Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies

FIIT-182905-73652

Lukáš Babula

User Model for Determining User's Motor Skills

Master's Thesis

Degree course: Intelligent Software Systems
Fields of study: 9.2.5. Software Engineering (primary), 9.2.8. Artificial Intelligence (secondary)
Institute: Institute of Informatics, Information Systems and Software Engineering, FIIT STU Bratislava
Supervisor: Kamil Burda

2019, April

Zadanie diplomovej práce

Meno študenta: **Bc. Lukáš Babula**

Študijný program: Inteligentné softvérové systémy

Študijný odbor: Softvérové inžinierstvo – hlavný študijný odbor
Umelá inteligencia – vedľajší študijný odbor

Názov práce: **Model používateľa pre určenie jeho motorických schopností**

Samostatnou výskumnou a vývojovou činnosťou v rámci predmetov Diplomový projekt I, II, III vypracujte diplomovú prácu na tému, vyjadrenú vyššie uvedeným názvom tak, aby ste dosiahli tieto ciele:

Všeobecný cieľ:

Vypracovaním diplomovej práce preukážte, ako ste si osvojili metódy a postupy riešenia relatívne rozsiahlych projektov, schopnosť samostatne a tvorivo riešiť zložité úlohy aj výskumného charakteru v súlade so súčasnými metódami a postupmi študovaného odboru využívanými v príslušnej oblasti a schopnosť samostatne, tvorivo a kriticky pristupovať k analýze možných riešení a k tvorbe modelov.

Špecifický cieľ:

Vytvorte riešenie zodpovedajúce návrhu textu zadania, ktorý je prílohou tohto zadania. Návrh bližšie opisuje tému vyjadrenú názvom. Tento opis je záväzný, má však rámcový charakter, aby vznikol dostatočný priestor pre Vašu tvorivosť.

Riaďte sa pokynmi Vášho vedúceho.

Pokiaľ v priebehu riešenia, opierajúc sa o hlbšie poznanie súčasného stavu v príslušnej oblasti, alebo o priebežné výsledky Vášho riešenia, alebo o iné závažné skutočnosti, dospejete spoločne s Vaším vedúcim k presvedčeniu, že niečo v texte zadania a/alebo v názve by sa malo zmeniť, navrhnite zmenu. Zmena je spravidla možná len pri dosiahnutí kontrolného bodu.

Miesto vypracovania: Ústav informatiky, informačných systémov a softvérového inžinierstva, FIIT STU v Bratislave

Vedúci práce: **Ing. Kamil Burda**

Termíny odovzdania:

Podľa harmonogramu štúdia platného pre semester, v ktorom máte príslušný predmet (Diplomový projekt I, II, III) absolvovať podľa Vášho študijného plánu

Predmety odovzdania:

V každom predmete dokument podľa pokynov na www.fiit.stuba.sk v časti:
home > Informácie o > štúdiu > harmonogram štúdia > diplomový projekt.

V Bratislave dňa 12. 2. 2018

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4
1

prof. Ing. Pavol Návrat, PhD.
riaditeľ Ústavu informatiky, informačných systémov a softvérového inžinierstva

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ

Návrh zadania diplomovej práce

Finálna verzia do diplomovej práce [1]

Študent:

Meno, priezvisko, tituly:	Lukáš Babula, Bc.
Študijný program:	Inteligentné softvérové systémy
Kontakt:	lukas.babula@gmail.com

Výskumník:

Meno, priezvisko, tituly:	Kamil Burda, Ing.

Projekt:

Názov:	Model používateľa pre určenie jeho motorických schopností
Názov v angličtine:	User Model for Determining User's Motor Skills
Miesto vypracovania:	Ústav informatiky, informačných systémov a softvérového inžinierstva, FIIT STU, Bratislava
Oblasť problematiky:	modelovanie používateľa, strojové učenie

Text návrhu zadania[2]

V súčasnosti sú stále viac a viac rozšírené aplikácie ovládané telesnými pohybmi (gestami) pomocou senzorov ako napr. Kinect alebo Leap Motion. Okrem hier sa čoraz viac skúma využitie pohybových senzorov na rehabilitáciu pacientov či personalizáciu aplikácií.

Motoricky menej zruční používatelia majú nevýhodu v škálovaní obťažnosti gest naprieč rôznymi aplikáciami. Často sa nedokážu prispôsobiť stanovenej náročnosti a používanie takýchto aplikácií, pri ktorých je potrebná interakcia pomocou gest rukami, je pre nich náročne alebo aj nemožné.

Analyzujte súčasný stav modelovania používateľa pre rozpoznanie úrovne jeho motorických schopností na základe jeho gest rukami. Analyzujte možnosti získavania charakteristík, podľa ktorých sú posudzované motorické schopnosti ľudí a úspešnosť koordinácie ich rúk. Navrhnite a implementujte model používateľa na základe jeho gest rukami, ktorý dokáže adekvátne reprezentovať rôznu úroveň motorických schopností používateľov. Implementuje takto vytvorený model do aplikácie, aby bola schopná prispôsobiť sa svojou náročnosťou motorickým schopnostiam používateľa. Navrhnite a vykonajte sériu experimentov zameraných na stanovenie úrovne motorických schopností používateľa na základe jeho gest rukami. Na základe výsledkov jednotlivých experimentov určte, ktoré charakteristiky majú najväčší vplyv na presnosť stanovenia úrovne motorických schopností používateľa a zhodnoťte presnosť vybraných metód.

[1] Vytlačiť obojstranne na jeden list papiera

[2] 150-200 slov (1200-1700 znakov), ktoré opisujú výskumný problém v kontexte súčasného stavu vrátane motivácie a smerov riešenia

Literatúra[3]

- JIANG, F. et al.: Viewpoint-independent hand gesture recognition with Kinect. In: Signal, Image and Video Processing, 8(1), 2014, pp.163-172.
- NUGRAHANINGSIH, N., PORTA, M., SCARPELLO, G.: A Hand Gesture Approach to Biometrics,. In: New Trends in Image Analysis and Processing -- ICIAP 2015 Workshops, V. Murino, E. Puppo, D. Sona, M. Cristani, and C. Sansone, Eds. Springer International Publishing, 2015, pp. 51–58.

Vyššie je uvedený návrh diplomového projektu, ktorý vypracoval(a) Bc. Lukáš Babula, konzultoval(a) a osvojil(a) si ho Ing. Kamil Burda a súhlasí, že bude takýto projekt viesť v prípade, že bude pridelený tomuto študentovi.

V Bratislave dňa 13.12.2017

Podpis študenta | Podpis výskumníka

Vyjadrenie garanta predmetov Diplomový projekt I, II, III

Návrh zadania schválený: áno / ~~nie~~[4]

Dňa: ....12. 2. 2018........

Podpis garanta predmetov

---

[3] 2 vedecké zdroje, každý v samostatnej rubrike a s údajmi zodpovedajúcimi bibliografickým odkazom podľa normy STN ISO 690, ktoré sa viažu k téme zadania a preukazujú výskumnú povahu problému a jeho aktuálnosť (uveďte všetky potrebné údaje na identifikáciu zdroja, pričom uprednostnite vedecké príspevky v časopisoch a medzinárodných konferenciách)

[4] Nehodiace sa prečiarknite

I hereby declare that this thesis and work described in it are my own work based on consultations and using denoted references.

Bratislava, April 30th, 2019

Lukáš Babula

Acknowledgments

I would first like to express sincere gratitude to my supervisor Kamil Burda for his patience, willing and helpful approach, and very valuable proofreading.

I would like to acknowledge employees of Research Institute of Child Psychology and Psychopathology in Bratislava, especially a physiotherapist Viera Fedorová, for an opportunity to conduct the experiments, their great cooperation and meaningful consultations.

I would like to acknowledge also all participants of the experiments. It was not very short, I know, but I am very glad that you did not give up. Especially my fellow researches, who were first participants of experiments in demonstration versions and helped me with new ideas how to improve the experiments.

I would like to thank my parents and grandparents for their support and patience with me and my unpredictable schedule during this work.

Finally, I would like to specially thank all my friends for tolerating who I have become. You know who you are.

Lukáš Babula

Annotation

Slovak University of Technology in Bratislava
FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES
Degree course: Intelligent Software Systems

Author: Lukáš Babula
Master's thesis: User Model for Determining User's Motor Skills
Supervisor: Kamil Burda
2019, April

Determining the users' skills from human computer interaction can be challenging even if it is the only goal of an application. Users with motor skills and coordination disorders are usually not very alike in their movements, not even when the motor impairments are the same. This problem introduces many difficulties in designing a user model for determining user's motor skills. To avoid confusion about differences of motor skills among similar users we propose explicit specification of motor skill categories. Once the categories are established, it is important to choose the correct approach to adapt the application to regarding the user's restrictions. This can result in opening new interaction possibilities to impaired users who have severe difficulties interacting with the application otherwise. We present three motor skills categories and three gestures to use for category recognition. Recognition of motor skills categories and gesture adaptations to users were put into test in two experiments. The results of the experiments are very favorable and encourage further work in this domain.

Anotácia

Slovenská Technická Univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ
Študijný program: Inteligentné softvérové systémy

Autor: Bc. Lukáš Babula
Diplomová práca: Model používateľa pre určenie jeho motorických schopností
Vedúci diplomovej práce: Ing. Kamil Burda
apríl 2019

Určovanie motorických schopností používateľa na základe jeho interakcie s počítačom môže byť výzvou, aj keď je to určovanie jediným účelom danej aplikácie. Pohyby používateľov, ktorí majú poruchu motorických schopností alebo problémy s koordináciou, sú pomerne rozličné, aj keď trpia tou istou poruchou. Tento fakt má za následok veľa komplikácií a spôsobuje ďalšie prekážky pri návrhu modelu používateľa pre určenie jeho motorických schopností. Aby sme sa vyhli problémom rozmanitosti motorických schopnosti medzi veľmi podobnými používateľmi, navrhujeme zreteľnú špecifikáciu kategórií motorických schopností. Keď sú kategórie presne určené, je dôležité zvoliť správny spôsob toho, ako sa bude aplikácia prispôsobovať používateľom na základe ich obmedzení. Toto prispôsobenie by malo vytvoriť postihnutým používateľom nové možnosti interakcie, pretože bez toho majú veľmi veľké ťažkosti s aplikáciou vôbec interagovať. Za týmto účelom prezentujeme tri rôzne kategórie motorických schopností, ktoré boli použité spolu s našimi troma vlastnými obojručnými gestami. Rozpoznávanie kategórií motorických schopností a následného prispôsobovania sa používateľovi boli otestované v dvoch experimentoch. Výsledky týchto experimentov sú veľmi podnetné a povzbudzujú ďalšiu prácu v rámci tejto domény.

Table of Contents

1 Introduction

Nowadays, several systems and applications are controlled by user gestures. Usage of these gestures offers many useful benefits, because user's gestures are behavioral characteristics. Thus, we can consider them as unique for every user in one or other way. Extracting possible features from gestures and representation of these features is a very wide research area itself. This area covers at least biometrics and user modeling research fields, which are becoming more and more popular with technology evolution.

Users' gestures can be utilized in many different ways by applications. Currently, the most common approach is their usage for interaction, albeit usage of users' gestures for user authentication or user identification during interaction are arising. Regardless of the gesture usage, the gesture needs to be recognized by the application. The captured gesture can be compared with gestures known by the application and then possibly recognized as one of the supported gestures. The captured gesture can be also assigned to a particular user or a group of users. This is how the user or the group can be modeled. User modeling based on gestures is currently less developed than a gesture modeling of the user.

Many problems related to gesture representation can occur during the usage of applications based on a gesture interaction. Such an application can and usually does react slightly differently to each user. These differences are projected into the differences of controlling the application by different users, despite them controlling the same application. Due to the differences, users may experience different levels of difficulty in interaction by different gestures. Therefore, the application can constrain users based on their physique. A naïve example could be requesting continuity of some gesture execution for a very large horizontal distance, such as swipe with hand for at least 100 cm in one sway. This could be a problem for people without arms long enough to reach the given distance. Nonetheless, another problem can also be the difference of motor skills among users. From the interaction point of view, it could be a very difficult and serious challenge, because a user with a higher level of motor skills can perform gestures much easier than a user with worse motor skills.

In order to remedy or at least partly eliminate this problem caused by different motor skills of users, applications can introduce gesture adaptation whilst the gesture is being recognized. There are many possible adaptations, such as the change of the whole gesture, or a simple adaptation of a threshold necessary to accept a gesture with regards to the chosen evaluation. Some corrections can be executed directly by data sensor (device observing gestures) while processing the raw data (sensing movements of a user). This correction is usually sufficient for regular users, because the sensor can handle users' small differences (Bigdelou et al., 2012). Though many problems are under circumstances, where user has rapidly worse motor skills (for instance user has some physical impairment such as cerebral palsy).

Most of users are, naturally, healthy, so it is not common for applications to bother with users outside this bracket. We want to discover, whether it is even possible for an application to recognize different motor skills among its users. To be able to do this, the application has to be first exposed to several different users with different motor skills and be explicitly told which user belongs to which category. Therefore, categories of different motor skills must be established first. Not every hand gesture is appropriate for such recognition, so gestures the application must be designed right after with regards to motor skills categories. With the categories and the gestures, it also takes a lot of experimenting in order to make any conclusion about ability to recognize users and adapt to them.

Analysis of motor skills, their possible applications in virtual reality, gesture recognition and adaptation is described in chapter 2 based on related work in similar research fields. The domain of biometric user modeling and classification is described in chapter 3. Chapter 4 concludes the analysis of related work discussing different sensing devices which can be used for sensing gestures.

Chapter 5 introduces hypothesis of this work in form of our assumptions based on the detailed analysis. These assumptions are the basis for our proposed system in chapter 6. This proposal consists of all parts of solution needed to evaluate the hypothesis. The specification of motor skills categories, gesture types, experiments or adaptations of the system are all proposed in this chapter. The following chapter 7 detailly describes the realization of the all parts of the proposed system.

Evaluation of the proposed system is summarized in chapter 8. The chapter discusses evaluation of each part of the system and also presents results achieved using the system. Final chapter 9 offers a conclusion of our work and hands over future work with motor skills determination.

2 Motor Skills and Virtual Reality

Virtual reality is defined as an immersive and interactive system that provides users with the illusion of entering a virtual world (Heim, 2000). The user is connected to the virtual reality system as part of the input/output loop, allowing individuals to provide input to the virtual environment and experience the result of that input (Lange et al., 2010). A user can provide the input in many different ways, from regular computer input devices, through specialized input devices for virtual reality and various gestures up to the eye movement (Deng et al., 2010).

None of these interactive approaches is new, albeit the Electro-Oculography based approaches are still not very widely used nowadays. The most important approach for obtaining motor skills of users is the one based on gestures. Closer to look to gesture more specifically, as a hand gestures, there is the obvious benefit, such that humans already use their hands and fingers in different situations to manipulate real world objects (Aslan et al., 2014).

2.1 Gestures

“An important distinction is that between *explicit* (or *control*) gestures – when their purpose is to provide some form of input to the computer, such as a command – and *implicit* gestures – when they are exploited to obtain indirect information about the user and his or her environment, such as activity recognition” (Nugrahaningsih et al., 2015). This is an important distinction in the context of this work since the motor skills levels could be obtained from these types of gestures, because they may require certain coordination skills. In addition, both types mostly apply to hand gesture recognition.

2.1.1 Gesture Recognition

Gesture recognition topic is currently very active and widely discussed within research field. In order to solve the problem of huge differences between users’ motor skills it is necessary to focus on hand gestures, because movements of hands are providing the best insight of users’ motor skills.

Two main approaches exist to hand gesture recognition, depending on the kind of sensor employed for data acquisition: techniques which require the user to move a physical object (e.g. an accelerometer) and vision-based methods in which hand movements are detected with one or more cameras. The second case is more complex, but fortunately there are now sensors, which greatly simplify the tasks of hand recognition and tracking (Nugrahaningsih et al., 2015).

There are many approaches how to obtain data of users’ hands and how to recognize gestures from the data. Currently the most widely used sensors are Leap Motion and

Kinect. However, their approaches differ as well despite their similarities. Leap Motion models hands and gestures from hand positions and space orientation while Kinect models the depth of objects. Both sensors provide satisfying results, but when combined, they can recognize even static gestures in real time. This combined method can provide a precise description of users' hands. This description is needed in order to build an adequate user model (Marin et al., 2014).

Thus, the biggest challenge is to obtain hand data and recognize users' gestures as precisely as possible (Manresa et al., 2005a). Very closely related to this challenge is to match the recognized gesture to its user and therefore to recognize both the gesture and the user. In a similar manner, groups of users can be recognized. Different gestures underwent static user verification based on their hand gestures. Results of this experiment are very promising (Imura and Hosobe, 2016a). Gestures used in this experiment were not interactive, so the promising results and very high success rate around 90% can be misleading for authentication based on interactive hand gestures.

Gesture recognition is nowadays executed under various conditions and many application domains. One of them is a low-cost solution for clean rooms, where systems need to recognize users by their gestures (Aslan et al., 2014). Authors discuss many possibilities and approaches to implement the whole system for clean rooms including interaction and user authentication. Another approach was proposed using viewport-independent hand gesture recognition (Jiang et al., 2014), but the proposed approach to improve gesture recognition is sensor-dependent. A very nice survey about gesture recognition was done in order to improve human computer interaction itself (Rautaray and Agrawal, 2015).

2.1.2 Gesture Adaptation

Gesture adaptation to a user using device with an accelerometer is another interesting concept in this research field. Adapted gestures and their complexity are suitable for every user. This approach uses user identification and user authentication based on these adapted gestures (Liu et al., 2009). In the context of motor skills, it is necessary to perform user identification the other way around, because the user needs to be recognized first for proper user-based gesture adaptation. This can be achieved by making use of more features obtained from the user's hand, which can altogether provide much more specific data than accelerometer.

A well-suited gesture for an interactive activity can be totally contrasting among the users, because users are trying to achieve the same functionality through different gestures, which are more natural for each user. The users' goal is the same, but their gestures are not alike. Thus, it is difficult to map certain system functionality to specific gestures unless the gestures are very basic and well known. An interactive system built on the users' gesture recognition has to adapt itself to every user, because every user is unique. The most important differences are users' motility and quickness. This whole concept is based on user identification, where the user is recognized by the system and was registered in advance. The registration builds and stores user models, so the users can be identified using these models (Zhang et al., 2014).

2.1.3 Gesture Interaction

Usability of human computer interaction via users' hand gestures and its real-time evaluation are mostly dependent on a concrete implementation of such an interactive system. A comparison of several interaction methods (e.g. navigation or manipulation) and data obtained by them appear to have promising results in favor of using hand gestures (Cabral et al., 2005a).

Interactive hand gestures in the identification context achieved fairly good results despite a small sample used for training and user modeling. During closed-set identification a model of a user who is being recognized is already known. Therefore, it is enough to compare the current user with all the user models stored in the database, because it is guaranteed that model of current user is stored in the database. Closed-set identification is much simpler than the identification of a new user who is using the system for the first time. Of course, it is not possible to identify the user precisely, but one can consider identifying a user group to which the current user belongs, e.g. group of users with similar levels of motor skills.

2.2 Motor Impairments and Interaction

A recent study (Caro, 2014) had been researching the improvements of motor skills of people with coordination disorder, also known as *dyspraxia*. A game was developed for children with this disorder. During the game the players had to execute multiple coordination-based movements. These movements were aimed specially at dyspraxia, so the exercise was enjoyable, because it helped them to improve in game, but useful, because at the same time their motor skills were improving.

Gestures used for the application were interactive, so the children were able to control the entire application only by their gestures. This example shows that people with dyspraxia do not have to be excluded from interactive applications in which hand gestures are required. However, the application needs to be adapted for this case and has to both support these special users and be able to adapt to them. The application itself was rather simple and aimed for children, hence many problems can theoretically emerge using more advanced interaction or more complex gestures (Caro, 2014).

Children's motor skills also underwent research based on gaming applications regardless of coordination movements (Landry et al., 2013). These gaming applications were also controlled by interactive user gestures, but system obtained all children movements. Movements required to control applications were designed by a specialist in order to achieve the highest variability and to be the most beneficial for children. A level of motor skills was computed from a number of distinct movements and different positions assuming that if the child did not perform a movement necessary to control the application, then the child would not be able to perform the movement in real life.

Determining motor skills via applications is not that rare nowadays as it used to be (Singh and Aggarwal, 2016). The authors were discovering motor skills of vocational workers whose job required a certain level of motor skills. The work was focused on the fine motor skills, which are much more difficult to measure. Touch screen was chosen for measurements, because it is easier to obtain precise information from a touch screen than from sensors monitoring the space. Users were executing specific movements with fingers on the screen surface. These movements were designed for obtaining the motor skills and not for interaction, so it was not possible to control the application with them. Nevertheless, authors were also discussing coordination and gross motor skills. In this manner they used gyroscope and accelerometer of the device with the touch screen. The application was designed to include several shoulder and elbow movements, acting as the source of gross motor skills information and coordination of both hands. The level of motor skills was approximated using linear regression and Support Vector Machine. The success rate of workers was compared with the results of manual tests driven by physiotherapy specialists.

“The games and virtual environments must allow the user to interact in a way that is appropriate for their level of impairment, and must be easily changed to increase the level of challenge as the user improves” (Lange et al., 2010).

The gesture recognizer dynamically analyzes a subset of all body skeleton data detected by the sensing device (considering only information related to torso and upwards joints) and progressively checks if such data correspond to a gesture included in the set of the possible ones (Valoriani, 2013). From the obtained information, it is possible to post-process the gesture and perform gesture management, which can be considered as adaptation to the current user. The way the gesture is interpreted and executed is not given beforehand, so it can be well suited for every user.

2.2.1 Exergames

Interesting connection between physical activity of users and virtual reality is the idea of exergaming. Exergaming was originally conceived as the use of video games in an exercise activity (Sinclair et al., 2007), but nowadays there are many applications which are more or less connecting field of exergaming with virtual reality (Altanis et al., 2013; Caro, 2014; Landry et al., 2013; Tanaka et al., 2012; Yoo et al., 2017). This adds a new dimension to exergaming, which was originally considered to have limited impact to a user (Hsu et al., 2011; Wollersheim et al., 2010).

Exergames can also be easily connected with rehabilitation, which can result in an important new usage of virtual reality (Tanaka et al., 2012). The authors are researching the possibilities of exergames usage within the rehabilitation framework, which turn out as a fairly feasible.

2.2.2 Rehabilitation in Virtual Reality

Early attempts for rehabilitation within virtual reality relevant to context of our work were using an interaction glove (i.e. CyberGlove or a Rutgers Master II-ND haptic glove), allowing to process data from users' hands (Adamovich et al., 2004; Baker et al., 2004). These approaches are indicating potential feasibility of exercise systems in the virtual reality in order to rehabilitate the users' hand disfunction.

On the other hand, there is a number of works and researches trying to rehabilitate problems caused by *cerebral palsy* (Chang et al., 2013; Huang, 2011; Mousavi Hondori and Khademi, 2014; Oliveira et al., 2016). Cerebral palsy has many different impacts on human body, which vary from person to person, because they are mostly individual. Problems caused are mostly motor skills problems, and this is where rehabilitation takes its place. Since the problems differ, the solutions proposed by authors are different and authors also introduce different approaches to achieve such rehabilitation.

Interaction with hand and foot gesture is another approach to rehabilitation using virtual reality (Lv and Li, 2015). This particular touch-less approach is aiming to the rehabilitation achieved by stretching hand and legs whilst using the application and manipulating with it.

2.3 Adaptation

Last but not least, there is an application adaptation approach arising among approaches associated with our field of study. Such an adaptation can be done in multiple ways based on multiple different areas. One of the adaptations which are consistent to previous fields is adaptation of an exergame to its user (Yoo et al., 2017). This is how personalized exergames can be made. It can help users to enjoy the game more, or exercise more according to their skills.

Psychomotor activity modeling, proposed in (Santos, 2017; Santos and Eddy, 2017), is another part of adaptation approach because of the model needed to execute user prediction. Prediction can be built up on the predictive performance of such a model (Pelánek, 2017). This can also be considered as an evaluation overview for the application, because the success rate of user's prediction is the rate of application successful adaptation. It is simply because the adaptation cannot be done after a gesture had been executed, but the gesture needs to be predicted and adapted while not executed by the application.

3 User Model and Classification

For classification of user models and determining whether a user model belong to one class or another, it is crucial to build a proper user model first. Then it is important to know the purpose of classification, which can be useful in choosing a proper classifier.

3.1 User Modeling

User modeling is process of creating a computer representation of user (Fischer, 2001). The computer cannot recognize a person the same way as a human, because it is not able to perceive all the person's features. With the purpose of helping computer to recognize the person, we need to simplify the person's features and select the most varying features among all users for purpose of recognition by the computer. The most common case of person recognition by computer is user recognition.

The list of features, or a computer representation of the user, is the user model (Allen, 1997). The user model should be independent from a concept of the system within user's mind, because it could decrease the quality of such a user model. Once the user knows how they are represented, they are influenced by this information and can act differently.

3.2 Biometrics

Biometrics, also known as biometric recognition, is a science of discovering a persons' identity from their physiological or behavioral features (Jain et al., 2011). The main idea of biometrics is that a person's identity can be uniquely determined by certain features or their combination, e.g. fingerprint or face. One cannot select a single feature that can fulfill all possible scenarios, because different features are suitable for different usages. Selecting proper features is tightly related to the purpose of the application (Jain et al., 2004).

In the context of determining user's motor skills, it is desirable to consider behavioral biometrics, because features obtained from a user's hand are dynamic and exhibit the user's unique behavior.

3.2.1 Biometric System

A biometric system is any system which can distinguish different user models. The system requires three modules to be able to compare user models (Bolle et al., 2013):

- biometric sensor – a device obtaining raw data from the user;

- feature extractor – a unit processing the raw data from the sensor, obtaining features from them as a vector and either inserting the instance of a user model represented by the vector into the model database or returning the model for comparison;
- model database – storage of all known user models.

With these modules, the system is able to perform user model comparison, because when it is needed, the system can build the user model of the current user and get all known user models from model database (Jain et al., 2007). There are many ways to compare models or to assign identity to the current user depending on purpose of the biometric system, i.e. identification or verification.

3.3 Verification

A biometric system needs to decide whether the user is truly the one they claim to be. Thus, during the verification process the user has to explicitly proclaim identity to be verified. A user model of the proclaimed user is then selected from the model database and compared with the user model obtained from the current user. If these two models are similar, the proclaimed identity is confirmed, otherwise the access for the current user is denied and the user is considered to be an impostor. Verification is usually used in applications which need protect their content from unauthorized users (Jain et al., 2011; Wayman, 2015).

3.4 Identification

User identification is the recognition of a user's identity by a biometric system without any identity proclaimed by user. The biometric system in this case does not have a specified user model to compare the obtained model of the current user with. It implies that system needs to fetch and compare all user models from the database against the current user model (Jain et al., 2000).

Identification can be divided into two types (Wayman, 2015):

- *open-set identification* – the current user can be unknown by the biometric system, so the system can result in an unsuccessful identification if none of the user models from database are not similar enough;
- *closed-set identification* – the current user is always known by the biometric system, so the system has to select the most similar user model from the model database, even though none of the stored models may be similar.

3.5 Classification

During the process of user identification, the current user model has to be compared with all user models stored in the model database. To assign a user's identity to the current user model, the most common method is classification. In the machine learning terminology, users' identities are labels or classes. Each user identity corresponds to one class. When a biometric system identifies a user model, the corresponding class is assigned to the user model (Aggarwal, 2014).

Classes in general do not have to necessarily represent user models, they can represent almost anything that can be divided into multiple distinct categories, i.e. groups of users according to their category of motor skills. That is why the classification is a very general and powerful approach. Classification was designed for distinguishing two classes, but nowadays many classifiers can handle multiple classes as well, usually by generalization of the original method (Aggarwal, 2014).

In order to set the classifiers properly, the main properties and parameters have to be settled. Choosing kernel, distance measure, depth or any other key characteristic is crucial and has severe impact on results of classifier. The values are usually determined using a validation set or using cross-validation (Aly, 2005).

For classification it is necessary to have instance of a user model, which belong to concrete user, whose identity or attributes we are trying to classify. These instances are referred as *samples* in machine learning. They are basically vectors of features, which were extracted during a user modeling. These samples can be processed by classifiers for training and testing purposes. The brief summary of the most commonly used classifiers follows.

3.5.1 Decision Trees

Decision trees are considered to be a powerful method for classification. They can be divided into two main categories: classification and regression trees. Two most adopted algorithms for building decision trees currently are ID3 and C4.5. During a decision tree construction, building algorithm has to develop a split the dataset according to its values for every feature to create branches leading to the class labels. The split represented by every node is typically based on the maximum information gain, but it can vary depending on the building algorithm. The feature giving the desirable gain is chosen to be split by. When the algorithm proceeds the end of the branch – leaf of the tree – a class of the unknown sample is unambiguously assigned. In order to assign class to the sample, it has to follow a path from root of the tree to any leaf of the decision tree making a decision at each node according to splitting feature of such a sample.

The algorithms for decision tree building and searching are very well applicable also to multiple class classification problems, because the class label stored in the leaf can be one out of any number of classes concerned. (Aly, 2005).

3.5.2 *k*-Nearest Neighbors

k-Nearest Neighbors (*k*-NN) is one of the oldest classification algorithms not using parametrization. In this algorithm, the key is its distance measure (e.g. Euclidean or Cosine). It is used when classifying an unknown sample to compute the distance from that particular sample to each previously obtained training sample. When distance from each sample is determined, the *k* closest samples are taken into the consideration. The most represented class among them is considered to be the most probable class for unknown sample and it is resulting label of the classification. (Aly, 2005).

3.5.3 Support Vector Machines

"Support Vector Machines are among the most robust and successful classification algorithms " (Aly, 2005). We can distinguish different types according to used kernels. The most widely used kernels are linear and gaussian kernel. The main idea of this classification is to divide the space of samples into two maximizing the minimum distance from one of them to the other. When using linear kernel, the separation is linear (i.e. line for 2D sample or plane for 3D sample), while with gaussian kernel, the algorithm would try to enclose samples of one class from outside from the other one. Therefore, the classical SVM approach supports just binary classification. There are also several strategies available, which can handle multiclass classification as well. First well-known approach is *one-vs.-all* approach, where space is separated to multiple subspaces, one for each class. Second standard approach is *one-vs.-one*, where the problem is split into multiple classification problems and the classes are compared to each other classical way – two at the same time.

4 Sensors

When it comes to the device that can obtain data from user gestures there are theoretically many options because of a technology boom, which is getting more and more intense every year. However according to recent research work in this field, there are three reasonable practical choices:

- *interaction gloves* (Adamovich et al., 2004; Baker et al., 2004),
- *Leap Motion* (Chan et al., 2015; Cui and Sourin, 2014; Marin et al., 2014; Potter et al., 2013) and
- *Kinect* (Altanis et al., 2013; Bigdelou et al., 2012; Chang et al., 2013; Huang, 2011; Jiang et al., 2014; Marin et al., 2014; Mousavi Hondori and Khademi, 2014; Zhang et al., 2014).

Approaches using interaction gloves are not very relevant nowadays, because other named sensors – Leap Motion and Kinect – are able to obtain comparable information from users' hands while being much more convenient, because they do not need any wires or sensors directly attached to the user. Thus, the user perceives greater freedom of movement and it is more comfortable for the user to interact using modern sensors.

Kinect and Leap Motion are both three-dimensional sensors able to obtain data from cameras. Kinect processes and combines information from multiple depth cameras into a depth map (Wilson and Benko, 2010) and Leap Motion is combining grayscale images of two infrared cameras and computing hands and all their part positions (*How Does the Leap Motion Controller Work?*, 2014). The range of the Kinect sensor is much greater than the range of Leap Motion, but it is caused by mentioned differences in cameras and input processing. Both of the sensors are affordable, since they are only typically additional devices for customers wanting to enhance their VR or gaming experience. The price of both sensors is similar, because the core technology – infrared lit scene and multiple cameras perception – is either.

4.1 Kinect

Kinect is a 3D motion sensing input device, which is designed and produced by Microsoft Corporation. First Kinect was introduced in 2010 for gaming console Xbox 360. Kinect monitors the space in front of the sensor itself. Kinect is able to capture objects within distance up to four meters in direction where the front of the sensor is pointing (Wilson and Benko, 2010).

Since its introduction, Kinect drew a lot of attention because of its gaming and research potential. Despite this popularity, Microsoft decided to discontinue support and production of this sensor (Wilson, 2017). Kinect is still available for purchase, but research potential dropped rapidly down, because in near future the availability may change as well as its popularity.

4.1.1 Data Provided by Kinect

Kinect can provide several different types of data obtained from depth cameras. It is possible to receive data on a different level of abstraction, from raw video images up to the human skeletons. Focusing on hand gestures we are interested in hand positions in the 3D space. Hand positions can be accessed from skeleton information, because the coordinate system for the skeleton data is a full 3D system with values in meters (*Kinect Sensor*, 2012).

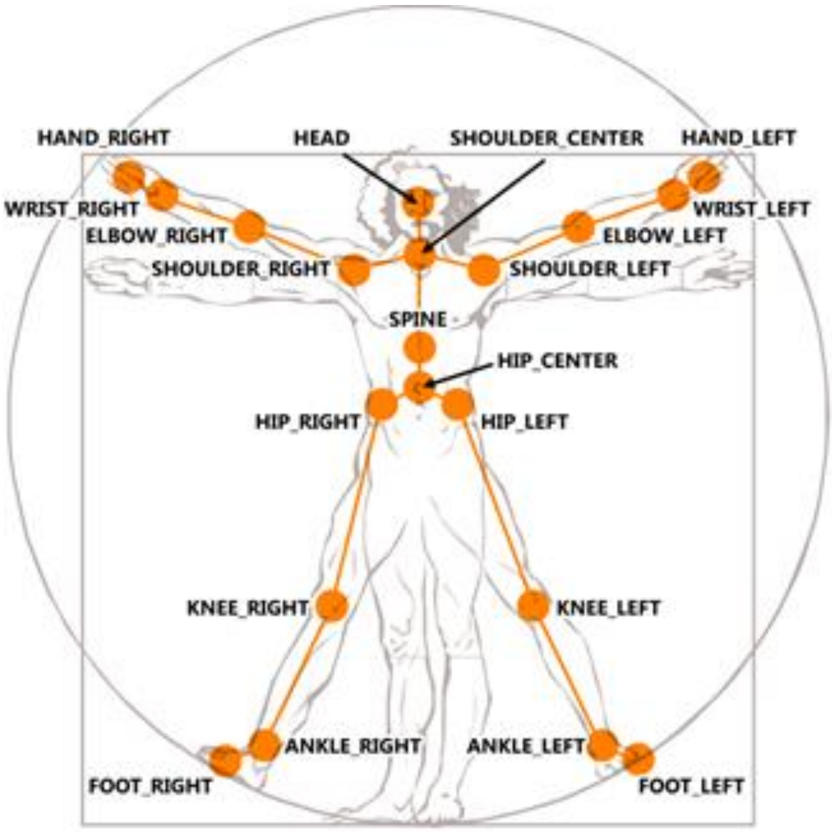


Figure 1. Kinect skeleton representation (Kinect Sensor, 2012) .

From Figure 1 , it is clear that for motor skills classification can be used more than just hands, because suitable information can be also derived from the positions of wrists, elbows and shoulders.

4.2 Leap Motion

Leap Motion is, like Kinect 3D, a motion sensor input device, but it is developed by company Leap Motion, Inc. The range of the sensor is much smaller, only up to 80 centimeters with viewing angle 120°. The range is influenced by powering the sensor via

USB, so three LED infrared lights are able to light only the described interaction area, which is also depicted in Figure 2 (*How Does the Leap Motion Controller Work?*, 2014).

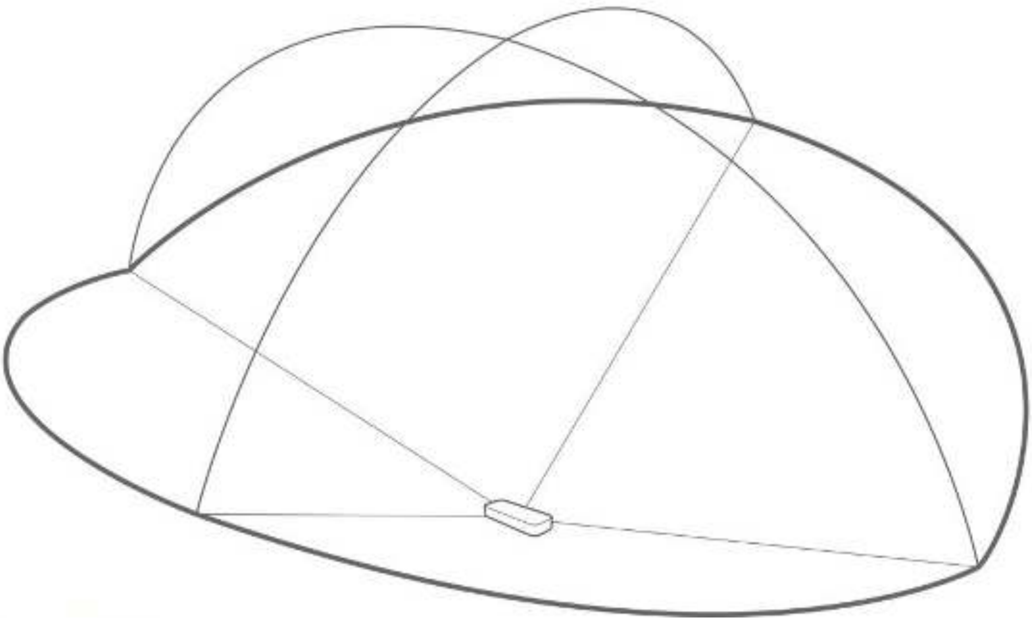


Figure 2. Leap Motion Interaction Area (How Does the Leap Motion Controller Work?, 2014).

4.2.1 Data Provided by Leap Motion

Leap Motion is able to transform data from grayscale stereo video image into hand positions (*Image API Now Available for v2 Tracking Beta*, 2014). In comparison with Kinect, Leap Motion provides data at a much finer level of detail, because it is able to provide positions of fingers, bones and joints. It is also possible to obtain information about wrists.

For every part of each hand, the position, direction and velocity are provided by Leap Motion. For every bone, Leap Motion also provides its length and width. Additionally, the sensor tracks the palm, providing palm position, palm direction, palm width and palm normal.

5 Assumptions

According to the domain analysis, people with a certain degree of motor impairment have difficulties performing interaction with hand gestures, have to overcome disadvantages while using these systems and sometimes it is totally impossible for them to use such an application at all. In order to help these people with using hand interactive applications, an application needs to recognize type of motor impairment of the user and classify user behavior with respect to supported motor disabilities as well as adapt the interface according to the recognized motor impairment. We assume that it is possible and feasible to:

- Classify a person's motor impairment with Leap Motion sensor with respect to specified categories of motor impairment from user's coordination-based gesture execution. This coordination-based gesture in the context of this work can either be a rotate gesture, scale gesture or a carry gesture, all of which are executed with both hands.
- Classify a person's motor impairment with Leap Motion sensor as described above in real time.
- Adapt hand gesture-controlled applications for users according to their motor impairment.

It is possible to discover specific users' skills by letting them perform exercises designed to evaluate these skills (Singh and Aggarwal, 2016). Since the coordination gestures we are focusing on are not designed in this or a similar manner, we assume that it is possible to classify users' motor skills on a higher level of abstraction. Thus, the application will not obtain exact users' motor skills, but will be able to recognize specific patterns for different types of motor disabilities instead.

6 Proposed Biometric System for Motor Skills Detection

For motor skills detection we need to specify the skills first. Therefore, we propose three categories of motor skills, which are simply describable but at the same time clearly specifying the abstract concept of motor skills.

Collecting data from the Leap Motion sensor for every of these categories is very important for classification of these categories. The data represent users' hands, their movements, positions, directions and other information obtainable from Leap Motion API. Collecting data needs to be performed during the first of users' experiments, which will be also aimed to choose the best classifier in order to allow real-time adaptation of gestures.

Before the classification itself, the raw data need to be processed to extract useful features. These features will be used for training the classifier, resulting in the creation a of user model representing the category of motor skills.

Classification can be done in multiple ways depending on the classifier itself and its hyperparameters as well. We consider a prior selection the best classifier to be very tedious and error-prone, because each classifier is strongly dependent on data, which cannot be described good enough before they will be obtained. Therefore, we propose to experimentally choose the best classifier after comparing their results.

After successful classification and user model creation for every category it is necessary to add real-time identification and gesture adaptation to build the whole system. Afterwards, it is necessary to test this system with users during another experiment. After collecting results of the latter experiment, we compute the classification success rate and discuss effectiveness and efficiency of adaptations after consultations with experiment participants and specialists observing the participants.

6.1 Motor Skills Categories

In order to adapt an application for a user with respect to the user's motor impairment, it is necessary to specify disabilities to be taken into consideration. We propose to distinguish three different categories of users' motor skills:

- motor skills of healthy users,
- motor skills of users with dyspraxia and
- motor skills of users with cerebral palsy.

These three categories can be also categorized as *regular category*, *discoordination category* and *movement-constrained category*, respectively, because the categorization of the impairments is in a simplified manner based on those principles.

6.2 Gestures

For determining a category of motor skills from hand movements, we propose to focus on short and well-defined actions – gestures – of both hands. Obtaining the category could be very difficult if the gestures were considered for only one hand, because there is no adequate response from people with dyspraxia regarding their impairment using only one hand. We propose three different coordinated gestures of both hands.

6.2.1 Rotate Gesture

As the name suggests, the gesture is derived from a real-life hand movement used for tuning a radio using a button or screwing a lightbulb. Hand fingers must surround the object of interest and then turn in the desired direction. For coordination purposes we propose to enhance the gesture to be applicable for both hands at the same time. Once the fingers of each hand surround different objects, the user then performs rotation of both hands at the same time in an outward direction – the right hand is being rotated clockwise and left hand anticlockwise as depicted in Figure 3.

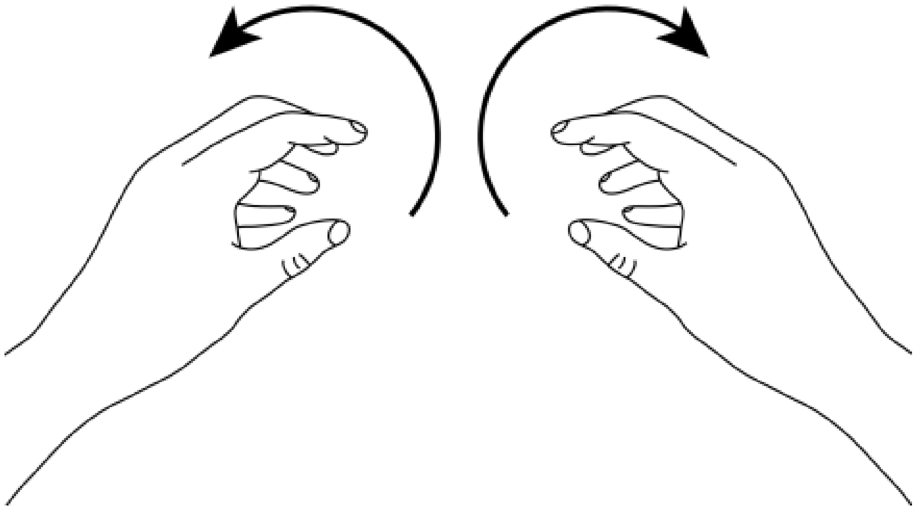


Figure 3. Rotate gesture.

The gesture must be executed at the same time with both hands, what can be guaranteed by comparing the angle of both hands. If the angles are different, the gesture is not accepted. For humans it is very difficult to control hand rotations on the scale of one angle degree. Therefore, a gesture will be accepted if the difference of angles is relatively small to requested angle to execute. It is feasible to require the gesture to be around half of the circle long, because it is comfortably reachable with regards to the hand anatomy.

6.2.2 Scale Gesture

The name of the scale gesture is not based on its real time usage of opening, but on the touchscreen representation of the zooming feature. The gesture in Figure 4 is to be executed with two hands as follows:

- both hands pinch the same object with all fingers,
- hands move in the horizontal direction only and
- hands move a certain distance apart from each other.

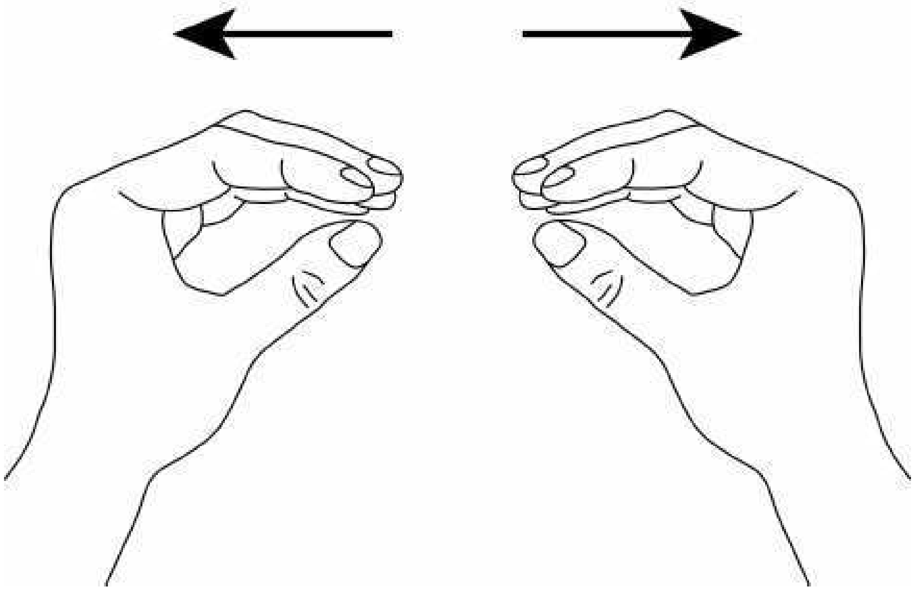


Figure 4. Scale gesture.

Some small fluctuations of vertical movements and forward backward movements are also taken into consideration and will be tolerated in order to make the gesture conveniently executable. To involve coordination, the criterium of regularity of movements with both hands will be similar as in the previous gesture. In this case the distance from either hand to the center of the object must be the same, with a small amount of tolerated difference.

6.2.3 Carry Gesture

Our last proposed coordination gesture is the carry gesture shown in Figure 5. It is the least complex gesture for users to understand, because all of its constraints are real-life based. The gesture simulates carrying an object from the bottom with open hands. Therefore, the hands must stay during the execution of the gesture in a similar distance as they were at the beginning. If the distance is shorter, the hands will pass the center of gravity and the object will fall. If the distance is longer, the hands will be farther apart as the size of the object, resulting in the fall of the object. If the positions of both hands are not horizontal (palms are not facing upwards), the object will slide and fall.

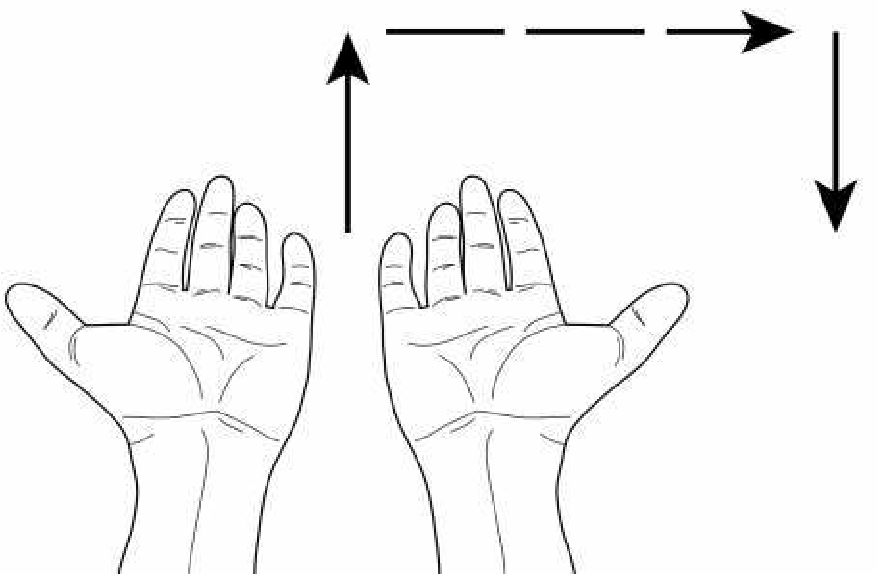


Figure 5. Carry gesture.

The transport distance using the carry gesture is meant to be rather small to allow execution of the gesture while sitting. We exclude other body movements than hand movements from contributing into the carry gesture conditions for our purpose of user modelling.

6.3 Experiments

We propose to perform two experiments:

- Obtaining data from users for choosing the best classifier for our problem and analyze the success rate of the proposed method and
- evaluating obtained results and observe the ability of the system to adapt to the user.

Both experiments are to be performed using the same settings within the same interactive scene, so users cannot without interaction distinguish whether the system is already trying to adapt to them or not. We want to avoid affecting users in order to ask them about their point of view on interaction.

We propose to involve all three selected gestures in a simple game, where users will be distracted by in-game abstractions of problems and effects of their movements and will not concentrate on performing gestures themselves – making gestures means instead of goals. The gamification of experiments also gives users motivation to participate and perform the same gestures over and over, making their progress visible directly by their actions.

6.3.1 Game Logic of Experiments

Involving all the gestures into the interactive game should be very straightforward. Thus, we propose to use the rotation gesture to actually rotate objects in the scene and the carry gesture to carry an object. The nickname for the scale gesture comes from our usage of the gesture in the application, but the gesture is natively used to open sliding doors or curtains. Though, in virtual reality, the gesture is being used for scaling objects and we decided to follow this convention. We do not consider it unintuitive for this use case, because widely spread and well-known zoom gestures on touchscreens employ a very similar logic.

The goal of the game is to build a two-dimensional pyramid (triangular stack) out of three-dimensional cubes. The game consists of constructing cubes of the correct size to be possible to use in building of such a pyramid. Each cube at the beginning is too small to be used in the final pyramid, so it needs to be properly adjusted. Using the scale gesture, we scale one dimension of the cube. To scale the whole cube to a larger size, it is necessary to rotate the cube at least twice using the rotate gesture. When the cube is properly constructed, the cube is placed on a stockpile, from where the cube can be carried to the final pyramid.

For evaluation purposes, we need all three gestures to be represented evenly. To include them approximately evenly into the game, we decided to add one more step, to create a cube to be scaled using the gesture for rotation. Hence, the count of successful scale and rotate gestures is exactly the same. The only deficit is with carry gestures, so we propose to introduce levels in the game, where each successive level involves building an increasingly larger pyramid. It gives us the reason to clear the pyramid building progress – adding a base tile. All cubes from the pyramid thus can be moved to the stockpile pyramid and the user must carry the cubes again into the pyramid being built.

For building the final pyramid with a base consisting of five tiles, the user must create, rotate and properly scale 15 cubes. The minimum requirements for completing a proper cube are to use the rotate gesture 3 times and the scale gesture 3 times, amounting to 45 gestures of each type. From the pyramid base of length 1 up to the base of length 5, the user needs to always carry all cubes required to create the corresponding pyramid. Summing the first five triangular numbers gives 35, which we consider an appropriate number compared to the 45 occurrences of other two gestures.

6.3.2 Selecting Candidates for Experiments

Selecting appropriate candidates for experiments is very important, because the entire classification and evaluation of the proposed system depend on participants of the experiments. There are two approaches to build a good user model for each category: either to have a large number of participants or a relatively small sample of participants, which will be very heterogenous and can satisfactorily cover each category.

Covering the category with a large number of participants is very time-consuming and infeasible, because each category imposes some constraints for participants which have to be fulfilled. In this case it is necessary for participants to distinguish their motor skills category. It is not acceptable to let participants declare their category, because self-estimation of participants' category of motor skills is subjective and varies greatly among the participants. On the other hand, some participants may have already had their motor impairment classified by a specialist.

Consequently, we propose the candidate list to be carefully chosen by specialists to minimize the number of participants while covering the variety of categories.

6.4 User Models and Features

User model in biometric systems is dependent on the users' actions. The actions in a biometric system are gestures proposed earlier in this chapter. We do not want to create a biometric system depending on all of the gestures performed evenly as it is set in our experiments. Thus, we propose three distinct user models, one for each gesture. Whether the system has other gestures or not, independent on their sequences we want to have the user model to be able to be bound to one particular gesture and represent the user only by that gesture.

Instances in user models are represented by feature vectors. In our case it is feasible to propose concrete features in the vectors and then generalize the models out of the obtained data from the first experiment. Consequently, for each gesture it is necessary to have certain features which could represent all the motor skills categories and differentiate between them according to their values.

We identified several common features for all of the proposed gestures. All features marked with asterisk are computed for the left and the right hand separately:

- minimum, maximum and average velocity of palms*
- velocity of palms at beginning and end of the gesture*
- normal of palms at the beginning and end of the gesture*
- direction of hands at the beginning and end of the gesture*
- minimum, maximum and average x, y and z coordinate of positions of palms*
- duration of the gesture

Since values for velocities, normals and directions in real life are three-dimensional, they will be represented as separate features according to their components along the three axes. All rotations will be represented as four components of quaternion representing the rotation in three-dimensional space.

6.4.1 Specific Features for Carry Gesture

The carry gesture has basically determined only the start point and the end point, so the path of transportation could vary not only user to user, but also according to the difficulties with keeping hands coordinated. Specific features for this gesture include:

- minimum, maximum and average distance between palms
- minimum, maximum and average difference of height of palms
- distance travelled by palms*
- minimum, maximum and average distance between normals of palms and vector facing upwards*

6.4.2 Specific Features for Rotate Gesture

The rotate gesture is the only gesture using rotation of hands for its purpose. Thus, we have to obtain information about rotations of hands. Also positions of hands during the gesture can reveal specific information about the user. When rotating, the user is holding an invisible object, therefore the five-finger pinch tightness is very important. Palm can be rotating around any finger, so tracking how much is outermost finger moving is very important. For this gesture specific features include:

- rotation of hands at beginning and in the end of the gesture*
- average distance between thumb and ring finger*
- average distance between palm and ring finger*
- distance travelled by pinky*

6.4.3 Specific Features for Scale Gesture

The scale gesture is the least unique in comparison with the other gesture. We want to enhance the common features only by following additional three groups:

- average distance between thumb and ring finger*
- average distance between palm and ring finger*
- distance travelled by palms*

6.5 Biometric Sensor

After comparing available and suitable sensors in the analysis, we propose to use the Leap Motion sensing device due to its better future support and also the precision of the smaller hand movements and gestures. Since we are focused on the hand gestures, Leap Motion sensor's abilities are very satisfying. Future support allows further future work on this or similar topics of other researchers.

Regarding the Leap Motion abilities, we propose to monitor and store most of the available data the sensor provides, because we do not to want to lose any opportunities during the feature extraction or data analysis stage. These values will be stored in a file easy to read for feature extraction and data analysis processes. Values which can be easily ignored from the sensor without any loss are attributes of bones in fingers. Attributes of fingers are providing all the necessary information in our case, because they fairly precisely summarize and represent the whole fingers, thus information about each bone are not needed.

The frequency of providing data by the sensor depends on the number of hands the sensor monitors. For two hands, as all the gestures are proposed for, it is approximately 60 times per second. Such a large amount of information is in favor for our usage for very precise tracking of hands in space and time.

6.6 Feature Extractor

The feature extractor will read the raw data obtained from sensing device and compute values of each feature in the vector according to their definition based on the user model. Since the user model is different for each gesture, the feature extractor must be different for each gesture.

It is not desired to process data which are obtained during the time when no gesture was performed. Therefore, during the preprocessing of raw data we make sure that only the relevant data segments are chosen for extraction.

6.6.1 Raw Data Preprocessing

We propose to store also the event progression, so one can easily distinguish whether a specific gesture is being executed or not. Each gesture needs to have its separate progression file with gesture events. We propose the same three events for each gesture:

- start of execution,
- acceptance and
- failure of gesture.

Along with events, a timestamp is included for each entry in the gesture progression file.

6.7 Model Database

After the feature extraction, the vectors of features have to be persisted. The information about class, in our case motor skills level category, must be assigned to each vector. Our proposal is to store these vectors of numbers in comma-separated values form to be easily readable by both human and computer for their analysis and later for use in classifiers.

Each gesture is associated with a separate file, because every gesture is using a different user model. It is also desired to distinguish between successful and failed gesture executions, so we are considering six separate files to be used for storing the vectors of features (i.e. the instances of user models).

6.8 Identification Method

In the second experiment, participants are identified by our proposed system according to their motor skills category. This problem can be represented either as *open-set identification*, where it is possible to classify the user's motor skills as not belonging to any category, or *closed-set identification*, where the system must decide even if none of known categories is very similar to the user's motor skills.

We propose the usage of the closed-set identification, because there is always an action during the adaptation process that needs to be executed. This allows the system to perform adaptation even if the current user's motor skills are in some way different from all of the specified categories, but still mostly similar to one of them, which we consider as a preferred way of handling special cases.

It is crucial to identify a user's motor skills category not only by successfully performed and accepted gestures, but also by failed gestures. If the user is not able to perform gesture properly for multiple times in row, then adaptation should take its place at the time. Thus, it is not feasible wait for classification based on successful gestures. We propose to split these cases and focus heavily on the unsuccessful gestures.

For our proposed *closed-set identification,* we consider choosing best classifier by tuning the parameters of following classifiers in coherence with our domain analysis on classifiers.

- *k*-NN using
    - *k* from one up to a feasible ceiling according to the number of obtained training samples and
    - different distance measures – Euclidean, Manhattan and cosine similarity.
- SVM using

  - linear kernel and
  - gaussian kernel.
- Decision tree using
  - C4.5 and
  - ID3 constructing algorithm.

The parameters explicitly stated for each classifier are not the only parameters to be tuned in fact but are the main parameters available regardless of the implementation or chosen library to perform hyperparameter tuning. We suggest tuning other available parameters of classifiers, such as algorithm learning rate or tolerance for stopping criterion, depending on the implementation.

After the best classifier is determined for all the gestures and both successful and unsuccessful gestures, we propose to properly train them and adjust identification for real-time usage to allow adaptations of the system.

6.9 Adaptation

From experiment observations and data analysis we propose determining the most violated gesture conditions for each gesture and motor skills category. We also propose comparing violated conditions of the active user to the expected conditions belonging to their group. We do not consider generally loosening the conditions if user is not able to meet them. Instead, we propose adapting conditions according to the category and allowing this adaptation only to successfully classified users of the category.

In some cases, the visual representation of conditions could also be helpful, when the user is able to fulfil all of the gesture conditions, but they are unable to do so naturally. Users focusing on their gesture during their performance with qualitative feedback could help them to meet all the conditions without any further adaptation of conditions. However, another visual feedback would not be appropriate in our proposed system, because two out of three gestures already have color and movement feedback, which we consider much more important. Therefore, the system adaptation will have to make do without visual changes relying solely on category-based condition adaptations.

We propose the adaptations to be unnoticeable by users because we want users to interact with the system in the exact same manner, whether it is adapting to them or not. If the user is heathy, the application will not perform the adaptation in any way, even if the user is making no progress, because the user is easily able to.

If the user does not belong to the category of healthy users but is not constrained in any way to perform one of the gestures, they may be misclassified based on their movements for that particular gesture. However, it does not mean the user is able to perform other gestures without any constraints. We propose classifying the user’s motor skills and adapting the system for each gesture separately, even if one of the gestures is classified completely differently than others (e.g. heathy category for carry gesture and cerebral palsy category for scale and rotate gesture). Even if the user is eligible for adaptation according to their real category and classification of other gestures, we do not find it necessary to adapt a gesture that the user can perform without problems.

6.9.1 Adaptation to Users with Dyspraxia

After observations of movements of users with dyspraxia and consultations with a specialist we propose adapting only a strictness of the coordination of both hands. Users in this category are fully able to perform all the required movements but cannot unconsciously perform them with both hands at the same time. Adapted gestures will still require performing all of their originally proposed movements with both hands cooperatively, even if not perfectly coordinated.

6.9.2 Adaptation to Users with Cerebral Palsy

All adaptations made to the gestures for users with dyspraxia are applied and we propose additional adaptations because cerebral palsy imposes stricter constraints on movements. The gestures still need to stay coordinative and involve both hands cooperating on the gestures. Cerebral palsy often does not allow users with this condition to cooperate with both hands evenly. Therefore, we propose adapting gestures such that users must involve both hands but are able to perform mostly with the healthy hand.

7 Realization of Proposed System

The experimental three-dimensional scene is implemented in game engine Unity 2018.3.8f1, what is currently the most recent supported engine for our sensor – Leap Motion. This engine is using assets of Leap Motion 4.0.0, allowing to naturally use and render hand movements in scene built in Unity. Therefore, all of the functionality is currently implemented in order to be usable within the scene, which is the core of the entire system.

Unity allows to use three different scripting languages: C#, UnityScript (JavaScript for Unity) and Boo. The entire solution except classification is implemented in C#, since all the Leap Motion assets are in C#, and it is the only language usable for combining these technologies. Classification itself is performed outside of this framework using the *scikit-learn* library in Python programing language.

7.1 Scene of Experiments

In order to make the scene as clear as possible and easy to understand, we decided to visually divide areas of the scene for all the gestures. In the Figure 6 it is also possible to see that the whole interaction area is divided into two stacks of cubes. The interaction area consists of three parts, one for each gesture.

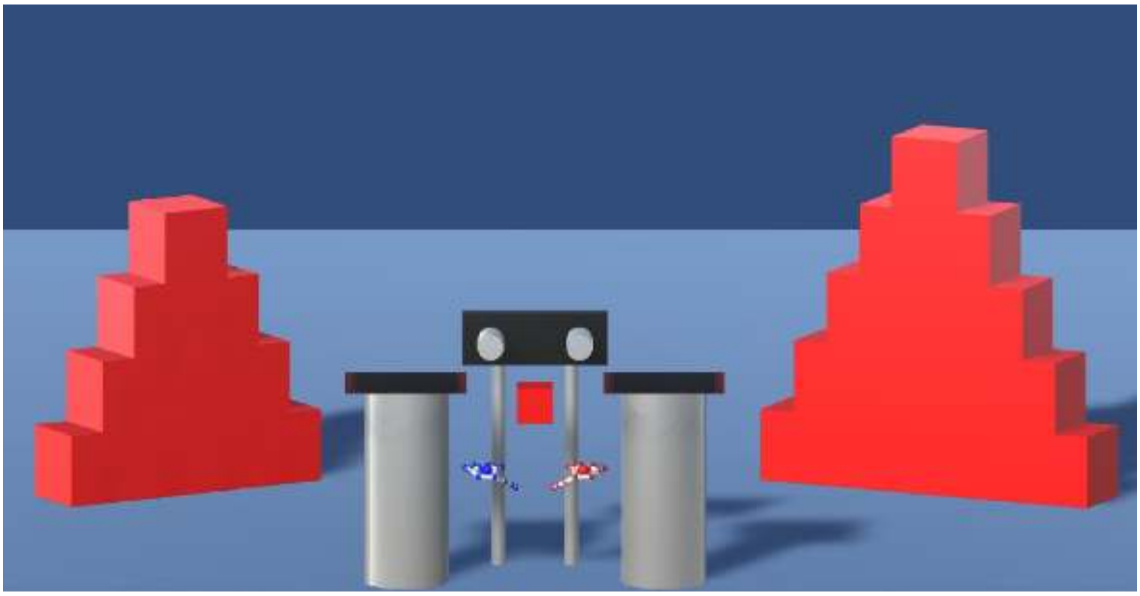


Figure 6. Placement of game objects in the scene of experiments.

There are two pedestals with platforms for carrying the cubes in the Figure 6. The left pedestal is where a cube will be spawned after its creation. The right pedestal is the receiving pedestal, from where the cube will be teleported to the final pyramid on the right-hand side. The small cube in the middle is the area for scale interactions, where it is needed to pinch the cube and scale each side to the bigger size. For the rotation gesture there is area for the two buttons on the box with two narrow legs. These legs were added after initial feedback, because it was very difficult to estimate the depth of a floating box in the air.

7.2 Gestures

The proposed gestures suitable for our use are very specific and cannot be found in the generic gesture library of Leap Motion classes, so their recognition must be implemented almost from scratch. However, an abstract class providing access to the hands and their attributes was available and proved very useful for the gesture wrapping. The abstract class could not help with evaluating our specific conditions of gestures, though.

7.2.1 Scale Gesture

The implementation of the scale gesture is based on the principle that scaling is starting directly from the small range around the center of the cube to scale. If the cube is not scaled yet, the range is exactly of the size of the unscaled cube. This range remains the same throughout the whole process. Therefore, even the last scaling gesture must be started from the center of the cube, despite the fact that much larger area is visible on the sides. The side of the small cube is three centimeters long.

After positioning the five-finger pinches of both hands into the sides of the range, the gesture is activated. Then the hands must stay in the horizontal tube along the *x-axis* of the cube. Hands must move at least 20 centimeters each, while the difference between these movements cannot be more than 7 centimeters. While this threshold might seem quite large at first, we had to take extra space into consideration to account for sensor errors and imperfections. If each hand is dislocated only very slightly (1 cm), then the threshold is shrunk to only 5 centimeters. If hands are moving fast, the dislocations cannot be avoided by the hardware and occur regularly.

7.2.2 Rotate Gesture

The rotate gestures of both hands are performed on the radio buttons in the center of the scene. That means that the distance between the left and the right hand and the left and the right button, respectively, must be small, indicating that the hand is close to the button. Each button must be touched by at least four fingertips. The fingertips must either collide with the surface of the button or be positioned inside the button. Since the button is not a physical object, it is not possible to properly touch the object, so positioning the fingers inside only indicates a tighter pinch of the fingers. Also, only four fingers are required for the same reason. The four fingers must stick to each button during the whole execution and the closeness of the hands to the correct button must be fulfilled.

The rotation angle of the gesture is 90 degrees for each hand. The difference cannot be more than 30 degrees between the rotations of both hands. The right hand is rotating clockwise and the left hand anticlockwise. A challenge in implementation was to compute the actual rotation of the hand, because after reaching 360 degrees the rotation continues from zero. It was solved by shifting the rotations by 120 degrees from zero. Thus, the hands do not rotate over the zero value anymore.

7.2.3 Carry Gesture

The beginning and the end of the carry gesture are defined exactly the same, only on different platforms. The platform on the left pedestal is designated for the beginning and the right for the end. Both hands cannot be farther than 18 centimeters from the center of the platform in any direction. Furthermore, in the *y-axis* direction it cannot be more than three centimeters, so platform and hands are on the same vertical level.

During the whole gesture, both hands must be no more than 40 centimeters apart and at the same time no less than 20 centimeters apart. These distances represent the size of the cube, so the cube will appear to fall through if the hands are more than the maximum distance apart. The hands also must stay at the same vertical level of difference no more than 5 centimeters, because otherwise the cube could slide and fall. Both hands also must face upwards the whole time. This is achieved with the normal of each hand and its distance from a vector facing directly upwards. If the distance is under the experimentally chosen threshold, the hand is facing upwards enough to accept the gesture. The threshold was chosen to represent approximately 15 degrees of tolerance between the hand normal and the vector facing upwards.

7.2.4 Visual Feedback of Gestures

Some of the conditions for gestures are easier to fulfil than others. To provide feedback to the user, if their gesture is still active or already failed, we decided to implement a color scheme for cubes. While the scale and rotate gestures are inactive, the cube is red. While being active, the color turns yellow. After a failure of gesture, the color returns to red and after the acceptance of the gesture the cube turns green for a short time to give a signal of gesture success and then defaults to red. These two gestures also have feedback in the form of partly executing their actions – during the performance of the gestures, the cube is being partly rotated as the hands are being rotated or scaled as the hands are moving apart.

The feedback for the carry gesture is simpler. When the gesture is active, the cube is moving with the hands as if carrying the cube in real life. After a carry gesture failure, the cube disappears from hands and the cube is respawned on the left pedestal's platform, representing the actual fall of the cube.

7.3 Raw Data Logging

Data obtained from the Leap Motion sensor during each scene update (performed automatically by the Unity engine approximately 60 times per second) are written into the comma-separated values file in following form and order:

- timestamp,
- hand ID,
- boolean representing whether the hand is right
- palm width,
- palm position (x, y and z),
- palm normal (x, y and z),
- palm velocity (x, y and z),
- hand direction (x, y and z),
- wrist position (x, y and z),
- hand rotation (w, x, y and z).

Each finger has also its own features logged:

- length,
- width,
- tip position (x, y and z),
- direction (x, y and z).

For the usability of the dataset in different contexts we decided to include physiological features of the hand – the length and width of each bone in a hand to the end of each line.

7.4 Gesture Events Logging

Information about gestures is also logged in the comma-separated values format. Each gesture has its own log file with only two columns: event timestamp and event ID, where the event ID is for all the gestures defined as follows.

- 0 for the beginning of the gesture;
- 1 for the acceptance of the gesture;
- 2 for the failure of the gesture.

7.5 Feature Extraction

Feature extraction is implemented and used real-time, because motor skills categories are being recognized for each unsuccessful gesture using classifiers immediately after its failure point. In every scene update performed by the Unity engine, all raw data are persisted into the comma-separated values file. We had two options how to gather all necessary raw data and extract features from them:

- read the file with persisted raw data after the gesture failure according to the event timestamps (beginning of the gesture and failure of the gesture) or
- start temporarily storing only the necessary raw data for the ongoing gesture during each scene update into the memory and stop storing the data as soon as the gesture fails.

Due to inconveniences associated with reading files while being written to, we decided to use the latter approach. Also, we do not have to store all the raw data, because it contains data which we are not interested in our work, but they are still being persisted for potential future usage of the dataset. The downside of this approach is, however, that the system to keep in the memory necessary raw data during every gesture, even if it is in the end successful. The system is easily capable to do so, and it is very simple to delete everything when the gesture is successful.

The extracted features are then persisted into a separate file without any further modifications or transformations. Scaling or standardization is applied afterwards and only in the classification stage.

7.6 Classification

For real-time adaptation, our system needs to perform real-time classification of the motor skills category based on the gestures performed by users. Classification is not performed within the Unity framework supporting C# programing language, because this programing language does not offer any suitable library for data science. Therefore, we decided to use the Python programing language providing the *scikit-learn* library and perform classification in this external environment.

This Python program accepts feature vectors on standard input and returns the class label on output. In C# we can easily communicate these necessaries between the Unity framework and this external Python program if we treat the program as an external process. Each gesture has own classification program, and therefore has own external classification process. After each failed gesture is this external process triggered with feature vector, and the program runs preprocessing and classification on the feature vector. The result is then received by Unity framework and passed to the corresponding gesture script to enable adaptations in the scene.

The Python program usable in this manner must be standalone and runnable without any dependencies. Since the training of a classifier takes a very long time and we have three external processes with one classifier each running on startup, we trained classifiers beforehand and persisted their trained representations. On every startup these classifiers are loaded, so the Unity engine does not have to wait a significant amount of time and postpone the startup of interactive modules. The external processes are running all the time from the startup of the application until its shutdown and passively waiting for feature vectors to be classified.

7.6.1 Preprocessing

After the first experiment focused on obtaining data from users, we completed our training dataset. Our preprocessing is very simple, because all our features are numeric. Although, there are negative values in our training dataset. The values come from the sensor; therefore, we expected the values in future be out of the range of values collected in the training dataset. These reasons led us into conclusion that the Z-Score would be a very appropriate preprocessing step for our data.

Our feature vectors are not very large, but they are not very small either. Feature vectors for scale gesture, carry gesture and rotate gesture consist of 79, 87 and 95 features, respectively. A performance of training process with the entire feature vector is pretty good, but we tried to reduce dimensionality due to the fact that some of the features in the feature vector may be linearly dependent. For this purpose, we tried to use Principal Components Analysis, but the less data we used, the significantly worse were the results achieved. Therefore, we kept all features in each feature vector.

7.7 Adaptation

The adaptation is applied for each gesture separately. In the implementation for the second experiment the adaptation was carried out after each failed gesture. Every adaptation was based only on the one last failed gesture, therefore any previous adaptations of the same gesture for the same user were not taken into any consideration. Later after evaluating the second experiment we, however, found out, that it would be more appropriate to take the most recent gestures into consideration and apply the adaptation on multiple failed gestures rather than a single gesture.

After each gesture failure, the vector of features undergoes preprocessing and classification. When the class is assigned, the system passes the class only to the script handling that type of gestures. This script stores the class and uses it within every gesture recognition until another class is passed to the script. The script for each gesture is handling the class in own way with regards to the original gesture specifications and success conditions. However, there is one basic concept, that all the gesture scripts follow – all adaptations for users with dyspraxia are used also for users with cerebral palsy. These adaptations can either be used as-is or further improved in order to better fit users with cerebral palsy. Also, no adaptations apply to healthy users. These adaptations were designed to fit users' needs in controlling the application. Users are therefore not presented these adaptations and are not aware of them. Their interaction with application from their point of view should stay exactly the same.

7.7.1 Adaptation of Rotate Gesture

There are two adaptations for users with dyspraxia. The first adaptation is enlarging the buttons where the gesture is performed. Visibly there is no change, but the colliders of both buttons are of 120 % size compared to the original. The second adaptation consists of removing the threshold of maximum difference between the angles of rotations of both hands. Both hands must stay on the buttons during the whole gesture and both hands must rotate at least 90 degrees.

The adaptation for users with cerebral palsy is also in enlarging the buttons, but the main adaptation is that only one hand is required to perform 90 degrees rotation. The other hand must be on the button the whole time but can rotate in opposite direction or not rotate at all.

7.7.2 Adaptation of Scale Gesture

The adaptation for users with dyspraxia consists of removing the threshold of maximum difference between movements of both hands. Both hands still must move at least 20 centimeters away from the center of scaling cube. Also, they both must stay in the horizontal tube along the x-axis of the cube. The adaptation for users with cerebral palsy is very similar to the previous adaptation, but instead of both hands moving at least 20 centimeters, it is enough if one hand travels the required distance, while the other stays in the horizontal tube.

7.7.3 Adaptation of Carry Gesture

There are overall four adaptations of this gesture. Three of them apply to both categories and the last adaptation only to users with cerebral palsy. For users with cerebral palsy it is very difficult to keep the impaired hand upwards together with moving it with the other. Therefore, we decided to require only one hand to be facing upwards for this gesture.

The other three adaptations modify thresholds for both categories. Thresholds of maximum distance between hands during the gesture and distance between normal vector of palm and vector facing upwards were both modified to 150 % of their original values. The threshold of minimum distance between hands was modified to be 75 % of its original value.

8 Evaluation of the System

The proposed system was continuously undergoing evaluation of some parts and want through further evaluation alongside with the second experiment and calculating its results. For the evaluation different approaches are involved according to their suitability.

8.1 Motor Skills Categories

We attended a meeting with a professional physical therapist from the Research Institute of Child Psychology and Psychopathology in Bratislava in order to evaluate our proposals and ideas and to gain inspiration and feedback for improvements. Regarding the recognition of motor skills categories, our initial proposal was met with understanding from the specialist and was regarded as very feasible to use in our context.

8.2 Gestures

We distinguish two approaches to the evaluation of gestures, because the first gestures need to be properly constructed and then they must undergo testing in order to match their realization of the proposed method. Even a perfectly tuned and performed gesture is useless if the gesture does not fulfil its purpose, thus the emphasis is placed on the first approach.

8.2.1 Proposed Gestures

We originally proposed two gestures – throw gesture and catch gesture. We were informed by specialist during the consultation meeting that even when our proposed gestures are still quite appropriate for our usage, there are many more feasible concepts.

We were discussing multiple aspects of hand movements, one of which is the pinch position. Very suitable our purposes were usages of two-finger pinch or five-finger pinch. Another proposed gesture is crossing the center of body with a hand executing the gesture. We decided not to involve this gesture in the end, because we could not find appropriate usage of this gesture. Also, the range of the sensor is slightly restricting the area where the gesture could be performed.

The mentioned possible gestures also included pushing an object with both hands, pressing a balloon from the left and right side in order to pop it, catching handles and balls. Our final gestures were also provided to us with very useful description of possible problems and the way users from different motor skills categories practice the movements in order to master them. That was one of the reasons why we finally decided to choose the carry gesture, rotation gesture and scale gesture.

8.2.2 Realized Gestures

The gesture realization was due to many conditions of each gesture very tedious. After finishing the gameplay logic of the experiment and designing the gestures, we presented the application to other researchers and ask them for their thoughts on the gesture handling. Most of the researchers were in fact not able to execute gestures, which were designed to be very fluent and not difficult to perform. Therefore, we collected the gesture problems from their experiences and from our observations.

The altered and straightforward gesture implementation was exposed to the same procedure. After improving the gesture detection, we identified new problems in sensor range and our scene setting. At the end the scene, the gesture conditions and experiment were much better tuned and ready for proper experimenting and data obtaining than at the beginning.

8.3 User Models

Training and evaluating the user models with cross-validation did not confirm their correctness due to lack of the data. After the first experiment we were doubting the proposed user models, but during and after the second experiment it became clear that the user models are usable in the context of recognition of motor skills category. The user model for the carry gesture yields underwhelming performance compared to other gestures, but it might be also due to the type of the gesture or its requirements of the space, because the sensor might not be very precise when it is operating on the border of its range. However, the user models for the rotate and scale gestures appear to perform much better, what was observable during the second experiment and can also be observed from the results in section 8.7.

8.4 Experiments

During the evaluation of gesture realization feasibility, experiment scenario and gameplay was also discussed and reviewed. Suggestions of fellow researchers were taken into the consideration, resulting in some changes in our experiment scene. Camera perspectivity was one of the problems, which was resolved by changing the type of the camera from perspective to orthogonal. Another problem was in placement of objects in the scene, bases of pyramids were repositioned along with the occlusive appearance of the control units.

After changes were made, some of the fellow researchers were additionally questioned if the problems they identified were resolved by changes. Their consent to execute the experiment as it was proposed with applied changes we considered as feasible evaluation of the experiment setup.

8.4.1 Participants of Experiments

All participants of both experiments with dyspraxia or cerebral palsy were carefully selected or approved by the specialist in physical therapy. Healthy participants were selected either by the specialist or by their tutor with no motor skills impairments. The participants of both experiments are patients of Research Institute for Child Psychology and Pathopsychology in Bratislava. We can rely on the proper selection of the users, because we cannot think of better classification of users than one done by doctor.

8.5 Classification

Classification results depend solely on the classifier. We can help the classifier with appropriate preprocessing or other data manipulation based on the domain or data knowledge. In our work we evaluate selection of the most appropriate classifies and also effects of manipulation with class labels in order to simplify classification problem with reduction of class labels.

8.5.1 Classifiers Comparison

For the comparison of classifiers, we decided to include the proposed algorithms along with some easily usable default algorithms from our chosen *scikit-learn* library. We also added ensemble algorithm *XGBoost* to improve properties of decision tree. This is the list of classifiers we evaluated on our training dataset: Linear Discriminant Analysis (LDA), *k*-nearest neighbors (*k*-NN), decision tree (DT), Gaussian Naïve Bayes (NB), Support Vector Machines (SVM), and XGBoost (XGB).

Our training dataset for each gesture consists of 15 users: 4 users with dyspraxia, 4 users with cerebral palsy and 7 heathy users. We have two approaches to evaluate the performance of our classifiers based on selecting data onto the test dataset:

1. selecting a subset of feature vectors from each user and
2. selecting all feature vectors from one user (or multiple users).

The dataset was slightly altered before classifier comparison because not every category was represented equally. One of the users with cerebral palsy performed an excessive number of failed gestures of each type. For each gesture the threshold was specified and if user exceeded the number of failed gestures, the rest of the gestures were not taken into consideration. The threshold was chosen for each gesture and each category very carefully, so very little data were omitted. The thresholds are listed in Table 1.

After this alteration of dataset, we equally represent each category of motor skills. There are 314 feature vectors for carry gesture, 1987 feature vectors for rotate gesture and 508 feature vectors for scale gesture by each motor skills category. The overall number of feature vectors is therefore large enough to represent the gestures adequately.

Table 1. Thresholds for number of user gestures.

Category	Carry Gesture T.	Rotate Gesture T.	Scale Gesture T.
Cerebral Palsy	101	1112	172
Dyspraxia	314	1987	508
Healthy	58	358	138

The participants of our first experiment were carefully selected to cover the spectrum of all categories. If one of them is left out from training process completely, there is a very high risk that the spectrum will not be covered accurately by the classifier and during the evaluation the user could be heavily misclassified. However, if we take feature vectors from each user, so each user is represented in training dataset and evaluating dataset, there is a risk that the classifier will also take the user identity along with the motor skills category into consideration.

We decided to perform the comparison based on both approaches of selecting the test dataset. In the first approach we used 5-fold cross-validation, in the latter there are 15 splits corresponding to 15 users, so classifier training and evaluation will be performed 15 times, once for each split. We performed comparison based on two different scoring systems: accuracy and the macro-average of F_1 score. Hyperparameter optimization was, however, based only on the accuracy. The best achieved results are summarized for each gesture separately in Table 2, Table 3 and Table 4. We did not have to hesitate about selecting the best classifier, because in each comparison there is one classifier clearly standing out, and for all gestures it is the Support Vector Machines algorithm.

From the tables we can see that our concerns were on point, so we cannot make any conclusions based on any of these numbers. Although, the combination of both approaches is quite sufficient to determine the best classifier, it is hard to tell how successful it will in reality be.

During the hyperparameter optimization with *scikit-learn* library we encountered multiple surprising moments – after large amount of time classifiers tended to perform the best with most of the default parameters. Our final classifier has exactly the same parameters for each gesture (only non-default parameters are listed): *C* = 16, *probability* = True, *tol* = 0.5.

The *C* parameter is *Penalty parameter C of the error term*. The parameter *tol* stands for *Tolerance for stopping criterion*. The *probability* parameter handles *whether to enable probability estimates*. A very important parameter, which is the default but worth of mention, is the *kernel*. It defaults to value *rbf*, which represents *Gaussian Radial Basis Function* (Pedregosa et al.,).

Table 2. Comparison of classifiers with 3 categories for rotate gesture.

Classifier	**Accuracy 1**	**Accuracy 2**	**F_1 macro 1**	**F_1 macro 2**
LDA	0.7654	0.3374	0.7662	0.1501
k-NN	0.9302	0.3511	0.9302	0.1653
DT	0.7552	0.3033	0.7544	0.1517
NB	0.6160	0.3447	0.6087	0.1596
SVM	0.9405	0.3688	0.9404	0.1674
XGB	0.9219	0.3340	0.9219	0.1459

Table 3. Comparison of classifiers with 3 categories for scale gesture.

Classifier	**Accuracy 1**	**Accuracy 2**	**F_1 macro 1**	**F_1 macro 2**
LDA	0.7797	0.3502	0.7799	0.1837
k-NN	0.9226	0.3920	0.9229	0.1882
DT	0.7462	0.3426	0.7535	0.1613
NB	0.6695	0.3300	0.6571	0.1680
SVM	*0.9220*	*0.3947*	*0.9220*	*0.2032*
XGB	0.8957	0.3236	0.8955	0.1548

Table 4. Comparison of classifiers with 3 categories for carry gesture.

Classifier	**Accuracy 1**	**Accuracy 2**	**F_1 macro 1**	**F_1 macro 2**
LDA	0.7892	0.2796	0.7882	0.1332
k-NN	0.8718	0.3135	0.8714	0.1498
DT	0.7500	0.3453	0.7532	0.1635
NB	0.6758	0.4054	0.6762	0.1785
SVM	*0.9036*	*0.3518*	*0.9042*	*0.1731*
XGB	0.8919	0.3419	0.8918	0.1564

8.5.2 Classification with Redefined Categories

While observing the results of the second approach (selecting all feature vectors of a user into evaluation dataset) we noticed that there are interesting patterns in confusion matrices. There was a much larger confusion between healthy users and users with dyspraxia, than between healthy users and users with cerebral palsy. This can be observed also on two-dimensional representation of our dataset for rotate gesture shown in Figure 7.

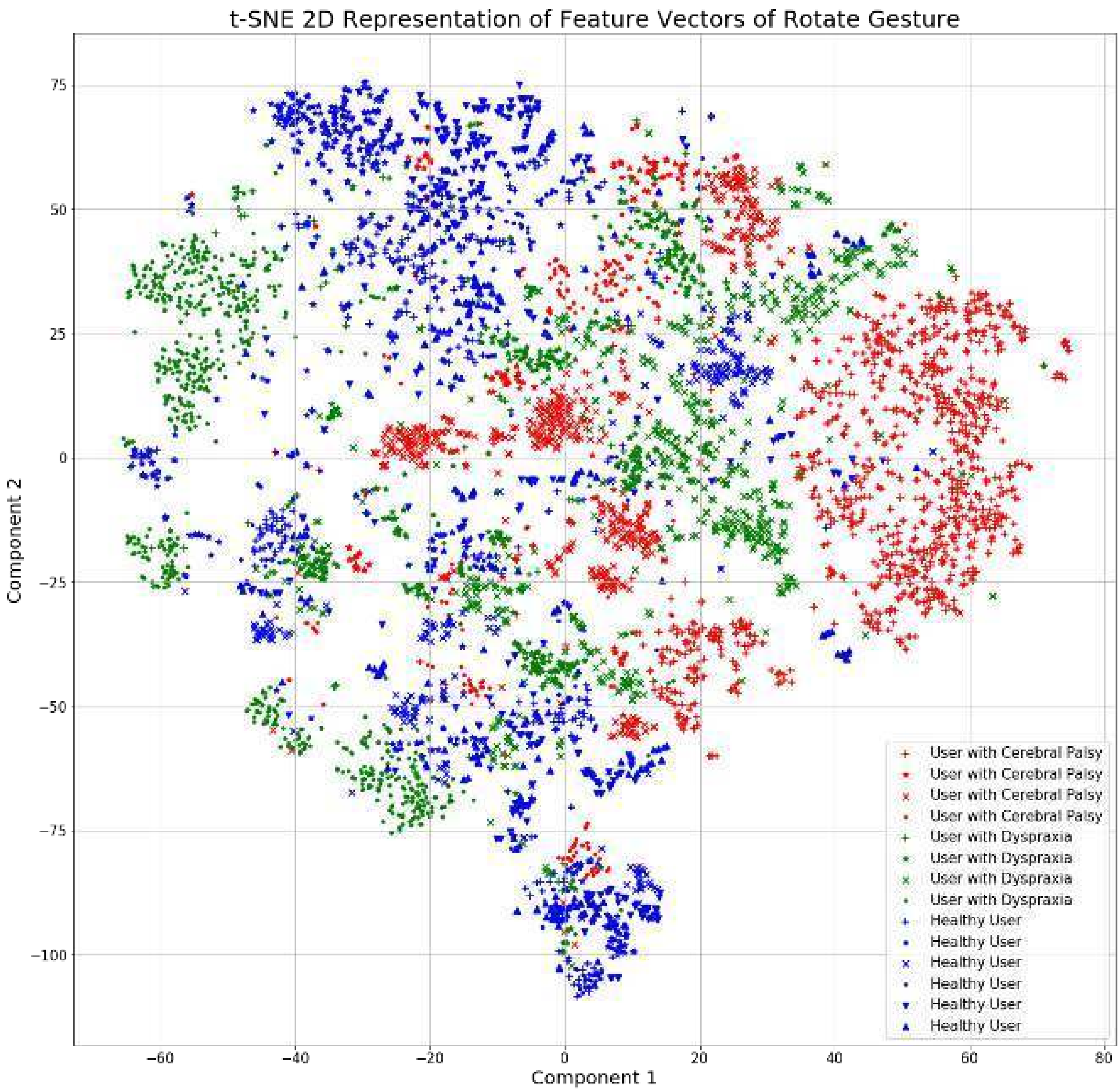


Figure 7. t-SNE two-dimensional representation of feature vectors of the rotate gesture.

The three categories can be ordered by a severity of impairment, so we decided to create only two categories and test the second approach again. We propose three approaches to create only two categories:

a) omit category of users with dyspraxia,

b) merge category of users with dyspraxia with category of users with cerebral palsy and

c) merge category of users with dyspraxia with category of healthy users.

Table 5 shows the comparison of all of described approaches for two scoring systems, accuracy and macro-average of F_1 score using the second approach for the test dataset selection. In most of the cases there is no particular improvement. Both scoring systems show improved results, but we have to take into the consideration that we now have only 2 categories. However, the third approach shows very promising results for the scale gesture and for the rotate gesture as well. It is overall the most successful approach. The computation of these results was carried out after the second experiment, because we were not originally focused on this use case. Nevertheless, the results are very intriguing, and we strongly encourage any further research in this particular case.

Table 5. Comparison of SVM performance.

Gesture	**Acc. 2a**	**F_1 2a**	**Acc. 2b**	**F_1 2b**	**Acc. 2c**	**F_1 2c**
rotate	0.6829	0.3949	0.5662	0.3463	0.6874	0.3911
scale	0.6522	0.4277	0.5303	0.3288	0.7526	0.4987
carry	0.5558	0.3320	0.6126	0.3624	0.5694	0.3939

8.6 Adaptation

The constrains of gestures were discussed with the specialist and according to them the adaptations were designed. However, the evaluation of these adaptations proved difficult, because it is highly subjective. Therefore, not only we asked participants about their opinions, but we also carefully observed their actions together with the specialist. Consulting the participants' opinions on adaptations and our observation, we made a conclusion that the adaptations met the needs of impaired participants.

We were especially concerned about mistakenly adjusting the gestures to the healthy users which could ruin the purpose of the game for them. These concerns were mostly proven wrong after evaluating the effort and duration necessary to complete the game. While the effort and duration, based on our observations, significantly dropped for users with cerebral palsy, it could not be significantly observed with users with dyspraxia. Healthy users were from this point of view affected minimally or not at all, because we did not notice differences between the first and the second experiment.

The significant decrease in time and effort for users with cerebral palsy was very appropriate, because without any adaptations these users were barely able to finish the game. With adaptations they were still performing noticeably slightly slower overall, but the difference was negligible.

8.7 Results

Our system is applying adaptations based on only one failed gesture using the classification. These class labels after each failed gesture were persisted into the separate file for each user and each gesture. We decided to compute accuracy for each gesture and each user. We have 10 participants of the second experiment, and we decided to make user-based average of the accuracy for each category. Therefore, if one of the users has a lot of failed gestures and it is being recognized very successfully, it will make the same impact as a user having a very small number of failed gestures and was recognized poorly.

Adaptations in the application were applied after one failed gesture. However, we found out that taking more than one gesture of the same type leads to much improved results with regards to accuracy. For each array of class labels, we decided not to take only one, but wait until *k* failed gestures indicated the same category. If this threshold is 2, then the system will wait until there is a second occurrence of any of the already classified categories. After reaching the threshold, the class label is then carried out and the counter resets. If the threshold is set to 1, the same result is obtained as in the original use case. In Figure 8, Figure 9 and Figure 10 is shown how the accuracy of recognition of each category changes with regards to this threshold. It is clear, that threshold 1 is definitely not the most accurate, therefore our system could be easily improved after integrating this continuous idea of adaptation. The data used for figures are in tables. *N* stands for the number of gestures required for decision, accuracy is abbreviated as *acc.* and F_1 represents macro-average of F_1 score. The motor skills categories are abbreviated with their initial letters – *CP* for cerebral palsy, *D* for dyspraxia and *H* for healthy users' motor skills category.

The class arrays of carry gesture in Figure 10 for users with dyspraxia fluctuated widely and were inaccurate, so there is not enough data to cover a threshold of more than 3. That is also the reason why we do not consider threshold for any gesture greater than 9. Values are either stable at that point or not available.

We consider the rotate gesture results to be better than the result of scale gesture, because the rotate gesture has a much better accuracy with healthy users. If the accuracy with healthy users is very high, then it is not an issue if the accuracy for other categories are lower, because every now and then the impaired user is able to make progress and the user is still feeling involved. However, if the impaired user recognition is perfect, but the healthy users' recognition is poor, then most of the users are most of the time mistreated. One has to keep in mind that the majority of users are healthy. If we keep treating such users almost perfectly correctly, then every, even small, adaptation to non-healthy user is extra value of the system.

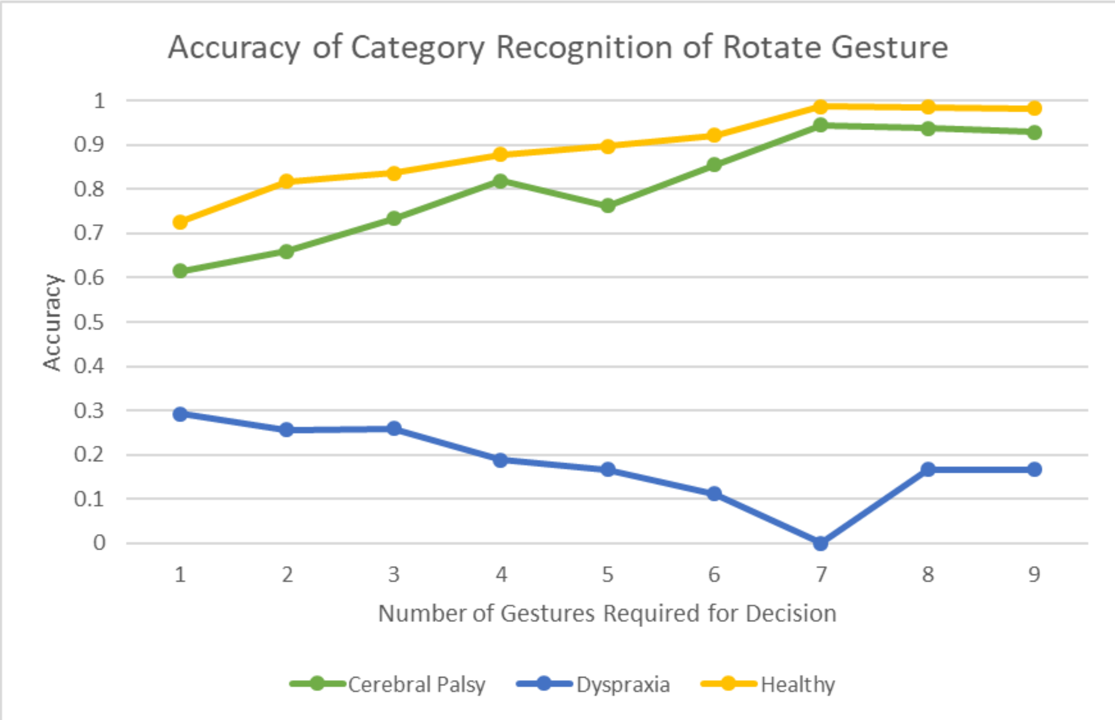


Figure 8. Accuracy of category recognition of rotate gesture.

Table 6. Scoring of category recognition of rotate gesture.

N	Acc. CP	F_1 CP	Acc. D	F_1 D	Acc. H	F_1 H
1	0.6152	0.2529	0.2929	0.1497	0.7261	0.2784
2	0.6596	0.2632	0.2564	0.1645	0.8173	0.3598
3	0.7333	0.2815	0.2593	0.1611	0.8366	0.3955
4	0.8194	0.4496	0.1889	0.1429	0.8788	0.5393
5	0.7619	0.4308	0.1667	0.1111	0.8976	0.6707
6	0.8545	0.4603	0.1111	0.0833	0.9216	0.7770
7	0.9444	0.7353	0.0000	0.0000	0.9867	0.8966
8	0.9375	0.7333	0.1667	0.1111	0.9846	0.8960
9	0.9286	0.7308	0.1667	0.1111	0.9818	0.8952

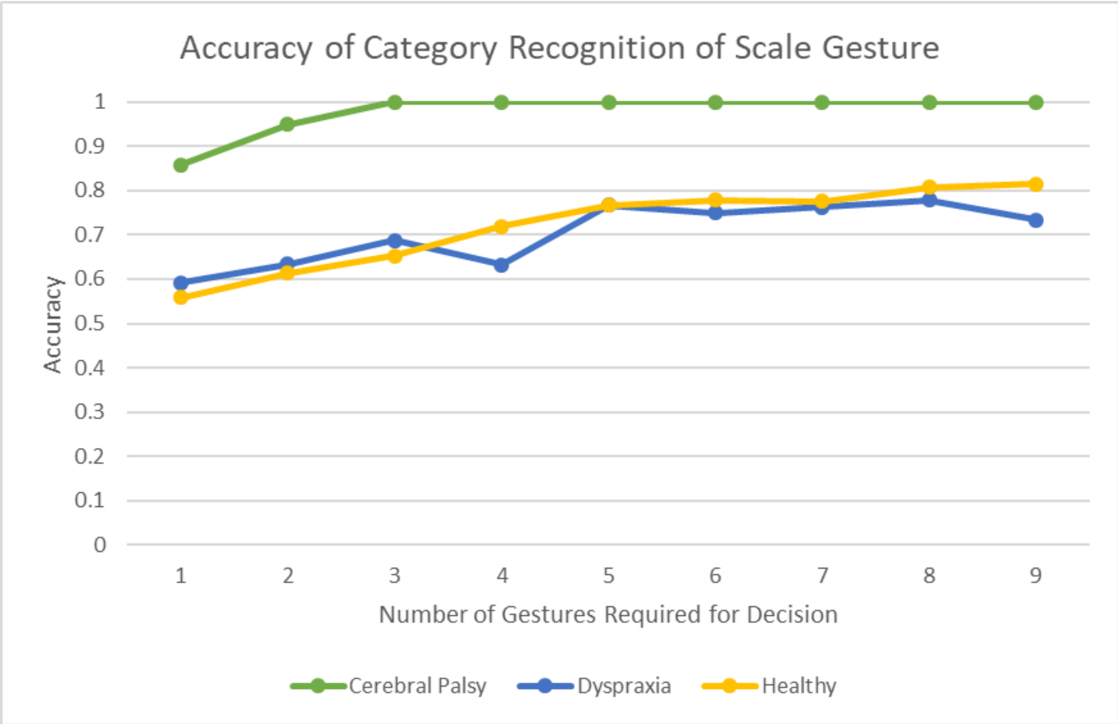


Figure 9. Accuracy of category recognition of scale gesture.

Table 7. Scoring of category recognition of scale gesture.

N	Acc. CP	F_1 CP	Acc. D	F_1 D	Acc. H	F_1 H
1	0.8586	0.3080	0.5921	0.2900	0.5582	0.2361
2	0.9495	0.4870	0.6336	0.4669	0.6140	0.3317
3	1.0000	1.0000	0.6877	0.5278	0.6526	0.3680
4	1.0000	1.0000	0.6325	0.5083	0.7197	0.4902
5	1.0000	1.0000	0.7667	0.7436	0.7671	0.6030
6	1.0000	1.0000	0.7500	0.7333	0.7786	0.6084
7	1.0000	1.0000	0.7619	0.7407	0.7758	0.6071
8	1.0000	1.0000	0.7778	0.7500	0.8067	0.6157
9	1.0000	1.0000	0.7333	0.7222	0.8139	0.6188

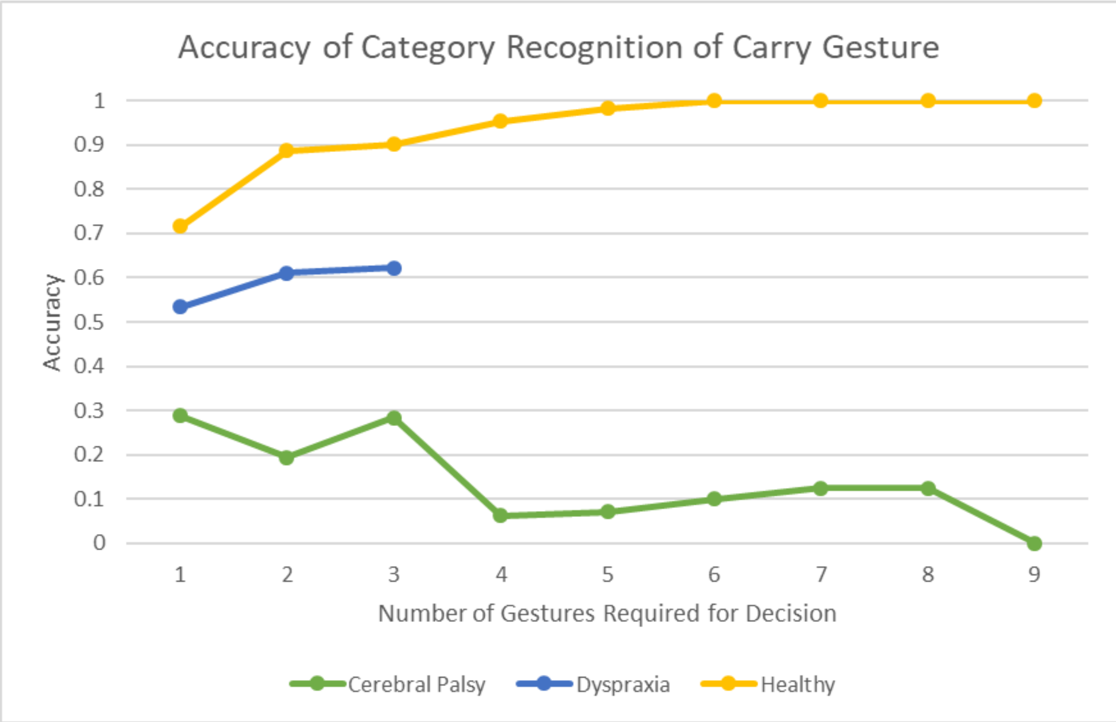


Figure 10. Accuracy of category recognition of carry gesture.

Table 8. Scoring of category recognition of carry gesture.

N	Acc. CP	F_1 CP	Acc. D	F_1 D	Acc. H	F_1 H
1	0.2887	0.1493	0.5344	0.2937	0.7166	0.2780
2	0.1944	0.1320	0.6111	0.4848	0.8863	0.5390
3	0.2833	0.1667	0.6222	0.4881	0.9014	0.6412
4	0.0625	0.0556	-	-	0.9533	0.7875
5	0.0714	0.0625	-	-	0.9818	0.8952
6	0.1000	0.0833	-	-	1.0000	1.0000
7	0.1250	0.1000	-	-	1.0000	1.0000
8	0.1250	0.1000	-	-	1.0000	1.0000
9	0.0000	0.0000	-	-	1.0000	1.0000

8.7.1 Results of Combined Gestures

After computing the results for each gesture with multiple feature vectors we decided also to evaluate the accuracy of motor skills category recognition using combination of gestures. For each user we combined class labels from all gestures (the first three values of new array were the first labels from three original arrays in order). Since the carry gesture itself did not exhibit a satisfactory performance, we decided to omit it from combination and combine only rotation and scale gesture arrays. As we expected, the latter combination outperformed the first in every single motor skills category. Figure 11 depicts the accuracy of the combination consisting of rotate and scale gesture class arrays. Values of the scoring methods are in Table 9.

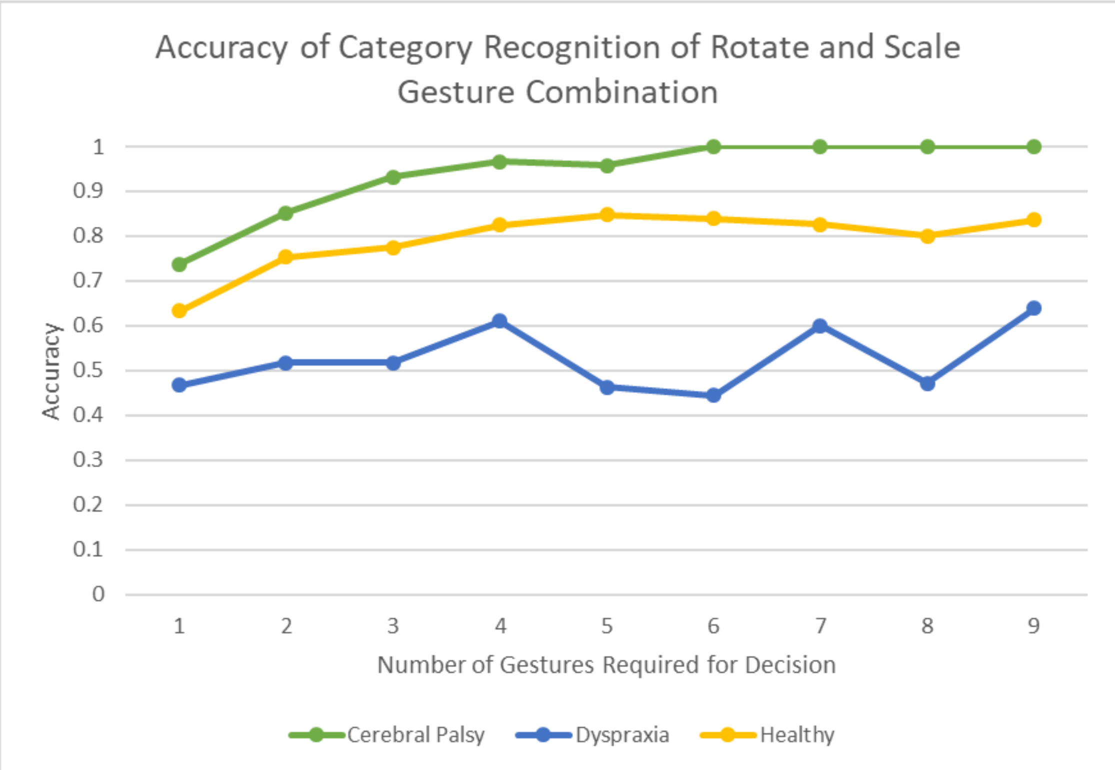


Figure 11. Accuracy of category recognition of rotate and scale gesture combination.

Table 9. Scoring of category recognition of rotate and scale gesture combination.

N	Acc. CP	F_1 CP	Acc. D	F_1 D	Acc. H	F_1 H
1	0.7380	0.2829	0.4670	0.2087	0.6339	0.2568
2	0.8516	0.3800	0.5176	0.2988	0.7532	0.3158
3	0.9324	0.4824	0.5167	0.2939	0.7740	0.3485
4	0.9667	0.7414	0.6104	0.3313	0.8254	0.5207
5	0.9583	0.7391	0.4635	0.2650	0.8474	0.6277
6	1.0000	1.0000	0.4444	0.2626	0.8396	0.6535
7	1.0000	1.0000	0.6000	0.3577	0.8267	0.6485
8	1.0000	1.0000	0.4722	0.3095	0.8004	0.6380
9	1.0000	1.0000	0.6389	0.3873	0.8367	0.6524

9 Conclusion

In our work we analyzed common approaches and methods in our researched field and similar fields. This analysis was used as basis for our hypotheses and assumptions and created our expectations in their confirmation. In order to do so we proposed the system. Our proposals were based on the research field analysis together with consultations with the specialist in physical therapy. Our collaboration with the specialist was very fruitful throughout multiple part of our work including proposals, execution of experiments, adjustments and results representation.

The realization of our proposals was very tightly connected to their evaluations step by step. These multiple feasible approaches overall evaluated our whole solution. Based on the evaluation we consider our hypotheses confirmed, supported by the data we were able to obtain and process and by the specialist evaluation as well. The results show that for motor skills category determination were suitable two out of three proposed gestures, and both gestures performed very well in the adaptation context. The results of scale gesture recognition of motor skills categories for 3 gestures in row were better than our expectations: 100 % for the healthy users' motor skills category 68.77 % for the category of motor skills of users with dyspraxia and 65.26 % for the users with dyspraxia motor skills category.

Experiments we used for data collection and system evaluation were not involving very large number of participants, because of the nature of experiments. Requirements for users to be involved in experiments were very strict for the cerebral palsy motor skills category and for the dyspraxia motor skills category, because users had to have the diagnosis confirmed by a doctor. We encourage further experimenting with larger groups of participants.

Our solution was built to decide whether it is possible for system to recognize motor skill category of a user and adapt to the user based on this category. Now, when these hypotheses are confirmed we see a large potential in developing this idea further and create similar systems with a different goal – to create gestures and techniques able to adapt to user more conveniently.

Resumé

V súčasnosti sa do popredia dostávajú aplikácie ovládané gestami používateľov. Kým pre mnohých takéto gestá predstavujú rôzne výhody, sú používatelia, ktorí majú problém takéto gestá vykonávať. Dôvody môže byť napríklad fyzické obmedzenie používateľa nejakým postihnutím a teda jeho neschopnosť tieto gestá bezchybne vykonávať. Keďže však gestá možno zaradiť medzi behaviorálne charakteristiky, tak sa nám otvára možnosť takéto skupiny rozpoznávať a následne im metódy interakcie prispôsobovať.

Biometrické charakteristiky sú často využívané na identifikáciu alebo autentifikáciu konkrétneho používateľa, no my sa v práci venujeme inému prístupu – rozpoznávanie kategórií motorických schopností. Našim cieľom je zistiť, či aplikácia je schopná takéto kategórie adekvátne rozpoznávať a na základe toho sa potom používateľom adekvátne prispôsobiť. Na to však treba najprv stanoviť kategórie, ktoré sa snažíme rozpoznávať a definovať gestá, počas ktorých budeme rozpoznávanie vykonávať. Naše predpoklady treba, samozrejme, experimentálne overiť a na základe toho určiť, či sú takéto prispôsobenia na základe rozpoznávania motorických schopností možné.

Motorické schopnosti a virtuálna realita

Virtuálna realita je pohlcujúcim interakčným systémom, ktorý v používateľovi vytvára ilúziu vstupu do virtuálneho sveta (Heim, 2000). Používateľ je teda priamo zapojený do systému, s ktorým môže interagovať rôznymi spôsobmi cez špecializované vstupné zariadenia a prežívať výsledky svojich vstupov vo virtuálnom svete (Deng et al., 2010; Lange et al., 2010). Hoci tieto prístupy nie sú nové, pre získavanie údajov o motorických schopnostiach je najdôležitejší ten, kde používateľ interaguje gestami rúk, pretože takto interaguje s objektami aj v reálnom svete (Aslan et al., 2014).

Gestá

Jedným zo zaujímavých aspektov gest s rukami pre našu prácu je schopnosť systému rozpoznať vykonanie gesta používateľom. Na to možno využiť dva prístupy: používateľ drží v ruke zariadenie, pomocou ktorého vykonáva gestá alebo ruky používateľa sú snímané kamerou a pomocou vizuálnych vstupov počítač vyhodnotí, či používateľ vykonal gesto (Nugrahaningsih et al., 2015). Je viacero spôsobov, ako možno získať a vyhodnotiť dáta z pohybu rúk používateľa a hoci sa rôznia aj zariadenia, napríklad Leap Motion a Kinect, cieľom je stále čo najlepšia úspešnosť rozpoznania vykonania gest (Manresa et al., 2005b; Marin et al., 2014). Okrem rozpoznania gesta samotného sú aj aplikácie, kedy sa rozpoznáva aj používateľ, ktorý gesto vykonáva práve na základe vykonávaného gesta (Aslan et al., 2014; Imura and Hosobe, 2016b; Jiang et al., 2014).

Ďalším zaujímavým aspektom je interakcia pomocou gest. Možno sa pomocou gest v aplikácii navigovať alebo v rámci nej niečím manipulovať (Cabral et al., 2005b).

Na základe takýchto interakčných gest je zložitejšia presná identifikácia používateľa, no identifikácia kategórie motorických schopností je trocha všeobecnejšia a preto by takéto využitie mohlo byť dostatočne presné pre naše využitie.

Motorické obmedzenia a interakcia

Motorické schopnosti používateľov sa vyskytujú už aj dnes v interakčných aplikáciách. Či už je to hra pre deti s dyspraxiou (Caro, 2014), aplikácia zameraná na určenie pohyblivosti (Landry et al., 2013) alebo aplikácia pre zisťovanie, či je miera motorických schopností používateľov dostatočná pre určitý druh práce (Singh and Aggarwal, 2016).

Existujú však aj prístupy, kde sa pomocou interakčných systémov dosiahla rehabilitácia používateľov po detskej mozgovej obrne, pretože aplikácie primali používateľov vykonávať ich zdraviu prospešné gestá za zásterkou hry (Chang et al., 2013; Huang, 2011; Mousavi Hondori and Khademi, 2014; Oliveira et al., 2016).

Používateľský model a klasifikácia

Pre klasifikáciu vzoriek jednotlivých používateľov a určovanie, ktorej triede patria je prvoradé vytvoriť správny používateľský model. Ďalej je však potrebné aj vedieť účel klasifikácie, čo môže uľahčiť výber vhodného klasifikátora. Model používateľa v našom prípade predstavujú črty, ktoré sú počítačovou reprezentáciou používateľa (Allen, 1997). Určovaním identity na základe týchto čŕt sa zaoberá biometria (Jain et al., 2004). V našom prípade sa nezaoberáme konkrétnou identitou, ale kategóriou motorických schopností – tá je rovnaká pre viacerých používateľov, zatiaľ čo identita je jedinečná.

Počas procesu rozoznávania, používateľský model extrahovaný počas chodu aplikácie v danom momente musí byť porovnaný so všetkými ostatnými modelmi v databáze. Pre takéto porovnávanie je najrozšírenejšou metódou klasifikácia (Aggarwal, 2014). Veľmi rozšírené sú v súčasnosti klasifikátory *k*-NN (z angl. *k-Nearest Neighbors*), SVM (z angl. *Support Vector Machines*) a rozhodovací strom (Aly, 2005).

Senzory

Na základe výskumu a vedeckej práce v tejto oblasti máme tri rozumne praktické možnosti:

- *interakčné rukavice* (Adamovich et al., 2004; Baker et al., 2004),
- *Leap Motion* (Chan et al., 2015; Cui and Sourin, 2014; Marin et al., 2014; Potter et al., 2013) a

- *Kinect* (Altanis et al., 2013; Bigdelou et al., 2012; Chang et al., 2013; Huang, 2011; Jiang et al., 2014; Marin et al., 2014; Mousavi Hondori and Khademi, 2014; Zhang et al., 2014).

Prístupy využívajúce interakčné rukavice dnes už nie sú veľmi zastúpené, pretože ostatné dva senzory dokážu získať porovnateľné výsledky bez potreby priameho obmedzovania používateľa senzorom a káblami priamo pripojenými k jeho telu. Leap Motion aj Kinect sú si veľmi podobné, obe zariadenia sú trojdimenzionálne senzory zachytávajúce údaje infračervenými kamerami. Oba boli aj pomerne populárne, až kým sa Microsoft nerozhodol zastaviť produkciu a aj podporu senzora Kinect (Wilson, 2017).

Predpoklady rozpoznávania motorických schopností a prispôsbenia

Na základe doménovej analýzy predpokladáme, že je možné a vhodné:

- Klasifikovať motorické obmedzenie pomocou senzora Leap Motion v rámci špecifikovaných kategórií motorických obmedzení na základe koordinačného gesta vykonaného používateľom. Takéto koordinačné gesto v kontexte našej práce môže byť gesto rotácie, gesto škálovania alebo gesto prenosu, pričom každé z nich je vykonané pomocou oboch rúk.
- Klasifikovať motorické obmedzenie pomocou senzora Leap Motion (ako popísané vyššie) v reálnom čase za chodu aplikácie.
- Prispôsobiť aplikáciu kontrolovanú gestami rúk používateľovi na základe jeho motorického obmedzenia.

Návrh biometrického systému pre detekciu motorických schopností

Pre detekciu motorických schopností používateľa potrebujeme najprv špecifikovať kategórie týchto schopností. Je potrebné určiť aj gestá, na základe ktorých chceme tieto kategórie klasifikovať. Pred klasifikáciou samotnou, musíme dáta týchto gest získať zo senzora Leap Motion a extrahovať z nich črty. Tieto črty môžeme klasifikovať pomocou rôznych algoritmov, z ktorých treba vybrať ten najúspešnejší. Na záver treba vedieť túto klasifikáciu použiť za chodu aplikácie v reálnom čase a zahrnúť prispôsobenia pre jednotlivé kategórie motorických schopností.

Kategórie motorických schopností

Pre účely rozpoznávania navrhujeme nasledovné tri kategórie motorických schopností:

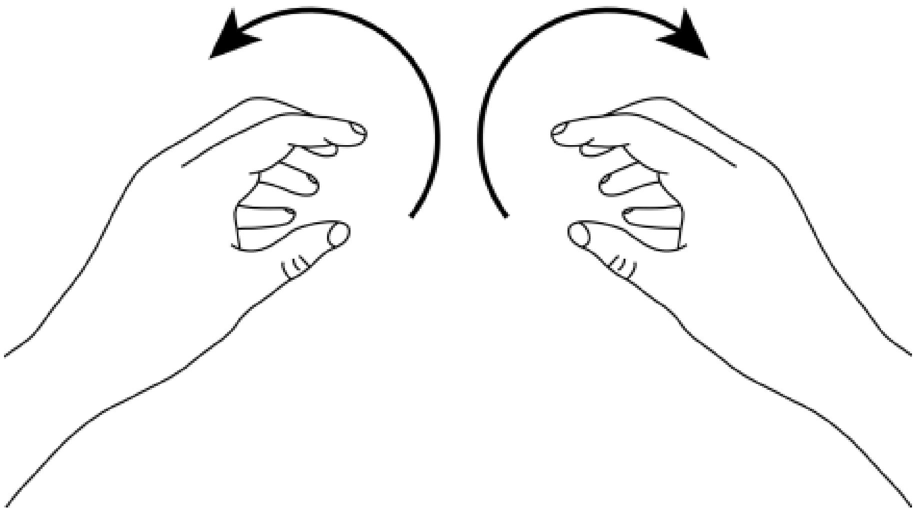
- motorické schopnosti zdravých používateľov,
- motorické schopnosti používateľov trpiacich dyspraxiou a

- motorické schopnosti používateľov po detskej mozgovej obrne.

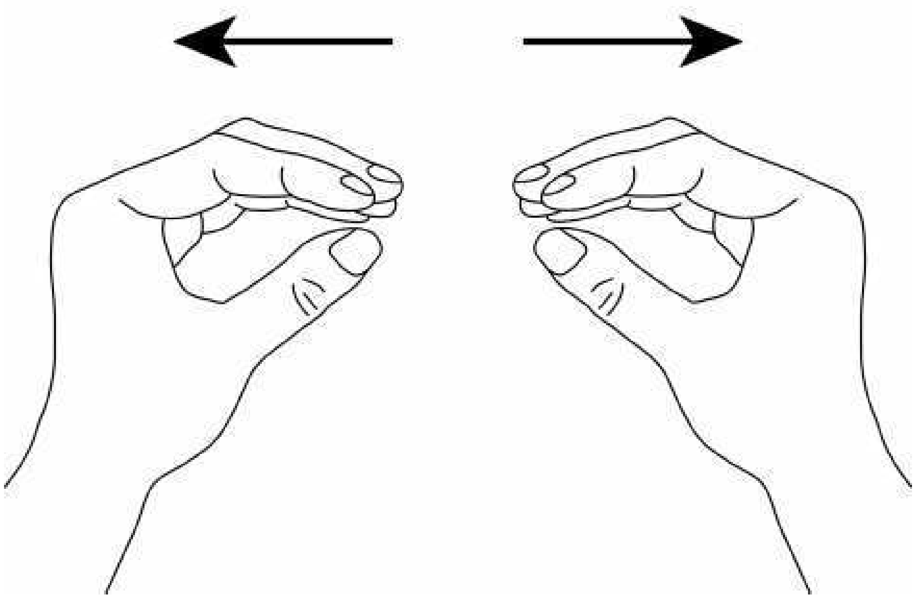
Gestá

Berúc do úvahy povahu kategórií motorických schopností, navrhujeme tri rôzne koordinačné gestá vykonávané obojručne: gesto rotácie, gesto škálovania a gesto prenosu.

Gesto rotácie je špecifikované ako súčasná rotácia oboch rúk – ľavej ruky v protismere hodinových ručičiek a pravej ruky v smere hodinových ručičiek, ako znázorňuje aj obr. 2. Aby sa zaručilo, že rotácia oboch rúk bude naozaj súčasná, uhol, o ktorý sa obe ruky otočili musí byť takmer rovnaký – treba brať do úvahy istú toleranciu.



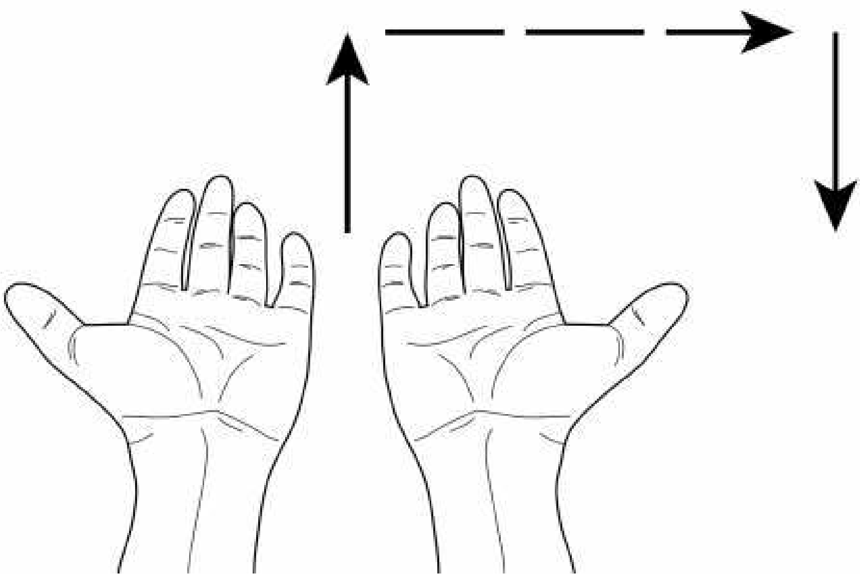
Obr. 2. Gesto rotácie.



Obr. 1. Gesto škálovania.

Gesto škálovania je špecifikované ako súčasné vzďaľovanie sa rúk od seba v horizontálnej rovine. Obe ruky počas tohto gesta musia uchopiť objekt všetkými prstami, pričom tento úchop aj smer pohybu rúk sú znázornené na obr. 1. Rovnako ako pri predošlom geste, obe ruky sa musia od stredu vzďaľovať súčasne a navrhujeme istú toleranciu pre rozdiel týchto vzdialeností.

Posledným gestom je gesto prenosu znázornená na obr. 3. Gesto simuluje prenášanie objektu v reálnom svete, napríklad tácky s pohármi. Ruky teda musia byť celý čas otočené smerom nahor a v rovnakej výške, inak by sa objekt skĺzol. Ruky musia byť aspoň v určitej vzdialenosti od seba, inak by sa objekt preklopil keby boli ruky príliš blízko. Ak by boli, naopak, priďaleko od seba, tak by objekt prepadol pomedzi ne, takže obe ruky musia zostať v nejakej maximálnej vzdialenosti od seba. Objekt treba zdvihnúť z jedného miesta, preniesť a položiť na druhé, ako znázorňuje aj obrázok.



Obr. 3. Gesto prenosu.

Experimenty

Navrhujeme vykonať dva experimenty, kde prvý bude slúžiť na zber dát, na základe ktorých vyberieme najlepší klasifikátor a druhý overí našu metódu adaptácie a výber klasifikátora. Pre experimenty navrhujeme jednoduchú hru, ktorá zahrnie všetky tri gestá. Cieľom hry bude postaviť pyramídu z kociek. Každú kocku však treba na začiatky vyrobiť (pomocou gesta rotácie) a každú jej dimenziu zväčšiť (pomocou gesta škálovania). Po zväčšení každej dimenzie treba kocku otočiť (gestom rotácie), aby mohla byť zväčšená aj ďalšia dimenzia, až kým nie je celá kocka zväčšená v každej dimenzii. Takto zväčšená kocka sa sama presunie na ľavý podstavec, odkiaľ ju treba vziať a preniesť (gestom prenosu) na pravý piedestál, odkiaľ sa už sama presunie na správne miesto v pyramíde. Navrhujeme postaviť pyramídu pozostávajúcu z 15 kociek, aby boli gestá vykonané dostatočný počet krát.

Metóda rozpoznávania

Navrhujeme použitie minimálne klasifikátorov *k*-NN, SVM a rozhodovací strom, pričom ďalšie možno vyskúšať v závislosti od knižnice, v ktorej bude klasifikácia implementovaná. V rámci výberu najlepších parametrov navrhujeme vykonanie optimalizácie hyperparametrov pre navrhnuté klasifikátory.

Prispôsobovanie

Prispôsobovanie sa aplikácie navrhujeme oba pre kategórie motorických schopností používateľov po detskej mozgovej obrne a používateľov s dyspraxiou. Kvôli fyzickým obmedzeniam, ktoré používatelia po detskej mozgovej obrne majú navrhujeme odstránenia kritéria pre súčasné vykonávanie gest oboma rukami. Obe ruky sa musia podieľať na každom geste, ale stačí, keď jedna z nich vykoná svoju časť gesta, zatiaľ čo druhá, v mnohých prípadoch postihnutá, ruka bude dôležitá najmä pre začatie gesta, kde zaujme svoju pozíciu, ale nemusí vykonať dostatočný pohyb na splnenie podmienok gesta.

Prispôsobovanie sa používateľom s dyspraxiou je menej radikálne, pretože ich ruky sú samostatne schopné vykonať svoj podiel na geste, majú však problémy s ich koordináciou. Preto navrhujeme také prispôsobenie, ktoré v konečnom dôsledku bude stále vyžadovať vykonanie gesta oboma rukami, no bez väčšieho dôrazu na ich súčasné a koordinované pohyby.

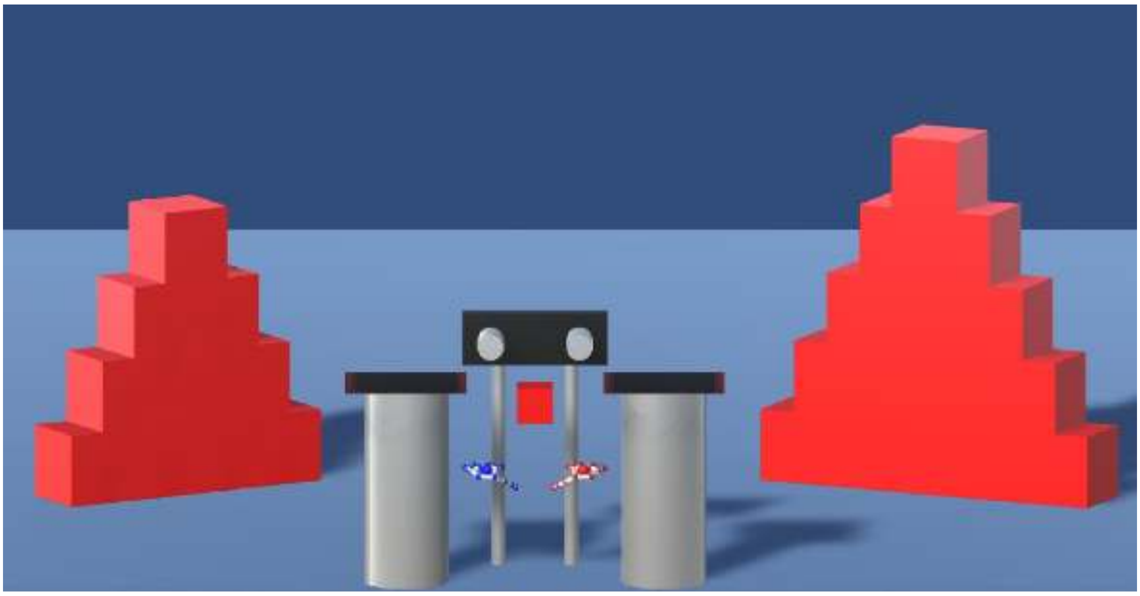
Realizácia navrhnutého systému

Scéna experimentov na obr. 4 je implementovaná v prostredí Unity 2018.3.8f1 v spolupráci s ovládačom zariadenie Leap Motion verzie 4.0.0. Herná logika je implementovaná v jazyku C#, zatiaľ čo klasifikácia je vykonávaná mimo Unity prostredia v jazyku Python za použitia knižnice *scikit-learn*.

Keďže gestá v návrhu naše vlastné a veľmi špecifické, museli byť implementované od základu. Počas vykonávania gest, kocka mení farby na základe toho, ako sa používateľovi darí. Pokiaľ gesto prebieha, je žltá. Ak sa gesto nepodarí, tak zostane červená a ak je gesto úspešne vykonané, na krátku chvíľu sa zmení na zelenú, a potom opäť na červenú.

Klasifikácia

Klasifikácia je implementovaná ako samostatná aplikácia pre každé gesto. Pri spustení systému sa pre každé gesto spustí jeden zvlášť proces s klasifikátorom, ktorý očakáva vyextrahované črty na štandardnom vstupe a navráti rozpoznanú kategóriu motorických schopností na výstupe. Vstup aj výstup z týchto samostatných aplikácií zabezpečuje samotný extraktor čŕt pre každé gesto pomocou samostatného procesu v jazyku C#.



Obr. 4. Rozloženie herných objektov v rámci scény experimentov.

Vyhodnotenie systému

Kategórie motorických schopností a aj gestá, ktoré umožnili ich rozpoznávania boli stanovené po konzultáciách s fyzioterapeutkou z Výskumného ústavu detskej psychológie a patopsychológie v Bratislave. Prvého experimentu sa zúčastnilo 15 používateľov a druhého experimentu 10 používateľov, ktorí boli v oboch prípadoch vybraní zo strany výskumného centra, aby dostatočne pokryli požiadavky vyplývajúce z povahy našej práce.

Najlepší klasifikátor pre náš systém nám vyšiel klasifikátor SVM, ktorý sme zakomponovali aj do druhého experimentu a ním stanovené kategórie sme použili na vyhodnotenie úspešnosti nášho systému. Vyhodnocovali sme pomocou dvoch rôznych metrík – presnosti a makro-priemerovaného skóre F_1. Okrem úspešnosti na základe jedného vyhodnoteného gesta sme však vyhodnotili aj úspešnosť, keby sme na určenie kategórie použili viacero po sebe idúcich gest. Treba podotknúť, že zaznamenávame a klasifikujeme neúspešne prevedené gestá, pretože našim cieľom je prispôsobovanie aplikácie, ktoré nie je také žiadúce, pokiaľ je používateľ schopný gesto vykonať úspešne.

Z obr. 5 a obr. 6 možno pozorovať, že takto dosiahnuté výsledky sú presnejšie ako určovanie kategórie len na základe jedného gesta. Podrobné výsledky pre gesto rotácie sa nachádzajú v tbl. 1 a pre gesto škálovania v tbl. 2. Výsledky gesta prenosu sú horšie, čo je spôsobené aj nedostatkom chybových gest tohto typu počas druhého experimentu. DMO predstavuje kategóriu používateľov po detskej mozgovej obrne, D predstavuje kategóriu používateľov s dyspraxiou a Z kategóriu zdravých používateľov. N predstavuje počet gest potrebných na určenie kategórie.

Prispôsobovanie sa používateľovi na základe kategórie jeho motorických schopností nebolo možné exaktne odmerať, pretože je to veľmi subjektívne a záleží to od konkrétneho používateľa. Na základe pozorovaní používateľov počas oboch experimentov, ich porovnávaním a konzultáciou s fyzioterapeutkou sme však došli k záveru, že používateľom po detskej mozgovej obrne sa bol systém významným spôsobom schopný prispôsobiť. Pri používateľoch s dyspraxiou sme taktiež spozorovali isté náznaky, no neboli také výrazné. Pri zdravých používateľoch sa systém správal zväčša presne tak, ako v prvom experimente a nezaznamenali sme takmer žiadne náznaky, že by sa systém snažil uľahčovať interakciu zdravým používateľom.

Záver

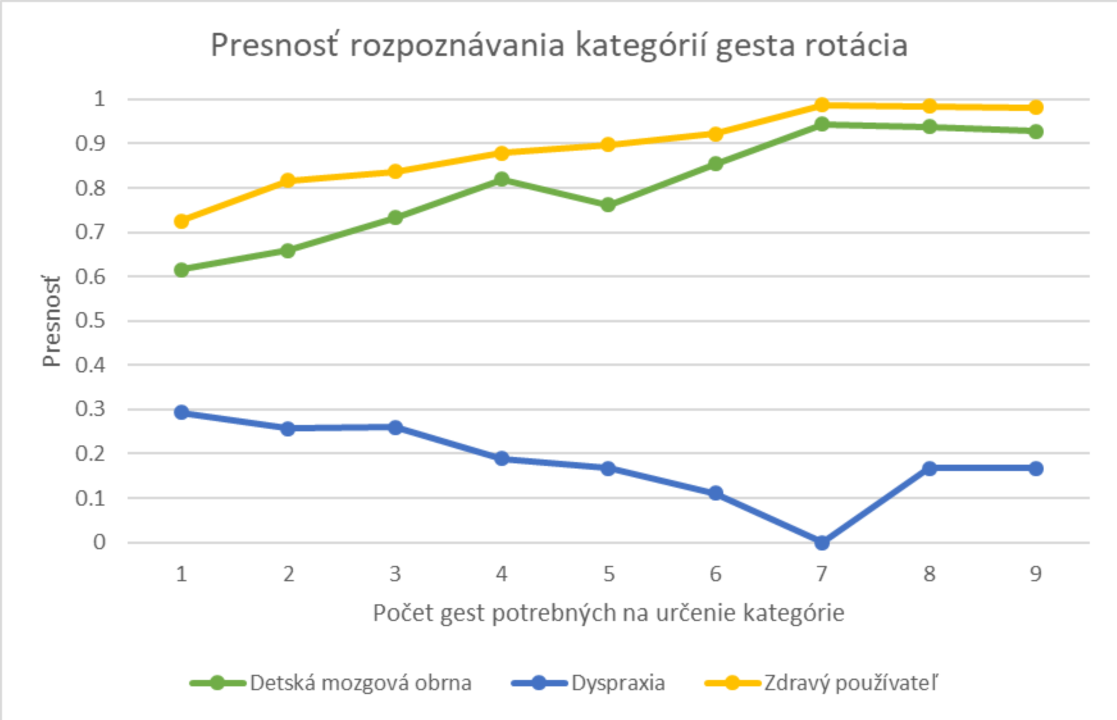
V tejto práci sme vytvorili systém, ktorý bol schopný počas chodu rozoznávať motorické schopnosti používateľa a adekvátne sa u na ich základe prispôsobovať. Na základe výsledkov konštatujeme, že sa potvrdili všetky tri predpoklady, ktoré sme si vytýčili ako ciele tejto práce. Výsledky dokonca prekonali naše očakávania, najmä v prípade vyhodnocovania viacero za sebou idúcich gest, a preto vrelo povzbudzujeme k ďalšiemu výskumu v tejto oblasti a posúvaním pôvodnej myšlienky ďalej – skúšaním nových gest či aplikácií našich záverov v reálnom systéme a nie len v experimente.

Tbl. 1. Metriky pre rozpoznávanie kategórií gesta rotácie.

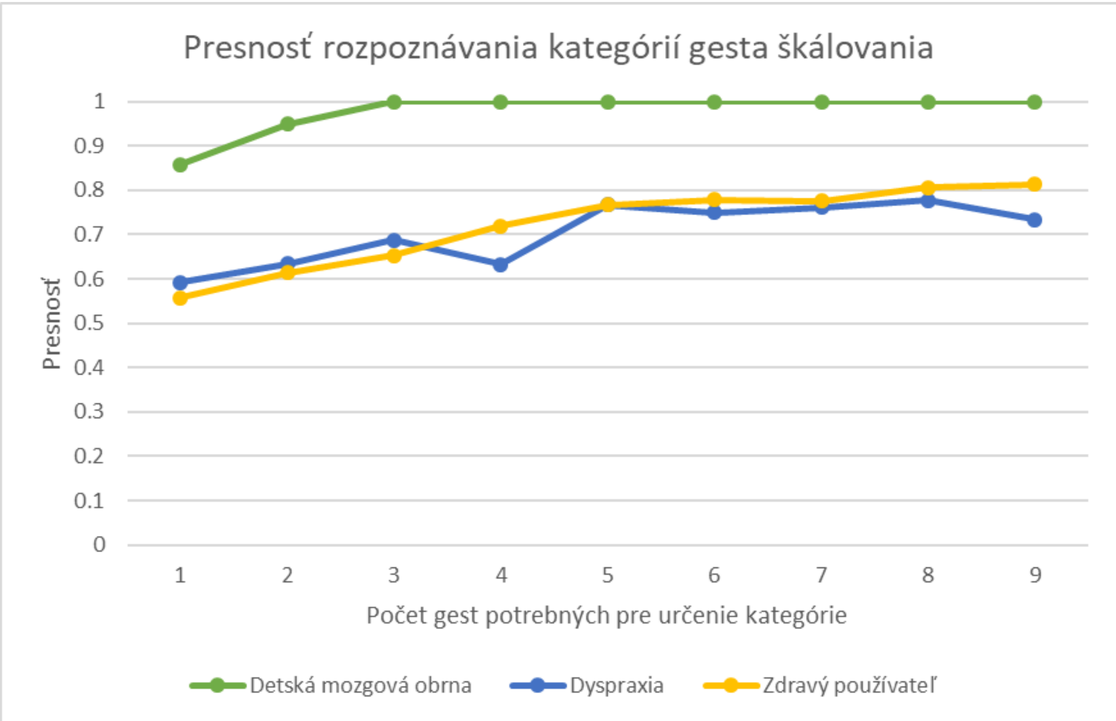
N	Pr. DMO	F_1 DMO	Pr. D	F_1 D	Pr. Z	F_1 Z
1	0.6152	0.2529	0.2929	0.1497	0.7261	0.2784
2	0.6596	0.2632	0.2564	0.1645	0.8173	0.3598
3	0.7333	0.2815	0.2593	0.1611	0.8366	0.3955
4	0.8194	0.4496	0.1889	0.1429	0.8788	0.5393
5	0.7619	0.4308	0.1667	0.1111	0.8976	0.6707
6	0.8545	0.4603	0.1111	0.0833	0.9216	0.7770
7	0.9444	0.7353	0.0000	0.0000	0.9867	0.8966
8	0.9375	0.7333	0.1667	0.1111	0.9846	0.8960
9	0.9286	0.7308	0.1667	0.1111	0.9818	0.8952

Tbl. 2. Metriky pre rozpoznávanie kategórií gesta škálovania.

N	Pr. DMO	F_1 DMO	Pr. D	F_1 D	Pr. Z	F_1 Z
1	0.8586	0.3080	0.5921	0.2900	0.5582	0.2361
2	0.9495	0.4870	0.6336	0.4669	0.6140	0.3317
3	1.0000	1.0000	0.6877	0.5278	0.6526	0.3680
4	1.0000	1.0000	0.6325	0.5083	0.7197	0.4902
5	1.0000	1.0000	0.7667	0.7436	0.7671	0.6030
6	1.0000	1.0000	0.7500	0.7333	0.7786	0.6084
7	1.0000	1.0000	0.7619	0.7407	0.7758	0.6071
8	1.0000	1.0000	0.7778	0.7500	0.8067	0.6157
9	1.0000	1.0000	0.7333	0.7222	0.8139	0.6188



Obr. 5. Presnosť rozpoznávania kategórií gesta rotácie.



Obr. 6. Presnosť rozpoznávania kategórií gesta škálovania.

References

ADAMOVICH, S.V. et al. (2004): A virtual reality based exercise system for hand rehabilitation post-stroke: transfer to function. In: *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2004. s. 4936–4939.

AGGARWAL, C.C. (2014): Data classification: algorithms and applications. Boca Raton: CRC Press, Taylor & Francis Group, 2014. 671 s. ISBN 978-1-4665-8674-1.

ALLEN, R.B. (1997): Mental models and user models. In: *Handbook of human-computer interaction*. 1997. Vol. 1, s. 49–63.

ALTANIS, G. et al. (2013): Children with Motor Impairments Play a Kinect Learning Game: First Findings from a Pilot Case in an Authentic Classroom Environment. In: *J Interact Design Architect*. 2013. s. 14.

ALY, M. (2005): Survey on Multiclass Classification Methods. In: *Neural Netw*. 2005. s. 1–9.

ASLAN, I. et al. (2014): Mid-air Authentication Gestures: An Exploration of Authentication Based on Palm and Finger Motions. In: *Proceedings of the 16th International Conference on Multimodal Interaction*. New York, NY, USA: ACM, 2014. s. 311–318.

BAKER, L.L. et al. (2004): Rehabilitation of the arm and hand following stroke - a clinical trial with BIONs trade; In: *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2004. s. 4186–4188.

BIGDELOU, A. et al. (2012): Simultaneous categorical and spatio-temporal 3D gestures using Kinect. In: *2012 IEEE Symposium on 3D User Interfaces (3DUI)*. 2012. s. 53–60.

BOLLE, R.M. et al. (2013): Guide to biometrics. Springer Science & Business Media, 2013. ISBN 1-4757-4036-0.

CABRAL, M.C. et al. (2005): (a): On the Usability of Gesture Interfaces in Virtual Reality Environments. In: *Proceedings of the 2005 Latin American Conference on Human-computer Interaction*. New York, NY, USA: ACM, 2005. s. 100–108.

CABRAL, M.C. et al. (2005): (b): On the Usability of Gesture Interfaces in Virtual Reality Environments. In: *Proceedings of the 2005 Latin American Conference on Human-computer Interaction*. New York, NY, USA: ACM, 2005. s. 100–108.

CARO, K. (2014): Exergames for Children with Motor Skills Problems. In: *SIGACCESS Access. Comput.* 2014. no. 108, s. 20–26.

CHAN, A. et al. (2015): Leap Motion Controller for Authentication via Hand Geometry and Gestures. In: *Human Aspects of Information Security, Privacy, and Trust*. Springer International Publishing, 2015. s. 13–22.

CHANG, Y.-J. et al. (2013): A Kinect-based upper limb rehabilitation system to assist people with cerebral palsy. In: *Research in Developmental Disabilities*. 2013. Vol. 34, no. 11, s. 3654–3659.

CUI, J., SOURIN, A. (2014): Feasibility Study on Free Hand Geometric Modelling Using Leap Motion in VRML/X3D. In: *2014 International Conference on Cyberworlds*. 2014. s. 389–392.

DENG, L.Y. et al. (2010): EOG-based Human–Computer Interface system development. In: *Expert Systems with Applications*. 2010. Vol. 37, no. 4, s. 3337–3343.

FISCHER, G. (2001): User Modeling in Human–Computer Interaction. In: *User Modeling and User-Adapted Interaction*. 2001. Vol. 11, no. 1–2, s. 65–86.

HEIM, M. (2000): Virtual Realism. Oxford University Press, 2000. 263 s. ISBN 978-0-19-535009-8.

HSU, J.K. et al. (2011): A “Wii” bit of fun: The effects of adding Nintendo Wii® Bowling to a standard exercise regimen for residents of long-term care with upper extremity dysfunction. In: *Physiotherapy Theory and Practice*. 2011. Vol. 27, no. 3, s. 185–193.

HUANG, J.-D. (2011): Kinerehab: A Kinect-based System for Physical Rehabilitation: A Pilot Study for Young Adults with Motor Disabilities. In: *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility*. New York, NY, USA: ACM, 2011. s. 319–320.

IMURA, S., HOSOBE, H. (2016): (a): Biometric Authentication Using the Motion of a Hand. In: *Proceedings of the 2016 Symposium on Spatial User Interaction*. New York, NY, USA: ACM, 2016. s. 221–221.

IMURA, S., HOSOBE, H. (2016): (b): Biometric Authentication Using the Motion of a Hand. In: *Proceedings of the 2016 Symposium on Spatial User Interaction*. New York, NY, USA: ACM, 2016. s. 221–221.

JAIN, A. et al. (2004): An introduction to biometric recognition. In: *IEEE Transactions on Circuits and Systems for Video Technology*. 2004. Vol. 14, no. 1, s. 4–20.

JAIN, A. et al. (2000): Biometric Identification. In: *Commun. ACM*. 2000. Vol. 43, no. 2, s. 90–98.

JAIN, A. et al. (2007): Handbook of Biometrics. Springer Science & Business Media, 2007. 551 s. ISBN 978-0-387-71041-9.

JAIN, A. et al. (2011): Introduction. In: *Introduction to Biometrics*. Springer US, 2011. s. 1–49. ISBN 978-0-387-77325-4.

JIANG, F. et al. (2014): Viewpoint-independent hand gesture recognition with Kinect. In: *Signal, Image and Video Processing*. 2014. Vol. 8, no. 1, s. 163–172.

LANDRY, P. et al. (2013): Design Strategy to Stimulate a Diversity of Motor Skills for an Exergame Addressed to Children. In: *Proceedings of the 12th International Conference on Interaction Design and Children*. New York, NY, USA: ACM, 2013. s. 84–91.

LANGE, B.S. et al. (2010): The Potential of Virtual Reality and Gaming to Assist Successful Aging with Disability. In: *Physical Medicine and Rehabilitation Clinics of North America*. 2010. Vol. 21, no. 2, s. 339–356.

LIU, J. et al. (2009): uWave: Accelerometer-based personalized gesture recognition and its applications. In: *Pervasive and Mobile Computing*. 2009. Vol. 5, no. 6, s. 657–675.

LV, Z., LI, H. (2015): Imagining in-air interaction for hemiplegia sufferer. In: *2015 International Conference on Virtual Rehabilitation (ICVR)*. 2015. s. 149–150.

MANRESA, C. et al. (2005): (a): Hand Tracking and Gesture Recognition for Human-Computer Interaction. In: *ELCVIA Electronic Letters on Computer Vision and Image Analysis*. 2005. Vol. 5, no. 3, s. 96–104.

MANRESA, C. et al. (2005): (b): Hand Tracking and Gesture Recognition for Human-Computer Interaction. In: *ELCVIA Electronic Letters on Computer Vision and Image Analysis*. 2005. Vol. 5, no. 3, s. 96–104.

MARIN, G. et al. (2014): Hand gesture recognition with leap motion and kinect devices. In: *2014 IEEE International Conference on Image Processing (ICIP)*. 2014. s. 1565–1569.

MOUSAVI HONDORI, H., KHADEMI, M. (2014): A Review on Technical and Clinical Impact of Microsoft Kinect on Physical Therapy and Rehabilitation. In: *Journal of Medical Engineering*. 2014. [Accessed 2018-05-31]. Available from: <https://www.hindawi.com/journals/jme/2014/846514/>.

NUGRAHANINGSIH, N. et al. (2015): A Hand Gesture Approach to Biometrics. In: MURINO, V. et al.: *New Trends in Image Analysis and Processing -- ICIAP 2015 Workshops*. Springer International Publishing, 2015. s. 51–58. ISBN 978-3-319-23221-8.

OLIVEIRA, D. et al. (2016): Novel Virtual Environment for Alternative Treatment of Children with Cerebral Palsy. In: *Computational Intelligence and Neuroscience*. 2016. [Accessed 2018-05-31]. Available from: <https://www.hindawi.com/journals/cin/2016/8984379/abs/>.

PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. In: *MACHINE LEARNING IN PYTHON*. s. 6.

PELÁNEK, R. (2017): Measuring Predictive Performance of User Models: The Details Matter. In: *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*. New York, NY, USA: ACM, 2017. s. 197–201.

POTTER, L.E. et al. (2013): The Leap Motion Controller: A View on Sign Language. In: *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*. New York, NY, USA: ACM, 2013. s. 175–178.

RAUTARAY, S.S., AGRAWAL, A. (2015): Vision based hand gesture recognition for human computer interaction: a survey. In: *Artificial Intelligence Review*. 2015. Vol. 43, no. 1, s. 1–54.

SANTOS, O.C. (2017): Psychomotor Learning in Martial Arts: An Opportunity for User Modeling, Adaptation and Personalization. In: *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*. New York, NY, USA: ACM, 2017. s. 89–92.

SANTOS, O.C., EDDY, M.H. (2017): Modeling Psychomotor Activity: Current Approaches and Open Issues. In: *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*. New York, NY, USA: ACM, 2017. s. 305–310.

SINCLAIR, J. et al. (2007): Considerations for the design of exergames. In: *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*. ACM Press, 2007. s. 289.

SINGH, B.P., AGGARWAL, V. (2016): Apps to Measure Motor Skills of Vocational Workers. In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. New York, NY, USA: ACM, 2016. s. 340–350.

TANAKA, K. et al. (2012): A Comparison of Exergaming Interfaces for Use in Rehabilitation Programs and Research. In: *Loading...* 2012. Vol. 6, no. 9.

VALORIANI, M. (2013): Technological and Methodological Tools for Personalized Touchless Applications. In: *CHItaly (Doctoral Consortium)*. 2013. s. 10.

WAYMAN, J.L. (2015): Biometric Verification/Identification/Authentication/Recognition: The Terminology. In: LI, S.Z., JAIN, A.: *Encyclopedia of Biometrics*. Springer US, 2015. s. 263–268. ISBN 978-1-4899-7487-7.

WILSON, A., BENKO, H. (2010): Combining Multiple Depth Cameras and Projectors for Interactions On, Above, and Between Surfaces. 2010. ISBN 978-1-4503-0271-5.

WILSON, M. (2017): Exclusive: Microsoft Has Stopped Manufacturing The Kinect. In: *Co.Design*. 2017. [Accessed 2018-05-31]. Available from: <https://www.fastcodesign.com/90147868/exclusive-microsoft-has-stopped-manufacturing-the-kinect>.

WOLLERSHEIM, D. et al. (2010): Physical and Psychosocial Effects of Wii Video Game Use among Older Women. In: *International Journal of Emerging Technologies and Society; Hawthorn*. 2010. Vol. 8, no. 2, s. 85–98.

YOO, S. et al. (2017): Towards a Long Term Model of Virtual Reality Exergame Exertion. In: *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. New York, NY, USA: ACM, 2017. s. 247–255.

ZHANG, H. et al. (2014): A Personalized Gesture Interaction System with User Identification Using Kinect. In: *PRICAI 2014: Trends in Artificial Intelligence*. Springer, Cham, 2014. s. 614–626.

How Does the Leap Motion Controller Work? In: *Leap Motion Blog*. 2014. [Accessed 2016-12-06]. Available from: <http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>.

Image API Now Available for v2 Tracking Beta. In: *Leap Motion Blog*. 2014. [Accessed 2016-12-09]. Available from: <http://blog.leapmotion.com/image-api-now-available-v2-tracking-beta/>.

Kinect Sensor. In: *Microsoft Developer Network*. 2012. [Accessed 2018-06-06]. Available from: <https://msdn.microsoft.com/en-us/library/hh438998.aspx>.

Appendix A Technical Documentation

The core of the technical documentation in our case is the way of realization of our system, which is described in the main work in detail. However, we would like to also include a class diagram depicted in Figure 12 to complement the text description of the realization for better understanding. In diagram there are not included external processes – classifiers. Each Extractor possess one connection to corresponding classifier.

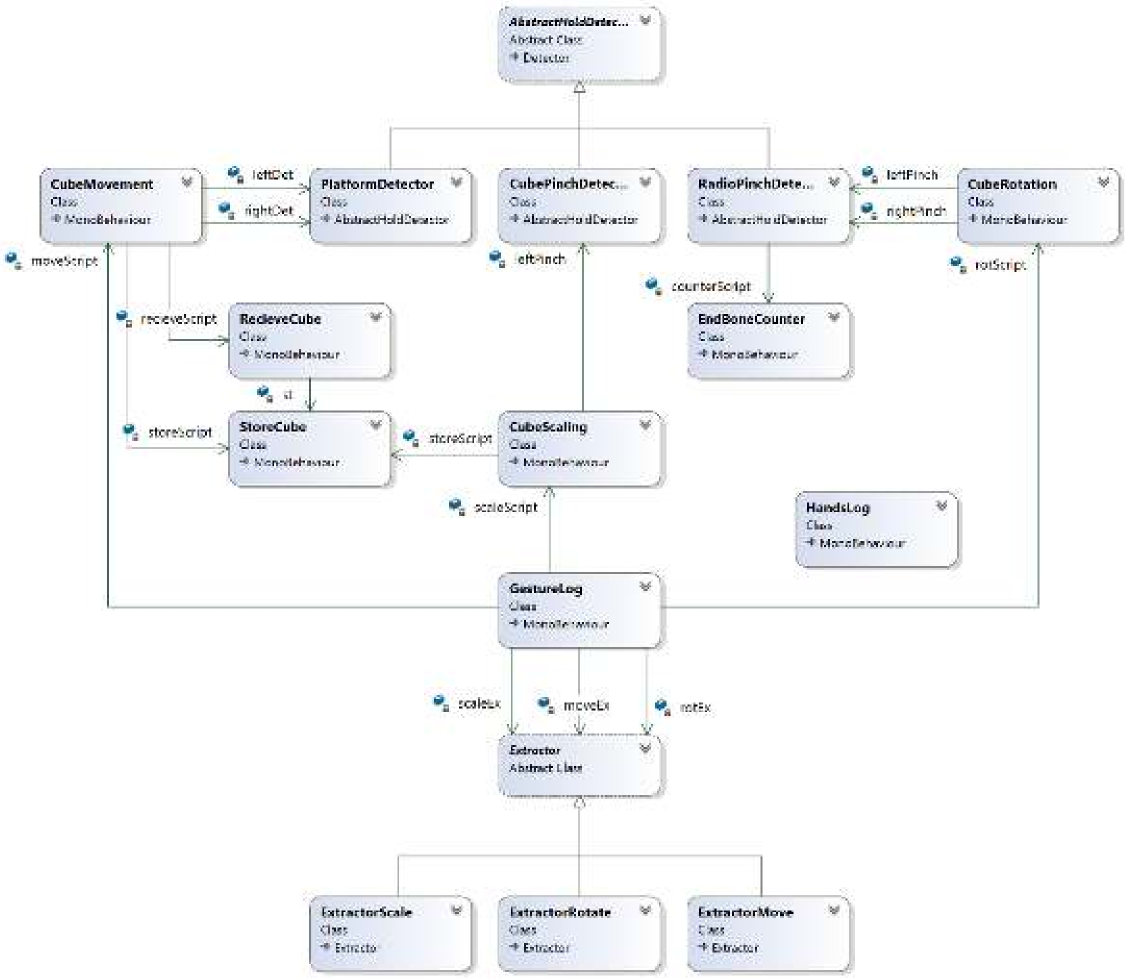


Figure 12. Class diagram of system for experiments.

A.1 Requirements

The system for motor skills determination and adaptation, where both experiments were executed, is designed in Unity engine, therefore does not require anything else but Leap Motion Sensor to be able to run. There are required versions to run our system.

- Unity 2018.3.8f1
- Leap Motion Controller 4.0.0

The data science part, where all results were computed, is included in this work as Jupyter Notebook, and has following main requirements.

- Python 3.6 or newer
- numpy 1.16.1
- pandas 0.24.2
- py-xgboost 0.80
- scikit-learn 0.20.1
- scipy 1.2.1

Appendix B Project Work Plan

B.1 Spring Semester Plan (DP1)

- Research motor skills usage within virtual reality, gesture possibilities for virtual reality, adaptation of systems to users, user modeling, biometrics and available sensors.
    - A lot of researching and reading took place at the beginning of the semester, however, after while it was postponed and delayed and finished at the end of the spring semester.
- Refine assumptions after each research.
    - Done continuously during reading related work.
- Summarize research findings and final assumptions.
    - Summarization was done at the very end of the semester, several weeks after the original due date. The delay was caused by excessive amount of work required by both universities at the same time.
- Create solution idea based on assumptions and related work summarization.
    - Solution was being created continuously during the whole semester.

B.2 Winter Semester Plan (DP2)

- Design training experiment.
    - Designing training experiment was delayed because of evaluation of our proposed gestures and motor skills categories by specialist.
- Choose candidates for training experiment.
    - Candidates were chosen by specialist according to the plan.
- Implement data miner.
    - Data miner implementation had to be postponed due to lack of experiment design and was finished with delay.
- Execute experiment obtaining training data.
    - Experiment execution postponed to DP3, because of communication problems with our specialist.

    - Experiment was executed during the break between semesters even before DP3 officially stared.
- Train multiple classifiers.
    - Postponed to DP3 due to lack of data.
    - Classifiers were trained and evaluated continuously with several different ideas after first experiment almost until the second experiment execution.
- Select the best classifier.
    - Postponed to DP3 due to lack of data.
    - The best classifier was selected much later than expected, but still sufficiently ahead of execution of testing experiment.

B.3 Summer Semester Plan (DP3)

- Design testing experiment.
    - Testing experiment was designed along with training experiment during the DP2.
- Implement real time identification.
    - Real time identification was ready already in DP2, missing classifier to be just plugged in to carry out the result.
    - Real time identification was plugged in with some dependency troubles, which did not delay execution of testing experiment.
- Enhance selected application by user adaptations.
    - Adaptations were implemented according to plan, before execution of testing experiment.
- Execute testing experiment.
    - Testing experiment was executed according to our plan in coordination with our specialist.
- Collect results.
    - Results were computed at the end of the semester.
- Evaluate solution.
    - Solution has been continuously evaluated during DP2 as well.

- Overall solution evaluation was carried out at the end of the semester as planned.

Appendix C Description of Digital Part of Thesis

Evidence number of thesis in information system: FIIT-182905-73652

Content of digital part of thesis (ZIP archive):

```
Content                                     Description
DP_prilohy_digital_LukasBabula.zip
¦   Lukas_Babula-Master's_Thesis.pdf        digital version of thesis
¦   preprocess-train.ipynb                  data processing and evaluation notebook
¦
+---classifiers                             trained classifiers
¦       move_svm_20190320-043305-569627
¦       move_xgb_20190319-234025-402902
¦       rot_svm_20190320-043312-957246
¦       rot_xgb_20190319-234033-593410
¦       scale_svm_20190320-043313-469252
¦       scale_xgb_20190319-234035-579548
¦
+---data-dp                                 data from the first experiment
+---data-dp-eval                            data from the second experiment
+---experiment1                             scene and assets of the first experiment
+---experiment2                             scene and assets of the second experiment
```

Digital part of thesis consists of 2.13 GB of data. Therefore, it is stored in G Suite for Education.

Name of the turned in archive: DP_prilohy_digital_LukasBabula.zip