



Article

Evaluation of Pattern Recognition Methods for Head Gesture-Based Interface of a Virtual Reality Helmet Equipped with a Single IMU Sensor

Tomasz Hachaj *  and Marcin Piekarczyk 

Institute of Computer Science, Pedagogical University of Krakow, 2 Podchorazych Ave, 30-084 Krakow, Poland; marcin.piekarczyk@up.krakow.pl

* Correspondence: tomekhachaj@o2.pl

Received: 4 November 2019; Accepted: 6 December 2019; Published: 8 December 2019



Abstract: The motivation of this paper is to examine the effectiveness of state-of-the-art and newly proposed motion capture pattern recognition methods in the task of head gesture classifications. The head gestures are designed for a user interface that utilizes a virtual reality helmet equipped with an internal measurement unit (IMU) sensor that has 6-axis accelerometer and gyroscope. We will validate a classifier that uses Principal Components Analysis (PCA)-based features with various numbers of dimensions, a two-stage PCA-based method, a feedforward artificial neural network, and random forest. Moreover, we will also propose a Dynamic Time Warping (DTW) classifier trained with extension of DTW Barycenter Averaging (DBA) algorithm that utilizes quaternion averaging and a bagged variation of previous method (DTWb) that utilizes many DTW classifiers that perform voting. The evaluation has been performed on 975 head gesture recordings in seven classes acquired from 12 persons. The highest value of recognition rate in a leave-one-out test has been obtained for DTWb and it equals 0.975 (0.026 better than the best of state-of-the-art methods to which we have compared our approach). Among the most important applications of the proposed method is improving life quality for people who are disabled below the neck by supporting, for example, an assistive autonomous power chair with a head gesture interface or remote controlled interfaces in robotics.

Keywords: IMU sensors; pattern recognition; motion capture; head gestures; dynamic time warping; quaternions; principal components analysis

1. Introduction

Virtual reality helmet (VRH) is a head-mounted device that allows a user to interact with a three-dimensional virtual reality environment by displaying stereovision images. VRH also analyses user's head position and adjusts the virtual camera position in the virtual reality world in order to enhance the immersive experience.

Virtual reality systems might contain several types of user interface depending on the application of such solution. It can be either full-body motion capture (MoCap), both visual and sensor-based, hand gestures-based interface supported by various handled devices, gaze-based interface or head motion interface (also called head gestures interface). The last type of interface might be used without any additional handled devices and motion acquisition systems beside internal measurement unit (IMU) sensors in VRH. Gaze gestures are a promising input technology for wearable devices, especially in smart glasses form because gaze gesturing is unobtrusive and leaves the hands free for other tasks [1,2]. Gaze gestures utilize eye tracking technology and, to be reliable, it requires dedicated hardware. Alternatively, a system can track user head motion trajectory and analyze spatial motion patterns. This type of interface depends less on specialized cameras and can be developed using even IMU of

smartphones. The contemporary IMU of smartphone devices has many applications. Among them is analysis and recognition of human activity—a survey on this subject can be found in [3,4]. Beside IMU sensors, an up to date smartphone device contains efficient GPU and high resolution screen. Thanks to those features, smartphones can be used to construct a relatively affordable component of virtual reality glasses/helmets [5]. Head gestures require a pattern recognition algorithm that would enable unambiguous recognition of user head trajectory. In contrast to other types of human activity recognition this problem has not been investigated as much as, for example, gait or hand motions classification and analysis. In the following section we will present an up-to-date summary of the most effective methods for head motion analysis and for similar human motion evaluation tasks.

1.1. Head Motion Analysis and Classification

Surprisingly, to our best knowledge there are only a few papers that directly address the problem of head gestures of application for user interface or who discuss the broader subject of head gestures recognition. Some of them present a very initial study and in some cases, a lack of available data and detailed description of applied methods makes the presented results impossible to reproduce. In [6] the authors have applied Dynamic Time Warping (DTW) algorithm in order to calculate the similarity between time sequences acquired with IMU sensors in order to classify four classes of actions. Application of weighted DTW classifier for head motions acquired by Smart Glasses is reported in [7]. However, due to a lack of detailed description of methodology and data it is hardly possible to reproduce the method and results. In [8,9], Hidden Markov Models (HMM) are used for head motion classification to support sign gestures recognition and recognition of four types of actions respectively. Those two papers also do not use publicly available datasets. In summary, methods that are commonly applied in head gestures classification works on unprocessed time series, which means that those classifiers take as an input data a multivariate time series that is segmented from an incoming data stream. Especially popular is DTW time series classification algorithm which utilizes the similarity function to determine the similarity of the time series pairs [10,11]. Head gestures recognition, however, is a special case of other body parts motion recognition or more generally human motions or gait recognition. There are no contradictions in applying the most effective methods of those types to head trajectory classification. We will discuss this in the following section.

1.2. Effective Methods of Human Motion Analysis and Classification

While head motion in VRH is often tracked using a single IMU sensor, the produced time series involve either a three or four dimensional signal depending of type of motion description that is applied. It is three dimensional time series if the data is represented as rotation using Euler angles or if it represented as coordinates of normalized vectors that indicate head orientation (colloquially speaking, the direction of those vectors shows direction of the user's nose). It is a four dimensional time series if rotation is represented as quaternions. Due to this fact, some methods present in literature also utilize three or four dimensional time series with motion descriptions (as rotation or vector direction) that can also be applied for head gestures recognition. Those methods (for example, for hand motion classification) utilize various classifiers depending on motions features selection, for example DTW, HMM or when features are derived after processing original motion capture data nearest neighbor classifier (NN), support vector machine (SVM) or neural network [12,13]. A good example of this is [14] where the authors apply HMM to classify multidimensional hand motion data composed of Euler angles rotation, gyroscope and accelerometer data. An algorithm from [15] uses the Naive-Bayes-nearest-neighbor classifier to measure distance between joint-based histogram representation of full body motion and the action to be classified.

Analysis of full body MoCap is a more general problem, because it utilizes several dozen body joints measurements. Those tasks require methodologies that enable scaling influence (importance) of separate joints on overall data recognition. Among the methods that are becoming more and more popular in the last years are neural network (ANN) and deep neural network (DNN)-based

methods. If action recognition is performed on raw video data authors prefer to use convolution neural networks where convolution layer is used to generate features. Those features are then processed by a fully connected neural network which performs classification [16]. Sometimes an input raw signal is processed by convolution layer followed by recurrent network to avoid a sliding window design and then classified by a fully connected neural network [17]. In situations when appropriate motions features are already present an ANN can be applied directly [18,19]. In some research, authors also utilize DTW [20]. However, as can be seen in papers that contain a comprehensive survey on this subject [12,13,21,22] Principal Components Analysis (PCA) is a commonly used procedure for dimensionality reduction and features set generation for human motions analysis and classification. Because application of PCA in various steps of motion analysis and classification seems to be a standard approach for gait/full body motion analysis (although it was not mentioned in state-of-the-art papers devoted to head gestures recognition) the rest of this survey will be devoted to PCA.

Generally, PCA-based time series processing is used either for dimensionality reduction before a signal is classified by a pattern recognition algorithm or another type of analyses. This is done because in many cases distance-based classifiers perform better in lower dimensional space. The choice of following pattern recognition method or analysis procedure depends on what the goal authors wanted to obtain. In the case of classification, authors use, for example, HMM [23], DTW [24], nearest neighbors [25–27], and SVM [28]. Authors also perform various types of analysis for example aligning signals with Two Step Aligning algorithm [29], drawing direct conclusions from principal components [30–33] or data segmentation [34]. The second application of PCA is for generating motion features of fixed size that are vector of real values which represents features of linear combination of eigenvectors that can be also classified by general-purpose classifiers like, for example, SVM [35–37] or nearest neighbors [38]. The classification effectiveness depends mostly on type of activity authors wanted to recognize the quality and variability of data. There are also some methods that utilize both dimensionality reduction and Eigen-based features to perform motion classification, for example, [39].

1.3. Motivations of This Paper

As can be seen in previous section head gestures classification and its application for virtual reality interfaces have not been a target of extensive research yet. This might be because of the fact that VRG based on smartphone hardware (IMU, GPU and screen) has become widely available in last few years and there was not much interest in this type of pattern recognition solution before. MoCap technology was either used for whole body (gait/posture recognition and analysis) by general purpose MoCap systems or a dedicated IMU-based hardwares were constructed for certain tasks (see for example [6]). The motivation of this paper is to fill that gap by examining the effectiveness of state-of-the-art and newly proposed MoCap pattern recognition methods on the task of head gestures classifications. We will validate a classifier based on PCA features with various numbers of dimensions for the two-stage PCA recognition method [39]. In [18], the author performs classification of a high-frequency EMG seven-dimensional signal using several pattern recognition techniques among them artificial neural network (ANN) and decision trees (DT). We adapted methods from that paper to our pattern recognition tasks. We have replace however DT with a Random Forest (RF) classifier to prevent a single DT overfitting to training dataset. In this paper we introduce DTW classifier trained with extension of DBA algorithm that utilizes quaternion averaging [40] and we will also proposes a bagged variation of previous method that utilizes many DTW classifiers that perform voting to obtain final classification. The last classifier and its training procedure based on [40] is our original contribution. Both an original data set that was used for this evaluation, Android OS software for data acquisition and all R-language source codes we used to conduct the research is available to download in order to make our research reproducible and to check methodology and implementation details (The source codes and data can be download from <https://github.com/browarsoftware/headmotions>). Some methods that we are comparing with our approach, although they were developed several years ago, are extensively used in up-to-date researches like for example DTW-based classification [14] or various

application of PCA-based dimensionality reduction and features selection [21,32]. Also, SVM and KNN classifiers [15] are often utilized to classify data acquired by IMU sensors. The most important difference between all those papers is that depending on the motion acquisition technique, authors need to perform various preprocessing of the MoCap signal to make it applicable to dimensionality reduction and pattern recognition techniques. Those methods are specific to signal domain (for example raw video data, EMG, full body MoCap, so-called skeleton) and are not within the scope of this paper. As we previously mentioned in Section 1.1, there are not many up to date papers devoted to head gestures so we evaluate the proposed approaches with methods that were successful in other motion types.

A very important factor that was not taken into account in previous research on head motion analysis is supporting a user who performs the head motion with some visual feedback: an actual trajectory of his/her head. Such type of feedback (for example drawing points that indicate actual head direction in virtual or augmented reality technology) might highly increase the awareness of motion trajectories that are drawn by a user's head. The dataset we used in this research was gathered using a VRH with systems that draw such a trajectory. This is why we used whole VRH rather than only an IMU sensor.

Methods presented in this paper have many important applications. Among the most important are the possibility of improving life quality for disabled people below the neck for example by supporting assistive autonomous powerchair with head gestures interface (see [6]) or remote controlling interfaces in robotics [41,42]. Technologies that utilize head-mounted devices have been mentioned in several researches in the last years. Among its possible applications is augmented reality technology for pedestrian collision warning [43], driver drowsiness detection systems [44], eye tracking-based interfaces [45], and drivers' distraction detection [46,47]. Especially the last type of application might be a promising field of application for our algorithm because some head positions and motions might be an indicator that a driver is sleepy or falls asleep. We have to remember, however, that while using IMU sensors in high speed vehicles like cars, we should utilize an additional method to distinguish the motion of the vehicle and the device's motion in the vehicle [48] and also compensate for vehicle shaking [49].

2. Materials and Methods

2.1. Evaluation Dataset

Because we have not found a publicly available dataset that contains head MoCap data acquired by smartphone-based VHR the dedicated dataset has been created. The data we gathered is saved as a three-dimensional stream of Euler angles rotations of head and four-dimensional stream of quaternions that represents the same data. Let us recall some elements of quaternion's algebra that are necessary to define the methods we have used. Let us assume that we have two quaternions: $q_1 = [x_1, y_1, z_1, w_1]$, $q_2 = [x_2, y_2, z_2, w_2]$ where x, y, z and w are quaternion coordinates.

Quaternion normalization:

$$U_{q_1} = \frac{q_1}{\sqrt{x_1^2 + y_1^2 + z_1^2 + w_1^2}} \quad (1)$$

Multiplication of quaternions q_1 and q_2 :

$$q_1 \cdot q_2 = \begin{bmatrix} w_2 \cdot x_1 + x_2 \cdot w_1 + y_2 \cdot z_1 - z_2 \cdot y_1, \\ w_2 \cdot y_1 + y_2 \cdot w_1 + z_2 \cdot x_1 - x_2 \cdot z_1, \\ w_2 \cdot z_1 + z_2 \cdot w_1 + x_2 \cdot y_1 - y_2 \cdot x_1, \\ w_2 \cdot w_1 - x_2 \cdot x_1 - y_2 \cdot y_1 - z_2 \cdot z_1 \end{bmatrix} \quad (2)$$

Inverse of quaternion q_1 :

$$q_1^{-1} = \frac{[-x_1, -y_1, -z_1, w_1]}{d} \quad (3)$$

$$d = \sqrt{x_1^2 + y_1^2 + z_1^2 + w_1^2}$$

Angle between quaternions q_1 and q_2 :

$$q_3 = U_{q_1} \cdot U_{q_2}^{-1}$$

$$\alpha = 2 \cdot \text{acos}(w_3) \quad (4)$$

$$\angle(q_1, q_2) = \begin{cases} \alpha & \text{if } \alpha \leq \pi \\ -\alpha & \text{if } \alpha > \pi \end{cases}$$

where $U_{q_2}^{-1}$ is an inverse of normalized quaternion q_2 and w_3 is a w component of quaternion q_3 .

The evaluation dataset consisted of motions performed by twelve persons: eight males and four females whose ages varied from 15 to 61. All those persons were asked to perform seven types of head gestures at least ten times each; however, some of them performed many more repetitions. The dataset contains seven types of head gestures: clockwise rotation, counterclockwise rotation, left horizontal motion, right horizontal motion, head nodding, m-shaped move to left and m-shaped move to right. The “idealized” shape of motions are visualized in Figure 1. We have chosen those templates basing on reports from other papers, where authors have found them to be natural, especially for commands in design gestures-based controlling interface [6,7,50]. We have chosen this set of gestures for several reasons. All of them can be easily explained to and repeated by system users/experiment participants. Also, although there is a visible difference between each gesture, they have some important similarities, for example. circular trajectory is present both in clockwise and m-shaped actions. Also, actions have to be considered as the time-varying signals, because the spatial patterns that are drawn in space between some classes are undistinguishable when temporal information is not used. This situation happens, for example, for clockwise and counterclockwise, left and right, m left and m right actions pairs. The motion range of the head which is defined as a set of all possible normalized vectors coordinates that indicate head orientation has a semi hemispherical shape (see Figure 1, left). It is of course not possible to perform head motions that are 360 degrees rotation in axial (horizontal) plane; however, it is possible to make circular or m-shaped trajectories on the hemisphere of the motion range. The dataset we used is summarized in Table 1. We have acquired 975 motion recordings. MoCap data acquisition was performed every 22.06 +/− 5.83 ms. As can be seen there is large standard deviation in acquisition time that happens in each recording. That is because of the fact that a data gathering application does not have any mechanism that assures constant acquisition frequency. We have collected head gesture data using the hardware system based on smartphone Samsung Galaxy S7 (SGS7) and virtual reality headset equipment namely Google Cardboard. The phone has been powered by the Exynos 8890 with a Quad-Core 2.3 GHz Mongoose + Quad-Core 1.6 GHz Cortex A53 configuration and with memory LPDDR4 RAM of 4GB in size. The device operated under Android OS in version 8.0.

The phone of this model is equipped with an LSM6DS3 sensor manufactured by ST MICROELECTRONICS [51] with the full IMU functionality of 6-axis accelerometer and gyroscope. From technical point of view the LSM6DS3 sensor is a system-in-package featuring a 3D digital accelerometer and a 3D digital gyroscope performing at 1.25 mA (up to 1.6 kHz ODR) and able to work in different modes with linear acceleration measurement range of ±2 g/ ± 4 g/ ± 8 g/ ± 16 g and angular rate measurement range of ±125 dps/ ± 245 dps/ ± 500 dps/ ± 1000 dps/ ± 2000 dps appropriately. The data was acquired using an Android OS JAVA application developed by us, which visualizes the head motion trajectory of the user wearing VHR in the form of colored balls that were displayed in a virtual three-dimensional room, see Figure 2, for example. This additional visual feedback helped experiment participants to visualize themselves and which head motion they performed. As can be seen in Figure 3, each experiment participant might perform the same action in many ways: although motions are similar to idealized trajectory the obtained spatial representations

of motions are very diverse. Figure 4 presents real head motion trajectories obtained by one of the users. Compare this with Figure 1.

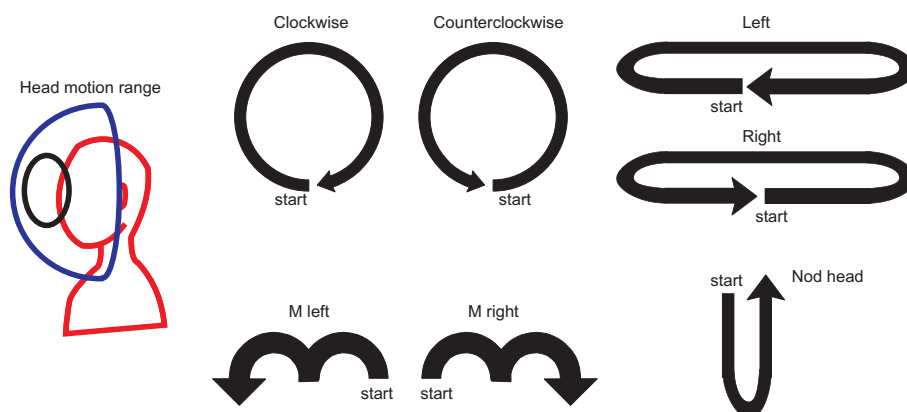


Figure 1. The left side of the figure presents schematic image of head motion range (a blue hemisphere) with example circular head gesture trajectory (black line). The right side of the figure presents “idealized” shape of motions in the evaluation dataset.

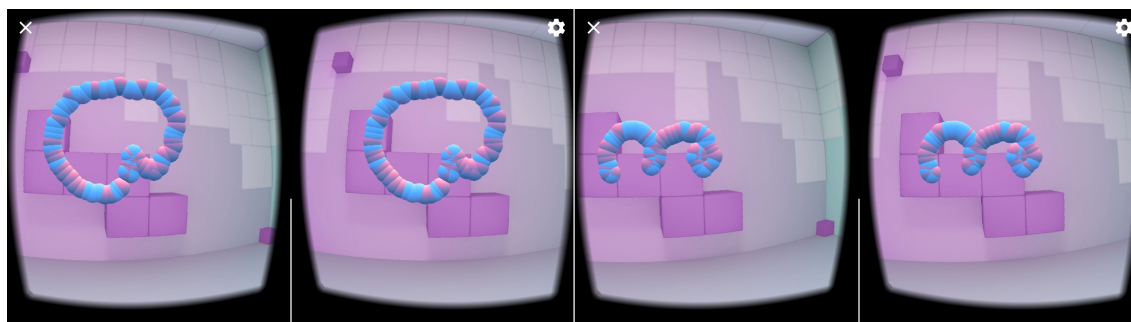


Figure 2. This figure presents stereovision view from the virtual reality helmet (VRH) system. The trajectory on the left is counterclockwise motion while the trajectory on the right is m right gesture.

Table 1. This table summarizes data in evaluation dataset. Rows represents persons while columns represents motion classes. MIN and MAX are minimal and maximal number of samples for each motion class that are present in the recordings. For example, the shortest clockwise recording has 85 samples while the longest has 332 samples.

	Clockwise	Counterclockwise	Left	Nod Head	M Left	M Right	Right
1	10	11	11	12	12	12	12
2	12	13	11	13	11	11	11
3	14	11	11	11	11	11	11
4	10	10	11	11	13	13	11
5	12	11	12	23	11	11	11
6	11	11	12	11	12	11	11
7	11	12	10	12	11	10	10
8	11	12	11	13	11	12	11
9	11	12	11	12	12	12	11
10	12	11	11	22	12	12	11
11	11	12	11	11	11	11	11
12	14	11	11	10	10	11	10
SUM	139	137	133	161	137	137	131
MIN	85	86	81	50	75	76	81
MAX	332	412	528	183	255	264	406

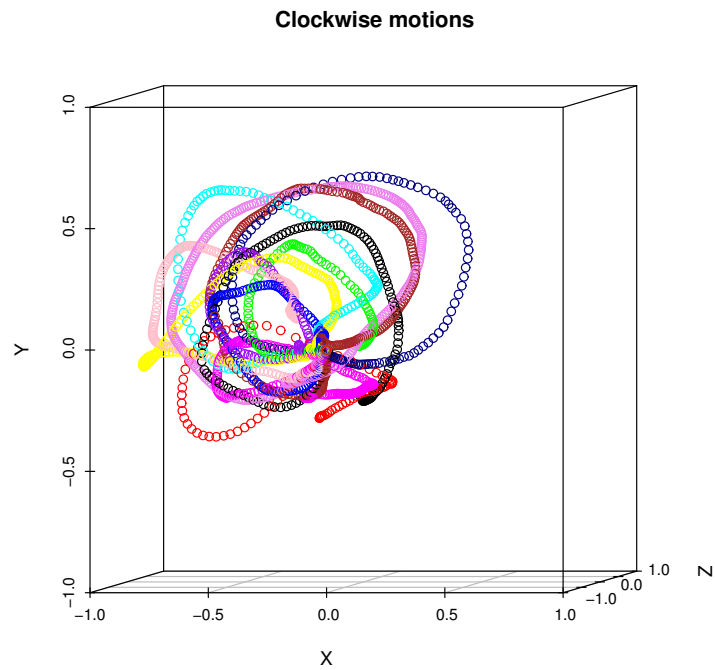


Figure 3. This figure presents twelve examples of a clockwise motion performed by each person. A trajectory that belongs to the single person is color-coded.

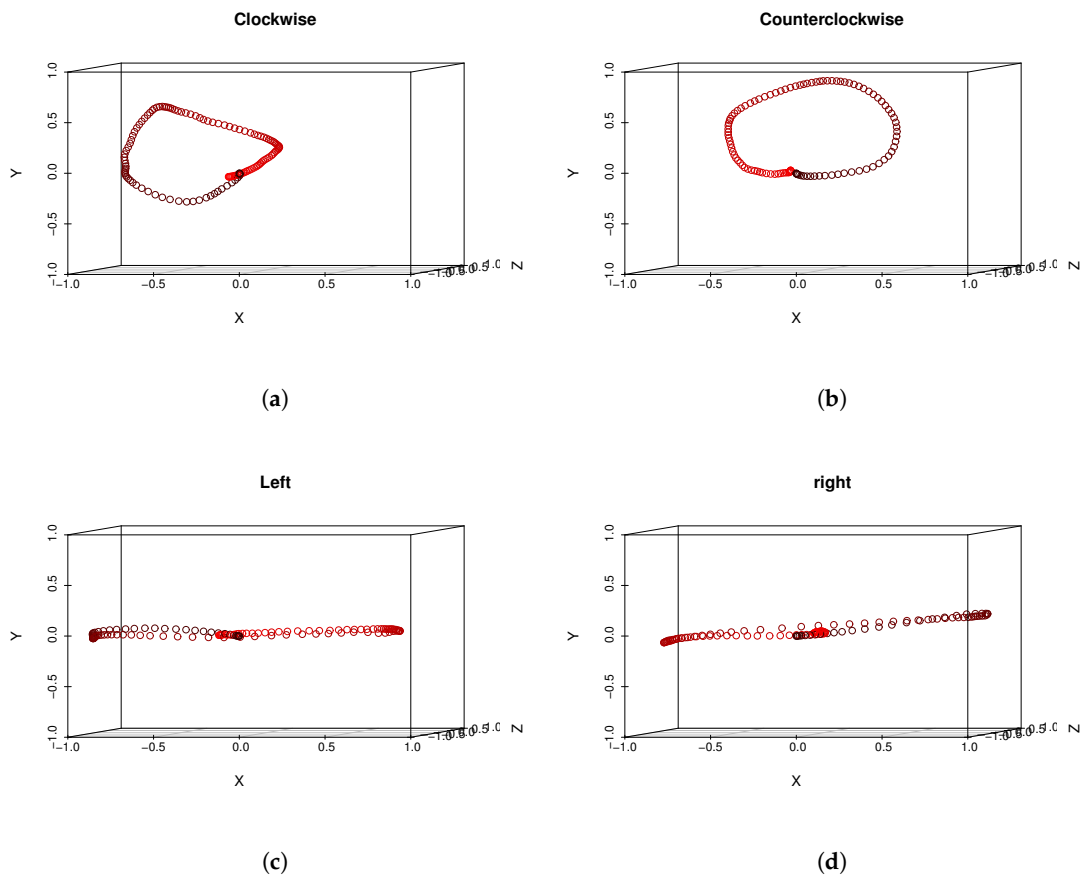


Figure 4. Cont.

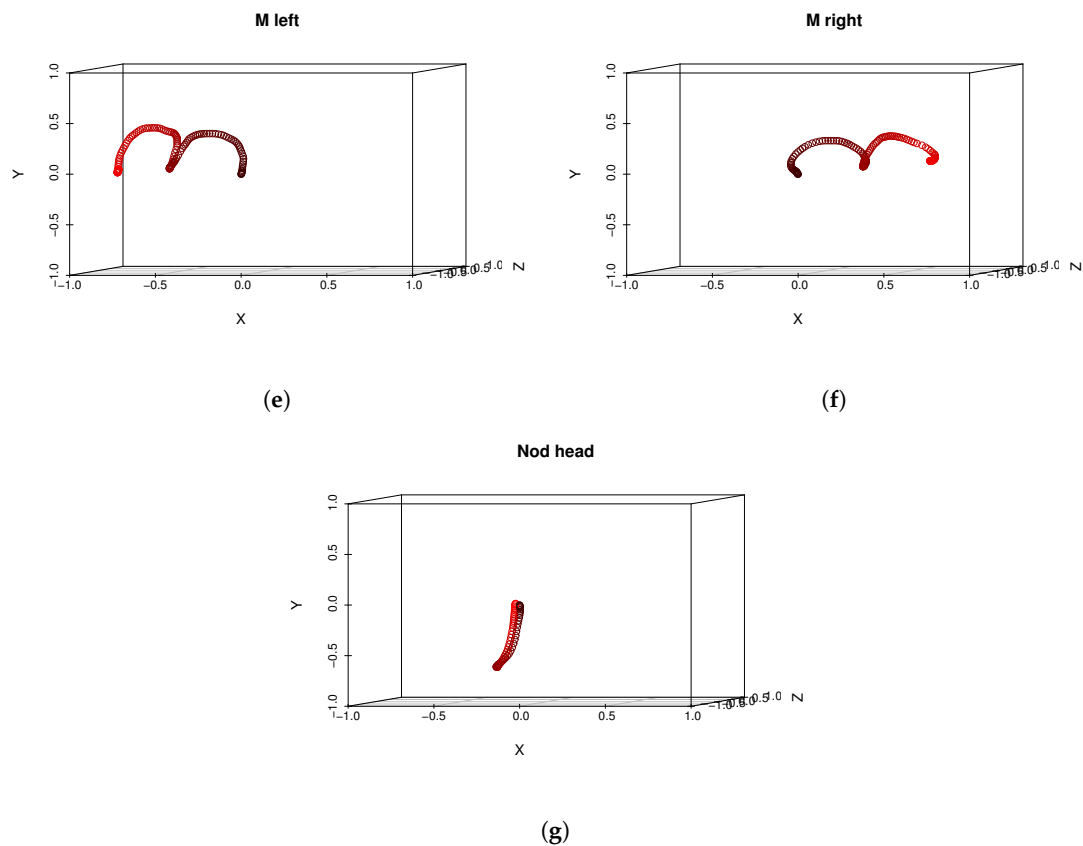


Figure 4. This figure presents real head motion trajectories obtained by one of the users – compare with Figure 1. Sampled head positions are color-coded. Early samples are darker while end samples of motions are red.

The majority of recordings that were acquired by our application contained multiple actions performed one after another by experiment participants. In order to segment motions recordings into samples that contain only one example of a certain gesture (as it is present in Figure 4 and summarized in Table 1) we have used a heuristic segmentation algorithm presented in pseudocode Algorithm 1. It can be seen that the algorithm is dependent on three adaptive parameters: median filter window size, angle difference threshold (in degrees) and minimal length threshold. Values of those samples might differ depending on the MoCap system acquisition frequency and in our case we used values: $mk = 15$, $t_1 = 1$, $t_2 = 10$. After applying our algorithm to those values, all motions samples were correctly segmented from our MoCap data. After segmentation head gestures were manually labeled in order to assign them to correct classes. Example segmentation results are presented in Figure 5. We decided to use our own segmentation method because it have returned anticipated results; however, one can use any method proposed in the literature, for example, [34]; however, some tuning of those methods (for example, adaptation to acquisition frequency) might be unavoidable. Those issues, however, are beyond the scope of this paper.

Algorithm 1: Head gestures segmentation from quaternion signal

Data: Q - quaternion signal, mk - median filter window size, t_1 - angle difference threshold, t_2 - minimal length threshold.

Result: R - a set of indexes pairs that indicates beginning and the end of head gestures.

```

for a in 1:(length(Q)-1) do
  | D[a]  $\leftarrow \angle(Q[a], Q[a + 1])$ 
end
F  $\leftarrow$  MedianFilter(D, mk)
F[F <  $t_1$ ]  $\leftarrow$  0
F[F >  $t_1$ ]  $\leftarrow$  1
s  $\leftarrow$  0
a  $\leftarrow$  1
R  $\leftarrow$  EMPTY.SET
while a < length(F) and s = 0 do
  | if F[a] = 1 then
  | | s = a
  | end
  | a  $\leftarrow$  a + 1
end
if s  $\neq$  0 and s < length(F) then
  | for a in (s + 1):length(F) do
  | | if F[a] = 0 and s = 0 then
  | | | else
  | | | | if F[s] = 1 and s = 0 then
  | | | | | s  $\leftarrow$  a
  | | | | | else
  | | | | | | if F[s] = 1 and F[a] = 0 then
  | | | | | | | if (a-1) - s >  $t_2$  then
  | | | | | | | | append(R, [s, a-1])
  | | | | | | | | s  $\leftarrow$  0
  | | | | | | | end
  | | | | | | end
  | | | | | end
  | | | | end
  | | | end
  | | end
  | end
end

```

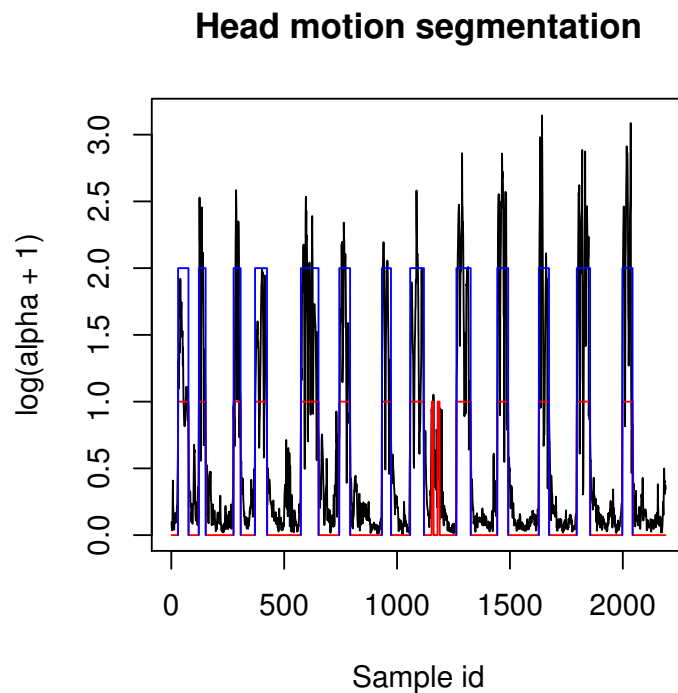


Figure 5. This figure presents example segmentation results of Algorithm 1. Black lines are angles between two neighboring quaternions (motion velocity), red segments are values above t_1 threshold of median-filtered velocity signal, blue segments are values above t_2 threshold. Those blue segments are segmentation results returned in list R.

Each recording has been also rotated so that the initial rotation Euler angle has become $[0, 0, 0]$ (that means $[0, 0, 0, 1]$ for quaternion). In order to do so we have generated new rotation signal Q_r from input rotation signal Q by multiplying the inverse quaternion of the first acquired frame $Q[1]^{-1}$ by each acquired frame $Q[a]$. After this operation, first the rotation equals $[0, 0, 0, 1]$ and then the rest of them are transformed by the inverse of the initial rotation (see Equation (5)).

$$Q_r[a] = Q[1]^{-1} \cdot Q[a] \quad (5)$$

In next sections we will describe the motion classification procedures we have tested in this research. The dataset preprocessed by methods described in this section is used by all the following algorithms.

2.2. Head Gestures Recognition with DTW Classifier

The first classification method we used is a DTW consisting of seven templates generated with modified DBA algorithm originally proposed in paper [40]. That method is a modification of DBA from [52]; however, it is adapted to operate on a quaternion signal. The original barycenter averaging method has been replaced by norm-preserving Markley's algorithm [53] where the averaged quaternion is found by solving the following eigenvalue problem (see Algorithm 2).

Algorithm 2: Markley's quaternion averaging

Data: Q - quaternion signal, which values we want to average, W - non-negative weights vector.

Result: Matrix A, which eigenvector corresponding to the largest eigenvalue is averaged quaternion of all quaternion in Q.

$$A \leftarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

sum \leftarrow 0

for a in 1:length(Q) **do**

$$\begin{array}{l} q_a \leftarrow Q[a] \\ w_a \leftarrow W[a] \\ A \leftarrow w_a \cdot q_a * q_a^T + A \\ \text{sum} \leftarrow \text{sum} + w_a \end{array}$$

end

$$A \leftarrow \frac{1}{\text{sum}} \cdot A$$

where * is matrix (not quaternion!) multiplication operator.

The DTW cost function is defined as [20]:

$$\text{cost}(q_1, q_2) = 1 - |q_1 \circ q_2| \quad (6)$$

Where $|q_1 \circ q_2|$ is an absolute value of a dot (scalar) product of quaternions q_1 and q_2 .

The DTW distance is calculated as normalized DTW distance.

Our DTW classifier operates on spatial representation of head motion which is calculated using Equation (7). We can express head position in three-dimensional space as a unit vector $[0, 0, 1]$ transformed by a quaternion that represents head rotation. Each head motion is represented as a three-dimensional time varying signal of spatial coordinates that denote normalized vectors that indicate head orientation. The coordinates of that vector drawn in three-dimensional space nicely represent head motion trajectory.

$$\begin{aligned} q_1^* &= [-x_1, -y_1, -z_1, w_1] \\ q_v &= [0, 0, 1, 0] \\ q_r &= (q^* \cdot q_v) \cdot q_1 \\ v_1 &= [x_r, y_r, z_r] \end{aligned} \quad (7)$$

where q_1 is a rotation quaternion and v_1 is a return vector from that calculation.

The DTW classifier utilizes Euclidean distance and assigns input motion to the class for which the normalized distance between input signal and template signal that represents the motion class (also recalculated using Equation (7)) has the minimal value.

2.3. Bagged DTW Classifier

The approach presented in Section 2.2 does not exploit the information from the training dataset about which the user performed a certain action. That is because, in the method presented in Section 2.2, each DTW template is generated from all MoCap recordings from training dataset. In order to apply a well-known bagging schema [54] to generate classifier from our dataset we can construct a set of "weak classifiers" where each of them is trained on data that come from the single user. This approach will generate as many DTW classifiers trained on motion classes as we have different experiment participants in our training dataset. Thanks to this approach, user-specific motions trajectories will not be averaged over all other trajectories. In the motion classification process, the set of "weak classifiers"

performs voting and assigns input motion to the class to which motion was assigned by the majority of “weak classifiers”.

2.4. Head Gestures Recognition with PCA-Based Features

As we already mentioned in Section 1.2 PCA-based features are among most popular approaches used in MoCap data analysis and classification. That approach is described for example in paper [55] so we will summarize only the basic concept.

2.4.1. Features Generation (Training)

At first, input data is processed in the same manner as in Section 2.1, also Equation (7) is used to recalculate quaternion representation to obtain spatial representation of head motion. Then we have to resample training data to the uniform size: each recording should have a common length. Also, all recordings that we would like to classify in the future will have to be resampled to that length. In the next step we perform the operation analogical to “flattening” known from deep neural networks. In this step we generate a single-dimensional vector from n -column input signal by putting values from each column one after another in a single vector. Suppose we have m rows and n columns. The flattened vector will have $n \cdot m$ rows and 1 column. Then we compose a matrix A , which columns are “flattened” vectors of training dataset. We create a mean vector $V^{[n \cdot m]}$ (in square brackets we present dimensionality of a vector, this is a column vector) whose values are calculated as averaged values of each row of matrix A . We generate a new matrix A' by subtracting $V^{[n \cdot m]}$ from each column of A . In the next step we calculate covariance matrix C from matrix A' and we find eigenvalues and eigenvectors of C . We order eigenvalues by descending order and we take k eigenvectors that correspond with most significant eigenvalues. The PCA-based features that are used to describe a MoCap recording are coefficients of a linear combination of those most significant eigenvectors (or simply: vector coordinates) that are obtained after performing projection of input $n \cdot m$ -dimensional MoCap on k -dimensional space generated by Principal Components Analysis. That projection of an input signal $S^{[n \cdot m]}$ can be computed using the mean vector and a matrix composed of k eigenvectors corresponding to k most significant eigenvalues:

$$S^{[k]} = X^{[k, n \cdot m]} \cdot (S^{[n \cdot m]} - V^{[n \cdot m]}) \quad (8)$$

where $X^{[k, n \cdot m]}$ is a matrix which rows are k eigenvectors corresponding to the k most significant eigenvalues.

We use the training dataset recalculated to be k -dimensional to train any desired classifier. In this paper we will utilize a nearest neighbor classifier.

2.4.2. Classification

The classification step requires resampling of the recording we want to classify to the same length as it was done for a training dataset, flattening the input signal that represents motion and computing projection of obtained $n \cdot m$ vector to k -dimensional space using coefficients calculated in feature generation (training) step (8). Then we classify the input recording using appropriate classifier (as we already mentioned in our case it is a nearest neighbor classifier). In our implementation we have used unoptimized NN algorithm, where we compare input vector to each other vector in the dataset to find the nearest neighbor. We are aware of the existence of searching algorithms that accelerates computation like kd-trees [56] Semi-Convex Hull Tree [57] or Dynamic Continuous Indexin [58]; however, they will not improve the recognition rate of the approach, which was a goal of our research.

2.5. Two-Stage PCA-Based Method

In some approaches, the authors perform an additional initial projection of input data to lower dimensional space with PCA before calculating PCA-based features. For example, in [39], the authors

project whole dataset to two-dimensional space with PCA, then they calculate PCA-based features using approach presented in Section 1. The classification is performed using SVM classifier. Input data for the algorithm is processed in the same manner as in Section 2.1, also Equation (7) is used to recalculate quaternion representation to obtain spatial representation of head motion.

2.6. Neural Network and Random Forest

Similarly to [18] we have applied ANN (a feedforward artificial neural network) and DT-based approaches (namely RF) to perform classification of our data. Dimensionality reduction is a commonly-used procedure that is applied on a high-dimensional dataset or signals with a large number of samples. However, linear projection might be redundant while the classification process is done with a classifier capable of partitioning the features space with appropriate decision hyperplanes. Also, in our case it is not necessary to apply convolution layers to calculate input features from time series, because the coordinates of vectors that represent head motion trajectory (7) are enough to train a fully connected network or a decision tree. In our case the only requirement for the training dataset is to perform resampling to the uniform size and “flattening”, as described in Section 2.4.1. That resampled and “flattened” signal is used as an input data for NN and RF algorithms. The NN activation function was hyperbolic tangent [18] and the network was trained using a parallelized version of stochastic gradient descent using back-propagation [59]. We have implemented these machine learning methods using the H2O library (<https://github.com/h2oai/h2o-3>).

3. Results

We have evaluated the methods presented in Sections 2.2–2.5 on the dataset from Section 2.1. All classifiers have been evaluated using leave-one-out cross validation method, where recordings from the single person have been used for testing while recordings from eleven persons have been used for classifier training. In case of the NN classifier with PCA-based features from Section 2.4, we have evaluated various lengths of features vectors: with 5, 10, 15, 20, 25, 30, 35 and 40 coefficients. Because highest recognition for that method was obtained for a 15-dimensional vector, we have used the same length of vector in the second step of Two-stage PCA-based method from Section 2.5. Several ANN architectures have been tested with a different number of neurons in the hidden layer. We have used a hidden layer with a single row of 50, 100 or 200 neurons and a hidden layer with two, fully connected rows of neurons, each of them with size 50, 100 or 200. Random forest contained 50 decision trees with maximal depth 20. In the case of Bagged DTW classifiers, they contained eleven “weak classifiers”, each of them trained on data from a different person. In Table 2 we present the error rate of each examined classifier and total error rate of each classifier is presented. The recognition rate is defined as number of correctly classified objects of that class divided by all objects of that class, the error rate equals $1 - \text{recognition rate}$. Total recognition rate is defined as number of correctly classified objects of each class divided by all objects of all classes. Total error rate is $1 - \text{total recognition rate}$. Figure 6 visualizes the results from Table 2 in the form of a bar plot. The highest value of recognition rate has been obtained for the DTW bagged classifier. It equals 0.975 and is higher than the second-best classifier, namely DTW, by 0.013. DTW bagged error rate is the smallest or it shares smallest position in case of clockwise, counterclockwise, left and right head gestures. The multiclass confusion matrix for the DTW bagged classifier is presented in Table 3. Table 4 presents the same confusion matrix, however each value is calculated relatively to the count of head gestures of each type. Table 4 contains an additional column that contains the classification error for each motion class.

Table 2. This table presents error rate of each examined classifier and total error rate of each classifier.

	Clockwise	Counterclockwise	Left	Nod Head	M Left	M Right	Right	Error
Eigen 5	0.148	0.074	0.090	0.035	0.054	0.020	0.147	0.080
Eigen 10	0.101	0.074	0.030	0.075	0.007	0.027	0.049	0.052
Eigen 15	0.054	0.095	0.028	0.087	0.020	0.020	0.049	0.051
Eigen 20	0.074	0.115	0.014	0.110	0.020	0.074	0.021	0.063
Eigen 25	0.067	0.088	0.028	0.104	0.013	0.047	0.042	0.057
Eigen 30	0.081	0.095	0.042	0.098	0.034	0.081	0.049	0.069
Eigen 35	0.074	0.115	0.042	0.098	0.067	0.081	0.056	0.077
Eigen 40	0.087	0.122	0.042	0.110	0.094	0.074	0.042	0.082
Two-Stage PCA	0.134	0.061	0.069	0.046	0.034	0.040	0.014	0.057
DTW	0.087	0.014	0.000	0.029	0.054	0.067	0.014	0.038
DTW bagged	0.020	0.007	0.000	0.040	0.027	0.060	0.014	0.025
NN 50	0.087	0.115	0.035	0.150	0.013	0.013	0.112	0.077
NN 100	0.094	0.088	0.028	0.127	0.027	0.013	0.084	0.067
NN 200	0.087	0.115	0.035	0.069	0.020	0.007	0.091	0.061
NN 50x50	0.081	0.122	0.028	0.087	0.020	0.020	0.063	0.061
NN 100x100	0.114	0.095	0.028	0.150	0.020	0.007	0.112	0.077
NN 200x200	0.107	0.101	0.035	0.168	0.027	0.007	0.084	0.078
RF	0.081	0.101	0.021	0.104	0.027	0.007	0.077	0.061

Table 3. This table presents the multiclass confusion matrix for DTW bagged classifier.

	Clockwise	Counterclockwise	Left	Nod Head	M Left	M Right	Right
clockwise	146	0	0	2	0	0	1
counterclockwise	0	147	1	0	0	0	0
left	0	0	144	0	0	0	0
nod head	0	7	0	166	0	0	0
m left	0	4	0	0	145	0	0
m right	5	0	0	4	0	140	0
right	0	0	2	0	0	0	141

Table 4. This table presents the same confusion matrix as in Table 3; however, each value is calculated relatively to the count of head gestures of each type.

	Clockwise	Counterclockwise	Left	Nod Head	M Left	M Right	Right	Error
clockwise	0.980	0.000	0.000	0.013	0.000	0.000	0.007	0.020
counterclockwise	0.000	0.993	0.007	0.000	0.000	0.000	0.000	0.007
left	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000
nod head	0.000	0.040	0.000	0.960	0.000	0.000	0.000	0.040
m left	0.000	0.027	0.000	0.000	0.973	0.000	0.000	0.027
m right	0.034	0.000	0.000	0.026	0.000	0.940	0.000	0.060
right	0.000	0.000	0.014	0.000	0.000	0.000	0.986	0.014

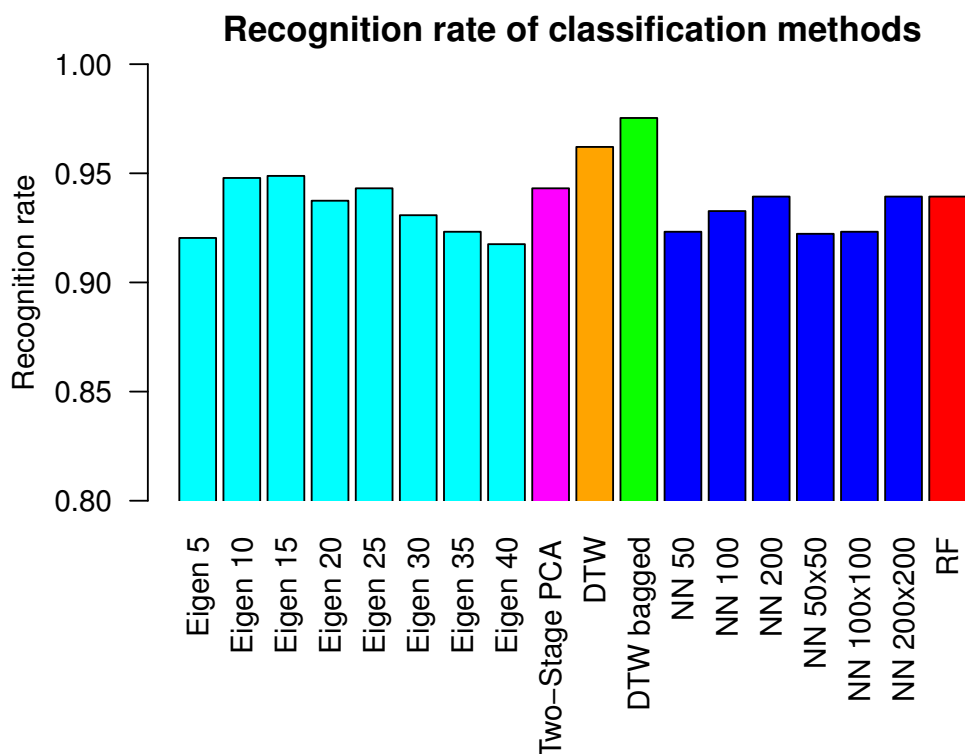


Figure 6. This figure visualizes results from Table 2 in the form of bar plot.

4. Discussion

As can be seen in the previous section, all methods that were applied for head gestures classifications have obtained very good results on our test dataset. Each classifier that was applied for this task obtained total recognition rate above 91%. Figure 3 visualizes why the classification was not an easy task: motions of each person varied in execution speed, range of head rotation and spatial head position. The reason for such a high recognition rate is the fact that, for this evaluation, we have chosen pattern recognition methods that have either already proven their usefulness in MoCap data identification, or we have anticipated them to be very promising. Error rates obtained by various architectures of ANN and a RF are on a similar level as error rates of the NN algorithm with 25–35 PCA features. This might be explained by the fact that ANN and RF utilize decision hyperplanes that might substitute dimensionality reduction performed by PCA. We have observed the lowest error rates for ANN with a single hidden layer of 100 neurons and two rows of 50 neurons each. In case of networks that contained less or more neurons or layers, results were worse. That might be caused by classifiers' underfitting or overfitting appropriately. Two methods that utilized DTW with template signals averaged by algorithm [40] have a higher recognition rate than performing methods that utilized various numbers of PCA-based features. The most probable reason for this is the fact that PCA-based features deal very well with MoCap data that contain signals from several body joints because they have the ability to linearly map the initial signal space to lower-dimensional space that maximizes the variance between recordings. In the case of motions gathered from the single sensors, however, it is possible to obtain higher recognition rate using direct matching of motions trajectories to averaged signal using DTW. Also, the DTW method does not require input signals resampling to the uniform length. The bagged version of the DTW classifier has slightly higher recognition rate than the non-bagged version. The bagged classifier has higher computational complexity than non-bagged; however, classification in each "weak" classifier is performed independently and can be paralleled without much influence on overall classification time. The evaluation proved that in the case of all classifiers, there were some recognition errors. Because of the leave-one-out procedure, the training

dataset did not contain data acquired from a person on whose data validation was done. In all validations, there was no event when all test data was misclassified; however, there were situations when not all recordings in the validation dataset were assigned to the correct class. That means that all applied classifiers were biased due to variability, which happens during several repetitions of same action by people in the dataset.

5. Conclusions

In this paper, we propose and evaluate several methods capable of classifying head gestures acquired by VRH powered by of-the-shelf smartphone hardware. We have tested all methods on a large dataset of head motions for which trajectories were recommended as the benchmark motions for gesture-based interfaces. The results we have obtained proved that the bagged DTW classifier trained with modified DBA algorithm that averages quaternion-based signal is the most promising algorithm for that task. DTW also has an important advantage over methods that utilized PCA-based features because DTW does not require input signal resampling to the uniform length. Due to this fact, DTW can directly compare the input signal to the templates after segmentation and processing with (5). As we already mentioned in the first section of this paper the most important application of our method is head motion-based interface for disabled people. The results we have obtained are not only limited to head gestures acquired with VRH with smartphone hardware but can be generalized to any MoCap method that allow the measuring of head motion. In future work, we plan to extend our research on eye tracking technology and eye motions and gaze-based interfaces. This is also a very challenging subject because eyes perform both voluntary and involuntary movements and we anticipate that the main obstacle to overcome is to distinguish one type of these motions from another.

Author Contributions: T.H. was responsible for conceptualization, proposed methodology, software implementation and writing of original draft; M.P. was responsible for data curation, validation and co-writing of original draft.

Funding: This work has been supported by the National Science Centre, Poland, under project number 2015/17/D/ST6/04051.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Kim, M.; Choi, S.H.; Park, K.B.; Lee, J.Y. User Interactions for Augmented Reality Smart Glasses: A Comparative Evaluation of Visual Contexts and Interaction Gestures. *Appl. Sci.* **2019**, *9*, 3171. [[CrossRef](#)]
2. Kangas, J.; Rantala, J.; Akkil, D.; Isokoski, P.; Majaranta, P.; Raisamo, R. Vibrotactile Stimulation of the Head Enables Faster Gaze Gestures. *Int. J. Hum. Comput. Stud.* **2017**, *98*, 62–71. [[CrossRef](#)]
3. Morales, J.; Akopian, D. Physical activity recognition by smartphones, a survey. *Biocybernetics Biomed. Eng.* **2017**, *37*, 388–400. [[CrossRef](#)]
4. Farooq, M.; Sazonov, E. Accelerometer-Based Detection of Food Intake in Free-Living Individuals. *IEEE Sens. J.* **2018**, *18*, 3752–3758. [[CrossRef](#)]
5. Ahuja, K.; Islam, R.; Parashar, V.; Dey, K.; Harrison, C.; Goel, M. EyeSpyVR: Interactive Eye Sensing Using Off-the-Shelf, Smartphone-Based VR Headsets. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2018**, *2*, 57. [[CrossRef](#)]
6. Mavuş, U.; Sezer, V. Head gesture recognition via dynamic time warping and threshold optimization. In Proceedings of the 2017 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA), Savannah, GA, USA, 27–31 March 2017; pp. 1–7. [[CrossRef](#)]
7. Yi, S.; Qin, Z.; Novak, E.; Yin, Y.; Li, Q. GlassGesture: Exploring head gesture interface of smart glasses. In Proceedings of the IEEE INFOCOM 2016—The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9. [[CrossRef](#)]

8. Kelly, D.; Delannoy, D.; McDonald, J.; Markham, C. Automatic recognition of head movement gestures in sign language sentences. In Proceedings of the 4th China-Ireland Information and Communications Technologies Conference, Maynooth, Ireland, 19–21 August 2009; pp. 142–145.
9. Morimoto, C.; Yacoub, Y.; Davis, L. Recognition of head gestures using hidden Markov models. In Proceedings of the 13th International Conference on Pattern Recognition, Vienna, Austria, 25–29 August 1996; Volume 3, pp. 461–465. [[CrossRef](#)]
10. Hasna, O.L.; Potolea, R. Time series—A taxonomy based survey. In Proceedings of the 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 7–9 September 2017; pp. 231–238. [[CrossRef](#)]
11. Shokoohi-Yekta, M.; Hu, B.; Jin, H.; Wang, J.; Keogh, E. Generalizing DTW to the Multi-dimensional Case Requires an Adaptive Approach. *Data Min. Knowl. Discov.* **2017**, *31*, 1–31. [[CrossRef](#)]
12. Xue, Y.; Ju, Z.; Xiang, K.; Chen, J.; Liu, H. Multimodal Human Hand Motion Sensing and Analysis—A Review. *IEEE Trans. Cognitive Dev. Syst.* **2019**, *11*, 162–175. [[CrossRef](#)]
13. Cheng, H.; Yang, L.; Liu, Z. Survey on 3D Hand Gesture Recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 1659–1673. [[CrossRef](#)]
14. Dalmazzo, D.; Ramírez, R. Bowing Gestures Classification in Violin Performance: A Machine Learning Approach. *Front. Psychol.* **2019**, *10*, 344. [[CrossRef](#)]
15. Parnandi, A.; Uddin, J.; Nilsen, D.M.; Schambra, H.M. The Pragmatic Classification of Upper Extremity Motion in Neurological Patients: A Primer. *Front. Neurol.* **2019**, *10*, 996. [[CrossRef](#)]
16. Huang, J.; Zhou, W.; Li, H.; Li, W. Attention-Based 3D-CNNs for Large-Vocabulary Sign Language Recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 2822–2832. [[CrossRef](#)]
17. Liu, J.; Li, Y.; Song, S.; Xing, J.; Lan, C.; Zeng, W. Multi-Modality Multi-Task Recurrent Neural Network for Online Action Detection. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *29*, 2667–2682. [[CrossRef](#)]
18. ur Rehman, M.Z.; Waris, M.; Gilani, S.; Jochumsen, M.; Niazi, I.; Jamil, M.; Farina, D.; Kamavuako, E. Multiday EMG-Based Classification of Hand Motions with Deep Learning Techniques. *Sensors* **2018**, *18*, 2497. [[CrossRef](#)] [[PubMed](#)]
19. Zhao, H.Y.; Wang, Z.; Qiu, S.; Xu, F.; Wang, Z.; Shen, Y. Adaptive gait detection based on foot-mounted inertial sensors and multi-sensor fusion. *Inf. Fusion* **2019**, *52*. [[CrossRef](#)]
20. Switonski, A.; Josinski, H.; Wojciechowski, K. Dynamic time warping in classification and selection of motion capture data. *Multidimension. Syst. Signal Process.* **2019**, *30*, 1437–1468. [[CrossRef](#)]
21. A Survey on Gait Recognition. *ACM Comput. Surv.* **2018**, *51*, 1–35. [[CrossRef](#)]
22. Berman, S.; Stern, H. Sensors for Gesture Recognition Systems. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2012**, *42*, 277–290. [[CrossRef](#)]
23. Hachaj, T.; Ogiela, M.R. Classification of Karate Kicks with Hidden Markov Models Classifier and Angle-Based Features. In Proceedings of the 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Beijing, China, 13–15 October 2018; pp. 1–5. [[CrossRef](#)]
24. Billon, R.; Nédélec, A.; Tisseau, J. Gesture Recognition in Flow Based on PCA and Using Multiagent System. In Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology, Bordeaux, France, 27–29 October 2008; pp. 239–240. [[CrossRef](#)]
25. Bottino, A.; Simone, M.D.; Laurentini, A. Recognizing Human Motion using Eigensequences. *J. WSCG* **2007**, *15*, 135–142.
26. Świtoński, A.; Mucha, R.; Danowski, D.; Mucha, M.; Polanski, A.; Cieslar, G.; Wojciechowski, K.; Sieron, A. Diagnosis of the motion pathologies based on a reduced kinematical data of a gait. *Przegląd Elektrotechniczny* **2011**, *87*, 173–176.
27. Mantovani, G.; Ravaschio, A.; Piaggi, P.; Landi, A. Fine classification of complex motion pattern in fencing. *Procedia Eng.* **2010**, *2*, 3423–3428. [[CrossRef](#)]
28. Choi, W.; Ono, T.; Hachimura, K. Body Motion Analysis for Similarity Retrieval of Motion Data and Its Evaluation. In Proceedings of the 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kyoto, Japan, 12–14 September 2009; pp. 1177–1180. [[CrossRef](#)]
29. Skurowski, P.; Pruszowski, P.; Pęszor, D. Synchronization of Motion Sequences from Different Sources. *AIP Conf. Proc.* **2016**, *1738*, 180013. [[CrossRef](#)]

30. Hinkel-Lipsker, J.; Hahn, M. Coordinative structuring of gait kinematics during adaptation to variable and asymmetric split-belt treadmill walking – A principal component analysis approach. *Hum. Movement Sci.* **2018**, *59*. [[CrossRef](#)] [[PubMed](#)]
31. Yang, Y.; Zeng, L.; Leung, H. Keyframe Extraction from Motion Capture Data for Visualization. In Proceedings of the 2016 International Conference on Virtual Reality and Visualization (ICVRV), Hangzhou, China, 24–26 September 2016; pp. 154–157. [[CrossRef](#)]
32. Lee, M.; Roan, M.; Smith, B. An application of principal component analysis for lower body kinematics between loaded and unloaded walking. *J. Biomech.* **2009**, *42*, 2226–2230. [[CrossRef](#)] [[PubMed](#)]
33. Zago, M.; Pacifici, I.; Lovecchio, N.; Galli, M.; Federolf, P.; Sforza, C. Multi-segmental movement patterns reflect juggling complexity and skill level. *Hum. Movement Sci.* **2017**, *54*. [[CrossRef](#)]
34. Peng, S. Motion Segmentation Using Central Distance Features and Low-Pass Filter. In Proceedings of the 2010 International Conference on Computational Intelligence and Security, Nanning, China, 11–14 December 2010; pp. 223–226. [[CrossRef](#)]
35. Fotiadou, E.; Nikolaidis, N. Activity-based methods for person recognition in motion capture sequences. *Pattern Recognit. Lett.* **2014**, *49*, 48–54. [[CrossRef](#)]
36. Choi, W.; Li, L.; Sekiguchi, H.; Hachimura, K. Recognition of gait motion by using data mining. In Proceedings of the 2013 13th International Conference on Control, Automation and Systems (ICCAS 2013), Gwangju, Korea, 20–23 October 2013; pp. 1213–1216. [[CrossRef](#)]
37. Choi, W.; Sekiguchi, H.; Hachimura, K. Analysis of Gait Motion by Using Motion Capture in the Japanese Traditional Performing Arts. In Proceedings of the 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kyoto, Japan, 12–14 September 2009; pp. 1164–1167. [[CrossRef](#)]
38. Chalodhorn, R.; Rao, R.P.N. Learning to Imitate Human Actions through Eigenposes. In *From Motor Learning to Interaction Learning in Robots*; Sigaud, O., Peters, J. Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 357–381.
39. Das, S.R.; Wilson, R.C.; Lazarewicz, M.T.; Finkel, L.H. Two-Stage PCA Extracts Spatiotemporal Features for Gait Recognition. *J. Multimedia* **2006**, *1*, 9–17.
40. Hachaj, T.; Piekarczyk, M.; Ogiela, M.R. Human Actions Analysis: Templates Generation, Matching and Visualization Applied to Motion Capture of Highly-Skilled Karate Athletes. *Sensors* **2017**, *17*, 2590. [[CrossRef](#)]
41. Hachaj, T. Head Motion—Based Robot’s Controlling System Using Virtual Reality Glasses. In *Image Processing and Communications*; Choraś, M., Choraś, R.S., Eds.; Springer: Cham, Switzerland, 2020; pp. 6–13.
42. Field, M.; Pan, Z.; Stirling, D.; Naghdy, F. Human motion capture sensors and analysis in robotics. *Ind. Rob.* **2011**, *38*, 163–171. [[CrossRef](#)]
43. Kim, H.; Gabbard, J.L.; Anon, A.M.; Misu, T. Driver Behavior and Performance with Augmented Reality Pedestrian Collision Warning: An Outdoor User Study. *IEEE Trans. Visual Comput. Graphics* **2018**, *24*, 1515–1524. [[CrossRef](#)]
44. Li, G.; Chung, W. Combined EEG-Gyroscope-tDCS Brain Machine Interface System for Early Management of Driver Drowsiness. *IEEE Trans. Hum. Mach. Syst.* **2018**, *48*, 50–62. [[CrossRef](#)]
45. Bao, H.; Fang, W.; Guo, B.; Wang, P. Real-Time Eye-Interaction System Developed with Eye Tracking Glasses and Motion Capture. In *Advances in Human Factors in Wearable Technologies and Game Design*; Ahram, T., Falcão, C., Eds.; Springer: Cham, Switzerland, 2018; pp. 72–81.
46. Chui, K.T.; Alhalabi, W.; Liu, R.W. Head motion coefficient-based algorithm for distracted driving detection. *Data Technol. Appl.* **2019**, *53*, 171–188. [[CrossRef](#)]
47. Zhang, C.; Wu, X.; Zheng, X.; Yu, S. Driver Drowsiness Detection Using Multi-Channel Second Order Blind Identifications. *IEEE Access* **2019**, *7*, 11829–11843. [[CrossRef](#)]
48. Karatas, C.; Liu, L.; Gruteser, M.; Howard, R. Single-Sensor Motion and Orientation Tracking in a Moving Vehicle. In Proceedings of the 2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Hong Kong, China, 11–13 June 2018; pp. 1–9. [[CrossRef](#)]
49. Zhao, Y.; Görne, L.; Yuen, I.M.; Cao, D.; Sullman, M.; Auger, D.J.; Lv, C.; Wang, H.; Matthias, R.; Skrypchuk, L.; et al. An Orientation Sensor-Based Head Tracking System for Driver Behaviour Monitoring. *Sensors* **2017**, *17*, 2692. [[CrossRef](#)] [[PubMed](#)]
50. Kela, J.; Korpipää, P.; Mäntyjärvi, J.; Kallio, S.; Savino, G.; Jozzo, L.; Marca, S.D. Accelerometer-based gesture control for a design environment. *Pers. Ubiquitous Comput.* **2005**, *10*, 285–299. [[CrossRef](#)]

51. LSM6DS3. iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope. Available online: www.st.com/web/en/resource/technical/document/datasheet/DM00133076.pdf (accessed on 29 October 2019).
52. Petitjean, F.; Ketterlin, A.; Gançarski, P. A Global Averaging Method for Dynamic Time Warping, with Applications to Clustering. *Pattern Recogn.* **2011**, *44*, 678–693. [[CrossRef](#)]
53. Markley, L.; Cheng, Y.; Crassidis, J.; Oshman, Y. Averaging Quaternions. *J. Guidance Control Dyn.* **2007**, *30*, 1193–1196. [[CrossRef](#)]
54. Breiman, L. Bagging Predictors. *Mach. Learn.* **1996**, *24*, 123–140. [[CrossRef](#)]
55. Hachaj, T. Improving Human Motion Classification by Applying Bagging and Symmetry to PCA-Based Features. *Symmetry* **2019**, *11*, 1264. [[CrossRef](#)]
56. Liu, W.; Sun, J.; Li, W.; Hu, T.; Wang, P. Deep Learning on Point Clouds and Its Application: A Survey. *Sensors* **2019**, *19*, 4188. [[CrossRef](#)]
57. Chen, Y.; Zhou, L.; Bouguila, N.; Zhong, B.; Wu, F.; Lei, Z.; Du, J.; Li, H. Semi-Convex Hull Tree: Fast Nearest Neighbor Queries for Large Scale Data on GPUs. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17–20 November 2018; pp. 911–916. [[CrossRef](#)]
58. Li, K.; Malik, J. Fast K-nearest Neighbour Search via Dynamic Continuous Indexing. In Proceedings of the 33rd International Conference on International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 671–679.
59. Recht, B.; Ré, C.; Wright, S.J.; Niu, F. Hogwild: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent. In *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F., Weinberger, K.Q., Eds.; NIPS Proceedings: Granada, Spain, 2011; pp. 693–701.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).