The Cognitive Motivation-based APBMR Algorithm in Physical Rehabilitation

Tibor Guzsvinecz[1], Veronika Szucs[2] and Attila Magyar[3]

[1]Department of Information Technology and Its Applications, University of Pannonia, 8900, Zalaegerszeg, Gasparich M. utca 18/A, Hungary

[2,3]Department of Electrical Engineering and Information Systems, University of Pannonia, 8200, Veszprem, Egyetem utca 10, Hungary

[1]guzsvinecz.tibor@zek.uni-pannon.hu, [2]szucs@virt.uni-pannon.hu, [3]magyar.attila@virt.uni-pannon.hu

Abstract: *This article presents a new, alternative method of gesture recognition using the cognitive properties of intelligent decision-making systems to support the rehabilitation process of people with disabilities: the Asynchronous Prediction-Based Movement Recognition (APBMR) algorithm. The algorithm "predicts" the next movement of the user by evaluating the previous three with the goal to maintain motivation. Based on the prediction, it creates acceptance domains and decides whether the next user-input gesture can be considered the same movement. For this, the APBMR algorithm uses six mean techniques: the Arithmetic, Geometric, Harmonic, Contrahamonic, Quadratic and the Cubic ones. The purpose of this article besides presenting this new method is to evaluate which mean technique to use with the three different acceptance domains. The authors evaluated the algorithm in real-time using a general and an advanced computer, as well as they tested it by predicting from a file and also compared the algorithm to one of their earlier works. The tests were done by four groups of users, respectively, each group doing four gestures. After analyzing the results, the authors concluded that the Contraharmonic mean technique gives the best average gesture acceptance rates in the ±0.05 m and ±0.1 m acceptance domains, while the Arithmetic mean technique provides the best average gesture acceptance rate in the ±0.15 m acceptance domain when using the APBMR algorithm.*

Keywords: *cognitive infocommunications; human-computer interaction; Kinect; mean techniques; motivation; prediction-based gesture recognition; real-time gesture recognition; rehabilitation*

1 Introduction

Stroke is one of the most frequent diseases of the modern day. As shown in [1, 2], 48% of people who survived brain-to-asthma disease suffer from half-side paralysis. Not only that, in more than 60% of cases, cognitive decline is detectable. Up to 12-18% of people with this disease are aphasic. 24-53% of stroke patients are partially or completely dependent on other people. Due to this, modern technology should be involved.

The development of stroke rehabilitation methods that do not place additional burdens on the overburdened healthcare systems was stimulated by the increasing number of stroke patients in need of post-stroke rehabilitation [3]. Researchers are working on new rehabilitation methods as virtual reality can be used in neurorehabilitation. There are many initiatives in healthcare, primarily in the field of movement rehabilitation, where some form of gameplay is used. "Serious games" (in other words, games that develop something) that can complement physiotherapy as motion elements are used that are controlled by the motion therapist [4]. Even virtual reality-based games and video games are new, well-used technologies that can be effectively combined with the traditional rehabilitation of an upper limb injury following a stroke.

In these mentioned software, the movement of the patient is monitored through an optical device to indicate if they are practicing properly. There are some applications where the user is placed in a virtual environment and, for example, they handle a phobia or reduce the frustration associated with the rehabilitation procedure [5].

There are many neurorehabilitation techniques based on virtual reality technology that are promising on solving this problem [6-14], however they did not spread across the field of healthcare. These techniques are well-developed judging from an IT perspective, but the users in the healthcare field found them difficult to use and to customize, thereby the patient lost motivation [15, 16]. In some cases, even the sensors should be customized [17].

Telemedicine can also be an important factor [18, 19]. It is more important nowadays than previously, as the hospitals are overcrowded and rehabilitation at the home of the patients is much more convenient for both the patient and the therapist. Therefore, the authors want to present an alternative method besides the existing ones, thus the workers in the field of healthcare could have one more method to choose from.

Fortunately, the area of Cognitive InfoCommunications (CogInfoCom) [20] is ready to highlight new capabilities of ICT in human-machine interactions [21-23]. This provides an opportunity to examine a number of human factors using modern cognitive IT methods [24-26].

In an earlier article in the CogInfoCom environment [27], the authors came to the conclusion that the acceptance domain-based gesture classification can be used in real-time with the Kinect. This could adapt to the current capabilities of the patients, while maintained their motivation in the rehabilitation process. However, it was not always accurate. Therefore, the authors propose the Asynchronous Prediction-Based Movement Recognition (APBMR) algorithm that can even be used at the home of patients if they have a low-cost sensor such as the Kinect. This algorithm is presented and evaluated in this article.

The APBMR algorithm “predicts” the next movement of the user by evaluating the previous three and decides whether the next user-input gesture can be considered the same movement with the goal to maintain motivation. It also follows the position of the user and matches their speed, to make the decision of accepted gestures easier.

This article is structured as the following: section 2 deals with the materials and methods. Section 3 presents the results, containing both real-time and file-based evaluations, section 4 discusses them and in Section 5, conclusions are made.

2 Materials and methods

The authors divided this section into three subsections: subsection 2.1 presents the idea which led to the development of the APBMR algorithm, while subsection 2.2 defines it in detail and subsection 2.3 presents the collected data.

2.1 The idea of the APBMR algorithm

The authors frequently use the Kinect v1 and v2 for research regarding motion analysis with the main goal to develop new methods to help the physical rehabilitation of people with disabilities. While these devices can be used in the medical field by substituting more expensive sensors [28], with [29] or without data filtering [30], there are two problems with both Kinects regarding gesture recognition: the first is that when the user stands at a different distance from the sensor than before, it returns new x, y, z coordinates. Therefore, when repeating the same gestures at a different position, it may not be recognized. Another problem is the speed: when doing the same gestures at a different speed, the gestures may not be recognized as well.

Thus, the authors devised and developed the APBMR algorithm in 2019 using C#, which follows the position of the users and also, tracks the speed of the movement. This algorithm is similar to two of the previous algorithms of the authors, namely the Reference Distance Based Synchronous / Asynchronous Movement Recognition (RDSMR/RDAMR) algorithms [27]. They are similar as they use the same gesture acceptation domain principle, but they will be compared and elaborated in subsection 3.3 and section 4, respectively.

Therefore, the following is hypothesized: *The average of accepted gestures is larger when the APBMR algorithm is used than in the case of the previous algorithm it was based upon and it can also be used for telerehabilitation.*

It should also be mentioned, that even though the APBMR is developed for the Kinects; in principle, it can be used with any sensor that sends movement descriptor data in real-time and evaluates gestures by using coordinates.

2.2 Presenting the APBMR algorithm

In this subsection, the APBMR algorithm is presented. Before defining the steps of the algorithm, two pieces of crucial information should be noted. The first is that the algorithm only evaluates one axis at a time. After it completes the evaluation on one axis, it starts the evaluation on another. When all three axes are evaluated, the gestures are either accepted or rejected. The other piece of information is that the algorithm looks for repeating gestures in the movement descriptors by searching for the farthest and the closest coordinates from the starting point on the currently evaluated axis. This can be seen in Figure 1.

Imagine that the starting coordinate of a gesture is the leftmost X in the figure. First, the algorithm determines whether the starting coordinate is at the bottom or at the top of a "slope". Then, the algorithm searches for the coordinate that is the farthest from the starting coordinate (illustrated with the second X): in the figure the starting coordinate is at 0.5220696 and the farthest is at 0.1476125. Since the starting coordinate was at a top of a "slope" in the case of this example, then it looks for the farthest coordinate at the bottom. Reaching this coordinate means that the gesture is about halfway done. Afterward, the algorithm searches for the coordinate that is the closest to the starting coordinate (illustrated with the third X). In this case, it is at 0.5437541. As can be suspected, it is not closest numerically, because searching for this coordinate has a few criteria: If the starting coordinate is at the top of a slope, then the closest coordinate also has to be at the top. Naturally, if the starting coordinate is at the bottom, then the closest coordinate has to be at the bottom as well. Also, the closest coordinate must be after the previous farthest coordinate. If this closest coordinate is reached, then the algorithm can conclude that this coordinate is the end of the first gesture. Afterward, the coordinate symbolized by the third X in the figure becomes the new starting coordinate for the next gesture and the algorithm repeats these steps.

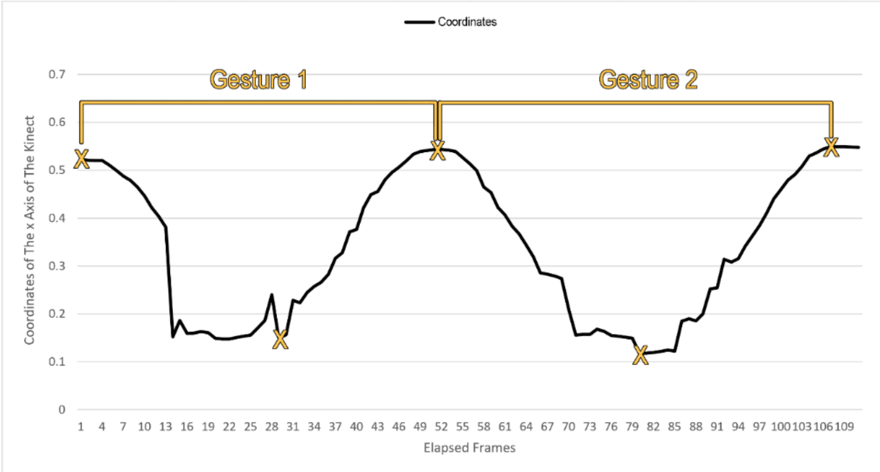

Figure 1
Illustration of how algorithm calculates the number of gestures and the length of the gestures.

After defining how the algorithm finds the gestures, let us look at how it works, step-by-step:

1. Scans the number of done gestures by searching for the closest and farthest coordinate points (referred to as “clofarpoint” later on) to the starting coordinate point in the movement descriptors.
2. Calculates the average length of the scanned gestures.
3. Predicts the possible next movement on the x axis and its acceptance domains based on the last three done gestures using mean techniques. This step has multiple substeps:
   a. While $i < clofarpointnumber - 6$, it calculates the length of the previous three movements based on the following rules:

$$x_1 = clofarpoint_{i+2} - clofarpoint_i \quad (1)$$

$$x_2 = clofarpoint_{i+4} - clofarpoint_{i+2} \quad (2)$$

$$x_3 = clofarpoint_{i+6} - clofarpoint_{i+4} \quad (3)$$

, where $clofarpointnumber$ is the number of all “clofar” in the gesture descriptors. Variable i is incremented by 2 in each cycle.

   b. It creates an average of these lengths using a mean technique (mtk). This can be selected by the user ($k \in [1,6]$). The used mean techniques were the Arithmetic average (4), Geometric average (5), the special case of Harmonic average for three numbers (6), Contraharmonic average (7), Quadratic average (8) and the Cubic average (9). The special case of Harmonic average was required as the regular Harmonic average equation gave “Not a Number” (NaN) results during measurements.

$$mt_1 = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4}$$

$$mt_2 = \sqrt[n]{\prod_{i=1}^{n} x_i} \tag{5}$$

$$mt_3 = \frac{3x_1x_2x_3}{x_1x_2+x_1x_3+x_2x_3} \tag{6}$$

$$mt_4 = \frac{x_1^2+x_2^2+\cdots+x_n^2}{x_1+x_2+\cdots+x_n} \tag{7}$$

$$mt_5 = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} \tag{8}$$

$$mt_6 = \sqrt[3]{\frac{1}{n}\sum_{i=1}^{n} x_i^3} \tag{9}$$

, where, similarly, as in equations (1-3), x_i is the length of the i^{th} gesture. Also, since the algorithm uses the previous three gestures, n = 3 in all mt_k equations.

c. Generates a new coordinate called $predictedc_j$ at frame j, while $j < mt_k$ and $minmax_{i+4} + j + 1 < x_i$ by using one of the mentioned mean techniques and the following rules:

$$y_1 = \begin{cases} c_{clofarpoint_i + j}, & \frac{x_i - j}{mt_k} \geq \frac{mt_k - j}{mt_k} \\ \frac{c_{clofarpoint_i+j} + c_{clofarpoint_i+j+1}}{2}, & otherwise \end{cases} \tag{10}$$

$$y_2 = \begin{cases} c_{clofarpoint_{i+2}+j}, & \frac{x_i - j}{mt_k} \geq \frac{mt_k - j}{mt_k} \\ \frac{c_{clofarpoint_{i+2}+j} + c_{clofarpoint_{i+2}+j+1}}{2}, & otherwise \end{cases} \tag{11}$$

$$y_3 = \begin{cases} c_{clofarpoint_{i+4}+j}, & \frac{x_i - j}{mt_k} \geq \frac{mt_k - j}{mt_k} \\ \frac{c_{clofarpoint_{i+4}+j} + c_{clofarpoint_{i+4}+j+1}}{2}, & otherwise \end{cases} \tag{12}$$

$$predictedc_j = \begin{cases} \frac{1}{3}\sum_{l=1}^{3} y_l, & k = 1 \\ \sqrt[3]{\prod_{l=1}^{3} y_l}, & k = 2 \\ \frac{3y_1y_2y_3}{y_1y_2+y_1y_3+y_2y_3}, & k = 3 \\ \frac{y_1^2+y_2^2+y_3^2}{y_1+y_2+y_3}, & k = 4 \\ \sqrt{\frac{1}{3}\sum_{l=1}^{3} y_l^2}, & k = 5 \\ \sqrt[3]{\frac{1}{3}\sum_{l=1}^{3} y_l^3}, & k = 6 \end{cases} \tag{13}$$

, where c is the coordinate of the previous gestures.

d. Creates three ADs for each $predicted_j$ coordinate. It creates a very strict acceptance domain ($predictedc_j \pm 0.05\ m$), a medium strict one ($predictedc_j \pm 0.10\ m$) and the least strict one ($predictedc_j \pm 0.15\ m$).
e. The algorithm does the previous steps for the remaining two axes (y and z).
f. Calculates the percentage of the coordinates inside all three acceptance domains on all axes and evaluates whether the gesture is accepted.
g. Waits until the user does a following gesture, then the algorithm pulls the earliest movement descriptor from the stack and starts over again with the remaining ones.
h. The algorithm runs until the user turns it off.

By doing these steps, a possible next gesture of the user and its acceptance domains are created. As substep 3/e stated, the algorithm generated not only the possible movement descriptor of the user but their acceptance domains as well. See Figure 2 for the sequence diagram of the APBMR algorithm.

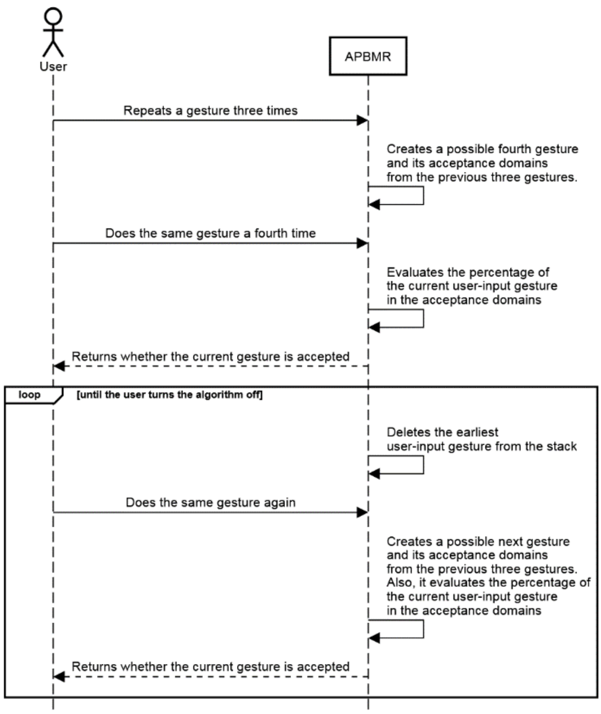

Figure 2

Sequence diagram of the APBMR algorithm.

In Figure 3, the previously mentioned acceptance domains are represented with six thin blue lines, while the original gesture is drawn with a black line and the predicted movement with an orange line. The acceptance domains are generated around the predicted movement descriptors.

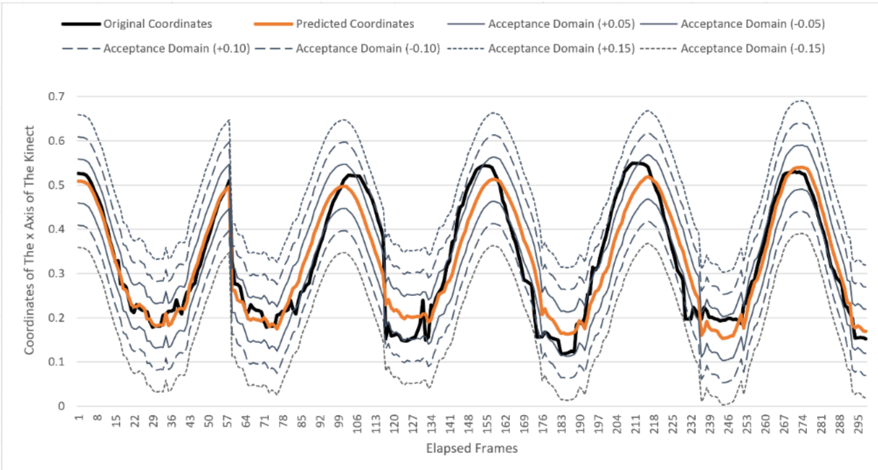

Figure 3
Graphical representation of the APBMR algorithm on the x axis.

If the reader looks at Figure 3, it could be observed that the acceptance domains and the predicted movement descriptors share the same shape on the x axis. The goal for the original movement is to stay inside these acceptance domains in each frame. Since all frames are evaluated, the gesture is accepted if it is inside these acceptance domains. This means at least 50% of the number of frames of a movement. If a gesture is accepted inside the strictest acceptance domain, that means that the APMBR algorithm can very accurately predict and classify the gesture of the user based on the previous three movements. Figure 3 tells us that for the strictest acceptance domain, 70% of the gestures are inside. For the medium strict one, 98.65% are inside, and for the least strict, 100% are inside.

Keep in mind, that the algorithm predicts gestures on one axis at a time, therefore before deciding on an accepted gesture, the algorithm has to be run on all three axes. Fortunately, the algorithm is very fast and this does not result in a problem – even in real-time.

The strength of this algorithm is that it can accept gestures when done in another position than the previous gestures. By default, this a problem with the Kinect, as it has a built-in 3D coordinate system and it only accepts those gestures that are done in the same position as the previous gestures. Since the APBMR algorithm follows the movement of the user, it predicts the position where the following gesture will be made. See Figure 4.

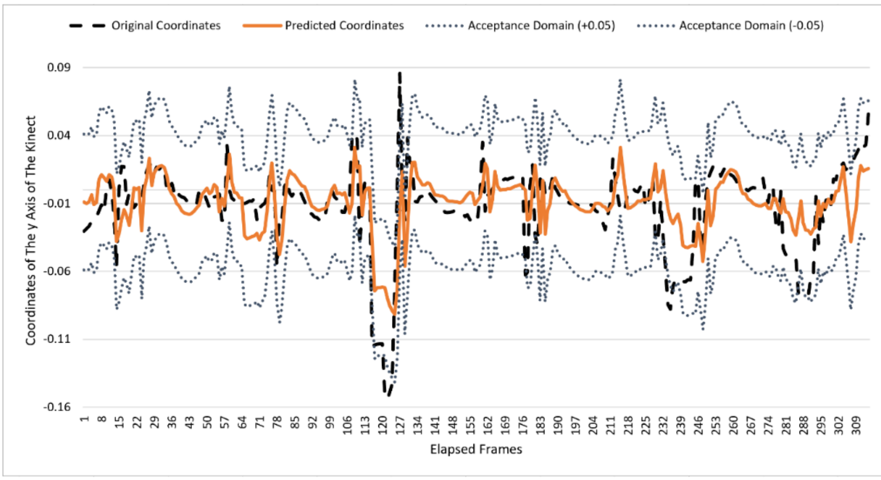

Figure 4
Graphical representation of the changing position of the right hand on the Kinect's y axis.

Another strength of this algorithm is that it is asynchronous, meaning that when the movements are done with a different speed (thus, having fewer frames), they are accepted as well. See Figure 5 for graphical representation. In the figure, the first three gestures are done with "normal" speed, while the next six are done faster and the last few are slower. Only the strictest acceptance domain is shown in the figure.

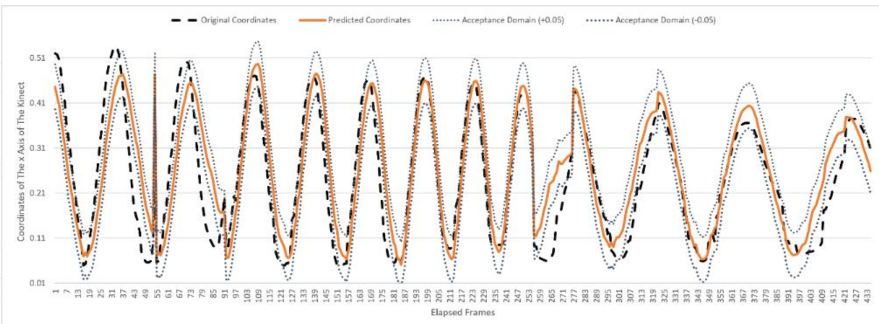

Figure 5
Graphical representation of the changing movement speed.

2.3 Data collection

Data collection was done in the second half of 2019 at the University of Pannonia. Four groups of people tested the algorithm. Out of these four groups, two groups measured in real-time, while the data of the other two was logged in a file. This algorithm was evaluated using two different computers. The authors will refer to these computers as General or Advanced. Their specifications are the following:

- General: Intel Core i7-720QM 1.60GHz, 6GB DDR3 1333MHz, ATI Mobility Radeon HD 5850 1GB.
- Advanced: Intel Core i9-9900K 3.60GHz, HyperX 32GB Predator DDR4 3200MHz, ASUS ROG Strix GeForce RTX 2080 8GB GDDR6 SUPER.

Also, the speed of the algorithm when predicting from a file was also assessed. As the APBMR algorithm uses a similar principle to one of the earlier algorithms of the authors, they were compared as well. Table 1 presents the data-gathering phase.

Table 1

Data collection.

Computer	People	Gestures	Repetition	Evaluation	Algorithm
Advanced	16	4	10	File	APBMR
Advanced	32	4	10	Real-time	APBMR
General	32	4	10	Real-time	APBMR
Advanced	32	4	10	File	APBMR and RDSMR/RDAMR

As can be seen, there are 16*4*10+32*4*10+32*4*10+32*4*10=4480 cases to evaluate the accuracy and speed of the APBMR algorithm. However, there are six different mean techniques and each was measured, therefore total number of cases is 4480*6=26880. This means that 32*4*10*2*6=15360 cases were evaluated in real-time and 16*4*10*6+32*4*10*6=11520 cases were evaluated from a file.

It should be noted that in the second and the third row, the users who tested the algorithm were the same. Also, the measured gestures were the same in each row: A circular movement, a waving movement, a diagonal movement forwards and a diagonal movement upwards. Lastly, in the fourth row of Table 1, it can be seen that both algorithms were assessed. Here, the testers recorded gesture descriptors and the data were saved in a file, since it was critical to assess the same coordinates of the gestures. Therefore, in the case of the last row, both algorithms loaded the data from the mentioned file and evaluated it.

3 Results

This section is broken into four subsections. Subsection 3.1 presents the real-time results of both computers when evaluating the algorithm. Subsection 3.2 deals with the results when predicting movement descriptors from a file. Next, subsection 3.3 compares the APBMR algorithm to the old algorithm of the authors. Lastly, subsection 3.4 evaluates all results of the APBMR algorithm, by taking every previous data into account.

Also, from the next subsection onwards, abbreviations are used instead of the frequently occurring words or phrases. These are the following:

- Average Gesture Acceptance Rate (AGAR)
- Acceptance Domain (AD)
- Arithmetic Mean Technique (AMT)
- Geometric Mean Technique (GMT)
- Harmonic Mean Technique (HMT)
- Contraharmonic Mean Technique (CHMT)
- Quadratic Mean Technique (QMT)
- Cubic Mean Technique (CMT)

3.1 Real-time results

First, the results received with the general computer were investigated (Figure 6). In the case of circular gestures, the CHMT gives the best AGAR for the strictest AD (26.95%), the HMT gives the best AGAR for the medium strict AD (64%) and the Arithmetic mean gives the best AGAR for the least strict AD (87.1%). In the case of waving gestures, the HMT gives the best AGAR for both the strictest and medium strict ADs: 76.1% and 95.7%, respectively. For the least strict one, the CHMT gives the optimal results with an AGAR of 97.265%. In case of the forward-diagonal gestures, the CHMT gives the best AGAR for both the strictest and the medium strict ADs, which are 84.765% and 99.218% respectively. For the least strict AD, the HMT gives the best AGAR of 100%. This means that the HMT accepted every forward-diagonal gesture done by the users. Lastly, in case of the upward-diagonal gesture, the CHMT gives the best AGAR of 19.921% for the strictest AD. In the medium strict and the least strict ADs, the AMT gives the best AGARs of 52.343% and 75.39%, respectively.

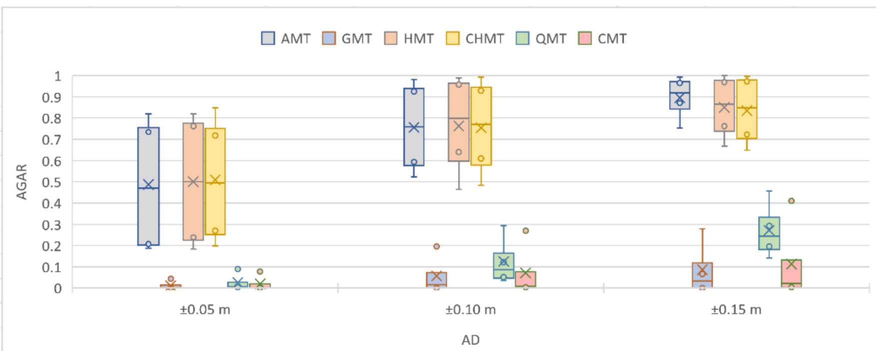

Figure 6

Results received with the general computer (real-time).

Secondly, the results received with the advanced computer were investigated (Figure 7). When circular gestures are assessed, the CHMT provides the best average acceptation rate of the strictest AD with 23.437%. For the medium strict and the least strict AD, the QMT provides the best average acceptation rates: 73.046% and 98.437%, respectively. After evaluating the waving gesture using the advanced computer, the results show that the CHMT provides the best AGAR of the strictest AD with 79.296%. Meanwhile, the AMT has the best AGAR of 96.875% medium strict AD. For the least strict AD, the HMT gives the best AGAR with 99.609%. In case of the forward-diagonal gestures, the CHMT provides the best AGAR using the strictest (65.625%) and the medium strict (92.187%) ADs. The AMT yields the best AGAR for the least strict AD with 96.093%. Lastly, in case of upward-diagonal gestures, the CHMT gives the best AGAR for all ADs: 48.828% for the strictest AD, 89.843% for the medium strict and 94.921% for the least strict one. It should be noted that for the least strict AD, the HMT has the same AGAR as the CHMT.

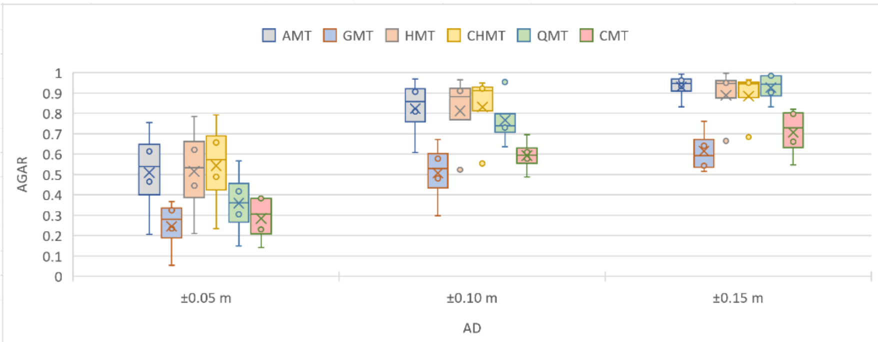

Figure 7

Results received with the advanced computer (real-time).

Lastly, their speed was compared and that can be seen in Table 2. In the table, “1” refers to the circular movements, “2” to the waving gestures, “3” to the forward-diagonal movements and “4” to the upward-diagonal gestures. Also, “G” refers to the general, while “A” to the advanced computer.

Table 2

Comparisons between the averages of time (ms).

Mean Technique	G1	G2	G3	G4	A1	A2	A3	A4
AMT	3.047	3.048	1.466	1.770	0.603	0.549	0.485	0.451
GMT	1.672	1.964	0.918	1.023	0.383	0.339	0.304	0.271
HMT	1.193	1.878	0.825	0.896	0.355	0.323	0.265	0.245
CHMT	1.574	3.528	0.924	1.221	0.453	0.432	0.325	0.305
QMT	1.525	2.983	0.900	1.164	0.410	0.392	0.308	0.297
CMT	0.985	3.199	0.955	1.323	0.272	0.413	0.333	0.314

Naturally, the advanced computer performs the task faster. However, it can also be seen that the APBMR algorithm on the general computer is not slow either. This is good as this fact adds to the possibility of using this algorithm in a home environment.

3.2 File-based results

In this subsection the file-based results are evaluated (Figure 8). Only the advanced computer was used in this regard. The first gesture to be evaluated was the circular movement: the CHMT gave the best average accepted ratio in the strictest AD with 37.5%, while the AMT gave the best average accepted ratio in the medium strict AD with 64.322%. The QMT gave the best average accepted ratio in the least strict AD with 84.895%. The following gesture to be evaluated was the waving gesture: the HMT gives the best AGAR in the strictest AD with 62.934% and the CHMT gives the best AGARs in all other ADs: 91.545% and 97.118%, respectively. It should be noted that the HMT and CMT returns the same results as the CHMT in the least strict AD. Next, the forward-diagonal gesture was investigated: in this case the CHMT gives the best AGAR in the strictest AD with 72.135%. Similarly, in the medium strict AD, also the CHMT gives the best AGAR with 88.281%. For the least strict AD, the AMT gives the best AGAR of 94.791%. Finally, the upward-diagonal gesture was examined: the CHMT gives the best AGARs in the strictest and medium strict ADs with 45.520% and 76.562%, respectively. In contrast, the HMT gives the best AGAR in the least strict AD with 86.718%.

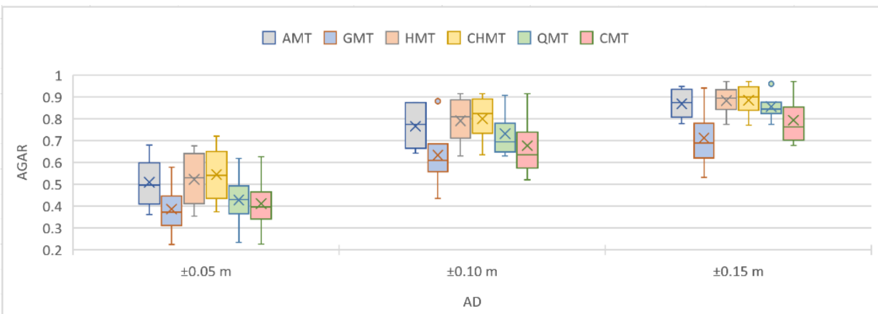

Figure 8
Results received with the advanced computer (file-based).

3.3 Comparison to the RDSMR/RDAMR algorithms

This subsubsection shows the comparison between the APBMR and the RDSMR/RDAMR algorithms. However, the authors omitted the RDSMR from the comparison. With it, the elapsed time between two movement descriptors can influence the results, in contrast to the RDAMR where it does not: By definition, the RDAMR can provide a better average of accepted movements than its synchronous counterpart.

The RDAMR works similarly to the APBMR: it creates the same ADs as the APBMR, but it only uses the first three gestures for their creation. Therefore, the ADs do not change during the gesture recognition. While the RDAMR works and can be used, the not-changing ADs could be a problem later on, since the algorithm does not follow the speed and the position of the user. If one of these factor changes, the algorithm would not accept the gesture, even if its shape is the same. Another difference between the algorithms is that the RDAMR evaluates whether the gesture is accepted during the time the user does the movements, while the APBMR evaluates it immediately after one is finished.

When comparing these two algorithms, the authors analyzed the AGARs of each mean technique (APBMR) and the RDAMR algorithm. Also, the authors only tested the APBMR in three ADs (±0.05 m, ±0.10 m and ±0.15 m). It quickly became apparent that the APBMR returned improved results than the RDAMR. Therefore, the authors increased the ADs when using the RDAMR algorithm until it gave similar AGARs as the APBMR. It should be noted that their execution times could not be compared, as the APBMR evaluates after the gesture is done, while the RDAMR evaluates during the movement in each frame. The results of the comparison can be seen in Figure 9.

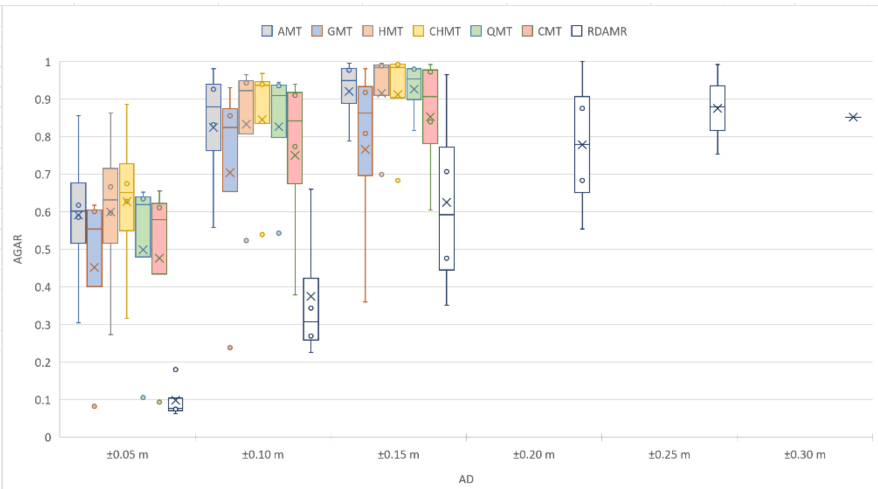

Figure 9

Comparing the mean techniques of the APBMR to the RDAMR algorithm.

Similarly, to before, the circular gesture was the first to be compared. Better AGARs are provided by all MTs of the APBMR than the by the use of the RDAMR algorithm. The difference between the AGARs of the two algorithms is very high in the cases of the AMT, HMT and CHMT. The AGARs when the GMT, QMT and the CMT are used are quite similar. Therefore, the APBMR is superior to the RDAMR in case of the circular gestures. Although, the results are more interesting in the case of the waving gesture: the difference of the AGARs of the RDAMR between the ±0.05 m and ±0.10 m ADs is quite large. Also, the AGAR of the RDAMR in the ±0.10 m AD (66.0%) is similar to the AGARs of the APBMR algorithm in the ±0.05 m AD (61.7% - 67.5%, depending on the used mean technique). Contrarily, in the case of the circular gesture, the ADs of the RDAMR are needed to be increased to ±0.15 m to have the same AGARs as the APBMR in the ±0.05 m AD. In the case of the forward-diagonal gesture, worse AGARs are returned by the RDAMR than in the case of the waving gesture. An AGAR of 88.7% is provided by APBMR with the use of the CHMT in the ±0.05 m AD which is the optimal mean technique to be used in this case. Meanwhile, a similar AGAR is provided by the RDAMR with 87.5% in the ±0.20 m AD, which is quite a large AD. Similarly to the forward-diagonal movement, the results regarding the upward-diagonal gesture are alike to it. In the ±0.20 m AD, an AGAR of 68.4% can be reached with the RDAMR algorithm which is slightly better than the ones in the ±0.05 m AD using the APBMR algorithm. The AGARs of the latter are between 58.6%-62.8% depending on the mean technique used.

Although, as can be suspected during the comparison, superior AGARs are provided by the APBMR. In the ±0.05 m AD, the increase of AGARs is between 358.2%-535.3% depending on the mean technique used, while in the ±0.10 m and ±0.15 m ADs it is 87.8%-125.4% and 22.7%-47.3%, respectively.

5 Discussion

According to the presented results, the APBMR algorithm also works well in real-time, and different mean techniques give the best AGARs in case of each gesture and ADs: in the ±0.05 m AD, the CHMT presents the best AGARs in three out of four gestures, while the remaining one gives the best AGAR with the HMT. In the ±0.1 m AD, the CHMT presents the best AGARs in two out of four gestures, while one of the remaining two gives the best AGAR with the AMT and the other with the HMT. In the ±0.15 m AD, the HMT presents the best AGARs in two out of four gestures, while the other two movements give the best AGARs with the AMT. However, the GMT gives the worst average acceptance rates in case of each gesture: for the circular movements, the AGARs of the GMT are the following: 11.272%, 28.013% and 39.397%, which are quite bad compared to the other mean techniques. For the waving gestures, the AGARs are: 34.709%, 55.133% and 59.486%. For the forward-diagonal gestures, the AGARs are the following: 21.316%, 39.397% and 43.415%. Last, but not least, for the upward-diagonal gesture, the AGARs are: 28.995%, 51.339% and 60.825%.

Based on the results, the GMT, QMT and CMT should not be used for predictive-based gesture recognition as the AMT, HMT and CHMT provide better results. From the strictest AD to the least strict, the former has AGARs of 24.073%, 43.470% and 50.781%, while the middle has AGARs of 29.334%, 56.848% and 70.896%; and the latter has AGARs of 26.317%, 47.990% and 57.435%.

It can also be concluded that when using the APBMR algorithm for prediction-based motion analysis, the CHMT presents the optimal AGARs in both the ±0.05 m and ±0.1 m ADs with 53.392% and 79.562%, respectively. For the ±0.15 m AD, the use of the AMT results in the optimal AGAR of 89.620%.

Different numerical differences exist between the AGARs in case of each gesture: With the circular movements, the numerical differences are between 0.28125 - 0.58371. In the case of the waving gestures, the numerical differences are between 0.22686 - 0.32187. With the forward-diagonal gesture, the numerical differences are between 0.21652 - 0.35714, while the numerical differences are between 0.3183 - 0.46897 with the upward-diagonal gesture.

As was mentioned previously, the APBMR evaluates each axis. However, during these evaluations, it can be observed that numerical differences exist between the AGARs on each axis: on the x axis, the numerical differences are between 0.17383 - 0.20117, while the numerical differences are between 0.19076 - 0.34147 and 0.06803 - 0.07129 on the y and z axes, respectively.

Lastly, when comparing the AMPBR to the RDAMR, the following can be concluded: in the ±0.05 m AD, the APBMR algorithm has an AGARs between 45.195%-62.656% depending on the used mean technique, while the RDAMR algorithm only has an AGAR of 9.863%. In the ±0.1 m AD, the APBMR algorithm provides AGARs between 70.410%-84.524% depending on the used mean technique, while the RDAMR algorithm only provides an AGAR of 37.5%. In the ±0.15 m AD, the use of the APBMR algorithm produces AGARs between 76.660%-92.571% depending on the used mean technique, while the RDAMR algorithm only produces an AGAR of 62.5%. Meanwhile, in the ±0.25 m AD, the AGAR of the RDMAR algorithm reaches a percentage that is similar to the AGAR of the APBMR in the ±0.15 m AD.

Conclusions

The authors of this article proposed a new method, called the Asynchronous Prediction-Based Movement Recognition algorithm for physical rehabilitation using sensors. While this algorithm was developed for and tested with the Kinects at the laboratory, it can be used with any sensor that returns coordinates in real-time.

The APBMR algorithm predicts the next gesture of the users from the previous three by using six different mean techniques and decides whether the next user-input movement is accepted. By doing so, it can follow the speed and the position of the user, making the decision to accept the next gesture easier.

Eight research questions and eight hypotheses were made at the beginning of the research. After a thorough analysis, eight theses were formed. The most important is to get the optimal AGARs, the mean technique to use differs from gesture to gesture as well as from AD to AD, while the mean techniques not to use are the same in each case. According to these results, the AMT, HMT and CHMT should be used for prediction-based gesture recognition, but these mean techniques should be changed depending on the gesture and the AD. See Table 3 which shows which mean technique to use in case of different gestures and ADs when evaluating on all three axes.

Table 3
Which mean technique to use in case of different gestures and ADs when evaluating on all three axes?

AD	Circular	Waving	Forward-diagonal	Upward-diagonal
±0.05 m	CHMT (0.304)	HMT (0.711)	CHMT (0.738)	CHMT (0.391)
±0.1 m	AMT (0.619)	HMT (0.941)	CHMT (0.925)	CHMT (0.723)
±0.15 m	AMT (0.837)	HMT (0.977)	AMT (0.964)	HMT (0.833)

When evaluating the gestures on all three axes, it can be concluded that when doing a simpler gesture where most of the time only one axis is necessary, such as the waving movement, the HMT provides the best AGARs. In all other cases, where the gestures move on more axes, the best AGARs are with the CHMT and the AMT in most cases.

However, when taking the authors' whole database of gestures into consideration, the CHMT gives the optimal average of accepted gestures out of all six mean techniques in the ±0.05m and ±0.1m ADs with 53.392% and 79.562%, respectively. Meanwhile, in the ±0.15m AD, the AMT provides the optimal average of accepted gestures with 89.620%.

Keep in mind, that the measurements were done with the Kinect which has its own coordinate system with positive and negative values. Due to the possible negative values, the GMT, QMT and CMT gave worse results. It is possible that with other sensors – that do not return negative coordinate values – or with some shift in the returned coordinates of the Kinect, they may provide better results.

In conclusion, the prediction-based gesture recognition method is more accurate than the older (RDAMR) algorithm of the authors. The APBMR algorithm can adapt to the current capabilities of the user, which is a criterion for maintaining motivation in the patients and for successful physical rehabilitation. Since the older RDAMR algorithm could be used at home, the APBMR algorithm can be as well, making the rehabilitation process easier for both the therapist and the patient.

Acknowledgement

The authors would like to thank Balazs Havasi for his help in the development of the WPF application to make the tests easier for the users. Project No. 131501 has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the K_19 funding scheme. A.M. was supported by the Janos Bolyai Research Scholarship of the Hungarian Academy of Sciences. Attila Magyar was supported by the ÚNKP-21-5 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund.

References

[1] Yang, Q., et al.: Vital signs: Recent trends in stroke death rates - United States, 2000-2015. MMWR. Morbidity and mortality weekly report, vol. 66, no. 35, 2017, p. 933. https://doi.org/10.15585/mmwr.mm6635e1

[2] Benjamin, E.J., et al.: Heart Disease and Stroke Statistics'2017 Update: A Report from the American Heart Association. Circulation, vol. 135, no. 10, 2017, pp. e146–e603. https://doi.org/10.1161/CIR.0000000000000485

[3] Pietrzak, E., Cotea, C.: Pullman, S. Using commercial video games for upper limb stroke rehabilitation: Is this the way of the future? Topics in Stroke Rehabilitation, vol. 21, no. 2, 2014, pp. 152–162. https://doi.org/10.1310/tsr2102-152

[4] Johnson, D., Deterding, S., Kuhn, K.A., Staneva, A., Stoyanov, S., Hides, L.: Gamification for health and wellbeing: A systematic review of the literature. Internet Interventions, vol. 6, 2016, pp. 89–106. https://doi.org/10.1016/j.invent.2016.10.002

[5] Rothbaum, B.O., Hodges, L., Smith, S., Lee, J.H., Price, L.: A controlled study of virtual reality exposure therapy for the fear of flying. Journal of Consulting and Clinical Psychology, vol. 68, no. 6, 2000, p. 1020. https://doi.org/10.1037/0022-006X.68.6.1020

[6] Cameirão, M.S., Badia, S.B.I., Oller, E.D., Verschure, P.F.M.J: Neurorehabilitation using the virtual reality based Rehabilitation Gaming System: Methodology, design, psychometrics, usability and validation. Journal of NeuroEngineering and Rehabilitation, vol. 7, no. 1, 2010, pp. 1–14. https://doi.org/10.1186/1743-0003-7-48

[7] Henderson, A., Korner-Bitensky, N., Levin, M.: Virtual reality in stroke rehabilitation: A systematic review of its effectiveness for upper limb motor recovery. Topics in Stroke Rehabilitation, vol. 14, no. 2, 2007, pp. 52–61. https://doi.org/10.1310/tsr1402-52

[8] Kim, J.H., Jang, S.H., Kim, C.S., Jung, J.H., You, J.H.: Use of virtual reality to enhance balance and ambulation in chronic stroke: A double-blind, randomized controlled study. American Journal of Physical Medicine and Rehabilitation, vol. 88, no. 9, 2016, pp. 693–701. https://doi.org/10.1097/PHM.0b013e3181b33350

[9] Laver, K.E., Lange, B., George, S., Deutsch, J.E., Saposnik, G., Crotty, M.: Virtual reality for stroke rehabilitation. Cochrane Database Systematic Reviews, 2017, p. 11. https://doi.org/10.1002/14651858.CD008349.pub4

[10] Merians, A.S., Tunik, E., Adamovich, S.V.: Virtual reality to maximize function for hand and arm rehabilitation: Exploration of neural mechanisms. In Proceedings of the Studies in Health Technology and Informatics, vol. 145, 2009, p. 109. https://doi.org/10.3233/978-1-60750-018-6-109

[11] Moreira, M.C., De Amorim Lima, A.M., Ferraz, K.M., Benedetti Rodrigues, M.A.: Use of virtual reality in gait recovery among post stroke patients-a systematic literature review. Disability and Rehabilitation: Assistive Technology, vol. 8, no 5., 2013, pp. 357–362. https://doi.org/10.3109/17483107.2012.749428

[12] Park, Y.-H.; Lee, C.; Lee, B.-H. Clinical usefulness of the virtual reality-based postural control training on the gait ability in patients with stroke. Journal of Exercise Rehabilitation, vol. 9, no. 5, 2013, p. 489. https://doi.org/10.12965/jer.130066

[13] Rizzo, A.A.; Schultheis, M.; Kerns, K.A.; Mateer, C. Analysis of assets for virtual reality applications in neuropsychology. Neuropsychological Rehabilitation, vol. 14, no. 1-2, 2004, pp. 207–239. https://doi.org/10.1080/09602010343000183

[14] Slater, M., Steed, A., McCarthy, J., Maringelli, F.: The influence of body movement on subjective presence in virtual environments. Human Factors, vol. 40, no. 3, 1998, pp. 469–477. https://doi.org/10.1518/001872098779591368

[15] Ustinova, K.I., Leonard, W.A., Cassavaugh, N.D., Ingersoll, C.D.: Development of a 3D immersive videogame to improve arm-postural coordination in patients with TBI. Journal of NeuroEngineering and Rehabilitation, vol. 8, no. 1, 2011, pp. 1–11. https://doi.org/10.1186/1743-0003-8-61

[16] Lanyi, C.S., Szucs, V.: Motivating rehabilitation through competitive gaming. In Modern Stroke Rehabilitation Through e-Health-Based Entertainment, 2016, pp. 137–167. Springer, Cham. https://doi.org/10.1007/978-3-319-21293-7_5

[17] Pagliari, D., Pinto, L.: Calibration of Kinect for Xbox One and Comparison between the Two Generations of Microsoft Sensors. Sensors, vol. 15, 2015, p. 27569. https://doi.org/10.3390/s151127569

[18] Sik-Lányi, C., Szücs, V.: Games applied for therapy in stroke tele-rehabilitation. International Journal of Stroke, vol. 9, 2014, pp. 300–303.

[19] Szücs, V., Lanyi, C.S.: Abilities and limitations of assistive technologies in post-stroke therapy based on virtual/augmented reality. In Assistive Technology: From Research to practice, 12th European AAATE conference, 2013, pp. 1087–1091. IOS Press. https://doi.org/10.3233/978-1-61499-304-9-1087

[20] Baranyi, P., Csapo, A., Sallai, Gy.: Cognitive Infocommunications (CogInfoCom), Springer, 2015. https://doi.org/10.1007/978-3-319-19608-4

[21] Kovari, A. Human-machine communication from an industry 4.0 perspective and its relationship with education 4.0, New challenges and pedagogical innovations in VET & HE, 2018, pp. 637–647.

[22] Magdin, M., et al.: Automatic detection and classification of emotional states in virtual reality and standard environments (LCD): comparing valence and arousal of induced emotions. Virtual Reality, vol. 25, no. 1, 2021, pp. 1–13. https://doi.org/10.1007/s10055-021-00506-5

[23] Kovari, A.: Study of Algorithmic Problem-Solving and Executive Function, Acta Polytechnica Hungarica, vol. 17, no. 9, 2020, pp. 241–256. https://doi.org/10.12700/APH.17.9.2020.9.13

[24] Demeter, R. et al.: Importance of digital simulation in the competence development of engineers defining the society of the future, Civil Szemle, vol. 17, no. 2, 2020, pp. 89–101.

[25] Kövecsesné Gősi, V.: Supporting the learning of teaching students with digital tools. Journal of Applied Technical and Educational Sciences, vol. 10, no. 4, 2020, 105–124. https://doi.org/10.24368/jates.v10i4.208

[26] Gogh, E. et al.: Experience of Self-Efficacy Learning among Vocational Secondary School Students, Acta Polytechnica Hungarica, vol. 18, no. 1, 2021, pp. 101–119. https://doi.org/10.12700/APH.18.1.2021.1.7

[27] Szucs, V., Guzsvinecz, T., Magyar, A.: Movement Pattern Recognition in Physical Rehabilitation - Cognitive Motivation-based IT Method and Algorithms. Acta Polytechnica Hungarica, vol. 17, no. 2, 2020, pp. 211–235. https://doi.org/10.12700/APH.17.2.2020.2.12

[28] Guzsvinecz, T., Szucs, V., Sik-Lanyi, C. Suitability of the Kinect Sensor and Leap Motion Controller—A Literature Review. Sensors, vol. 19, 2019, p. 1072. https://doi.org/10.3390/s19051072

[29] Niu, J., Wang, X., Wang, D., Ran, L.: A Novel Method of Human Joint Prediction in an Occlusion Scene by Using Low-Cost Motion Capture Technique. Sensors, vol. 20, 2020, p. 1119. https://doi.org/10.3390/s20041119

[30] Mateo, F., Soria-Olivas, E., Carrasco, J.J., Bonanad, S., Querol, F., Pérez-Alenda, S.: HemoKinect: A Microsoft Kinect V2 Based Exergaming Software to Supervise Physical Exercise of Patients with Hemophilia. Sensors, vol. 18, 2018, p. 2439. https://doi.org/10.3390/s18082439