

Article

2.5D Multi-View Gait Recognition Based on Point Cloud Registration

Jin Tang ¹, Jian Luo ¹, Tardi Tjahjadi ² and Yan Gao ^{1,*}

¹ School of Information Science and Engineering, Central South University, Changsha 410083, China; E-Mails: tjin@csu.edu.cn (J.T.); delphifx@csu.edu.cn (J.L.)

² School of Engineering, University of Warwick, Gibbet Hill Road, Coventry CV4 7AL, UK; E-Mail: t.tjahjadi@warwick.ac.uk

* Author to whom correspondence should be addressed; E-Mail: gaoyan@csu.edu.cn; Tel./Fax: +86-0731-888-79463.

Received: 14 January 2014; in revised form: 24 March 2014 / Accepted: 24 March 2014 /

Published: 28 March 2014

Abstract: This paper presents a method for modeling a 2.5-dimensional (2.5D) human body and extracting the gait features for identifying the human subject. To achieve view-invariant gait recognition, a multi-view synthesizing method based on point cloud registration (MVSM) to generate multi-view training galleries is proposed. The concept of a density and curvature-based Color Gait Curvature Image is introduced to map 2.5D data onto a 2D space to enable data dimension reduction by discrete cosine transform and 2D principle component analysis. Gait recognition is achieved via a 2.5D view-invariant gait recognition method based on point cloud registration. Experimental results on the in-house database captured by a Microsoft Kinect camera show a significant performance gain when using MVSM.

Keywords: gait; person identification; 2.5D modeling; point cloud registration

1. Introduction

Gait recognition is a means of using the behavioral biometrics of gait to identify a human subject. Gait is difficult to disguise and can be easily observed in low-resolution video sequences. The need for a means for counter-terrorism, security and medical-related subject behavior analysis makes accurate modeling of human gait and effective extraction of gait signatures for view-invariant subject

identification have significant theoretical and practical value. For example Chowdhury and Tjahjadi [1] proposed a gait recognition method that combines spatio-temporal motion characteristics, statistical and physical parameters of a human subject to achieve robustness and high accuracy in subject identification.

In surveillance applications, most of the challenging factors that affect existing gait recognition systems [2], e.g., variation in human walking posture for different camera views, make the performance of a gait recognition method that is designed to operate on a particular camera view degrade significantly for other views. Furthermore, for gait recognition to be used in surveillance applications, it is impractical to use many cameras to achieve multi-view gait recognition. Thus, achieving view-invariant gait recognition has become a major challenge.

There are several approaches to view-invariant gait recognition. One approach is to reconstruct 3-dimensional (3D) gait models using a calibrated multi-camera system and extract 3D gait features. Shakhnarovich *et al.* [3] explored the use of an image-based visual hull to reconstruct the 3D model and rotate the model to realize view-invariant gait recognition. Gu *et al.* [4] proposed viewpoint-free gait recognition from recovered 3D human joints. Sivapalan *et al.* [5] proposed the use of a 3D voxel model derived from multi-view silhouette images. However all current examples of 3D modeling of the human body are mostly based on images from multiple cameras. Due to the need for multiple equipment and the increased complexity of the resulting recognition algorithm, such an approach is usually only feasible under laboratory conditions. In addition, although radar (e.g., laser radar) can also be used for the 3D modeling, the resolution of the resulting model is low.

The second approach is to use view transformation model (VTM) to achieve multi-view gait recognition. VTM transforms gait features from different views onto the same view. The VTM is constructed by decomposing a matrix comprising features from different views and of different subjects into subject-independent matrix and view-independent matrix. Makiyara *et al.* [6] used VTM to transform gallery features onto the same view for multi-view gait recognition. Muramatsu *et al.* [7] proposed an arbitrary gait view transformation scheme using 3D gait database and VTM method. Kusakunniran *et al.* [8,9] developed the VTM model by using correlated motion regression and multi-layer perceptron. Although the VTM gait recognition approach demonstrates the advantages of multi-view gait recognition, it requires multiple-view images to generate VTM. Furthermore, the model accuracy is determined by the number of multi-view gaits used in the VTM construction.

The third approach is to use a multi-view fusion classifying method. By fusing gait classification from multi-view data captured by multiple cameras, view-invariant gait recognition is realized. For example, Nizami *et al.* [10] explored the use of Extreme Learning Machine (ELM) multiclass classifier for classification, and the results are fused at score level subject to some fusion rules to realize the view-independent gait recognition. However, the ELM based system does not address the problem of using multi-view images. To address this problem, Jean *et al.* [11] proposes an approach to compute view-normalized body part trajectories. The normalized trajectories are extracted as view-invariant gait feature for gait recognition. However human gait information cannot be fully represented using only trajectories of head and feet. Thus, when the gait views are significantly changed or self-occlusion is encountered, the method performs poorly.

To address the above-mentioned problems, in this paper we propose the use of a single Kinect camera to obtain point cloud data of a human body and construct 2.5D voxel gait model that includes

only one-side surface portion of the human body. A point cloud registration method is proposed to synthesize multi-view gait features using two reconstructed gait models from two different views. Dense point cloud and view-invariant Gaussian curvature are extracted to represent the gait features. The 2.5D data is mapped onto the 2D space, and Gaussian curvature based gait color images are used to facilitate the gait feature extraction, classification and identification of the human subject.

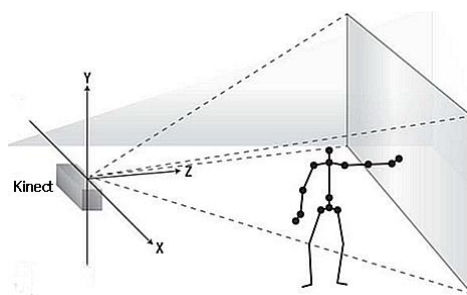
This paper is organized as follows: Section 2 presents the construction of 2.5D gait voxel model using a Kinect device with point cloud data simplification. Section 3 presents point cloud registration for multi-view 2.5D gait voxel model. Section 4 introduces the extraction of 2.5D gait features and the multi-view gait recognition method of the proposed gait recognition system. Section 5 presents our experimental results and Section 6 concludes the paper.

2. Preliminary Steps

2.1. Construction of 2.5D Gait Voxel Model

2.5D data that contains depth information is used to construct gait surface voxel model, and a Kinect is used to capture the 2.5D data which is a simplified 3D (x,y,z) surface representation (Figure 1). 2.5D data contains at most one depth value $d(x,y)$ which denotes the distance between the RGB image pixel (x,y) of a point on the body surface and the Kinect. 2.5D is a suitable trade-off solution between 2D and 3D approaches. It is restricted to a given viewpoint that is called 2.5D information [12].

Figure 1. World coordinates of the Kinect sensor-based system.



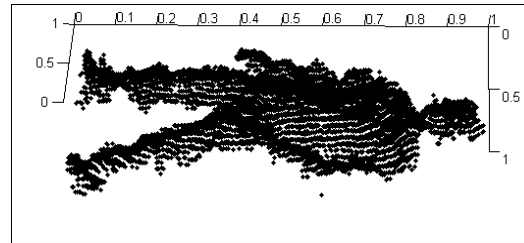
As a 3D measuring device, Kinect comprises an IR pattern projector and an IR camera. It can output three different images: IR image, RGB image and Depth image. The 2.5D data of the depth image and RGB image are used to construct a 3D voxel model for a given viewpoint by calculating all the 3D points from the measurement (x,y,d) in the depth image. 3D point cloud data are calculated using the Kinect geometrical model [13], *i.e.*:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{c_1 d + c_0} \text{dis}^{-1} \left(K^{-1} \begin{bmatrix} x + u_0 \\ y + v_0 \\ 1 \end{bmatrix}, k \right) \quad (1)$$

where d is depth value along the z -axis, c_1 and c_0 are parameters of the model, u_0 and v_0 are respectively the shifted parameters of IR and depth images, dis is distortion function, k is distortion parameter of the Kinect IR camera and K is the IR camera calibration matrix.

Before constructing the 2.5D gait point model, gait silhouettes are extracted from the depth image by foreground subtraction and frame difference methods [14]. The gait silhouettes and RGB images are then used to calculate all the 3D point cloud data for the gait using Equation (1). The 3D point cloud gait model is constructed for a given viewpoint by normalizing all the gait point cloud data to 3D space. Since only a single Kinect depth camera is used, the gait point cloud data includes only one side surface portion of the human body as shown in Figure 2. We call it a 2.5D voxel model.

Figure 2. The normalized point cloud data of human body.



2.2. Point Cloud Data Simplification for Gait Voxel Model

Since the point cloud data is large, it is simplified while preserving its features. This is achieved by using curvature features of the point cloud by Hausdorff distance [15]. A bounding box method is first used to derive the relationship between a point cloud data P and its K nearest neighbors. Denote the two principal curvatures of P and one its neighboring points respectively as $\{K_1^P, K_2^P\}$ and $\{K_1^Q, K_2^Q\}$. The Hausdorff distance H of the two data sets is:

$$H = \max_{i=1,2} \min_{j=1,2} \left(\frac{\|K_i^P - K_j^Q\|}{\|K_i^P\| + \|K_j^Q\|} \right) \quad (2)$$

The Hausdorff distance is defined for P as $H^P = \max(H^Q), Q=1,2,\dots,k$. By calculating the Hausdorff distance of every point within the bounding box, a threshold ε is selected to remove the less important point cloud data, and thus complete the data point simplification as shown in Figure 3. The choice of the threshold ε directly influences the efficiency of the simplification and the computational cost of the algorithm. A bigger ε will reduce computational cost but less simplifying efficiency, while a smaller ε has the opposite effect. We conducted experiments to determine the optimum ε value. 2.5D gait voxel models are selected for point cloud simplifying experiment and ε is set to 10^{-6} , 10^{-5} , 10^{-4} , 10^{-3} and 10^{-2} .

Figure 3. Simplification of point cloud data: (a) raw data; (b) $\varepsilon = 10^{-6}$; (c) $\varepsilon = 10^{-5}$; (d) $\varepsilon = 10^{-4}$; (e) $\varepsilon = 10^{-3}$; and (f) $\varepsilon = 10^{-2}$.

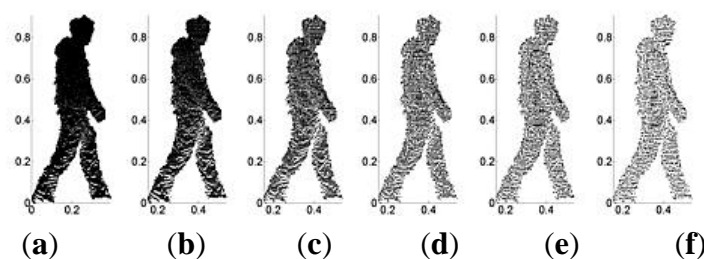


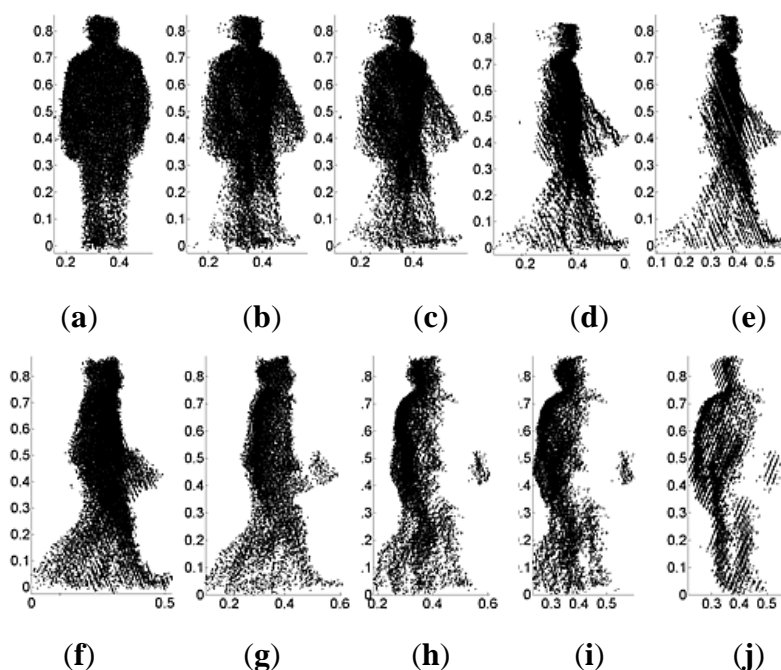
Figure 3a shows a raw gait point cloud data (including 25,862 point cloud data) before simplification, and Figure 3b–f is the results after simplification with 13,286, 8,392, 6,381, 4,592 and 2,392 point cloud data, respectively. The computational times are 518, 432, 327, 273 and 228 ms, respectively. From the experiment results, we set $\varepsilon = 10^{-4}$ with mean computational time and sufficient simplification.

3. Point Cloud Registration for Multi-View 2.5D Gait Voxel Models

3.1. Overview

Since the 2.5D gait voxel model is constructed from data captured using a single Kinect camera, view-invariant gait recognition cannot be realized by just rotating the model to obtain gait features for different views due to self-occlusion as illustrated in Figure 4.

Figure 4. Self-occlusion caused by rotating 2.5D gait model: (a–e) 0° view rotated counterclockwise; and (f–j) 90° view rotated clockwise.



In order to realize multi-view gait recognition and overcome the self-occlusion problem with 2.5D gait models, a point cloud registration method is proposed to synthesize different view gait features. And two training galleries with θ_{\min} and θ_{\max} views are used. Let P_{\min} be the 2.5D gait point cloud data for θ_{\min} view and P_{\max} for θ_{\max} view where $P_{\min} \cap P_{\max} \neq \Phi$. P_{\min} is registered with P_{\max} , and the new β view point cloud data after registration is represented by:

$$S_{\beta} = F_{reg}(P_{\min}, P_{\max}) = (P_{\min} \cdot R_{(\theta_{\min} \rightarrow \beta)} + T_1) \cup (P_{\max} \cdot R_{(\theta_{\max} \rightarrow \beta)} + T_2) \quad (3)$$

If $\hat{\alpha}$ is set to θ_{\min} then:

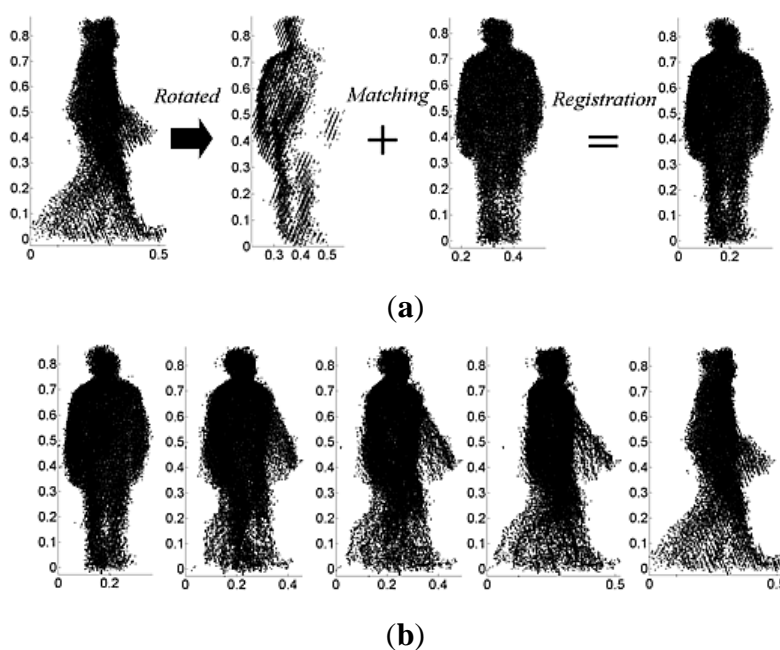
$$S_{\beta} = S_{\theta_{\min}} = P_{\min} \cup (P_{\max} \cdot R_{(\theta_{\max} \rightarrow \theta_{\min})} + T) \quad (4)$$

where $R_{(\theta_{\max} \rightarrow \beta)}$ denotes 3D rotation matrix from view θ_{\max} to \hat{a} , and T denotes the translation matrix. Based on the registered point cloud data $S_{\hat{a}}$ with \hat{a} view, the 2.5D gait data between θ_{\min} and θ_{\max} views can be determined without encountering any self-occlusion problems using:

$$P_{\theta} = S_{\beta} \cdot R_{(\beta \rightarrow \theta)} \quad (5)$$

The self-occlusion problem is addressed by the point cloud registration and 2.5D curvature features extraction method as illustrated in Figure 5. This is why in our approach only two different view galleries are needed for multi-view gait training and recognition.

Figure 5. Self-occlusion problem overcome by point cloud registration method: (a) gait point cloud registration with $\theta_{\min} = 0^{\circ}$, $\theta_{\max} = 90^{\circ}$ and $\hat{a} = 0^{\circ}$, and (b) view rotated from registered gait without self-occlusion.



3.2. Gait Point Cloud Alignment in a Cycle

Gait features are represented by a complete cycle as shown in Figure 6. It can be seen that a gait cycle must include several dynamic frames. 2.5D gait models are reconstructed by data from corresponding frames. Therefore point cloud registration cannot be made directly between two random 2.5D models with different views and cycles. Most often, the number of frames in a gait cycle is different, e.g., in Figure 6 the sample gait with 0° has eight frames while the corresponding gait with 90° has only seven frames. It is thus difficult to directly register models between two different view cycles. In order to overcome these problems, only the head point cloud data as shown in Figure 7 is used to calculate the rotation matrix and translation matrix for registration. The resulting matrices are used in registration process between full gait models. The first step is to align the gait point cloud data in the same cycle by using the centroids of head point cloud data. The head information is used because it is static information when compared with legs and arms.

Figure 6. Gait point cloud alignment and mixed gait modeling in a cycle: (a) 0° gait view; and (b) 90° gait view.

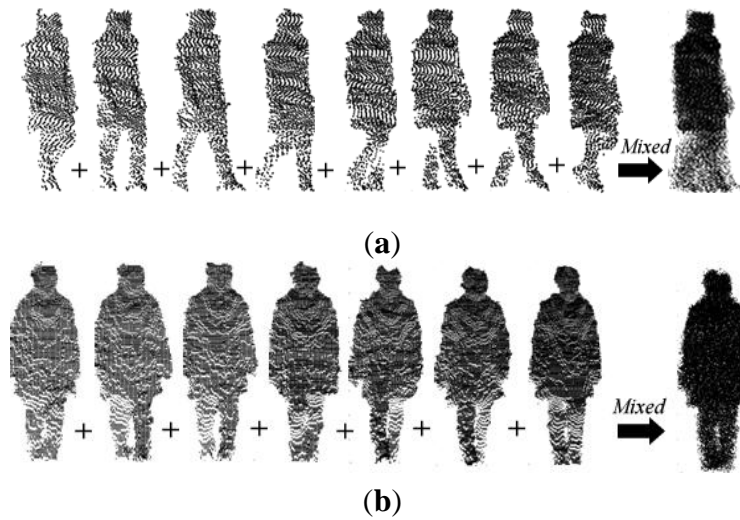
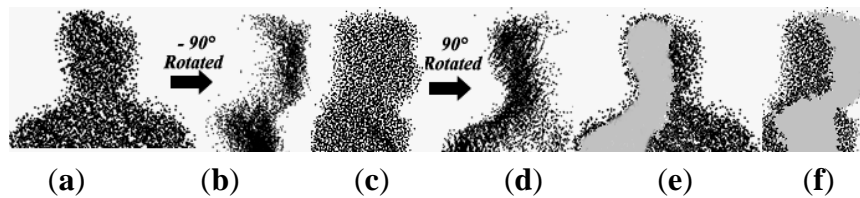


Figure 7. Mixed head point cloud: (a) 0° head view; (b) −90° rotated; (c) 90° head view; (d) 90° rotated; (e): overlapped areas of (a) and (d); and (f) overlapped areas of (b) and (c).



Assume that after gait phase estimation N frames in a gait cycle are extracted, and the corresponding 2.5D gait models are constructed. The 2.5D gait models are denoted as $P_\theta = \{p_\theta^i, i=1 \dots N\}$. The extracted head models are denoted by $H_\theta = \{h_\theta^i, i=1 \dots N\}$. The centroids of all head models are calculated as $\mu(H_\theta) = \{\mu(h_\theta^i), i=1 \dots N\}$. In order to complete alignment, the first gait model is set as reference. The translation matrices that align with the first gait model are then calculated using $T^i = \mu(h_\theta^i) - \mu(h_\theta^1), i=1 \dots N$. The final mixed 2.5D gait model after alignment is given by $M_\theta = \{\cup(p_\theta^i + T^i), i=1 \dots N\}$.

3.3. Gait Point Cloud Registration

After gait point cloud alignment in a cycle, a mixed 2.5D gait model is obtained from different gait models in a gait cycle, which represents the 2.5D gait features. Gait point cloud registration is then conducted between two mixed gait models of different views.

In order to complete point cloud registration between two 2.5D mixed gait models with $\hat{a} = \theta_{\min}$ in Equation (4), the rotation matrix $R_{(\theta_{\max} \rightarrow \theta_{\min})}$ and translation matrix T need to be determined using iterative closest point algorithm (ICP) [16]. Let H_{\min}^{mix} be the head point cloud in θ_{\min} mixed training model and $L_{\min} = \{l_{\min}^i \in R^3, 1 \leq i \leq n\}$ represents the overlapped area of head point cloud data with θ_{\max} view. H_{\max}^{mix} denotes the head point cloud data in θ_{\max} mixed training model, and $L_{\max} = \{l_{\max}^i \in R^3, 1 \leq i \leq m\}$ represents the overlap area of head point cloud data with θ_{\min} view.

There are common areas of two different view head surfaces as shown in Figure 7. The accurate detection of the overlapped region will aid the gait point cloud registration. The optimization process that determines the rotation matrix $R_{(\theta_{\max} \rightarrow \theta_{\min})}$ and translation matrix T is a nonlinear least squares optimization [17], i.e.:

$$\min E(g) = \sum_{i=1}^{\min(m,n)} \|g l_{\max}^i - l_{\min}^i\|^2 \quad (6)$$

Let \tilde{g} be the optimum solution, then the point cloud set $L'_{\max} = \{l_{\max}^i = \tilde{g} l_{\max}^i, 1 \leq i \leq n\}$ has the same centroid with L_{\min} , and let $\mu_{(L_{\min})} = \mu_{(L'_{\max})}$. We then calculate the centroid of L_{\min} and L_{\max} as $\mu_{(L_{\min})}$ and $\mu_{(L_{\max})}$, where $\mu_{(L'_{\max})} = \tilde{g} \mu_{(L_{\max})}$. Let $d_i = l_{\max}^i - \mu_{(L_{\max})}$, $d'_i = l_{\min}^i - \mu_{(L_{\min})}$, Equation (6) then becomes:

$$\min E(R) = \sum_{i=1}^{\min(m,n)} \|R d_i - d'_i\|^2 \quad (7)$$

The optimization then decomposes into determining the rotated matrix R , and calculating the translation matrix $T = \mu_{(L_{\min})} - R \mu_{(L_{\max})}$. Singular value decomposition (SVD) method is used to calculate R as follows. First, the covariance matrix D between L_{\min} and L_{\max} is calculated as:

$$D = \sum_{i=1}^{\min(m,n)} d_i d_i^T \quad (8)$$

The matrix D is then decomposed by SVD, and let $D = UVV^T$, $X = VU^T$ and:

$$A = \begin{cases} X & \det(U) \det(V) \geq 0 \\ \text{diag}(1, 1, -1) & \det(U) \det(V) < 0 \end{cases} \quad (9)$$

The determinant of U is denoted by $\det(U)$, and $\text{diag}(1, 1, -1)$ denotes the 3×3 matrix that has diagonal values of 1, 1, -1 . If $\text{rank}(D) \geq 2$, then R and T are respectively given by:

$$R = U A V^T \quad (10)$$

$$T = \mu_{(L_{\min})} - R \mu_{(L_{\max})} \quad (11)$$

L_{\max} point cloud data is then transformed using R and T . After transformation a new point cloud data set L'_{\max} is obtained by $L'_{\max} = R \cdot L_{\max} + T$.

We then repeat the previous steps to calculate new R and T by conducting iterative transformation until the square distance error $e = \sum_{i=1}^{\min(m,n)} \|R_k l_{\max}^i + T_k - l_{\min}^i\|^2$ satisfies the smallest requirements. The matrices R and T are then used to construct the final registration gait model:

$$S_{\beta} = S_{\theta_{\min}} = M_{\theta_{\min}} \cup (M_{\theta_{\max}} \cdot R_{(\theta_{\max} \rightarrow \theta_{\min})} + T) \quad (12)$$

The key of the algorithm is to detect the overlapped region of two views accurately and construct the covariance matrix. The method to determine the matching point set is as follows. First, we calculate the Gaussian curvature K of the point cloud data for the head part of 2.5D gait models H_{\min}^{mix} and H_{\max}^{mix} . Gaussian curvature is invariant to the affine transformation and is used as the basis of the matching. The similar point cloud data are then searched between H_{\min}^{mix} and H_{\max}^{mix} . Let $\forall q_i \in H_{\min}^{mix}, p_i \in H_{\max}^{mix}$, and we define the curvature distance between two point cloud data as

$Dis(q_i, p_i) = |K(q_i) - K(p_i)|$. The similar point cloud data is then determined and forms the matching point set:

$$S = \{(p_i, q_i) \mid q_i \in H_{\min}^{mix}, p_i \in H_{\max}^{mix}, \delta < Dis(p_i, q_i) < \varepsilon\} \quad (13)$$

where $\varepsilon = \frac{1}{2}(\max Dis(q_i, p_i) + \min Dis(q_i, p_i))$ and $\delta = \frac{1}{N} \sum_{i=1}^n |K(q_i)|$.

Since one point cloud data may be similar to many point cloud data in another point cloud data set, one to one correspondence analysis are performed by matching similar triangles. We first select three points in S from $H_{\theta_{\min}}^{mix}$, and search the most similar three points in S from H_{\max}^{mix} using similar triangles, where the distance between two triangles is

$$T_{dis} = \frac{1}{3} \times \sum_{1 \leq i < j \leq 3} \frac{\|p_i p_j\| - \|q_i q_j\|}{\|p_i p_j\| + \|q_i q_j\|} \quad (14)$$

The points with the smallest distance are selected as matching cloud point. When all the points are matched, the covariance matrix D is determined for computing the matrices R and T . to achieve two view gait point cloud model registration as Equation (12). The registration model can then be rotated to obtain gait features for different views to achieve view-invariant gait recognition using Equation (5).

4. 2.5D Gait Features Extraction and Multi-View Gait Recognition

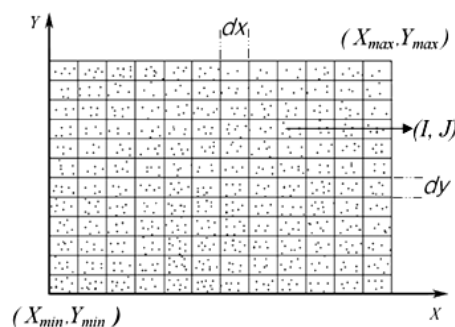
4.1. 2.5D Gait Features Extraction

The density of point cloud is utilized to extract silhouette data of the human subject, and Gaussian curvature and mean curvature [18] are used to extract 2.5D gait features. The color gait curvature image (CGCI) for gait recognition is formed by mapping the 2.5D gait features to a color gait image.

4.1.1. Point Cloud Density, Gaussian Curvature, Mean Curvature and CGCI

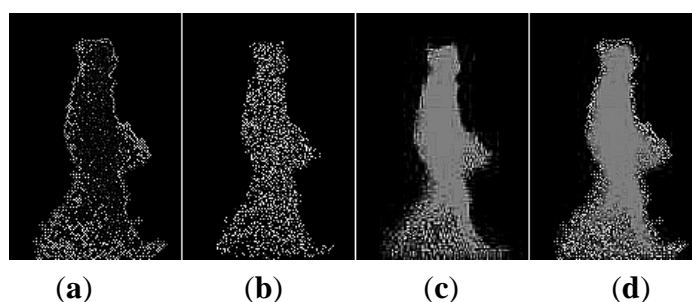
The normalized point cloud data is first projected onto the XY plane into $N_I \times N_J$ blocks as shown in Figure 8, where dx and dy are respectively the horizontal and vertical sampling intervals. Each point cloud data is located in the corresponding block (I, J) , and each block may have several point cloud data. In order to extract the silhouette information of the gait, the density of point cloud in each block (I, J) is calculated first, denoted by $Density(I, J)$.

The curvature information is extracted next. Since only a single Kinect camera is used, the point cloud data includes only one side surface portion of the human body, and the surface point set is denoted by $S = [x, y, z], (x, y) \in D$, where D is the projected grid from 2.5D gait surface onto the XY plane. Before extracting the curvature information, the mean value of z are calculated for all point cloud data located in the same block (I, J) , denoted by $z_{mean(I, J)}$. The discrete surface point set is then obtained $S_{dis} = [I, J, z_{mean(I, J)}]$.

Figure 8. Projection of normalized point cloud data onto blocks.

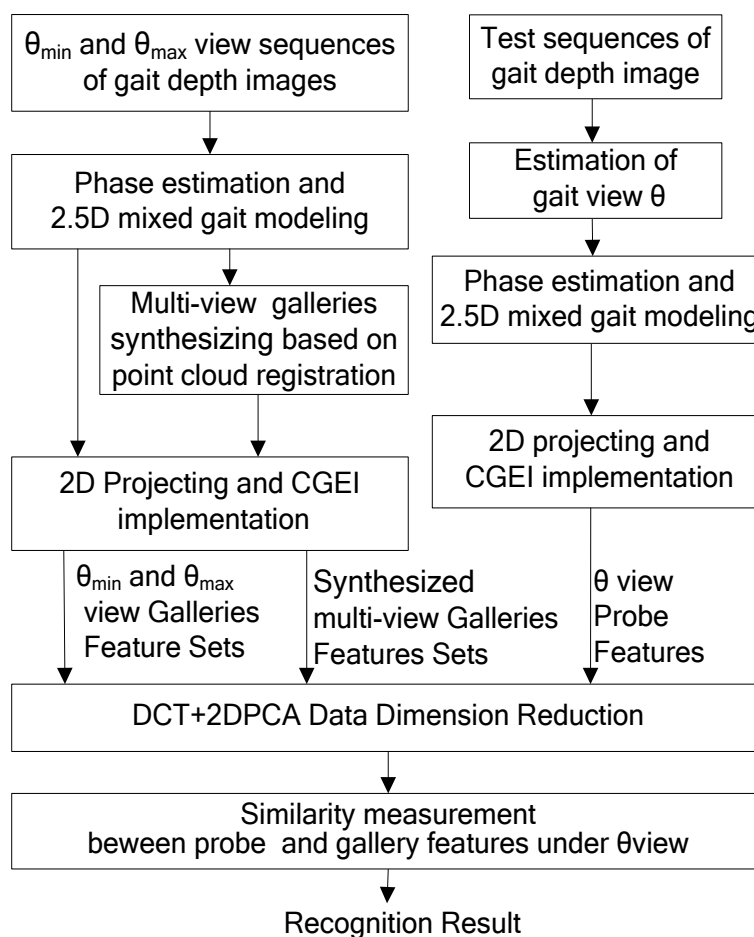
Gaussian curvature and mean curvature are view-invariant under certain class of geometric transformation including rotation, scaling and shearing [18]. Gaussian curvature K and mean curvature H are then computed through the discrete surface point set S_{dis} by the method in [19], respectively denoted as $K_{(I,J)}$ and $H_{(I,J)}$ for each block (I,J) .

The curvatures K and H in 2.5D space are normalized to the range $[0, 2^{16}]$, and the density of point cloud is also normalized to $[0, 255]$. They are then projected onto 2D RGB space to facilitate the gait feature extraction, classification and identification of the human subject. A pseudo 2D RGB color image is obtained by mapping $K_{(I,J)}$ and $H_{(I,J)}$ of each block to the red (R), green (G) and blue (B) components of the image. The R and G components of the image respectively denote the most significant 8 bit value and the low significant 8 bit values of the curvature. The B component of the image denotes the density of gait point cloud data that indicates the average silhouette information of human subject. The size of the resulting RGB image, *i.e.*, CGCI, is $N_I \times N_J$ and is shown in Figure 9. The final CGCI is denoted as $A = \{R(I, J), G(I, J), B(I, J)\} \in R^{N_I \times N_J}$.

Figure 9. Curvature based 2D color gait feature image (CGCI): (a) R component; (b) G component; (c) B component; and (d) RGB image.

4.2. Multi-View Gait Recognition

To realize multi-view gait recognition with a single Kinect depth camera, two standard reference gait views θ_{min} and θ_{max} are needed as a training gait set. Figure 10 shows the 2.5D view-invariant gait recognition method based on point cloud registration using CGCI in detail. In paper, 0° and 90° view of gait depth images are selected as standard reference training gait sets.

Figure 10. 2.5D view-invariant gait recognition method based on point cloud registration.

4.2.1. Estimation of Phase and View Angles for 2.5D Gait

In this paper, the gait phase is estimated using the width information of two legs in motion by silhouette depth images. The view angle of a test sample also needs to be estimated to enable it to be transformed or compared with the corresponding training samples. However it is difficult to obtain a reliable estimation from 2D images. Several methods have been proposed to estimate the gait angles. The method in [11] uses body part trajectories during walk to realize gait view normalization. This method does not work well when the difference in view angles is large. The method in [20] uses the regression models learned from training gait database. The estimation is poor when there is self-occlusion. The method in [21] uses perspective projection to estimate the walking angle in the 3D world from a video sequence of a planar scene. This method requires camera calibration and also performs poor when the difference in view angles is large.

A reliable method is proposed for view angle estimation in 2.5D space from video sequences. The subject is assumed to be walking along a straight line AC and line AB is parallel to the Z axis as shown in Figure 11. First, N th and $(N + k)$ th depth image frames are selected in a gait cycle. The two selected depth images are then used to construct 2.5D voxel gait model in the world coordinate. Let (x_i, y_i, z_i) , $i \in \{1, 2, \dots, N\}$ represent the point cloud data of gait where N denotes the total number of point cloud data. The centroids of the N th and $(N + k)$ th 2.5D gait model are then calculated, denoted

by (X_{1c}, Y_{1c}, Z_{1c}) and (X_{kc}, Y_{kc}, Z_{kc}) respectively. The estimation azimuth angle θ is given by $\tan\theta = (X_{kc} - X_{1c}) / (Z_{kc} - Z_{1c})$.

Figure 11. Estimation of view angles for 2.5D gait.

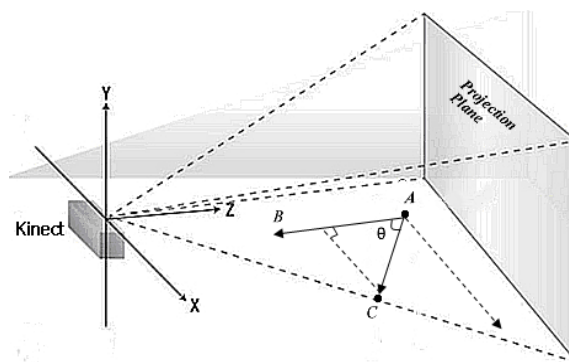


Table 1 shows the gait view angle estimation experiment results using 100 subjects for each subject for 0° , 15° , 30° , 45° , 60° , 75° , 90° views. The experimental results demonstrate that the proposed method is feasible.

Table 1. Gait view angle estimation results.

θ View Angle of Test Data	Estimation Result θ_{est}	
	$\bar{E}[\theta_{est}]$	σ^2
0°	0.55°	0.17
15°	15.62°	0.32
30°	29.47°	0.26
45°	44.31°	0.21
75°	74.29°	0.23
90°	90.48°	0.09

4.2.2. Multi-View Galleries Synthesizing

Before synthesizing multi-view galleries, two standard reference mixed gait models $M_{\theta_{\min}}$ and $M_{\theta_{\max}}$ must be constructed by the method in Section 3.2. This is followed by point cloud registration to form a new 2.5D mixed gait model $S_{\hat{a}}$. Multi-view galleries are then synthesized. It is based on rotating the new registered gait model $S_{\hat{a}}$ in $\Delta\theta$ step interval to obtain gait features for different view between two reference views. Let P_k^θ denotes the θ view synthesized 2.5D gait galleries, *i.e.*:

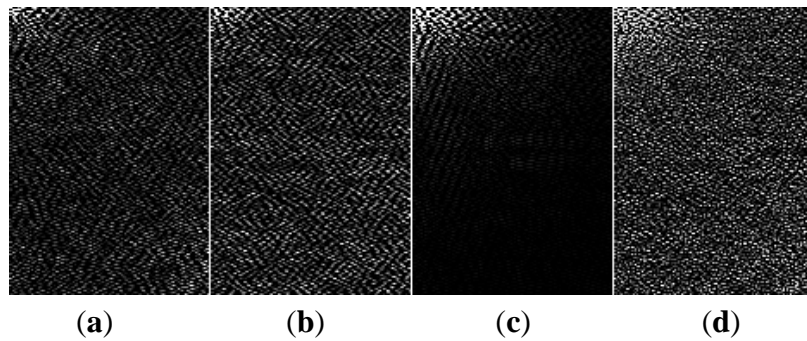
$$P_k^\theta = S_{\hat{a}} \cdot R_{(\beta \rightarrow \theta)}, k = 1 \dots N, \theta = 0^\circ, \Delta\theta, 2\Delta\theta, \dots, 90^\circ \quad (15)$$

where N represents the number of training objects. The density of point cloud data and 2.5D curvature features are then extracted from P_k^θ . Finally gait CGCI features with different view from 0° to 90° are obtained denoted as $A_k^\theta \in R^{N_I \times N_J}$, $k = 1 \dots N$.

4.2.3. DCT and 2DPCA Based Gait Feature Dimension Reduction

The size of a CGCI is determined by the sampling intervals dx and dy (see Section 4.1). 2D discrete cosine transforms (2D-DCT) [22] and 2-dimensional Principle Component Analysis (2DPCA) is used to reduce the dimensionality of the gait feature space. DCT is applied to R, G and B components of the CGCI separately. The DCT coefficient matrices are then obtained with the same size as CGCI. The low frequency components containing the most important information of the image are concentrated in the upper left corner of the DCT matrix, while the high frequency components are distributed in the lower right corner as shown in Figure 12. Since the high frequency components are less important, we only need to retain the low frequency components when reconstructing the image.

Figure 12. DCT transform of CGCI: (a–d) are respectively the DCTs of (a–d) in Figure 9.



A $m \times m$ matrix is used to extract the low frequency DCT coefficients in the upper left corner. The gait features extracted from DCT coefficients from N CGCIs with α view are then denoted by $A_k^\alpha \in R^{m \times m}, k=1 \dots N$, 2DPCA is then applied to further reduce the dimensions. Unlike PCA which involves vectors, 2DPCA deals with matrices corresponding to images, and uses a matrix to construct a covariance matrix [23]. After extracting multi-view CGCI-DCT gait features, the data dimensionality of gait features for each view is then conducted further reduction. Let CGCI-DCT feature matrices with N objects be denoted by:

$$A_k^\alpha \in R^{m \times m}, k=1 \dots N, \alpha=0^\circ, \Delta\theta, 2\Delta\theta, \dots, 90^\circ \quad (16)$$

The mean matrix \bar{A}^α and covariance matrix S^d of the N DCT matrices are then calculated. Eigenvalue decomposition is then performed on S^d , i.e.:

$$S^\alpha = (X^\alpha)^T D^\alpha X^\alpha \quad (17)$$

where:

$$D^\alpha = \text{diag}\{\lambda_1^\alpha, \lambda_2^\alpha, \dots, \lambda_m^\alpha\}, \lambda_1^\alpha \geq \lambda_2^\alpha \geq \dots \geq \lambda_m^\alpha \quad (18)$$

and $X^\alpha \in R^{m \times m}$ comprises the corresponding eigenvector. The optimum projective matrix X_{opt}^α comprises d ($d \leq m$) eigenvectors corresponding to the largest eigenvalues. Thus, the reduced gait features A_k^α is:

$$Y_k^\alpha = A_k^\alpha \times X_{opt}^\alpha, k=1 \dots N \quad (19)$$

The energy of 2DPCA is:

$$E^\alpha = \sum_{i=1}^d \lambda_i^\alpha / \sum_{i=1}^N \lambda_i^\alpha \quad (20)$$

The value of d is selected according to E , where E denotes the information rate of reducing the dimensionality. It is usually around $[0.9, 1]$. In paper, E is set to 0.95 which gives a good reduction and recognition result. Since a CGCIs-DCT is used to represent gait features, each image has three matrices, one for the R component $A_k^{\alpha R}$, another for the G component $A_k^{\alpha G}$ and the third for the B component $A_k^{\alpha B}$. 2DPCA is performed on each of these matrices.

4.2.4. Recognition

Each probe gait sequence is first processed to estimate their view \hat{e} and generate the corresponding probe gait features. The gallery gait set that has the most similar view is selected for gait recognition.

2.5D probe gait features with \hat{e} view are projected onto a plane to generate the CGCI and 2DPCA based DCT coefficients matrix denoted by $Y_t^\theta = \{Y_t^{\theta R}, Y_t^{\theta G}, Y_t^{\theta B}\}$, where the superscripts R , G and B respectively denote the R , G and B components. We define the Euclidean distance:

$$D(Y_t^{\theta col}, Y_k^{\theta col}) = \sqrt{|Y_t^{\theta col} - Y_k^{\theta col}|^2}, k = 1, \dots, c \quad (21)$$

where Y_k^θ is a gait feature matrix belonging to the k th class, $col = \{R, G, B\}$, and c is the number of class in training samples. Denote W_i as the i th class of training samples. The smallest distance is chosen as the recognition result, *i.e.*, $Y_t^{\theta col} \in W_i$ if:

$$D(Y_t^{\theta col}, Y_k^{\theta col}) = \min_{k=1}^c \sqrt{|Y_t^{\theta col} - Y_k^{\theta col}|^2} \quad (22)$$

We define the final fused distance measure:

$$D_{fused} = k_1 \cdot D(Y_t^{\theta R}, Y_k^{\theta R}) + k_2 \cdot D(Y_t^{\theta G}, Y_k^{\theta G}) + k_3 \cdot D(Y_t^{\theta B}, Y_k^{\theta B}) \quad (23)$$

where $k_1 = k_2 = k_3 = 1/3$, Let $Y_t^\theta \in W_i$ if $D(Y_t^\theta, Y_k^\theta) = \min D_{fused}$.

5. Experiments

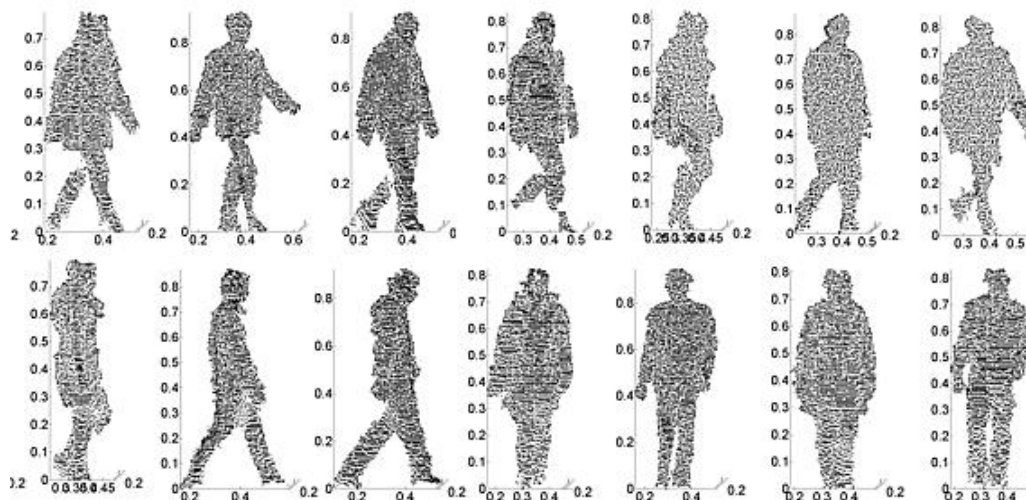
5.1. 2.5D CSU Point Cloud Gait Database

Hofmann *et al.* [24] presented a 2.5D TUM-GAID Database with depth information. However it is a gait database with depth information only. It is neither a 2.5D point gait database nor a multi-view database for gait. Since there are no publicly available 2.5D multi-view gait databases, we created the CSU database to evaluate the extraction of 2.5D gait features (e.g., as shown in Figure 13) and the identification algorithm using CGCI.

The database consists of 100 subjects, the data of each subject has 0° , 15° , 30° , 45° , 60° , 75° , 90° views. Each gait of the sample is captured three times. We use a single Microsoft Kinect camera to capture the videos. Each video sequence is of 8 s duration, recorded at full frame rate (30 frames/s). The original video format is 24-bit full-colour JPG and depth image files with resolution

of 640×480 . We extracted the subjects' data that have been segmented from background using OpenNI and generate 2.5D point cloud sets that contain the subjects' gait features.

Figure 13. 2.5D point cloud gait features with different views.



5.2. Point Cloud Registration

The point cloud registration algorithm using head point cloud data is proposed in Section 3.3. Object No. 8 in CSU Point Cloud Gait Database is taken as an example. The original views of mixed head point cloud data are 0° and 90° . Figure 14a shows the mixed head point cloud data in $\beta = 45^\circ$ after rotation from the original views. Figure 14b represents the data after registration with $\theta_{min} = 45^\circ$, $\theta_{max} = 90^\circ$ and $\beta = 45^\circ$ in Equation (3). Figure 15 shows the relation between distance error e_k and the number of iteration. We set the end condition for iteration as:

$$e_k - e_{k-1} > \varepsilon, e_k = \sum_{i=1}^{\min(m,n)} \|R_k l_{\theta_{max}}^i + T_k - l_{\theta_{min}}^i\|^2 \quad (24)$$

where k denotes the number of iteration, and $\varepsilon = 10^{-5}$. Table 2 shows the point cloud registration result. Matrices R and T matrix are then used in full gait body to gain full registration gait model. By rotating the full registration gait model $S_{\hat{a}}$ using Equation (5), the gait features for different views are obtained.

Figure 14. Registration result: (a) before registration; and (b) after registration.

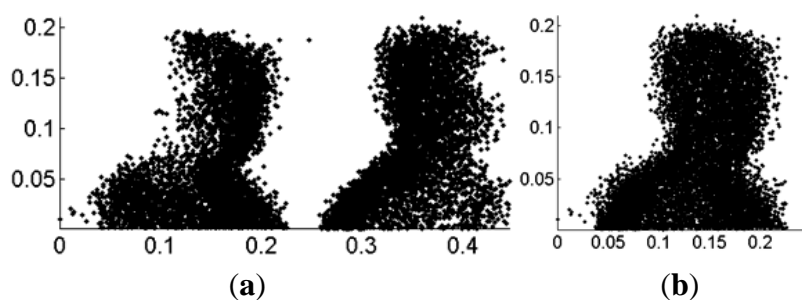
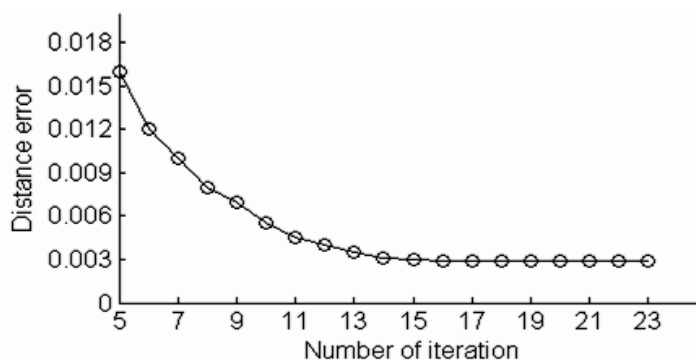


Figure 15. Relation between distance error and the number of iteration.**Table 2.** Point Cloud Registration Result.

Number of Iterations	Pixels	<i>R</i> Matrix	<i>T</i> Matrix
18	10,062	$\begin{pmatrix} 0.998 & -0.002 & 0.329 \\ 0 & 0.999 & 0.008 \\ -0.352 & -0.007 & 0.991 \end{pmatrix}$	$\begin{pmatrix} -0.2118 \\ -0.0012 \\ -0.0011 \end{pmatrix}$

5.3. Multi-View Gait Recognition Experiments and Results

In order to evaluate the effectiveness of the proposed algorithm, experiments are carried out to compare the multiple-view gait recognition performances using three different methods. The first method uses the VTM technique and GEI as gait features [6]. The second method uses the 3D-based VTM technique and GEI as gait features [7]. It uses a 3D gait database comprising visual hulls with intact 360 degree body surfaces to construct the VTM model and realize view-invariant gait recognition by the VTM technique. In our experiment, we use our 2.5D voxel model which only has one side surface portion of the human body instead of the 3D data used in [7]. The third method uses our multi-view synthesizing method based on point cloud registration (MVSM).

The 2.5D CSU Point Cloud Gait Database is used in the experiments, where each gait sample is captured three times for each view. The database is divided into two sets. In all experiments, the two sets of gaits with different views are used. But only 0° and 90° views data from one set are retained for training and also as the gallery data sets for evaluating performance of multi-view gait recognition. The other set is used as the probe data set with different views.

Figure 10 illustrates the overview structure of the proposed gait recognition method. The three methods are trained using two standard reference gait view sets instead of the multi-view gait data in [6, 7]. The reference gait view θ_{min} and θ_{max} are set to 0° and 90°, respectively. The step interval $\Delta\theta$ is set to 15°. Therefore, the VTM in VTM-GEI [6] method is constructed from GEI features with just 0° and 90° views from the training data set. The VTM in the 3D-based VTM-GEI [7] method is also constructed from GEI features but with more views. It includes 0° and 90° of the original views and others, *i.e.*, 15°, 30°, 45°, 60°, 75° views data that are generated by rotation from 0° and 90° (not from database). The GEI features are extracted from the B component in CGCI that indicates the average silhouette information of gait.

To compute the comparison chart, in the VTM method the probe gait data from one view is transformed to a feature data under two other views that match one of the views (0° or 90°) in gallery gait database. In our proposed approach, the probe gait data from one view is matched with the corresponding synthesized galleries from (0° or 90°) data, and the gait similarity measurement is then calculated. Figure 16 shows the performances of the three methods.

Additional experiments that set different step intervals in multi-view synthesizing process are conducted to show the improvement of the view-invariant 2.5D gait recognition. The results are shown in Figures 17 and 18. Cumulative Match Scores (CMS) are used to illustrate our 2.5D view-invariant gait recognition results. The CMS value α corresponding to rank r indicates a fraction $100\alpha\%$ of probes whose top r matches must include the real identity matches.

Figure 16. Comparison of recognition performance using different approaches.

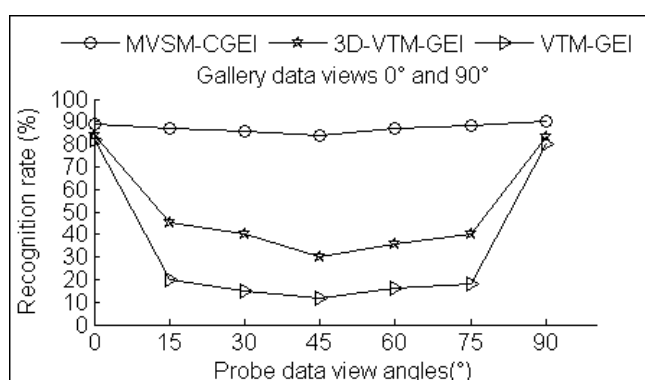


Figure 17. Recognition results with step interval $\Delta\theta = 15^\circ$.

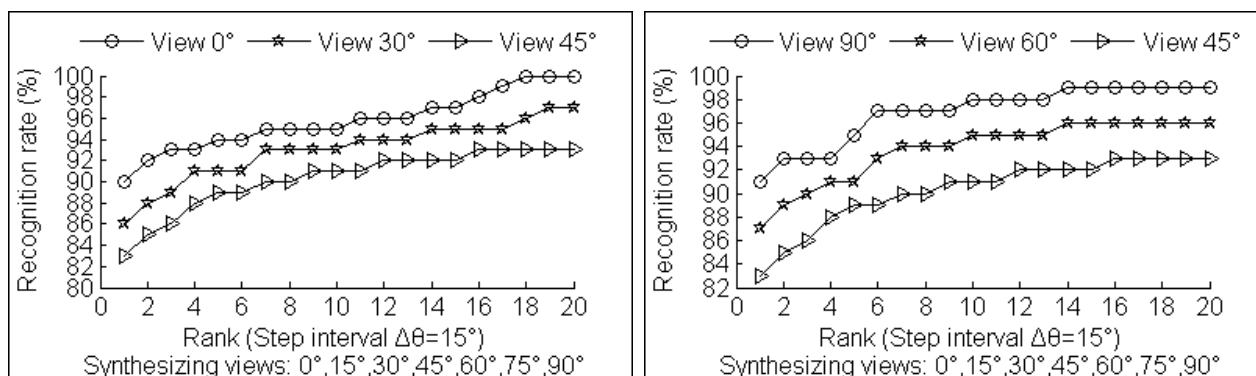


Figure 18. Recognition results with step interval $\Delta\theta = 20^\circ$.

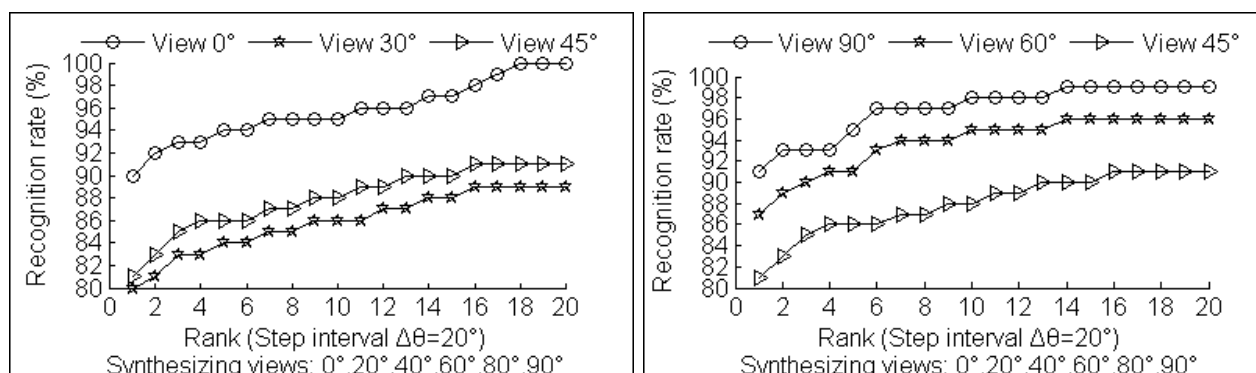


Figure 16 shows that the proposed 2.5D view-invariant gait recognition approach outperforms the VTM method and the 3D-based VTM method. The 3D-based VTM method has better performance than the original VTM method because it can rotate the 2.5D gait data to extract multi-view gait features for VTM construction. The original VTM method has the worst performance, especially for the 30°, 45° and 60° views. This is because during training these views do not include the gallery sets that lead to bigger VTM angle transformation before the similarity measurement. The experimental results in [6–9] indicate the recognition rate degrades dramatically when the probe and gallery views differ by 30°. Another reason is that the number of multiple images from the gallery sets used in VTM construction and recognition directly influences its performance. In our method, only gallery data from two views are needed for training while three times more are used in [6–9]. Unlike the VTM-based method, our approach uses data registration and a 3D rotation method to synthesize different gallery data. As a result, five additional synthesized gallery data for views 15°, 30°, 45°, 60° and 75° are used that overcome the big angle transformation problem. This is the main reason why our method has achieved higher recognition results for probe views of 15°, 30°, 45°, 60° and 75° in Figure 6.

There are several other reasons why our method achieves significantly better performance. The first is that our multi-view 2.5D gait recognition method uses a single Kinect camera whereas VTM and other multi-view gait recognition approaches use multiple cameras or multi-view images. The experiment shows the method in [7] requires an intact gait 3D surface. Its recognition result drops significantly when our two standard reference gait view sets with only one side surface portion of the human body are used. This is because of the self-occlusion problem when using incomplete 3D sets to rotate the 3D gait models to construct VTM.

The second reason is that view-invariant 2.5D features are effectively extracted through Gaussian curvature, where Gaussian curvature is invariant to affine transformation. It is view-invariant under certain class of geometric transformation including rotation, scaling and shearing. This improves the performance of 2.5D gait feature extraction for multi-view gait recognition.

The last key point is that compared to GEI, our CGCI contains more surface information. The CGCI are gray images with curvature information instead of binary silhouette images used in GEI. Furthermore, Figures 17 and 18 show that the test gait view closer to reference training views or synthesized gallery views has significantly better recognition rate. Thus, in real applications, step interval $\Delta\theta$ can be set to be small to gain more accuracy in arbitrary view gait recognition, but with a larger size for synthesizing gait databases.

6. Conclusions

In this paper, a 2.5D voxel gait model is constructed by point cloud data captured by a Kinect camera. In order to achieve multi-view gait recognition with one single Kinect camera, a point cloud registration method is proposed to synthesize different view gait features. Density of point cloud, Gaussian curvature and mean curvature are then introduced for extracting 2.5D gait features, which are projected to 2D RGB space to generate CGCI as expression of gait features.

The experimental results show that the proposed method is more effective than VTM-based multi-view gait recognition and other multi-view gait recognition methods using a single camera. Our 2.5D view invariant gait recognition based on point cloud registration approach needs only one camera

and does not use multi-view images to achieve gait training and recognition. It achieves arbitrary view gait recognition without any camera calibration information. These advantages enable the proposed method to be used in many practical surveillance applications.

Acknowledgments

This work was supported partly by the National Natural Science Foundation of China (91220301), the Science and Technology Program of Hunan, China (2013WK3026).

Author Contributions

All authors participated in designing and implementing the study. All authors discussed the basic structure of the manuscript. Jin Tang and Jian Luo drafted the main parts of manuscript. Tardi Tjahjadi and Yan Gao reviewed and edited the draft. All authors read and approved the final manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Das Choudhury, S.; Tjahjadi, T. Silhouette-based gait recognition using Procrustes shape analysis and elliptic Fourier descriptors. *Pattern Recognit.* **2012**, *45*, 3414–3426.
2. Das Choudhury, S.; Tjahjadi, T. Gait recognition based on shape and motion analysis of silhouette contours. *Comput. Vis. Image Underst.* **2013**, *117*, 1770–1785.
3. Shakhnarovich, G.; Lee, L.; Darrell, T. Integrated face and gait recognition from multiple views. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001; Volume 1, pp. 439–446.
4. Gu, J.; Ding, X.; Wang, S.; Wu, Y. Action and gait recognition from recovered 3-D human joints. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2010**, *40*, 1021–1033.
5. Sivapalan, S.; Chen, D.; Denman, S.; Sridharan, S.; Fookes, C. Gait energy volumes and frontal gait recognition using depth images. In Proceedings of the 2011 International Joint Conference on Biometrics, Washington, DC, USA, 11–13 October 2011; pp. 1–6.
6. Makihara, Y.; Sagawa, R.; Mukaigawa, Y.; Echigo, T.; Yagi, Y. Gait recognition using a view transformation model in the frequency domain. In Proceedings of the 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; pp. 151–163.
7. Muramatsu, D.; Shiraishi, A.; Makihara, Y.; Yagi, Y. Arbitrary view transformation model for gait person authentication. In Proceedings of the 2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems, Arlington, MA, USA, 28–30 September 2012; pp. 85–90.
8. Kusakunniran, W.; Wu, Q.; Zhang, J.; Li, H. Gait recognition under various viewing angles based on correlated motion regression. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 966–980.
9. Kusakunniran, W.; Wu, Q.; Zhang, J.; Li, H. Cross-view and multi-view gait recognitions based on view transformation model using multi-layer perceptron. *Pattern Recognit. Lett.* **2012**, *33*, 882–889.

10. Nizami, I.F.; Hong, S.; Lee, H.; Ahn, S.; Toh, K.-A.; Kim, E. Multi-view gait recognition fusion methodology. In Proceedings of the 3rd IEEE Conference on Industrial Electronics and Applications, Singapore, Singapore, 3–5 June 2008; pp. 2101–2105.
11. Jean, F.; Albu, A.B.; Bergevin, R. Towards view-invariant gait modeling: Computing view-normalized body part trajectories. *Pattern Recognit.* **2009**, *42*, 2936–2949.
12. Suau, X.; Casas, J.R.; Ruiz-Hidalgo, J. Real-time head and hand tracking based on 2.5D data. In Proceedings of the 2011 IEEE International Conference on Multimedia and Expo, Barcelona, Spain, 11–15 July 2011; pp. 1–6.
13. Smisek, J.; Jancosek, M.; Pajdla, T. 3D with Kinect. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops, Barcelona, Spain, 6–13 November 2011; pp. 1154–1160.
14. McHugh, J.M.; Konrad, J.; Saligrama, V.; Jodoin, P. Foreground-adaptive background subtraction. *IEEE Signal Process. Lett.* **2009**, *16*, 390–393.
15. Yu, Z.; Kang, B.; Li, H.; Shi, F. Improved algorithm for point cloud data simplification. *J. Comput. Appl.* **2012**, *32*, 521–523.
16. Besl, P.J.; McKay, H.D. A method for registration of 3D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256.
17. Qin, X.; Wang, J.; Zheng, H.; Liang, Z. Point clouds registration of 3D moment invariant feature estimation. *J. Mech. Eng.* **2013**, *49*, 129–134.
18. Tosranon, P.; Sanpanich, A.; Bunluechokchai, C.; Pintavirooj, C. Gaussian curvature-based geometric invariance. In Proceedings of the 6th International Conference on Electrical, Electronics, Computer, Telecommunications and Information Technology, Pattaya, Thailand, 6–9 May 2009; Volume 2, pp. 1124–1127.
19. Besl, P.J.; Jain, R.C. Segmentation through variable-order surface fitting. *IEEE Trans. Pattern Anal. Mach. Intell.* **1988**, *10*, 167–192.
20. Zhang, D.; Wang, Y.; Zhang, Z.; Hu, M. Estimation of view angles for gait using a robust regression method. *Multimed. Tools Appl.* **2013**, *65*, 419–439.
21. Kale, A.; Chowdhury, A.K.R.; Chellappa, R. Towards a view invariant gait recognition algorithm. In Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance, Miami, FL, USA, 21–22 July 2003; pp. 143–150.
22. Haque, M. A two-dimensional fast cosine transform. *IEEE Trans. Acoust. Speech Signal Proc.* **1985**, *33*, 1532–1539.
23. Yang, J.; Zhang, D. Two-dimensional PCA: A new approach to appearance-based face representation and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 131–137.
24. Hofmann, M.; Geiger, J.; Bachmann, S.; Schuller, B.; Rigoll, G. The TUM gait from audio, image and depth (GAID) database: Multimodal recognition of subjects and traits. *J. Vis. Commun. Image Represent.* **2014**, *25*, 195–206.