RECOGNITION OF HUMAN-COMPUTER INTERACTION GESTURES ACQUIRED BY INERTIAL MOTION SENSORS WITH THE USE OF HIDDEN MARKOV MODELS

Aleksander Sawicki[1], Kristina Daunoravičienė[2], Julius Griskevicius[2]

[1] Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

[2] Faculty of Mechanics,Vilnius Gediminas Technical University, Lithuania

Abstract: The paper presents the algorithm of recognition of selected Human-Computer Interaction (HCI) gestures acquired by inertial motion sensors. The possibilities of using Hidden Markov Models as classifiers have been verified. The experiments investigated the possibility of using a methodology dedicated to the recognition of virtual reality (VR) game gestures to classify HCI gestures. The paper compares the accuracy of classification depending on the method of discretization of the forearm orientation signals. The evaluation of the accuracy of the classification was carried out with the use of 3-fold cross validation. The paper uses author's data corpus containing in total 720 time series acquired from 20 human subjects.

Keywords: HMM, Classification, HCI, IMU

1. Introduction

Gestures can be used to transmit messages through the body's significant, meaningful motions. Consequently, they may constitute a form of non-verbal communication. Gestures, with particular emphasis on hand movements, are considered to be one of the most important in our daily communication. They are considered the most promising in the field of Human-Computer Interaction [1]. Therefore, nowadays a significant number of studies concerning their recognition are carried out.

The scientific work can be divided into three groups in relation to the type of used signals. In the first one, the classification of the gesture is based on the orientation signal which represents information about the rotation of the sensor or the body.

Advances in Computer Science Research, vol. 15, pp. .., 2021. DOI: 10.24427/acsr-2021-vol15-0001

In the second case, the orientation is used simultaneously with other types of signals. In the last group, orientation information is ignored and recognition is performed with raw sensor values.

The first of these groups includes the approach presented in the publications [2,3]. In the first one, the authors recognized the gestures used in VR games based only on the forearm orientation. As a result of IMU data fusion filter operation, a quaternions describing rotations in 3D space were determined. The data in this form were transformed into three Euler angles and further processed. In publication [3] the authors used the hand orientation data recorded as a video. The presented approach uses information on hand tilt in a 2D system that was oriented in accordance with the camera lens. The developed gesture recognition algorithm used only one rotation angle.

The second group of approaches includes the systems described in [4,5]. In the first one, the authors used information about hand orientation as well as its speed and location. Here, the video recordings were also used as the input signals of the system. In another approach [5], the feature vector containing the quaternion was expanded by the pressure signal data. The described system used the signals from the IMU unit and a specialized glove equipped with the pressure sensors.

In the last group of papers, the authors completely abandoned the use of orientation signals in favour of other types of data. In [6] the authors used the signals acquired by an inertial unit such as acceleration and angular velocity. The paper does not use the data fusion algorithm, which would allow to determine the orientation. It should be noted that there are publicly available and open source software packages that allows such a process. The lack of using this procedure is a deliberate and conscious action. A very similar approach was also presented in [7].

It should be emphasized that in the literature there are many articles in which unprocessed signals are used [6,7]. The use of different types of signals [4,5], is not an individual approach either. At the moment, however, there are work no extending the method presented in [2]. There is no major work group on HMM applications to classify gestures on the basis of three-dimensional spatial orientation signals.

After reading the above mentioned paper, the two following questions arise: “Is the methodology presented in [2] adequate for recognizing another group of gestures?” and "If an equally divided division of all Euler’s angles ensures the best accuracy of classification?"could be asked. This paper presents an attempt to answer those questions. The article begins with a brief description of the available database with a specification of the devices used to acquire the motion signal. The following section describes the methodology derived from the literature and the author’s modification. Finally, the results and conclusions are presented.

2. Methodology

The paper contains results of gesture recognition with the use of orientation signals in the form of quaternion time series. The developed pattern recognition algorithm is divided into three main blocks: "Prepocessing", "Vectorization" and "Classification" (Fig. 1.). The forearm orientations in the quaternion time series form are delivered to

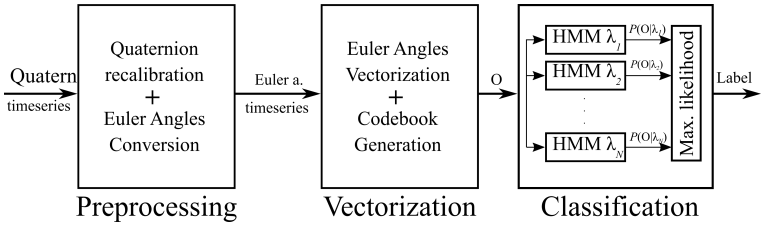

Fig. 1. Simplified structure of the gesture detection algorithm

the system input. In the first "Prepocessing" block, the quaternion series is normalized and converted to Euler angles series.The information in this form is vectorised and then used to generate a sequence of observations (O). In the "Classification" block, with the use of the Viterbi algorithm, a probability parameter (likelihood) is calculated for the four trained hidden Markov models. The result of the block is a single label describing the most probable gesture.

2.1 Data Set

The paper uses an authors collection of gestures that can be used in Human-Computer Interaction (HCI). The gestures described as "Come", "Turn Right", "Turn Over" and "Sit Down" were performed with the right hand. Motion dataset was inspired by the list of motions used in the renowned article [1]. The visualization of gestures is shown in Fig 2. The data corpus was created during the participation in the international "PROM" internship (financed by the Polish National Agency for Academic Exchange) in the Department of Biomechanics at the Gediminas University of Technology in Vilnius. The database consist of motion tracking sessions performed by 20 participants, including 8 men and 12 women. The average age of the participant was 26.15 years with a standard deviation of 6.44. There were no participants with illnesses or injuries that could affect the realization of particular gestures. The study participants performed 9 repetitions of individual gestures, which allowed to obtain

Fig. 2. Visualisation of the used gestures

a total of 720 movement sessions. Following manual segmentation, the single repetition time was of the order of second. Currently, discussions are taking place with the "PROM" coordinators in order to make the data available on the Zendodo or similar platform.

The acquisition of measurement data was carried out with the use of commercially available inertial motion tracking system called "Perception Neuron". This device consists of a set of 17 IMUs (Inertial Measurement Units). Each of the sensors provided the measurement of quantities as acceleration, angular velocity (magnetic field strength is proprietary), and due to the implemented algorithm of data fusion, orientation. The sensors were distributed evenly on the body, which allowed to track the entire skeleton. Fig. 3 A) presents the visualization of the person during the Turn Right gesture. As a result of preprocessing (Section 2.2), each of the registered persons was oriented according to the X axis of the coordinate system. In Fig. 3 B) forearm rotation axes was displayed by additional lines. It should be noted that the inertial system for the right forearm allows for the determination of 3 rotation angles around 3 axes.

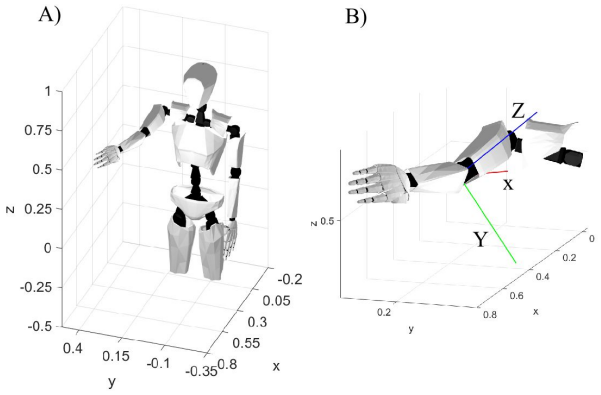

Fig. 3. A) Body visualisation in MATLAB software B) Forearm rotation axis visualisation

In the next part of the study, according to the literature [2], signal concerning the orientation of the forearm were used. The use of measurement data obtained from sensors located on the forearm is a popular solution which is applied in many scientific works [8]. IMU sensors are responsible for determining only their own orientation. The determination of skeletal limb orientation is done by calibration, in which the T-pose calibration gesture is performed by a participant. The perception neuron device enables calibration to be performed when all 17 sensors were detected. Therefore, at the moment of tracing one limb it was necessary to wear the complete suit. Moreover, the torso orientation was used as additional data in signal preprocessing, in elimination of Yaw/Azimuth offset angle approach.

Through signal processing, the orientation of the forearm in the form of a quaternion was converted to Euler's angles in Yaw-Pitch-Roll convention. This means that first the Z axis was rotated by an angle of Ψ, then the Y axis by an angle of θ and finally the X axis was rotated by an angle of Φ. It should be noted that the sensors of the Perception Neuron system are capable of conducting measurements at a frequency of 120 Hz. The precision of the measurements is not specified by the manufacturer. The accuracy of a similar class of devices (e.g. UM7-LT) for dynamic situations is about $\pm 5^{\circ}$ for *Pitch* (θ) and *Roll* (Φ), whereas $\pm 8^{\circ}$ for *Yaw* (Ψ) angle.

2.2 Preprocessing

The data acquisition process was carried out within a few days. Due to the participation of many participants, and thus the long duration of the experiments, the measurements were carried out in more than one room (caused by the organisational issues).

Inertial motion systems allow to determine the orientation of limbs, in relation to the global coordinate system oriented according to the Earth's magnetic pole.

While performing the gestures, some of the participants were directed towards the magnetic South while others were directed towards the magnetic East of the Earth. In further codebook generation work, leaving the data unchanged would affect the *Yaw* angle and interfere with the classification accuracy. Therefore, the preprocessing procedure involved the artificial rotation of the individual persons in order to orient them in the same direction. In general, researchers use various types of normalisation techniques. In [9] the authors multiplied the data by a quaternion conjugated to q0 where "q0 is the heading offset with respect to the magnetic north". Since the perception neuron was equipped with a sensor located on the back, information about its orientation was used. The use of artificial rotation is a necessary process, commonly exploited also in motion data analysis [10].

According to the methodology presented in the literature [2,4], the codebook was generated using orientation information. First, the time series of quaternions were converted into a series of Euler angles. For this purpose, reverse trigonometric functions presented in equation (1) were used.

$$\begin{aligned} \Psi &= atan\tfrac{2q_yq_w-2q_xq_z}{1-2q_y^2-2q_z^2} \\ \theta &= asin(2q_xq_y+2q_zq_w) \\ \Phi &= atan\tfrac{2q_xq_w-2q_yq_z}{1-2q_x^2-2q_z^2} \end{aligned} \tag{1}$$

where:
q_w, q_x, q_y, q_z -quaternion coponents;
Ψ, θ, Φ -rotation angle *Yaw*,*Pitch*,*Roll*.

It should be emphasized that in the proposed instead of the function `atan` the procedure `atan2` was used. As a result, the angles *Yaw* (Ψ) and *Roll* (Φ) are in the range of $\pm 180^\circ$ while *Pitch* (θ) in the range of $\pm 90^\circ$.

2.3 CodeBook generation

In the next stage of the study, according to the methodology presented in [2,4], the angles of rotation were discretized. The three states sequences (resulting from the 3 rotation axes) were used to generate final observations. In order to discretize the Euler angles, the parameter L [2] was defined, which describes the number of states into which the range of 180° was divided. For example, for a parameter L equal to 3, 180° was divided into ranges of 60°. In this case, the *Pitch* (θ) angle generated one of the states in the $0 \div (L-1))$ range, which means one element from $\{0,1,2\}$. As

a result of the `atan2` function, angles *Yaw*(Ψ) and *Roll*(Φ) have a range of 360°. The state determination process is described in Algorithm 1.

Algorithm 1 State (angle, range, L)

Require: $angle \in \langle 0;180\rangle, range \in \{180,360\}, L \in \{3,4,5,6,7,8\}$

```
if range=180 then
    thr=linspace(0,180,L+1)
else
    thr=linspace(0,360,2L+1)
end if
for i = 2 to i < size(thr) do
    if angle<=thr(i) then
        state=i-2
        break
    end if
end for
return state
```

In Fig.4. an exemplary division of Euler angles into states for case A) L=3 and B) L=4 is presented.

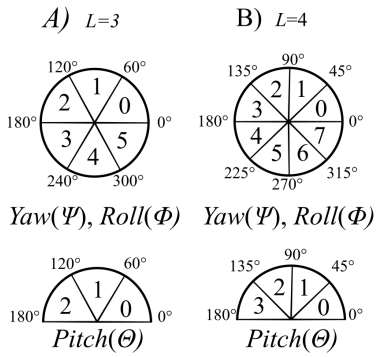

Fig. 4. Generation of angle states based on parameter L parameter A) L=3 B) L=4

In the case described in the literature [2], the final observation was a combination of states of three angles of rotation. Additionally, when the angle *Pitch* (θ) was equal to 0 or $L-1$, the angle state *Roll* (ϕ) was ignored (Algorithm 2). This assumption has reduced the number of observable states.

Algorithm 2 Observation (Yaw, Pitch, Roll, L)

Require: $Yaw, Roll \in \langle 0;360 \rangle, Pitch \in \langle 0;180 \rangle, L \in \{3,4,5,6,7,8\}$
1: **if** $state180(Pitch, L) == 0$ **or** $L-1$ **then**
2: $O = 0 + 2L \cdot state(Pitch, 180, L) + 2L^2 \cdot state(Yaw, 360, L) + 1$
3: **else**
4: $O = state(Roll, 360, L) + 2L \cdot state(Pitch, 180, L) + 2L^2 \cdot state(Yaw, 360, L) + 1$
5: **end if**
6: **return** O

In the original formula (algorithm 2), the value of the *Roll* angle significantly affects the generation of a sequence of observations. It should be noted that this angle has a full range of 360°. Therefore, for each L parameter, it is possible to generate $2L$ states based only on this angle. For example, for parameter $L = 6$, the range of 360 ° will be divided into 12 states with a range of 30°. This assumption significantly increases the requirement for a training base. Therefore, in this paper we propose a modified method of generation of observable states, in which, regardless of the value of the L parameter, the *Roll* was divided into equal states with a width of 120°(Algorithm 4).

Algorithm 3 Proposed observation (Yaw, Pitch, Roll, L)

Require: $Yaw, Roll \in \langle 0;360 \rangle, Pitch \in \langle 0;180 \rangle, L \in \{3,4,5,6,7,8\}$
1: **if** $state(Pitch, 180, L) == 0$ **or** $L-1$ **then**
2: $O = 0 + 2L^2 \cdot state(Pitch, 180, L) + \cdot state(Yaw, 360, L) + 1$
3: **else**
4: $O = (4L^2) \cdot state(Roll, 360, 3) + 2L^2 \cdot state(Pitch, 180, L) + \cdot state(Yaw, 360, L) + 1$
5: **end if**
6: **return** O

2.4 Classification

The aim of the paper was to verify whether the methods described in the article [2] can be used to classify HCI gestures. In their original form, the presented methods were dedicated to the gestures used in VR games. In the conducted studies, the influence of the L parameter, determining the vectorization of data on the accuracy of classification, was examined.

The conducted experiments concerned the recognition of 4 gestures described as "Come", "Turn Right", "Turn Over" and "Sit Down". Each of these gestures was

represented by a sequence of observations related to the forearm orientation. The paper presents the influence of the $L = \{3,4,5,6,7,8\}$ discretization parameter on the results of gesture recognition. In this study, 3-fold cross-validation repeated 5 times was used.

For the classic method taken from the literature, a total of 360 hidden Markov models were trained (4 gestures · 6 variants L parameter · 3 cross validation · 5 repetition). The article presents a modification of Euler angles discretizing parameter. Therefore additional 360 models were trained.

The observations were related to the method of Euler angles discretization. Therefore, the number of observable states M depended on the parameter L. For the classic method (Algorithm 2), the total number of observed states is described by Equation 2.

$$M = 4 \cdot L^3 - 8 \cdot L^2 + 4 \cdot L + 1, \tag{2}$$

The number of observable states for the modified discretization algorithm (Algorithm 3) is described by Equation 3.

$$M = 6 \cdot L^2 - 8 \cdot L + 1, \tag{3}$$

The Literature [2] does not specify the number of hidden states N . Therefore, pilot studies have been carried out in which a fixed number of states of 4 has been selected. For example, for the parameter L=3 (the least complicated models), hidden Markov models were described by parameters with dimensions: 1×4, vector of initial state probabilities; 4×4, matrix of transition probabilities; 4×49, matrix of emission probabilities (classic approach) or 4×31, matrix of emission probabilities (proposed approach).

Training and prediction of hidden Markov models was carried out with the use of *seqHMM* package in *R* programming language. The model parameters were set using the Baum-Welch algorithm. Due to the gradient character of the method, the learning process was restarted 100 times. The classification of individual observations O was carried out using the Viterbi algorithm and the required determination of likelihood indicators for each of the four trained models. The series was classified as the most likely gesture. The observation could therefore be classified correctly, incorrectly or not at all.

3. Resuts

Fig.5. shows 3 classification results described as "Classic Approach", "Proposed Approach" and "Literature Results". The first two use the author's database. The "Proposed Approch" refers to our methodology, whereas the "Classic Approach" to the

methodology used in the publication [2]. the "Literature Results" are results taken directly from the publication [2] and relate to outcomes obtained using the authors' database (which is not public available). In our opinion, a significant difference between "Classic Approach" and "Literature Results" is due to the different types of gestures used in the various databases.

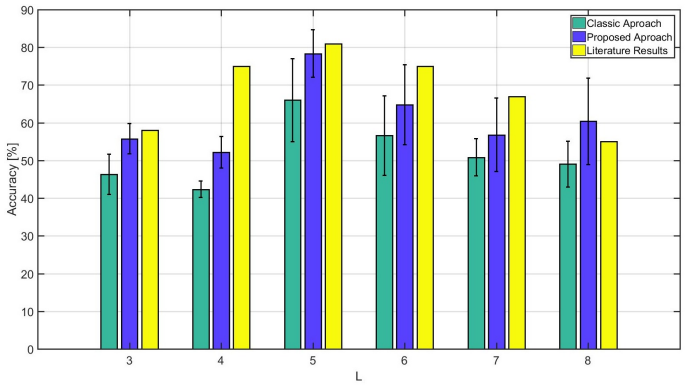

Fig. 5. Classification accuracy as a function of parameter L. Error bars represent standard deviation.

The figure shows the averaged results for the 3-fold cross validation, repeated 5 times. The classification accuracy was presented on the y-axis, whereas L parameter values were presented along the x-axis. Please note that the publication [2] presents only averaged results as a graph, and do not provide any information about standard deviation. Therefore, "Literature Results" bars do not contain error plots.

The highest value for "Proposed Approach" as well "Classic Approach" classification accuracy was achieved for $L = 5$, which is consistent with the "Literature Results". The use of classic codebook generation methodology provides maximum classification accuracy of about 65 %, while the modified method has increased accuracy by about 10 percentage points. The p-score coefficient (Student's t-test) for individual values of L parameter equals: 8.66×10^{-6}; 1.02×10^{-8}; 7.84×10^{-4}, 4.26×10^{-2}; 4.27×10^{-2}; 2.08×10^{-3} respectively. For all cases of comparison of the accuracy of the "Proposed Approach" to the "Classic Approach", a significant statistical difference is observed(p<0.05). A comprehensive summary of the average classification accuracy for particular groups of gestures is presented in Table 1. The presented data are related to the average of a total 15 iterations (3-fold cross-validation repeated 5 times).

Table 1. Classic and Proposed approach gesture classification results. The table presents the mean values and associated standard deviation.

Classic approach accuracy[%]						
Gesture \ L	3	4	5	6	7	8
Turn Right	65.0 ± 8.2	37.8 ± 3.1	84.2 ± 1.8	55.9 ± 17.8	60.0 ± 3.6	44.9 ± 4.8
Sit Down	16.4 ± 3.6	21.4 ± 2.0	48.9 ± 24.9	54.3 ± 23.2	23.7 ± 7.2	31.0 ± 10.4
Turn Over	61.3 ± 10.8	38.6 ± 6.3	44.8 ± 29.6	29.4 ± 5.5	28.8 ± 20.8	34.3 ± 13.3
Come	42.6 ± 6.1	71.7 ± 4.7	86.1 ± 3.1	86.7 ± 5.4	90.9 ± 4.5	85.9 ± 10.3
Proposed approach accuracy[%]						
Gesture \ L	3	4	5	6	7	8
Turn Right	65.0 ± 8.2	52.2 ± 8.6	83.3 ± 11.9	61.7 ± 20.5	60.7 ± 16.3	44.7 ± 28.9
Sit Down	15.6 ± 3.1	25.3 ± 7.8	47.3 ± 32.7	45.0 ± 31.8	22.3 ± 11.5	36.9 ± 25.8
Turn Over	100.0 ± 0.0	60.6 ± 2.1	97.2 ± 2.1	60.6 ± 2.1	72.3 ± 30.1	79.1 ± 17.2
Come	42.4 ± 6.1	70.4 ± 7.8	85.6 ± 3.9	91.9 ± 3.4	71.9 ± 15.9	80.9 ± 16.9

From the presented data, it can be stated that the maximum recognition accuracy is observed for different values of the parameter L for each gesture. For the classic discretization method, the "Turn Right" and "Turn Over" gestures classification accuracy is lower than the "Turn Right" or "Come" motions. In the proposed approach, "Turn over" gestures recognition accuracy has significantly increased. At the same time, recognition of the remaining gestures has not changed significantly. No significant decrease in the recognition of "Turn Right" or "Come" motion patterns was observed. On the other hand, the accuracy of "Sit Down" gesture recognition remained low.

4. Conclusions

As a part of the work, the author developed a comprehensive algorithm for recognizing selected HCI gestures registered with the inertial sensors. The paper uses author's data corpus containing the signals representing a set of four gestures described as "Come", "Turn Right", "Turn Over" and "Sit Down" (20 participants, 720 timeseries). As a consequence of the conducted experiments it was found that the methodology described in the literature [2] and dedicated to the recognition of VR gaming gestures cannot be directly applied to HCI gestures datasets. In the case of using the classic methods on the author's data, the classification accuracy of approximately 65% was obtained.

The paper proposes modification of codebooks generating algorithm, in particular limiting the number of states generated from the *Roll* angle of forearm rotation.

In this study an uneven division of Euler's angles was proposed, in which the state of *Roll* angle assumed only one of three values. As a consequence of the changes, the classification accuracy increased about 10 % points in comparison with the results obtained with the classic algorithm (Fig. 5).

This work provides comprehensive information about the impact of Euler angle discretization on the classification accuracy. It should be emphasized that regardless of the L parameter value, for the new algorithm of codebook generation a higher average accuracy of classification was obtained (in relation to the classic method). Despite the effort of modifying the codebook algorithm, the presented approach is still not universal. The accuracy of the "Sit Down" gesture classification can be considered as insufficient. Therefore, optimization of the algorithm will be the subject of further experiments.

References

[1] Wu, Y., Chen, K., Fu,C.: Natural Gesture Modeling and Recognition Approach Based on Joint Movements and Arm Orientations, IEEE Sensors Journal, Volume: 16, Issue: 21, 2016.

[2] Arsenault, D., Whitehead, A.D.: Gesture recognition using Markov Systems and wearable wireless inertial sensors, IEEE Transactions on Consumer Electronics, Volume: 61, Issue: 4, 2015.

[3] Elmezain, M., Al-Hamadi, A., Michaelis, B.: A hidden markov model-based isolated and meaningful hand gesture recognition, International Journal of Electrical, Computer, and Systems Engineering 3.3: 156-163, 2009.

[4] Elmezain, M., Al-Hamadi A., Michaelis, B.: Hand Gesture Spotting Based on 3D Dynamic Features Using Hidden Markov Models, Communications in Computer and Information Science, vol 61. Springer, Berlin, Heidelberg, 2009.

[5] Di Benedetto A., Palmieri F.A.N., Cavallo A., Falco P.: A Hidden Markov Model-Based Approach to Grasping Hand Gestures Classification, Advances in Neural Networks. WIRN 2015. Smart Innovation, Systems and Technologies, vol 54. Springer, Cham, 2016.

[6] Georgi, M.; Amma, C.; Schultz, T.: Recognizing Hand and Finger Gestures with IMU based Motion and EMG based Muscle Activity Sensing, BIOSTEC 2015 Proceedings of the International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 4, 2015.

[7] Amma, C., Georgi, M.,Schultz, T.: Airwriting: a wearable handwriting recognition system Personal and Ubiquitous Computing, Volume 18, Issue 1, 2014.

[8] Chen, C., Jafari, R., Kehtarnavaz, N.: A Real-Time Human Action Recognition System Using Depth and Inertial Sensor Fusion 6th International Workshop on Advances inSensors and Interfaces (IWASI), 2015.

[9] Comotti, D., Caldara, M., Galizzi, M., Locatelli, P., Re, V.: Inertial based hand position tracking for future applications in rehabilitation environments IEEE SENSORS JOURNAL, VOL. 16, NO. 3, FEBRUARY 1, 2016.

[10] Li, Q., Wang, Y., M., Sharf, A., Cao, Y., Tu, C., Chen, B., Yu, S.: Classication of gait anomalies from kinect The Visual Computer,vol. 34, no. 2, 2018.

ROZPOZNAWANIE GESTÓW INTERAKCJI CZŁOWIEK-KOMPUTER ZAREJESTROWANYCH PRZY UŻYCIU INERCYJNYCH CZUJNIKÓW RUCHU POPRZEZ NIEJAWNE MODELE MARKOVA

Streszczenie Artykuł przedstawia algorytm rozpoznawania wybranych gestów interakcji człowiek-komputer zarejestrowanych przy pomocy inercyjnych czujników ruchu. W niniejszej pracy zweryfikowano możliwości wykorzystania niejawnych Modeli Markova jako klasyfikatora. Zbadano możliwość zastosowania metodyki dedykowanej rozpoznawaniu gestów gry VR do klasyfikacji gestów HCI. W pracy dokonano porównania skuteczności klasyfikacji w zależności od sposobu dyskretyzacji zarejestrowanych sygnałów orientacji przedramienia. Ocena skuteczności klasyfikacji odbyła się z wykorzystaniem trójkrotnej walidacji krzyżowej. W pracy wykorzystano autorski korpus danych zawierający 20 uczestników oraz łącznie 720 szeregów czasowych.

Słowa kluczowe: HMM, Klasyfikacja, HCI, IMU

The work was supported by grant WI/WI/1/2019 from Białystok University of Technology and funded with resources for research by the Ministry of Science and Higher Education in Poland. Funded by the PROM Project: “International scholarship exchange of PhD candidates and academic staff” within the Operational Programme Knowledge Education Development, co-financed from the European Social Fund. The author is grateful to Sławomir Krzysztof Zielinski for substantive contribution.