailed results are presented in the following Sections.

6.1 TUM-IITKGP data set

In this section, we first describe the camera setup and types of occlusion captured in the TUM-IITKGP data set [36,37]. Next, silhouette extraction and preprocessing methods are presented.

6.1.1 Data set description

To address the gap in existing gait data sets, we have built a new data set that includes two types of occlusion. One type considers the subject to be occluded by dynamic objects (*dynamic occlusion*), i.e., another person walking in the field of view of the camera. In the second type of occlusion, the subject is occluded by static objects (*static occlusion*). The setup for recording is shown in Fig. 7. The camera is placed in a narrow hallway to reflect real-world surveillance application scenario. It is positioned at a height of 185 cm in a direction perpendicular to the hallway. The subject walks along the line AR to AL. The perpendicular distance of the subject from the camera is 517.5 cm. The subject walks 444 cm while within the field of view of the camera. For dynamic occlusion, the subject starts from point AR, and two other occluding persons start from points BL and CL, respectively. Thus, these two occluding persons occlude the subject of interest in two different positions. The same setup is repeated for left-to-right motion, only with the starting points reversed (AL, BR, CR, respectively). One such sequence is shown in

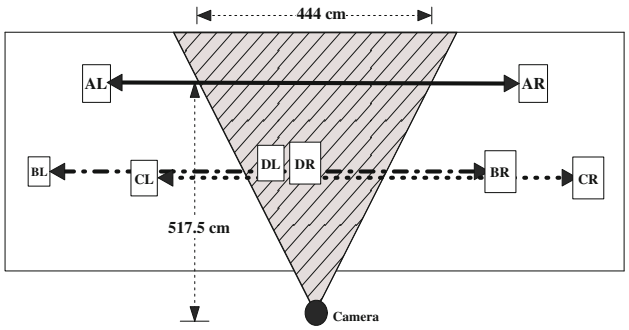

Fig. 7 Camera setup for recording

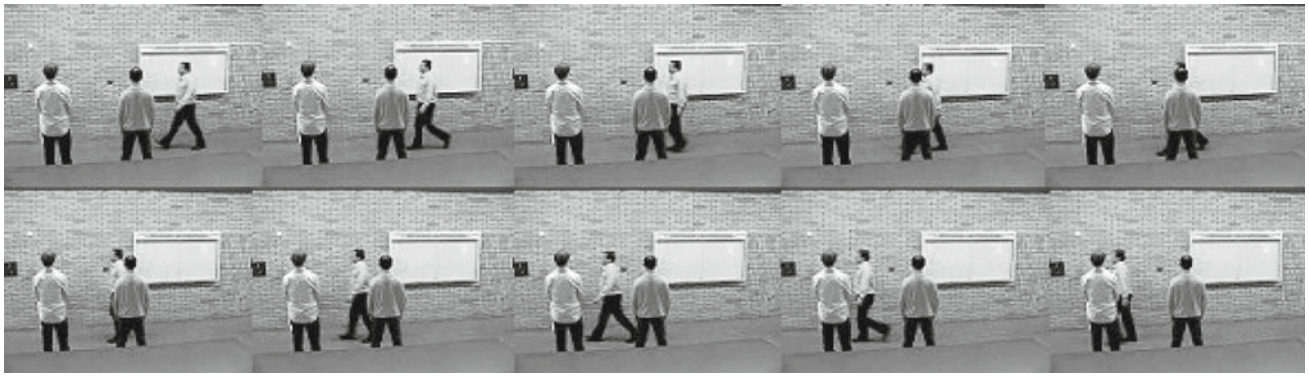

Fig. 8 A sequence of frames where a subject is occluded by static objects (static occlusion)

Fig. 2, where the subject of interest is in right-to-left motion. For static occlusion, we place two persons in points DR and DL. One such type of static occlusion sequence is shown in Fig. 8.

The movement of each subject is captured in three different situations. Each is first recorded in a regular non-occluded situation. The other two situations are dynamic and static occlusions. We capture a total of four sequences (two sequences for right-to-left motion and two sequences for left-to-right motion) for each subject in each situation. Thus, there are 12 sequences for each subject. The data set currently consists of 35 subjects.

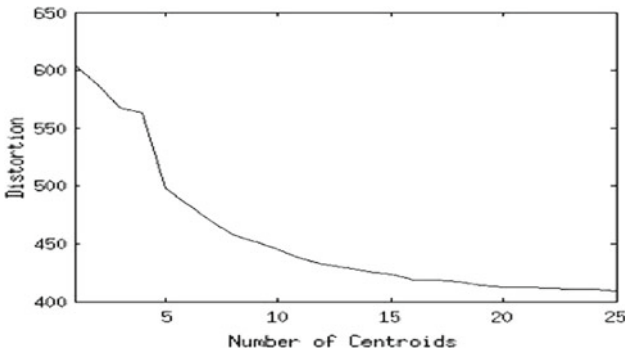

Fig. 9 Rate-distortion plot

6.1.2 Silhouette extraction

Once the videos are captured, moving objects are separated from background using Gaussian Mixture Model (GMM) [38]. We use the shadow elimination method proposed in Lu and Zhang [21] to get clean silhouettes. For removing spurious pixels and small holes inside the extracted human silhouette region, we apply morphological operators. First, dilation operator with 3×3 structuring element is applied, which fills up the holes and expands the silhouette region. To get back the original silhouette region, we use erosion operator with same structuring element. Finally, the silhouettes are normalized by height scaling and centering.

6.2 Results of key pose estimation and occlusion detection

Given a test silhouette sequence, where each silhouette frame is vertically scaled and horizontally aligned, the first step is to classify each silhouette into one of the key poses. If occlusion is present, then we classify those degraded silhouettes as occluded and the remaining as respective most probable key poses.

To decide the optimum number of key poses produced by K-means clustering, we consider the rate-distortion curve as used by Kale et al. [9]. Rate-distortion curve plots the average distortion as a function of number of clusters, an example of which is shown in Fig. 9. It can be observed from the plot that beyond sixteen clusters, the average distortion does not decrease significantly. Thus, for our experimentation using TUM-IITKGP data set, we choose sixteen clusters that give a set of sixteen key poses. From mean weight vectors $(P_1, P_2, \ldots, P_k)$, the key poses are reconstructed for visualization. Figure 10 shows the sixteen reconstructed key poses over one gait cycle obtained from the clustering.

Each of the key poses is associated with one state in the proposed state transition diagram as shown in Fig. 5. However, unlike only the three states shown in this example figure, we actually have sixteen states, one for each key pose and one for the occluded state. Based on this state transition diagram, the graph is constructed with seventeen nodes for each frame. The key pose detection problem is then solved as the most probable path search problem in this graph. The output of the key pose detection algorithm for a subject never occluded is shown in Fig. 11. The indexes indicate the key pose classes in which the corresponding frames are mapped.

Occlusion detection and key pose estimation in the presence of static occlusion are shown in Fig. 12. Here, the silhouettes are degraded by inclusion of background pixels. It can be observed from the figure that when the silhouette is

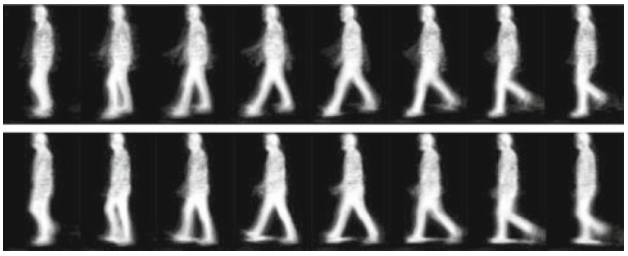

Fig. 10 Reconstructed key poses obtained from K-means clustering in eigen space

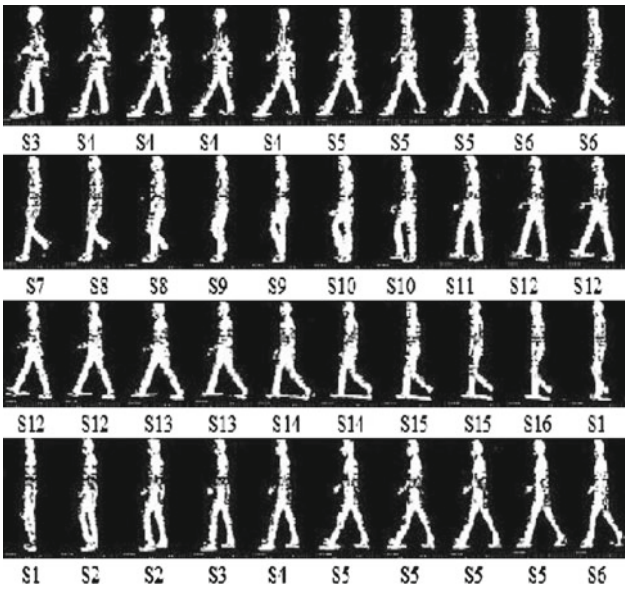

Fig. 11 Example mapped sequence for a subject never occluded. In the first gait cycle, starting from frame no. 1 (S3) and ending at frame no. 33 (S2), all the key poses are present, thus clean. Second gait cycle starts from frame no 34, but these remaining seven frames do not form a complete gait cycle

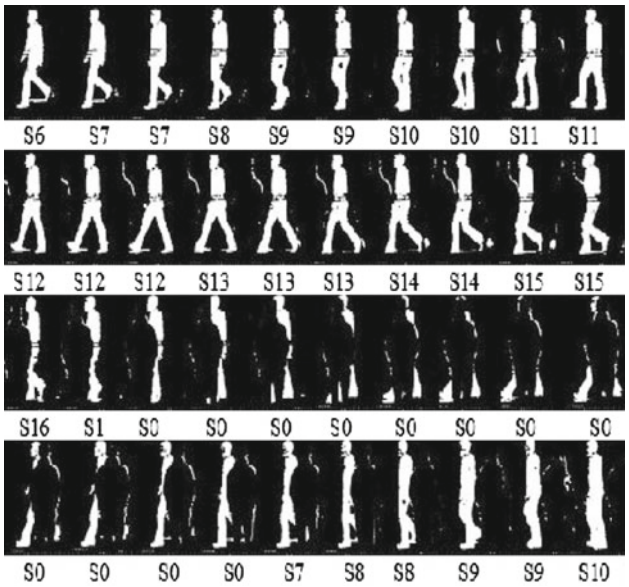

Fig. 12 Example mapped sequence for a subject occluded by static objects. First gait cycle starts from frame no. 1 (S6), but the end is overlapped with the next gait cycle due to occlusion. Thus both the gait cycles are detected as unclean

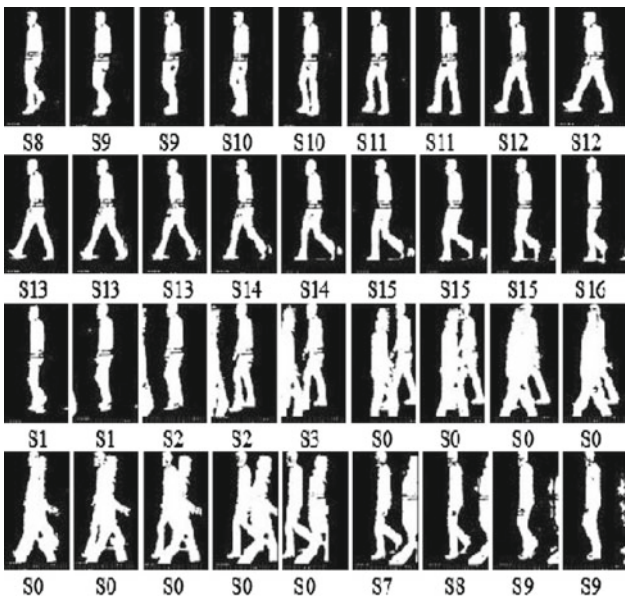

Fig. 13 Example mapped sequence for a subject occluded by dynamic objects. First gait cycle, starting from frame no. 1 (S8) and ending at frame no. 33 (S7), is detected as unclean as occluded poses are present or all the key poses are not present. Second gait cycle, starting from frame no. 34, is incomplete

partially occluded, then also our algorithm is able to predict the pose of the silhouette.

Figure 13 shows occlusion detection and key pose estimation result in case of dynamic occlusion. Here, the silhouettes are degraded by addition of foreground pixels. In this case, also our proposed algorithm performs reasonably well and predicts the partially occluded silhouettes.

Table 1 shows the accuracy of the proposed algorithm for key pose detection, occlusion detection, and partially occluded pose prediction. The ground truth is obtained manually. It can be observed that the algorithm detects the presence of occlusion with high accuracy. Moreover, it never detects unoccluded frames as occluded, i.e., false positive rate is zero (not shown in table). Key pose detection accuracy is also high. The frames for which key poses are classified incorrectly are mainly intermediate frames between two sequences of occluded frames. Since the temporal context information used by dynamic programming is low in those situation, frames are misclassified. Partially occluded poses are predicted reasonably well. It shows the robustness of the dynamic programming-based pose detection approach, which is able to handle silhouette degradation to a considerable extent.

Table 1 Pose detection result

	Key pose detection (%)	Occlusion detection (%)	Partially occluded pose prediction (%)
Static occlusion	93.0	99.0	82.0
Dynamic occlusion	94.0	98.0	86.0

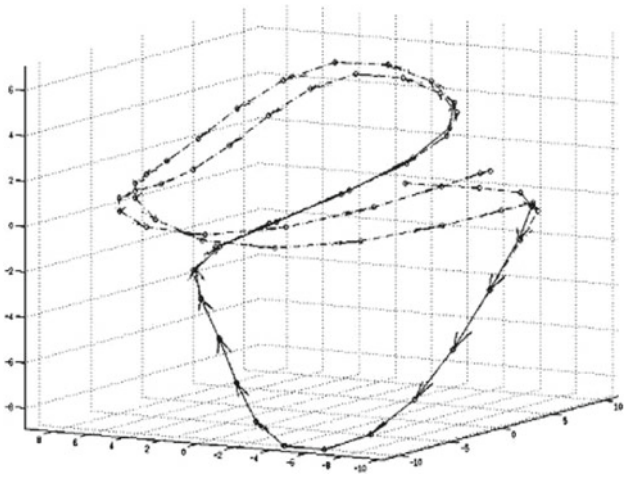

Fig. 14 Latent positions and corresponding trajectories learnt from a silhouette sequence of two gait cycles using BGPDM

Sect. 5.3. Figure 14 shows 3-dimensional latent trajectories obtained for a test sequence of silhouettes. The silhouette sequence contains two gait cycles of size 34 frames. Twenty silhouettes were occluded, thus considered as missing. The latent trajectory for the existing silhouette data is depicted by dash-dot curve, and the regular curve shows the estimated missing silhouette latent positions. It can be observed that the estimated positions closely follow the latent positions of existing data.

Figure 15 shows the reconstructed missing silhouettes obtained from reverse mapping of the estimated latent coordinates. The second row and fourth row show twenty reconstructed silhouettes of two different persons. Corresponding original silhouettes are shown in first row and third row. It can be observed that the predicted silhouettes are almost visually indistinguishable from the ground truth. In spite of this, to measure the numerical accuracy of reconstruction, we use Tanimoto similarity measure, commonly known as Tanimoto coefficient [39]. It computes the similarity between the original and reconstructed silhouette, I and I^* respectively, as follows:

$$T(I, I^*) = \frac{I^T I^*}{I^T I + I^{*T} I^* - I^T I^*} \tag{19}$$

6.3 Results of silhouette reconstruction

For reconstruction of the missing silhouettes, BGPDM is first trained with the normal walking sequences of ten subjects each of one gait cycle length. Then, this trained model is used for predicting the missing silhouette frames as discussed in

For two normalized overlapping binary silhouettes, this measure computes the ratio of the number of intersection pixels to the number of union pixels. The accuracy of the proposed silhouette reconstruction approach using Tanimoto similarity measure is found to be 90.7% for dynamic occlusion, and 88.9% for static occlusion.

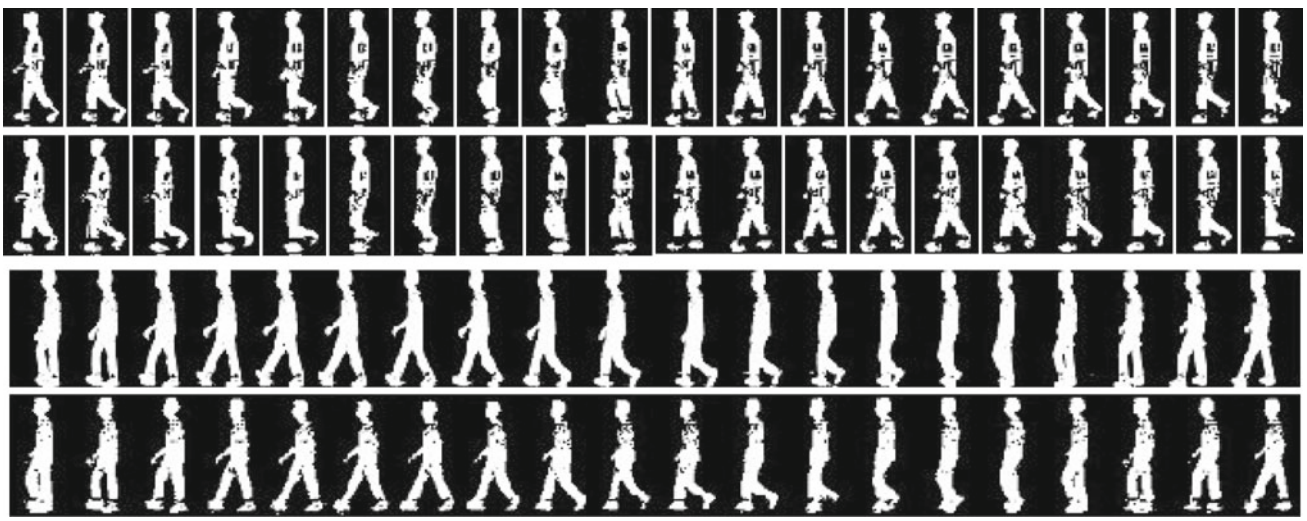

Fig. 15 Original silhouettes and reconstructed missing silhouettes of two persons. *First* shows the original silhouettes, and *second row* shows the reconstructed missing silhouettes of one subject. Similarly, *third row* shows original silhouettes, and *fourth row* shows the reconstructed silhouettes of another subject

6.4 Results on CMU MoBo data set with synthetic occlusion

We also used CMU's MoBo data set that has varying walking styles [33]. The data set consists of indoor video sequences of 25 subjects walking on a treadmill. Videos were captured in different modes of walking, i.e., walking on an inclined plane, walking with a ball in two hands, fast walk and slow walk. Each sequence is 11.33 s long, recorded at a frame rate of 30 frames per second.

For our experimentation, we consider sequences of 2 gait cycles long. Each sequence starts from the midstance mode after left leg forward position. Then, to introduce occlusion in these sequences, we consider three different degrees of occlusion, namely *low*, *medium*, and *severe*. To model low-degree occlusion situation, we degrade frames taking normal distribution with $\mathcal{N}(8, 3)$. For medium occlusion, frames are degraded according to $\mathcal{N}(12, 3)$, and for severe occlusion, frames are degraded according to $\mathcal{N}(17, 3)$. For each distribution, we generate 100 samples that indicate the number of silhouette frames degraded under this distribution. Degradation of silhouette frames is done in three different positions of the input sequence to observe the effect of occlusion on different parts of a gait cycle. The three selected positions are (i) start of the sequence (midstance mode to right leg forward position), (ii) middle of the sequence where the previous gait cycle ends and the next gait cycle starts (transition from left leg forward position to right leg forward position), and (iii) end of the sequence (left leg forward position to midstance mode). The degraded frames are treated as if the full silhouette is occluded by some large static object.

In the pose detection step, to determine the key poses, silhouettes from slow walk are used. Here also sixteen key poses are used. After getting the key poses constructed from slow walk sequences, the pose detection accuracy is evaluated for fast walk, walking on inclined plane, and walking with a ball in hand. In all cases, ground truth is obtained by manually annotating the silhouette images into one of the key poses. The result of pose detection with different degrees of occlusion and different walking types is shown in Table 2. Here, the result is accumulated over all the three different parts of the gait cycle where occlusion is introduced. Occluded frames are always detected correctly. However, pose detection accuracy varies with degree of occlusion. With increasing degree of occlusion, pose detection accuracy drops. Table 3 shows the result of pose detection for different positions of occlusion and different walking types where the results have been accumulated over all the three degrees of severity of occlusion. From the results, it can be observed that the position of occlusion does not have any clear impact on the pose detection accuracy.

For pose reconstruction, the same procedure mentioned in Sect. 6.3 is followed. Reconstruction accuracy is computed using Eq. 19, with respect to the original silhouettes, for varied positions and degrees of occlusion. The results are presented in Tables 4 and 5. It can be seen from Table 4 that the reconstruction accuracy degrades gracefully with increased degree of occlusion. Even during severe occlusion, accuracy is reasonably high. Reconstruction accuracy for walking on inclined plane is lower due to the presence of background noise in the lower leg region. From Table 5, it can be observed that the reconstruction accuracy varies with position of occlusion. However, the variation is less for fast and slow walk, while it is slightly higher for walking in inclined plane and for walking with ball in hand.

Table 2 Pose detection result for different degrees of occlusion

	Fast walk (%)	Slow walk (%)	Incline (%)	With ball (%)
$\mathcal{N}(8, 3)$	91.2	95.6	88.8	92.9
$\mathcal{N}(12, 3)$	90.1	94.8	87.5	91.3
$\mathcal{N}(17, 3)$	87.8	94.1	86.3	90.8

Table 3 Pose detection result for different positions of occlusion

	Fast walk (%)	Slow walk (%)	Incline (%)	With ball (%)
Start position	90.1	93.3	86.6	90.2
Middle position	89.7	95.8	88.2	93.5
End position	89.4	95.4	87.7	91.4

Table 4 Silhouette reconstruction result for different degrees of occlusion

	Fast walk (%)	Slow walk (%)	Incline (%)	With ball (%)
$\mathcal{N}(8, 3)$	92.6	91.8	90.2	91.9
$\mathcal{N}(12, 3)$	91.9	91.5	89.8	90.7
$\mathcal{N}(17, 3)$	91.6	91.3	89.6	90.5

Table 5 Silhouette reconstruction result for different positions of occlusion

	Fast walk (%)	Slow walk (%)	Incline (%)	With ball (%)
Start position	91.8	91.3	90.7	92.1
Middle position	92.1	91.4	90.0	90.8
End position	92.3	92.0	89.1	90.7

7 Conclusions

Automated identification of humans from their gait is a challenging research problem. In this paper, we have considered situations where the subject gets occluded due to the presence of multiple objects in the field of view of camera, which is quite common in real-world surveillance scenarios. The

problem is quite challenging and is yet to receive enough attention. We have proposed a novel approach for detecting the presence of occlusion in a sequence of silhouette frames and their subsequent reconstruction. A dynamic programming- based maximum likelihood key pose detection algorithm simultaneously detects key pose class for each frame and also identifies the occluded frames. Clean and unclean gait cycles are segregated as the algorithm output. If all the subsequences of frames corresponding to a gait cycle are degraded by occlusion, then none of the existing methods can be used for recognition from this sequence. The need for reconstruction of the degraded silhouette frames to construct clean gait cycles becomes pertinent in such situations. A novel method based on BGPDM has been suggested in this paper, which is able to reconstruct the missing silhouettes considerably well. The reconstructed silhouettes can then be used for recognition using any of the existing methods.

We tested our algorithms on a new data set (TUM-IITKGP) featuring occlusion by static objects as well as dynamic objects. The result shows that the reconstruction accuracy is around 90%. We also evaluated the proposed algorithms on CMU's MoBo data set by synthetically introducing varied degree of occlusion. Here also we observed that the method can reconstruct silhouettes with high accuracy. Thus, it can be concluded that the proposed approach has the potential to solve the challenges that arise during recognition of humans using gait in the presence of occlusion.

Future work would involve further studies with more variations of occlusion situations, effect of camera positions, testing with larger data sets, and use of multiple cameras for occlusion reduction.

Acknowledgments This work is partially supported by the project grant 1(23)/2006—ME TMD, Dt. 07/03/2007, sponsored by the Ministry of Communication and Information Technology, Govt. of India, and also by Alexander von Humboldt Fellowship for Experienced Researchers. The authors would like to thank Martin Hofmann for his help in preparing the TUM-IITKGP dataset.

References

1. Larsen, P.K., Simonsen, E.B., Lynnerup, N.: Gait analysis in forensic medicine. J. Forensic Sci. **53**(5), 1149–1153 (2008)
2. Cunado, D., Nixon, M.S., Carter, J.N.: Automatic extraction and description of human gait models for recognition purposes. Proc. CVIU **90**(1), 1–41 (2003)
3. Bobick, A., Johnson, A.: Gait recognition using static, activity-specific parameters In: Proceedings of the IEEE Conference on CVPR, vol. 1, pp. 423–430 (2001)
4. Yam, C., Nixon, M.S., Carter, J.N.: Automated person recognition by walking and running via model-based approaches. Pattern Recognit. **37**(5), 1057–1072 (2004)
5. Jain, A., Dube, T., Ghosh, D.: A fuzzy approach to person identification using gait. In: Proceedings of the IET International Conference on Visual Information Engineering (VIE), pp. 174–179 (2006)
6. Zhang, R., Vogler, C., Metaxas, D.: Human gait recognition at sagittal plane. Image Vis. Comput. **25**(3), 321–330 (2007)
7. Lu, H., Plataniotis, K.N., Venetsanopoulos, A.N.: A full-body layered deformable model for automatic model-based gait recognition. EURASIP J. Adv. Signal Process. 2008(Article ID 261317) (2008). doi:10.1155/2008/261317
8. Huang, X., Boulgouris, N.V. Model-based human gait recognition using fusion of features. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 0, pp. 1469–1472 (2009)
9. Kale, A., Sundaresan, A., Rajagopalan, A.N., Cuntoor, N.P., Roy-Chowdhury, A.K., Kruger, V., Chellappa, R.: Identification of humans using gait. IEEE Trans. Image Process. **13**, 1163–1173 (2004)
10. Kale, A., Rajagopalan, A., Cuntoor, N., Krueger, V.: Gait-based recognition of humans using continuous HMMs. In: Proceedings of the International Conference on Automatic Face and Gesture Recognition, pp. 321–326 (2002)
11. Chen, C.H., Liang, J., Zhao, H., Hu, H., Tian, J.: Factorial HMM and parallel HMM for gait recognition. IEEE Trans. Syst. Man Cybern. C Appl. Rev. **39**(1), 114–123 (2009)
12. Sundaresan, A., Roy-Chowdhury, A.K., Chellappa, R.: A hidden markov model based framework for recognition of humans frem gait sequences. In: Proceedings of the IEEE Conference Image Processing, vol. 2, pp. 93–99 (2003)
13. Niyogi, S.A., Adelson, E.H.:Analyzing and recognizing walking figures in XYT. In: Proceedings of the CVPR, pp. 469–474 (1994)
14. Little, J., Boyd, J.: Recognizing people by their gait: the shape of motion. Videre J. Comput. Vis. Res. **1**(2), 1–32 (1998)
15. BenAbdelkader, C., Cutler, R., Davis, L.: Motion-based recognition of people in eigengait space. In: Proceedings of International Conference on Automatic Face and Gesture Recognition, pp. 267–274 (2002)
16. Vega, I., Sarkar, S.: Experiments on gait analysis by exploiting nonstationarity in the distribution of feature relationships. In: Proceedings of the International Conference on Pattern Recognition, vol. 1, pp. 1–4 (2002)
17. Sarkar, S., Phillips, P.J., Liu, Z., Robledo-Vega, I., Grother, P., Bowyer, K.W.: The human ID gait challenge problem: data sets, performance, and analysis. IEEE Trans. Pattern Anal. Mach. Intel. **27**(2), 162–177 (2005)
18. Hayfron-Acquah, J., Nixon, M., Carter, J.: Human identification by spatio-temporal symmetry. In: Proceedings of the International Conference on Pattern Recognition, vol. 1, pp. 632–635 (2002)
19. Lee L., Grimson, W.: Gait analysis for recognition and classification. In: Proceedings of the International Conference on Automatic Face and Gesture Recognition, pp. 155–162 (2002)
20. Boulgouris, N.V., Hatzinakos, D., Plataniotis, K.N.: Gait recognition: a challenging signal processing technology for biometrics identification. IEEE Signal Process. Mag. **22**(6), 78–90 (2005)
21. Lu, J., Zhang, E.: Gait recognition for human identification based on ICA and fuzzy SVM through multiple views fusion. Pattern Recognit. Lett. **28**, 2401–2411 (2007)
22. Nixon, M.S., Carter, J.N.: Advances in automatic gait recognition. In: Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, pp. 139–144 (2004)
23. Han, J., Bhanu, B.: Individual recognition using gait energy image. IEEE Trans. Pattern Anal. Mach. Intel. **28**(2), 316–322 (2006)
24. Yanga, X., Zhoua, Y., Zhanga, T., Shua, G., Yanga, J.: Gait recognition based on dynamic region analysis. Signal Process. **88**(9), 2350–2356 (2008)

25. Chen, C., Liang, J., Hu, H., Tian, J.: Frame difference energy image for gait recognition with incomplete silhouettes. Pattern Recognit. Lett. **30**(11), 977–984 (2009)
26. Zhanga, E., Zhao, Y., Xionga, W.: Energy image plus 2DLPP for gait recognition. Signal Process. **90**(7), 2295–2302 (2010)
27. Pullen, K., Bregler, C.: Motion capture assisted animation: texturing and synthesis. In: Proceedings of theof SIGGRAPH, pp. 501–508 (2002)
28. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models for human motion. IEEE Trans. PAMI **30**(2), 283–298 (2008)
29. Lawrence, N.D.(2004) Gaussian process latent variable models for visualisation of high dimensional data. In: Thrun, S., Saul, L., Schlkopf, B. (eds.). Advances in Neural Information Processing Systems, pp. 329–336 MIT Press, Cambridge, MA
30. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
31. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models. In: Proceedings of the NIPS, pp. 1441–1448 (2005)
32. Urtasun, R., Fleet, D.J., Fua, P.: 3D people tracking with Gaussian process dynamical models. In: Proceedings of the CVPR, pp. 238–245 (2006)
33. Gross, R., Shi, J.: The CMU Motion of Body (MoBo) Database. Technical report CMU-RI-TR-01-18, Robotics Institute, Carnegie Mellon University (2001)
34. Center for biometrics and security research, CASIA. http://www.cbsr.ia.ac.cn
35. Turk, M., Pentland, A.: Eigenfaces for recognition. J. Cogn. Neurosci. **3**(1), 71–86 (1991)
36. Hofmann, M., Sural, S., Rigoll, G.: Gait recognition in the presence of occlusion: a new dataset and baseline algorithms. In: Proceedings of the International Conference on Computer Graphics, Visualization and Computer Vision (WSCG). Plzen, Czech Republic (2011)
37. http://www.mmk.ei.tum.de/~hom/tumiit/tumiitgait.html
38. Staufferand, C., Grimson, W.E.L.: Adaptive background mixture modelsfor real-time tracking. In: Proceedings of the CVPR, pp. 246–252 (1999)
39. Tanimoto, T.T.: IBM Internal Report, 17 Nov (1957)