TALLINN UNIVERSITY OF TECHNOLOGY
School of Information technology
Department of Software Science

Jaroslav Kulikov 153147IAPM

Virtual Reality Aided Framework for Modeling Changes in Finger Motion and Orientation During Learning New Fine Motor Activity

Master thesis

Supervised by: Sven Nõmm, PhD
Prof. Aaro Toomela

Tallinn 2017

Author declaration

I declare that I have written this Master thesis independently and without the aid of unfair or unauthorized resources. Whenever content was taken directly or indirectly from other sources, this has been indicated and the source referenced.

Author: Jaroslav Kulikov

May 9, 2017

Abstract

This thesis presents a framework for detecting fine motor movement changes during learning new motor activity. The framework is implemented using virtual reality with partial immersion. Despite VR is similar to real world, user still needs training to get used to it. Additionally, virtual reality grants control over environment, enabling objective complexity changes. Another VR benefit is that sensor is not being eclipsed by exercise props.

Purposeful motor activities have already been a research subject for medicine and psychology. Knowledge about how individual is learning can help simulation and rehabilitation exercises evaluation.

This work narrows its' scope to palm and finger movement research. The Leap Motion controller enables detailed hand movement recording with reduced cost, thus it is chosen as a VR application controller.

A developed virtual reality application implements simple exercise, where user will be required to reposition figures. It also records user movement data. Hand orientation and kinematics parameters are calculated for recorded motions, such as grasping, replacing and releasing an object. Calculated parameter values in the beginning and ending of training are being compared using statistical hypothesis testing in order to detect changes, that occurred during training session.

Using comparison results, the application dynamic learning recognition has been developed. The main purpose of this module is objective difficulty increasing on the run depending on individual learning process.

The results have shown, that some parameter values do actually change during individual learning. Moreover, more significant changes can be detected if the objective was gradually becoming more difficult. Thus, changes in human motor movements have been successfully digitalized. Despite current results being exercise specific, motion parameter list may be widened to extent, that is suitable for any type of fine motor exercise characterization.

The thesis is in english and contains 49 pages of text, 5 chapters, 12 figures, 9 tables.

Annotatsioon

Antud töö esitab raamistiku, mille eesmärgiks on peenmotoorika muudatuste registreerimine ja analüüsimine uue motoorse tegevuse õppimise jooksul. Motoorsed oskused uuritakse psühholoogias, kus selle teemale pööratakse tähelepanu seoses laste arenguga. Spetsiifilised motoorsed oskused on vajalikud ka teistes erialades, näiteks kirurgia alal.

Uurimistöö läbiviimiseks on vaja rahuldada kolm olulist tingimust. Esimene tingimus on tagada vajadus õppima peenmotoorseid tegevusi. Täiskasvanute puhul see on eriti raske, arvestades seda et enamus peenmotoorseid tegevusi õpib inimene lapsepõlves. Teine tingimus seisneb selles et vältida reaalsete objektide kasutamist. Liigutuste registreerimissüsteemide puhul reaalsete objektide kasutamine põhjustab kaamerate varjutusi. Kolmas tingimus on võimalus registreerida sõrmede liigutusi ja asendid.

Virtuaal reaalsuse keskkond osalise kümblusega tagab esimese ja teise tingimuste täitmist. Virtuaal reaalsuse keskkonnas puudub vajadus kasutada reaalseid objekte, mis tihti põhjustavad kaamerate varjutusi. Samuti virtuaal reaalsuse keskkonnas on võimalik kiiresti muuta keskkonna parameetrid mille kaudu tagatakse vajadus kohandada oma liigutusi. Sõrmede liigutuste ja asendite registreerimiseks kasutab autor Leap Motion andurit, mille põhieelis seisneb selles et ta ei vaja keeruka seadistamisprotseduuri ja lisavarustust.

Ühe lihtsa harjutuse jaoks leiti parameetrid mille väärtused oluliselt erinevad edukate ja ebaõnnestunud katsete vahel. Samuti leiti parameetrid mille väärtused olulised muutuvad õppimisprotsesside jooksul. Selleks kasutati statistiliste hüpoteeside kontrollimine. Andmete kogumiseks ja analüüsiks arendati kaks rakendust.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 49 leheküljel, 5 peatükki, 12 joonist, 9 tabelit.

Acknowledgement

Partially supported by Research Funding Project of the Tallinn University of Technology B37.

The author would like to thank the group of undisclosed individuals, that helped to gather movement data.

List of abbreviations

LM	Leap Motion controller
VR	Virtual Reality
3D	Three Dimensional
PC	Personal Computer
MM	Motion Mass
PD	Parkinson's Disease
SDK	Software Development Kit
CSV	Comma Separated Values

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A virtual environment with partial immersion together with Leap Motion controller were used in this research to gather individual movement data while learning new fine motor activity. Collected data was analyzed afterwards with aim to detect motion changes during training session.

Purposeful motor activities have been a research subject for medicine, psychology and sports [2]. Those activities are divided to gross and fine motor movements. An example of fine motor activity lowest level is writing and drawing, while movements of separate fingers are on top level.

Learning new motor activity is done through a sequence of trials. Each trial is expected to improve the performance quality or an action outcome. Each motion is described with a set of speed, location and orientation parameters. Then the goal is to find motion characteristics, that differ significantly between the start and end of the training process.

Up to now testing fine motor activities was a routine for psychologists [2], the test evaluation is usually conducted by practitioner or nurse, but this approach has obvious drawbacks. Human naked eye can't notice every tiny motion detail. An evaluation without numeric scale is subjective. A numeric changes in fine motor activity provided by learning detection system can be used for evaluation of rehabilitation and surgeon training exercises [3] or diagnosis of diseases, that affect motion planning and execution [4]. Numeric measurements bring objectivity to exercise evaluation.

A key question is an exercise choice - an action, that will be learned by several individuals. All primitive actions were already learned during first years of childhood. We are also unaware of abilities, tested individuals may have, thus more complicated activities are not suitable neither. Moreover, a complicated processes may bring unnecessary complexity to this research. The problem may be solved by means of Virtual Reality (VR). Even if the simplest possible activity will be implemented, the virtual environment properties, for example object slipperiness and stickiness, may be slightly changed thus requiring user to adapt. "Build a tower" task is chosen as an exercise being modeled.

VR eliminates the necessity of any additional props for chosen exercise. Additionally, real objects would have blocked sensor line of sight decreasing the hand position recognition confidence level. However, we will avoid full immersion into virtual reality, because VR helmets may cause severe seasickness among children and they will vastly increase the hardware set cost.

The researches dedicated to gross motor function changes [5] modeling were already conducted and gave positive results, thus encouraging this work. A Kinect was used as a motion capturing tool. It shows great performance in reading limb movements, but do not detect fine motor motions. Another related research was conducted in Parkinson's disease diagnosis using tests implemented in tablet PC, where patient handwriting is recorded [6].

Since within the framework of this contribution the scope will be narrowed to detailed hand modeling, a proper tool, that is capable of capturing even such a tiny motions like finger movements, needs to be chosen. Leap Motion (LM) controller perfectly suits that role. It doesn't capture the whole body like Kinect does, but is optimized for hand movements tracking. It is cheap and compact too. "More expensive and complicated marker based systems do provide necessary abilities and precision but require one to construct model for each particular individual, which make it impossible to apply for medical research involving large number of participants" [7]. The experience of using LM for medicine purposes already exists [3] leaving no doubts about validity of chosen device.

Once data is gathered, it will be demonstrated, that user performance differs significantly in the beginning and in the ending of learning. To do so, the data will be split into two sets: *before* and *after*. We will also split data into *successful* and *failed* sets in order to show that action success depends on certain motion parameter difference. Those datasets will be processed using statistical hypothesis testing method.

Eventually, user may master an exercise. We need to avoid this undesired scenario, because the testee will stop learning. For this purpose the application is programmed to detect training progress. When user performance improvements are detected, an environment is automatically tuned to make exercise harder, thus suspend individual training. The continuous learning is achieved this way.

1.1 State of the art

An initial problem comes from the need to numerically model changes in human motor functions during learning new motor activities, thus allowing to precisely measure the exercise progress [5]. Learning new or improving existing motor activity is usually done through repetitive exercises. Each exercise may be presented as a sequence of trials. Each trial in its turn is split into motions, which are the target unit for measuring. Necessity of effectiveness digitalized tracking comes from rehabilitation and health monitoring [8] or sportive medicine fields [9]. Previous solutions were designed for specific problems, which led to precise results [10], but in order to develop universal modeling technique the generalization is needed.

[5] and [11] were dedicated to measuring motion parameters and modeling changes over training time. It was decided to use a Microsoft™Kinect™sensor for recording individual movements, because it allows to capture body position on limb level with adequate accuracy. There is a wide exercise choice for evaluation. For example, the

ball throwing task was implemented as computer application in [11]. Four *Motion Mass* (MM) parameters were presented. *Combined Euclidian distance* is a distance between starting and ending position of all joints. *Trajectory Mass* parameter a sum of trajectory lengths of all joint and describes the amount of movement made. *Acceleration Mass* can be defined as a sum of accelerations of each frame for each joint of interest. An acceleration describes the smoothness of the motion. A motion execution total time can be also useful. Due to physiological differences of people two more parameters are needed: ratio between *Combined Euclidean Distance* and *Trajectory Mass* and ratio between *Combined Euclidean Distance* and *Acceleration Mass*. Using those *Motion Mass* parameters any motion can be modeled independently from the matter of executed action. Depending on the actual test, some exercise success measurements can be added, such as amount of successful attempts. In [5] it was shown, that *Motion Mass* parameters differ significantly in the beginning and the end of exercises.

As a branch research, same comparison was made between human different arousal levels. Approach proposed in [12] "targets to relate measured values of the skin conductance to the observed locomotion amount of the human limbs". Were introduced a numeric measurements of motions by motion capture system instead of subjective human made observations. As a result of the research ability to estimate arousal level on the basis of measured parameters of the motion and vice versa was achieved. Also, with this knowledge the influence of the arousal level on the motions planning process may be researched.

Multilevel motion planning results into any purposeful movement [13]. On the first level, action purpose and execution way are decided. Second level is responsible for generating concrete motion patterns. On the third level the signal to action is sent to the spinal cord.

Lots of related research was addressed to Parkinson's disease (PD) diagnosis opportunities, which affects different levels of motion planning and execution [14], while on the other side, within the framework of this thesis, the functional system evolution will be modeled.

[15] describes the diagnostics of Parkinson's Disease using *Motion Mass* parameters of individuals performing the Up-and-Go test. To complete this test, the subject has to get up from a chair, walk forward, turn around and return to starting position [16]. An additional MM parameter was added: *Velocity Mass*, describing the speed of motion. It was shown, that there are distinguishable MM parameters of Parkinson disease patients compared to healthy individuals, thus "objective measures to characterize movements in PD patients in details would allow to assess progression of the disease" [15]. It was derived, that the *Acceleration Mass* has significant difference in most cases, which means, that "smoothness of the movement is the most affected characteristic in PD" [15]. Positive changes as a result of treatments can be detected as well.

The research on Parkinson disease diagnosis opportunities continues in [4] and [6]. While [15] concentrated on limb tracking, these articles describe an attempt to detect fine motor movements. The paper and pen test was implemented on tablet in both cases. The main benefit, which was achieved through digitalization, is an ability to measure stylus pressure, velocity and acceleration. Since the hand tightly holds the pen, its' position and pressure may be considered as hands' parameters. Based on these motion characteristics previously discussed Motion Mass parameters were estimated, so that PD patients' performance could be compared to healthy control group, together with comparison between motion characteristics of one individual over time.

The Luria's alternating series tests [13] were implemented in [4], which require a different complexity of motion planning, thus allow to distinguish disorders on the different levels related to motion planning and execution. This approach detects deviations on the second and the third motion planning levels.

In [6] the Poppelreuter's overlapping figures test [17] was digitalized. Test requires the testee to recognize the contours of overlapping figures. The problem of test digitalization lies in result validation, because of more meaningful shapes being drawn. One more worth to mention test is a Clock Drawing Test, where the test name doesn't need any additional explanations [18].

A Leap Motion (LM) controller was proposed as another device for fine motor movements detection in [7]. LM is a small controller that aims to replace physical controllers like mouse or keyboard. The controller can be embedded into most of existing virtual and augmented reality headsets or merely located on the table. Smaller detection range provides better resolution [19]. Tablets were widely used for hand writing motions research [20], but the ability to capture hand position itself gives a benefit to Leap Motion controller over tablet and stylus system, where palm location can only be deduced based on test performance. Only palm position and pressure can be extracted via stylus. This feature makes possible modeling not only arm as a whole, but also gives an ability to literally take hand apart and study each piece independently. Leap Motion controller was already used in applications for "motor coordination of children with developmental disorders" [21]. Children practiced exercises for fine and gross movements in virtual environment, which can be tuned for each individual needs and also enables tracking user performance. LM was also used for hand gesture recognition research [22] and in virtual reality based stroke recovery systems [23].

Objective learning assessment has gained lots of interest in recent years. In [24] was developed video processing method for tool detection and tracking for a simulated suture task. A surgeon exercise video was recorded and processed afterwards in order to track the needle driver used to grasp and guide the needle. Another experience of thoracoscopy surgeon skills training through virtual reality application was presented in [3]. Similarly to this thesis, the Leap Motion was chosen as a controller for simulator, because it benefits over other existing solutions. It is cheap, portable and enables to use real instruments during practice. Built application has three difficulty levels. The higher difficulty level, the smaller and finer are objects. Moreover, an application is capable of evaluating user performance.

[7] adds new parameters to MM category. Worth to mention parameters are *jerk*, the acceleration change rate [25], and diverse direction angles, which allow to address palm and fingers orientation. This thesis will use [7] as basis and develop it adding new training and data tracking features.

1.2 Formal problem statement

This research goal is to develop a virtual reality framework for detecting fine motor movement changes during learning new motor activity. For this reason, a simple motor exercise has to be implemented in a 3D application, that records user movements.

Two groups of untrained individuals will participate in research. The first group of ten testees has to complete the exercise once. Their successful and failed motions will be compared afterwards. The participants of second group will complete a training course of 20 trials. Motions in the beginning and ending of learning will be compared for each individual.

1.2.1 VR application functional requirements

In order to meet research needs, this features have to be implemented by application:

- 3D graphical interface.
- An ability to grasp and release objects has to be implemented using Leap Motion controller.
- Application must plot user hands inside game environment.
- “Build a tower“ exercise interface and evaluation has to be implemented.
- Application has to support several difficulty levels of an exercise.
- An exercise props must be restricted outside from Leap Motion line of sight.
- Application has to record the following information for each trial:
  - Events of hand interactions with objects.
  - The records or exercise progress.
  - Each hand joint position.
  - Interactive objects position and orientation.
- User motion basic parameters' value convergence and change trend has to be detected on the run.
- Based on user performance improvement a difficulty level has to increase automatically for the next trial.

1.2.2 Recorded data analysis

A user training session motion data has to be extracted from collected dataset. Analysis module must classify motions as successful or failed and calculate parameters, such as hand movement and orientation characteristics.

The working hypothesis of the present research is that there exist four subsets of motion parameters. The first set contains parameters, which values differ between successful and failed motions. Remaining three subsets comprise parameters, which values vary between beginning and the ending of training session in static, regularly increasing difficulty and complexity level, that depends on user performance, environments.

Formally, problem may be stated as follows. If θ is a set of parameters, that describe kinematics and position of hand and finger motions, then using method of statistical hypothesis testing find:

1. The subset of parameters $\theta_s \in \theta$, which values differ significantly between successful and failed motions.

2. The subset of parameters $\theta_{lstat} \in \theta$, which values differ significantly between the beginning and ending of learning process in static environment for successful movements.

3. The subset of parameters $\theta_{lreg} \in \theta$, which values differ significantly between the beginning and ending of learning process in environment with regularly increasing difficulty for successful movements.

4. The subset of parameters $\theta_{ldyn} \in \theta$, which values differ significantly between the beginning and ending of learning process in environment, where objective difficulty depends on user motion characteristics, for successful movements.

1.3 Outline

Method chapter

- Description of the application, that implements exercise and records movement data in section 2.1.

- Extraction of grasp, replacement and release motions from recorded data in section 2.2.

- List of parameters being calculated for extracted motions in section 2.3.

- Division of parameter values into samples for different comparison cases in section 2.4.

- Description of application module, which is responsible for user learning recognition on the run in section 2.5.

Analysis chapter This chapter will introduce and summarize parameters, that were detected to be different for two samples being compared.

Discussion chapter Work results together with future research goals will be discussed in fourth chapter.

Conclusion Conclusions are drawn in the final chapter.

Chapter 2

Method

2.1 Application

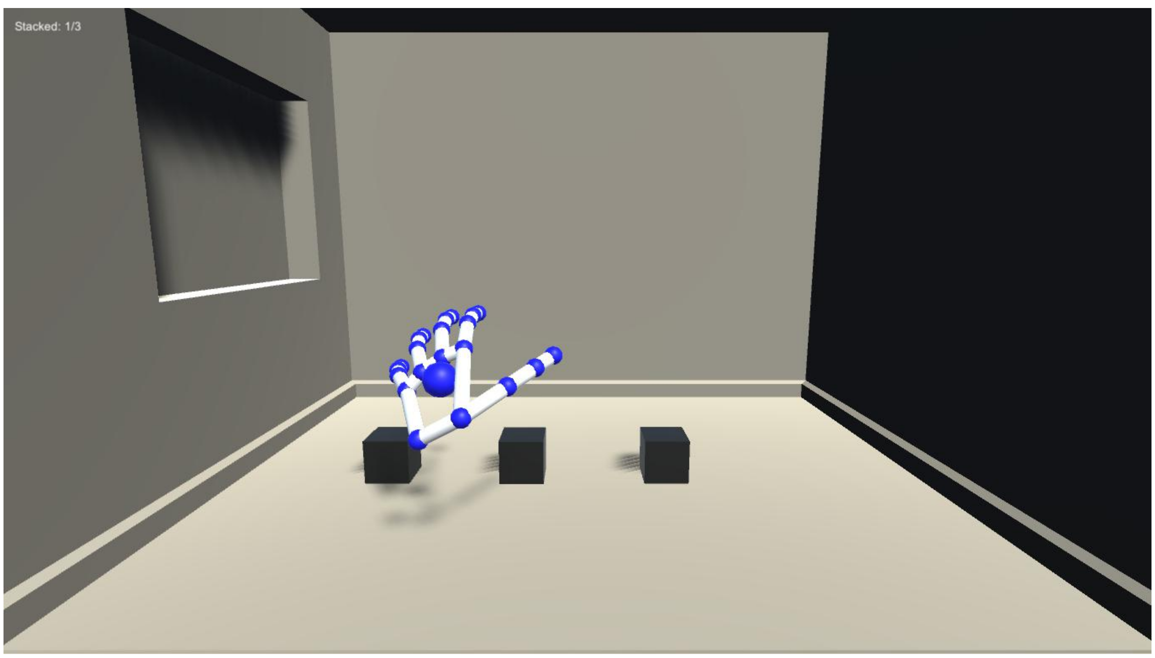

Figure 2.1: Application demo.

Frameworks and libraries

- Unity - a game development platform.
- Unity Assets for Leap Motion Orion - a ready to use LM integration for Unity platform.

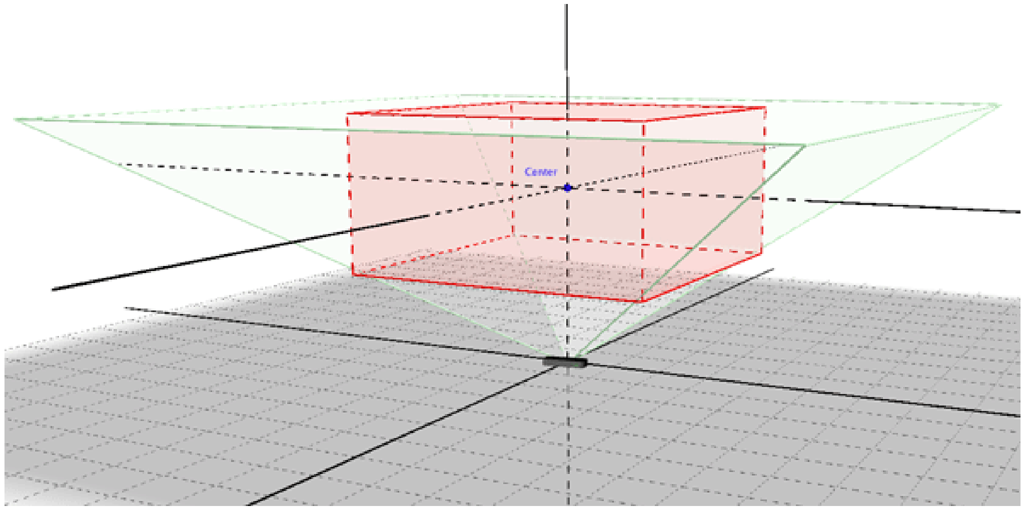

Figure 2.2: Interaction box [1].

2.1.1 Interface

Room Leap Motion is capable of capturing space with inverted pyramid shape in front of its' cameras, while it is more convenient to operate inside a rectangular box, thus a box has to be defined such, that it fits into pyramid, has adequate size and allows Leap Motion controller to show its' maximum capturing potential 2.2. The purpose if this box is to restrict objects from moving outside from LM field of view.

In order to make box more eye pleasing it was decided to give it a room view. Additionally, room plinths will not allow objects to stuck in the corner - the least visible part of the room for LM controller 2.1.

"Build a tower" exercise In order to complete an exercise, one is required to position cubes on top of each other 2.4. Program assumes, that one figure is on top of another, if both figures stay in contact and the lowest point of first is slightly higher, than the lowest point of another. Both objects must be in sleeping state, which means they weren't moving for a while, which makes it impossible to pass while hand is still interacting with any of the figures. The reader could notice, that two bottom points are compared instead of top and bottom. It was assumed, that eventually figures with non-standard shape may be used for this exercise, so that top and bottom comparison rule will not work any more.

2.1.2 Captured data

Everything, that is happening in the exercise room, has to be recorded, so that data can be processed afterwards. Every detail may be significant, because you never know beforehand, what motion parameters will show positive results. New record is made every frame with approximate frequency of 50 times per second.

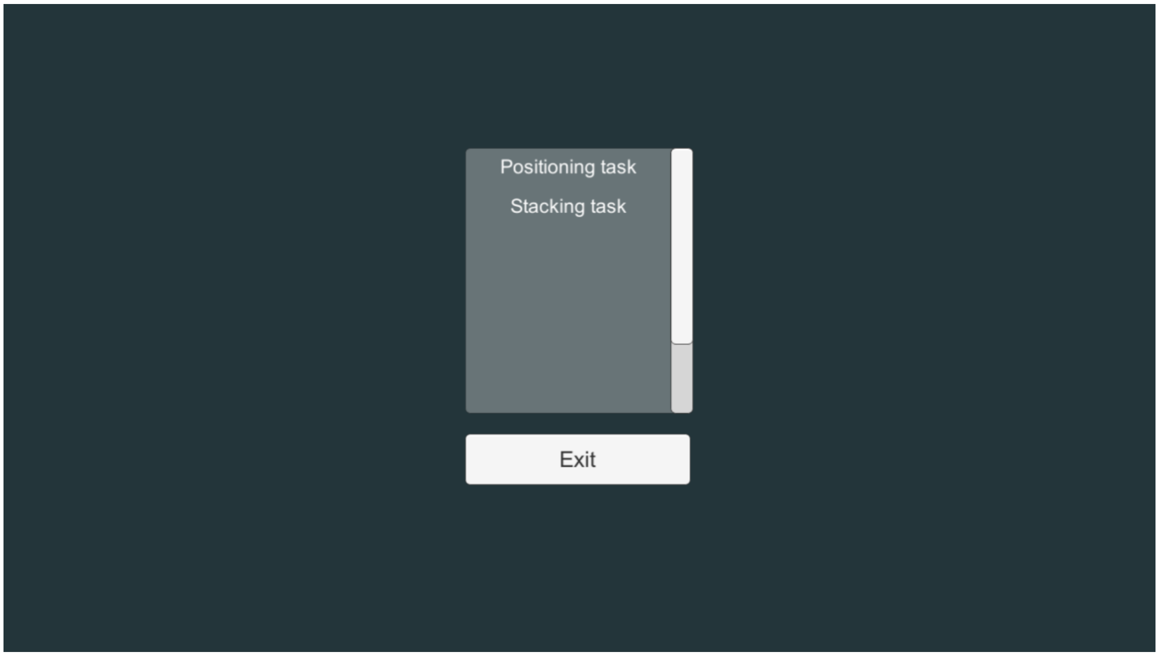

Figure 2.3: Application menu.

Figure 2.4: “Build a tower objective.

Coordinate system All coordinates are translated to Unity coordinate system by LM software 2.5.

Euler angles Most angles are written in Euler representation as pitch, yaw and roll. Pitch stands for object rotation around y, yaw - z and roll - x axes 2.6. Those three angles were "introduced by Leonhard Euler to describe the orientation of a rigid body with respect to a fixed coordinate system" [26]. Each angle describes rigid body rotation around one of the axes.

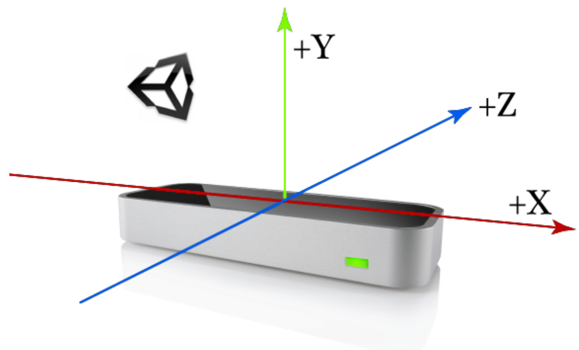

Figure 2.5: Unity axes [1].

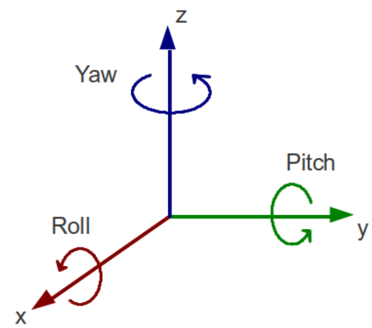

Figure 2.6: Euler angles.

Hand frame information Leap Motion SDK provides developer with detailed information. Hand object structure resembles real hand. User hands are described up to every bone position. Having each joint coordinates, hand position may be easily reconstructed later in details. Hand consist of five fingers: thumb, index, middle, ring and pinky. All fingers have four bones 2.7:

- Metacarpal - the bone inside the hand. Its' base is connected to wrist.
- Proximal Phalanx - the bone at the base of the finger.
- Intermediate Phalanx - the middle bone of the finger.
- Distal Phalanx - the bone at the tip of the finger.

Additionally to raw coordinates, Hand class provides developer with more abstract data regarding hand posture [27]. Together with raw coordinates, those parameters constitute one CSV file per each hand. We could make out with only joint coordinates, but additional parameters will simplify calculations. For each exercise application outputs two CSV files for *left* and *right* hand. Actually, during execution of an exercise, may appear more hands with different identifiers, because Leap Motion may eventually lose the trace of the hand and once trace is found again, new hand is identified. Software assumes, that only one individual is completing an exercise, so it writes each hand data to corresponding file without binding it to particular hand id.

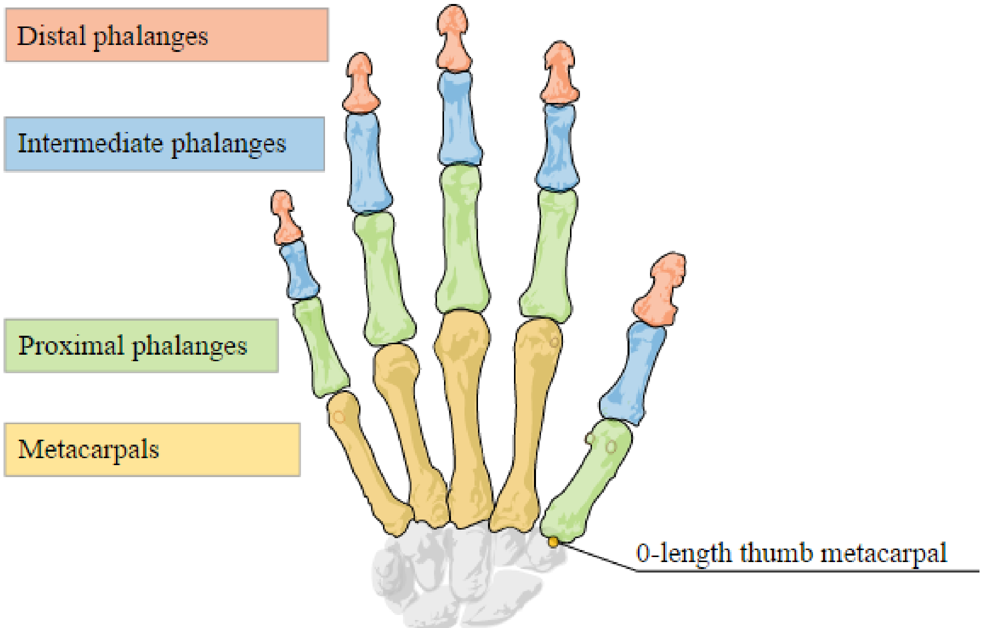

Figure 2.7: Finger bones [1].

Contents of hand CSV file

Hand parameters

- Confidence - software confidence in provided information. It is measured in scale from 0 to 1.
- Direction - the direction from the palm position toward the fingers.
- Palm normal - the normal vector to the palm 2.8. Finger bend angles are calculated comparing palm normal and finger direction.
- Palm position - the center position of the palm in millimeters from the Leap Motion Controller.
- Palm velocity - the rate of change of the palm position in millimeters/second.
- Grab angle - the angle between fingers and hand of a grab hand pose. It “is computed by looking at the angle between the direction of the 4 fingers and the direction of the hand. Thumb is not considered when computing the angle. The angle is 0 radian for an open hand, and reaches pi radians when the pose is a tight fist“ [27].
- Grab Strength - the strength of a grab hand pose.
- Wrist position - the position of the wrist of this hand.

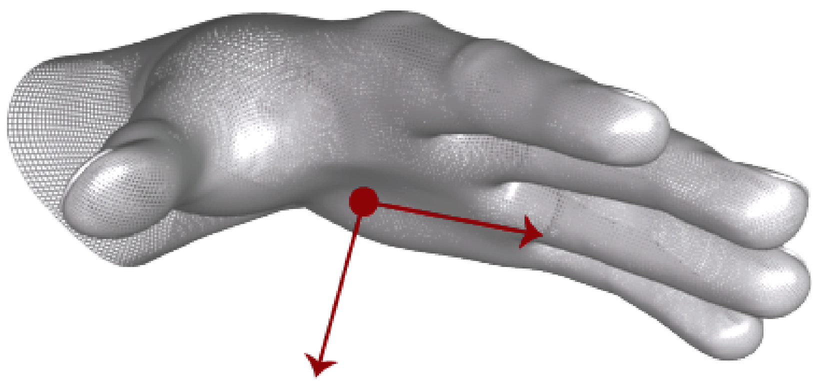

Figure 2.8: Palm vectors [1].

Figure 2.9: Finger tip direction [1].

Finger parameters

- Direction - the direction in which this finger is pointing.
- Tip position - the tip position in millimeters from the Leap Motion origin.
- Tip velocity - the rate of change of the tip position in millimeters/second.

Arm parameters

- Direction - arm orientation in space. An angle between arm and palm directions shows wrist bend angle.

Interaction events Raw hand position data may appear to be a chaotic trajectory. During objective completion hand is moving back and forth, thus we need to split this trajectory into motions, where each motion has logical start and sufficient for objective consequence. In case of exercise, where user is required stack figures on top of each other, such motion may be grasping object, moving it and releasing. It would be much easier to split hand position data into interactions with figures motions having interaction event records: object grasp and release. Those events are recorded separately into CSV file. Each record has:

- Event time
- Event type - grasp or release.
- Motion executing hand - left or right.
- Figure identifier - the figure, that was grasped or released.
- Angle between figure and palm normal on the moment of event.

Objective progress Motion is successful, if released object has become a top of the tower and failed otherwise. In order to simplify motion classification during data processing stage, objective progress records are made as well. Each record contains:

- Time.
- Amount of completed subtasks - for example, number of figures that already form a tower.
- Amount of subtasks needed to complete the objective.
- Last interacted object id - in case of stacking objective, the figure on top of the tower will be recorded.
- Accuracy - accuracy of last completed subtask. In case of stacking objective, a shift between centers of two highest figures in horizontal plane.

Object position For each interaction object (an available to grasp object), its' position and orientation is written to personal CSV file. In order to decrease the volume of data files, only coordinates of non-sleeping objects are written.

2.2 Motion extraction

Hand movement data for single exercise is put together into one CSV file, thus it needs to be split up into interaction actions at first. In current context interaction action will be defined as a set of three motions: grasping the object, moving it to particular place and releasing 2.11. We are not interested in remaining movement data, because motions without any interaction do not influence objective progress.

Grasping, moving and releasing motions should be addressed separately. While hand is moving an object, the position is changing, but posture usually stays the same. On the other side, when grasping or releasing an object, hand general position is static, but posture is changing a lot.

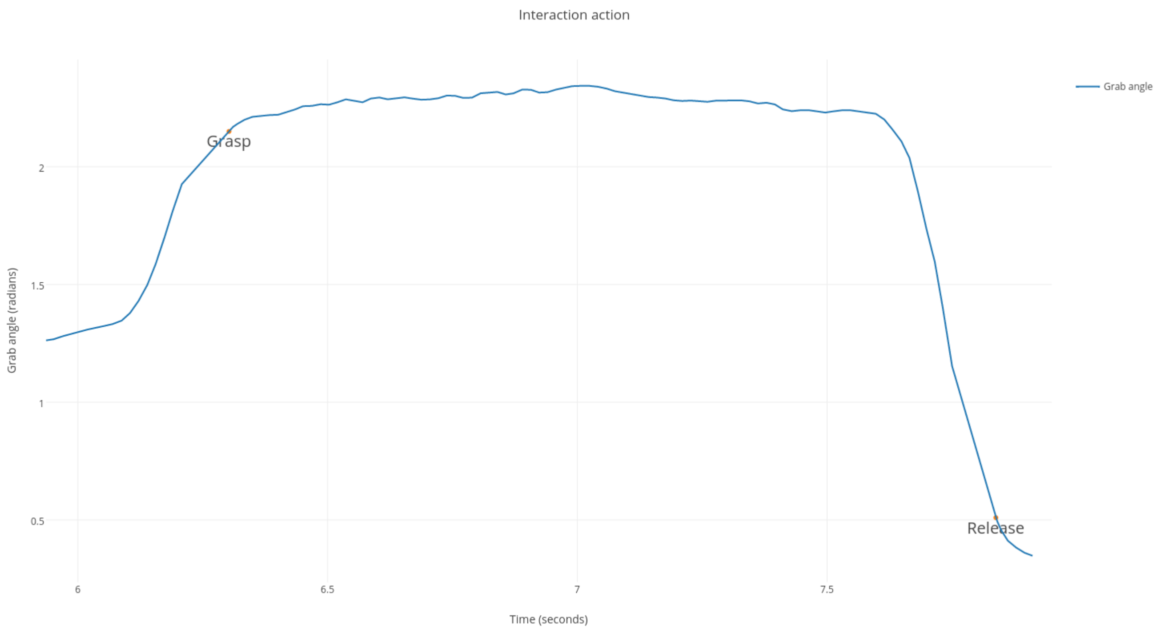

Figure 2.10: Interaction action described as a grab angle change function together with grasp and release events time.

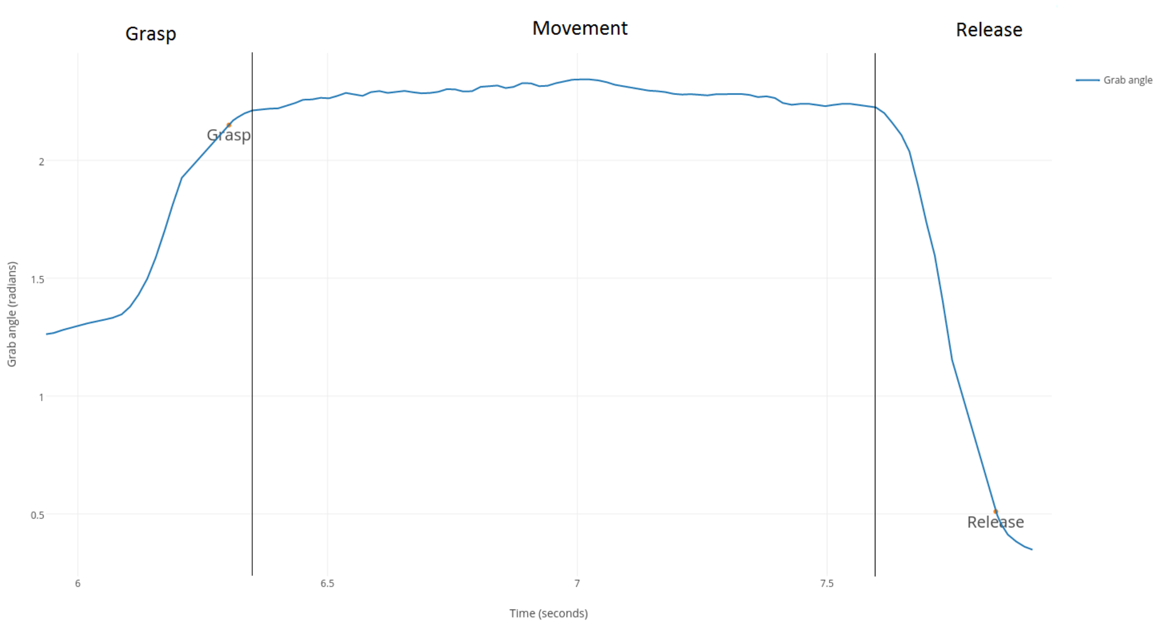

Figure 2.11: Interaction action with expected division into motions.

2.2.1 Interaction motion

Interaction events were previously recorded into separate file for splitting data purposes. Using *grasp* and *release* time, interaction frames interval may be easily found between those two events' time.

2.2.2 Release frames

Finding figure release frame interval isn't an elementary task. We may find a moment, when grab angle starts to decrease, but grab angle change trajectory isn't perfectly smooth 2.10. Filtering those tiny movements out isn't an option neither, because slow object releasing may be also excluded, making this method unreliable. Thus the most stiff slope of the grab angle change trajectory has to be found, such that figure release event is inside this interval. An example of expected motion division is shown in figure 2.11.

Considering this, more complicated division method has to be developed. Let $frames$ be a hand frame set between grasp and release records. Among $frames$, every grab angle local maximum frame is found, thus every found frame is a grab angle decreasing start point. Then for each grab angle extremum A_e an angle change coefficient ACC is calculated as a grab angle difference between this and the last frame A_{last} divided by time T passed between those two frames. Frame with the highest coefficient is considered to be a release motion start.

$$ACC = \frac{A_e - A_{last}}{T_e - T_{last}} \tag{2.1}$$

Figure 2.10 shows, that even though object is released, hand usually continues unclasping for a while. Frames from release record and until grab angle stops to decrease are defined as motion continuation.

Thus a release motion starting and ending points are defined.

2.2.3 Grasp frames

Grasp frames are found in similar way, but for this case grab angle increasing is being detected.

Motion has to start before grasp event record. Grab angle local minimum values are found, and the frame with maximum absolute value of angle change coefficient is chosen to be a squeezing motion start.

Squeezing motion ends after grasp event, when grab angle stops to increase 2.11.

2.2.4 Action classification

Once interaction action is found, it has to be labeled as successful or failed. Objective progress records will be helpful for this task. If interaction with object was successful, then new objective progress record would appear, where grasped object is mentioned as figure on top of the tower. Such record has to appear right after object was released.

After release event, hand does not influence figure any more, but possible inertial movement is taken into consideration, so 1 second is used as possible delay time.

If objective progress record, which meets described requirements, is found, then action is classified as successful.

2.3 Motion parameter choice and calculation

LM captures the highest level of fine motor movements and we should benefit from this preciseness as much as possible, so additionally to hand, every single finger is described with bunch of various parameters.

Motion parameters have to characterize hand position, orientation and their change rate on different levels of abstraction. The most abstract parameters rather play a generalization role. They describe hand in general and are often correlated with more specific ones.

Additionally to abstraction level, grasp, movement and release motions have their own parameter groups.

Movement motion parameters During movement motion, the hand posture is relatively stable, so attention is mostly paid to hand position movement and static posture description.

Palm parameters

- Time spent - amount of time waster on action.
- Distance - euclidean distance between palm center position in the beginning and ending of action.
- Trajectory length - isn't useful by itself, but shows motion effectiveness when divided by distance or time.
- Velocity - describes average movement speed. It is provided as mean and mass values.
- Acceleration - the rate of change of velocity. Describes motion smoothness.
- Jerk - acceleration change rate.
- Mean Euler angles - mean pitch, yaw and roll of palm normal. Those angles show hand orientation during carrying an object. Due to division into three separate angles, changes in rotation around each axes are separable.
- Euler angles change rate - those parameters describe the stability of palm posture during carrying an object.
- Mean grab angle - describes a position of four finders relative to palm direction while holding an object.

Finger parameters

- Mean Euler angles - the point direction of each separate finger relative to palm normal.
- Mean absolute Euler angles - the point direction of each finger relative to Leap Motion controller.
- Mean angles - a three point angle between metacarpal 2.7 base, end and a finger tip. Apart from Euler angles, which describe finger orientation for each axis, this angle directly shows a squeezing strength.
- Thumb to palm angle - since thumb is more mobile compared to other fingers, an additional angle between palm center thumb base and tip is introduced.

Release motion parameter list While releasing an objects, hand position stays the same, so release parameters are aimed to describe the changes in hand posture.

Palm parameters

- Speed - is measured in radians per second and shows grab angle velocity. It is presented as mean and mass values.
- Acceleration - the change rate of speed. It shows the smoothness of an unclasping motion.
- Time - time spent on releasing the object.

Finger parameters

- Euler angle speed - the change rate of tip direction pitch, yaw or roll during unclasping the hand.
- Euler angle acceleration - the change rate of tip direction speed.
- Angle speed - the change rate of three point angle between metacarpal base, end and tip position.
- Angle acceleration - the change rate of angle release speed.
- Thumb to palm angle speed and acceleration - additional three point angle between palm center, thumb base and tip, due to finger increased mobility.

Grasp motion parameter list Due to release and grasp motions similarity, identical parameters are used for their description.

2.4 Comparison of data samples

The research was conducted in two stages. At first stage, a group of ten untrained individuals has completed an exercise once. All their motions were gathered together into one dataset and divided into samples corresponding to the successful and failed attempts. Then for each parameter following statistical hypothesis testing was conducted. The statement of null hypothesis is that parameter values describing successful and failed trials belong to the independent random samples drawn from normal distributions with equal means $H_0 : \quad \mu_1 = \mu_2$. The alternative hypothesis is that parameter values describing successful and failed trials belong to the random samples drawn from normal distributions with unequal means $H_1 : \quad \mu_1 \neq \mu_2$. Whereas level of significance $\alpha = 0.05$. Those subsets differ by their outcome. During failed motions something has gone wrong, thus some of parameters has to be different from successful ones. Detected differences in those two subsets constitute a validness of chosen comparison method.

When the concept is proven, an actual learning was being detected. Several individuals has completed a training course of 20 trials. Then unsuccessful motions were filtered out for each testee. Two sets were randomly chosen from the beginning and ending of a training session and compared using statistical hypothesis testing method. Motion parameter value sets, that reject a null hypothesis, characterize an individuals' process of learning.

2.5 Continuous learning

Human is famous for his adaptivity to any environment. An individual will get used to the application properties fast enough, slowing down the training. Such unwanted behavior is shown in figure 3.4. The purpose of this section is to keep individual outside from his comfort zone, thus giving user a stimulus to keep his learning tempo.

In order to achieve this goal, the application is programmed to detect user learning on the run and deliberately complicate the task by adjusting some environment or object parameters if learning has been detected.

In previous section it was explained, that each motion may be described with several parameters. Significant changes in them indicate learning process.

If we present parameter values of motions set as a graphic with index representing x axis and parameter value representing y axis, then there are two types of graphics, that may represent the process learning: graphic monotonic growth or fall 3.3 and converging to certain value 3.5.

Unfortunately, a statistical hypothesis testing shows whether mean values of two samples are significantly different, while more narrow differences have to de detected. Thus another learning detection method is needed.

2.5.1 Parameter choice

Since user learning has to be detected on the run, calculating values for all parameters is not an option, because application has to work smoothly. Meticulous analysis is excess for this task, so only several parameters were chosen to represent the others. At first, characteristics, that have already shown changes during analysis are worth consideration. Secondly, palm and separate finger parameters usually correlate with each other, thus finger directions relative to room are covered with palm direction and fingers squeezing is described with hand grab angle property.

2.5.2 Data collection

Parameters are calculated for successful motions only, but since it is impossible to know whether motion will be successful before hand, once user grasps an object, application starts to write hand frame info to memory. Recording ends after object is released. Then it waits for quest progress update message. If in one second grasped figure appears on top of the tower, then motion is considered to be successful and all chosen parameters for this motion are calculated. Recorded data is erased otherwise in order to preserve memory resource.

Parameters for each hand are calculated independently and constitute independent datasets, because hands don't have equal skills.

2.5.3 Detection of increasing or decreasing trend

Linear regression is a workhorse of data science and it solves this problem too. If linear regression slope coefficient $\hat{\beta}_1$ is above or beyond certain threshold, then values are approved to be sufficiently changed, thus learning is detected. For linear regression equation

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i + \hat{\epsilon}_i \tag{2.2}$$

slope coefficient equals

$$\hat{\beta}_1 = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sum(X_i - \bar{X})^2} = \frac{n\sum(XY) - \sum X \sum Y}{n\sum X^2 - (\sum X)^2} \tag{2.3}$$

Data normalization Each parameter has its own range of possible values. For example, angle values vary from $-\pi$ to π, whereas other property range may be different. It was decided, that an interval values are adjusted to, will depend on dataset size. For example, values from sample with size 40 will vary from 0 to 40. This will prevent parameter values from being dominated by index values of bigger datasets.

Slope coefficient threshold value Noise tends to dominate over actual data in smaller datasets. That leads to dynamic threshold value. The larger dataset is, the smaller linear regression slope coefficient may be considered as sufficient changes in data sample. The behavior of parabola function perfectly suits for determining minimal slope coefficient value, because it infinitely approaches zero. An actual threshold T equation is

$$T = 3/n + 0.5 \tag{2.4}$$

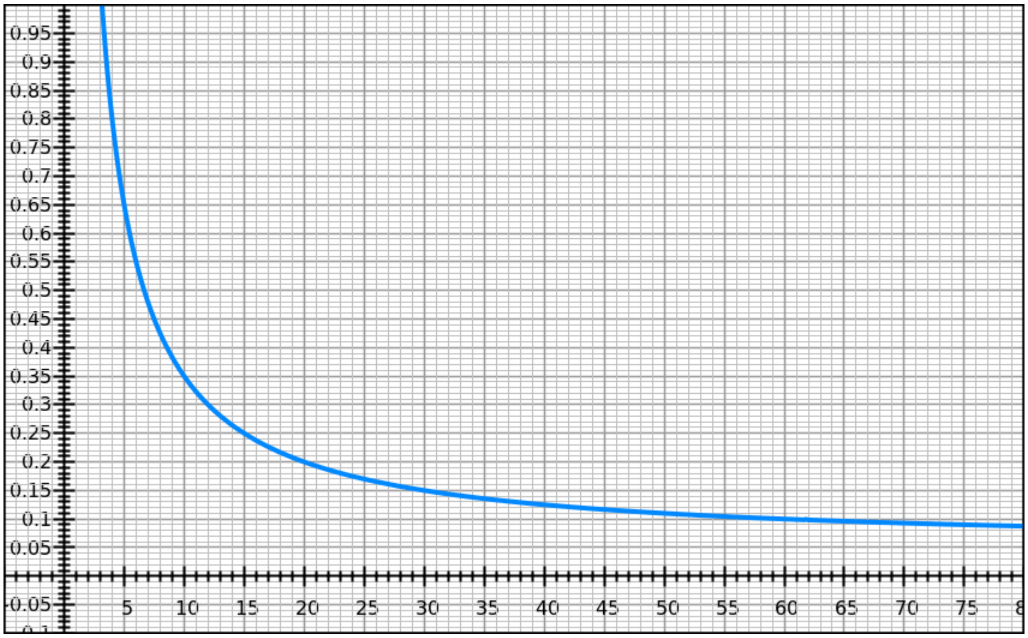

Figure 2.12: Minimal slope coefficient threshold function.

where n is dataset size 2.12. A function shape was adjusted according to actual analysis results while learning in static environment. Additionally, datasets below 20 values are rejected without verification.

2.5.4 Value convergence detection

Convergence detection problem may be converted to decreasing trend detection task, that was previously solved. If parameter values converge to certain value, then root-mean-square deviation with moving window compared to exponential moving average 2.13 graphic trend will decrease. Once parameter value sequence is converted to its' root-mean-square deviation, the linear regression slope coefficient of resulting array is calculated according to previously described method.

Exponential moving average An exponential moving average adapts to changes more quickly than simple one, which is useful for non-linear value sequences. An equation for calculation of exponential moving average value for time period t (EMA_t) may be written down as following [28]:

$$EMA_t = \alpha \times P_t + (1 - \alpha) \times EMA_{t-1} \tag{2.5}$$

$$\alpha = \frac{2}{N + 1} \tag{2.6}$$

$$EMA_1 = \frac{\sum_{t=1}^{N} P_t}{N} \tag{2.7}$$

where

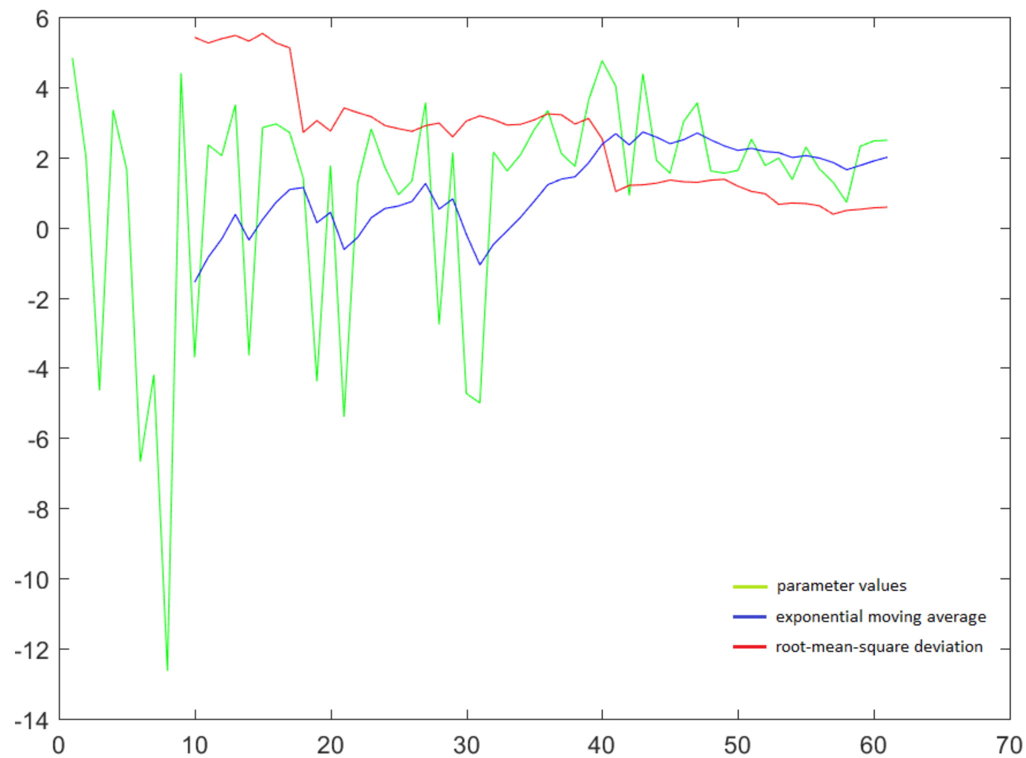

Figure 2.13: Root-mean-square deviation with moving window compared to exponential moving average.

- α - weight coefficient between 0 and 1, that determines the aging speed of older values.
- P_t - parameter value at time t.
- EMA_{t-1} - previous value of exponential moving average.
- N - size of moving window equal to 10 in the framework of this work.
- EMA_1 - the first element, which is calculated as a simple moving average.

Root-mean-square deviation A root-mean-square deviation with moving window $RMSD$ is estimated comparing to previously calculated exponential average EMA using following equation:

$$RMSD_t = \sqrt{\frac{\sum_{i=t}^{t+N}(EMA_t - P_i)^2}{N}} \tag{2.8}$$

2.5.5 Environment tuning

If at least one parameter is detected to have changing trend or value conversion, then the difficulty level is increased by one, all collected motion data is erased and the whole detection process is started again. It would be unnatural to tune environment variables on the run, so the difficulty is set in the beginning of each trial.

There are plenty parameters to play with. In dint of Unity physic material, the slipperiness and bounciness of objects may be increased with each level, but due to Leap Motion Interaction Engine implementation particular qualities, those physic properties do not affect hand and figure interaction. Additionally, high bounciness is noticeable only when object is thrown, which is not the case for "Tower building" exercise. Same applies to slipperiness. Figures usually have relatively slow velocity while being released and changes are noticeable only with slipperiness close to zero. Another candidate property is "stickiness", but unfortunately it has no effect on motion except releasing moment.

Contrariwise, the figure size has shown oneself to be the most influential property in terms of building a tower exercise. Smaller figures require greater accuracy when grasping and releasing an object, thus the choice was made in favor of figure size property. Starting from 5cm, every additional level the size is decreased by 0.5cm until the 3.5cm limit is reached. Since only successful motions are taken into account and the minimal dataset size accepted to learning detection is 20, it will take enough training time to reach maximal difficulty level, so there is no need to worry about reaching size limit too fast.

Chapter 3

Analysis

3.1 Successful and failed motions comparison

The comparison of motions, which have achieved their purpose, and those, which haven't, shows the validness of chosen research method. Since movements are already different by their outcome, the comparison should find some differences too.

Ten untrained individuals have completed the exercise once. Significantly different motion parameters for datasets with successful and failed motions of all testees together revealed by statistical hypothesis testing are presented in table 3.1.

In order to put figures on top of each other, an accuracy is needed, which may be achieved with higher time consumption, so motions with more time spent tend to be more successful 3.1. Same applies to motion velocity. Higher speed usually means less preciseness 3.2. Mean yaw and roll stand for hand orientation in space. Yaw shows hand horizontal bend and roll shows palm rotation around wrist. Finger absolute angles describe point direction relative to room, so their appearance in the table is connected to hand orientation.

3.2 Learning in static environment

Smaller number of individuals has participated in learning process. Tables 3.2 and 3.3 present parameters of successful motions, which values differ significantly between the beginning and ending of learning in static environment.

Unlike success and failure comparison, finger's and hand pitch angle - a vertical bending of the hand - has changed the most. Yaw and roll angles seem to be relatively stable for successful motions. Additionally, release speed parameters have appeared in the table. Hand unclasping speed and acceleration are decreasing, which means, that the releasing process becomes slower and smoother 3.4. Pinky also shows changes in angles different from pitch. That can be explained by pinky slightly deviated move trajectory.

3.3 Learning with regularly increasing difficulty

Same setting is used for this experiment with single modification. Every five trials the difficulty of an exercise was increased, so that user is kept out from comfort zone. A complete list of significantly changed parameters is presented in tables 3.4, 3.5 and 3.6.

There are noticeably more parameters, but changes in some of them are caused directly by difficulty growth and probably have nothing common with actual learning. Because of difficulty growth, user spends more time on each motion by slowing down his movements. Moreover, for this setting the only parameter being changed, was cube size, which describes palm mean grab angle and hold angles of separate fingers changes.

Unrecognized before, but in this case grasp velocity improvements are clearly visible, clasping movement parameters are becoming more stable 3.5. Those changes have affected all finger except thumb.

Release parameters have endured even greater changes 3.6. Additionally to results from previous cases, thumb movement improvements are detected. Its' velocity values are constantly becoming more stable.

3.4 Learning with improvement detection

For this section, an application was augmented with learning detection functionality. Every time, changes were detected, the exercise difficulty was increased by one level. Tables 3.7, 3.8 and 3.9 present movement, grasp and release motion parameters accordingly, which have significantly changed during learning full training period according to statistical hypothesis method.

Selected for this case parameters, are quite similar to those, found in previous section. There is a speed loss 3.11 and increased time consumption, because difficulty becomes higher.

Additionally, there are notable changes in hand orientation. Palm pitch 3.7 and roll angles has changed the most and finger corresponding absolute Euler angles 3.8 correlate with them.

Regarding grasp and release motions, every single finger has shown squeezing and unclasping speed differences compared to beginning of the training 3.10.

Parameter name		p-value	t-statistic
	Time spent	0.0033353	3.011
Palm	Mean velocity	0.042588	-2.0554
	Mean Yaw	0.012652	2.5416
	Mean Roll	0.002098	3.1629
Index	Mean Pitch	0.026249	2.2578
	Mean Absolute Yaw	0.021292	2.3416
	Mean Absolute Roll	0.0023397	3.1276
Middle	Mean Pitch	0.015792	2.4578
	Mean Absolute Yaw	0.033475	2.1577
	Mean Absolute Roll	0.0036425	2.9816
Ring	Mean Pitch	0.015624	2.4619
	Mean Absolute Yaw	0.01459	2.488
	Mean Absolute Roll	0.0067563	2.7693
Pinky	Mean Absolute Yaw	0.0068441	2.7647
	Mean Absolute Roll	0.033557	2.1566
Index	Yaw Grasp Speed Mass	0.04923	-1.9921

Table 3.1: Parameters which values differ significantly between the `successful and failed` attempts

Parameter name		p-value	t-statistic
Palm	Pitch Change Rate	0.0045023	-2.9212
	Mean Pitch	1.5145e-05	-4.601
Index	Mean Index Yaw	0.032692	2.1727
	Mean Index Absolute Pitch	0.0075329	-2.7402
Middle	Mean Absolute Pitch	0.00015532	-3.9665
Ring	Mean Absolute Pitch	0.0037563	-2.9832
Thumb	Mean Pitch	0.0031884	3.0386
	Mean Absolute Pitch	0.0072668	2.7531

Table 3.2: Significantly changed `movement` motion parameters during learning in `static environment`.

Parameter name		p-value	t-statistic
Palm	Mean Speed	0.012929	2.5413
	Speed Mass	0.019342	2.3859
	Mean Acceleration	0.038994	2.0979
	Acceleration Mass	0.0036659	2.9915
Middle	Mean Pitch Speed	0.0076217	2.736
	Pitch Speed Mass	0.031483	2.1884
	Pitch Acceleration Mass	0.005969	2.823
	Angle Mean Speed	0.0056363	-2.8432
	Angle Speed Mass	0.023252	-2.3126
	Angle Mean Acceleration	0.047578	-2.0113
	Angle Acceleration Mass	0.01282	-2.5445
Ring	Mean Pitch Speed	0.0080343	2.717
	Pitch Speed Mass	0.032138	2.1798
	Pitch Acceleration Mass	0.02745	2.2452
	Angle Mean Speed	0.0033459	-3.0224
	Angle Speed Mass	0.01861	-2.4011
	Angle Acceleration Mass	0.046034	-2.0258
Pinky	Mean Pitch Speed	0.0012726	3.3375
	Pitch Speed Mass	0.0031971	3.0377
	Mean Yaw Speed	0.028083	-2.2358
	Yaw Speed Mass	0.0063074	-2.8035
	Mean Roll Speed	0.019778	-2.3771
	Roll Speed Mass	0.019189	-2.389
	Angle Mean Speed	0.00032846	-3.7499
	Angle Speed Mass	0.0012641	-3.3396
Thumb	Angle Speed Mass	0.035524	-2.1376
	To Palm Angle Acceleration Mass	0.016314	2.4524

Table 3.3: Significantly changed `release` motion parameters during learning in `static environment`.

Parameter name		p-value	t-statistic
	Time Spent	0.015288	-2.4982
	Distance	0.0093643	2.6865
Palm	Mean Velocity	7.3261e-05	4.265
	Pitch Change Rate	0.010692	-2.6364
	Mean Pitch	0.0034653	-3.0459
	Mean Roll	0.00023885	3.9126
	Mean Grab Angle	0.0054746	2.8839
Index	Mean Absolute Pitch	0.0078837	-2.7507
	Mean Absolute Roll	0.0013181	3.373
Middle	Mean Roll	0.040467	-2.0951
	Mean Absolute Roll	0.0080773	2.7417
Ring	Mean Pitch	0.0019687	3.2395
	Mean Roll	8.4101e-06	-4.8802
	Mean Absolute Roll	0.01077	2.6336
Pinky	Mean Pitch	4.8488e-06	5.032
	Mean Roll	1.8717e-08	-6.5022
	Mean Absolute Pitch	0.035202	2.1556
	Mean Absolute Yaw	0.048243	-2.0171
	Mean Absolute Roll	0.012265	2.5839
	Mean Angle	0.0067711	-2.8067
Thumb	Mean Pitch	4.2212e-05	4.425
	Mean Roll	0.0054022	2.8887
	Mean Absolute Pitch	4.0637e-05	4.4359
	Mean Absolute Roll	1.6875e-05	4.6858
	Mean Angle	3.8065e-05	-4.4547
	Mean To Palm Angle	0.031107	-2.2085

Table 3.4: Significantly changed `movement` motion parameters during learning in `regularly changing environment`.

Parameter name		p-value	t-statistic
Palm	Speed Mass	0.015739	2.4868
Index	Mean Pitch Speed	0.0016236	-3.304
	Pitch Speed Mass	0.0002423	-3.9082
	Mean Roll Speed	1.8354e-06	-5.2962
	Roll Speed Mass	1.8388e-06	-5.2957
Middle	Mean Pitch Speed	0.023298	-2.3291
	Pitch Speed Mass	0.0070765	-2.7906
	Mean Roll Speed	2.679e-07	-5.8088
	Roll Speed Mass	3.0312e-07	-5.7763
	Mean Roll Acceleration	0.035033	2.1577
Ring	Mean Roll Speed	1.3912e-05	-4.74
	Roll Speed Mass	8.4903e-07	-5.503
Pinky	Mean Pitch Speed	0.0066576	2.8129
	Pitch Speed Mass	0.00047696	3.6991
	Mean Roll Speed	0.00028809	-3.8553
	Roll Speed Mass	3.7987e-05	-4.4553
	Mean Roll	0.0073019	-2.779
	Roll Acceleration Mass	0.00021119	-3.95
	Angle Speed Mass	0.02167	-2.3587

Table 3.5: Significantly changed `grasp` motion parameters during learning in `regularly changing environment`.

	Parameter name	p-value	t-statistic
	Time	0.014228	-2.5264
Palm	Mean Speed	0.0036577	-3.027
	Mean Acceleration	0.011663	-2.6032
	Acceleration Mass	0.01067	-2.6372
Index	Mean Pitch Acceleration	0.013727	-2.5403
	Pitch Acceleration Mass	0.01327	-2.5535
	Mean Roll Speed	0.0017662	3.2759
	Mean Roll Acceleration	0.0063147	2.8322
	Roll Acceleration Mass	0.0029846	3.0977
	Angle Mean Speed	0.021137	2.3688
	Angle Mean Acceleration	0.017838	2.4371
	Angle Acceleration Mass	0.036143	2.1443
Middle	Mean Pitch Speed	0.0083453	-2.7296
	Mean Pitch Acceleration	0.0098568	-2.6672
	Pitch Acceleration Mass	0.015323	-2.4973
	Mean Roll Speed	4.0907e-05	4.434
	Roll Speed Mass	0.042435	2.0742
	Mean Roll Acceleration	0.0074897	2.7697
	Roll Acceleration Mass	0.0065289	2.8201
	Angle Mean Speed	0.0335	2.1769
	Angle Mean Acceleration	0.013554	2.5453
	Angle Acceleration Mass	0.020264	2.3859
Ring	Mean Pitch Speed	0.00052642	-3.6681
	Mean Pitch Acceleration	0.012718	-2.5699
	Pitch Acceleration Mass	0.013349	-2.5512
	Mean Roll Speed	3.2058e-05	4.5039
	Roll Speed Mass	0.030877	2.2116
	Mean Roll Acceleration	0.0086893	2.7145
	Roll Acceleration Mass	0.013555	2.5452
	Angle Mean Speed	0.0038899	3.0054
	Angle Mean Acceleration	0.019299	2.4056
	Angle Acceleration Mass	0.016458	2.4691
Pinky	Mean Pitch Speed	0.0042237	-2.9764
	Mean Pitch Acceleration	0.026779	-2.2715
	Mean Roll Speed	0.0020459	3.2266
	Roll Speed Mass	0.025323	2.2947
	Roll Acceleration Mass	0.010569	-2.6408
	Angle Mean Speed	0.0076325	2.7627
	Angle Mean Acceleration	0.024866	2.3023
Thumb	Mean Pitch Speed	0.013379	-2.5503
	Angle Mean Acceleration	0.0038775	3.0066
	Angle Acceleration Mass	0.027712	2.2572
	To Palm Angle Mean Speed	0.040705	2.0925
	To Palm Angle Mean Acceleration	0.022417	2.3449

Table 3.6: Significantly changed `release` motion parameters during learning in `regularly changing environment`.

Parameter name		p-value	t-statistic
	Time Spent	0.0038799	-2.9803
	Distance	0.00076214	3.5099
	Trajectory Length	0.0065975	2.7944
Palm	Mean Velocity	1.9748e-09	6.8247
	Velocity Mass	0.0063609	2.8074
	Roll Change Rate	0.016999	2.4412
	Mean Pitch	2.3984e-05	-4.5048
	Mean Roll	0.026191	2.2683
Index	Mean Absolute Pitch	6.0429e-05	-4.2516
	Mean Absolute Roll	0.0035676	3.009
Middle	Mean Absolute Pitch	0.001597	-3.2758
	Mean Absolute Yaw	0.028445	2.2343
	Mean Absolute Roll	0.018289	2.4125
Ring	Mean Absolute Pitch	0.030152	-2.21
	Mean Absolute Roll	0.023037	2.3205
Pinky	Mean Absolute Pitch	0.011568	-2.5886
	Mean Absolute Yaw	0.033136	2.1705
	Mean Absolute Roll	0.010499	2.6249
Thumb	Mean Pitch	0.0013134	3.3386
	Mean Yaw	0.010002	-2.6429
	Mean Roll	0.021622	-2.346
	Mean Absolute Pitch	0.0044996	2.9292
	Mean Absolute Yaw	0.00039946	-3.7069
	Mean Angle	0.029767	-2.2154

Table 3.7: Significantly changed `movement` motion parameters during training in `learning detection environment`.

Parameter name		p-value	t-statistic
Index	Pitch Speed Mass	0.011392	-2.5944
Middle	Pitch Speed Mass	0.031961	-2.1857
	Mean Pitch Acceleration	0.023554	-2.3115
	Pitch Acceleration Mass	0.020542	-2.3665
	Angle Mean Acceleration	0.0088914	2.6863
	Angle Acceleration Mass	0.0089571	2.6836
Ring	Mean Pitch Acceleration	0.018363	-2.4109
	Pitch Acceleration Mass	0.01954	-2.3863
	Roll Speed Mass	0.012544	2.558
	Angle Mean Acceleration	0.0048096	2.906
	Angle Acceleration Mass	0.0037917	2.9882
Pinky	Mean Roll Speed	0.026992	2.2559
	Roll Speed Mass	0.0083575	2.709
Thumb	Mean Pitch Speed	0.013464	2.5311
	Pitch Speed Mass	0.028086	2.2395
	Mean Yaw Speed	0.002852	-3.0848
	Yaw Grasp Mass	0.0016641	-3.2624
	Mean Roll Speed	0.0022804	-3.1593
	Roll Grasp Mass	0.0033236	-3.0331
	To Palm Angle Mean Acceleration	0.0017978	3.2373
	To Palm Angle Acceleration Mass	0.0018213	3.2331

Table 3.8: Significantly changed `grasp` motion parameters during training in `learning detection environment`.

Parameter name		p-value	t-statistic
Palm	Mean Speed	0.0054603	-2.8615
	Mean Acceleration	0.0030812	-3.0587
	Acceleration Mass	0.0017111	-3.2534
Index	Mean Pitch Speed	0.00029327	-3.7989
	Pitch Speed Mass	0.012779	-2.551
	Mean Pitch Acceleration	0.00014139	-4.0115
	Pitch Acceleration Mass	2.2856e-05	-4.5178
	Mean Yaw Acceleration	0.012245	2.5672
	Yaw Acceleration Mass	0.032171	2.1829
	Mean Roll Speed	0.031556	2.191
	Mean Roll Acceleration	0.0048546	2.9028
	Roll Acceleration Mass	0.0003092	3.7833
	Angle Mean Speed	0.00038578	3.7173
	Angle Speed Mass	0.042352	2.0653
	Angle Mean Acceleration	0.0001241	4.0489
	Angle Acceleration Mass	1.4535e-05	4.6391
Middle	Mean Pitch Speed	0.0039938	-2.9704
	Mean Pitch Acceleration	0.00052328	-3.6253
	Pitch Acceleration Mass	0.00029231	-3.7999
	Mean Yaw Acceleration	0.0086158	2.6979
	Yaw Acceleration Mass	0.023358	2.3149
	Mean Roll Acceleration	0.044734	2.0414
	Angle Mean Speed	0.0036009	3.0058
	Angle Mean Acceleration	0.00027648	3.8164
	Angle Acceleration Mass	8.1672e-05	4.1674
Ring	Mean Pitch Speed	0.022875	-2.3233
	Mean Pitch Acceleration	0.00046238	-3.6628
	Pitch Acceleration Mass	0.0014511	-3.3066
	Mean Yaw Acceleration	0.0029647	3.0717
	Yaw Acceleration Mass	0.012586	2.5568
	Roll Speed Mass	0.01093	-2.6099
	Angle Mean Speed	0.020855	2.3604
	Angle Mean Acceleration	0.0002526	3.843
	Angle Acceleration Mass	0.00045246	3.6694
Pinky	Angle Mean Acceleration	0.028539	2.2329
	Angle Acceleration Mass	0.02593	2.2724
Thumb	To Palm Angle Mean Speed	0.0030777	3.0591
	To Palm Angle Mean Acceleration	0.00022552	3.8762
	To Palm Angle Acceleration Mass	2.699e-05	4.4729

Table 3.9: Significantly changed `release` motion parameters during training in `learning detection environment`.

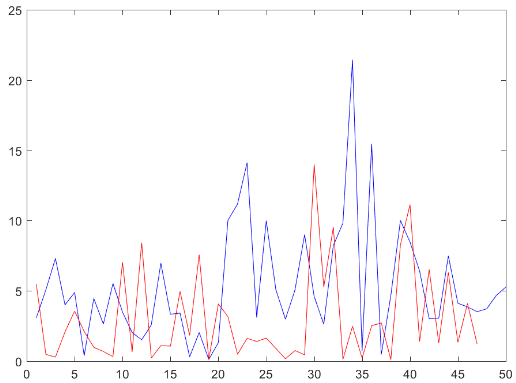

Figure 3.1: Time spent. Blue stands for success and red stands for failure.

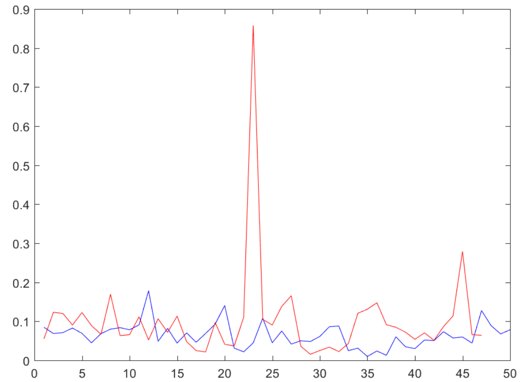

Figure 3.2: Mean velocity.

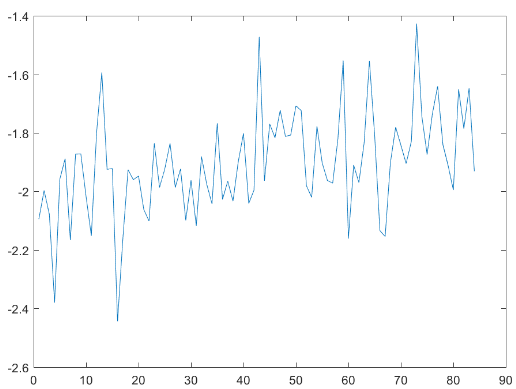

Figure 3.3: Mean pitch values during learning process. Value significantly increases.

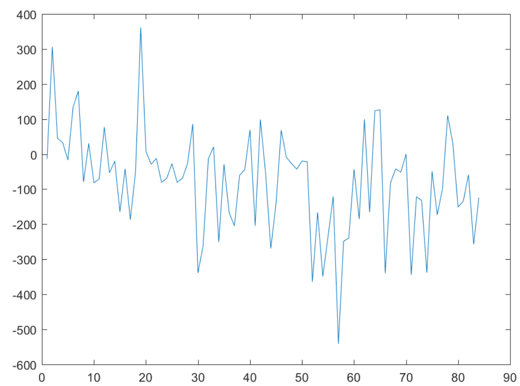

Figure 3.4: Release acceleration mass. Releasing becomes smoother, but parameter stops to change at some point.

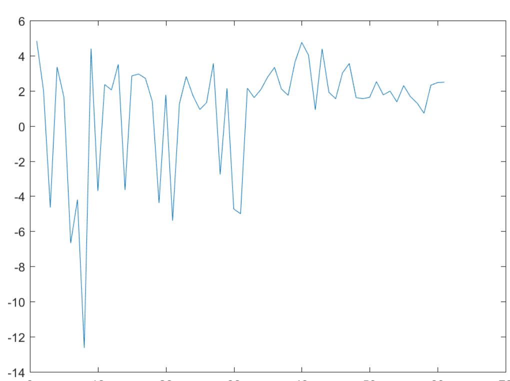

Figure 3.5: Index mean pitch grasp speed change history during learning process with regular difficulty growth.

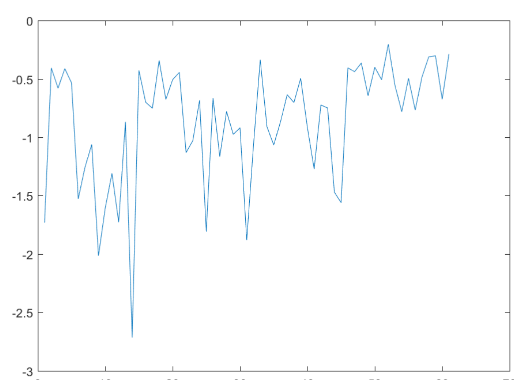

Figure 3.6: Mean release speed. Values converge to certain value.

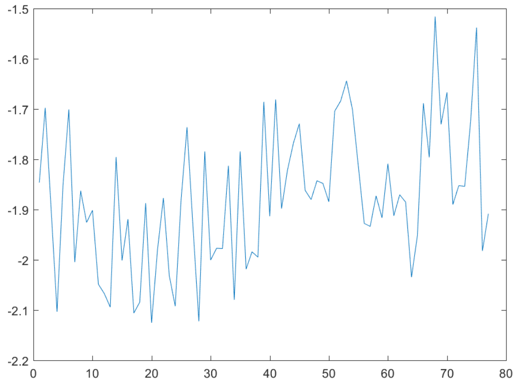

Figure 3.7: Mean palm pitch angle during training with learning detection.

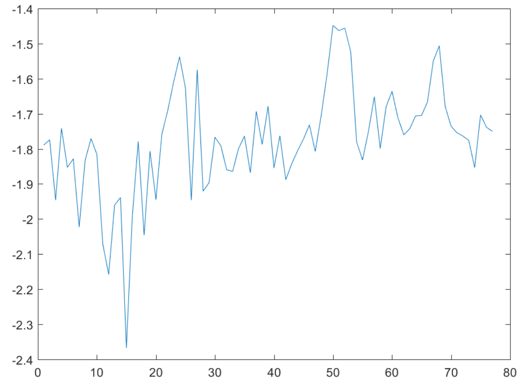

Figure 3.8: Index mean absolute pitch during training with learning detection correlates with palm pitch.

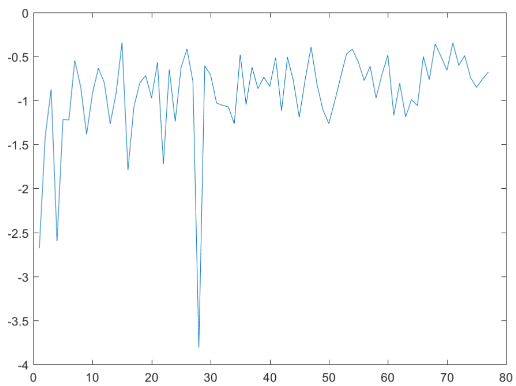

Figure 3.9: Mean release speed during training with learning detection.

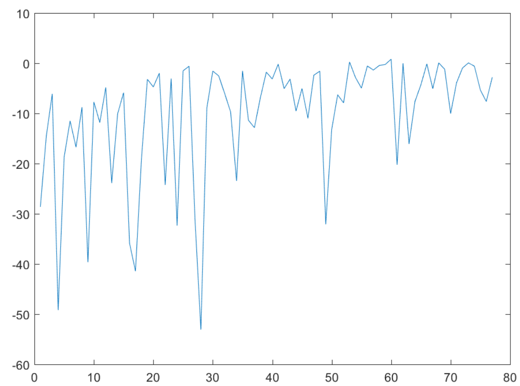

Figure 3.10: Index mean release acceleration during training with learning detection correlates with grab angle changes.

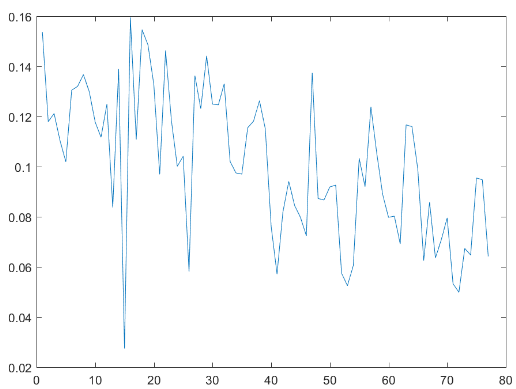

Figure 3.11: Mean palm velocity during training with learning detection.

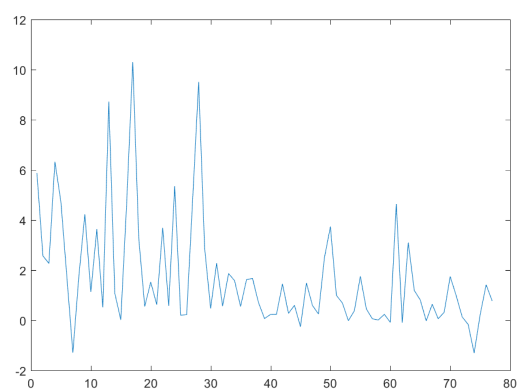

Figure 3.12: Thumb to palm angle release acceleration during training with learning detection.

Chapter 4

Discussion

Results of previous section has clearly shown, that hand and finger orientation parameters prevail over kinematic ones. It is also obvious, that release motion speed and acceleration have been affected the most during training. However, there is currently only one implemented task, so it is obvious, that motion parameters depend on that exercise. Having a set of different exercises, parameters that change for most of them could have been considered as exercise independent.

Additionally, a large variety of motion parameters have been introduced, but the list may be complemented with additional characteristics. Current parameter set is the most suitable for describing an exercises, where user is required to grasp objects, because there are lots of parameters for hand squeezing and unclasping. Additionally, a hand orientation from different perspectives has been described, but it is really true, that hand and finger movement trajectory analysis didn't receive much attention yet. It is a whole field to be discovered. Movement linearity, stability, hand jolting and many more - all those features can be described by the means of trajectory analysis.

Moreover, motion mass, described in "State of the art" section, can perfectly describe finger movement. Instead of tip position analysis, motion mass will put all finger joints together. However, finger joints are only intermediary, that put tip to certain position and orientation, which are an actual goal of the motion, so the description of tip position characteristics should be enough in the framework of this research.

It was decided to calculate parameters only for motions, when hand is interacting with objects. Those motions are most valuable in terms of completing a task. However, hand movements between interactions have a huge contribution as well. For example, they may describe the effectiveness of chosen movement trajectory.

Regarding the group of participants, group size was suitable for proving the concept of learning detection framework, but there definitely has to be more participants in order to model the way of learning new activity, so that correlating parameter set is picked from greater selection.

The Leap Motion also has several drawbacks. Controller work stability highly depends on lightening. If any source of light is pointed towards sensor, then controller often loses hands, so it works the best in poor lightening conditions.

If while working with Leap Motion user is required to hold his hand above controller for a long time, he can eventually feel tiredness in his hands. Unfortunately, fatigue affects his movement parameters, so breaks have to taken during training sessions.

Considering listed above, despite obvious changes in human motions have been demonstrated as an outcome of the framework, there is still no ending point in this research.

Chapter 5

Conclusion

The application, that records user movement data has been successfully developed for this research and based on recorded data four parameter sets were found. Parameters, which values differ between successful and failed motion constitute the first set. Remaining three sets contain parameters, which values are different in the beginning and ending of training session for three types of virtual environments: static, with regularly increasing difficulty and with complexity boost depending on user motion parameter value changes.

The main tool is virtual reality application with partial immersion. VR eliminates the sensor eclipse problem, making data gathering more reliable. Moreover, despite being similar to real world, even without purposeful changes user has to adapt to virtual reality. Thus, individual has to learn even basic motions such as grasping and releasing objects.

The idea to stimulate user learning by increasing objective difficulty has shown impressive results. A sets of parameters, which values have changed during training sessions with increasing difficulty contain greater amount of parameters and those parameter value graphics show more expressive picture, thus the control over game environment emphasizes the virtual reality role.

Taking everything listed above into account, I consider stated problem - develop a framework, that detects changes in user motions during learning new fine motor activity - being solved. Developed framework allows to detect parameters, which values have changed during training period. Thus, expanding this framework depending on needs of activity being learned, one will get exercise numeric evaluation material, becoming independent from subjective grading.

Bibliography

[1] “Leap motion sdk documentation.” `https://developer.leapmotion.com/documentation/index.html?proglang=current`. Accessed: 2017-04-25.

[2] V. Tammik and A. Toomela, “Relationships between visual figure discrimination, verbal abilities, and gender,” *Perception*, vol. 42, no. 9, pp. 971–984, 2013.

[3] J. A. Piedra and J. J. Ojeda, “Virtual environment for the training of the hands in minimally invasive thoracic surgery,” 2016.

[4] S. Nõmm, A. Toomela, J. Kozhenkina, and T. Toomsoo, “Quantitative analysis in the digital luria’s alternating series tests,”

[5] S. Nõmm, A. Toomela, J. Kozhenkina, and J. Borushko, “Alternative approach to model changes of human motor functions,” 2013.

[6] S. Nõmm, K. Bardõš, I. Mašarov, A. Toomela, and T. Toomsoo, “Recognition and analysis of the contours drawn during the poppelreuter’s test,”

[7] S. Nõmm, A. Toomela, and J. Kulikov, “Towards modeling of finger motions in virtual reality environment,” 2017.

[8] H. Ying, M. Schlösser, A. Schnitzer, T. Schäfer, M. E. Schläfke, S. Leonhardt, and M. Schiek, “Distributed intelligent sensor network for the rehabilitation of parkinson’s patients,”

[9] H. Yamaguchi and T. Kondo, “Analysis of motor skills for throwing darts: Measurement of release timing,” 2011.

[10] S. Nomm and K. Buhhalko, “Monitoring of the human motor functions rehabilitation by neural networks based system with kinect sensor,” *IFAC Proceedings Volumes*, vol. 46, no. 15, pp. 249 – 253, 2013.

[11] S. Nõmm and A. Toomela, “An alternative approach to measure quantity and smoothness of the human limb motions,” *Estonian Journal of Engineering*, vol. 19, no. 4, p. 298–308, 2013.

[12] S. Nõmm, T. Kõnnusaar, and A. Toomela, “Towards establishing relationships between human arousal level and motion mass,” 2016.

[13] R. A. Luria, *Higher Cortical Functions in Man*.

[14] L. M. de Lau and M. M. Breteler, "Epidemiology of parkinson's disease," *The Lancet Neurology*, pp. 525–535, 2006.

[15] S. Nõmm, A. Toomela, M. Vaske, D. Uvarov, and P. Taba, "An alternative approach to distinguish movements of parkinson disease patients,"

[16] J. Wall, C. Bell, S. Campbell, and J. Davis, "The timed get-up-and-go test revisited : Measurement of the component tasks," *Journal of Rehabilitation Research and Development*, vol. 37, no. 1, pp. 109–114, 2000.

[17] S. D. Sala, M. Laiacone, C. Trivelli, and H. Spinnler, "Poppelreuter-ghent's overlapping figures test: Its sensitivity to age, and its clinical use," vol. 10, pp. 511–534, 1995.

[18] J. de Knecht and K. Schutte, "Finding map correspondence using geometiric models," 1996.

[19] "Leap motion technology." `https://www.leapmotion.com/product/vr`. Accessed: 2017-03-22.

[20] C. Marquardt and N. Mai, "A computational procedure for movement analysis in handwriting," 1994.

[21] K. Caro, J. Beltran, A. I. Martinez-Garcia, and V. Soto-Mendoza, "Exercaveroom: A technological room for supporting gross and fine motor coordination of children with developmental disorders."

[22] W. Lu, Z. Tong, and J. Chu, "Dynamic hand gesture recognition with leap motion controller," 2016.

[23] R. G. Lupu, N. Botezatu, F. Ungureanu, D. Ignat, and A. Moldoveanu, "Virtual reality based stroke recovery for upper limbs using leap motion," 2016.

[24] A. C. Furtado, I. Cheng, E. Fung, B. Zheng, and A. Basu, "Low resolution tool tracking for microsurgical training in a simulated environment," 2016.

[25] P. Drotar, J. Mekyska, I. Rektorova, L. Masarova, Z. Smekal, and M. Faundez-Znuy, "Evaluation of handwriting kinematics and pressure for differential diagnosis of parkinson's disease," 2016.

[26] "Euler angles." `https://en.wikipedia.org/wiki/Euler_angles`. Accessed: 2017-04-03.

[27] "Leap.hand class reference." `https://developer.leapmotion.com/documentation/unity/api/gen-unity/class_leap_1_1_hand.html`. Accessed: 2017-04-03.

[28] "Exponential moving average." `http://allfi.biz/Forex/TechnicalAnalysis/Trend-Indicators/jeksponencialnoe-skolzjashhee-srednee.php`. Accessed: 2017-04-27.