

Program for controlling the correctness of physical exercises

A.I. Vybornyi¹, V.L.Rozaliev¹, Yu.A. Orlova¹, A.V. Alekseev¹

¹Volgograd State Technical University, Lenina Avenue 28 ,Volgograd, Russia, 400005

Abstract. This document contains the description of the program for the control of the correct physical exercises performance implementation using Microsoft Kinect, the method of the comparison between live motions performed by the user and recorded motions, description of the testing of the program and also information about different approaches for gesture recognition.

Keywords: Physical exercises, Microsoft Kinect, motion comparison, production rule system, gesture recognition.

1. Introduction

Nowadays the automation is used in many areas, including sport and physical culture. Most of the modern consoles have motion sensors. It gives developers big opportunities. The application of the new technologies and methods may be interesting and popular among people who care about their health, but do not have time for the gym and gamers who are interested in new experience. It is possible that in the near future people will give up going to the gym and hiring personal trainers and will use a virtual coach instead, doing physicals exercises at home in front of their consoles.

Program that is being developed may be useful both for the people who are recovering from injuries and for the people who just don't have enough time to go to the gym. The main objective of the program is to compare a predetermined sequence of the human movement with actual human movement captured via Kinect. The program should allow user to train at home controlling the way exercises are performed by the user and reporting to user about the mistakes he makes.

2. Different approaches for gesture recognition using mskinet

2.1. Hidden Markov Models

The method described by Jonathan Hall uses a Markov chain or a Markov Model. It is a typical model for a stochastic sequence of a finite number of states. These states are defined based on observations or data and these observations are essential for gesture recognition. In this approach, the observation data used are sequential 3D points (x, y, z) of Joints. A physical gesture can be understood as a Markov chain where the true states of the model $S = s_1, s_2, s_3, \dots, s_N$ define the 3D position of Joints for each state. A gesture is recognized based on the states as well as the transition between these states. These states are hidden and hence this type of Markov model is called a Hidden Markov Model (HMM). At each state an output symbol $O = o_1, o_2, o_3, \dots, o_M$ is emitted with some probability, and one state transitions to another with some probability. The emission and transition probabilities are learned

while training the model with known gesture data and these values are stored in the emission and transition matrices. Each trained model can then be used to determine the probability with which a given gesture appears in test data. In the manner described above, trained HMMs can be used to recognize gestures.[4, 10]

2.2. *Gesture Service with Windows SDK*

Gesture Service for Kinect project considers gestures to be made up of parts. In our context, parts refer to key poses of an Exercise and gesture refers to a sequence of key poses or in other words the complete Exercise. Each part of a gesture is a specific movement that, when combined with other gesture parts, makes up the whole gesture. Recognizing gesture parts are not sufficient to recognize a gesture. This is due to the fact that transitions between gesture parts play a crucial role in a gesture. To incorporate transitions, the method considers three results that a gesture part can return when it checks to see if it has been completed or not. The state of the gesture part is set to "Fail" if the user moved in a way that was inconsistent with the gesture part. The state of the gesture part is set to "Succeed" if the user performed a part of the gesture correctly and the system will automatically check for the next part of the gesture. Finally, the state of the gesture part is set to "Pausing" if the user is transitioning to the next gesture part. It indicates that the user did not fail the gesture but did not perform the next part either. The overall system comprises of three classes, namely, gesture controller, gesture and gesture part. The method uses a Gesture Controller to control the transition between gesture parts and updates the state of the gesture part. [4, 8]

2.3. *Kinect space*

The Kinect Space provides a tool which allows everybody to record and automatically recognize customized gestures using the depth images and skeleton data as provided by the Kinect sensors. This method is very similar to the Hidden Markov Model as discussed before. The software observes and comprehends the user interaction by processing the skeleton of the user. The unique analysis routines allow to not only detect simple gestures such as pushing, clicking, forming a circle or waving, but also to recognize more complicated gestures as, for instance, used in dance performances or sign language. In addition it provides a visual feedback how good individual body parts resemble a given gesture. The system can be easily trained for recognizing a gesture without writing any code. [4]

2.4. *Kinect SDK Dynamic Time Warping (DTW)*

Gesture Recognition Kinect SDK Dynamic Time Warping Gesture Recognition project allows developers to include fast, reliable and highly customizable gesture recognition in Microsoft Kinect SDK C-sharp projects. It uses Dynamic time warping (DTW) algorithm for measuring similarity between two sequences which may vary in time or speed. It uses skeletal tracking but the drawback with this software is that it currently supports only 2D vectors and not 3D. The software includes a gesture recorder that records the user's skeleton and trains the system. The recognizer software then recognizes the gestures that have been trained by the user. [5, 6]

2.5. *Neural network*

Neural networks are also one of the most commonly used and effective methods of gesture recognition. This is reflected in various works and sources of different authors. [7, 9] In most cases, the latter have a "pattern" character (reference gesture / movement, with which a coincidence must occur). The models for the gesture recognition have been constructed by using ten different Neural Networks (NN), one for each gesture, which have been trained providing a set of feature sequences of the same gesture as positive examples and the remaining sequences of other gestures as negative examples. Each NN has an input layer of 480 nodes corresponding to the feature vectors V_i for 60 consecutive frames, an hidden layer of 100 nodes and an output layer of one node trained to produce 1 if the gesture is recognized and zero otherwise. The Backpropagation Learning algorithm has been applied and the best configuration of hidden nodes has been selected in an heuristic way after several experiments. At the end of the learning phase, in order to recognize a gesture a sequence of features is provided to all the 10 NNs and the one which returns the maximum value is considered the winning

gesture. This classification procedure gives a result also when a gesture does not belong to any of the ten classes. For this reason a threshold has been introduced in order to decide if the maximum answer among the NN outputs has to be assigned to the corresponding class or not. [7]

3. Analysis methods of the control of the correct physical exercises performance

To implement the program it was decided for now to use angle comparison method. It wasn't mentioned in the section before, but the method is described below. The implementation of the program for the control of the correct physical exercises performance was divided into two main phases: recording phase and comparison phase.

3.1. Recording phase

In the first phase we needed to record the exercise and save it to file for the further use as a movement with a perfect form.

A few approaches have been tried for the recording. Firstly, we were trying to save only the coordinates and types of the joints. But this approach was too inefficient. Another suggested method was serialization and in the end it was decided to use it.

Serialization is the process of converting an object into a stream of bytes in order to store the object or transmit it to memory, a database, or a file. Its main purpose is to save the state of an object in order to be able to recreate it when needed.

The object is serialized to a stream, which carries not just the data, but information about the object's type, such as its version, culture, and assembly name. From that stream, it can be stored in a database, a file, or memory. [1]

Thus, with the use of the serialization the collection of frames with the data about the skeleton is saved to the file. This approach is good because we don't need to divide skeleton data and take only particular parts of it. Instead, the collection of frames is saved and each frame contains the whole information about the skeleton including coordinates, joints, type of joints, positions, orientation etc.

After we got the file with the information about the exercise we needed to read this file and process the data. To do it we used deserialization (the reverse process to the serialization).

3.2. Comparison phase

Now that we have saved motion data about the exercise that will be considered as a standard, we need to compare the standard with the user motion.

The basic idea is to compare the angles of the joints in the recorded motion standard with angles of the joints in the user motion. So, everything comes down to the calculation of the angles between vectors and comparison of these angles. In our case parts of the human body may be considered as vectors. Thus we need to know the values of the angles of the joints in the recorded motion standard and in the user motion. These angles are calculated with the use of the same method. [2]

Method takes 4 arguments: skeleton data and 3 arguments which represent the types of the joints. For example, in addition to the skeleton data it could be three joint types - `JointType.ShoulderCenter`, `JointType.ShoulderLeft`, `JointType.ElbowLeft`. If we pass these arguments in our method the angle will be calculated in the left shoulder joint (`JointType.ShoulderLeft`). Combining the joints, we can get two vectors that have one common point, and this point will be the joint in which we are calculating the angle. Coordinates of vectors are obtained, vectors are normalized, cross product and dot product are calculated and then the angle between vectors is calculated using `Atan2` method. This method returns the angle whose tangent is the quotient of two specified numbers.

The angles of the joints in the user motion are calculated every frame. For the motion standard angles are calculated in advance and stored in list.

Because it's almost impossible to repeat the motion standard with 100 percent accuracy program takes into account little errors in the user motion. At this point the error is 15 degrees in the value of the angles. Also, program compares current frame of the user motion not only with the one frame of the motion standard but with the 16 closest frames (8 previous and 8 subsequent) because user may do exercise a little faster or a little slower than standard motion demands. Angles of the joints in the

recorded motion standard and angles of the joints in the user motion are compared in the loop. If the difference between angles is less than 15 degrees it means that user is doing motion correctly.

To calculate the accuracy of the particular repetition in a particular exercise using described above algorithm the result of the comparison (whether the motion has been done correctly or not) is saved and stored. And the results of the comparison are stored for every joint involved in the exercise. When motion standard "ends" (all the frames of the recorded motion has been played) the analysis of the results starts. For the analysis the percent of the exercise performance correctness is calculated. The amount of frames in which the motions of the user were correct is divided by the total amount of frames in this exercise. It is done for every joint involved in the exercise. And after this the conclusion is made about whether this repetition may be considered as the correct or not. If the percent of correctness for each joint is above 85 and the arithmetic mean of all the percents is above 90 then this repetition may be considered as a correctly performed.

To analyze whether the motion is done by the user correctly or not we also used production rule system. All the exercises were divided into groups by the joints that are used in the exercises. And the production rules were set for each group. The example of the production rules for one of the exercises (overhead squats) is described below.

The next joints are used in this exercise: Shoulders, Elbows, Spine, Hips, Knees.

Production rules for this exercise:

Rule 1: IF (exercise = jumping jack) OR (exercise = squats) OR (exercise = overhead squats) OR (exercise = hip raises) THEN (compare angles in shoulder joints = yes) AND (compare angles in elbow joints = yes) AND (compare angles in spine = yes) AND (compare angles in hips = yes) AND (compare angles in knees = yes)

Rule 2: IF (difference between angles in shoulder joints < 15) THEN (result of the shoulder joints comparison in the current frame = true)

Rule 3: IF (difference between angles in shoulder joints >= 15) THEN (result of the shoulder joints comparison in the current frame = false)

Rule 4: IF (difference between angles in elbow joints < 15) THEN (result of the elbow joints comparison in the current frame = true)

Rule 5: IF (difference between angles in elbow joints >= 15) THEN (result of the elbow joints comparison in the current frame = false)

Rule 6: IF (difference between angles in spine < 15) THEN (result of the comparison in spine in the current frame = true)

Rule 7: IF (difference between angles in spine >= 15) THEN (result of the comparison in spine in the current frame = false)

Rule 8: IF (difference between angles in hips < 15) THEN (result of the comparison in hips in the current frame = true)

Rule 9: IF (difference between angles in hips >= 15) THEN (result of the comparison in hips in the current frame = false)

Rule 10: IF (difference between angles in knee joints < 15) THEN (result of the knee joints comparison in the current frame = true)

Rule 11: IF (difference between angles in knee joints >= 15) THEN (result of the knee joints comparison in the current frame = false)

Rule 12: IF (percent of correctness in each joint >= 85) AND (average percent of correctness >= 90) THEN (repetition is counted = true)

Rule 13: IF (percent of correctness in each joint < 85) AND (average percent of correctness < 90) THEN (repetition is counted = false)

The screenshots of the program while user is doing the exercise are shown on the figures 1 and 2.



Figure 1. Exercise is performed correctly.



Figure 2. Exercise is performed incorrectly.

4. Sections, subsections and subsections

The program has been tested by 10 users and each of them tested all 10 exercises represented in the program. The results of the testing are shown in the table I. Columns with the numbers from 1 through 10 represent 10 users. If the user was able to do a few repetitions of the exercise in the table this exercise was marked with +. If the user wasn't able to do even one repetition of the exercise in the table this exercise was marked with -. If the user was able to do a few repetitions of the exercise with some difficulties in the table this exercise was marked with +-.

Table 1. Results of the testing.

Exercise	№1	№2	№3	№4	№5	№6	№7	№8	№9	№10
Hand raises	+	+	+	+	+	+	+	+	+	+
Elbow rotation	+	+	±	+	+	+	+	+	±	+
Military press	+	+	+	+	+	±	+	+	+	+
Jumping Jack	+	+	±	±	+	±	+	+	+	±
Side bend	+	±	±	±	±	±	+	±	+	±
Squats	+	±	±	±	±	±	±	±	+	±
Overhead squats	+	+	±	±	±	+	±	±	+	+
Thrusters	±	-	-	-	-	-	-	-	±	-
Side lunges	+	+	+	+	+	+	+	+	+	+
Hip raises	±	-	-	-	-	-	-	-	-	-

Thrusters and hip raises caused the biggest difficulties among users. It means that we need to make these exercises easier (for example, by decreasing the percent of correctness threshold value) or replace them with the other exercises.

It is also worth mentioning that most of the users needed some time to get used to the way the exercises should be performed. Perhaps, decreasing the percent of correctness threshold value of the exercises would be the good idea not only for those movement that caused some difficulties among users but also for the other exercises.

Summing up the results of the testing we may say that users handled the performance of most of the exercises quite well. Nevertheless, some of the exercises still need corrections or replacing.

5. Conclusion

This document contains the description of the program for the control of the correct physical exercises performance implementation using Microsoft Kinect.

The implementation of the program was divided into two main phases: recording phase and comparison phase.

In the first phase we had to record the human-motion and save it into a file for later processing. A few approaches (on how to read and save those data from tracked human skeleton) have been tried here. The successful approach that is used now is a serialization – saving the collection of skeleton frames into a data structure in binary format.

The second phase is a comparison between live motions performed by the user and recorded motions. The main idea is to calculate recorded motion's joint angles and user's joint angles, compare them, considering a little error and then with the use of the production rule system analyze performed exercise to know whether the motion was correct or not.

At the moment there are ten exercises represented in the program, involving different joints and muscle groups. Program gives user the feedback about the performance of the exercises by marking the joints in which user make mistakes with red colour. Also, for more detailed information about the accuracy of the repetition performance user can open the output file that contains the percents of the accuracy for every joint in every exercise in every repetition.

The further development of the program is planned for the future. The plan is to combine the current method of the comparison with the gesture recognition methods to increase efficiency and accuracy and also to include more exercises to the program.

6. Acknowledgments

The work is partially supported by the Russian Foundation for Basic Research (16-07-00407, 16-07-00453, 16-47-340320 projects).

7. References

- [1] MSDN - Microsoft Developer Network: Serialization (C and Visual Basic). - URL: <https://msdn.microsoft.com/ruru/library/ms233843.aspx>.
- [2] Hemed, A. Motion Comparison using Microsoft Kinect 2012: FIT3036 /A.Hemed // Computer Science Project. – 2012. – P. 27.
- [3] Orlova, Y.A. Automation of the control of the physical exercises performance for rehabilitation using Microsoft Kinect / Y.A. Orlova, V.L. Rozaliev, A.A. Shpirko // Physical education and sports training. – 2013. –Vol. 1. –P. 53-58.
- [4] Ravi, A. Automatic Gesture Recognition and Tracking System for Physiotherapy / A.Ravi // Electrical Engineering and Computer Sciences University of California at Berkeley, Technical Report No. UCB/EECS-2013-112. – 2013. – 30 p.
- [5] Codeplex - open source project hosting: Kinect SDK Dynamic Time Warping (DTW) Gesture Recognition, 2011.
- [6] DTW Algorithm: Mining of gene expression time series with dynamic time warping techniques.
- [7] D’Orazio, T. A Neural Network Approach for Human Gesture Recognition with a Kinect Sensor/ T. D’Orazio, C. Attolico, G. Cicirelli, C. Guaragnella //ICPRAM2014 – InternationalConferenceonPatternRecognitionApplicationsandMethods ISSIA - CNR and DEI. – 2014. –P. 741-746.
- [8] MSDN - Microsoft Developer Network: Gesture service for the kinect with the windows sdk / MCS UK Solution Development, 2011.
- [9] Tang, A.A Real-time Hand Posture Recognition System Using Deep Neural Networks / A.Tang, K. Lu, Y. Wang, J. Huang, H. Li, // ACM Transactions on Intelligent Systems and Technology. – 2013. –Vol. 9(4). – P. 39-62.
- [10] Zoubin,G. An Introduction to Hidden Markov Models and Bayesian Networks / G. Zoubin // International Journal of Pattern Recognition and Artificial Intelligence.– 2012. –25 p.