An Evaluation Framework and Database
for MoCap-Based Gait Recognition Methods⋆

Michal Balazia (0000-0001-7153-9984) and Petr Sojka (0000-0002-5768-4007)

Faculty of Informatics, Masaryk University, Botanická 68a, 602 00 Brno, Czech Republic
xbalazia@mail.muni.cz and sojka@fi.muni.cz

Abstract As a contribution to reproducible research, this paper presents a framework and a database to improve the development, evaluation and comparison of methods for gait recognition from motion capture (MoCap) data. The evaluation framework comprises source codes of state-of-the-art human-interpretable geometric features as well as our own approaches where gait features are learned by a modification of Fisher's Linear Discriminant Analysis with the Maximum Margin Criterion, and by a combination of Principal Component Analysis and Linear Discriminant Analysis. It includes a description and source codes of a mechanism for evaluating class separability coefficients of feature space and four classifier performance metrics. This framework also contains a tool for learning a custom classifier and for classifying a custom probe on a custom gallery. We provide an experimental database along with source codes for its extraction from the general CMU MoCap database.

1 Introduction

Gait (walk) pattern has several attractive properties as a soft biometric trait. From a surveillance perspective, gait pattern biometrics is appealing in that it can be performed at a distance without requiring body-invasive equipment or subject cooperation.

Many research groups investigate the discrimination power of gait pattern and develop models that are applied to the automatic recognition of walking people from MoCap data. A number of MoCap-based gait recognition methods have been introduced in the past few years and new ones continue to emerge. In order to move forward with this competitive research, it is necessary to compare their innovative approaches with the state-of-the-art and evaluate them against established evaluation metrics on a benchmark database. New frameworks and databases have been developed recently [10,15].

As a contribution to reproducible research, this paper focuses on our framework for evaluating MoCap-based gait recognition methods and our benchmark MoCap gait database. We provide a large experimental database together with its extraction-and-normalization drive from the general CMU MoCap database, as specified in Section 2. Implementation details of thirteen relevant methods are summarized in Section 3. In Section 4 we describe the evaluation mechanism and define four class separability coefficients and four rank-based classifier performance metrics. Finally, Section 5 consists of a manual and comments on reproducing the experiments.

⋆ This is a companion paper to our papers [7,8].

2 Data

MoCap technology provides video clips of individuals walking which contain structural motion data. The format keeps an overall structure of the human body and holds estimated 3D positions of major anatomical landmarks as the person moves. These MoCap data can be collected online by a system of multiple cameras (Vicon) or a depth camera (Microsoft Kinect). To visualize MoCap data (see Figure 1), a simplified stick figure representing the human skeleton (graph of joints connected by bones) can be recovered from body point spatial coordinates in time. Recent rapid improvement in MoCap sensor accuracy has brought affordable MoCap technology to assist human identification in such applications as access control and video surveillance.
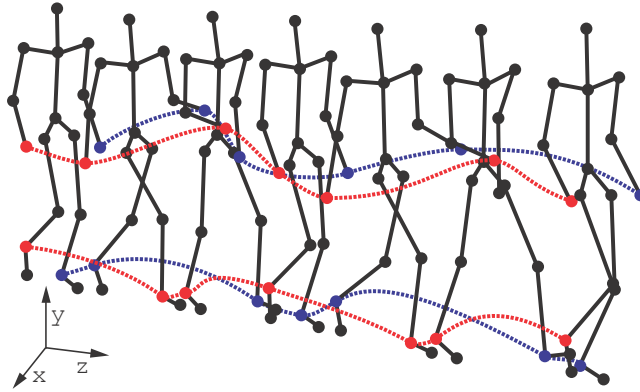
Figure 1. Motion capture data. A skeleton is represented by a stick figure of 31 joints (only 17 are shown here). Seven selected video frames of a walk sequence contain 3D coordinates of each joint in time. The red and blue lines track trajectories of hands and feet. [24]

For evaluation purposes we have extracted a large number of gait samples from the MoCap database obtained from the CMU Graphics Lab [11], which is available under the Creative Commons license. It is a well-known and recognized database of structural human motion data and contains a considerable number of gait sequences. Motions are recorded with an optical marker-based Vicon system. People wear a black jumpsuit with 41 markers taped on. The tracking space of $30\,\text{m}^2$ is surrounded by 12 cameras with a sampling rate of $120\,\text{Hz}$ at heights ranging from 2 to 4 meters above ground thereby creating a video surveillance environment. Motion videos are triangulated to get highly accurate 3D data in the form of relative body point coordinates (with respect to the root joint) in each video frame and are stored in the standard ASF/AMC data format. Each registered participant is assigned with their respective skeleton described in an ASF file. Motions in the AMC files store bone rotational data, which is interpreted as instructions about how the associated skeleton deforms over time.

These MoCap data, however, contain skeleton parameters pre-calibrated by the CMU staff. Skeletons are unique to each walker and even a trivial skeleton check could result in $100\,\%$ recognition. In order to fairly use the collected data, a prototypical skeleton is constructed and used to represent bodies of all subjects, shrouding the skeleton

parameters. Assuming that all walking individuals are physically identical disables the skeleton check from being a potentially unfair classifier. Moreover, this is a skeleton-robust solution as all bone rotational data are linked to one specific skeleton. To obtain realistic parameters, it is calculated as mean of all skeletons in the provided ASF files.

The raw data are in the form of bone rotations or, if combined with the prototypical skeleton, 3D joint coordinates. The bone rotational data are taken from the AMC files without any pre-processing. We calculate the joint coordinates using the bone rotational data and the prototypical skeleton. One cannot directly use raw values of joint coordinates, as they refer to absolute positions in the tracking space, and not all potential methods are invariant to person's position or walk direction. To ensure such invariance, the center of the coordinate system is moved to the position of root joint $\gamma_{\text{root}}(t) = [0, 0, 0]^\top$ for each time t and the axes are adjusted to the walker's perspective: the X axis is from right (negative) to left (positive), the Y axis is from down (negative) to up (positive), and the Z axis is from back (negative) to front (positive). In the AMC file structure notation it is achieved by setting the root translation and rotation to zero (`root 0 0 0 0 0 0`) in all frames of all motion sequences.

Since the general motion database contains all motion types, we extracted a number of sub-motions that represent gait cycles. First, an exemplary gait cycle was identified, and clean gait cycles were then filtered out using a threshold for their Dynamic Time Warping (DTW) distance on bone rotations in time. The distance threshold was explicitly set low enough so that even the least similar sub-motions still semantically represent gait cycles. Setting this threshold higher might also qualify sub-motions that do not resemble gait cycles anymore. Finally, subjects that contributed with less than 10 samples were excluded. The final database [2] has 54 walking subjects that performed 3,843 samples in total, which results in an average of about 71 samples per subject.

3 Implementation Details of Algorithms

Recognizing a person from their gait involves capturing and normalizing their walk sample $\mathbf{g}_n$, extracting gait features to compose a template $\widehat{\mathbf{g}}_n$, and finally querying the gallery for a set of similar templates $\widehat{\mathbf{g}}_{n'}$ – based on a distance function $\widehat{\delta}(\widehat{\mathbf{g}}_n, \widehat{\mathbf{g}}_{n'})$ – to report the most likely identity. This work focuses on extracting robust and discriminative gait features from raw MoCap data.

Many geometric gait features have been introduced over the past few years. They are typically combinations of static body parameters (bone lengths, person's height) [18] with dynamic gait features such as step length, walking speed, joint angles and inter-joint distances [3,6,18,20], along with various statistics (mean, standard deviation or maximum) of their signals [5]. Clearly, these features are schematic and human-interpretable, which is convenient for visualizations and for intuitive understanding, but unnecessary for automatic gait recognition. Instead, our approach [7,8] prefers learning features in a supervised manner that maximally separate the identity classes and are not limited by such dispensable factors.

What follows is a detailed specification of the thirteen gait features extraction methods that we have reviewed in our work to date. Since the idea behind each method has some potential, we have implemented each of them for direct comparison.

- **Ahmed** by Ahmed *et al.* [4] extracts the mean, standard deviation and skew during one gait cycle of horizontal distances (projected on the Z axis) between feet, knees, wrists and shoulders, and mean and standard deviation during one gait cycle of vertical distances (Y coordinates) of head, wrists, shoulders, knees and feet, and finally the mean area during one gait cycle of the triangle of root and two feet.
- **Ali** by Ali *et al.* [5] measures the mean areas during one gait cycle of lower limb triangles.
- **Andersson** by Andersson *et al.* [6] calculates gait attributes as mean and standard deviation during one gait cycle of local extremes of the signals of lower body angles, step length as a maximum of feet distance, stride length as a length of two steps, cycle time and velocity as a ratio of stride length and cycle time. In addition, they extract the mean and standard deviation during one gait cycle of each bone length, and height as the sum of the bone lengths between head and root plus the averages of the bone lengths between root and both feet.
- **Ball** by Ball *et al.* [9] measures mean, standard deviation and maximum during one gait cycle of lower limb angle pairs: upper leg relative to the Y axis, lower leg relative to the upper leg, and the foot relative to the Z axis.
- **Dikovski** by Dikovski *et al.* [12] selects the mean during one gait cycle of step length, height, all bone lengths, then mean, standard deviation, minimum, maximum and mean difference of subsequent frames during one gait cycle of all major joint angles, and the angle between the lines of the shoulder joints and the hip joints.
- **Gavrilova** by Gavrilova *et al.* [3] chooses 20 joint relative distance signals and 16 joint relative angle signals across the whole body, compared using the DTW.
- **Jiang** by Jiang *et al.* [14] measures angle signals between the Y axis and four major lower body (thigh and calf) bones. The signals are compared using the DTW.
- **Krzeszowski** by Krzeszowski *et al.* [16] observes the signals of rotations of eight major bones (humerus, ulna, thigh and calf) around all three axes, the person's height and step length. These signals are compared using the DTW distance function.
- **Kumar** by Kumar *et al.* [17] extracts all joint trajectories around all three axes. Gait samples are compared by a distance function of their covariance matrices.
- **Kwolek** by Kwolek *et al.* [18] processes signals of bone angles around all axes, the person's height and step length. The gait cycles are normalized to 30 frames.
- **Preis** by Preis *et al.* [19] takes height, length of legs, torso, both lower legs, both thighs, both upper arms, both forearms, step length and speed.
- **Sedmidubsky** by Sedmidubsky *et al.* [21] concludes that only the two shoulder-hand signals are discriminatory enough to be used for recognition. These temporal data are compared using the DTW distance function.
- **Sinha** by Sinha *et al.* [22] combines all features of Ball and Preis with mean areas during one gait cycle of upper body and lower body, then mean, standard deviation and maximum distances during one gait cycle between the centroid of the upper body polygon and the centroids of four limb polygons.

We are interested in finding an optimal feature space by maximizing its class separability, which is when gait templates are close to those of the same walker and far from those of other walkers. The method proposed in [7,8] learns gait features directly from joint coordinates by a modification of Fisher's Linear Discriminant Analysis [13] with Maximum Margin Criterion. The framework allows learning from bone rotations as well.

Let the model of a human body have J joints and all samples be linearly normalized to their average length T. Labeled learning data in the sample (measurement) space $\mathcal{G}_L$ are in the form $\{(\mathbf{g}_n, \ell_n)\}_{n=1}^{N_L}$ where $\mathbf{g}_n = [[\gamma_1(1) \cdots \gamma_J(1)]^\top \cdots [\gamma_1(T) \cdots \gamma_J(T)]^\top]^\top$ is a sample (gait cycle) in which $\gamma_j(t) \in \mathbb{R}^3$ are 3D spatial coordinates of a joint $j \in \{1, \ldots, J\}$ at time $t \in \{1, \ldots, T\}$ normalized with respect to the person's position and direction. See that $\mathcal{G}_L$ has dimensionality $D = 3JT$. Learning on bone rotations is analogical. Each learning sample falls strictly into one of the learning identity classes $\{\mathcal{I}_c\}_{c=1}^C$ determined by ℓ_n. A class $\mathcal{I}_c \subseteq \mathcal{G}_L$ has N_c samples. The classes are complete and mutually exclusive. We say that learning samples $(\mathbf{g}_n, \ell_n)$ and $(\mathbf{g}_{n'}, \ell_{n'})$ share a common walker if and only if they belong to the same class, i.e., $(\mathbf{g}_n, \ell_n), (\mathbf{g}_{n'}, \ell_{n'}) \in \mathcal{I}_c \Leftrightarrow \ell_n = \ell_{n'}$.

Apart from Maximum Margin Criterion (MMC) we also investigated the fusion of Principal Component Analysis (PCA) with Linear Discriminant Analysis (LDA) that has been used for silhouette-based (2D) gait recognition by Su *et al.* [23]. Feature extraction is given by a linear transformation (feature) matrix $\mathbf{\Phi} \in \mathbb{R}^{D \times \widehat{D}}$ from a D-dimensional sample space $\mathcal{G} = \{\mathbf{g}_n\}_{n=1}^N$ of not necessarily labeled gait samples to a $\widehat{D}$-dimensional feature space $\widehat{\mathcal{G}} = \{\widehat{\mathbf{g}}_n\}_{n=1}^N$ of gait templates where $\widehat{D} < D$ and gait samples $\mathbf{g}_n$ are transformed into gait templates $\widehat{\mathbf{g}}_n$ by $\widehat{\mathbf{g}}_n = \mathbf{\Phi}^\top \mathbf{g}_n$.

On given labeled learning data $\mathcal{G}_L$, Algorithm 1 and Algorithm 2 are efficient ways of learning the transforms $\mathbf{\Phi}$ for MMC and PCA+LDA, respectively. Both algorithms [7,8] are of quadratic complexity with respect to the number of learning identity classes due to the singular value decomposition and eigenvalue decomposition.

Algorithm 1 LearnTransformationMatrixMMC($\mathcal{G}_L$)

1: split $\mathcal{G}_L = \{(\mathbf{g}_n, \ell_n)\}_{n=1}^{N_L}$ into classes $\{\mathcal{I}_c\}_{c=1}^{C_L}$ of $N_c = |\mathcal{I}_c|$ samples
2: compute overall mean $\mu = \frac{1}{N_L} \sum_{n=1}^{N_L} \mathbf{g}_n$ and individual class means $\mu_c = \frac{1}{N_c} \sum_{n=1}^{N_c} \mathbf{g}_n^{(c)}$
3: compute $\mathbf{\Sigma}_B = \sum_{c=1}^{C_L} (\mu_c - \mu)(\mu_c - \mu)^\top$
4: compute $\mathbf{X} = \frac{1}{\sqrt{N_L}} [(\mathbf{g}_1 - \mu) \cdots (\mathbf{g}_{N_L} - \mu)]$
5: compute $\mathbf{\Upsilon} = [(\mu_1 - \mu) \cdots (\mu_{C_L} - \mu)]$
6: compute eigenvectors $\mathbf{\Omega}$ and corresponding eigenvalues $\mathbf{\Theta}$ of $\mathbf{\Sigma}_T$ through SVD of $\mathbf{X}$
7: compute eigenvectors $\mathbf{\Xi}$ of $\mathbf{\Theta}^{-1/2} \mathbf{\Omega}^\top \mathbf{\Sigma}_B \mathbf{\Omega} \mathbf{\Theta}^{-1/2}$ through SVD of $\mathbf{\Theta}^{-1/2} \mathbf{\Omega}^\top \mathbf{\Upsilon}$
8: compute eigenvectors $\mathbf{\Psi} = \mathbf{\Omega} \mathbf{\Theta}^{-1/2} \mathbf{\Xi}$
9: compute eigenvalues $\mathbf{\Delta} = \mathbf{\Psi}^\top \mathbf{\Sigma}_B \mathbf{\Psi}$
10: return transform $\mathbf{\Phi}$ as eigenvectors in $\mathbf{\Psi}$ that correspond to the eigenvalues of at least $^1\!/_2$ in $\mathbf{\Delta}$

**Algorithm 2** LearnTransformationMatrixPCALDA($\mathcal{G}_L$)

1: split $\mathcal{G}_L = \{(\mathbf{g}_n, \ell_n)\}_{n=1}^{N_L}$ into classes $\{\mathcal{I}_c\}_{c=1}^{C_L}$ of $N_c = |\mathcal{I}_c|$ samples
2: compute overall mean $\mu = \frac{1}{N_L} \sum_{n=1}^{N_L} \mathbf{g}_n$ and individual class means $\mu_c = \frac{1}{N_c} \sum_{n=1}^{N_c} \mathbf{g}_n^{(c)}$
3: compute $\mathbf{\Sigma}_B = \sum_{c=1}^{C_L} (\mu_c - \mu)(\mu_c - \mu)^\top$
4: compute $\mathbf{\Sigma}_W = \sum_{c=1}^{C_L} \frac{1}{N_c} \sum_{n=1}^{N_c} \left(\mathbf{g}_n^{(c)} - \mu_c\right)\left(\mathbf{g}_n^{(c)} - \mu_c\right)^\top$
5: compute eigenvectors $\mathbf{\Phi}_{PCA}$ of $\mathbf{\Sigma}_T = \mathbf{\Sigma}_B + \mathbf{\Sigma}_W$ that correspond to $\overline{D} = C_L$ largest eigenvalues
6: compute eigenvectors $\mathbf{\Phi}_{LDA}$ of $(\mathbf{\Phi}_{PCA}^\top \mathbf{\Sigma}_W \mathbf{\Phi}_{PCA})^{-1} (\mathbf{\Phi}_{PCA}^\top \mathbf{\Sigma}_B \mathbf{\Phi}_{PCA})$
7: return transform $\mathbf{\Phi} = \mathbf{\Phi}_{PCA} \mathbf{\Phi}_{LDA}$

In addition to the gait features extraction methods of our fellow researchers, we implemented our own methods as described below. Depending on whether the raw data are in the form of bone rotations or joint coordinates, the methods are referred to with BR or JC subscripts, respectively.

- **_MMC** learns gait features by MMC (Algorithm 1) and the gait templates are compared by the Mahalanobis distance [7,8].
- **_PCALDA** learns gait features by PCA+LDA (Algorithm 2) and the gait templates are also compared by the Mahalanobis distance [7,8].
- **_Random** has no features and classification is performed by picking a random identity that is present in the gallery.
- **_Raw** takes all raw data. The template vector, normalized to the average of $T = 150$ frames, results in a large feature space dimensionality $\widehat{D} = D = 3JT = 13{,}950$, which is why the raw data cannot be directly used for recognition on large databases.

4 Evaluation

Learning data $\mathcal{G}_L = \{(\mathbf{g}_n, \ell_n)\}_{n=1}^{N_L}$ of C_L identities and evaluation data $\mathcal{G}_E = \{(\mathbf{g}_n, \ell_n)\}_{n=1}^{N_E}$ of C_E identity classes have to be disjunct at all times. In the following, we introduce two setups of data separation: homogeneous and heterogeneous. The homogeneous setup learns the transformation matrix on $1/3$ samples of C_L identities and is evaluated on templates derived from the other $2/3$ samples of the same $C_E = C_L$ identities. The heterogeneous setup learns the transform on all samples in C_L identities and is evaluated on all templates derived from other C_E identities. An abstraction of this concept is depicted in Figure 2. Note that unlike in the homogeneous setup, no walker identity is ever used for both learning and evaluation at the same time in the heterogeneous setup.
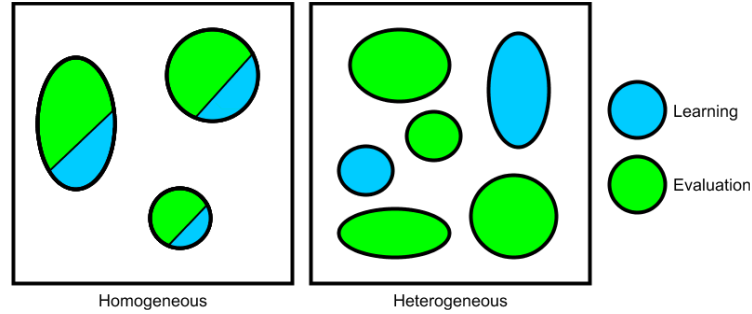

Figure 2. Data separation for the homogeneous setup of $C_L = C_E = 3$ learning-and-evaluation classes (left) and for the heterogeneous setup of $C_L = 2$ learning classes and $C_E = 4$ evaluation classes (right). The black square represents a database and the ellipses are the identity classes.

The homogeneous setup is parametrized by a single number $C_L = C_E$ of learning-and-evaluation identity classes, whereas the heterogeneous setup has the form (C_L, C_E)

specifying how many learning and how many evaluation identity classes are randomly selected from the database. The evaluation of each setup is repeated 3 times, selecting new random C_L and C_E identity classes each time and reporting the average result.

In the homogeneous setup, all results are estimated with nested cross-validation (see Figure 3) that involves the outer 3-fold cross-validation loop where templates in one fold are used for learning the features, while templates in the remaining two folds are used for evaluations. In the heterogeneous setup, the learning and evaluation parts are selected at random based on the given C_L and C_E, respectively. For both setups, this model is frozen and ready to be evaluated for class separability coefficients. Evaluation of rank-based classifier performance metrics advances to the inner 10-fold cross-validation loop taking one dis-labeled fold as a testing set and the other nine labeled folds as gallery. Test templates are classified by the winner-takes-all strategy, in which a test template $\widehat{\mathbf{g}}^{\text{test}}$ gets assigned with the label $\ell_{\arg\min_i \widehat{\delta}\left(\widehat{\mathbf{g}}^{\text{test}}, \widehat{\mathbf{g}}_i^{\text{gallery}}\right)}$ of the gallery's closest identity class.
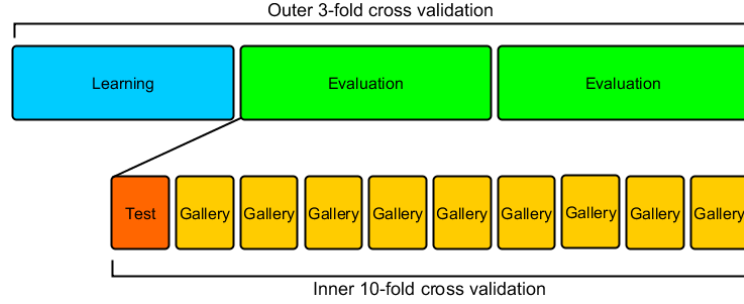
Figure 3. Nested cross-validation.

Correct Classification Rate (CCR) is often perceived as the ultimate qualitative measure, however, if a method has a low CCR, we cannot directly say if the system is failing because of bad features or a bad classifier. It is more explanatory to provide an evaluation in terms of class separability of the feature space. The class separability measures give an estimate on the recognition potential of the extracted features and do not reflect an eventual combination with an unsuitable classifier:

- *Davies-Bouldin Index:* **DBI**

$$\text{DBI} = \frac{1}{C_E} \sum_{c=1}^{C_E} \max_{1 \le c' \le C_E,\, c' \ne c} \frac{\sigma_c + \sigma_{c'}}{\widehat{\delta}\left(\widehat{\mu}_c, \widehat{\mu}_{c'}\right)} \tag{1}$$

where $\sigma_c = \frac{1}{N_c} \sum_{n=1}^{N_c} \widehat{\delta}\left(\widehat{\mathbf{g}}_n, \widehat{\mu}_c\right)$ is the average distance of all elements in identity class $\mathcal{I}_c$ to its centroid, and analogically for $\sigma_{c'}$. Templates of low intra-class distances and of high inter-class distances have a low DBI.

- *Dunn Index:* **DI**

$$\text{DI} = \frac{\min\limits_{1 \le c < c' \le C_E} \widehat{\delta}\left(\widehat{\mu}_c, \widehat{\mu}_{c'}\right)}{\max\limits_{1 \le c \le C_E} \sigma_c} \tag{2}$$

with σ_c from the above DBI. Since this criterion seeks classes with high intra-class similarity and low inter-class similarity, a high DI is more desirable.

- *Silhouette Coefficient:* **SC**

$$\text{SC} = \frac{1}{N_E} \sum_{n=1}^{N_E} \frac{b(\widehat{\mathbf{g}}_n) - a(\widehat{\mathbf{g}}_n)}{\max\{a(\widehat{\mathbf{g}}_n), b(\widehat{\mathbf{g}}_n)\}} \tag{3}$$

where $a(\widehat{\mathbf{g}}_n) = \frac{1}{N_c} \sum_{n'=1}^{N_c} \widehat{\delta}(\widehat{\mathbf{g}}_n, \widehat{\mathbf{g}}_{n'})$ is the average distance from $\widehat{\mathbf{g}}_n$ to other samples within the same identity class and $b(\widehat{\mathbf{g}}_n) = \min_{1 \leq c' \leq C_E, c' \neq c} \frac{1}{N_{c'}} \sum_{n'=1}^{N_{c'}} \widehat{\delta}(\widehat{\mathbf{g}}_n, \widehat{\mathbf{g}}_{n'})$ is the average distance of $\widehat{\mathbf{g}}_n$ to the samples in the closest class. It is clear that $-1 \leq \text{SC} \leq 1$ and SC close to one means that classes are appropriately separated.

- *Fisher's Discriminant Ratio:* **FDR**

$$\text{FDR} = \frac{\frac{1}{C} \sum_{c=1}^{C_E} \widehat{\delta}(\widehat{\mu}_c, \widehat{\mu})}{\frac{1}{N_E} \sum_{c=1}^{C_E} \sum_{n=1}^{N_c} \widehat{\delta}(\widehat{\mathbf{g}}_n, \widehat{\mu}_c)}. \tag{4}$$

High FDR is preferred for classes of low intra-class sparsity and high inter-class sparsity.

Apart from analyzing the distribution of templates in the feature space, it is schematic to combine the features with a rank-based classifier and to evaluate the system based on distance distribution with respect to a probe. For obtaining a more applied performance evaluation, we evaluate:

- *Cumulative Match Characteristic:* **CMC**
  Sequence of Rank-k (for k on X axis from 1 up to C_E) recognition rates (Y axis) for measuring ranking capabilities of a recognition method. Its headline Rank-1 is the well-known **CCR**.
- *False Accept Rate vs. False Reject Rate:* **FAR/FRR**
  Two sequences of the error rates (Y axis) as functions of discrimination threshold (X axis). Each method has a value e of this threshold giving Equal Error Rate (**EER**=FAR=FRR).
- *Receiver Operating Characteristic:* **ROC**
  Sequence of True Accept Rate (**TAR**) and False Accept Rate (**FAR**) with a varied discrimination threshold. For a given threshold the system signals both TAR (Y axis) and FAR (X axis). Area Under Curve (**AUC**) is computed as the integral of the ROC curve.
- *Recall vs. Precision:* **RCL/PCN**
  Sequence of rates with a varied discrimination threshold. For a given threshold the system signalizes both RCL (X axis) and PCN (Y axis). The value of Mean Average Precision (**MAP**) is computed as the area under RCL/PCN curve.

These measures reflect how well the feature is class-separated and how much it takes to confuse the identities of two people. They do not, in fact, provide complementary information, however, a quality evaluation framework should be able to evaluate the most popular measures. Each measure is evaluated in the context of a particular application. For example, a hotel lobby authentication system could use a high Rank-3 at the CMC, while a city-level person tracking system is likely to need the ROC curve leaning towards the upper left corner.

8

5 Reproducing the Experiments

This section provides a description of the framework we implemented and the database we extracted. With this manual, a reader should be able to reproduce the evaluation and to use the implementation for recognizing people. All source codes including (1) database extraction drive, (2) implementations of the proposed and all relevant methods, (3) classifier learning and classification mechanisms and (4) evaluation mechanism and metrics, are available at our departmental Git repository [1]. The original CMU MoCap database and extracted databases are available online at our research group web page [2].

Executor.java is the main execution class. Set all parameters of evaluation and file locations and distanceThreshold, then select which actions to perform and finally, select the evaluation setups. The class contains the main(String[] args) method. It contains four methods to select for execution:

extractDatabase() for extracting an experimental database from the original CMU MoCap database — this is also available for download at our page as original.zip. To run this method, unzip to get the files gaitcycle.amc (exemplary gait cycle) and skeleton.asf (prototypical skeleton) and the directory amcOriginal (original AMC files). Extraction begins with normalization with respect to a person's position and walk direction as provided in the normalized.zip file. Clean gait cycles are then filtered out by the distance threshold (see last paragraph of Section 2) that numerically expresses how much extracted motions resemble gait cycles, that is, the lower the distance threshold, the fewer and cleaner the gait cycles. Set a value for distanceThreshold to produce a folder of the extracted database. Evaluations in [7,8] are set with 302.0, extracting a database of 54 identities and 3,843 gait cycles. A higher distance threshold will qualify some non-gait motions.

learnClassifiers() for learning classifiers of all implemented methods on a subdatabase determined by the distance threshold. Set a value for distanceThreshold (such as 302.0) and provide the corresponding directory of the learning database (such as amc302.0 in extracted-302.0.zip) and the learned classifiers appear in the classifiers folder.

performClassification() for performing a classification of a custom probe/query on a custom gallery with a custom classifier. Set file locations for the classifier file customClassifier, the probe gait cycle customQueryFileAMC and the gallery directory customGalleryDirectory. Results are printed on the standard output.

evaluateMethods() for evaluating the implemented methods in homogeneous and heterogeneous setups. To skip database extraction, one could supply a provided extracted database (such as amc302.0 in extracted-302.0.zip) and a skeleton file (such as skeleton.asf in any extracted database ZIP file). Our page provides additional databases categorised according to various values of distanceThreshold. The results are set to be printed on the standard output but we suggest to redirect it to a CSV file. Results of individual evaluation attempts vary slightly as different learning, testing and gallery sets are randomly selected upon each attempt.

Compile to obtain Gait.jar. The main project location should also contain the lib directory and all necessary files and directories depending on which actions are to be executed. Run command $ java -jar Gait.jar > output.csv.

The output file (see structure in Table 1) contains the performance metrics as specified in Section 4 and the information about average distance computation time (**DCT**) in milliseconds and average template dimensionality (**TD**). The evaluation results are in the form of one value per coefficient (see results in Table 2) and the sequences CMC, FAR/FRR, ROC=TAR/FAR and RCL/PCN. The CMC sequence contains C_E values, one for each k in the Rank-k recognition rate. The other three pairs of sequences are normalized to 30 values by `method.setFineness(30)`. The FAR/FRR sequences of all methods are normalized to the discrimination threshold with respect to the first value of FAR=0 and FRR=1, and to the middle value that represents EER where all sequences cross. The ROC sequences are normalized with respect to the first value of TAR=FAR=0 and to the last value of TAR=FAR=1. Finally, the RCL/PCN sequences are normalized with respect to the first value of RCL=0 and to the last value of RCL=1.

Table 1. Structure of the output file.

{method name}, {distance threshold}
`DBI DI SC FDR CCR EER AUC MAP DCT TD`
{1 line – one value for each coefficient}
`CMC`
{C_E lines – the CMC sequence}
`FAR FRR TAR FAR RCL PCN`
{30 lines – all six sequences}

Table 2. First line results of all 20 implemented methods on the 302.0 database.

method	class separability coefficients				classification based metrics				scalability	
	DBI	DI	SC	FDR	CCR	EER	AUC	MAP	DCT	TD
Ahmed	216.2	0.842	−0.246	0.954	0.657	0.38	0.659	0.165	**<1**	24
Ali	501.5	0.26	−0.463	1.175	0.225	0.384	0.679	0.111	**<1**	2
Andersson	142.3	1.297	−0.102	1.127	0.84	0.343	0.715	0.251	**<1**	68
Ball	161	1.458	−0.163	1.117	0.75	0.346	0.711	0.231	**<1**	18
Dikovski	144.5	1.817	−0.135	**1.227**	0.881	0.363	0.695	0.254	**<1**	71
Gavrilova	185.8	1.708	−0.164	0.77	0.891	0.374	0.677	0.254	45	5,254
Jiang	206.6	1.802	−0.249	0.85	0.811	0.395	0.657	0.242	8	584
Krzeszowski	154.1	1.982	−0.147	0.874	0.915	0.392	0.662	0.275	35	3,795
Kumar	**118.6**	1.618	−0.086	1.09	0.801	0.459	0.631	0.217	8	13,950
Kwolek	150.9	1.348	−0.084	1.175	0.896	0.358	0.723	0.323	**<1**	660
Preis	1,980.6	0.055	−0.512	1.067	0.143	0.401	0.626	0.067	**<1**	13
Sedmidubsky	398.1	1.35	−0.425	0.811	0.543	0.388	0.657	0.149	**<1**	292
Sinha	214.8	1.112	−0.215	1.101	0.674	0.356	0.697	0.191	**<1**	45
_MMC$_{BR}$	154.2	1.638	**0.062**	1.173	0.925	**0.297**	**0.748**	0.353	**<1**	53
_MMC$_{JC}$	130.3	1.891	0.051	1.106	0.918	0.378	0.721	0.315	**<1**	51
_PCALDA$_{BR}$	182	1.596	−0.015	0.984	0.918	0.361	0.695	0.276	**<1**	54
_PCALDA$_{JC}$	174.4	1.309	−0.091	0.827	0.863	0.44	0.643	0.201	**<1**	54
_Random					0.042					**0**
_Raw$_{BR}$	163.7	**2.092**	0.011	0.948	**0.966**	0.315	0.743	**0.358**	70	8,229
_Raw$_{JC}$	155.1	1.954	−0.12	0.897	0.926	0.377	0.679	0.283	161	13,574

To reproduce the experiments in Table 2, follow the instructions in the README file at [1] in the `reproduce` folder. Please note that some methods are slow even on a leading edge hardware. Learning and evaluation times in Table 3 were measured on a computer with Intel® Xeon® CPU E5-2650 v2 @ 2.60GHz and 256 GB RAM.

Table 3. Evaluation times of the methods in Table 2. Units: s seconds, m minutes, h hours, d days.

method	time	method	time	method	time	method	time
Ahmed	48.6 m	Gavrilova	10.3 d	Preis	48.7 m	_PCALDA$_{BR}$	4.7 h
Ali	40.9 m	Jiang	1.9 d	Sedmidubsky	1.4 d	_PCALDA$_{JC}$	10.9 h
Andersson	45.7 m	Krzeszowski	8.1 d	Sinha	49.6 m	_Random	27.9 m
Ball	48.5 m	Kumar	1.8 d	_MMC$_{BR}$	2.6 h	_Raw$_{BR}$	16.1 d
Dikovski	50.7 m	Kwolek	1.1 h	_MMC$_{JC}$	3.0 h	_Raw$_{JC}$	36.7 d

The goal of the MMC-based learning is to find a linear discriminant that maximizes the misclassification margin. This optimization technique appears to be more effective than designing geometric gait features. Table 2 indicates the best results for the MMC on bone rotational data: highest SC, EER and AUC, and competitive DBI, DI, FDR, CCR and MAP. In terms of the Correct Classification Rate metric, our MMC method was only outperformed by the Raw method, which is implemented here as a form of baseline. We interpret the high scores as a sign of robustness.

Apart from the performance merits, the MMC method is also efficient: relatively low-dimensional templates and Mahalanobis distance ensure fast distance computations and thus contribute to high scalability. Note that even if the Raw method has some of the best results, it can hardly be used in practice due to its extreme consumption of time and space resources. On the other hand, Random has no features but cannot be considered a serious recognition method. To illustrate the evaluation time, calculating the distance matrix (a matrix of distances between all evaluation templates) took a couple minutes for the MMC method, almost nothing for the Random method, and more than two weeks for the Raw method. To conclude, the MMC method on bone rotational data appears to be an optimal trade-off between effectiveness and efficiency, and thus the new state-of-the-art in feature extraction for MoCap-based gait recognition.

6 Summary and Future Work

As our contribution to reproducible research, we have provided implementation details and source codes [1] of our evaluation framework for gait recognition [7,8]. The software implements the proposed method as well as all related methods. We include the evaluation database [2] together with source codes for its extraction from the general CMU MoCap database. We also attach the description and portable software for evaluating class separability coefficients of extracted features and classifier performance metrics. Finally, we provide documentation and installation instructions for easy and straightforward reproducibility of the experiments.

As demonstrated by outperforming other methods in four class separability coefficients and four classification metrics, the proposed features learning mechanism has a strong potential in gait recognition applications. Even though we believe that MMC is

the most suitable criterion for optimizing gait features, we continue to research further potential optimality criteria and machine learning approaches.

We hope that the evaluation framework and database presented here will contribute to smooth development and evaluation of further novel MoCap-based gait recognition methods. All used data and source codes have been made available [1,2] under the Creative Commons Attribution license (CC-BY) for database and the Apache 2.0 license for software, which grant free use and allow for experimental evaluation. We encourage all readers and developers of MoCap-based gait recognition methods to contribute to the framework with new algorithms, data and improvements.

References

1. Evaluation framework for mocap-based gait recognition methods, `https://gitlab.fi.muni.cz/xbalazia/GaitRecognition`, commit hash 0ac9b0dd from 25th November 2016
2. Extracted MoCap Gait Databases, `https://gait.fi.muni.cz/#database`, Faculty of Informatics, Masaryk University, Brno
3. Ahmed, F., Paul, P.P., Gavrilova, M.L.: DTW-Based Kernel and Rank-Level Fusion for 3D Gait Recognition Using Kinect. The Visual Computer 31(6), 915–924 (2015), `https://doi.org/10.1007/s00371-015-1092-0`
4. Ahmed, M., Al-Jawad, N., Sabir, A.: Gait Recognition Based on Kinect Sensor. Proc. SPIE 9139, 91390B–91390B–10 (2014), `http://dx.doi.org/10.1117/12.2052588B`
5. Ali, S., Wu, Z., Li, X., Saeed, N., Wang, D., Zhou, M.: Transactions on Computational Science XXVI: Special Issue on Cyberworlds and Cybersecurity, chap. Applying Geometric Function on Sensors 3D Gait Data for Human Identification, pp. 125–141. Springer, Berlin, Heidelberg (2016), `https://doi.org/10.1007/978-3-662-49247-5_8`
6. Andersson, V.O., Araujo, R.M.: Person Identification Using Anthropometric and Gait Data from Kinect Sensor. In: Proc. of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15). pp. 425–431. AAAI Press (2015), `http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9680`
7. Balazia, M., Sojka, P.: Learning Robust Features for Gait Recognition by Maximum Margin Criterion. In: Proc. of 23rd International Conference on Pattern Recognition, ICPR 2016. pp. 901–906. IEEE (Dec 2016)
8. Balazia, M., Sojka, P.: Walker-independent features for gait recognition from motion capture data. In: Robles-Kelly, A., Loog, M., Biggio, B., Escolano, F., Wilson, R. (eds.) Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop, S+SSPR 2016, Mérida, Mexico, November 29–December 2, 2016, Proceedings. pp. 310–321. Springer, Cham (2016), `https://doi.org/10.1007/978-3-319-49055-7_28`
9. Ball, A., Rye, D., Ramos, F., Velonaki, M.: Unsupervised clustering of people from 'skeleton' data. In: Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction. pp. 225–226. HRI '12, ACM, New York, NY, USA (2012), `http://doi.acm.org/10.1145/2157689.2157767`
10. Chen, X., Koskela, M.: Classification of RGB-D and Motion Capture Sequences Using Extreme Learning Machine. In: Kämäräinen, J.K., Koskela, M. (eds.) Proc. of Image Analysis: 18th Scandinavian Conference, SCIA 2013, Espoo, Finland, June 17–20, 2013. pp. 640–651. Springer (2013), `https://doi.org/10.1007/978-3-642-38886-6_60`

11. CMU Graphics Lab: Carnegie-Mellon Motion Capture (MoCap) Database (2003), http://mocap.cs.cmu.edu

12. Dikovski, B., Madjarov, G., Gjorgjevikj, D.: Evaluation of Different Feature Sets for Gait Recognition Using Skeletal Data from Kinect. In: 37th Intl. Convention on Information and Communication Technology, Electronics and Microelectronics. pp. 1304–1308 (May 2014), http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6859769

13. Fisher, R.A.: The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics 7(2), 179–188 (1936), https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

14. Jiang, S., Wang, Y., Zhang, Y., Sun, J.: Real Time Gait Recognition System Based on Kinect Skeleton Feature. In: Jawahar, C., Shan, S. (eds.) Computer Vision – ACCV 2014 Workshops, LNCS, vol. 9008, pp. 46–57. Springer (2015), https://doi.org/10.1007/978-3-319-16628-5_4

15. Kastaniotis, D., Theodorakopoulos, I., Theoharatos, C., Economou, G., Fotopoulos, S.: A Framework for Gait-Based Recognition Using Kinect. Pattern Recognition Letters 68, Part 2, 327–335 (2015), http://www.sciencedirect.com/science/article/pii/S0167865515001920

16. Krzeszowski, T., Switonski, A., Kwolek, B., Josinski, H., Wojciechowski, K.: DTW-Based Gait Recognition from Recovered 3-D Joint Angles and Inter-ankle Distance. In: Chmielewski, L.J., Kozera, R., Shin, B.S., Wojciechowski, K. (eds.) Proc. of Computer Vision and Graphics: International Conference, ICCVG 2014, Warsaw, Poland. LNCS, vol. 8671, pp. 356–363. Springer (Sep 2014), https://doi.org/10.1007/978-3-319-11331-9_43

17. Kumar, M.S.N., Babu, R.V.: Human gait recognition using depth camera: A covariance based approach. In: Proc. of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing. pp. 20:1–20:6. ICVGIP '12, ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2425333.2425353

18. Kwolek, B., Krzeszowski, T., Michalczuk, A., Josinski, H.: 3D Gait Recognition Using Spatio-Temporal Motion Descriptors. In: Proc. of Intelligent Information and Database Systems: 6th Asian Conference, ACIIDS 2014, Bangkok, Thailand, Part II. LNCS, vol. 8398, pp. 595–604. Springer (Apr 2014), https://doi.org/10.1007/978-3-319-05458-2_61

19. Preis, J., Kessel, M., Werner, M., Linnhoff-Popien, C.: Gait Recognition with Kinect. In: 1st International Workshop on Kinect in Pervasive Computing, New Castle, UK, June 18–22. pp. 1–4 (2012), https://www.researchgate.net/publication/239862819_Gait_Recognition_with_Kinect

20. Ramu Reddy, V., Chakravarty, K., Aniruddha, S.: Person Identification in Natural Static Postures Using Kinect. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) Computer Vision – ECCV 2014 Workshops: Zurich, Switzerland, September 6–7 and 12, 2014, Proceedings, Part II. LNCS, vol. 8926, pp. 793–808. Springer (2015), https://doi.org/10.1007/978-3-319-16181-5_60

21. Sedmidubsky, J., Valcik, J., Balazia, M., Zezula, P.: Gait Recognition Based on Normalized Walk Cycles. In: Advances in Visual Computing, LNCS, vol. 7432, pp. 11–20. Springer (2012), https://doi.org/10.1007/978-3-642-33191-6_2

22. Sinha, A., Chakravarty, K., Bhowmick, B.: Person Identification Using Skeleton Information from Kinect. In: ACHI 2013: Proc. of the Sixth Intl. Conf. on Advances in CHI. pp. 101–108 (2013), https://www.thinkmind.org/index.php?view=article&articleid=achi_2013_4_50_20187

23. Su, H., Liao, Z.W., Chen, G.Y.: A gait recognition method using L1-PCA and LDA. In: Proc. of the Eighth Intl. Conf. on Machine Learning and Cybernetics. vol. 6, pp. 3198–3203 (Jul 2009), https://dx.org/10.1109/ICMLC.2009.5212776

24. Valcik, J., Sedmidubsky, J., Zezula, P.: Assessing Similarity Models for Human-Motion Retrieval Applications. Computer Animation and Virtual Worlds 27(5), 484–500 (2016), http://dx.doi.org/10.1002/cav.1674