Capture and analysis of screen touch patterns for individual user recognition in a figure-matching game developed for the elderly

Submitted by:
Llano Jiménez Santiago
Zurzacherstrasse 24b
5200 Brugg
santiago.llanojimenez@students.fhnw.ch

FHNW
University of Applied Sciences and Arts
Computer Science - Profile iCompetence

Supervising lecturers:
Prof. Dr. André Csillaghy and Marco Soldati

Windisch, August 16, 2019

Contact Details

Name and role	Address	phone / e-mail
Marco Soldati, client and supervisor	University of Applied Sciences FHNW Institute for 4D technologies Bahnhofstrasse 6 5210 Windisch	+41 56 202 77 31 / marco.soldati@fhnw.ch
Prof. Dr. André Csillaghy, supervisor	University of Applied Sciences FHNW Institute for 4D technologies Bahnhofstrasse 6 5210 Windisch	+41 56 202 76 85 / andre.csillaghy@fhnw.ch
Heike Warnke, contact person Old people's home in Eigenamt	Flachsacherstrasse 2 5242 Lupfig	+41 56 464 22 14 / heike.warnke@haus-eigenamt.ch
Santiago Llano Jiménez, author/ student	Zurzacherstrasse 24b 5200 Brugg	+41 76 310 86 82 santiago.llanojimenez@students.fhnw.ch

Declaration on honour

The present work was produced independently and without the use of other sources and aids than those listed in the bibliography. The references taken literally or in terms of content from the sources and aids listed in the bibliography are identified in the work as quotations or paraphrases.

Signature

Santiago Llano Jiménez

Santiago Llano Jiménez

Windisch, August 16, 2019

Preface

I would like to take the opportunity to thank Prof. Dr. André Csillaghy who has served as the main advisor and Marco Soldati as advisor and contractor for the useful recommendations and feedback during the course of the project and elaboration of this work. Special thanks are also extended to Mrs. Heike Warnke, who works in the competence centre for elderly people in Lupfig AG for her full willingness to organise regular visits with the inhabitants of the centre and for her feedback. Thanks are now due to all those who participated in the development of the usability tests and the collection of vital data for the project.

Abstract

Automated recognition of users based on features of their interaction with touchscreens is a feature that is applicable in many fields such as user authentication. It may also hold promise for mobile games, especially for multi-player games. The purpose of this study was to develop a game for elders that would generate and store screen touch pattern data of the players, to process and analyze these data, and to identify steps of analytical procedures that may be useful in developing touch-based user recognition algorithms. The game ShapeMatch was developed using Unity and C# and a JSON structure was established to store the screen touch pattern data. Similarity or dissimilarity of trajectory characteristics within single users was calaculated applying the Dynamic Time Warping algorithm. Results showed high variability of the trajectory characteristics within single players suggesting that other features of the screen touch pattern data should be included in further analyses. Additional suggestions for future work in the area include the use of time series segmentation to identify and analyze segments of the trajectories or the application of classification algorithms to support descriptive analysis.

Table of Contents

1. Introduction

Automatically recognizing individual users or differentiating between groups of users sharing the same characteristics has many potential applications in a world where mobile technology has become almost ubiquitous and readily accessible for large proportions of society. The present project aimed at outlining and executing the steps necessary to evaluate whether user recognition based on screen touch patterns generated by players of a mobile game was feasible in a circumscribed use context.

The two-phase project was successful (a) in developing a game with high usability for the target group that generates screen touch pattern data useful for data analysis, and (b) in preprocessing, analyzing and descriptively interpreting the results of similarity analyses for the screen touch pattern data generated by players of the game during data collection phase.

ShapeMatch is a game in which the users are prompted to connect matching figures that randomly appear in a grid on the screen by passing their finger from one figure to the other. The game can be played in single-player mode or in two-player mode. A score is shown based on the number of matched pairs of figures, and the time since the start of the game is recorded. ShapeMatch was designed to be easily understandable especially for elders with potentially little experience with technology in general, and particularly with touch screen use.

The trajectories of the players' fingers on the screen are stored into a JSON structure. After preprocessing all data, trajectories, in the form of time series, of all successful matches of figures were compared within the most successful players. Dynamic Time Warping coefficients were calculated to determine the degree of similarity or dissimilarity between single time series. Within players, considerable dissimilarity was observed between a high number of pairs of trajectories. This suggests that the characteristics used in Dynamic Time Warping calculations show considerable variability within one single user. They may thus not be ideal characteristics of screen touch patterns to be used in automated user recognition. This finding will inform further investigation in the area and help determine useful and feasible mechanisms to eventually identify what characteristics of screen touch patterns should be used to identify users or user groups, and how those can be analyzed.

This report starts with a theoretical part that embeds the project and its aims in a wider use context. The potential utility of touch-based user recognition is outlined, and procedures in the field of pattern recognition are briefly introduced. The Dynamic Time

Warping algorithm is then presented more in detail as it holds promise for the main purposes of this project. The practical part describes the two phases of the practical work, beginning with an introduction to the architecture, design and usability of Shape-Match. Subsequently, the second practical phase is described, namely the data set, data preprocessing, and the steps in applying Dynamic Time Warping to the data. The results from the analyses are then presented and discussed with regards to potential limitations of the analyses, and suggestions for future work are outlined. The report is rounded off by a short summary and conclusion.

2. Theoretical part

The demographic development in most Western countries, including Switzerland, leads to a gradual increase of the proportion of the elderly in society [1]. Initiatives to foster contact across generations is becoming increasingly important if social isolation is to be reduced in old age. The Myosotis project aims to stimulate contact between generations, specifically elders and their relatives, through the medium of games played on tablet computers. By playing together, contact can be initiated, maintained, prolonged, and enriched, which should lead to increased well-being of the involved individuals [2].

When multiple users play a game, their performance and user experience can vary significantly as a function of both the players' characteristics and the context the game is played in [3], [4]. Examples for players' characteristics that influence performance and user experience are age, cognitive functioning, physical health impairments such as trembling of the hands, motivation and ambition, familiarity with the medium and/or the game, level of fatigue, and many more. Examples for characteristics of the context are lighting or level of ambient noise. If there is significant discrepancy between players' performance within a game, the experience of playing together may be less favorable. For example, the more competitive player becomes bored while the less competitive player feels frustrated, leading the two to quit the game and end the interaction.

As ending the interaction is an unwanted outcome, the aim is to keep all players entertained and engaged in the game. This could be achieved by establishing more balance between the players' performances. Ideally, the game automatically counteracts the imbalance in performance between players in order to improve satisfaction with the interaction for all involved players. Various options to counteract the imbalance are possible, ranging from individual scoring information in same-time multiplayer mode to automated adjustment of level of difficulty for all players.

The first step to implementing measures counteracting imbalance in player performance is automated user recognition. In order to recognize individual users or groups of users sharing certain characteristics, the game must analyze data that is generated by the users while playing the game. One such set of data are screen touch patterns generated by the players, i.e. the trajectories the players create on the screen while navigating through the game. The availability of built-in sensors in most mobile devices for measuring motion and orientation makes screen touch patterns a pertinent choice of data to be processed and analyzed for user recognition [5], [6].

Efforts at individual user recognition with the use of screen touch pattern data have been reported before. For example, touch and context behavioral features such as touch location, zoom length/curvature and Dynamic Time Warping (DTW) and One Nearest Neighbor (1NN) methods allow identifying gesture patterns on the touch screen which are subsequently used to compare biometric signatures for user authentication in uncontrolled environments [7].

The analysis of screen touch pattern data for user recognition requires the application of methods developed for pattern recognition. Pattern recognition is an important field of computational and data science that is gaining increasing attention in a variety of contexts: Biologists may use pattern recognition for the analysis of microscopy-derived image data sets [8], in psychology, pattern recognition is being used to study synchronization of heart rates and other psychobiological parameters of interaction partners [9], and pattern recognition is the base of automated image recognition and comparison that is already widely used in image searches, automatic alternative texts for images, and other applications.

Due to the variety of applicable fields, many methods of pattern recognition have been developed. One well-known example is Convolutional Neural Networks (CNN or ConvNet), a method that can take image input, assign importance to its aspects, and thus differentiate images from each other. The structure of a CNN is very much comparable to the neural networks in human brains, specifically the organization of the visual cortex. The CNN gradually reduces the image input into forms that are easier to process without omitting key defining parts [10]. Many other pattern recognition approaches haven been developed, varying greatly by field of application and process (for an overview see for example [11]).

For the purpose of analyzing screen touch patterns within the context of the Myosotis project, a very promising method of analysis is the Dynamic Time Warping (DTW) algorithm. DTW has been developed in the context of speech recognition, but it is also much more widely used in areas such as stock market analyses and decoding of the human genome, among many others. DTW is a mechanism to analyze time series, i.e. categorical or continuous data that are generated not simultaneously, but over a period of time, so the data structure is extended by a temporal dimension [12].

DTW was developed to counteract the disadvantages associated with analyzing time series with the Euclidean Distance. Euclidean Distance measures the distance between two points present in the same timeline. It is thus useful for classifying or recognizing two or more images of equal size when comparing the similarities between their pixels [13]. However, when measuring the distances between time series of two or more paths, the Euclidean distance has the disadvantage of only being able to operate efficiently with paths of the same length, i.e., with the same number of time series. Dynamic Time Warping (DTW), in contrast, allows finding correlations between two or more trajectories, including those of heterogeneous length, when determining the minimum distance between their time series. It does so by not only making an ordinary measurement, that is, exclusively comparing the positions between equal time series, but also comparing the distances between a time series of a trajectory with respect to multiple time series of the other trajectory.

DTW has been shown to be efficient in finding similarities between specific intervals of the compared trajectories regardless of whether these similarities occur at the same time or not [11]. This is an important feature that makes DTW promising for use in screen touch pattern recognition and comparison. When playing mobile games based on screen touch input, players are not restricted regarding where and at what speed they move across the screen1. The screen touch patterns generated by the players may thus vary greatly in duration and timing, a factor that can be remedied using DTW to compare the screen touch patterns generated by players.

Testing the application of Dynamic Time Warping in the analysis of screen touch patterns from users of a mobile game requires several steps. The aim of this P-5 project is thus to develop a game within the Myosotis framework that is structured such that it stores the generated screen touch patterns in a meaningful way to after be extracted and fed to the DTW algorithms. The following practical part will describe the development of the game, the processing and preparation of the data, the actual data analysis and the results that have been obtained during the analysis.

3. Practical Part

The aim of this part of the report is to describe in detail the two essential phases of the project: the development of the game ShapeMatch that was used to generate players' screen touch patterns, and the actual data analysis to compare single screen touch trajectories.

3.1 Phase 1: Development of the game ShapeMatch

ShapeMatch is a single and multiplayer game especially designed for the elderly with the purpose of collecting their touch gestures consisting of a grid containing figures which are repeated two or four times. The objective from the player's perspective is, through touch gestures, to join as many equal figures in the shortest possible time.

3.1.1 Elements of ShapeMatch

Game Scenes

Shapematch consists of three game scenes: The first scene corresponds to the configuration of the game in which the player determines how he wants to play before starting a round (Figure 1). In the second scene the player views the grid with the figures to match. Matching figures disappear once they are paired across a line created by the player's touch gesture (Figure 2). The third scene corresponds to the paused game. The player has the possibility to pause the game by pressing the ⏸ button which can be found in the grid scene described before (Figure 3).

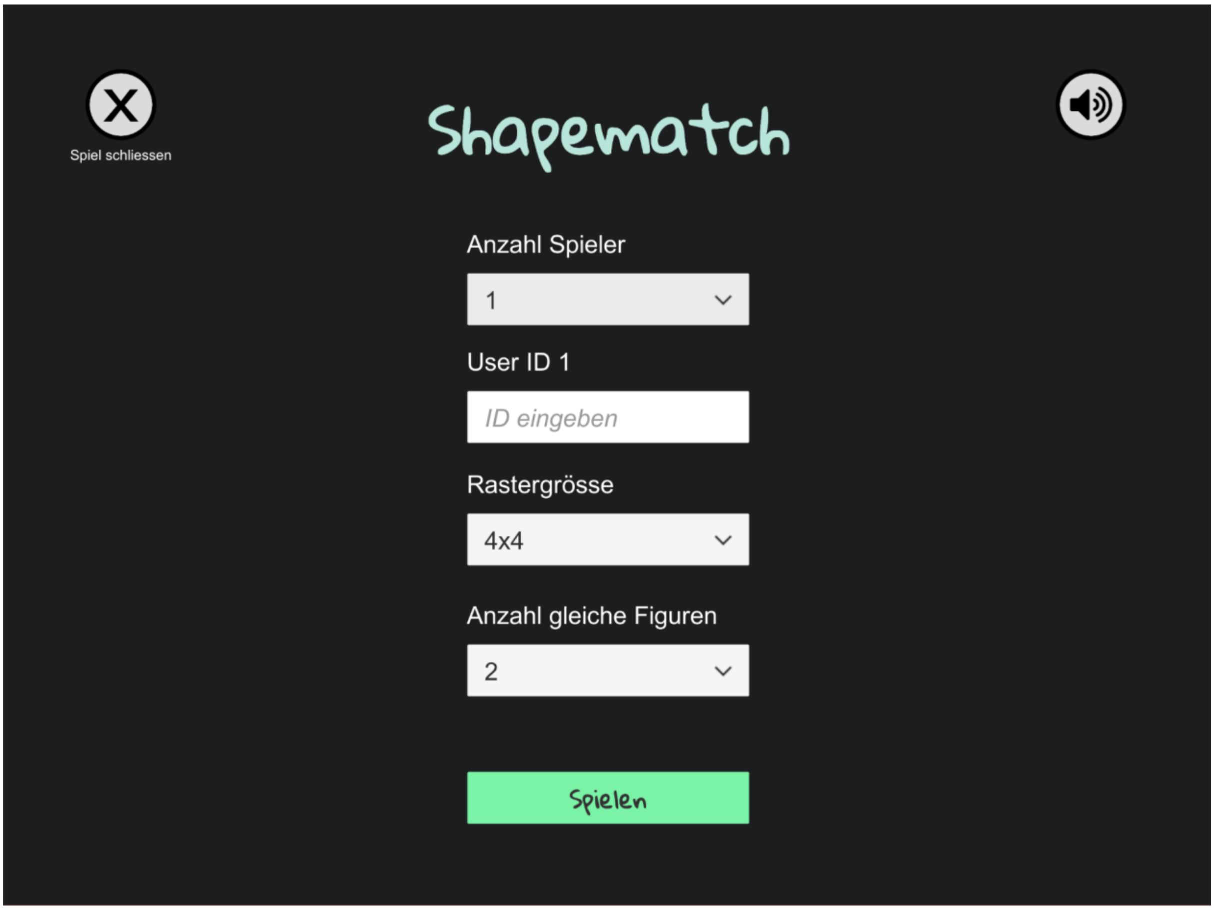


Figure 1: Illustrtion of the initial configuration scene containing a series of dropdown menus and text fields with which the player determines the way he wants to play the game round.

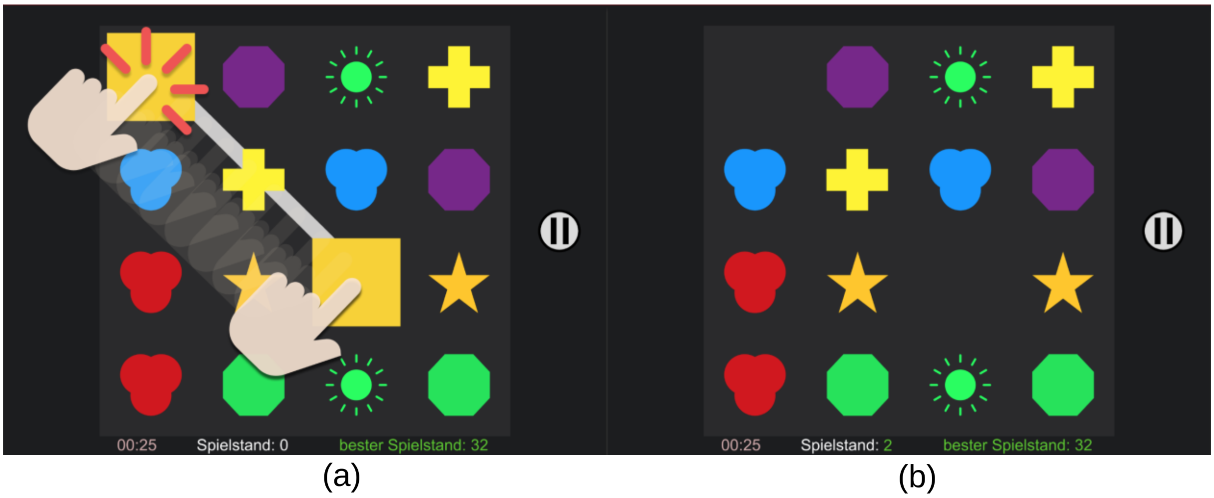


Figure 2: Illustrations of the Grid scene. The image on the left (a) simulates a touch gesture movement in which two figures of the same type are connected. The grey line is created once the player touches a figure and keeps moving his finger along the screen creating an independent trajectory. The image on the right (b) is the result of joining two figures of the same type in which both figures disappear. Note that now the player has received two points equivalent to the two figures joined together.

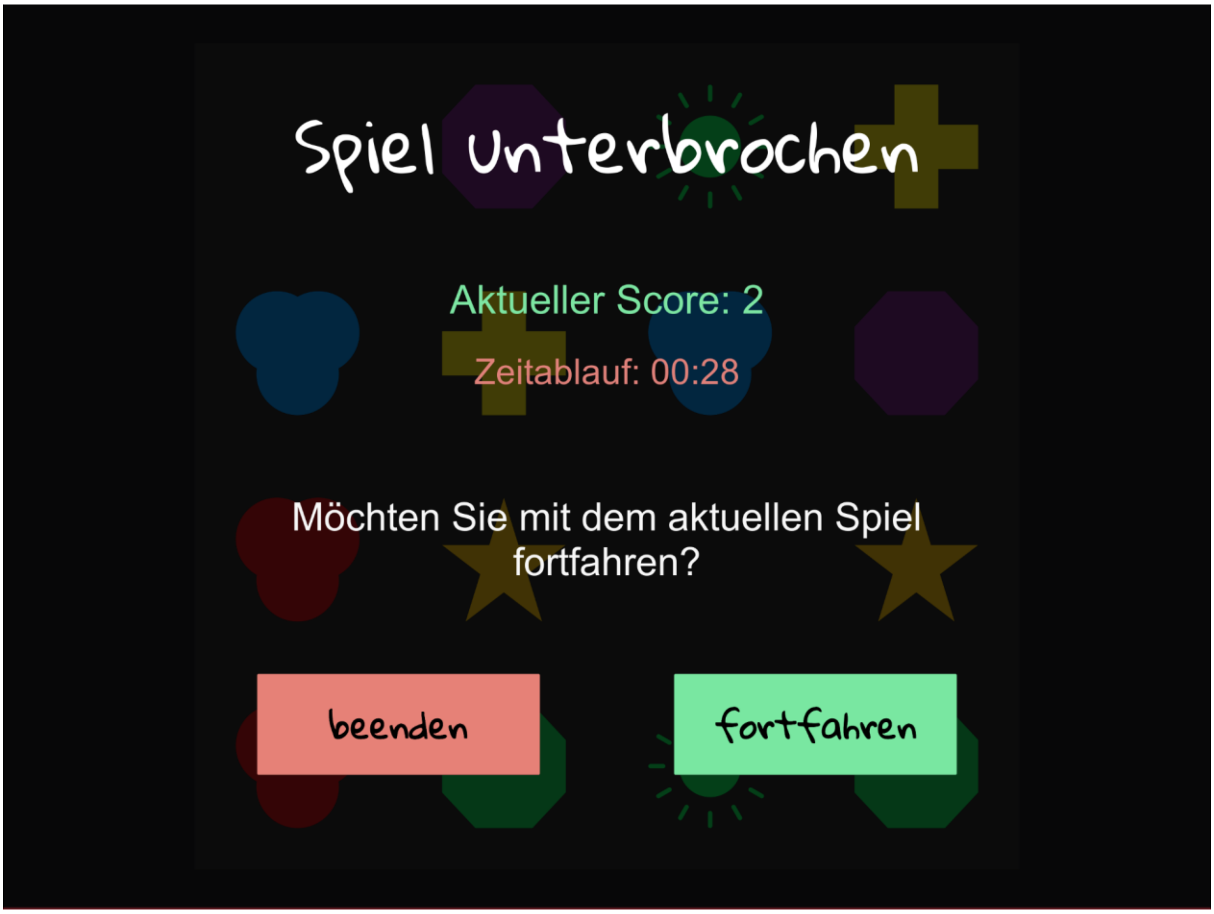


Figure 3: Illustrates the paused game scene in which the player can continue with the game round or return to the initial setup scene

Main Classes

Figure 4 is an overview of the most important classes required for ShapeMatch to function properly:

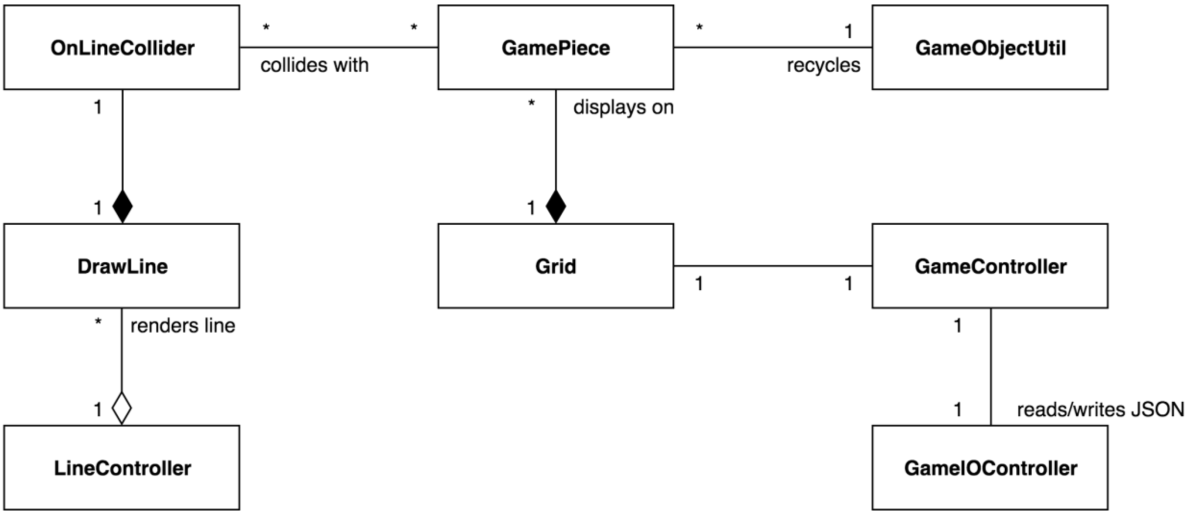


Figure 4: UML diagram displaying the main classes for ShapeMatch and its associations

Description of the relationship between classes in a real case scenario

When the player fills in the input fields in the initial configuration scene and clicks the "Starten" button, the input data is stored in the GameController class and the Grid scene is loaded. The values stored in the GameController are accessed by the Grid and LineController classes which are instantiated with the scene change. The Grid class has as its attribute a two-dimensional array of the GamePiece object which contains the figures to be displayed in the Grid. The LineController class creates an instance of the DrawLine class creating a visible line every time the player touches a GamePiece with his finger and moves his finger without removing it along the screen. The OnlineCollider class observes the collisions between the created line and each GamePiece, when the user joins with the line the number of GamePieces of the same type to join (2 or 4) these are recycled by the GameObjectUtil class, that is to say, they go to "inactive" state instead of being destroyed and disappear from the screen. Finally each inactive GamePiece is enabled again when the Grid becomes empty and is refilled as shown in Figure 2.a and Figure 2.b.

Textures

ShapeMatch uses 50 geometric shape images in a variety of colors to allow easy recognition from different viewing angles. Figure 5 illustrates the figures that are randomly displayed in the Grid.

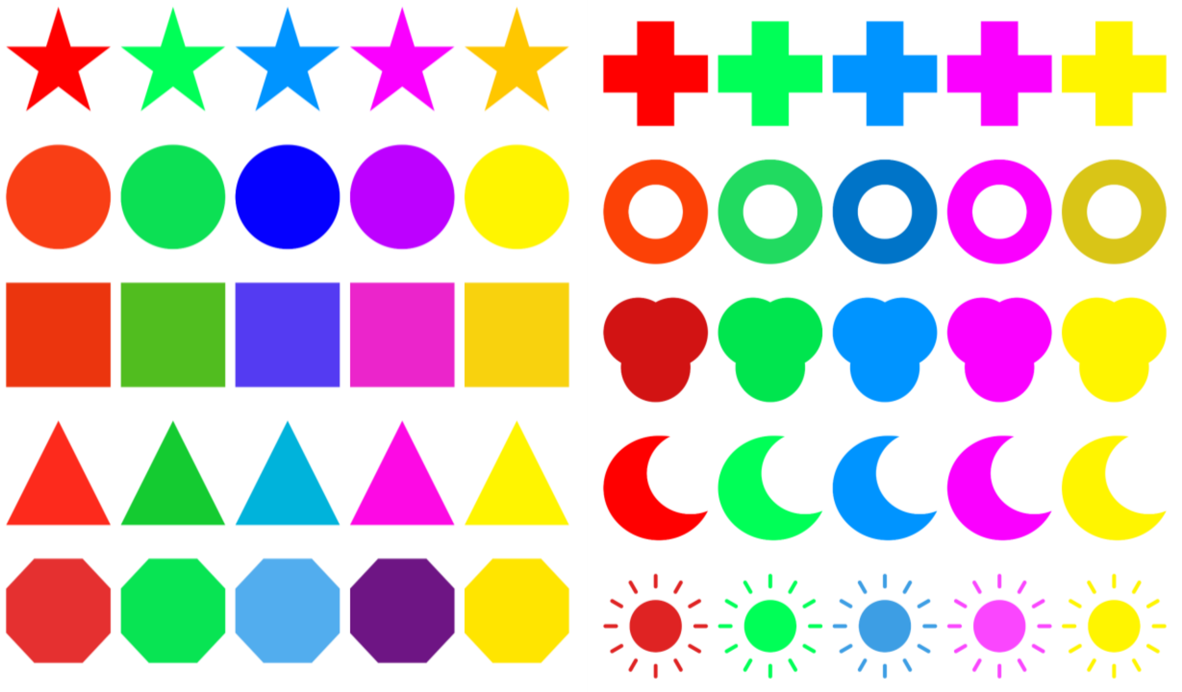
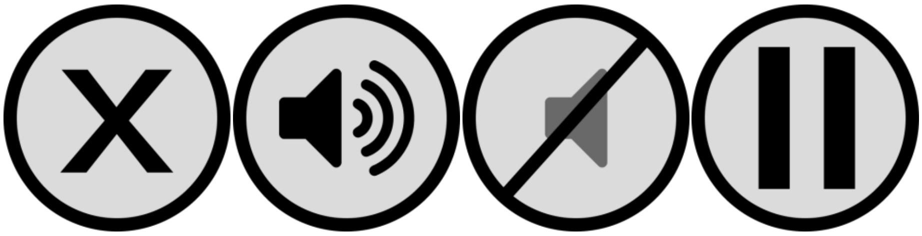


Figure 5: Set of textures (figures) that are randomly visible in the Grid scene

Icon Buttons

Figure 6 depicts the icons present in the various scenes of the game. The (a), (b) and (c) buttons appear in the initial scene and have the function of closing the game, de-activating and activating the sound accordingly while the button in (d) is visible in the grid scene and is used to go to the paused game scene.



(a) (b) (c) (d)

Figure 6: Button icons that appear in different scenes of the game

3.1.2 Collection of Screen Touch Pattern Data with ShapeMatch

When the player returns from the grid scene to the initial scene configuration scene by pressing the button, the game round id, the score as well as the collected touch data are converted into JSON format and stored in the application folder.

Below is a simplified version of the nodes contained in the JSON structure for each game round. The "//" are not part of the JSON structure since they only represent comments on the attributes contained in it. A more detailed information about user data touches can be found in 3.2.

```
{
 "id": 2, // the game round id
 "userData": {
   "uid1": "unbekannt",
   "uid2": ""
 },
 "score": {
   "matchedFigures": 2, // the total number of paired figures
   "time": { // start and end of the game round in  unix milliseconds
```

```
      "initialTime": 1561813049324,
      "finalTime": 1561813121469
    },
    "gridSize": "4x4", // 4 columns x 4 rows
    "numberOfPlayers": 1
  },
  "touchDataList": { // contains two lists of trajectories. Each trajectory is a list of points
    "matched": [ // list of trajectories (each trajectory is a list of points) of matched figures
      {
        "points": [ // the points list contains the points belonging to a trajectory
          {
            "deltaTime": 7.554210662841797, // time elapsed since the start of the game in which the point was created
            "x": 0.7624450922012329, // normalised position on the x-axis obtained by dividing the original position in pixels by the width of the screen
            "y": 0.6103515625
          }],
        "sameFigures": [ // it always contains the 2 or 4 figures paired by the trajectory
          {
            "x": 0.24444301426410676,
            "y": 0.8409090638160706,
            "deltaTimeMatch": 8.821630477905274 // time elapsed when the figure was matched
          }]
      }],
    "unmatched":[ // This list is analogous to that of the matched
      {
        "points": [
          {
            "deltaTime": 72.12589263916016,
            "x": 0.9194729328155518,
            "y": 0.50341796875
```

```
    }],
   "sameFigures": []
 }]}}
```


3.1.3 System Requirements for Installation

ShapeMatch was created with Unity and C# for Mac and is available for iPad 12.9" devices.

Technology/Device	Version / Model
Unity	2019.1.0f2
C#	7.3
iPad	Apple iPad Pro 12.9"

The source code of the game can be found in the following repository: git@gitlab.fhnw.ch:touchbasierte-nutzererkennung/game/shapematch.git

3.1.4 Usability Testing

The residents of the old people's home in Lupfig and people close to the executors of the present project were involved in conducting Usability Tests. Three visits during the development phase of the ShapeMatch game and one more visit during the user touch data collection phase (3.2) were carried out in the nursing home.

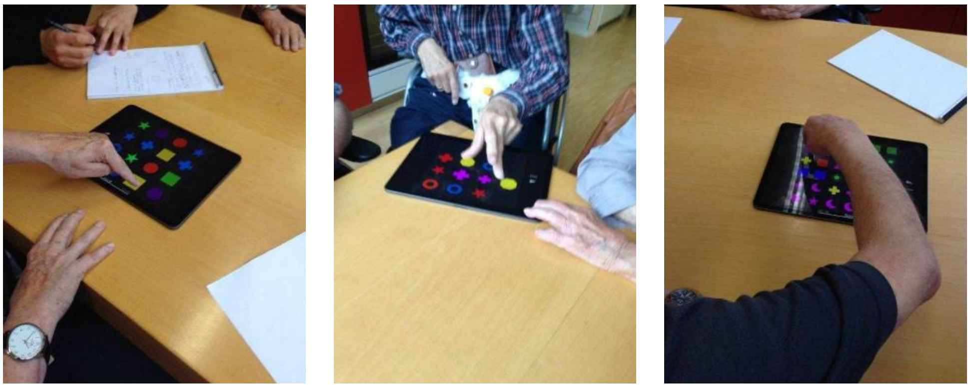


Figure 7 - Usability test images carried out with residents of the Competence centre for elderly people in Lupfig, AG

Results of the first usability test session

During the first visit, four participants (3 men and 1 woman) had the opportunity to make a first evaluation of the game and were asked to comment as clearly as possible on their impressions of the user experience with the following results:

The Figures as shown in Figure 8 visible in the Grid scene of the game were confusing for some participants. They had difficulty recognizing the figures to be matched. The main reason was that the shapes of most of the figures were very complex and tended to be distorted according to the position from which they were observed. The counter-measure taken to address this problem was to replace the set of figures with geometric figures allowing recognition from all angles of visibility.

Figure 8: A sample of old figures initially used in the game that were replaced by the current geometric figures.

All participants had inconveniences at the moment of making the collisions between the figures and the trajectories created from the touches of screen with the finger. The main reason was attributed to the fact that the size of the RigidBody 2D Component of each Figure, responsible for the collision recognition caused by each touch generated by the player, was too small. As a countermeasure it was decided to increase the size of each RigidBody even exceeding the original size of the figure itself, as illustrated in Figure 9.

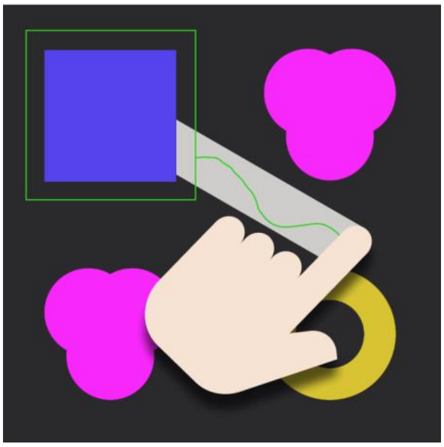


Figure 9: Simulation of RigidBody 2D component (green outline) responsible for detecting interaction between the touch created by a user and the figure.

Results of the usability test taken from the second visit

For the second usability test session, five elders were involved. The replacement of the figures shown in the Grid scene considerably improved the performance of the players of the game, however two of the players had to strive to differentiate figures with the same shape but different color. As a countermeasure it was decided to apply more variation in the selection of the colours for every geometric figure. A bug was detected as a result of a fast matching movement of figures executed by one of the participants in which the connected figures did not disappear from the screen. As a countermeasure, the problem was identified in the algorithm responsible for the temporary storage of paired figures which, due to the fast movement of touch gesture, did not succeed in being executed.

We received a recommendation from one of the participants to incorporate both sound and animation of the figures in such a way that the user's interaction with the game could be visible. To take up this suggestion we incorporated an animation to increase the size of the figure once it is touched with the finger. We also added a background sound in the initial scene of the game as well as sounds that occur when the figures have been or have not been successfully paired.

Results of the usability test of the third visit

The third round of usability tests was aimed at the first capture of touch data from the players. However, there were inconsistencies in the storage of data in JSON format as not all user touches appeared in the captured data. In addition to this, the positions of the paired figures did not correspond with the original position of the screen. Furthermore, the time data (represented in unix timestamps) did not correspond with the actual moment when the figure was paired. As a countermeasure, the erroneous positions of the figures were adjusted to the actual positions, the time format was changed to the deltatime value used by unity which represents the time elapsed in seconds when rendering the next frame.

Usability test results of the fourth visit

The fourth visit involved three residents and a visitor from the nursing home where they were asked to carry out continuous sets of Shapematch rounds in order to collect data for further analysis of screen touch patterns.

3.2 Phase 2: Analysis of Screen Touch Patterns by means of time series comparisons

The analysis phase referred to in this section consists of several steps. First we will begin by describing the characteristics of the participants and their origins as well as information concerning the touch data produced by them when playing different Shape-Match game rounds. In a second part we explain the relevant concepts that are used throughout this section concerning the touch data collected and how these are represented. In the third part we will explain the criteria taken into account for the processing, grouping and normalization of the data. Finally, we carried out the experimental analysis through an extended version of the method called Dynamic Time Warping aimed at calculating similarities between multidimensional data and explain the results of the analysis using the statistical criteria of mean and standard deviation.

3.2.1 Participant Characteristics

Data analysis was carried out using the data sets of 15 participants, six being women and nine being men. The majority of participants (13) were adults (aged between 19 and 92) coming from different professional, personal and educational backgrounds such as engineering, teaching, research assistance, social work, educational institution management, homemaking, packaging. Three adults are pensioners residing in a nursing home. The remaining participants are two middle school children aged 12 and 13.

The participants were asked to perform the same task of playing different rounds of ShapeMatch with the aim of joining as many figures of the same type as quickly as possible. The total number of touch points generated by all participants was 193.910. This study focuses exclusively on trajectories corresponding to successful figure pairings reaching a total of 10.674 trajectories of this type composed of 163.008 three-dimensional points.

User ID	Age	Number of trajectories
0	68	1072
1	29	672
2	62	927
3	38	1176
4	29	1127
5	52	1008
6	19	624
7	52	1082
8	48	760
9	75	518
10	92	116
11	84	104
12	84	337
13	12	346
14	13	384

Figure 10: Table showing the number of trajectories created by each participant.

3.2.2 Definition of trajectory, point, sequence and time series terms

When playing a round of ShapeMatch the player creates a set of trajectories corresponding to either successful figure pairs or unsuccessful figure pairs. Each trajectory $T = (P_1, P_2, \ldots P_i, \ldots, P_n)$ of length n is composed of a set of points P_i with $0 \leq i \leq n$. Each point is composed of three dimensions $P_i = (x_i, y_i, z_i)$ out of which the dimensions x_i, y_i correspond to the normalized horizontal and vertical screen positions deduced by dividing their original pixel values by the screen width and height accordingly, while z_i corresponds to the time of creation of that point with respect to the starting point of the trajectory.

For the creation of trajectories three phases are considered: the beginning of the screen touch corresponds to the initial point of the trajectory. When the player moves his finger along the screen without removing it, the trajectory acquires new points. The final point of the trajectory is generated once the player removes the finger from the screen.

The term time series refers to a collection of observations interpreted as a sequence of real numbers collected regularly in time. In our case study there are two collections of observations which represent the spatial positions across to the X and Y coordinates of a trajectory. [14, p. 215].

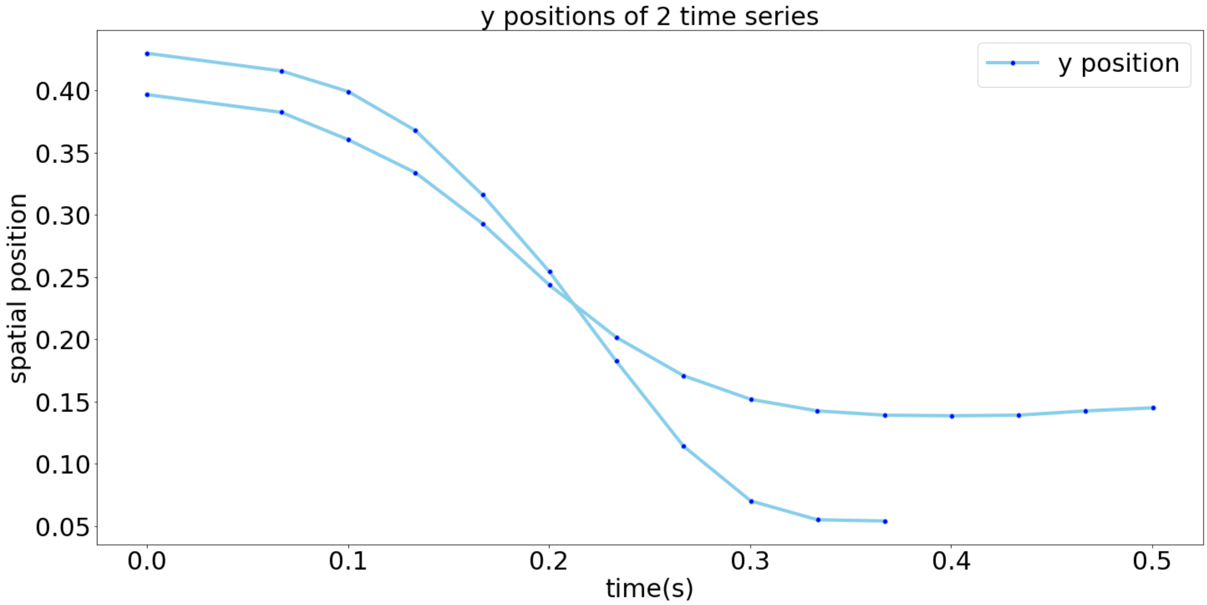


Figure 11: Plot of one dimensional time series corresponding to the positions of the vertical displacements (y-axis) of two different trajectories

3.2.3 Data Preprocessing

Data Analysis Constraints

Only the trajectories belonging to successful matches of figures in single player game rounds have been considered for the analysis. This choice is guided by the following reasonings:

- In multiplayer mode, there is a risk of "crossing" between both players in which, due to the risk of collision, players are forced to adapt their movements according to the behavior of their partner which implies limitations for an unrestricted free movement along the screen as seen in Figure 12.
- While some unsuccessful figure matching paths might be useful for comparing the level of matching success versus the level of matching failures, the noise level of low-value data and the lack of sufficient points per trajectory would not make it a good criterion when making comparisons. This is due to the fact that

in many cases the screen touches correspond instead to involuntary screen touches.

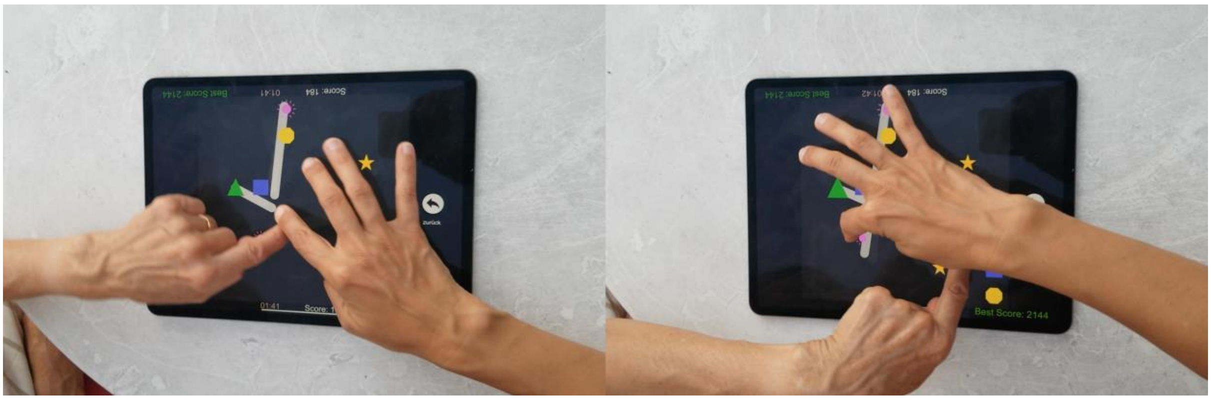


Figure 12: Illustration of the "crossing" effect in a multiplayer game round in which players' fingers collide involuntarily preventing free movement along the screen.

Listing and grouping of trajectories according to paired figure positions

The trajectories of all the game rounds have been grouped according to the position of the figures paired in each of them. The trajectories have been listed containing the following information:

Group id	Uid1	Uid2	X	Y
Group to which the trajectory pertains according to the position of the paired figures.	id for the user 1 who played the game round converted into a unique integer	id for the user 2 who played the game round converted into a unique integer	Array of positions on the x-axis of the trajectory points	Array of positions on the y-axis of the trajectory points

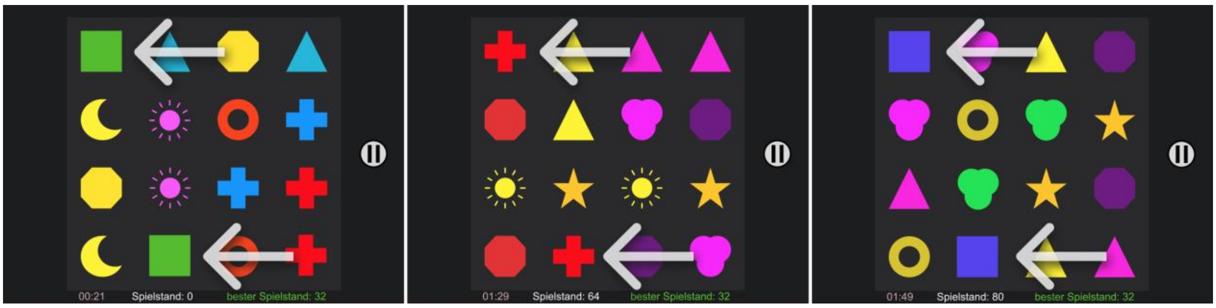


Figure 13: Illustration of figures that appear in the same position of the grid. The trajectories that match these figures belong to the same group.

Z-normalization of time series

Time series normalization is an important step in data preprocessing as it allows time series not to be scaled with respect to input units, thus avoiding unwanted shiftings. Since the time series from our dataset can have different units and scales, it is necessary to apply normalization in such a way that they are adjusted to a specific position range. For this case study, a procedure called z-normalization (1111) was used, also known as "Normalization to Zero Mean and Unit of Energy" [15, p. 2], and in which a transformation for the set of values belonging to the time series sets the mean value to 0 and the standard deviation approximately to 1.

$$x_i' = \frac{x_i - \mu}{\sigma}, \text{ where } i \in \mathbb{N} \tag{1}$$

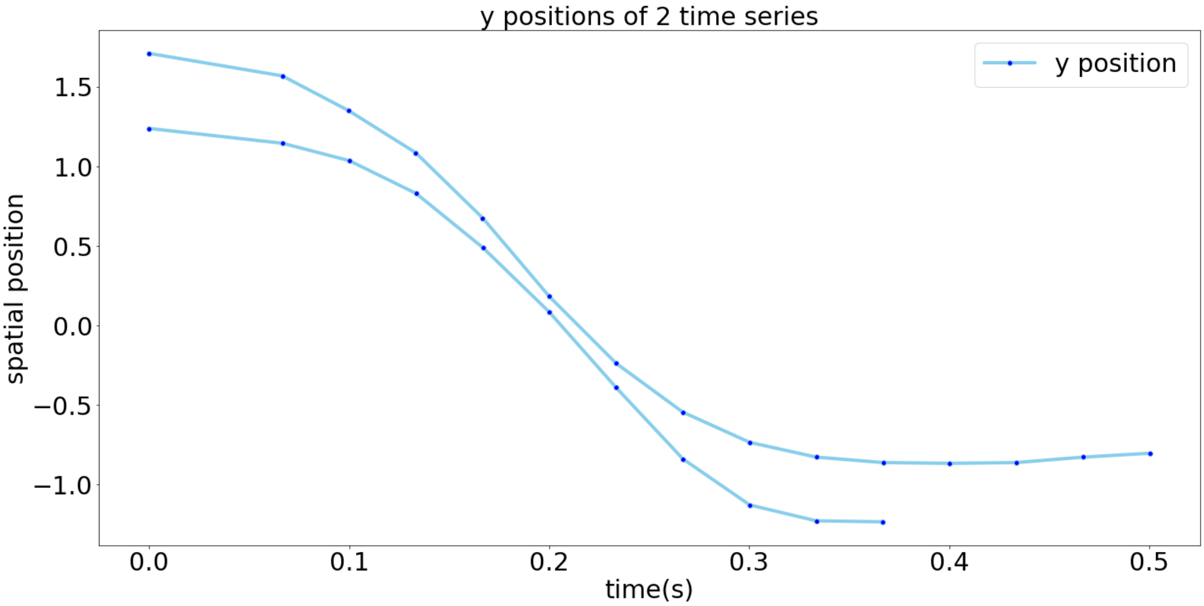


Figure 14: This graph illustrates the z-normalized version of the time series shown in figure x representing the vertical motion (y-axis) of a trajectory.

X and and Y time series Vectorization

The use of vector operations is common for the classification of multivariate time series as it performs the calculation of distances between sequences considering all the dimensions involved [16, p. 41]. Since each trajectory is composed by dependent spatial time series (both time series are captured in exactly the same moment assuring the inexistence of lags) when making the Dynamic Time Warping calculation between two trajectories we can not only make numerical comparisons (one-dimensional comparisons) but also vectorial comparisons (multidimensional comparisons) of the time series. Hence, we generate a vectorized version of both sequences combining them into a single multidimensional time series (2).

$$\vec{v_i} = \binom{x_i}{y_i}, \text{ with } i \geq 0 \leq |T| \tag{2}$$

3.2.4 Data Analysis

Understanding the Dynamic Time Warping approach

Regarding the similarity measurements among time series for pattern recognition, it is usually difficult to define a similarity function that allows time series similarity to be adequately described due to their non-linear nature since they are usually sequences in which the characteristics of mean, variance and covariance change over time. In order to calculate time series similarities, comparisons of the complete lengths or subsequences of each of the time series are involved. Based on previous studies of similar magnitude, this study addresses the non-linear time series similarity problem of the collected user touch data by means of a measurement technique known as Dynamic Time Warping (DTW).

The DTW algorithm

The DTW algorithm is a dynamic programming approach that is widely recognized when performing time series analysis. DTW was introduced in the 1960s on the basis of the Bellman Optimality Principle [17] and gained prominence in the 1970s when applied successfully in the area of voice recognition [18]. DTW makes it possible to determine the degree of similarity/dissimilarity of two time series by calculating an optimal non-linear alignment between them. Contrary to the Euclidean distance algorithm where only a linear comparison between two time series of the same length is possible, DTW achieves better results by not only being able to compare time series of different length but also by calculating the minimum distances across all sequences (4) in both time series and thus facilitating the recognition of similarities even when they do not appear at the same time.

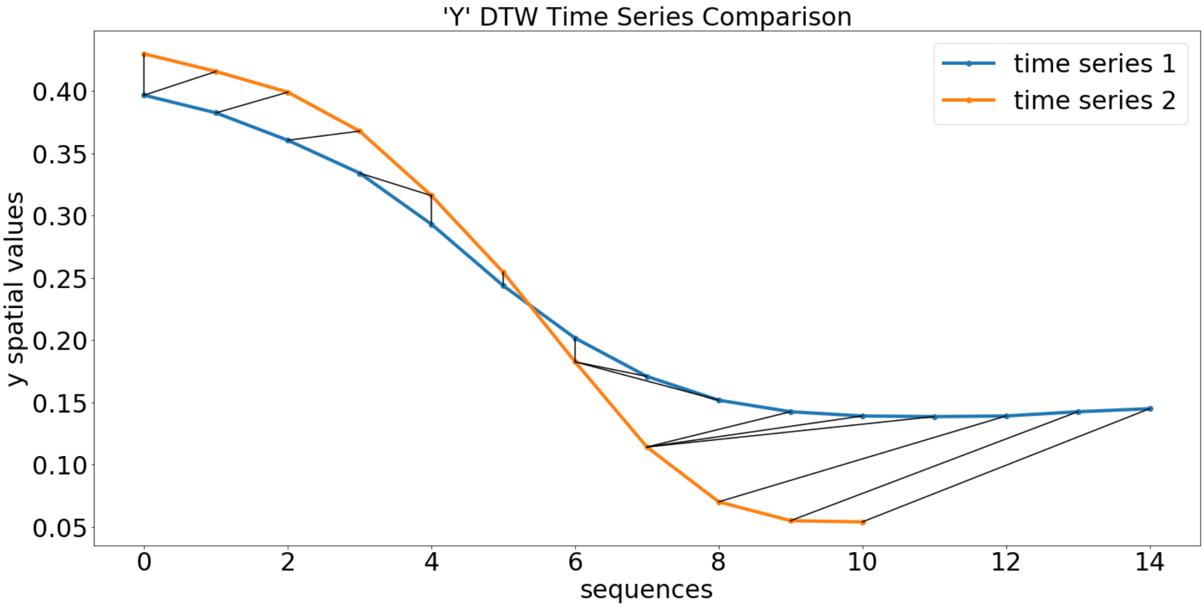


Figure 15: A time series comparison of two trajectories. The black lines correspond to the matched sequences that construct the distorted path (optimal path) which connects both time series.

Calculation of DTW

Given two time series $X = (x_1, x_2, ..., x_m)$ and $Y = (y_1, y_2, ..., y_n)$ of length m and n each. A distance cost matrix D of size $m\ x\ n$ is constructed where each element d_{ij} (with i representing the index of the time series X and j the index of the time series Y) expresses the distance (as seen in 3) between the elements x_{ij} and y_{ij} of the sequences X and Y.

$$D(x_i, y_j) = (x_i - y_j)^2 \tag{3}$$

The calculation of the minimum distance D_{ij} is given:

$$D(i,j) = Dist(i,j) + min\begin{cases} D(i-1,j), \\ D(i,j-1), \\ D(i-1,j-1) \end{cases} \tag{4}$$

Once the distances between all sequences are obtained, an optimal warping path that represents the best alignment (shortest distance) between both time series is determined. The warping path is defined as $W = (W_1, W_2, \ldots, W_k)$ with $max(m, n) \leq k \leq m + n - 1$. This path is continuous and each k^{th} element of W is defined $W_k = (i, j)$.

Restrictions of the warping path:

- The warping path starts at $W_k = (m, n)$ where k is the length of the warping path and ends at position $W_1 = (1,1)$ where the path starts and ends diagonally.
- Taking an element from the path $W_i = (a, b)$ then $W_{i-1} = (a', b')$ where $a \leq a'$ and $b \leq b'$. Therefore, the cells are adjacent.
- Taking an element from the path $W_i = (a, b)$ then $W_{k-1} = (a', b')$ where $a - a' \geq 0$ and $b - b' \geq 0$ forces the points of W to be continuously spaced.

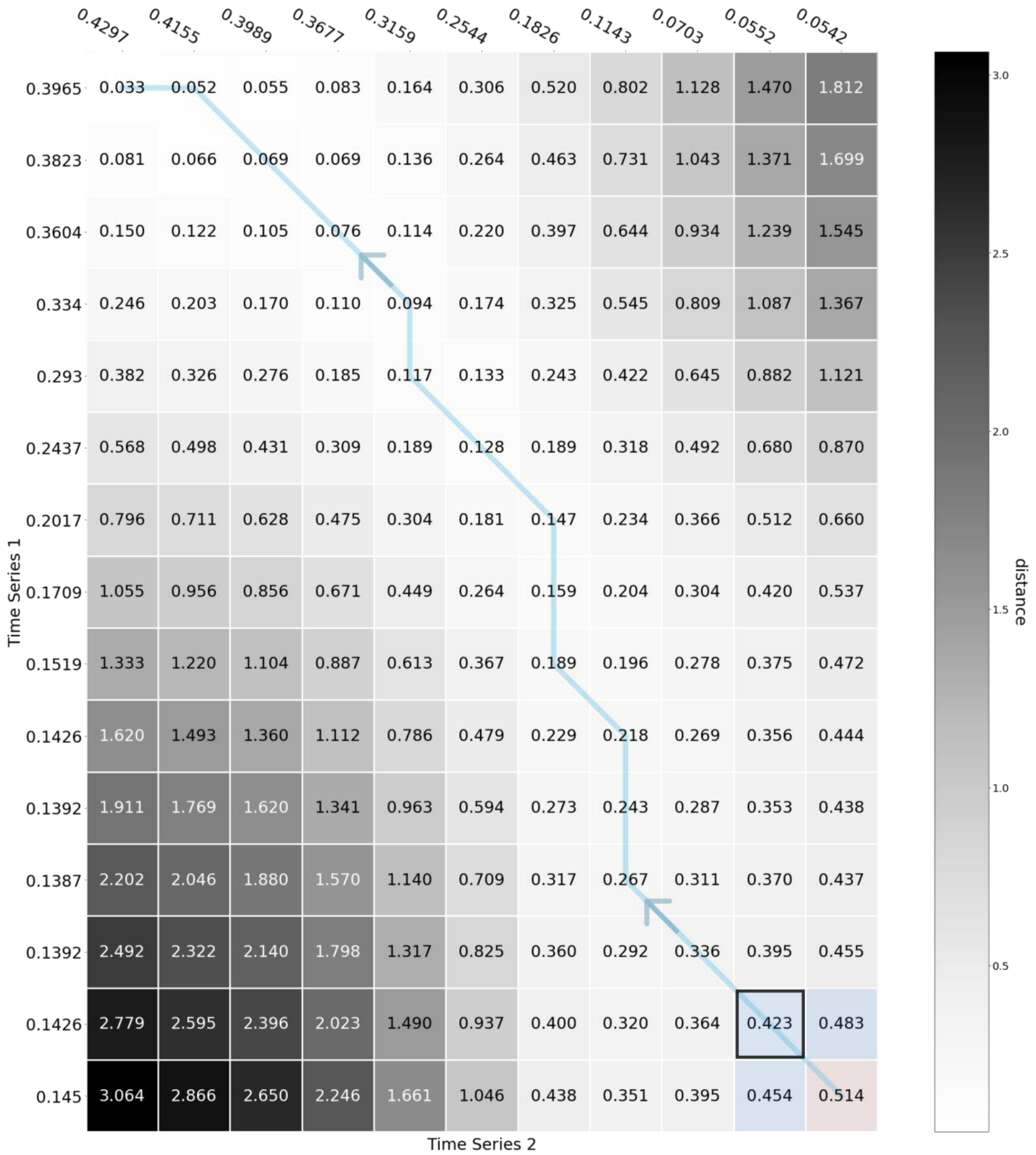


Figure 16: Illustration of the distance matrix $D = (m, n)$. Each cell indicates the distance between time series sequences. The distortion path W highlighted with the blue line is actually calculated in reverse order starting at $D = (m, n)$ and indicates the optimal distance between both time series. The three blue squares indicate comparisons of the neighbor distances with respect to the current position at $W_i = (i, j)$ (red square). The minimum distance (square with the black frame) represents the next sequence of the Warping path.

Extending DTW to Multidimensional Data

DTW is used for one-dimensional comparisons of time series, however, different extensions of the DTW algorithm allow multidimensional comparisons of time series. We decided to use the DTWD (Dynamic Time Warping Dependent) because this model has proven to be efficient in cases where the dimensions are perfectly aligned in time, i.e. there is no lag between them so that changes occur exactly at the same moment. The DTWD is obtained by almost following the same methodology as with DTW, but additionally the distances among the n-dimensions sequences from both trajectories are measured. We previously announced the vectorization for each of the X and Y time series sequences, this allows us to determine the distance calculation of sequences in form of a vector subtraction for both dimensions. The DTWD can be described as seen in (5).

$$DTW_D(T_1, T_2) = DTW\left(\begin{pmatrix} T_{1x} \\ T_{1y} \end{pmatrix}, \begin{pmatrix} T_{2x} \\ T_{2y} \end{pmatrix}\right) \tag{5}$$

Performing comparisons using the DTWD model

Using the DTWD algorithm, 104 trajectories belonging to the same group and created by 14 out of the 15 participants have been compared. With the aim of measuring the of similarity among them, the statistical measures of mean and standard deviation have been used in order to determine the dispersion degree of the resulting warping path distances for both the original and z-normalized time series values.

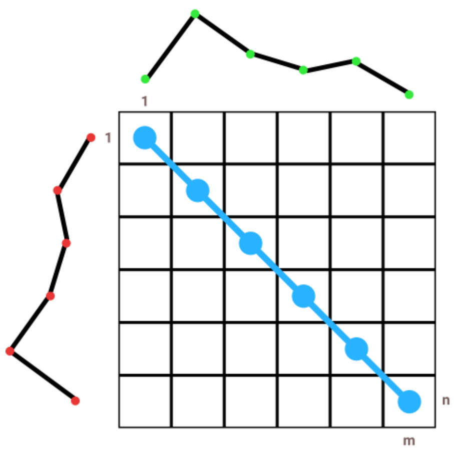


Figure 17: Optimal warping path between two equal trajectories

The similarity criteria for each comparison of trajectories is determined on the basis of the similarity level of the calculated distance of the warping path with respect to the optimum distance of the warping path, understanding such optimal path as a diagonal trace along the distance matrix whose value 0 would represent a perfect alignment between both trajectories. The analysis of patterns between trajectories has been undertaken in the following cases:

1. DTWD warping path distance calculations between trajectories created within a single user (the two players with the highest number of trajectories created in that group were compared individually)
2. DTWD warping path distance calculations between trajectories created by two users (comparisons of trajectories with the same user id excluded)
3. DTWD warping path distance calculations between trajectories created by all users of the group (comparisons of trajectories with the same user id excluded)

3.2.4 Results

The following two tables depict the mean and standard deviation results of the calculated warping path distances using the DTWD algorithm. It is important to highlight the significance of having included DTWD calculations for the z-normalized time series since the original position values of the sequences are always moving in a range close to 0 leading to misunderstandings with the metric of the optimal warping path distance that manifests a greater degree of similarity (correlation) between the comparative trajectories as the cost of the distortion path approaches zero, for that reason the z-normalized values can give us a clearer view of the degree of similarity/dissimilarity of the compared trajectories rather than the original values. The dispersion degree of the resulted calculations can be also inferenced by looking at the corresponding figures (xx,yy,zz) for each of the described cases.

Results for the original X and Y time series values				
Case	Number of trajectories	Number of DTWD comparisons	Mean	Standard deviation (error rate)
1.1	12	66	0.7813	0.7107
1.2	12	66	1.0813	0.9054
2	24	144	0.9309	0.7656
3	104	4917	0.8981	0.8745

Table 1: Results of the statistical analyses for the different cases resulting from the DTWD comparisons of the values of the time series belonging to each trajectory.

Results for the z-normalized X and Y time series values				
Case	Number of trajectories	Number of DTWD comparisons	Mean	Standard deviation (error rate)
1.1	12	66	11.7433	6.8898
1.2	12	66	14.1722	7.6096
2	24	144	12.6968	7.2141
3	104	4917	13.6802	7.7732

Table 2: Results of the statistical analyses for the different cases resulting from the DTWD comparisons of the z-normalized values of the time series belonging to each trajectory

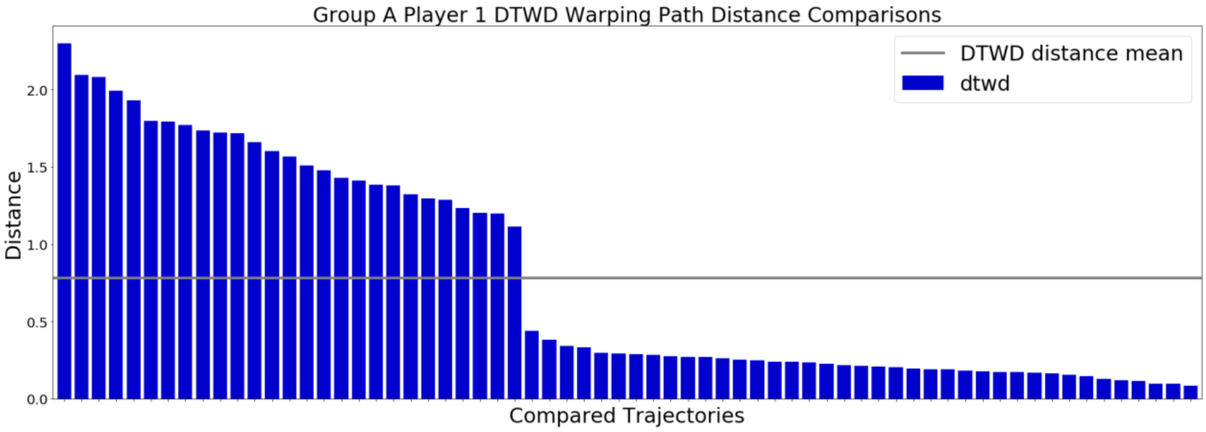


Figure 18

Figure 18 displays the calculated distances of the DTWD distortion path resulting from comparisons between the time series values (X,Y) of the trajectories belonging to the same group and which were generated by the same user.

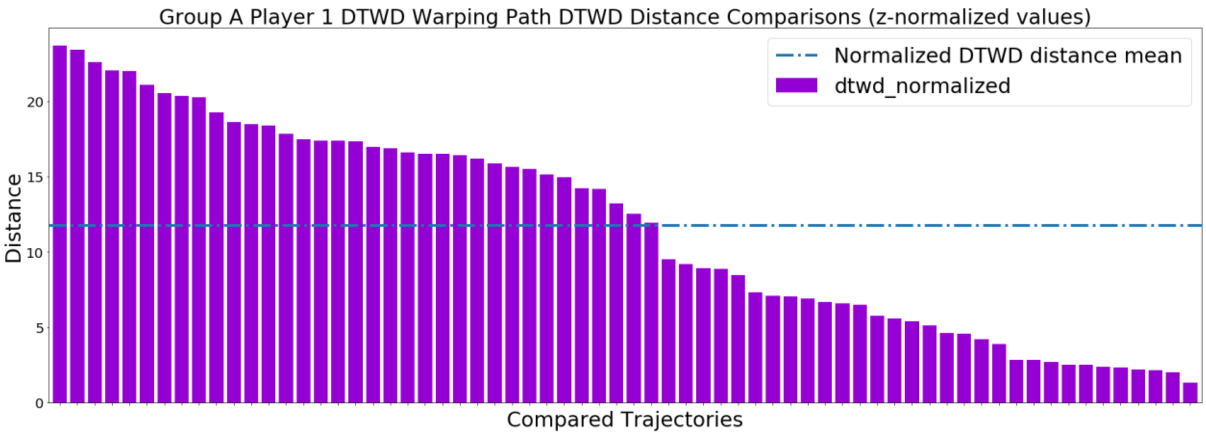


Figure 19

Figure 19 displays the calculated distances of the DTWD distortion path resulting from comparisons between the z-normalized time series values (X,Y) of the trajecto-ries belonging to the same group and which were generated by the same user.

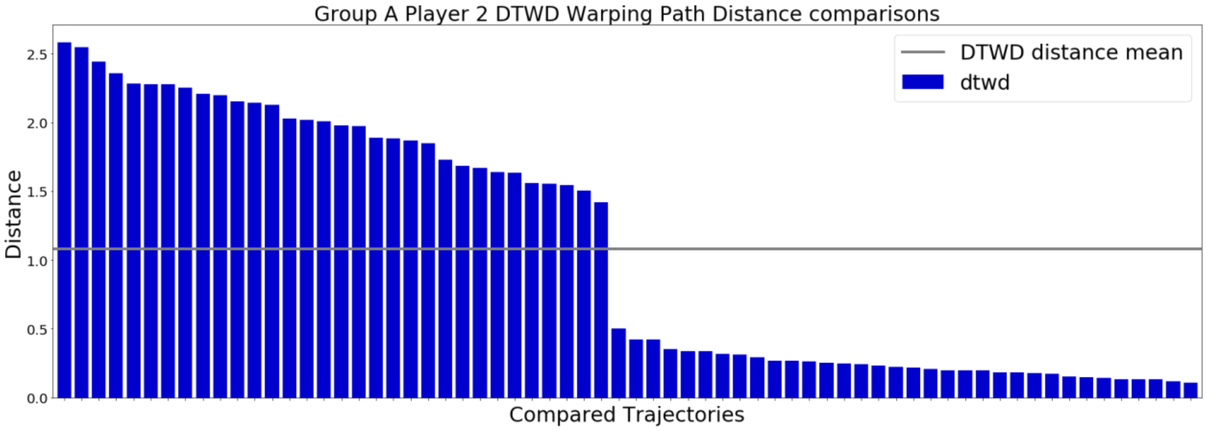


Figure 20

Figure 20 displays the calculated distances of the DTWD distortion path resulting from comparisons between the time series values (X,Y) of the trajectories belonging to the same group and which were generated by the same user.

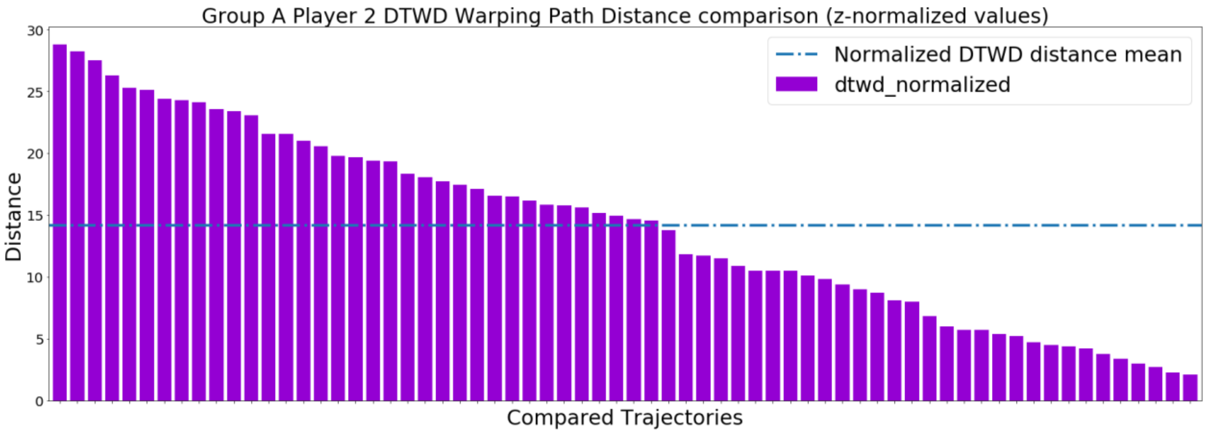


Figure 21

Figure 21 displays the calculated distances of the DTWD distortion path resulting from comparisons between the z-normalized time series values (X,Y) of the trajectories belonging to the same group and which were generated by the same user.

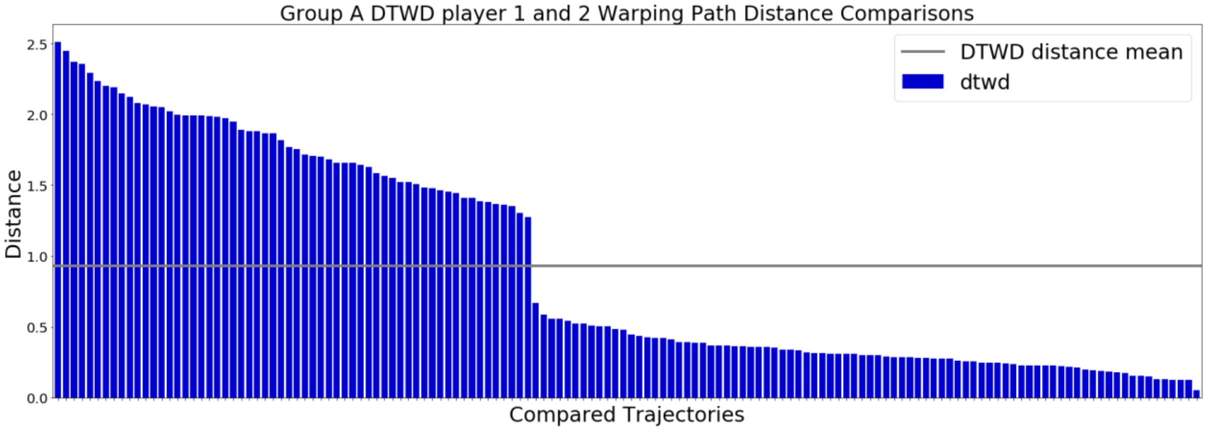


Figure 22

Figure 22 displays the calculated distances of the DTWD distortion path resulting from comparisons between the time series values (X,Y) of the trajectories belonging to the same group and which were generated by two users (trajectories belonging to the same user id excluded).

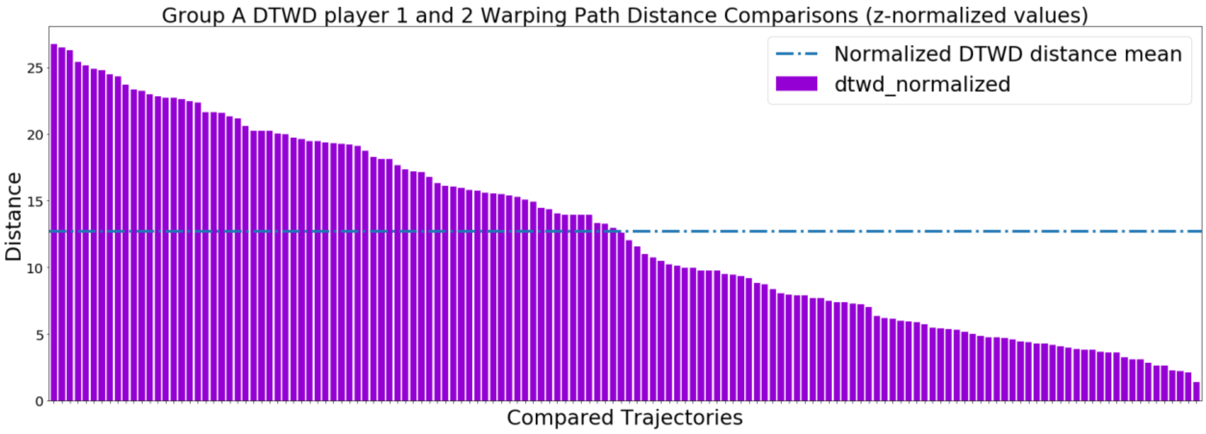


Figure 23

Figure 23 displays the calculated distances of the DTWD distortion path resulting from comparisons between the z-normalized time series values (X,Y) of the trajecto-ries belonging to the same group and which were generated by two users (trajectories be-longing tot he same user id excluded)

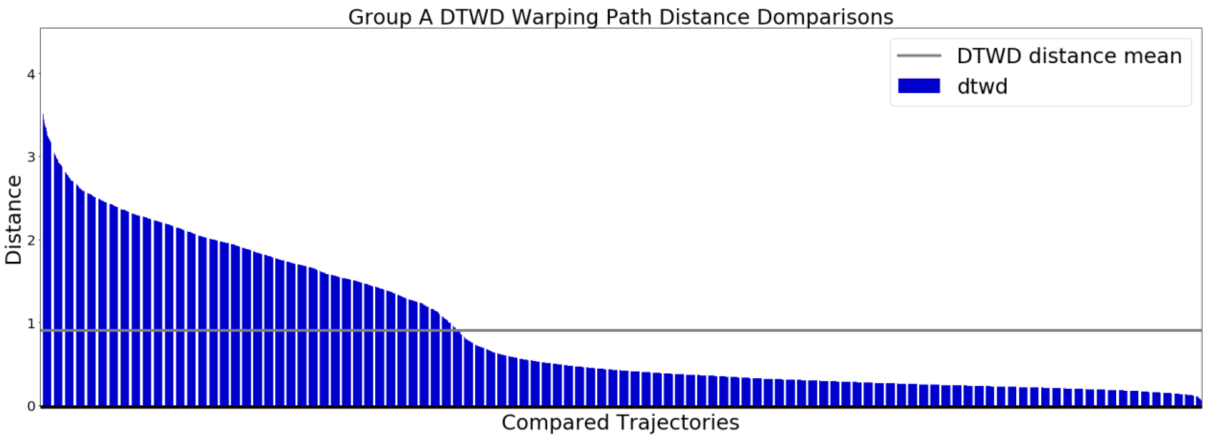


Figure 24

Figure 24 displays the calculated distances of the DTWD distortion path resulting from comparisons between the time series values (X,Y) of the trajectories belonging to the same group and which were generated by all users of the group (trajectories belonging tot he same user id excluded).

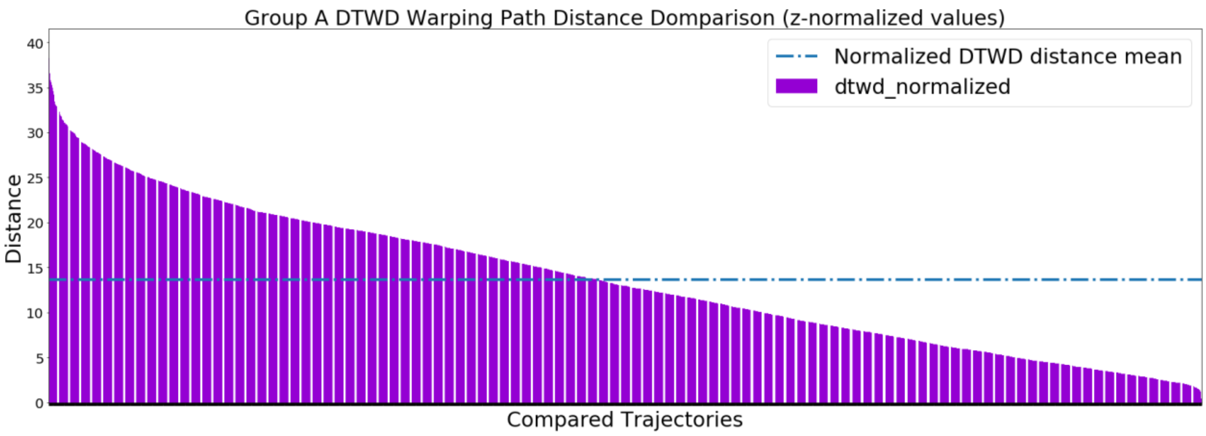


Figure 25

Figure 25 displays the calculated distances of the DTWD distortion path resulting from comparisons between the z-normalized time series values (X,Y) of the trajecto-ries belonging to the same group and which were generated by two users (trajectories be-longing to the same user id excluded).

In all sample values we find low data correlation. The measurement of z-normalization has been accurate because the values of the original time series have a diminute size (very close to 0) making it impossible to make a closer approximation to reality. By looking at the z-normalized DTWD warping path distance values as well as the statis-tical data of mean and standard deviation we can affirm that there is little similarity between the compared trajectories. As seen in table (1 and 2) It is also surprising that there is a greater correlation (degree of similitude) between the corresponding trajectories performed by all fourteen players (case 4) in contrast to the comparison of those trajectories created by the same player (case 2).

3.2.5 Discussion of the findings

Using the warping path distance comparison approach seems to be insufficient to determine the existence of similarities between the time series generated by the players of ShapeMatch. Several explanations may help to understand these findings, and possible countermeasures in extension to the present work will therefore be presented.

Runtime of DTW calculations

One of the major limitations to cover a greater number of comparisons of the data set has to do with the complexity of the DTW runtime. The performance of the primitive algorithm is O(mxn) with m corresponding to the length of time series 1 and n to the length of time series 2 which makes it very cost-intensive in memory for large-sized sequences and for large time series data calculations. There are extensions that significantly increase the DTW performance through lower bounding techniques. The best known extensions are the ones of Sakoe Chiba and the Itakura paralelogram which limit the distance matrix cells to a certain width across its diagonal reducing unnecessary calculations of non-optimal distances between time sequences, however, its use is strictly linked to time series of the same length, in case the time series to be compared are not of the same length one of the two must be reinterpolated. Extensions such as FTW and FastDTW [19] rely on such dimensioning models proving to be more efficient than the primitive DTW calculation in terms of runtime complexity.

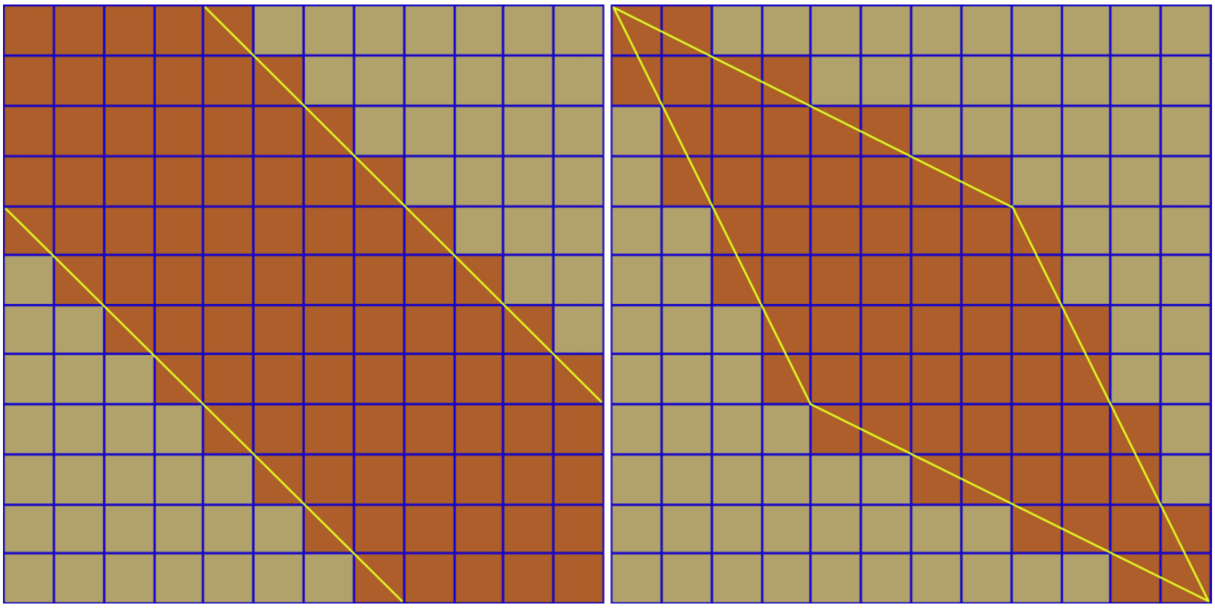


Figure 26: Illustrations of lower bounding as extensions to improve the calculation of the primitive DTW. The figure on the left is the diagonal band model proposed by Sakoe Chiba and the figure on the right is the parallelogram model proposed by Itakura.

Time series segmentation applied to classification algorithms

It is possible that the comparison of trajectories within one player does not show much similarity because the duration/length of one trajectory is dependent on the amount of intended matches and the location of the matching figure(s). ShapeMatch does not artificially limit the players' movements across the screen by imposing specific rules, e.g. that after every match or attempted match of two figures the trajectory needs to be interrupted by lifting the finger off the screen. That is, several strategies may coexist, both within and between players. In one scenario, a player might lift his finger off the screen after an attempt at matching two figures. This may be the case if in the end position of the match, he is blocking his own view on the screen with his arm. This strategy will generate a rather short trajectory. In other cases, he may well directly continue the trajectory to match another pair of figures without lifting the finger off the screen between two sets of matches. This will create a much longer trajectory and changing direction of the screen touch pattern. In this case, the trajectory/time series might best be broken down into two or more segments that each correspond to one attempt of matching.

Guijo-Rubio and colleagues [20] have introduced a novel proposal of time series classification based on the characterization of their segment topologies. The aim of the approach found in this work is to trim the time series into specific segments in order to recognize correlations between the shapes of each segment opening a different perspective in the time series clustering models. A hierarchical clustering based on an agglomerative or divisive algorithm (by linking a pair of clusters with a higher correlation and viceversa) could also be derived not only through the shape of the segment but also from its seasonality, trend, chaos, proper similarity, nonlinearity, etc.

Classification algorithms to support descriptive analysis

In this project, due to unexpected limitations in manpower, the analyses have been limited to descriptive analyses of similarity of screen touch patterns. Descriptive analyses can be extended with classification algorithms such as 1NN or K-nearest neigborhood for analysis. These classification algorithms might detect similarities that are not so pertinent in visual inspection of the data and descriptive comparisons. They also allow for statistical testing of the significance of similarity or dissimilarity. Another possibility to extend descriptive inspection of the DTW coefficients might be to compare the obtained values with cut-off values that indicate high, moderate, or low similarity.

Further features with potential significance for screen touch pattern analysis

Some characteristics such as speed, acceleration, displacement, distance, or direction are criteria that can become significantly relevant when finding similarities between trajectories. These features must be regarded not only for the overall trajectory but also within its different time intervals. Complementary to these analyses, clustering algorithms could be fed with such criteria.

4. Conclusion

The aim of this P-5 project was to outline and execute the steps necessary to evaluate whether user recognition based on screen touch patterns generated by players of a mobile game was feasible in a specifically designed use context. The process was divided into two main phases, the development of the game ShapeMatch whose players would provide screen touch pattern data to be analyzed regarding similarity within and between users, and the data processing and analysis phase in which the generated screen touch pattern data were preprocessed, analyzed, and conclusions were drawn. Dynamic Time Warping was used to compare the trajectories generated by the two users with most trajectories stored.

Within-player similarity of the trajectory time series was very low for both players. This may be attributable to several facts. Firstly, as the rules of ShapeMatch do not direct players' trajectories on the screen other than illiciting them to pair matching figures, the players have great freedom to apply different strategies depending on personal preferences and situational demands. This leads to high variability of the compared trajectories regarding length, direction and direction changes. One possible countermeasure is time series segmentation with the aim of splitting the trajectory time series up into meaningful segments that will subsequently be analyzed as separate time series.

References

[1] L. Rühli and D. Hauri, “Alterung: Global Trend,” Avenir Suisse, Zürich, 2017.

[2] “FHNW Myosotis-Garden: Gamen mit betagten Menschen,” *FHNW*. [Online]. Available: https://www.fhnw.ch/de/die-fhnw/hochschulen/ht/institute/institut-fuer-data-science/fhnw-myosotis-garden. [Accessed: 15-Aug-2019].

[3] A. I. Nordin, P. Cairns, M. Hudson, A. Alonso, and E. H. Calvillo Gámez, “The Effect Of Surroundings On Gaming Experience,” in *Proceedings of the 9th International Conference on the Foundations of Digital Games*, 2014.

[4] M. Erfani *et al.*, “The Effect of Age, Gender, and Previous Gaming Experience on Game Play Performance,” in *Human-Computer Interaction*, Berlin, 2010, pp. 293–296.

[5] S. M. Kolly, R. Wattenhofer, and S. Welten, “A Personal Touch: Recognizing Users Based on Touch Screen Behavior,” in *Proceedings of the Third International Workshop on Sensing Applications on Mobile Phones*, New York, NY, USA, 2012, pp. 1:1–1:5.

[6] M. Antal, Z. Bokor, and L. Z. Szabó, “Information revealed from scrolling interactions on mobile devices,” *Pattern Recognit. Lett.*, vol. 56, pp. 7–13, Apr. 2015.

[7] T. Feng, J. Yang, Z. Yan, E. M. Tapia, and W. Shi, “TIPS: context-aware implicit user identification using touch screen in uncontrolled environments,” in *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications - HotMobile ’14*, Santa Barbara, California, 2014, pp. 1–6.

[8] L. Shamir, J. D. Delaney, N. Orlov, D. M. Eckley, and I. G. Goldberg, “Pattern Recognition Software and Techniques for Biological Image Analysis,” *PLoS Comput. Biol.*, vol. 6, no. 11, p. e1000974, Nov. 2010.

[9] P. Dimitrijevic, “Einflussfaktoren auf die physiologische Synchronität zweier Interaktionspartner,” Universität Zürich, Zürich, unveröffentlichte Masterarbeit, 2019.

[10] S. Saha, “A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way,” *Medium*, 17-Dec-2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. [Accessed: 15-Aug-2019].

[11] J. Lin, S. Williamson, K. Borne, and A. Debarr, “Pattern Recognition in Time Series,” in *Advances in Machine Learning and Data Mining for Astronomy*, vol. 20124949, A. Srivastava, Ed. Chapman and Hall/CRC, 2012.

[12] D. J. Berndt and J. Clifford, “Using Dynamic Time Warping to Find Patterns in Time Series,” *AAAI Workshop Knowl. Discov. Databases*, pp. 229–248, 1994.

[13] V. Di Gesù and V. Starovoitov, "Distance-based functions for image comparison," *Pattern Recognit. Lett.*, vol. 20, no. 2, pp. 207–214, Feb. 1999.

[14] O. Michel, "Holger Kantz and Thomas Schreiber, Nonlinear Time Series Analysis," *J. Biol. Phys.*, vol. 24, no. 1, pp. 79–80, Mar. 1998.

[15] J. Tarango, E. Keogh, and P. Brisk, "Accelerating the dynamic time warping distance measure using logarithmetic arithmetic," in *2014 48th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, 2014, pp. 404–408.

[16] D. Smirnov and E. M. Nguifo, "Time Series Classification with Recurrent Neural Networks," p. 80.

[17] T. K. Vintsyuk, "Speech discrimination by dynamic programming," *Cybernetics*, vol. 4, no. 1, pp. 52–57, 1972.

[18] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 26, no. 1, pp. 43–49, Feb. 1978.

[19] M. Lee, S. Lee, M. Choi, Y. Moon, and H. Lim, "HybridFTW: Hybrid Computation of Dynamic Time Warping Distances," *IEEE Access*, vol. 6, pp. 2085–2096, 2018.

[20] D. Guijo-Rubio, A. M. Durán-Rosal, P. A. Gutiérrez, A. Troncoso, and C. Hervás-Martínez, "Time series clustering based on the characterisation of segment typologies," *ArXiv181011624 Cs Stat*, Oct. 2018.