Tomasz HACHAJ[1], Marek R. OGIELA[2], Danuta BARANIEWICZ[1]

Pedagogical University of Krakow (1), AGH University of Science and Technology (2)

doi:10.15199/48.2017.05.12

Evaluation of Gesture Description Language in the role of touchless interface for virtual reality environment

Abstract. *The main novelty presented in this paper is application and evaluation our Gesture Description Language (GDL) classifier in the role of touchless interface for virtual reality (VR) environment. In our VR system whole interaction is done by gestures and body movements analysis (so called natural user interface). For our needs we have adapted semi-realistic VR system (block engine). We have tested different aspects of proposed interface on a group of 26 persons with wide range of age (from 5 years to 40+) and both sexes. The results we obtained prove that GDL can be successfully applied in systems that require real time action recognition especially educational software games that aim at increasing the students' motivation and engagement while they learn.*

Streszczenie. *W tym artykule autorzy prezentują zastosowanie klasyfikatora o nazwie Język Opisu Gestów (GDL) w roli bezdotykowego interfejsu systemu wirtualnej rzeczywistości. Całość interakcji z zaprezentowanym w tej pracy systemem odbywa się poprzez rozpoznanie i analizę ruchu użytkownika (jest to tak zwany interfejs naturalny). Klasyfikator GDL został przetestowany w pseudo – realistycznym, wirtualnym środowisku, na grupie 26 osób obu płci w przedziale wiekowym od 5 do 40+ lat. Otrzymane rezultaty dowodzą, że zaproponowane podejście może być z powodzeniem użyte w aplikacjach wykorzystujących wirtualną rzeczywistość, w szczególności w grach edukacyjnych, których celem jest uatrakcyjnienie procesu zdobywania wiedzy. (**Ocena działania Języka Opisu Gestów w roli bezdotykowego interfejsu w środowisku wirtualnej rzeczywistości**).*

Keywords: virtual reality, Gesture Description Language, natural user interface, unsupervised learning
Słowa kluczowe: rzeczywistość wirtualna, Język Opisu Gestów, naturalny interfejs człowiek – komputer, uczenie bez nadzoru

1. Introduction

Virtual reality (VR) is among the most technically and scientifically challenging branches of computer science. A few years ago the possibility of creating an artificial but in the same time realistic environment in which user behavior in real world was mapped onto actions of avatar in VR was limited to specially equipped laboratories. Today off the shelf graphic cards are capable to render large and realistic VR. Also popular multimedia controllers like Kinect can be easily applied to segmenting and tracking user activities in real time. Due to this fact many up to date researches are conducted in the field of user movements analysis for VR systems. VR is about creating environments that are believable [1]. Among the most important features that increase the impression of user immersion in three-dimensional system are lack of distraction from the technology for example cabling or on-body wearable instruments. It is hardly possible to produce believable immersive context of virtual reality when user uses traditional mouse and keyboard - based interface to navigate the avatar. With the help of innovative wireless natural user interface provided by Kinect (Xbox 306 version) hardware and GDL classification library, we created the believable VR. The proof that the produced system was believable was successful interaction of users with the system, very good cooperation and true enthusiasm of participants of the experiment.

In this section we will present the state of the art of VR and gesture analysis. We will also present our motivation for this researches and our contribution to this field.

1.1 State of the art

Broad state-of-the art reports about different aspects of virtual-reality system can be found in [2][3][4]. The subject of this paper is presentation and evaluation of our new approach of touchless interface for three-dimensional virtual reality systems. Because of the scope of our paper we must focus on graphic rendering protocol and user interface designing.

A. Graphic protocol

The authors often choose computer graphic method that enables creation of three-dimensional environment which looks more or less realistic and can be easily modified and modeled for specialized tasks. Nowadays there are many commercial three-dimensional rendering engines that enable efficient rendering of photorealistic worlds. Those implementations often support programming interfaces and sophisticated CAD-engines that require deep computer graphic knowledge to be useful. That strongly limits the application of those modules to group of specialists in computer graphic. Because of that in scientific literature authors propose some additional “abstract layers” that facilitate the access to computer graphic support. For example application presented in [5] is based on a visual programming paradigm, supported by a semantic representation ontology for virtual environments. The ontology allows storing and organizing the components of a 3D scene, together with the knowledge associated with them. It is also used to expose functionalities in the given 3D engine. Based on a formal representation of its components, the proposed architecture provides a scalable VR system. Paper [6] discusses object-oriented virtual worlds building process, utilizing the software development technique of stepwise refinement, and possibilities of parallel processing in VR systems. Papers [7][8] present current research efforts and useful applications that employ 3D graphics with particular focus on addressing the unique features of mobile devices (ubiquity, connectivity, context-awareness, and multimodal interfaces). The 3D surface models are often created in offline preparation steps, which make it difficult to detect changes and to adapt the 3D model to these changes. Work [9] presents a 3D change detection and model adjustment framework that combines augmented reality (AR) techniques with real-time depth imaging to close the loop between dense 3D modeling and augmented reality. In paper [10] the ongoing development of a system, called VirDe is presented. It allows the designers to perform the whole design process, from the modelling phase to the finite element method (FEM) simulation analysis in a virtual reality environment. 3D model creation and visualization is often used in digital preservation of looks and structure of historical buildings. Paper [11] shows a preservation procedure, which consists of 3D scanning, 3D model creation, optimization and visualization on the historical building.

B. Interface and gesture recognition

The issues of touchless interface basing on body movements analysis are in the scope of interest of gestures/actions recognition and natural user interface (NUI). Paper [12] presents general state of the art discussion about gesture recognition approaches. Due to this fact in this section we will only discuss papers that directly mention the interfaces protocols used in virtual reality environments.

Historically three-dimensional virtual reality systems have used same interface as other computer applications (like keyboard, mouse etc.) [13]. Lately many novel data acquisition devices for gathering user gestures, position and spatial orientation have been introduced. Among them are precise motion capture systems that however require pre-prepared and dedicated environment to be functional or to wear special costume or accessories [14]. Beside those solutions there is also a group of multimedia devices that can be used for man-machine interactions. Among them are controllers that were designed initially for video games. Those devices already showed their usability in other tasks than entertainment, for example in medicine as a training device [15] (Nintendo Vii) or NUI [16] (MS Kinect), physical education [17] and rehabilitation [18][19] (also MS Kinect).

Microsoft Kinect is a very popular hardware module that can be used in various applications for example in medicine [20], intelligent home technology [21][22] nonverbal behavior analysis [23], 3-D kinematic measurement [24] or even zootechnics [25]. However its most popular applications are human – computer interfaces [26][27].

Quite often VR designers decide to use interfaces borrowed from AR marker-based [28][29][30][31] or marker less [32] solutions. One of the most important aspect of VR system is to supply users with an interface that is intuitive (natural) and precise. Authors in [33] emphasis that even in the most sophisticated and costly VR systems, people do not necessarily perceive and behave as they would in the real world. They suggest this might be related to our inability to use embodied (and thus often highly automated and effective) spatial orientation processes in VR. Also report [34] states that egocentric distances in virtual environments are commonly underperceived by up to 50 % of the intended distance.

We can summarize that designing proper interface for VR system is challenging tasks that requires not only adequate hardware and gesture recognition method but also exhausting process of calibration and validation.

1.2 The motivation for this paper

As can be seen in the state of the art section there are not many researches that evaluates VR systems that uses touchless interface. By the term “touchless” we mean the type of interface that does not require touch contact with traditional input devices (like mouse, keyboard or pads) and does not require any sensors installed on body. Those types of sensors increase the accuracy of user tracking however they require amount of time to put them on and to calibrate.

The main novelty presented in this paper is application and evaluation our Gesture Description Language (GDL) [12] classifier in the role of a touchless interface for virtual reality (VR) environment. In our VR system whole interaction is done by gestures and body movements analysis (so called natural user interface). Our proposed system does not require any prior calibration. For our needs we have adapted semi-realistic VR system (block engine). We have tested different aspects of proposed interface on a group of 26 persons with wide range of age (from 5 years to 40+) and both sexes.

Most of the popular computer games that utilize Kinect technology classify basic human actions like walking, running etc. however the possible navigation of avatar in game environment is very limited. Most often avatar travels through predefined path and the role of user is only to omit the obstacles by dodging or jumping what makes a game virtually one dimensional. This approach of game developers can be easily explained: it might be physically exhausting for some gamers to set his or her avatar in motion by walk or running. Because of this developers decide to give up the physical educational aspects of gameplay replacing it with not so much demanding entertainment. The introduction of fully human movements controlled user interface in fully three-dimensional environment which users in different age and body proportions can freely explore is very important novelty of our approach.

2. Materials and methods

In this section we present the description of VR system we have used for NUI evaluation. We also present the setup of GDL-based touchless interface.

2.1 Three-dimensional environment

To evaluate GDL in the role of body tracking touchless interface we have created virtual reality environment. The output from GDL classifier was mapped to input interface of VR engine. We have used Voxel Game [35] engine written in OpenTK (this is a C# wrapper for OpenGL [36]). Voxel Game is so called voxel engine. In this type of VR world is a construct of 3D cubes of identical size. Due to this approach one can very easily develop semi-realistic 3D map without necessity of exhausting modeling of triangle meshes and textures mapping. The graphic engine was enhanced by adding an active element called “Box”. When user reaches a Box instead of 3D environment the engine render 2D screen with additional tasks to be performed. We will write more about those tasks later in this section. In Figure 1 in the left we have presented how user is situated in our system. The NUI capture device is a single Microsoft Kinect sensor. The data from device is processed with Kinect SDK tracking library on PC class computer. The same machine also renders the VR on large screen that is situated directly in front of a user. All information that is necessary to walk through VR is displayed on this screen. User does not have any type of controller, the only way he or she can communicate with VR world is by body movements.

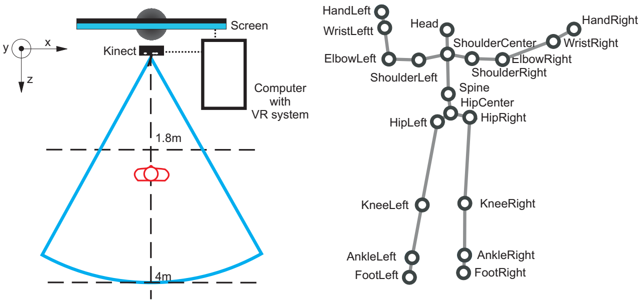

Figure 1. In this figure on the left we have presented how user is situated in our system. On the right we have presented set of Kinect SDK body joints (so called skeleton) that are automatically segmented from user body and tracked with frequency 30 Hz.

2.2 Gesture Description Language and unsupervised generation of key frames

The GDL is a rule-based classifier specialized in detection of body movements activities and gestures recognition. It utilizes elements of expert systems and uses GDL script (GDLs) language that is context-free grammar to define rules for the knowledge base. The definition of GDL classifier with its evaluation on set of different body movements were published in papers [12] [37]. Also those papers contains formal definition of GDLs grammar. The implementation of GDLs from [12] and [37] is accessible for free for scientific and educational purposes [38]. The implementation of classier requires third-party library that supply application with stream of body joints (so called skeletons) segmented from data acquisition sensor recording – see Figure 1. The gestures descriptions in knowledge base can be written manually and/or automatically generated with unsupervised learning method reversed-GDL (R-GDL) [39] from exemplar skeletons recordings of users (so called SKL recordings).

GDL approach bases on assumption, that a movement can be represented as a sequence of key frames and key frames should pay some crucial role for gesture identification. Because of this they should also be “remarkably visible” in a feature space. That means that key frames should create clusters in feature space that might be detected with machine-learning approach. We are interested in detection of central points of clusters (Gaussian distribution model is reported to be applied in human movements analysis [40]) which will create “the root” of key frames. Because we want to classify human movements we can more or less estimate how many important key frames are present in each movement activity. The k-means clustering algorithm satisfies all above conditions.

The Microsoft Kinect application programming interface contains two algorithm that enables gestures classification. First one is an AdaBoostTrigger with which is it possible to detect the presence of an action in incoming motion capture signal. This algorithm is a variation of AdaBoost algorithm and uses multiple weak classifiers. The second algorithm is an RFRProgress which uses Random Forest Regression that detects the progress of a gesture performed by the user. Their implementations have very important limitations. It is impossible to manually specify the features of actions that we want to classify - the approach is trained on the predefined features set. Also the only available method of training classifier is machine learning. Because of that even very simple actions have to be recorded and presented to the training algorithm. In contrary GDL can use the same GDLs file to store both manually defined gestures and results of machine learned actions. It is very simple to combine those actions together or to make extensions of already learned gestures. This approach is not possible in original Kinect algorithms which learning results are not designed to be interpreted by a user.

The output from our algorithm are GDLs descriptions of a key frames. They have following form (see Appendix H):

1. Features' definition that maps joints space into feature space, in which K-means clustering was computed.
2. Set of features generated by data clustering, those are epsilon values, centers of clusters computed as average value of all elements assigned to them (all features have the same weight), and spatial dimensionality of clusters computed as standard deviation value of all elements assigned to them plus epsilon value.
3. Key frames definition: each key frame corresponds to the single cluster. Single key frame has a following pattern of rule definition:

```
RULE abs(feature_1_value –
feature_1_average_value_in_a_given_cluster)
<=
feature_1_standard_deviation_in_a_given_clus
ter + feature_1_epsilon_value
AND (…)
RULE abs(feature_L_value –
feature_L_average_value_in_a_given_cluster)
<=
feature_L_standard_deviation_in_a_given_clus
ter + feature_L_epsilon_value
THEN Key_frame_for_a_given_cluster
```

Key frames are ordered into sequences with following GDLs formula:

```
    RULE key_frame_1
    & sequenceexists("[key_frame_2,
    key_frame_2_persistance_time]
    (…)
    [[key_frame_M,
    key_frame_M_persistance_time]")
    THEN Conclusion
```

The previous rule will be satisfied as long as key_frame_1 rule is satisfied and the sequence satisfies the time constraints. However if we would like to force GDL classifier to return the conclusions that a gesture has been identified only once per gesture that might be done with the following construction:

```
RULE Conclusion &
sequenceexists("[!Conclusion,1]") THEN
DetectedGestureName
```

This is especially useful when we want to use generated script not only to check if the particular gesture exists in given SKL recordings but also to count how many time it appeared.

The order of key frames is identify by finding most frequent n-gram [39]. After K-means algorithm each frame of skeleton recording (SKL) is assigned to cluster. This creates a stream of class symbols. Each set of neighboring elements with same class symbol is replaced by a single symbol. Now we can compute which n-gram of class symbols is this reduced stream appears most often. The n value in n-gram type is defined by the user, but n should not be less than number of clusters. The most frequent n-gram determinates the sequence of key frames. The time that pass between each key frame is computed as the longest time duration of the particular class in clustered SKL stream.

2.3 Application of Gesture Description Language as Natural User Interface

In order to apply NUI to virtual environment appropriate body movements have to be mapped onto commands to computer program. In our VR we wanted to test three types of interactions with the system:

1. Mouse – like two-dimensional point and click interface (P&C),
2. Movements commands, that are used to travel through VR (Mov),
3. Special body activities that are performed only during rare circumstances (Spec).

Our NUI interface is disabled when user is too close to the data capturing sensor. We set the distance threshold to 220 cm because basing on our observation, when an adult user (about 180 cm height) crosses this distance very often our data tracking software is unable to track all body joints (see Appendix A for GDLs definition of this rule).

In our implementation of P&C interface screen cursor is steered by right hand of the user. The position of cursor on screen is calculated as relative position of right hand to the point in the middle of the user chest. The “click” action is activated when a cursor is over “clickable” object for at least 3 seconds. This P&C interface is implemented outside the GDL paradigm.

In our VR environment there are four types of Mov commands: walking (or running) straight (walking/running might be performed simultaneously with any other Mov command), turning right, turning left and jumping. All of those commands are implemented as GDL heuristics. We designed those movements to be as much similar to natural movements as it can be. However we must be aware the fact, that NUI interface should be accessible for people with different age, agility and body proportions. Because of that some simplifications are necessary, and we decided not applied here R-GDL. It is recommended to use R-GDL method when a movement we want to recognize includes large parts of body and cannot be described by less than four time-varying features. In that case it is virtually impossible to generate manual descriptions of actions and machine learning approach has to be used. In experiment presented in this paper body weight squat and jumping jacks actions were trained with R-GDL method.

Jumping can be described as continues rise of y (horizontal) coefficient that represents the central part of the body (spine joint). In order to eliminate the tracking distortions that are generated by hardware and software the movement must last for a given period of time (in our case 0.3 sec) and the direction trend should be preserved - in 50% of captured frames the horizontal movement in up direction must be greater than 1 cm – see Appendix B. Jump also move a person forward in the same manner as the “forward” command was executed. In our VR one can jump over single voxel block, however two voxel height walls are too high to cross.

The right and left turning are realized by shoulder balance – see Appendix C. As can be seen there are two thresholds for each rotation direction: $[15°, 35°)$ and $[35°, 180°]$. The rotation speed for first threshold is 0.5 radians per second, in second case it is 2 radians per second. Because it is only possible to walk straight, in order to move the opposite direction one has to turn $180°$, what last about 1.5 sec.

Walking is represented by sequence of two key frames – see Appendix D. The GDL examines the right and left knee flection and relative vertical positions of knees. Those descriptions are satisfied when a person is walking (or walking in a place). The GDLs for running is very similar, however knees flection have to be greater than during “walking” and frequency of movement has to be at least two times faster than in “walking” definition. Those descriptions are satisfied when a person is running (also in a place) lifting knees high.

The walking speed is 4 voxel per second. The running speed is 16 voxels per second. The forward movements have inertia (implemented in VR engine) that modifies the movement speed in time:

$$\begin{cases} v(i) = v_0 \cdot \left(\frac{1}{1.1}\right)^i & when\ v(i-1) > 0.1\ or\ i = 0 \\ 0 & otherwise \end{cases}$$

where v_0 is initial speed of movement, *i* is an index of discrete time span (1/60 sec each) that passed from beginning of the movement when *i=0*.

The last - Spec - group of gestures we wanted to examine are two heuristic and two descriptions computed with R-GDL. The hand overhead definition (Appendix F) is obvious. The hand clapping GDLs contains sequence of two key frames (see Appendix G). The GDL first checks if distance between hands is greater than 20 cm then second the opposite condition. Due to this definition the description is invariant to the way a person situates hands relatively to body while clapping. The next two GDLs define popular gym exercises: body weight squat (BWS) and jumping jacks (JJ). Each of this exercise was trained separately as it was described in Section 2.2. We chose 3 as the number of cluster in k-means clustering, and 10 as epsilon parameter for all features. The training dataset consists of 87 complete body weight squat SKL recording gathered from 10 persons and 92 jumping jacks recordings also gathered from 10 adult persons with age 28+. The training group for those actions was completely different group than test group. The complete GDLs of jumping jacks are present in Appendix H. In case body weight squats in Appendix I we only present features that were used for unsupervised training because complete clustering results will not introduce much new information to the reader.

3. Experiment and results

The goal of our experiment was to evaluate the effectiveness and efficiency of GDL classifier in the role of touchless interface for VR system. We have prepared a virtual location in which users were immersed – see Figure 2. The path that a user has to go is surrounded by a wall that cannot be crossed (in Figure 2 the wall is composed of gray voxels). Each user was evaluated separately (only one user is present in virtual world at once). He or she has a goal to go from Start location to end location which is inside the House. This path has several important stages that are important for evaluation of our approach.

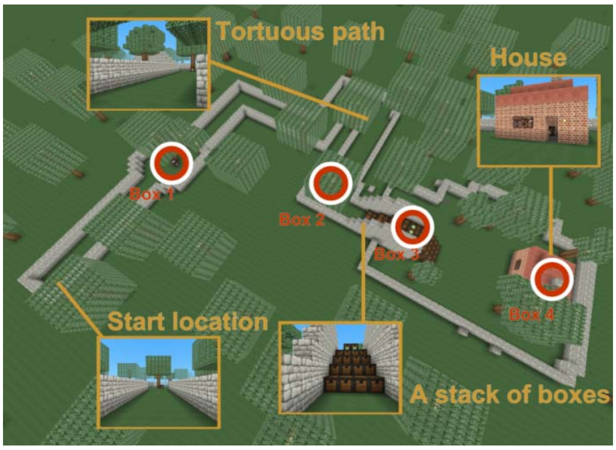

Figure 2. This figure presents aerial view of virtual world the experiment participants were asked to go through. We have marked a start location, boxes that pay the roles of checkpoints and some important obstacles that have to be overcome during the virtual travel.

After starting the virtual world simulation a set of 2D screens are visualized to user. Each 2D screen contains text commands (all people taking part in experiment were capable of reading beside a 5 year old child, to whom the commands were spoken). In this commands we asked a user to perform some activity. Those activities teaches basic actions that are necessary to steer avatar in VR. All of those actions (beside P&C) were evaluated by GDL module. The commands were presented automatically (by our system) one after another only after the previous one was satisfied. The tracked user skeleton was recorded into SKL file for offline evaluation. Because those commands intuitive and natural gestures and movements as the effectiveness of our solution we measure the time after which the system is capable to recognize the gesture that a user was intended to do. The shorter the time is the better interaction a user has with the system. None of the experiment participants have previous experience neither with this particular VR system nor with a GDL classifier.

The evaluation protocol goes as follow:

1) Before a user “arrives” to 3D world he is asked by the system to do a set of activities that teaches him or her basis of the NUI interface:

a. A user is asked to raise right hand over head.

b. To show correct image (one from two) with virtual cursor that is steered by his or her hand. The cursor has to persist for three seconds in the region of correct image.

c. To show correct image (one from three – see Figure 3 (c)).

d. To walk in place for about 6 seconds.

e. To run in a place for about 6 seconds (a user has to lift his or her knees higher than during walking and do movements a bit faster).

f. To turn right.

g. To turn left.

h. To jump two or more times.

2) After step h the VR 3D environment is automatically displayed to a user (see Figure 3 (a) and (b)). A user has to reach Box 1 from a start location. He or she can walk, run, turn left / right and jump. The aim of these tasks is to evaluate if user is capable to walk from one location to the another. A user sees his destination from the beginning. The path to this location is straight.

3) After reaching Box 1, user is asked to reach Box 2. One has to go through “tortuous path”, that have several turns. Box 2 is not initially visible from location of Box 1.

4) From Box 2 user should go to Box 3. To do so user has to jump onto a stack of boxes.

5) After reaching top of the boxes' stack, the user has to enter the House that has 1-voxel wide entrance (see Figure 3 (a) – the house entrance is visible from the top of the stack, in Figure 3 (b) we can see interior of the House). Inside the House is the last box (Box 4).

6) The last set of tasks evaluates GDL’s capability of recognition of complex gestures. The user is asked:

a. To clap hands.

b. To do three body weight squats.

c. To do three jumping jacks.

After task 6 – c the evaluation procedure ends.

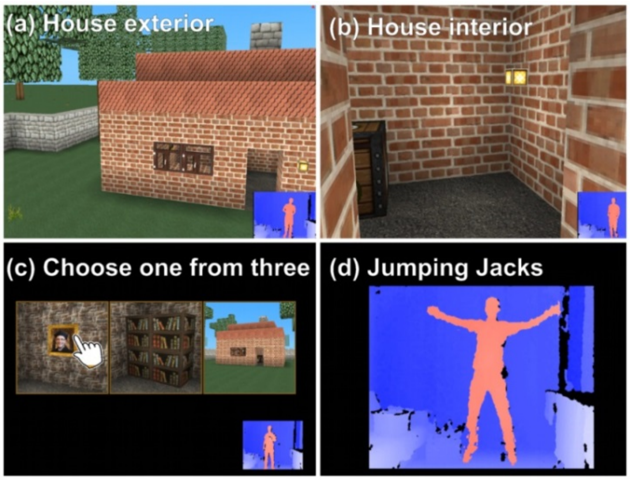

Figure 3. This figure presents the screenshots of visualizations of three different types of tasks that are performed by experiment participants in our virtual world. First two screenshots ((a) and (b)) are renderings of voxel-based environment, both exterior and interior of objects. The user is travelling across the environment by performing some predefined body movements (Mov). Screenshot (c) is an example of task when a user has to indicate the correct answer (one from three) by moving a virtual cursors with his hand (P&C, in this case to show a painting on the wall). The cursor has to remain three seconds in the region of particular image. The last example (d) shows visualization of segmentation of user’s body while he is performing jumping jacks, which is one of (Spec) tasks in our evaluation procedure.

Table 1. This table presents results of our experiment. Each row of a table describes one participant. Columns have information about the sex and age of participant and time he or she spends until finishing particular task. The order of columns in the table corresponds to order of those tasks. Last two rows summarizes the results giving the overall number of users of both sexes and average and standard deviation of time that has been spent for particular task.

No	Sex	Age	Hand up	1 from 2	1 from 3	Walking	Running	Turn r.	Turn l.	Jump	Box 1	Box 2	Box 3	Box 4	Clapping	BWS	JJ
1	F	24	1.1	3.9	6.1	7.8	5.5	4.1	1.9	2.0	7.2	41.5	11.7	14.6	1.6	4.8	3.6
2	M	24	0.5	3.5	4.8	5.4	6.7	3.0	2.4	2.4	7.2	53.7	7.7	13.6	2.0	4.7	3.8
3	F	24	3.3	3.5	8.0	16.4	6.2	4.5	1.0	2.8	11.3	136.5	19.9	12.6	0.9	5.9	3.4
4	F	24	3.4	4.5	3.7	5.7	4.9	4.2	6.8	5.7	7.2	51.2	75.9	77.4	1.1	10.6	6.1
5	M	24	0.6	3.0	6.1	4.5	6.5	3.5	2.1	3.1	8.4	42.2	16.2	13.5	1.8	5.6	10.6
6	M	24	1.8	3.3	6.3	5.4	5.5	8.2	1.8	1.2	11.1	29.1	21.7	10.0	1.3	6.4	4.2
7	F	20	0.3	4.0	6.7	15.0	5.8	5.0	1.9	2.2	7.2	88.2	15.4	14.0	0.9	10.3	3.4
8	F	20	0.9	4.5	5.1	8.7	4.6	3.9	2.4	2.0	14.6	42.9	6.1	26.3	2.5	6.1	24.6
9	F	20	0.2	3.3	5.1	2.5	5.7	3.0	2.6	1.7	11.2	45.5	17.9	12.6	1.1	5.3	3.6
10	F	20	0.5	4.4	6.6	4.9	4.7	2.3	3.8	2.7	14.0	34.1	11.8	11.6	0.9	4.8	3.4
11	M	20	1.3	4.4	4.7	4.3	4.8	3.0	2.6	1.8	10.9	20.4	10.6	4.6	1.3	5.8	4.5
12	F	20	1.9	3.4	5.2	5.9	5.8	4.2	2.9	1.9	11.7	38.9	6.4	7.0	1.7	4.6	3.5
13	F	20	1.1	4.2	3.3	9.0	5.1	4.2	4.8	1.1	5.5	36.9	3.5	47.7	1.6	5.6	3.8
14	F	40+	1.5	8.7	15.3	30.9	4.3	0.4	5.3	3.3	22.3	283.8	36.9	92.6	2.2	14.1	6.1
15	M	40+	0.8	4.9	5.0	8.5	4.5	4.5	3.5	5.5	8.0	55.5	11.5	14.0	1.0	17.3	5.2
16	M	40+	3.6	3.7	4.2	7.4	9.3	3.2	3.2	22.6	12.4	37.8	8.7	4.6	3.8	7.2	21.1
17	M	30-39	1.0	4.1	6.7	7.7	5.3	5.0	3.4	3.6	11.6	52.0	9.1	19.6	1.9	6.3	4.5
18	M	30-39	1.1	3.4	3.3	6.0	5.1	2.9	2.6	1.7	4.0	22.2	18.7	56.7	0.7	9.2	4.1
19	M	30-39	1.0	4.3	4.4	5.3	6.5	4.8	2.3	3.3	13.3	30.9	12.4	7.6	2.3	5.8	4.3
20	M	30-39	1.1	4.8	6.5	7.2	5.0	3.5	2.2	2.0	6.5	21.6	9.9	14.4	1.4	4.8	4.1
21	M	5	7.1	11.9	11.9	8.5	5.0	8.0	5.5	7.7	11.0	91.4	10.1	22.9	5.1	9.1	6.4
22	M	16	0.3	3.9	5.5	7.6	4.3	3.3	2.2	2.1	4.7	55.1	7.8	10.2	2.2	4.5	8.2
23	F	16	0.7	4.8	5.5	3.9	5.7	2.8	2.6	2.0	11.8	48.8	9.2	11.6	1.0	6.9	19.0
24	F	16	1.1	4.6	6.3	5.1	9.9	3.3	2.2	2.0	4.4	34.7	6.4	9.9	1.6	5.0	3.7
25	M	17	0.7	5.4	3.6	8.5	3.5	2.8	2.1	2.0	3.1	18.7	5.9	4.9	2.2	4.2	22.3
26	F	16	0.7	3.9	4.6	5.1	5.3	2.9	2.2	2.6	7.3	55.4	4.6	12.7	1.1	5.6	13.6
Sum M	13	Avg. value	1.5	4.6	5.9	8.0	5.6	3.9	2.9	3.5	9.5	56.5	14.5	21.1	1.7	6.9	7.7
SUM F	13	St. Deviation	1.5	1.8	2.6	5.6	1.4	1.6	1.3	4.2	4.2	52.8	14.3	22.4	1.0	3.1	6.6

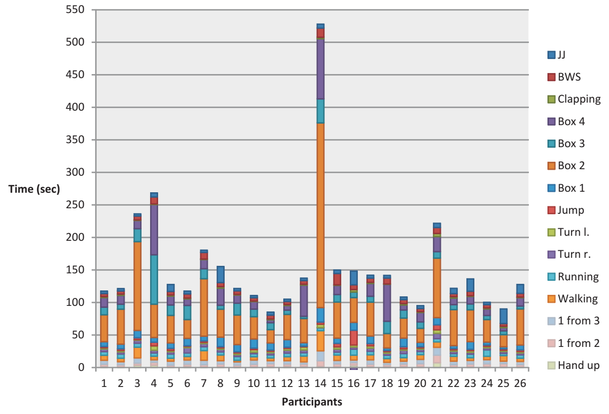

Figure 4. This plot presents visualization of results from Table 1.

26 people (13 men and 13 women) have taken part in our experiment. The age of participants differs from 5 to 40+ years. Thanks to this we have validated our approach on users' data that differs in dexterity, movement patterns and body proportions. The experiment's results are presented in Table 1. The experiment was performed on two consumer-quality PC computers. The first was equipped with an Intel Core i7-4470 CPU 3.40 GHz processor, 8 GB RAM, Nvidia GeForce GTX 770 graphics card, 64-bit Windows 7 Home Premium. Second one had Intel Core i5-2320 CPU 3.00GHz, 8 GB RAM, AMD Radeon HD 6570 graphic card, Windows 7 Home Premium 64 Bit.

In Figure 4 we have presented the plot bar that visualizes the results from Table 1. The role of this plot is to show how much time a participant spends doing particular task and to visualize difference between users. Each bar corresponds to a user, the color codes correspond to the task he or she did in virtual environment. Indexes of user are presented in horizontal axis. The time in seconds is in vertical axis. The higher the bar is, the more time a user spent finishing some activity.

The shortest path between start and Box 1 was 29 voxels (7.25 sec of walking), between Box 1 and Box 2 79 voxels (19.75 sec), between Box 2 and Box 3 14 voxels (3.5 sec) and between Box 3 and Box 4 16 voxels (4 sec). However it have to be remembered that above walking times (we will later call them “optimal”) do not include the activities of rotation or jumping that are often necessary to finish those tasks.

Figure 5 also presents visualization of results from Table 1, but this time the average time a user spent doing a task plus/minus standard deviation of that time. Each bar of the plot is average time that participant spent performing particular task measured in seconds. Red bar is plus/minus standard deviation of this time.

Table 2 and Figure 6 presents relative time of finishing Box tasks by all users compared to user No 25 who showed to be faster than most of other participants. We decided to use 25th participant as the reference person to compare the times of finishing boxes tasks because in most cases he was the fastest person. He finished Box 1 and Box 2 tasks as the first and there were only two persons faster than him in Box 3 (number 13 and 26) and only two persons faster than him in Box 4 (number 11 and 16). The time user No 25 spent doing Box tasks are divided by the time of a given user. The smaller the value is the more time a person spends completing a task. It might be interpreted as a type of scoring – the less points you get the worse your timing is. It is obvious that relative time for user No 25 equals 1. Colors of the bars correspond to users.

Table 2. This table presents relative times of finishing Box tasks of all experiment participants from Table 1.

No	Box1	Box2	Box3	Box4
1	0.4	0.5	0.5	0.3
2	0.4	0.3	0.8	0.4
3	0.3	0.1	0.3	0.4
4	0.4	0.4	0.1	0.1
5	0.4	0.4	0.4	0.4
6	0.3	0.6	0.3	0.5
7	0.4	0.2	0.4	0.4
8	0.2	0.4	1.0	0.2
9	0.3	0.4	0.3	0.4
10	0.2	0.5	0.5	0.4
11	0.3	0.9	0.6	1.1
12	0.3	0.5	0.9	0.7
13	0.6	0.5	1.7	0.1
14	0.1	0.1	0.2	0.1
15	0.4	0.3	0.5	0.4
16	0.3	0.5	0.7	1.1
17	0.3	0.4	0.6	0.3
18	0.8	0.8	0.3	0.1
19	0.2	0.6	0.5	0.6
20	0.5	0.9	0.6	0.3
21	0.3	0.2	0.6	0.2
22	0.7	0.3	0.8	0.5
23	0.3	0.4	0.6	0.4
24	0.7	0.5	0.9	0.5
25	1.0	1.0	1.0	1.0
26	0.4	0.3	1.3	0.4
Avg. value	0.4	0.5	0.6	0.4
St. Deviation	0.2	0.2	0.4	0.3

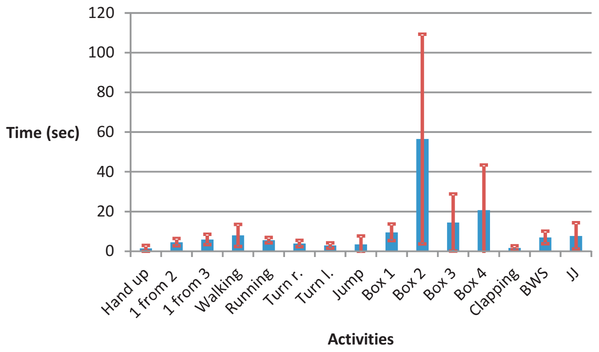

Figure 5. This plot presents visualization of results from Table 1 (more precisely the last two rows of that table)

4. Discussion

Before overall discussion on our approach we will discuss participant No 14 separately. This is a person that had worst results and we can conclude that our interface was not intuitive to her at all, however she managed to finish the experiment. After experiment she admitted, that she has no experience with any type of application that generates VR even most basic video games. We believe that that factor and stress that is indelible human factor resulted in such low performance of our NUI. We will not take this person among further discussion however we did not excluded this person's results from numeric calculations.

The results for other persons were far more satisfying. In case of P&C interface 1 from 2 and 1 from 3 tasks were quickly solved (the average times spend on them were 4.6 and 5.9 sec). That time contains additional 3 seconds in which a user cannot move a hand in order to “select” correct answer. The remaining time lasted selection of proper response.

The Mov commands were evaluated in “Walking”, “Running”, “Turn r”, Turn l”, “Jump” and “Box 1- 4” tasks. The tasks “Walking”, “Running”, “Turn r”, Turn l” were designed to force the user to perform each type of NUI activity correctly several times. As can be seen each person made those gestures quickly – the average times to those five tasks were 8.0, 5.6, 3.9, 2.9 and 3.5 sec. The “Box” tasks were done in 3D VR and times that users spent to finish them indicates how well our Mov NUI performs. The minimal possible time to do those tasks (while walking) are given in Section 3 however as we already mentioned they are unrealistic because they did not take into account jumping and left/right turns. However for example the participant No 25 managed even to overcome some of those times (!) because he was running. As can be seen in Table 2 and Figure 6 the average relative time differences between user No 25 and the others are in the range of 0.4 to 0.6. That is not much for this type of interface and means that our “Mov” NUI interface is intuitive for those users. The most common cause of longer time needed to complete a box task was due to the fact that user omitted some crucial part of the path (for example house doors in task Box 4) and had to turn and walk right direction.

The last type of NUI we evaluated was “Spec”. While the hand overhead and clapping are easy to detect actions and did not last more than a few second to be classified (avg. 1.5 and 1.7) the other two actions requires discussion. Jumping jacks and body weight squat are the physical activities that not every person is capable to do correctly. Our automatically generated descriptions with R-GDL were made from recordings that consisted only correct cases. The average time for BWS was 6.9 and for JJ was 7.7 which is a the time the average person without daily workout needs. We can observe the trend that older persons often needed more time to do BWS than teenagers (compare BWS results for cases No 14, 15 and from 22 to 26). Those persons were doing BWS much slower than other persons or have to repeat same exercise because they were unable to make it correctly at first attempt (for example to hold correct body position). In case of JJ we can observe that some peoples do not have coordination between legs and hands movements and they fail to make this exercise correctly in first attempt. Those are mostly young people (for example 5, 8, 23, 25 and 26) but also one from the group 40+ (case 16). Concluding, both manually designed and automatically generated GDLs descriptions are classified successfully.

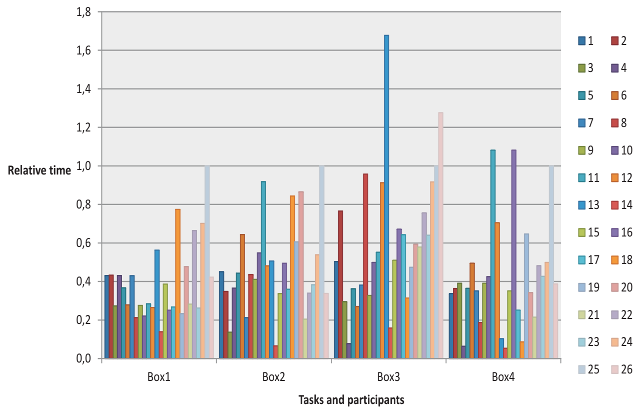

Figure 6. This figure presents relative time of finishing Box tasks of all experiment participants.

The standard deviations for all considered NUI elements are high. In case of Mov NUI they have value close to avg. This is of course the sign that times participants needed to finalize those tasks differs substantially. However as we discussed before even the longest time that occurs in our dataset does not disqualify our NUI interface due to long latency time or lack of intuitiveness.

The very interesting case we wanted to discuss separately is participant No 21 who had only 5 years. As can be seen in Table 1 and 2 he had more problems than the others with P&C interface however in Mov and Spec NUI his results were not far for average and were far for the worst. The data from single user is not enough to make generalization however it indicates that our NUI can be applicable even for a person whose movements coordination and body proportions differs much even from teenagers.

5. Conclusions

Basing on results from our experiment we can conclude that Gesture Description Language can be successfully applied in the role of touchless interface for virtual reality environment. The results we obtained for Mov and Spec NUI were satisfying. In case of Mov NUI the results were in most cases worse than “optimal” (see Section 3) however the difference in times in most cases were not too big to make the interaction inconvenient. In this paper we have also presented the complete setup of GDLs rules. This might be a valuable material for scientists or industry that would like to design VR with similar NUI interface as ours or directly apply our results in practice. Our approach can be also easily adapted to any other three-dimensional environment however the initial calibration of movements speeds and in-world physics have to be made to integrate GDL technology with graphic module.

Our future researches will be concentrated on applying our touchless interface and VR system in educational games. It has been already showed that educational software games aim at increasing the students’ motivation and engagement while they learn [12]. It has been suggested that VR technology allows knowledge-building experiences and in this way provides an alternative educational process [41][42]. However, if software games are targeted to school classrooms they have to be usable and likeable by all students. Because of that the content of the game has to be adequately tailored to age and psychical development of students. Due to this we have to investigate the efficiency of our GDL-based NUI on group of young pupils (first classes of elementary schools) in order to find the age limit of users for which our interface is operable. We will also expand our VR system prototype to enable easy designing of 3D worlds and its content even for person who has no experience with programming. Thanks to this our solutions will be usable for teachers from various domains of knowledge, which together with pupils will be the end-users of our future solution. With intuitive user interface for virtual world teachers will be capable to generate they own exercises, dedicated for particular age or even particular students. Those exercises will incorporate the features that were already explored in this paper that is Mov activity that are required to find “a Box” that contains some task for a student and Spec or P&C. We have observed that travelling through well designed and easily controlled virtual world is pleasant and solving tasks “inside boxes” satisfying. Basing on this observation we believe our approach will stimulate especially young people.

Appendix

This appendix contains scripts that we used in our NUI. The GDLs conclusions from this section with exclamation marks are directly mapped to appropriate commands of VR interface. The explanations of those scripts are in Section 2.

A. Distance from sensor

```
RULE distance(spine.xyz[0],[0,0,0]) < 2200 THEN
ToNear!
```

B. Jumping

```
RULE spine.y[0] - spine.y[1] > 10 THEN MoveUp
RULE rulepersists(MoveUp, 0.3, 0.5) THEN Jump!
```

C. Turning left and right

```
FEATURE 90 - angle(ShoulderRight.xyz[0] -
ShoulderLEFT.xyz[0], [0,0,1]) AS
MoveDirectionAngle
FEATURE sgnfuzzy(MoveDirectionAngle,15) AS
MoveDirection
RULE MoveDirection = 1 & (MoveDirectionAngle >=
15 & MoveDirectionAngle < 35) THEN Rights1!
RULE MoveDirection = 1 & (MoveDirectionAngle >=
35) THEN Rights2!

RULE MoveDirection = -1 & (MoveDirectionAngle <=
-15 & MoveDirectionAngle > -35) THEN Lefts1!
RULE MoveDirection = -1 & (MoveDirectionAngle <=
-35) THEN Lefts2!
```

D. Walking

```
FEATURE angle(HipRight.xyz[0] - KneeRight.xyz[0],
AnkleRight.xyz[0] - KneeRight.xyz[0]) AS
RightKnee
FEATURE angle(HipLeft.xyz[0] - KneeLeft.xyz[0],
AnkleLeft.xyz[0] - KneeLeft.xyz[0]) AS LeftKnee

RULE KneeRight.y[0] - KneeRight.y[1] > 0 THEN
RightKneeUp
RULE KneeLeft.y[0] - KneeLeft.y[1] > 0 THEN
LeftKneeUp

RULE RightKnee > 150 & LeftKnee > 150 THEN
WalkingStart
RULE RightKnee > 150 & LeftKnee < 150 &
RulePersists(LeftKneeUp, 0.25, 0.5) THEN
WalkingLeft
RULE RightKnee < 150 & LeftKnee > 150 &
RulePersists(RightKneeUp, 0.25, 0.5) THEN
WalkingRight

RULE WalkingStart &
sequenceexists("[WalkingLeft,0.5][WalkingStart,0.
5]") THEN WalkingStepOne1
RULE WalkingStart &
sequenceexists("[WalkingRight,0.5][WalkingStart,0
.5]") THEN WalkingStepOne2

RULE WalkingStepOne1 | WalkingStepOne2 THEN
Walking!
```

E. Running

```
RULE RightKnee > 130 THEN WalkingRight1f
RULE RightKnee < 130 THEN WalkingRight2f
RULE LeftKnee > 130 THEN WalkingLeft1f
RULE LeftKnee < 130 THEN WalkingLeft2f
RULE WalkingRight1f &
sequenceexists("[WalkingRight2f,0.25][WalkingRigh
t1f,0.25]") THEN WalkingStepOne1f
RULE WalkingLeft1f &
sequenceexists("[WalkingLeft2f,0.25][WalkingLeft1
f,0.25]") THEN WalkingStepOne2f
RULE WalkingStepOne1f | WalkingStepOne2f THEN
WalkingF!
```

F. Hand overhead

```
RULE HandRight.y[0] > Head.y[0] & Not(ToNear!)
THEN HandOverHead
```

G. Clapping

```
FEATURE
distance(HandRight.xyz[0],HandLeft.xyz[0]) AS
HandsDistance
RULE HandsDistance > 200 THEN HandsSeparete
RULE HandsDistance < 200 THEN HandsTogether
RULE HandsSeparete &
SequenceExists("[HandsTogether,0.5][HandsSeparete
,0.3]") THEN Clapping!
```

H. Jumping Jacks

```
//------------Input features ----------------
FEATURE angle(ShoulderRight.xyz[0] -
ElbowRight.xyz[0],
WristRight.xyz[0] - ElbowRight.xyz[0])
AS RightElbow
FEATURE angle(ShoulderLeft.xyz[0] -
ElbowLeft.xyz[0],
WristLeft.xyz[0] - ElbowLeft.xyz[0])
AS LeftElbow
FEATURE angle(ShoulderCenter.xyz[0] -
ShoulderRight.xyz[0],
ElbowRight.xyz[0] - ShoulderRight.xyz[0])
AS RightShoulder
FEATURE angle(ShoulderCenter.xyz[0] -
ShoulderLeft.xyz[0],
ElbowLeft.xyz[0] - ShoulderLeft.xyz[0])
AS LeftShoulder
FEATURE angle(ShoulderRight.xyz[0] -
ElbowRight.xyz[0],
ShoulderLeft.xyz[0] - ElbowLeft.xyz[0])
AS BetweenWrists
FEATURE angle(KneeLeft.xyz[0] - HipLeft.xyz[0],
KneeRight.xyz[0] - HipRight.xyz[0])
AS BetweenLegs

//------------Generated features-------------
FEATURE 10 AS rightelbow_EPS
FEATURE 10 AS leftelbow_EPS
FEATURE 10 AS rightshoulder_EPS
FEATURE 10 AS leftshoulder_EPS
FEATURE 10 AS betweenwrists_EPS
FEATURE 10 AS betweenlegs_EPS

FEATURE 147 AS rightelbow_MEAN_0
FEATURE 13 AS rightelbow_DEV_0
FEATURE 154 AS leftelbow_MEAN_0
FEATURE 11 AS leftelbow_DEV_0
FEATURE 118 AS rightshoulder_MEAN_0
FEATURE 11 AS rightshoulder_DEV_0
FEATURE 118 AS leftshoulder_MEAN_0
FEATURE 10 AS leftshoulder_DEV_0
FEATURE 108 AS betweenwrists_MEAN_0
FEATURE 26 AS betweenwrists_DEV_0
FEATURE 26 AS betweenlegs_MEAN_0
FEATURE 7 AS betweenlegs_DEV_0

FEATURE 157 AS rightelbow_MEAN_1
FEATURE 13 AS rightelbow_DEV_1
FEATURE 158 AS leftelbow_MEAN_1
FEATURE 10 AS leftelbow_DEV_1
FEATURE 157 AS rightshoulder_MEAN_1
FEATURE 10 AS rightshoulder_DEV_1
FEATURE 159 AS leftshoulder_MEAN_1
FEATURE 11 AS leftshoulder_DEV_1
FEATURE 125 AS betweenwrists_MEAN_1
FEATURE 31 AS betweenwrists_DEV_1
FEATURE 10 AS betweenlegs_MEAN_1
FEATURE 7 AS betweenlegs_DEV_1

FEATURE 153 AS rightelbow_MEAN_2
FEATURE 11 AS rightelbow_DEV_2
FEATURE 151 AS leftelbow_MEAN_2
FEATURE 10 AS leftelbow_DEV_2
FEATURE 127 AS rightshoulder_MEAN_2
FEATURE 21 AS rightshoulder_DEV_2
FEATURE 127 AS leftshoulder_MEAN_2
FEATURE 23 AS leftshoulder_DEV_2
FEATURE 32 AS betweenwrists_MEAN_2
FEATURE 15 AS betweenwrists_DEV_2
FEATURE 10 AS betweenlegs_MEAN_2
FEATURE 11 AS betweenlegs_DEV_2

//------------Generated rules----------------
RULE abs(rightelbow -rightelbow_MEAN_0) <=
rightelbow_DEV_0 + rightelbow_EPS
& abs(leftelbow -leftelbow_MEAN_0) <=
leftelbow_DEV_0 + leftelbow_EPS
& abs(rightshoulder -rightshoulder_MEAN_0) <=
rightshoulder_DEV_0 + rightshoulder_EPS
& abs(leftshoulder -leftshoulder_MEAN_0) <=
leftshoulder_DEV_0 + leftshoulder_EPS
& abs(betweenwrists -betweenwrists_MEAN_0) <=
betweenwrists_DEV_0 + betweenwrists_EPS
& abs(betweenlegs -betweenlegs_MEAN_0) <=
betweenlegs_DEV_0 + betweenlegs_EPS
THEN JumpingJacks0

RULE abs(rightelbow -rightelbow_MEAN_1) <=
rightelbow_DEV_1 + rightelbow_EPS
& abs(leftelbow -leftelbow_MEAN_1) <=
leftelbow_DEV_1 + leftelbow_EPS
& abs(rightshoulder -rightshoulder_MEAN_1) <=
rightshoulder_DEV_1 + rightshoulder_EPS
& abs(leftshoulder -leftshoulder_MEAN_1) <=
leftshoulder_DEV_1 + leftshoulder_EPS
& abs(betweenwrists -betweenwrists_MEAN_1) <=
betweenwrists_DEV_1 + betweenwrists_EPS
& abs(betweenlegs -betweenlegs_MEAN_1) <=
betweenlegs_DEV_1 + betweenlegs_EPS
THEN JumpingJacks1

RULE abs(rightelbow -rightelbow_MEAN_2) <=
rightelbow_DEV_2 + rightelbow_EPS
& abs(leftelbow -leftelbow_MEAN_2) <=
leftelbow_DEV_2 + leftelbow_EPS
& abs(rightshoulder -rightshoulder_MEAN_2) <=
rightshoulder_DEV_2 + rightshoulder_EPS
& abs(leftshoulder -leftshoulder_MEAN_2) <=
leftshoulder_DEV_2 + leftshoulder_EPS
& abs(betweenwrists -betweenwrists_MEAN_2) <=
betweenwrists_DEV_2 + betweenwrists_EPS
& abs(betweenlegs -betweenlegs_MEAN_2) <=
betweenlegs_DEV_2 + betweenlegs_EPS
& HandRight.y[0] < ShoulderRight.y[0] &
HandLeft.y[0] < ShoulderLeft.y[0]
THEN JumpingJacks2

//---N-gram based analysis of sequences-------
RULE JumpingJacks2
&
sequenceexists("[JumpingJacks1,1][JumpingJacks0,1
][JumpingJacks1,2][JumpingJacks2,1]")
THEN JumpingJacksOK
RULE JumpingJacksOK &
sequenceexists("[!JumpingJacksOK,0.5]")
THEN JumpingJacks!
```

I. Body weight squat

Below features were used for unsupervised training of body weight squat movement:

```
FEATURE angle(ShoulderRight.xyz[0] -
ElbowRight.xyz[0],
WristRight.xyz[0] - ElbowRight.xyz[0]) AS
RightElbow
FEATURE angle(ShoulderLeft.xyz[0] -
ElbowLeft.xyz[0],
WristLeft.xyz[0] - ElbowLeft.xyz[0]) AS LeftElbow

FEATURE angle(ShoulderCenter.xyz[0] -
ShoulderRight.xyz[0],
ElbowRight.xyz[0] - ShoulderRight.xyz[0]) AS
RightShoulder
FEATURE angle(ShoulderCenter.xyz[0] -
ShoulderLeft.xyz[0],
ElbowLeft.xyz[0] - ShoulderLeft.xyz[0]) AS
LeftShoulder

FEATURE angle(ShoulderRight.xyz[0] -
ElbowRight.xyz[0],
ShoulderLeft.xyz[0] - ElbowLeft.xyz[0]) AS
BetweenWrists


FEATURE angle(HipRight.xyz[0] - KneeRight.xyz[0],
AnkleRight.xyz[0] - KneeRight.xyz[0]) AS
RightKnee
FEATURE angle(HipLeft.xyz[0] - KneeLeft.xyz[0],
AnkleLeft.xyz[0] - KneeLeft.xyz[0]) AS LeftKnee

FEATURE angle(ShoulderCenter.xyz[0] -
HipCenter.xyz[0],
KneeRight.xyz[0] - HipRight.xyz[0]) AS RightHip
FEATURE angle(ShoulderCenter.xyz[0] -
HipCenter.xyz[0],
KneeLeft.xyz[0] - HipLeft.xyz[0]) AS LeftHip
```

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

***Authors**: Tomasz Hachaj, Pedagogical University of Krakow, Institute of Computer Science and Computer Methods, 2 Podchorazych Ave, 30-084 Krakow, Poland, tomekhachaj@o2.pl, Marek R. Ogiela, AGH University of Science and Technology, Cryptography and Cognitive Informatics Research Group, 30 Mickiewicza Ave, 30-059 Krakow, Poland, mogiela@agh.edu.pl, Danuta Baraniewicz, Pedagogical University of Krakow, Institute of Special Needs Education, 30-060 Kraków, R. Ingardena 4, 30-060, Krakow, Poland, danbar3@op.pl*

REFERENCES

[1] Virtual Realities, Dagstuhl Seminar 2008, Editors: Guido Brunnett, Sabine Coquillart, Greg Welch, Springer Vienna, 2011, DOI 10.1007/978-3-211-99178-7
[2] Zhou N-N, Deng Y-L, Virtual Reality: A State-of-the-Art Survey, International Journal of Automation and Computing 6(4), November 2009, 319-325, DOI: 10.1007/s11633-009-0319-9
[3] Houliez C., Gamble E., Dwelling in Second Life? A phenomenological evaluation of online virtual worlds, Virtual Reality (2013) 17:263–278, DOI 10.1007/s10055-012-0218-1
[4] Zhao QP, A survey on virtual reality, Science in China Series F: Information Sciences, March 2009, Volume 52, Issue 3, pp 348-400
[5] García-Rojas A., Gutiérrez M., Thalmann D., Visual creation of inhabited 3D environments An ontology-based approach, The Visual Computer, July 2008, Volume 24, Issue 7-9, pp 719-726
[6] Sobota B., Korečko Š., Hrozek F., On building an object-oriented parallel virtual reality system, Central European Journal of Computer Science, October 2012, Volume 2, Issue 3, pp 261-271
[7] Noguera J. M., Torres J. C., Interaction and visualization of 3D virtual environments on mobile devices, Personal and Ubiquitous Computing, October 2013, Volume 17, Issue 7, pp 1485-1486
[8] Piórkowski A., Jajesnica Ł., Szostek K., Creating 3D Web-Based Viewing Services for DICOM Images, Computer Networks, Communications in Computer and Information Science Volume 39, 2009, pp 218-224

[9] Kahn S., Reducing the gap between Augmented Reality and 3D modeling with real-time depth imaging, Virtual Reality (2013) 17:111–123, DOI 10.1007/s10055-011-0203-0

[10] Ingrassia T., Cappello F., VirDe: a new virtual reality design approach, International Journal on Interactive Design and Manufacturing (IJIDeM), February 2009, Volume 3, Issue 1, pp 1-11DOI 10.1007/s12008-008-0056-2

[11] Hrozek F., Sobota B., Szabó C., Digital preservation of historical buildings using virtual reality technologies, Central European Journal of Computer Science, October 2012, Volume 2, Issue 3, pp 272-282

[12] Hachaj T., Ogiela M. R., Rule-based approach to recognizing human body poses and gestures in real time, Multimedia Systems, February 2014, Volume 20, Issue 1, pp 81-99, (DOI) 10.1007/s00530-013-0332-2

[13] Lee M. W., Lee J. M., Generation and Control of Game Virtual Environment, International Journal of Automation and Computing 04(1), January 2007, 25-29, DOI: 10.1007/s11633-007-0025-4

[14] Sherstyuk A., Jay C., Treskunov A., Impact of hand-assisted viewing on user performance and learning patterns in virtual environments, Vis Comput (2011) 27: 173–185, DOI 10.1007/s00371-010-0516-0

[15] Middleton K. K., Hamilton T., Tsai P-C, Middleton D. B., Falcone J. L., Hamad G., Improved nondominant hand performance on a laparoscopic virtual reality simulator after playing the Nintendo Wii, Surgical Endoscopy, November 2013, Volume 27, Issue 11, pp 4224-4231

[16] Ruppert G. C., Reis L. O., P. H. Amorim, T. F. de Moraes, J. V. da Silva, Touchless gesture user interface for interactive image visualization in urological surgery, World J Urol. 2012 Oct;30(5), doi: 10.1007/s00345-012-0879-0. Epub 2012 May 12, pp. 687-91

[17] Hachaj T., Ogiela M. R., Piekarczyk M., Real-Time Recognition of Selected Karate Techniques Using GDL Approach, Image Processing and Communications Challenges 5, Advances in Intelligent Systems and Computing, Volume 233, 2014, pp 99-106

[18] Clark R. A., Pua Y-H, Fortin K., Ritchie C., Webster K. E., Denehy L., Bryant A. L., Validity of the Microsoft Kinect for assessment of postural control, Gait & Posture 36 (2012), pp. 372–377

[19] Chang Y-J., Chen S-F., Huang J-D., A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities, Research in Developmental Disabilities 32 (2011) pp. 2566–2570

[20] Webster D., Celik O., Systematic review of Kinect applications in elderly care and stroke rehabilitation, Webster and Celik Journal of NeuroEngineering and Rehabilitation 2014 Jul 3;11:108. doi: 10.1186/1743-0003-11-108

[21] Mastorakis G., Makris D., Fall detection system using Kinect's infrared sensor, Journal of Real-Time Image Processing, December 2014, Volume 9, Issue 4, pp 635-646, doi: 10.1007/s11554-012-0246-9

[22] Planinc R., Kampel M., Introducing the use of depth data for fall detection, Personal and Ubiquitous Computing, August 2013, Volume 17, Issue 6, pp 1063-1072 , doi: 10.1007/s00779-012-0552-z

[23] Stevenson Won A., Bailenson J. N., Stathatos S. C., Dai W., Automatically Detected Nonverbal Behavior Predicts Creativity in Collaborating Dyads, Journal of Nonverbal Behavior, September 2014, Volume 38, Issue 3, pp 389-408, doi: 10.1007/s10919-014-0186-0

[24] Dutta T., Evaluation of the KinectTM sensor for 3-D kinematic measurement in the workplace, Applied Ergonomics 43 (2012) pp.645-649, doi 10.1016/j.apergo.2011.09.011

[25] Kongsro J., Estimation of pig weight using a Microsoft Kinect prototype imaging system, Computers and Electronics in Agriculture, Volume 109, November 2014, Pages 32–35, doi:10.1016/j.compag.2014.08.008

[26] Galna B., Barry G., Jackson D., Mhiripiri D., Olivier P., Rochester L., Accuracy of the Microsoft Kinect sensor for measuring movement in people with Parkinson's disease, Gait & Posture, Volume 39, Issue 4, April 2014, Pages 1062–1068, doi:10.1016/j.gaitpost.2014.01.008

[27] Castro C., Kinect Remote-Controlled Vehicles, Chapter, Arduino and Kinect Projects, pp. 207-241, Apress, 2012, DOI 10.1007/978-1-4302-4168-3_9

[28] Ng L. X., Oon S. W., Ong S. K., Nee A. Y. C., GARDE: a gesture-based augmented reality design evaluation system, International Journal on Interactive Design and Manufacturing (IJIDeM), June 2011, Volume 5, Issue 2, pp 85-94

[29] Leith K. Y. Chan, Henry Y. K. Lau, MagicPad: the projection based 3D user interface, International Journal on Interactive Design and Manufacturing (IJIDeM), May 2012, Volume 6, Issue 2, pp 75-81

[30] Lee J. Y., Rhee G. W., Woo Seo D. W., Hand gesture-based tangible interactions for manipulating virtual objects in a mixed reality environment, The International Journal of Advanced Manufacturing Technology, December 2010, Volume 51, Issue 9-12, pp 1069-1082

[31] Lee H., Billinghurst M., Woo W., Two-handed tangible interaction techniques for composing augmented blocks, Virtual Reality (2011) 15:133–146, DOI 10.1007/s10055-010-0163-9

[32] Hürst W., van Wezel C., Gesture-based interaction via finger tracking for mobile augmented reality, Multimed Tools Appl (2013) 62:233–258, DOI 10.1007/s11042-011-0983-y

[33] Riecke B. E., Sigurdarson S., Milne A. P., Moving through virtual reality without moving?, Cognitive processing(2012) 13 (Suppl 1):S293–S297, DOI 10.1007/s10339-012-0491-7

[34] Kelly J. W., Donaldson L. S., Sjolund L. A., Freiberg J. B., More than just perception–action recalibration: Walking through a virtual environment causes rescaling of perceived space, Attention, Perception & Psychophysics, (2013) 75:1473–1485, DOI 10.3758/s13414-013-0503-4

[35] Official website of Voxel Game engine http://www.voxelgame.com/

[36] Szostek K., Piórkowski A., OpenGL in Multi-User Web-Based Applications, Innovations in Computing Sciences and Software Engineering, 2010, pp 379-383

[37] Ogiela M. R., Hachaj T., Natural User Interfaces in Medical Image Analysis, Springer Verlag, Cham Heidelberg, 2015, ISBN 978-3-319-07799-4

[38] Official website of GDL technology http://cci.up.krakow.pl/gdl/

[39] Hachaj T., Ogiela M. R., Full-body gestures and movements recognition: user descriptive and unsupervised learning approaches in GDL classifier, Proc. SPIE 9217, Applications of Digital Image Processing XXXVII, 921704 (September 23, 2014); doi:10.1117/12.2061171

[40] Quattoni A., Darrell T., Latent-Dynamic Discriminative Models for Continuous Gesture Recognition, Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on, IEEE, pp. 1-8 (2007)

[41] Bossard C., Kermarrec G., Buche C., Tisseau J., Transfer of learning in virtual environments: a new challenge?, Virtual Reality (2008) 12:151–161, DOI 10.1007/s10055-008-0093-y

[42] Richard E., Tijou A., Richard P., Ferrier J-L, Multi-modal virtual environments for education with haptic and olfactory feedback, Virtual Reality (2006) 10:207–225, DOI 10.1007/s10055-006-0040-8