Literature

Omar Chatila

May 2024

1 Main Paper

1.1 Introduction

- VR Coaching system that detects motor errors and provides RT verbal and visual feedback
- Requirements:
  1. Compatibility to existing feedback systems
  2. Real-time feedback: Depending on the task, RT feedback is required to prevent injury
  3. Interpretability: info on classified errors should be able to gen augmented feedback and verifiable by experts
  4. Conservative size of data sets: system should perform with limited, high quality data
  5. Minimal manual work: low manual input / expert knowledge should be required

Contributions: New RT pipeline which uses reference-based DTW of movement prefixes as basis for feature selection with Random Forests for classification with SVM. Beats KNN-DTW and CNN approach:

1. better classifiacation results
2. better suited for generation of augmented feedback
3. higher versatility and compatibility due to use of skeleton data, tracking joints (common interface)

1.2 Related work

1NN-DTW Yurtman and Barshan: Extension of DTW that can detect multiple exercise types and error patterns by comparing the input trajectory to multiple reference trajectories where each reference trajectory contains one error type. Classification through calculating the single best match based on

DTW distance. Problem: Combinations of error patterns can only be spotted if pre-recorded template with multiple errors exists. Also this approach is only suitable for single-subject evaluation.

Problems with data based approaches

1. bad generalization to new subjects. System needs to be retrained for each user.
2. Unprecise error assessment (good or bad). Small number of joints. Individual styles are not considered, rigid comparison to reference
3. Overall difference is regarded. Problem: irrelevant body parts with deviations lead to wrong classification. hand movement irrelevant for squats but wrong wrist movement leads to high deviation $\Rightarrow$ error-detection

1.3 General approaches for human activity recognition

DTW Dynamic Time Warping is an algorithm used in the analysis of time series to measure the similarity between two time series $X = x_1, x_2, \ldots, x_n$ and $Y = y_1, y_2, \ldots, y_n$ that may vary in time or speed. It uses dynamic programming to calculate the distance between two matching points (x_i, y_j), aiming to minimize the overall alignment cost, calculated by a pre-determined distance function $d(x_i, y_j)$.
Highly suitable for motion classification.

KNN-DTW K-Nearest-Neighbors (KNN) is an algorithm which finds the k nearest data points for a query input based on a pre-determined distance measure. KNN-DTW employs DTW distance to calculate the similarity between two time series. Once the DTW distance has been calculated between the query time series and all other time series in the dataset, the algorithm identifies the k-nearest neighbors based on the smallest DTW distances.

1.4 Domain and data set

- Identification of error patterns with help of coaches
- Record motion data with Optitrack motion capture system, motion capture suit with markers
- 19 joints at 120 Hz: k joint rotations and positions
- joint rotations as quaternions/euler angles, **root joint: hips**. positions as vectors: translate to root

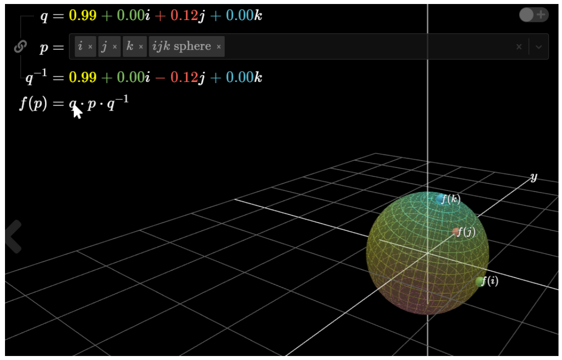

Figure 1: Quaternions

1.5 Classification

Pipeline:

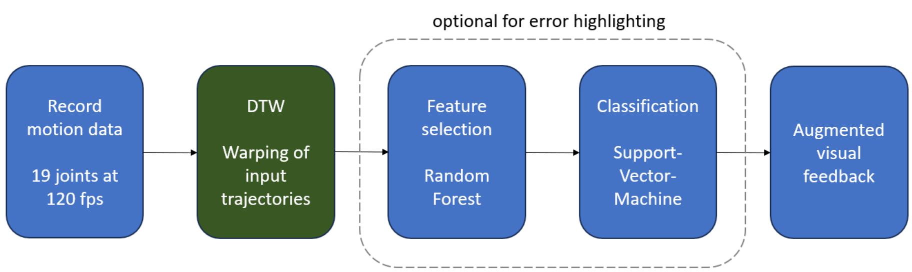

Figure 2: Hülsmann's classification pipeline

Input: Stream of frames of skeleton data from motion capture system
Output: Label w.r.t. each error pattern

- Motion data is recorded at 120 Hz for 19 joints, capturing rotation and position for each joint per frame.
- Dynamic Time Warping aligns each frame of the data to a reference trajectory of fixed time frame.
- Random Forests filter out irrelevant features, like wrist rotations during a back exercise.
- Support Vector Machine classifies motion data, providing the basis for visual augmented feedback. This feedback highlights faulty areas on the trainee's body.

Classic DTW can only classify once the motion is completed.

Two extensions:

1.5.1 Weight-Optimized Open-End DTW

- Source: Accurate Online Alignment of Human Motor Performances (Hülsmann et al.)
- Open-End-DTW can lead to poor alignments, much worse than offline DTW
- Problem: if alignment fails, such that algo decides final frame belongs to earlier reference frame, alignment becomes useless because further steps will fail
- WOOEDTW with path-length weighting combined with joint weights is proposed
- data driven weighting: low min manual labor
- Related approaches on feature weighting:
  Existing approaches focus on overall movement or variance of features which may neglect important joints with small movement. Unimportant joints with mostly non-functional movement might be prioritized over important ones
- Proposal of alternative approach using optimization of DTW weights. Introduce an error measure for DTW alignments to optimize weights
- Path-length weighting is applied to make DTW independent from assumptions on the movements' timing, mitigating the bias induced by DTW penalties.
- Each feature on the whole temporal axis is weighted equally in the path-length weighting approach, unlike some previous methods where later frames implicitly gain more weight for DTW.
- Only requires one additional matrix storing the paths' lengths
- Secnario and Dataset:
  - joint angles as features, quaternion representation accommodating to different body types
  - root axis: hip. Reference of joints: parents
- OE-DTW: Source: Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation (Tormene, 2009) Open-End DTW (OE-DTW) allows aligning a prefix T1 of a query trajectory with a complete reference trajectory T2 It provides a warp and estimates which frame in the reference matches the last frame of T1, allowing the backtracing step to start from a different point than

in traditional DTW. OE-DTW computes the dissimilarity or distance between one input and the best matching forepart of a reference trajectory.

$$D_{OE}(X,Y) = min(D_{DTW}(X, Y^{(j)}), j = 1, \ldots, M$$

- **Path-length-weighting**:
  - Accumulated cost matrix biased towards shorter paths, problem because in practice shorter alignments are not better than longer ones
  - Scenario: two similar actions with same speed lead to alignment along diagonal. If one pauses, optimal alignment should leave diagonal. Focus on shorter paths however would compel algorithm to stay on diagonal.
  - Proposal: Path-length weighing with Matrix L containing lengths of optimal paths to mitigate bias towards short paths:

$$\mathrm{D}(i,j) = \mathrm{M}(i-1,j-1) + \mathrm{D}\left(\operatorname*{argmin}_{(k,l)}\left(\frac{\mathrm{D}(k,l)+\mathrm{M}(i-1,j-1)}{\mathrm{L}(k,l)+1}\right)\right), \quad \text{where } (k,l) \in \{(i-1,j-1),(i-1,j),(i,j-1)\}. \tag{3}$$

Matrix L contains the path-lengths of each optimal path. It is updated together with $\mathrm{D}(i,j)$ based on the just calculated values for k and l. After calculating D and L, we determine the optimal path via backtracing from $\mathrm{D}(|T_1|+1, \Omega)$. In each step, we divide all examined cells of the accumulated cost matrix D by their corresponding path-lengths from L and select the one with the smallest result.

Figure 3: M: distance betw i, j. D: Accumulated cost matrix

- **Priority based weighting of joints**
  - Certain joints are more important than others.
  - Distance between X and Y with regard to priority weighting:

$$M(i,j) = \sum_{d=1}^{k} w_d(1 - |q_{i,d} \cdot q_{j,d}|$$

  - data-driven approach to find optimal weights

1.5.2 prefix based approach

2 Algorithms

asdfdasf