Decision Support and Business Intelligence

Information Technologies for Business Intelligence

Master Thesis

Oky Purwantiningsih

Visual Analytics on Human Body Movement Data Applied on Healthcare

---

prepared at LIRMM and Université Paul Valéry Montpellier

Submitted on September 11, 2015

Advisor :	Arnaud Sallaberry	-	LIRMM	arnaud.sallaberry@lirmm.fr
	Jerômé Azé	-	LIRMM	jerome.aze@lirmm.fr
Supervisor :	Nacéra Bennacer	-	Centrale Supélec	nacera.bennacer@supelec.fr
In collaboration with :	Sébastien Andary	-	NaturalPad	sebastien@naturalpad.fr

Abstract: Serious game is used in healthcare to perform patients' rehabilitation and training. An example of such game is Hammer and Planks which is used to rehabilitate patient with balance disorders. Utilizing motion sensing input devices (Kinect, Wii Balance Board), the game is played by moving player's body. However, it is difficult to assess the rehabilitation progress from the game. In this Master thesis we design and develop visualization interface to help healthcare professional make correct diagnostic of patients' progress and therefore enable them to set the difficulty level for future rehabilitation session.

To achieve this objective, we developed a visualization application which enables healthcare professional to analyse gameplay from two different views: (i) Session Visualization which allows analysis on one session. With this view, users are able to identify the frequency of movement related to objects on the game (ii) Summary Visualization which allows movement analysis over several sessions. This view enables users to navigate and explore the evolution of movement throughout all of the sessions. Here we propose a clustering method based on hierarchical clustering to group similar movement pattern over sections of game horizontal x-axis. This view also enables users to analyse in which area of x-axis the movement frequently happened. The proposed visualization is illustrated with two case studies which demonstrate the ability of the application to assess the rehabilitation progress.

Keywords: Information Visualization, Visual Analytic, Body Movement Visualization, Movement Analysis, Hierarchical Clustering

Acknowledgments

In the name of Allah, the Most Gracious and the Most Merciful.

Alhamdulillah, all praises to Allah for the strengths and His blessings in completing this thesis.

Firstly, I would like to express my sincere gratitude to my advisors Arnaud Sallaberry and Jerômé Azé for their valuable guidance, detailed feedbacks, and continuous support. Without their help, I wouldn't be able to finish the research and complete the writing of this master thesis. Especially for Arnaud Salaberry, his patience and attention has made the development of the visualization application possible. I would also like to thank my supervisor Nacéra Bennacer for her insightful comments and feedbacks.

I am also thankful to Sébastien Andary of NaturalPad for his help regarding the Hammer and Planks game as well as providing the game application and data for the research. His valuable support has made the completion of this research possible.

My sincere thanks also goes to the Advanse Team in LIRMM, who gave me the opportunity to join their team as an intern and who provided access to their research facilities. Most importantly, for their understanding and support when I had medical difficulties during my internship. My immense gratitude towards my colleague for their time and participation in providing the gameplay data set.

My gratitude goes to the IT4BI Consortium for giving me the opportunity and making all of this possible. My gratitude also goes to Université Paul Valéry Montpellier for funding the research.

I thank my friends for their company, support and motivation during the past six months. Thank you for the friendship and memories. I would also like to thank my parents for their continuous support and prayers.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advancement of gaming device technology such as Kinect, Wii Balance Board, Wii Remote, PlayStation Move, etc. has enabled players to control and interact with the game console through body movement. In healthcare, such technologies are used in serious game which can help users (doctors, patients, researchers, etc.) perform health related activity such as patients' rehabilitation and training[22, 6, 14]. One example of such game is Hammer and Planks. Developed by NaturalPad[1], Hammer and Planks was designed to train the equilibrium of patient with balance disorders (specifically for hemiplegic people)[9]. A person with hemiplegic is paralized on one side of the body[2]. Therefore, the gameplay is designed so that the player has to move their body to the right, left, forward and backward in order to train their affected side of the body. To support the purpose of rehabilitation, the healthcare professionals need to analyse the movement to make a correct diagnostic of patients' progress and to adjust the difficulty level for the next rehabilitation session. In this thesis, we discuss the design of an interface to help healthcare professional understand the data generated from the game.

1.1 Motivation

Hammer and Planks is a vertical shooter game. The game world is in a 2D environment vertically scrolling from top to bottom in which a player navigate a ship from left to right through moving his body. It tells the story of a pirate named John K. One day a meteor fell down on John's ship and ruin it. There is a little left from his boat but it is still enough to build a new basic boat with what's left. While navigating his ship to collect driftwood/plank to upgrade it (hence the name Hammer and Planks), he also wants to find the ship which showered meteor and destroyed his ship. Therefore, in the game a player has to defeat all enemies which come on his way and he has to avoid being destroyed by bullets, reefs and other obstacles. Throughout the game the player has to collect bonuses (planks) to improve the ship. The game is usually played in short and intense phase and thus requires a lot of concentration[9]. Figure 1.1 shows the interface of the game. The boat in the middle represents the player, the reefs are obstacles which should be avoided, driftwood/planks shown in yellow circle are bonuses which should be collected, and the object emanating fires is an enemy which should be killed.

Currently, the game provides some charts which visualize player's body movement with respect to the horizontal axis and vertical axis (Figure 1.2). However, the information that can be gathered from the visualization is not enough for the healthcare professional to be able to establish an informed diagnostic. It's hard to know how often the player

[1] www.naturalpad.fr

[2] http://www.hemihelp.org.uk/hemiplegia/what_is_hemiplegia

move to the right or left. It's also not possible to know to which type of events (ie. avoiding an enemy, catching the bonuses) the movement is related to. Which is crucial since the therapist need to know if the player is able to develop strategy to play the game overtime. The existing visualization also provides chart to show the evolution of player's performance and total movement for all game sessions. However, the evolution of player's body movement is not depicted.

Figure 1.1: Hammer and Planks Screenshot

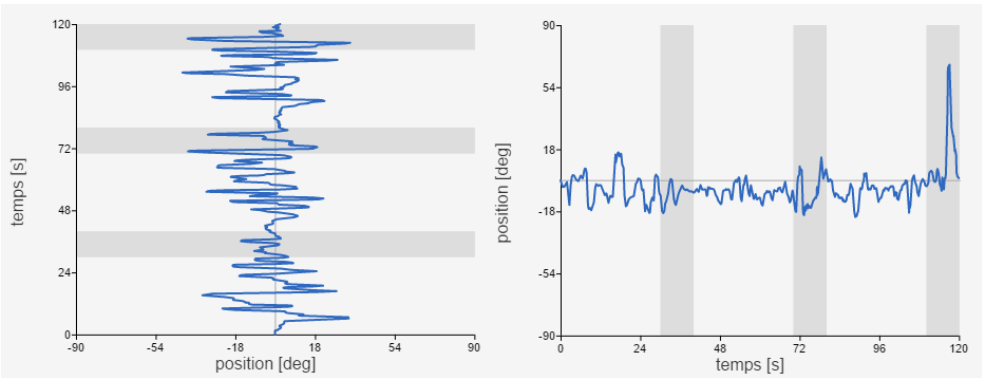


Figure 1.2: Current visualization used in the game to show movements over horizontal and vertical axis

The purpose of this thesis is to address the problem mentioned by proposing a visualization interface to help healthcare professionals analyse gameplay data generated from the game so that they can determine the frequency and direction of player movement as well as movement pattern evolution over time. It is important for the interface to be simple and intuitive but at the same time still able to correctly shows the information needed by healthcare professionals. In this case, the visualization design process is difficult due to the heterogeneity and the size of the data. For instance, Hammer and Planks produces log file with tens of thousands records. Another challenge is how to visualize movement

data that can be interpreted intuitively. Currently, there have been several approaches proposed on how to perform visual analytics on movement data. However, most of them deal with "geographical" movements [2] and only a few has been dealing with human body movement [4]. It is also challenging to identify interesting movement pattern from the data as well as choosing the right approach to visualize changes of pattern.

1.2 Methodology

To ensure that the visualization to be designed would satisfy the information needed by healthcare professionals, we followed the Nested Process Model proposed by Tamara Munzner [19]. The model is divided into 4 levels: Domain Problem Characterization, Data/Operation Abstraction Design, Encoding/Interaction Technique Design, and Algorithm Design. These levels are nested; the output of a higher level will be the input for the lower level as shown in Figure 1.3.

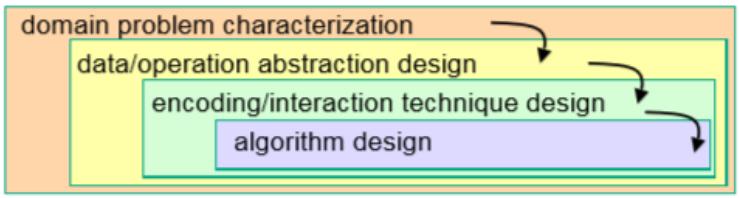


Figure 1.3: Munzner's Visualization Design Model with four nested layers

In domain problem characterization level, the types of information needed by health professional from the visualization are defined. The output of this level would be a list of tasks that need to be solved by the visualization application. We then identify data structures which can support these tasks in the Data/Operation Abstraction Design level. In the third level, a good visualization and interaction techniques which can support the tasks are defined. At last in Algorithm Design level, an algorithm to support the visualization is proposed.

1.3 Contribution

In this thesis, we proposed a visualization interface for healthcare professionals to visualize Hammer and Planks gameplay data to help them analyse player movement during the game as part of rehabilitation process. The visualization provides two type of visualizations:

- A visualization which provides information on frequency and direction of body movements which are related to objects in the game for one game session.
- An interactive visualization where healthcare professionals can analyse the evolution of player's body movement throughout all sessions. Here, we proposed an approach to easily navigate and highlight movement pattern. We also proposed a clustering algorithm based on hierarchical clustering to identify similar movement pattern. A

distance function is defined to quantify movement pattern similarity which consider both movement evolution and proportion of movement frequency.

1.4 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 discuss the domain problem characterization. Chapter 3 explores related work. The data abstraction is presented in Chapter 4. The Visual Mapping and Interactive Functionality of the proposed visualization are discussed in details in Chapter 5. Chapter 6 provides some case studies used to evaluate the approach and finally, chapter 7 concludes the thesis.

Chapter 2

Domain Problem Characterization

In order to clearly understand the problems faced by the healthcare professional in interpreting the gameplay into a meaningful therapy routine, it is important to have an understanding of how the game is played. Based on this understanding, then it will be possible to find out what kind of information needed by defining questions usually asked by the users. In the end, visualization requirement elicitation will be done by translating each question into list of tasks. This chapter discusses each one of these steps in details.

2.1 Hammer and Planks Game Dynamics

Played with Kinect, it is possible to play Hammer and Planks in three different ways:

1. BodyTilt: Player puts both arms in his/her hips and move the upper body (from the waist up) to the right, left, forward or backward to navigate the boat(Figure 2.1 (left)).

2. HandPoint: Player lifts one of his/her forearm in front of the body with the palm facing forward. Navigating the boat can be done by moving the forearm to the right, left, forward, or backward(Figure 2.1 (right)).

3. ShoulderCGE: Player lifts one of his/her arm in front of the body and bend the elbow. Moving the elbow up and down will navigate the boat up and down the screen.

For both the BodyTilt and HandPoint there are three directions available: (i) Horizontal: the screen scrolls from top to bottom and player navigates the ship from left to right (ii) Vertical: the screen scrolls from right to left and player navigates the ship from top to bottom of the screen (iii) Both: the screen scrolls from top to bottom and player navigates the ship from left to right. He/she can also move the boat faster or slower by bending the upper body (BodyTilt) or arm (HandPoint) forward or backward . For ShoulderCGE there is only vertical direction. In this thesis, we only interested in games played using BodyTilt and HandPoint movement for both directions since the information generated are richer, thus harder for the specialist to understand.

Before each session, the healthcare professionals will set the number of objects (enemies, bonuses, obstacles), activity duration and repetition, as well as area in which the objects can appear. Therefore he can adjust the difficulty of the game for different session.

2.2 Target User Questions

A traditional Hemiplegic therapy routine usually involved the therapist ordering a patient to perform several movement repetitively [22]. By the end of the session, the therapist

(a) BodyTilt (b) HandPoint

Figure 2.1: Hammer and Planks movement type

will analyse how the patient has performed based on the quality of movement as well as how the patient has progressed compared to the previous session. Based on this analysis, the therapist will then configure a new routine to further the patient's progress, if needed.

However, by using a game to facilitate the therapy, it is difficult to monitor how often the patient has moved his/her arm, to which direction and to which objects this movement is associated. Based on these reasons, we identified 5 types of questions usually inquired:

(Q_1) For a given session, to which direction (right/left) the player moved more?

(Q_2) For a given session, how does the player perform based on the number of objects collected, avoided, or killed with respect to the area of the movement?

(Q_3) For a given session, how does the player perform based on the number of objects collected, avoided, or killed with respect to the area of movement and the speed in which the game is played?

(Q_4) For a given patient, has he/she improved in the game overtime?

(Q_5) For a given patient, has he/she improved in a certain area overtime?

2.3 Visualization Requirements

The gameplay of each game session is logged in a json file which contains information of the player, game setting, player movement throughout the game and every events (i.e. enemy killed, bonus collected, etc.) happened in the game. Each event has an event type, which is the type of action performed by a player (represented with a boat in the

game) towards an object or the other way around. In the log file it is recorded as Kill, Catch, Dodge, etc. From the log, it is also possible to calculate how fast the player's boat move within the game world when an event happened. For future reference, the pace boat movement will be called screen speed. Based on these information and the question defined in the previous section, the visualization tasks can be grouped into: task related to a session for a particular player ($T1$) and task related to the summary of all sessions for a player ($T2$). The following are the tasks defined for each task group:

($T1.1$) Visualize and be able to compare the number of events within the same or among different event type at a given x area $(Q_1)(Q_2)$.

($T1.2$) Visualize and be able to compare the number of events and its screen speed of the same or among different event type at a given x area $(Q_1)(Q_2)(Q_3)$.

($T1.3$) Select and visualize the number of events for a certain object at a given x area $(Q_1)(Q_2)$.

($T1.4$) Select and visualize the number of events and its screen speed for a certain object at a given x area $(Q_1)(Q_2)(Q_3)$.

($T2.1$) Visualize, navigate and be able to compare the evolution of number of events throughout all sessions within a certain x area.$(Q_4)(Q_5)$.

($T2.2$) Select and visualize the number of events of a certain event type in a certain x area throughout all sessions $(Q_4)(Q_5)$.

($T2.3$) Visualize, navigate and be able to compare the distribution of a certain number of events over x area among all sessions$(Q_4)(Q_5)$.

($T2.4$) Select and visualize the distribution of certain number of events over x area for a certain event type throughout all sessions $(Q_4)(Q_5)$.

($T2.5$) Extract and visualize similar pattern of number of events evolution throughout all sessions over a certain x area $(Q_4)(Q_5)$.

Chapter 3

Related Works

There have been several serious game for hemiplegic rehabilitation developed in the last few years. Similar to Hammer and Planks, these games also have some visualization feature which shows how the player performed so that the therapist is able to make the correct diagnosis. Thus, in this chapter, we first review some of these visualization. Then, since the nature of the input data is time series and movement data, we present some work in visualization which are related to this type of data.

3.1 Visualization of Serious Game Result

Game result visualization is an integral part of a serious game used for rehabilitation since it's the feature which influence the accuracy of therapist analysis. Most serious game have an analytic feature, however the type of analysis presented depends on the nature of the game and the framework used in the rehabilitation. Therefore, for the purpose of this thesis, we only focus on reviewing serious game which are directed to hemiplegic patients rehabilitation.

[22] present a rehabilitation framework for hemiplegic patients which combines the use of Kinect and LEAP[1] hand-tracking devices. These devices are attached to a 3D based game environment which was set to accommodate a set of primitive therapy motion such as forearm pronation/supination, shoulder and hip joint adduction/abduction, etc. Similar to Hammer and Planks, one of the game used in the framework requires user to navigate a plane by moving the hand to the right and left (hand-elbow flexion-extension). The recorded movement is then presented in line chart depicting the range of axis of elbow joint (180 degrees when fully extended and 20 degrees when fully flexed) over number of frames captured (Figure 3.1). Similarly, current visualization in Hammer and Planks also uses line chart to show average body movement over time. At first, line chart is used to represent Hammer and Planks gameplay, however in the end this approach is abandon since it's not intuitive enough. Details of this attempt can be found in chapter 4.

In [14], a virtual reality rehabilitation system for children with hemiplegia was developed using TUI[2]. The game itself is displayed on LCD and the player interact with the game by placing the TUI on top of moving targets shown on the LCD. In this system, performances are measured by speed, accuracy and trajectory(mean movement efficiency). However, unlike [22], this system doesn't provide an interface in which therapist can analyse the gameplay.

Similar to Hammer and Planks, [23] introduced a framework which uses Kinect attached to Second Life[3] serious game environment. The mission of the game is to follow

[1] https://www.leapmotion.com/product/desktop
[2] https://en.wikipedia.org/wiki/Tangible_user_interface
[3] http://secondlife.com/

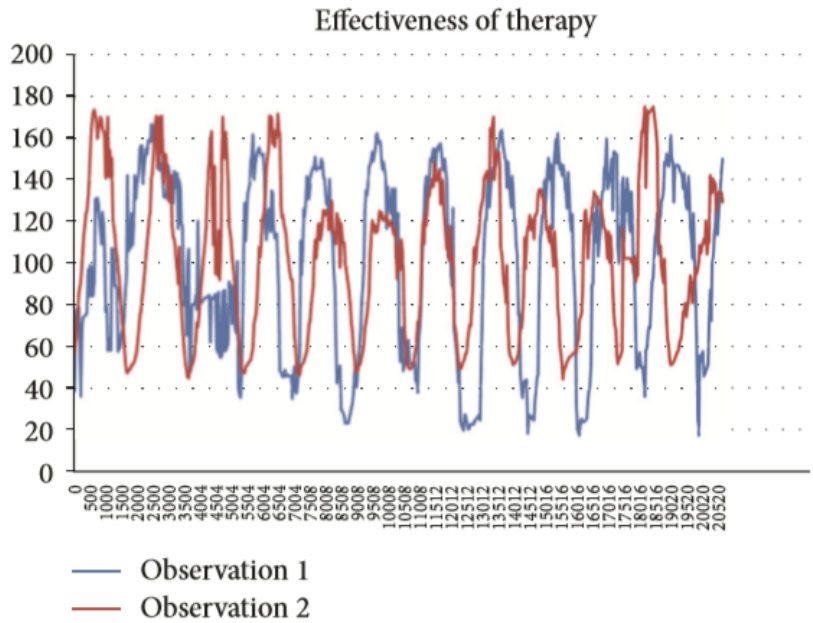


Figure 3.1: Visualization used in [22] depicting the degree of forearm movement overtime

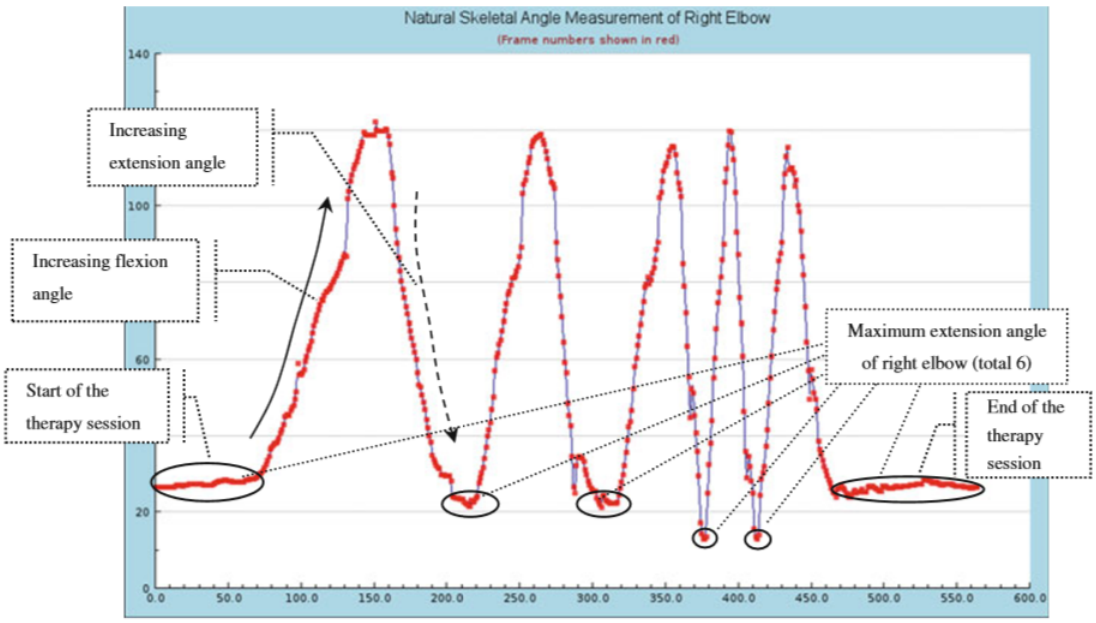


Figure 3.2: Visualization used in [23] depicting the speed of movement (m/s) of forearm overtime

a set of movement that have been configured beforehand by the therapist. During the game, the movement of each body joint is recorded and saved in Session Recorder. Afterwards, a Kinematic Analytic component will process this data and visualize the quality of improvement metrics of each body joint movement. Each metric is visualized with a dotted line chart over time as shown in Figure 3.2. Even though it's possible to see which line curve indicate an elbow flexion or extension, the therapist needs to count the number of the curve manually. This is not very efficient when the session is longer and there are more curve to count.

3.2 Visualization of Time Series Data

Since one of the requirement of the interface is to have the information of movement evolution over time, it is interesting to review how a time series data is usually visualized. Tominski and Aigner discuss at length about the techniques of time series data visualization in their book[1]. The visual survey of this book can be found on their website[4]. This section reviews some of these interesting techniques.

Considering that the recorded gameplay data contains spatial information (location of an event happened on the screen), we reviewed techniques which concern visualizing spatio-temporal data. **Flow Map** depicts movements of object over time. Object movements are usually represented by directed trajectories over spatial space(i.e: map) with different color, width, angle of trajectories represent additional information. In order to overcome overlapping trajectories for huge amount of data, usually aggregation techniques (clustering, self organizing map, etc.) are introduced to group similar data point. Figure 3.3 shows an example of flow map depicting photographers movement between cities in Germany [3]. In this case, the aggregation considers three parameters: initial location, destination location, and time period in which the movement happened. Trajectories width indicate the number of photographers who move between the cities. Another visualization technique worth to mention is **Spatio-Temporal Event Visualization** which uses the space-time cube concept. In this concept, the x and y axis usually represent two spatial dimensions while the third axis represent temporal dimension. The events are then represented as graphical objects which are mapped to the space-time cube location. Different events attribute can be represented in different size, colors, shape, or texture. Figure 3.4 shows space-time cube which depict convective clouds [27], human health [26] and earthquake events [12] from left to right. As we can see, the spatial dimension of the left chart is area in pixel while the middle and right chart is a map. The events on the chart are represented with sphere objects with different color and different sizes. Even though space-time cube can portrays the spatio-temporal data, it has some downside. When there are too many events, occlusion is inevitable. It should be coupled with an appropriate interaction technique to allow users see the data from different perspective.

One example of time-series visualization technique which doesn't concern spatial data is Theme River. First introduced in [15], Theme River is used to visualize thematic changes over time of document collection. Each theme is represented as different colors which flows from left to right with different width over different time point. The width

[4] http://survey.timeviz.net

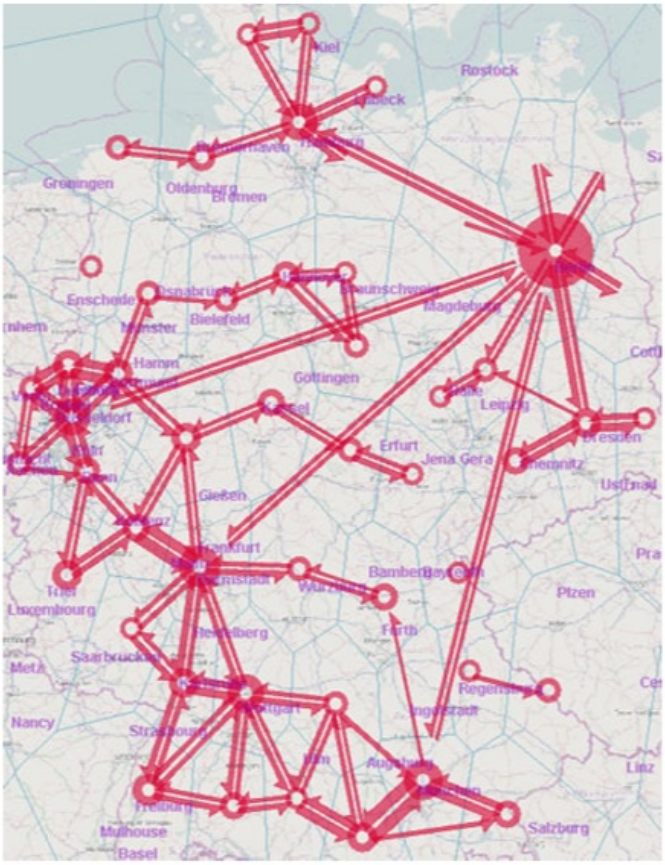


Figure 3.3: Flow Map depicting photographers movement between cities in Germany

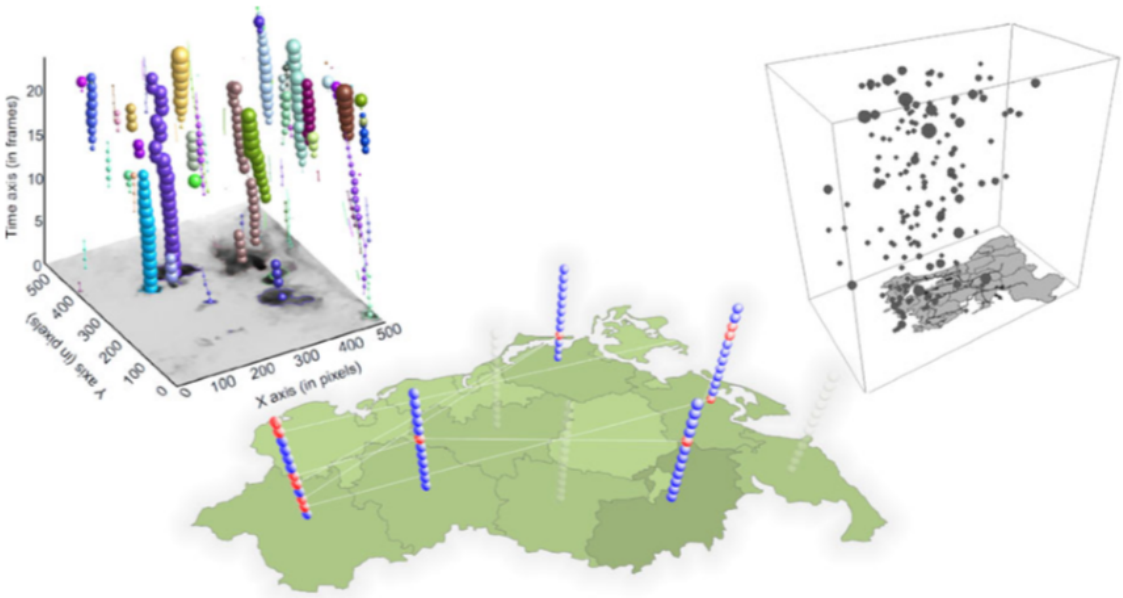


Figure 3.4: Spatio-Temporal Event Visualization

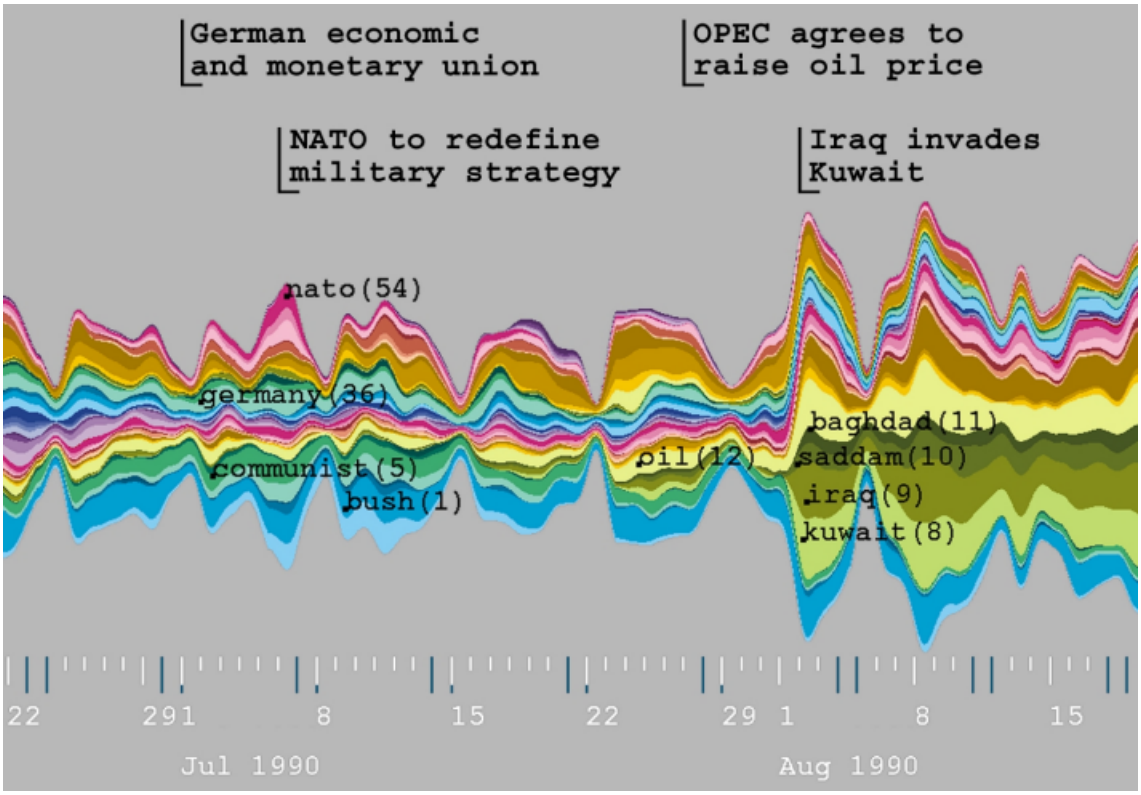


Figure 3.5: Theme River showing evolution of theme in document collection in 1990

depicts theme strength over temporal axis. The purpose of this technique was to easily understand the evolution of theme strength over time. Figure 3.5 shows an example of Theme River representation of 1990 Associated Press newswire data. It can be seen on the chart that the theme baghdad, saddam, iraq, and kuwait are gaining strength around the time Iraq invaded Kuwait on August 2, 1990. By following the flow of a certain color (theme) we can easily see the changes in theme strength and associate it with the events that affects the changes. Theme River should be supported with interaction techniques which allow user to rearrange river positioning over horizontal axis.

Consequently, the Theme River technique is chosen due to its ability to show evolution of a certain data variable over time. Further details on the implementation can be found on Chapter 4.

3.3 Visualization of Movement Data

Movement data usually represents an object which moves over a certain space [2]: data of moving car, birds migration, etc. It's usually recorded as series of location (latitude/longitude, x/y coordinates, etc.) and time. On the other hand, body movement data are recorded as vector representation of human pose [4] over time. On this section, we first review visualization for movement data in general and then discuss visualization for body movement focusing on visualization for skeleton animation[5] data.

[5] https://en.wikipedia.org/wiki/Skeletal_animation)

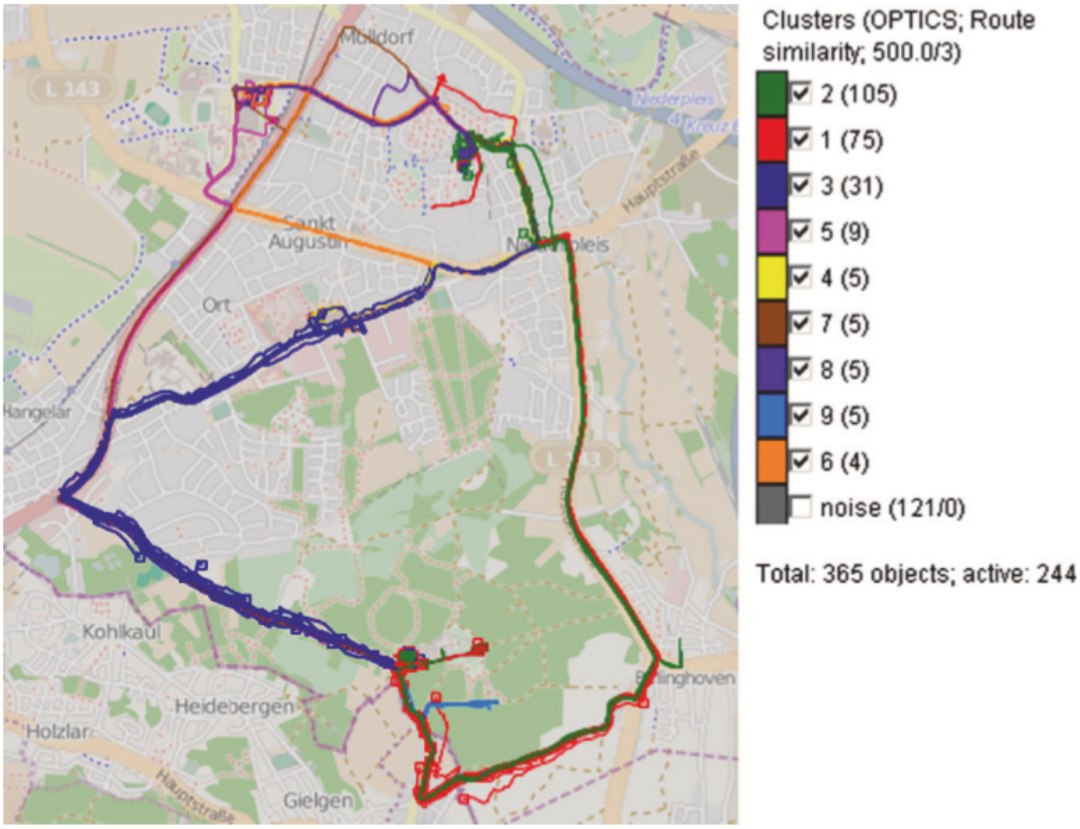


Figure 3.6: Car Trajectories clustered by route similarity

There have been numerous method and application developed to analyse movement data. [3] gives an overview on these methods and applications. Movement data for discrete entities are usually represented as linear symbol over a map or space time cube. However, this technique has problem with occlusions for huge amount of data. Therefore, it's usually accompanied with other graph such as time graph. Other solution to this problem is to use clustering on the trajectories. Apart from minimizing the number of trajectories presented on the view at the same time, clustering also help user to find interesting pattern of the movement. Figure 3.6 [2] gives an example of trajectories of a single car from gps data over several days. The trajectories are divided by stop duration at least 3 hours and clustered by route similarity represented in different colors. Therefore, it is possible to know which route are often or less taken by car owner.

Patterns can also be found by introducing aggregation and generalisation technique on spatial or temporal properties of the movement. For example, the movement data can be aggregated spatially into a discrete grid and for each grid, the number of movement (total or average) happened within the grid can be represented with color or objects in different size. Figure 3.7 [2] shows the presence of cars in Milan in certain geographical area (generated with Voronoi tessellation [20]) during certain time period. The number of cars in the area is represented with a circle in different size which indicate intensity of traffic. As we can see, there are more traffic between 05-06h (left) compared to 22-23h (right). This is understandable since most people leave work around 5 to 6 pm and are

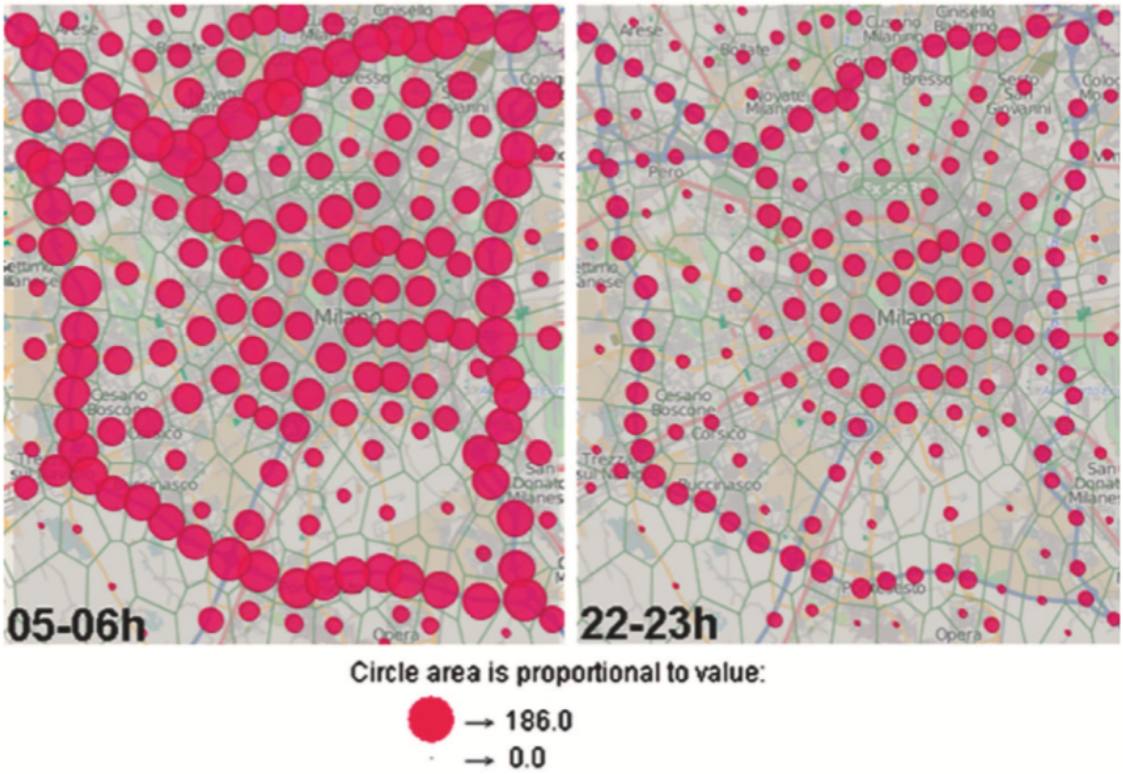


Figure 3.7: Presence of cars in Milan in different time period

already home at 22-23 pm.

Recognizing and understanding human movement has many benefits in different application domain: arts[16, 24], sports[4], healthcare[21], etc. There are numerous research has been done concerning human movement analysis as discussed in [13] which surveyed different methodologies and approaches. Most of the methodologies discussed focus on identifying a certain type of movement. On the other hand, to our knowledge, there hasn't been many research which focus on human body movement visualization in which user can explore and analyse a certain data set.

[8] proposes a system to track and visualize body movement on a virtual environment in real time. In this system, body parts which desired to be tracked are attached to an optical system with twelve infrared cameras. Once user move the tracked body parts, a "motion trail" will be shown in the virtual environment in which then user can manipulate its representation by changing the color, shape, smoothness, etc. These interaction also conducted directly in the virtual environment. Figure 3.8 [8] shows the motion trail produced in the virtual environment while a user move the tracking device in his hand. On the right is the interface where user can interact with the visualization.

Another approach to visualize body movement is by using color belt [25]. In this approach, movement data collected from motion capture system with 11 sensors attached in body joints (Figure 3.9 left) are grouped into 4 limbs movement: Right Arm, Left Arm, Right Leg, Left Leg. Each of the limb representation is arranged in vertical axis sequentially forming a belt. The horizontal axis represents time (from left to right) and

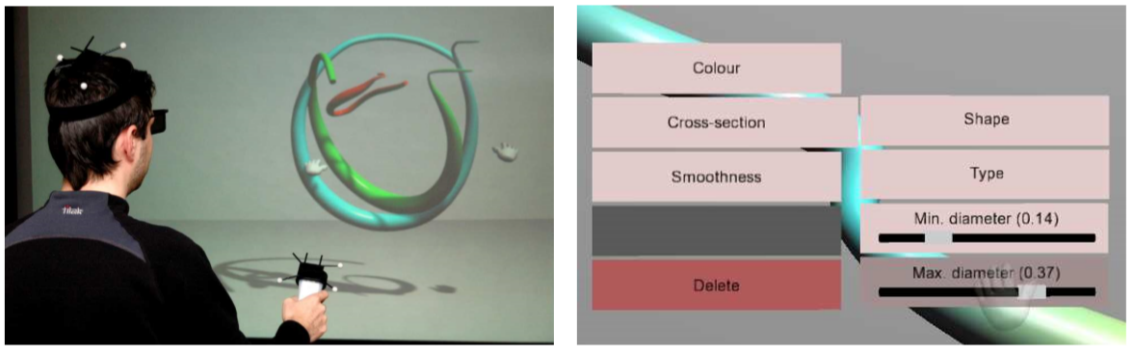


Figure 3.8: Motion Trail

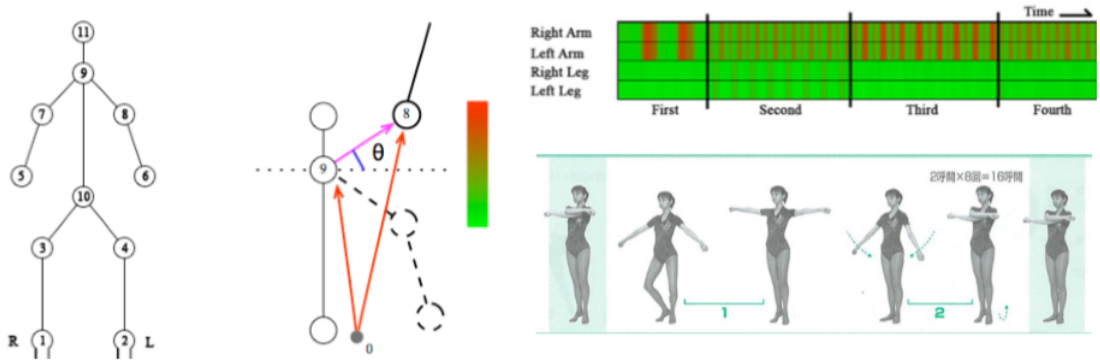


Figure 3.9: Color Belt depicting gymnast body movement

the sections represent sets of movement. Limb motions are shown as gradation of red-green colors in this belt by extracting the position and angle of associated body joints data (Figure 3.9 middle). Positive angle is represented with red color, while negative angle is represented with green color. Figure 3.9 (right) shows a color belt and how it is related to the body movement done by a gymnast. The color belt shows the first to fourth movement set. As can be seen in the picture, the gymnast move her leg on the second exercise and the second section on the color belt depicts the movement for right leg and left leg.

MotionExplorer[4] introduced human motion exploratory search using hierarchical aggregation. This approach are directed towards the need to explore huge quantity of motion data and be able to identify interesting sequence of movements. Implemented on database which contains various motions in multiple repetitions, first each human pose data is clustered using k-means algorithm. A pose cluster comprises of a large numbers of similar human pose and is represented as a circular glyph with human stick-figure pose as the centroid and set of pose in the cluster as deviating, transparent figures. The cycle around cluster glyph are colored based on color legend and shows similarity among clusters. MotionExplorer provides 4 views (Figure 3.10): (i) *Pose hierarchy explorer* (top left) allows user to explore all available pose cluster in the data sets hierarchically. The pose cluster hierarchy is shown as a dendogram and calculated with a divisive clustering algorithm. The aggregation level is adjustable. (ii) *Motion explorer* (top right) shows

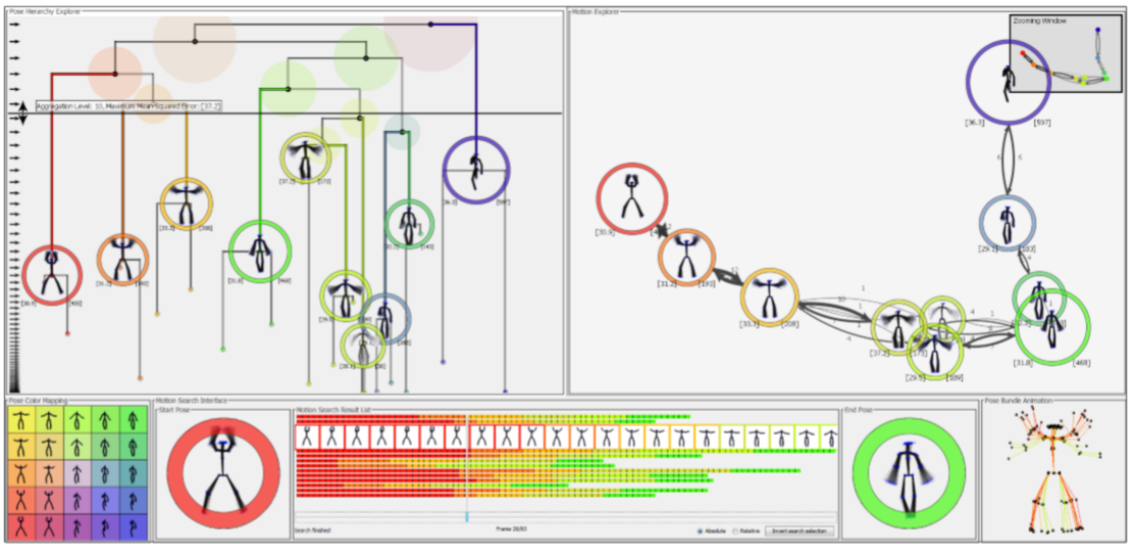


Figure 3.10: MotionExplorer interface

sequences between pose clusters at current aggregation level. Human poses are represented as nodes while edges represent motion sequences. Two nodes are connected if there is at least one motion sequence which connects both pose clusters. (iii) *Motion search* interface (bottom right) allows user to input query for a specific motion sequence by inputting start pose and end pose. An interactive search result is provided where user can explore each style variations. The interface is divided into 4 parts: start pose field, search result field, end pose field and pose bundle animation. A user can make search query by choosing/dragging a pose cluster to the start and end pose field. The resulting motion sequences are then shown in the search result field on the granularity of a single human pose. (iv) *Pose color mapping* (box on bottom left) is a color legend related to each pose cluster. The color grid is built using SOM algorithm trained with all feature vectors in a vector quantization scheme. By clicking one of the clusters in the color legend, the aggregation level will be adjusted to the granularity level of the selected pose cluster. Each visualization window is linked. When user changes the aggregation level in the pose hierarchy explorer, the motion explorer is automatically adjusted to the new aggregation level. When user clicks on one of the clusters in the color legend, the system automatically adjust the aggregation level to the same granularity of the selected cluster. If a cluster in one of the view is selected, then the same cluster pose is also highlighted in every view.

3.4 D3.js as Data Visualization Tool

The popularity of data visualization has been supported with the development of a lot of data visualization tool. To support the development of the interface in this thesis, a survey were conducted focusing on tools which allows creation of different visualization technique without restriction of a ready made template or certain chart type.

One of the tools surveyed is D3[5]. **D3** or Data-Driven Documents is a javascript library which allows user to create desired visualization either the standard one (bar, line,

Figure 3.11: Some of visualizations developed with D3.js

scatter plot, etc.) or the customized one (limited only by one's creativity) in 2D. Built by incorporating HTML5, CSS, and SVG standards, D3 enables user to load data in various format(.json,.csv,.txt, etc.)[6] and present it as a chart. The various functionality offered by D3 and its detailed API documentation has encourage a lot of people to explore it. This resulted in numerous amount of D3 demo online which can be a good reference source for new user[7]. Thus, starting using D3 is quite easy with so many example online supported with big community. On top of that, D3 provides different ways to manipulate and interact with the data which enables development of interactive and intuitive visualization. Figure 3.11 shows some of the visualizations built in D3[8]. Considering the visualization requirement identified in the previous chapter as well as the features offered by D3, consequently D3 is chosen as the tool for interface development in this thesis.

[6] https://github.com/mbostock/d3/wiki/API-Reference
[7] http://techslides.com/over-2000-d3-js-examples-and-demos
[8] http://d3js.org

Chapter 4

Data Abstraction

In this chapter we discuss the design of data structure and clustering technique used to support the visualization requirement. First, an overview of the input data generated from the game will be explained. Then a description on how this data is extracted to be the input of the visualization interface will be given. Finally, a clustering algorithm proposed to be used in the visualization will be discussed.

4.1 Game Events Structure

4.1.1 Log Data Overview

The log of each game session is stored in MongoDB database in JSON format. Each log file contains information of the player, game setting (movement type, game direction, game duration and repetition), objects apparition, events happened during the game, as well as player's body joint position. For object apparition, the record contains object id, apparition location, object type (Reef, Planks, etc.), and timestamp. For events, the record contains object id, event location, event type, and timestamp. For body joint position, the record contains the position of 20 body joints (head, neck, left elbow, left wrist, etc.) and the time when this exact body joint position is captured. Each location data is recorded in (x,y,z) coordinates.

Based on the data available in the log file, there are two approaches can be done to visualize body movements. First, by exploring the body joint position data. With this approach, it's possible to know which part of the body is moved as well as the degree of the movement by calculating the position of each body joint. The second approach is by exploring the events data. Since boat movement is navigated by player, boat location in the game world can be interpreted as the position of the body. Hence every event happened to the boat also indicate the position of the body and therefore represent body movement. Considering the time constraint, in the end we decided to use the second approach and thus putting aside the body joint position data.

Considering the size of the data and to have an acceptable application response time when user perform an interaction on the visualization, the log data is first extracted to reduce it's size. Since we only interested to events data, we process the log data so that in the input of the visualization we have records which consist of game setting information and events data. For each event record, we extracted event type, event location on x axis, time, and screen speed. This input is used for $T1$.x. For $T2$.x, another extraction is performed by combining the data from all sessions and adding the session information for each event record.

4.1.2 Event Category

The goal of Hammer and Planks game is to kill all of the enemies while avoiding any attack from the enemies and obstacles[9]. Along the way, player can also catch bonuses to increase their score. Based on these, we identified three different objects within the game: Enemy, Bonus, and Obstacle. For each of these object, there are certain events associated. Each event has an event type. In total, there are 8 event types:

(1) Catch: when a bonus is catched

(2) Miss: when a bonus is missed or player's attack on enemy is missed

(3) Dodge: when an obstacle is avoided

(4) Collision: when the player's boat collide with an enemy or obstacle

(5) Kill: when an enemy is destroyed by player's boat

(6) Hit: when the player's attack hit an enemy

(7) Hurt: when the enemy's attack hit player's boat

Based on the level of impact of each event to the user's boat, we characterize the event by assigning it with Positive, Neutral, or Negative as shown in Table 4.1.

Event Type	Bonus	Obstacle	Enemy
Positive	catch	-	kill,hit
Neutral	miss	dodge	miss
Negative	-	collision	hurt, collision

Table 4.1: Event Type grouping

4.1.3 Game World Coordinates

The game world is a never ending ocean which is shown on the screen such that players see everything what happened from the sky. Each object and event in the game are assigned with 3D location coordinates (x,y,z). An x axis of this coordinate indicates horizontal axis of the screen. Zero x axis is in the middle of the screen. Thus, an event happened on the right half of the screen will have positive x value and an event happened on the left half of the screen will have negative x value. y axis indicates vertical axis in the game world which means -y is a location under the ocean and $+y$ is above the ocean. z axis indicates vertical axis of the screen. Zero z axis is in the middle of the height of the screen, thus -z is a location on the lower half of the screen and $+z$ is a location on the upper half of the screen. The visualization interface uses x coordinate information to represent body movement to the right or left and z axis to calculate screen speed as explained in the following sub section.

4.1.4 Screen Speed

In the game, a big number of positive events indicate a good player's performance. However, it is important to consider whether the events happened when the player's boat move fast or slowly $(T1.2)(T1.4)$. Getting all the bonuses while moving fast requires precise hand/body movement which indicates improvement in rehabilitation process. Boat speed while navigating the sea is basically the speed in which the screen scroll. This is calculated by identifying the location(apparition z coordinate θ_{apr}) and time (apparition time t_{apr}) of an object when it first appears on the screen, and location(event z coordinate θ_{evt}) and time(event time t_{evt}) when an event happened on that object. Screen speed(υ_{scr}) is distance between apparition location and event location divided by duration between apparition time and event time.

$$\upsilon_{scr} = \frac{\theta_{evt} - \theta_{apr}}{t_{evt} - t_{apr}}$$

4.2 Clustering

In understanding the evolution of movements among different sessions over x-area, it is interesting to see what the common evolution of different section of the x-area$(T2.5)$. The idea is to cluster similar distribution of movements (which is represented by events in the game) so that consecutive sections which have similar evolution is represented by a single representation. This section explains how the clustering is done. First, a distance function used to calculate the difference between two consecutive section will be presented. Then, a clustering algorithm based on hierarchical clustering which incorporate this distance function will be explained.

4.2.1 Distance Calculation

Let S be a gameplay data set of n_{ses} sessions. S is an ordered list of sections $s_i, 0 \leq i < n_{sec}$. Each section contains events occurred on an x-axis unit among all sessions. More precisely, a section s_i is a sequence of triplets $s_i[j], 0 \leq j < n_{ses}$. Each triplet represents data set of a certain game session of a particular section s_i. The triplet consists of the number of negative, neutral and positive events. The profile of each section can be represented by a matrix of n_{ses} x 3 dimensions. For instance, the following matrices represents a gameplay with 2 sessions and 3 sections:

$$s_1 = \begin{bmatrix} 10 & 20 & 6 \\ 20 & 5 & 18 \end{bmatrix} s_2 = \begin{bmatrix} 20 & 40 & 10 \\ 40 & 10 & 30 \end{bmatrix} s_3 = \begin{bmatrix} 10 & 20 & 5 \\ 16 & 4 & 12 \end{bmatrix}$$

In this example, we see that on the left part of the x-axis (s_1), the player got 10 negative, 20 neutral and 6 positive events for the first session. He then got 20 negative, 5 neutral and 18 positive events for the second session. On the middle part of the x-axis (s_2), on the first session he got 20 negative, 40 neutral and 10 positive events, while on the the second session he got 40 negative, 10 neutral, and 30 negative events.

Since the idea was to aggregate consecutive sections with similar movement pattern, a function to quantify the similarity between two sections is needed. In this case, we

quantify the similarity by defining a distance function. Thus, distance equals 0 indicates that both sections are similar, while distance equals 1 indicates that both sections are different. We identified that there are two types of distance need to be considered: (i) how different both sections in term of event types proportion within each section (ii) how different both sections in term of the evolution of each event type throughout the sessions. For instance, with the (i) approach, s_2 and s_3 are similar, because triplets of each sessions are proportional ([20,40,10] is proportional to [10,20,5] and [40,10,30] is proportional to [16,4,12]). With the (ii) approach, s_1 and s_2 are similar, because the corresponding event types are proportional ([10,20] is proportional to [20,40], [20,5] is proportional to [40,10], and [6,18] is proportional to[10,30]).

Thus, for each pair of consecutive sequences (s_1,s_2) of S, distance is defined as weighted sum of (i) and (ii), represented as $f(s_1, s_2)$ and $g(s_1, s_2)$ in the following formula:

$$d(s_1, s_2) = \alpha f(s_1, s_2) + (1 - \alpha) g(s_1, s_2)$$

Distance function $f(s_1, s_2)$ is based on a normalized euclidean distance between two triplets of the same session between two consecutive sections. The overall distance of both sections is the average euclidean distance of each triplets pair. To get a value of distance between 0 and 1, each Euclidean distance value is normalized by dividing it with maximum distance $\sqrt{3}$.

$$f(s_1, s_2) = \frac{\sum_{i=0}^{i<|s_1|} NED(s_1[i], s_2[i])}{|s_1|}$$

$$NED(s_1[i], s_2[i]) = \frac{\sqrt{\sum_{j \in \{0,1,2\}} (s'_1[i][j] - s'_2[i][j])^2}}{\sqrt{3}}$$

In this first distance function, s'_1 and s'_2 represent normalized value of s_1 and s_2. Since the first distance is to see the difference of event type proportion in each section, thus the values are normalized by the maximum value of each sessions in each section. In this case, two consecutive sections with different number of events but similar proportion of event type will have distance equals 0. For example, the matrices presented previously can be normalized into the following matrices:

$$s'_1 = \begin{bmatrix} \frac{1}{2} & 1 & \frac{6}{20} \\ 1 & \frac{1}{4} & \frac{18}{20} \end{bmatrix} s'_2 = \begin{bmatrix} \frac{1}{2} & 1 & \frac{1}{4} \\ 1 & \frac{1}{4} & \frac{3}{4} \end{bmatrix} s'_3 = \begin{bmatrix} \frac{1}{2} & 1 & \frac{1}{4} \\ 1 & \frac{1}{4} & \frac{3}{4} \end{bmatrix}$$

Thus, distance f of s_1 and s_2 equals 0.06, and distance f of s_2 and s_3 equals 0.

The second distance function $g(s_1, s_2)$ is based on a normalized euclidean distance between the same event type j from different section. The overall distance of both sections is the average euclidean distance of each event type pair. For each event type distance, there are three different distance value defined: (i) if there are no events in both event type pairs, the distance is 0 (ii) if there is at least one event in one section and there are no events in the other section, the distance is 1 (iii) if both event type pairs has any events then the euclidean distance is calculated. Distance (i) and (ii) are defined to

isolate empty regions. Thus when both sections are empty, it will be considered similar and will be clustered together. On the other hand, when only one section is empty while the other has some events, it will be considered as different session and therefore will not be clustered together. To get a value of distance between 0 and 1, each Euclidean distance value is then normalized by dividing it with maximum distance $\sqrt{|s_1|}$.

$$g(s_1, s_2) = \frac{g_0(s_1, s_2) + g_1(s_1, s_2) + g_2(s_1, s_2)}{3}$$

$$\text{and for } j \in \{0, 1, 2\},\ g_j(s_1, s_2) = \begin{cases} 0 & \text{if } s_k[i][j] = 0 \text{ for each } 0 \leqslant j < |s_k| \text{ and } k \in \{1, 2\} \\ 1 & \text{if } s_k[i][j] = 0 \text{ for each } 0 \leqslant j < |s_k| \text{ and } \exists s_p[i][j] \neq 0, \\ & k, p \in \{1, 2\},\ k \neq p \\ \dfrac{\sqrt{\sum_{i=0}^{|s_1|} (s_1''[i][j] - s_2''[i][j])^2}}{\sqrt{|s_1|}} & \text{otherwise} \end{cases}$$

Here, s_1'' and s_2'' represent normalized value of s_1 and s_2. In order to compare the evolution of the same event type between consecutive sections, the values are normalized by dividing it with maximum value of each event type in the section. Thus, the matrices presented previously can be normalized into the following:

$$s_1'' = \begin{bmatrix} \frac{1}{2} & 1 & \frac{1}{3} \\ 1 & \frac{1}{4} & 1 \end{bmatrix} s_2'' = \begin{bmatrix} \frac{1}{2} & 1 & \frac{1}{3} \\ 1 & \frac{1}{4} & 1 \end{bmatrix} s_3'' = \begin{bmatrix} \frac{5}{8} & 1 & \frac{5}{12} \\ 1 & \frac{1}{5} & 1 \end{bmatrix}$$

Here, distance g of s_1 and s_2 equals 0, and distance g of s_2 and s_3 equals 0.06.

4.2.2 Clustering Algorithm

The clustering algorithm follows Hierarchical Clustering algorithm [18] which is a clustering analysis method used to build hierarchy of clusters. Overall, the clustering is done as follows: Initially, the data set is divided into sections the size of x-axis unit. Then, distance of consecutive sections are calculated. Two consecutive sections with distance below a specified threshold are then merged. This resulting in a new set of sections. Then, the distance of each consecutive sections of this new set of sections are calculated and again, if the new distance satisfies the threshold, the sections will be merged. This process is repeated until there is no distance falls below the threshold.

Chapter 5

Visual Mappings And Interactive Functionalities

This chapter describes different visualizations and interaction methods developed based on the visualization requirements and data abstraction discussed in Chapter 2 and 4. In general, the visualization is divided into two parts: (i) Session Visualization which visualize movement in one particular session (for $T1.x$) and (ii) Summary Visualization which visualize movement over different sessions (for $T2.x$). Both visualization is organized in an application where user can select player and sessions he has played. At first, the earlier version of visualization will be explained. This earlier version is not used in the final version since it's difficult to get any information intuitively. Then, each type of visualization and its interaction used in the final version will be discuss. In the end, the application which encapsulate both visualization will be presented.

5.1 Early version of Session Visualization

The first visualization method chosen to represent ($T1.1$) is line chart. In this approach, the x area is shown as a horizontal axis and the number of events shown in vertical axis with the line signify the changes of number of events for different x area unit. In the log file, each event is recorded with distinct 3D coordinate location. The x value from this coordinate is a decimal, therefore visualizing each one of this x value will require a lot of space. To solve this, the events are then grouped by the rounded x value. In Figure 5.1 below, Negative events are shown in red line and Positive events are shown in blue line. As we can see, it's possible to know which event type happened more in a certain x unit, however it is difficult to see how big a percentage is it compare to total number of events happened in the same x unit.

Visualization method chosen to represent ($T1.2$) is scatter plot. At first, each event type is presented in three different chart area: top are for Positive events, middle are for Neutral events, and bottom area for Negative events. Similar to the line chart, the x value from the 3D coordinate location are represented in horizontal x axis. However, the vertical axis here represents screen speed. For the scatter plot, each event is shown as a

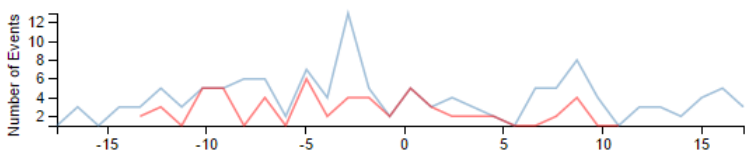


Figure 5.1: First visualization version for ($T1.1$) using Line Chart

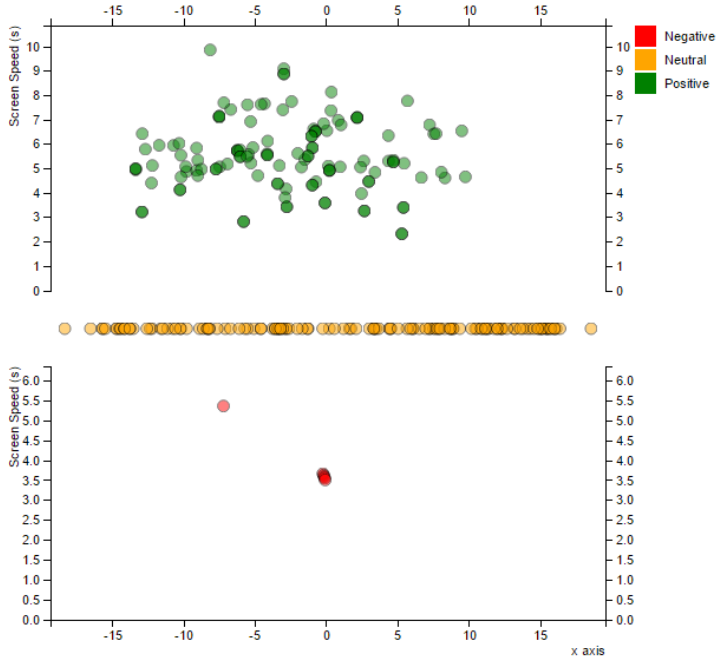


Figure 5.2: First visualization version for (T1.2) using Scatter Plot

plot in the chart area according to it's x value and screen speed as shown in Figure 5.2. Even though similar pattern can be seen on the scatter plot, occlusion problem prevents users to know how many event are actually happened in a certain x area.

5.2 Session Visualization

The Session Visualization shows events within a game session. The requirement can be split into two: (i) knowing the distribution of events and movements (ii) knowing the distribution of events, movements and screen speed. Basically, (ii) is a detailed view of (i). Therefore, there are two chart developed to meet these requirements: stacked area graph for (i) and heatmap for (ii), each of which will be explain in details in this section.

5.2.1 Stacked Graph

Build on layered area graph, Stacked Graph is widely used to visualize evolution of variable over times such as document theme [15], box office movie revenue[1], listening history in Last.fm [7],etc. Stacked Graph is chosen because its ability to show individual value of a variable, the difference between values of different variables as well as the total of overall value. In our approach, instead of using this metaphor to show evolution over time, it is used to show distribution of events over spatial coordinate (T1.1) as shown in Figure 5.3. Here, the horizontal axis represents x coordinate and vertical axis represents number of events. Each event type is represented as an area with different color: Red (Negative),

[1] http://www.nytimes.com/interactive/2008/02/23/movies/20080223_REVENUE_GRAPHIC.html?_r=0

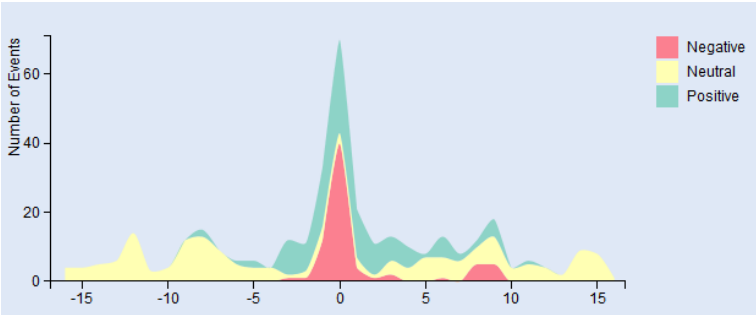


Figure 5.3: Stacked Graph with Linear Layout

Yellow (Neutral), and Green (Positive). However, with this approach it is difficult to see the trend for each individual events which is not on the base of the chart. [10] propose an interactive solution to this problem by sinking the selected category to the horizontal axis. Thus, in our visualization, different layout of stacked graph is provided:

1. Linear (Figure 5.3): zero y axis is used as the baseline, with the stack ordered from the bottom as Negative, Neutral, Positive.

2. Silhouette (Figure 5.4a): the graph is centered as in streamgraphs.

3. Positive (Figure 5.4b): zero y axis is located at the top of the chart and is used as the baseline with the stack ordered from the top as Positive, Neutral, Negative.

4. Neutral-Negative (Figure 5.4c): zero y axis is located in the middle of the chart. Neutral and Positive events are shown on the positive area of y axis and Negative events are shown on the negative area of y axis.

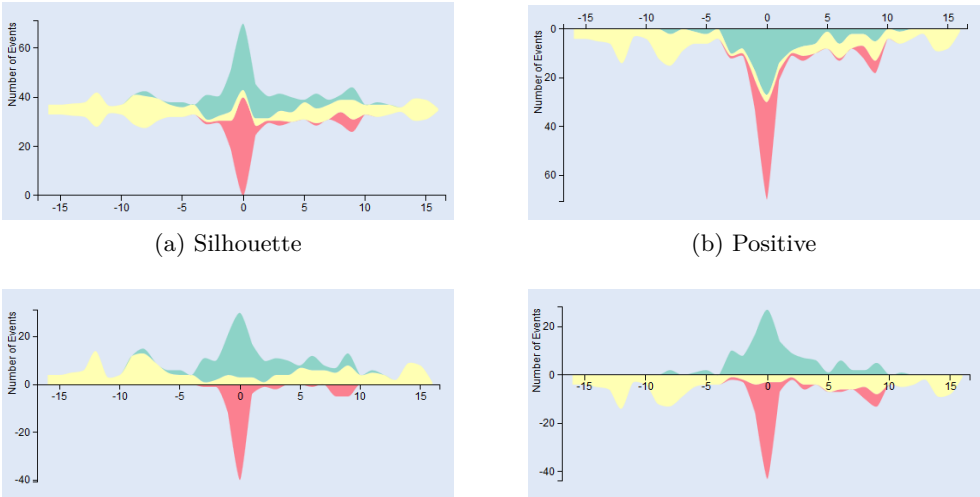
5. Positive-Neutral (Figure 5.4d): zero y axis is located in the middle of the chart. Positive events are shown on the positive area of y axis, while Neutral and Negative events are shown on the negative area of y axis.

For each stacked graph layout, user can choose which object type to show on the graph (T1.3). Options are available as radio button on top of the chart. Therefore, choosing Bonus will show only Positive and Neutral events, choosing Obstacle will show only Neutral and Negative events, and choosing Enemy will show all event type.

5.2.2 Heat Map

Heat Map is a quite popular visualization method nowadays due to its ability which allows user to see variable with the highest value at one glance. Most of the time, heatmap is implemented on geographical map to represent variable value over certain area on map, i.e: Natural Disaster Risk by Location[2], population density[3], Number of picture taken in an area[4], etc. Heat map is also used to track eye movement or mouse click on a website, and representing DNA microarray data in the form of cluster heat map[11]. Heat map

[2] http://www.rms.com/
[3] https://en.wikipedia.org/wiki/Population_density
[4] http://sightsmap.com/



(a) Silhouette (b) Positive

(c) Neutral-Negative (d) Positive-Neutral

Figure 5.4: Different Layout of the Stacked Graph representing number of events over x-axis

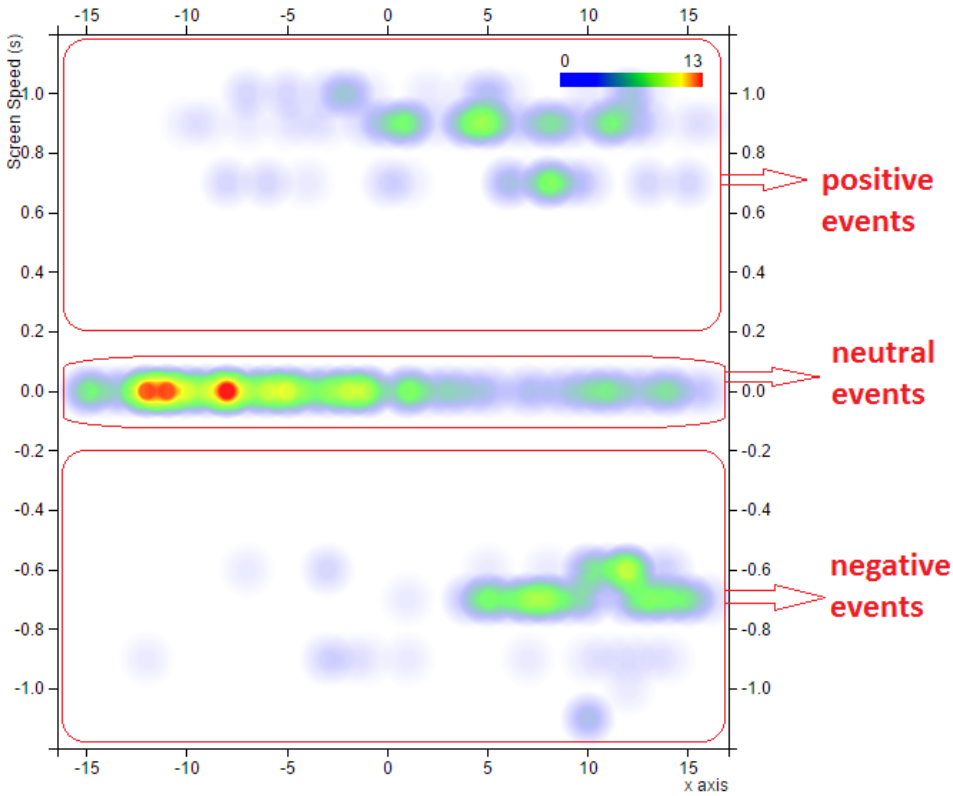


Figure 5.5: Heatmap showing movement distribution over x-axis and screen speed

uses color gradation to represent the hotness level of a variable. Usually, red color is used to represent the high value (hot) and blue is used to represent the low value (cold). However, other color combination can also be used. To represent distribution of events and screen speed over x axis ($T1.2$), the events are first grouped based on their x values and normalized screen speed. The number of events is then represented as heat map on the graph with highest number of events in red color and the lowest number of events in blue color. Normalized screen speed is represented as vertical axis and x value is represented as horizontal axis. Using the same approach used in scatter plot chart, each event type is presented in different area: top for Positive, middle for Neutral, and bottom for Negative (Figure 5.5). For Neutral events, the screen speed is not calculated since it basically mean an object has been avoided or missed. For Negative events, the screen speed is represented in negative to show that it's an uncalculated movement. For the heatmap, user can also choose to show a specific object ($T1.4$) by clicking the radio button associated with the desired object.

5.3 Summary Visualization

The Summary Visualization fulfills the requirements concerning movement evolution over time ($T2.x$). In this case, a single session is considered as a single time point. Since user are interested in evolution over a certain x-area, the visualization are divided into sections of x-area. There are three ways of division: by the range of x-area, by the number of events within an area, and by clustering. The following explains the three approaches and its interaction technique in details.

5.3.1 Visualization by range of x-area

To fulfill ($T2.1$), a streamgraph metaphor is chosen to show evolution of movement (represented by events) over time. Time is often displayed on x-axis, but here, x-axis represent the x-axis of the screen, so time is displayed on the y-axis. Some visualizations are also based on this approach, like Visual Sedimentation [17]. Here, sessions represent time with the earliest one shown at the bottom and the latest one shown on the top. The x axis is then divided into sections of the same range based on user input. For each section, events are then filtered to the one which happened within the section x boundary. The filtered events are grouped based on session number and event type. Number of events within this group is then presented in vertical streamgraph layout with event type represented using the same color used in the Session visualization (Figure 5.6).

Shown in Figure 5.7 a section in the chart. Here, the lower x boundary is -20 and the upper x boundary is -7.33. Within this area, the evolution of events throughout all session can be seen($T2.1$). It is also possible to see which session has the most or least number of events by comparing the total length of all event type in one session.

5.3.2 Visualization by number of events

This second type of Summary Visualization uses the same approach explained previously. However, a section is calculated based on the total number of positive and negative events instead of the range of x-axis ($T2.3$). Therefore, based on the distribution of positive and

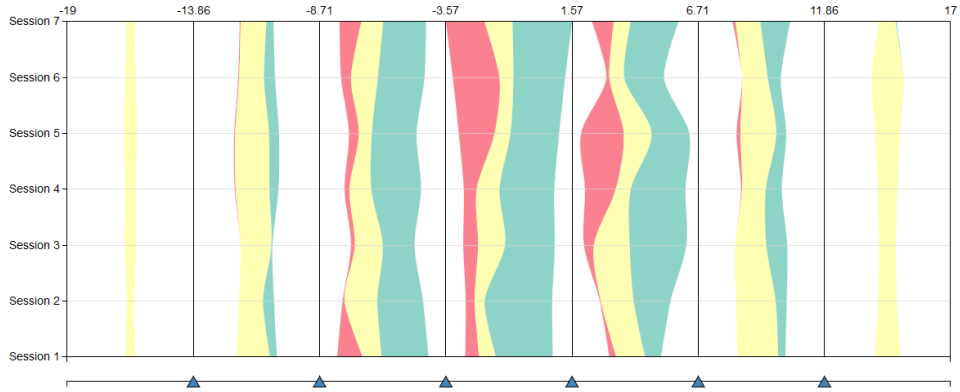


Figure 5.6: Summary Visualization divided by range of x-area

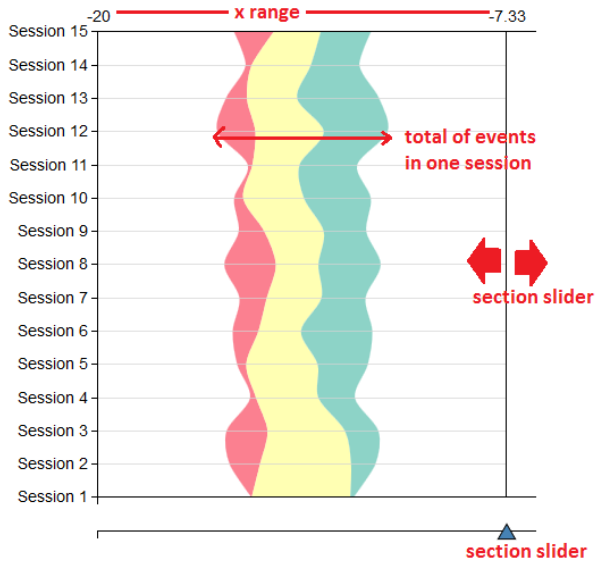


Figure 5.7: A section in Summary Visualization

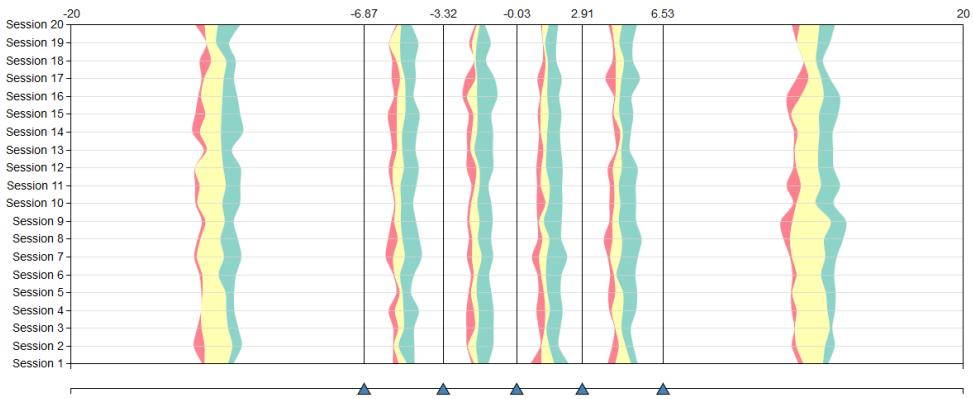


Figure 5.8: Summary Visualization divided by number of events

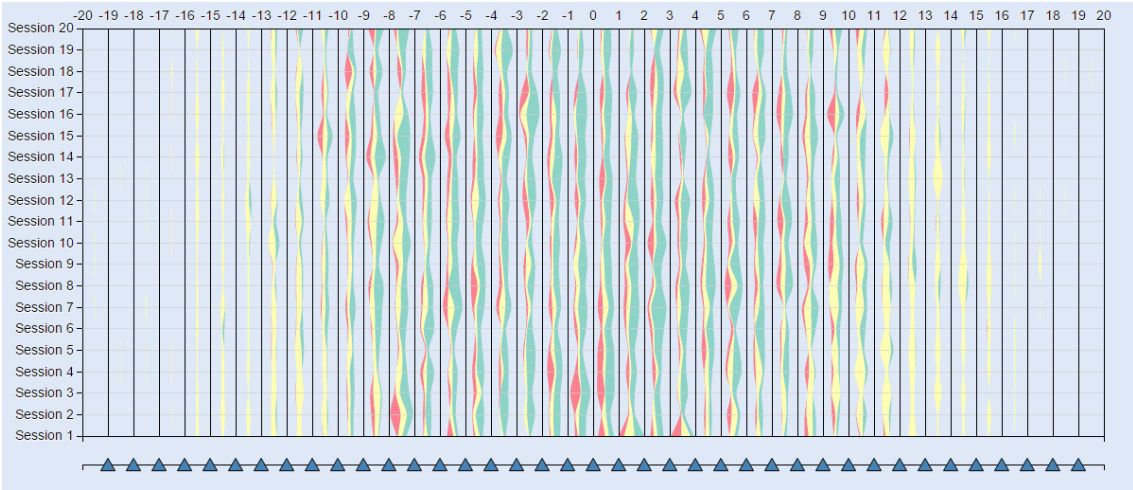


Figure 5.9: Summary Visualization divided by movement evolution similarity, clustered with threshold = 0

negative events, one section in the chart may have bigger x-range than the other section. Only by comparing the size of sections, it's possible to know in which area most of the events are concentrated. Figure 5.8 shows that the events are more concentrated in the middle area of the screen. Here, it can be concluded that on the far right and far left of the screen, there are more neutral events compare to the middle area.

5.3.3 Visualization by clustering

This third type of visualization fulfills (T2.5). As explained in section 4.2, initially the chart is divided into sections with range equals to an x-axis unit. Depending on the threshold value inputted by user, consecutive sections with distance below the threshold will be merged. This process is repeated until there is no sections with distance below the threshold. The input threshold ranges from 0 to 1. Thus, when user inputted threshold = 0, none of the section will be merged. On the other hand, when user inputted threshold = 1, all of the sections will be merged. Figure 5.9 shows when user input threshold = 0, while Figure 5.10 shows threshold = 0.24. As we can see, there are some sections which are merged, indicated with bigger section size. 4 sections are merged together forming one section with x range between 16-20 and 12-16. On the left side of the chart, 7 sections are merged together forming one big section ranging from -13 to -20.

5.3.4 Interaction Technique

On top of the chart, an interaction bar is provided where user can interact with and change some variable in the chart. In first panel of the interaction bar (Figure 5.11), three sliders are provided: (i) slider to input the number of slices so that each section will have the same x-range (ii) slider to input the number of slices so that each section will have the same total number of positive and negative events (iii) slider to input threshold so that each section will have a cluster of similar movement evolution. When user uses slider (i) and (ii), the value on the slider defines the number of sections on the chart. For

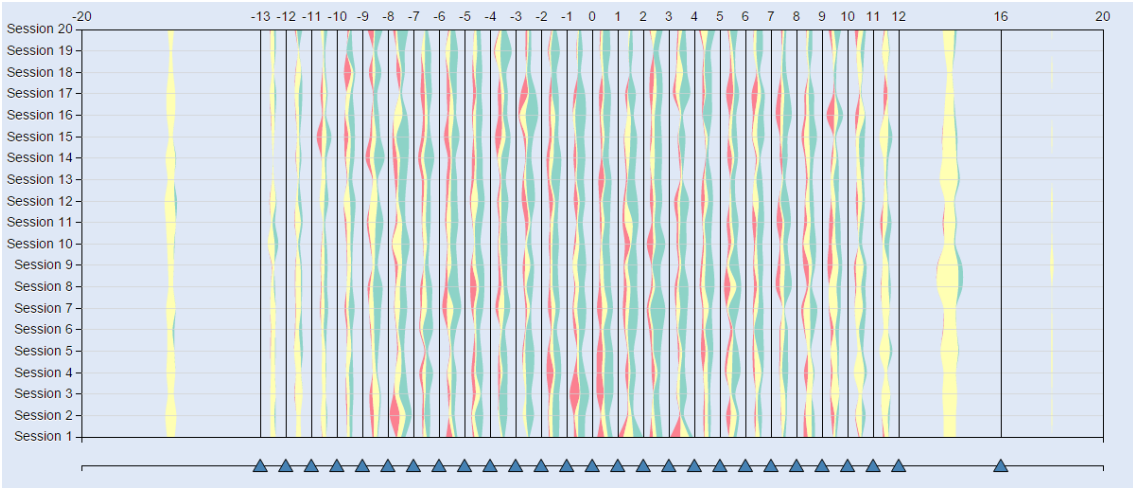


Figure 5.10: Summary Visualization divided by movement evolution similarity, clustered with threshold = 0.24

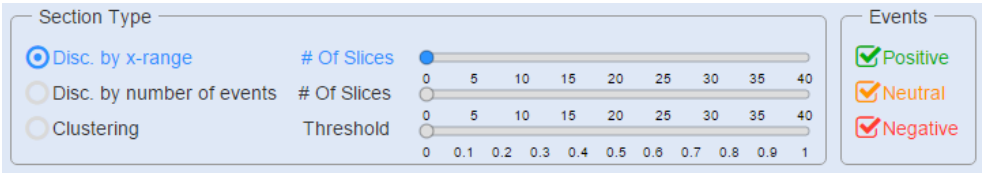


Figure 5.11: Interaction Bar for Summary Visualization

(i), the more number of sections, the smaller the x-range. While for (ii), the value selected on the slider is a denominator. The input is calculated by dividing total number of all positive and negative events by the value selected on the slider. Thus, the smaller the number chosen on the slider, the bigger the number of events. In the second panel, user has the options to choose which event type to show on the chart. This fulfills requirement ($T2.2$) and ($T2.4$). By default all event type will be shown.

Once the chart is generated, user has the ability to slide/drag the line between each session or the small triangle at the bottom of the line to the right or left(see Figure 5.7) to change the range of it's neighbouring sections($T2.1$)($T2.3$). While dragging, the text on top of the line changes based on the current x value of the dragged line. When a line is dragged over another line, the two sections will be merged creating a new section with different x range. Therefore, user may be able to gain the information in which particular area a certain type of events starting to happened. Figure 5.12 shows changes on affected sections when line 11.63 is dragged to the left to position 10.7. We can see that Negative events on the circled area started to appears from x = 10.7. It is also possible to divide a section into two sections by clicking the top area of the chart in between the lower and upper section boundary text. This allows user to know the distribution of event type within a section.

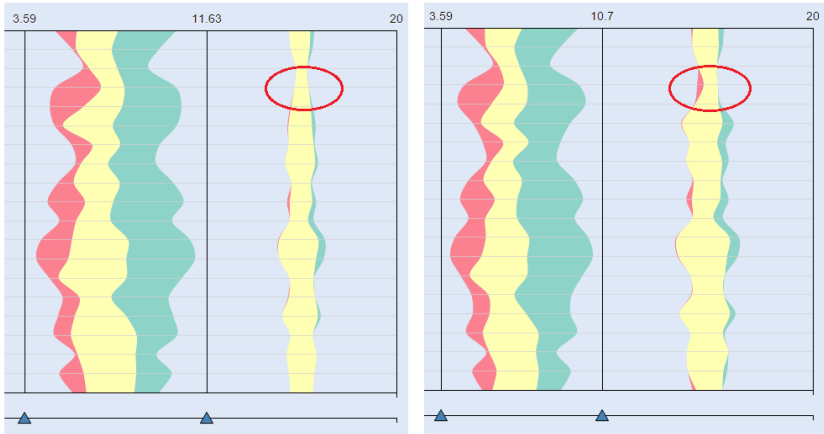


Figure 5.12: Navigating section line to highlight pattern

5.4 General Interface

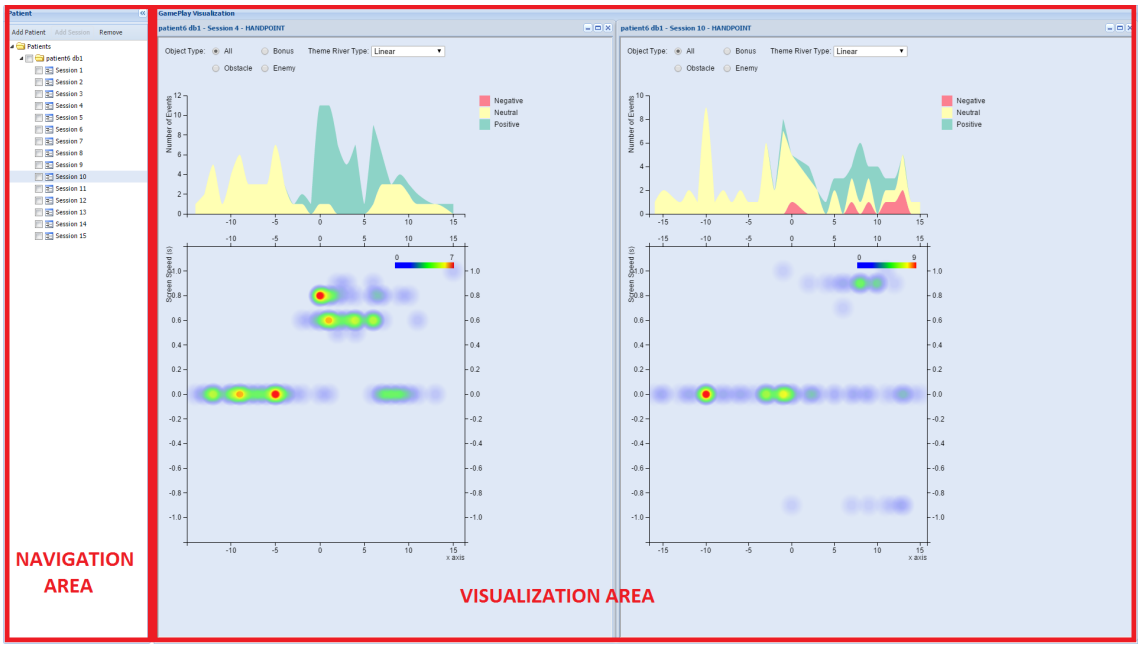


Figure 5.13: Application Interface

Both the Session and Summary visualization are attached into an application which allows user to navigate different player and session. Unlike the visualization interface, the general interface is built using extjs library which allows the development of desktop-like web application. The interface of this application is divided into two areas: Navigation area and Visualization area (Figure 5.13). In Navigation area, user can choose patient and the sessions they have played. On clicking a session, a Session visualization of this session will be shown on the visualization area. It is also possible to open more than one Session visualization and rearrange the visualization window to compare gameplay be-

tween sessions (similar to navigating multiple windows in desktop). On clicking patient's name, Summary visualization for the chosen session will be shown.

Chapter 6

Case Studies

To evaluate the visualization functionality developed in this thesis, two case studies will be presented. There are two types of data collected: log files from healthy person who played the game and log files from patient. The log files from healthy person are gathered in duration of three weeks with each session played in different day. The patients' data are collected by NaturalPad[1]. Unfortunately, it is not possible to get information concerning patients' pathology due to confidentiality reason.

6.1 Case Study 1: Healthy Player

The first case study is based on log files of game played by our colleague over the course of three weeks. Each game is set to be played in 2 minutes duration with 3 repetitions. In total, there are 20 sessions of BODYTILT game collected. Figure 6.1 presents a comparison between the very first and last session played by the player. It shows that in the first session, there are lots of events missed (yellow area) on the far right and far left of the screen which indicate that the player doesn't move her body to that extent. From the peak of area between 0-5 x axis, it can be concluded that the player moves more to the right. The green and red area which indicates positive and negative events only appears around -10 to 10 x axis which indicates that player only moves around the middle of the screen(T1.1). It is understandable for first session because usually player needs time to get use to the game and get the feeling of how far he/she should move his/her boy to reach/avoid an object. On the other hand, in session 20, the peak of the area are more spread out and there are even green area at the far right(T1.1) which indicates that the player is able to move to that extent. It seems that the player already has the feeling of how to play the game.

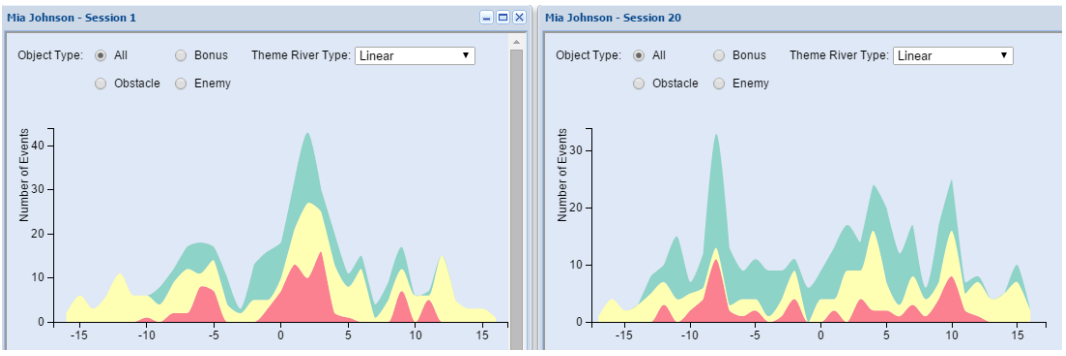


Figure 6.1: Stacked Graph comparison of Session 1 and Session 20 played by healthy player

[1] http://www.naturalpad.fr/

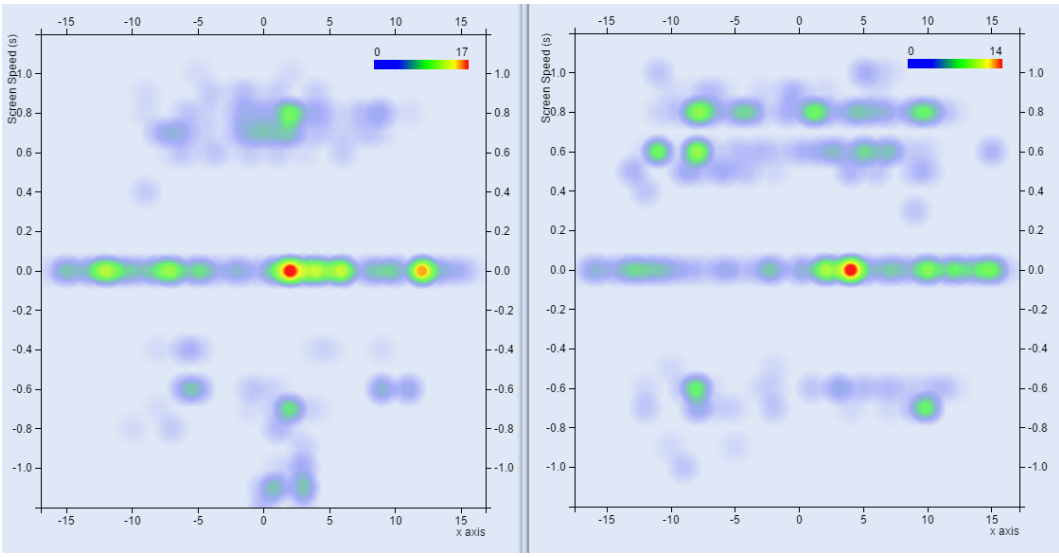


Figure 6.2: Heatmap comparison of Session 1(left) and Session 20(right) played by healthy player

At first glance on Figure 6.2, it is noticeable that there are more spot on the top part of the right heatmap (positive events) and less spot on the bottom part of the right heatmap (negative events). It also can be concluded that the player is able to control the boat well on the right chart since there are less spot with high speed of the negative events($T1.2$). Which means that the player are getting better on playing the game.

Focusing on events for Enemy ($T1.3$) as shown in Figure 6.3, at a glance we can see that there are more negative events and very little positive events on the first session. While on session 20, there are less negative events and more positive events. This supports the conclusion made previously that the player has gotten better performance over time. Similar notes are depicted by Figure 6.4. Here, on the left heatmap, there are more negative events which happened at faster screen speed($T1.4$). This supports the fact that first time player are usually confuse on which direction to move their body/hand to make the boat go faster or slower.

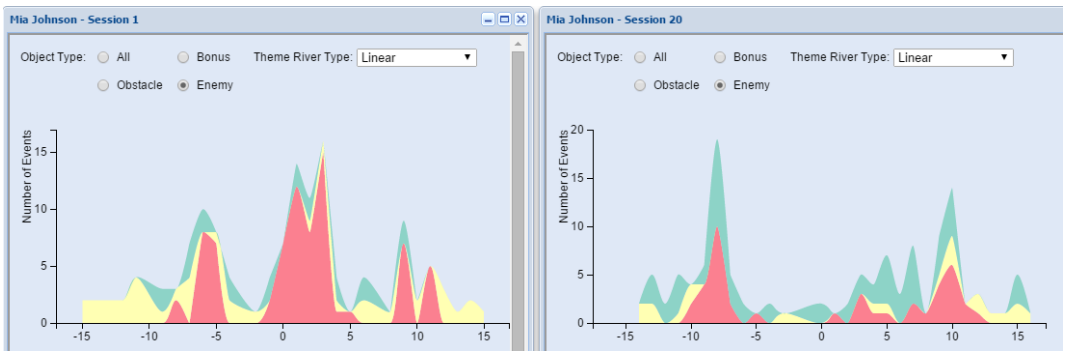


Figure 6.3: Stacked Graph comparison of Session 1 and Session 20 for Enemy played by healthy player

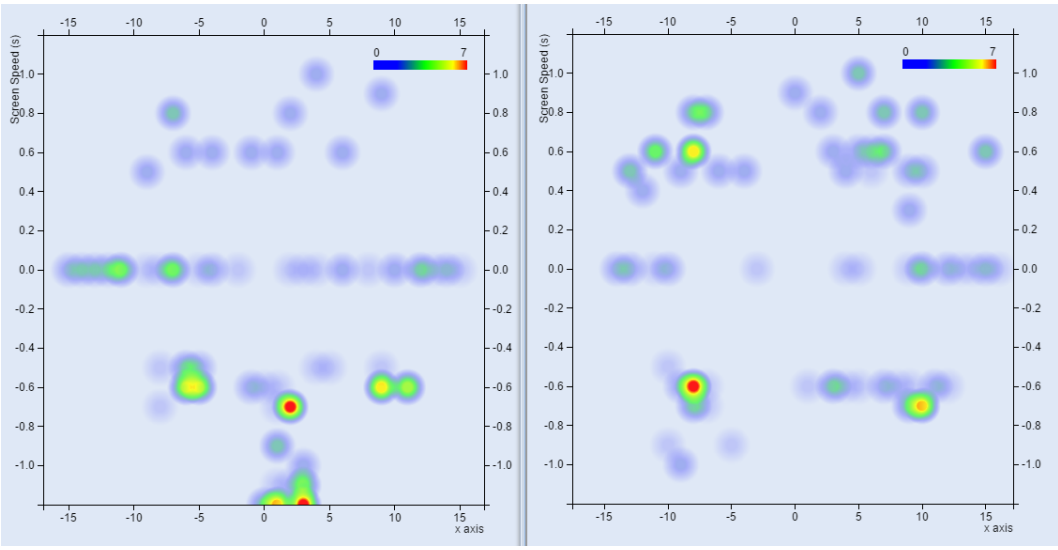


Figure 6.4: Heatmap comparison of Session 1(left) and Session 20(right) for Enemy played by healthy player

The summary of all the session can be seen in Figure 6.5. As we can see, the number of positive events are steadily increasing though fluctuate ($T2.1$). On session 19, there are very small number of negative events on all sections which indicate an improvement in the gameplay.

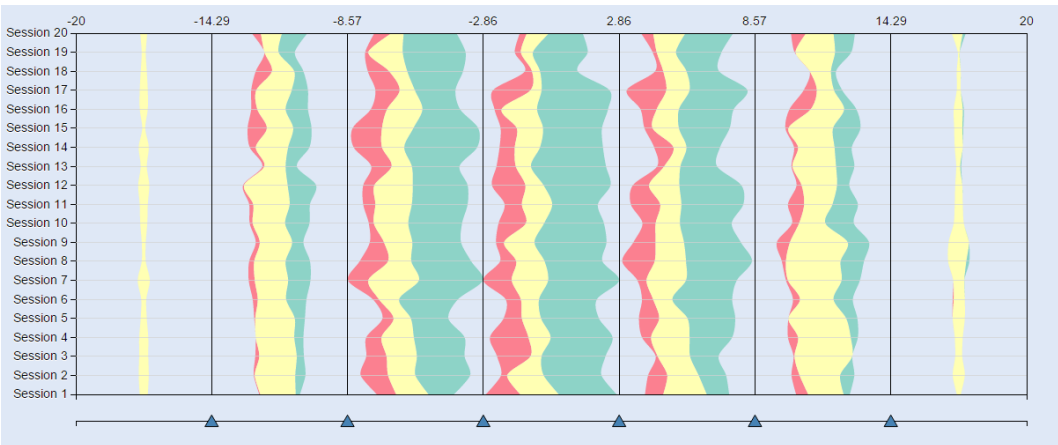


Figure 6.5: Summary Visualization by x-range, played by healthy player

In Figure 6.6, we can see the evolution of the positive events ($T2.2$). Overall, the number are steady except for a certain session. However, on the right most section there are some positive events can be found which indicates that overtime, the player move more to the right. From Figure 6.7 we can conclude that the movement are more concentrated in the middle area of the screen [($T2.3$)($T2.4$)].

In Figure 6.8, two clustered sections are presented. In Figure 6.8a left, we can see that all three sections (12-13, 13-14, 14-15) are similar in which each one of them has a little positive events on the lower half of the sections. Section 15-16 is similar to section

14-15 in term of the proportion as well as the evolution of neutral events. Therefore, these four sections are merged together forming one sections on the right($T2.5$). Similarly, in Figure 6.8b, we can see that all three sections has similar proportion of negative, neutral and positive events as well as similar evolution of each event type throughout the session. Thus forming one section on the right($T2.5$).

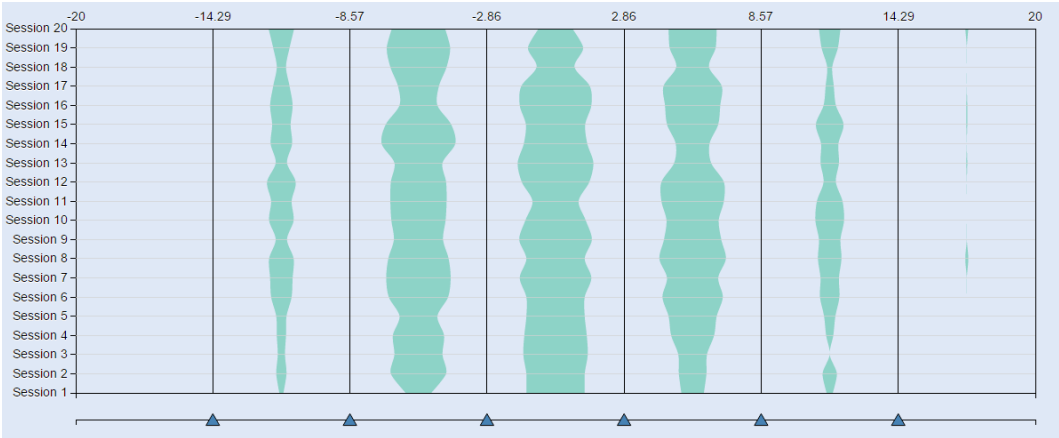


Figure 6.6: Summary Visualization by x-range, filtered for positive events, played by healthy player

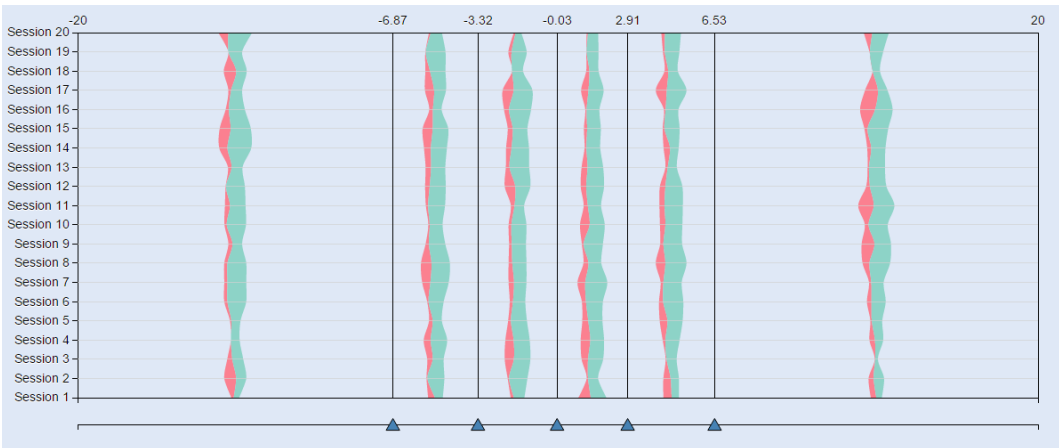
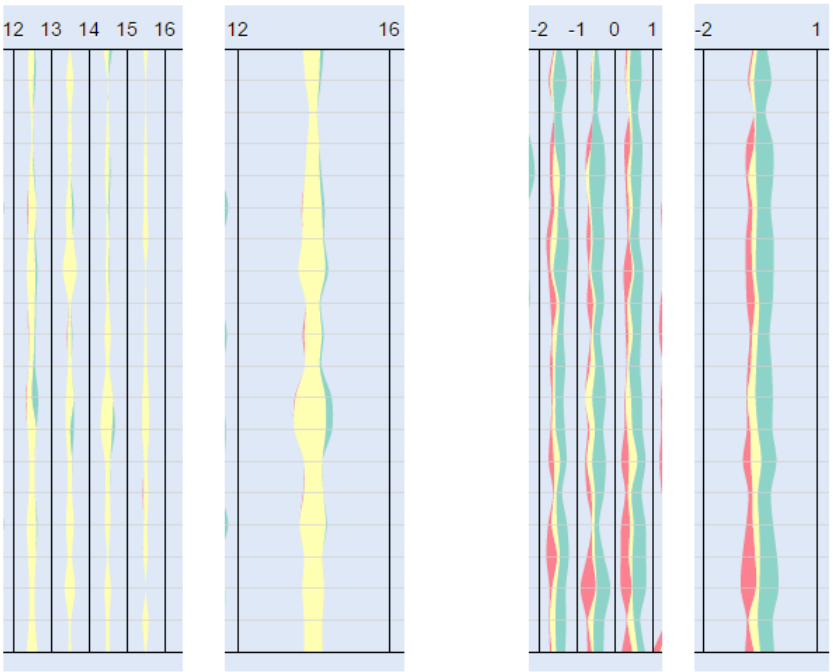


Figure 6.7: Summary Visualization by number of events, filtered for positive and negative events, played by healthy player



(a) 4 sections merged, threshold 0.22

(b) 3 sections merged, threshold 0.26

Figure 6.8: Summary Visualization by clustering, played by healthy player

6.2 Case Study 2: Patient

We have several patient data set, however most of these data set have very small number of session. Thus for the second case study, we chose data set from patient with enough sessions so that we can confirm the functionality of the developed visualization interface. Consequently, log files of game played by Patient 6 is used for this case study. There are 15 sessions which are played over the course of three weeks. However, these sessions are of two game type: HANDPOINT (6 sessions) and BODYTILT (9 session). Here, only the sessions of HANDPOINT exercise will be discussed. The game for this patient is set to show only obstacles and bonuses.

Focusing on the second session of HANDPOINT shown in Figure 6.9, it can be seen that the positive and negative events only appear on the right half of the chart indicating that the player only move to the right ($T1.1$). The yellow area are bigger compared to the green and red areas, showing that there are a lot of missed/avoided objects.

From Figure 6.10, we can see that there are many Neutral events on the left side of the screen. While Positive and Negative events are happened only on the right side with similar screen speed indicating that the player didn't change the pace of the game ($T1.2$). Figure 6.11 and 6.12 shows events related to bonus ($T1.3$)($T1.4$). Therefore, there are only Positive and Neutral events. Here, it can be seen that on the far right, there are no bonus is missed, however, on the left side all bonuses are missed. Based on these charts,

we may conclude that the player has some difficulty to move to the left. It's important to see if this is an isolated case or happened all the time. Therefore, we need to see the pattern of movement for all sessions which will be discussed shortly.

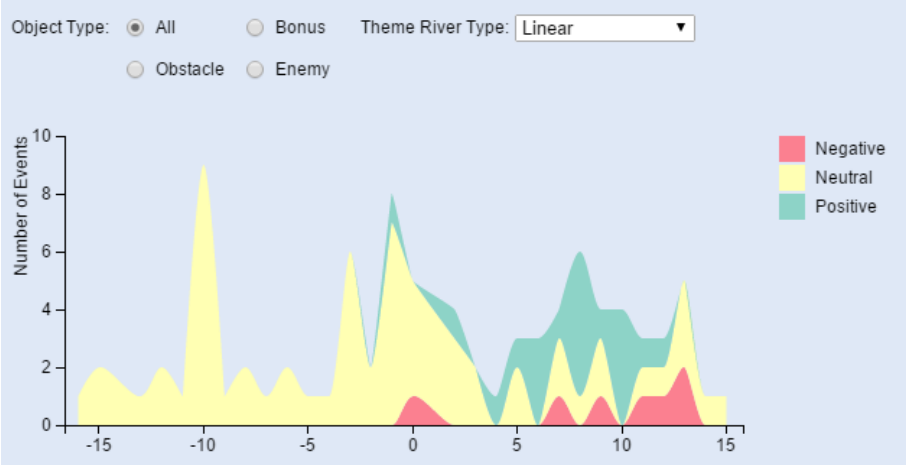


Figure 6.9: Stacked Graph of Patient 6 on second session

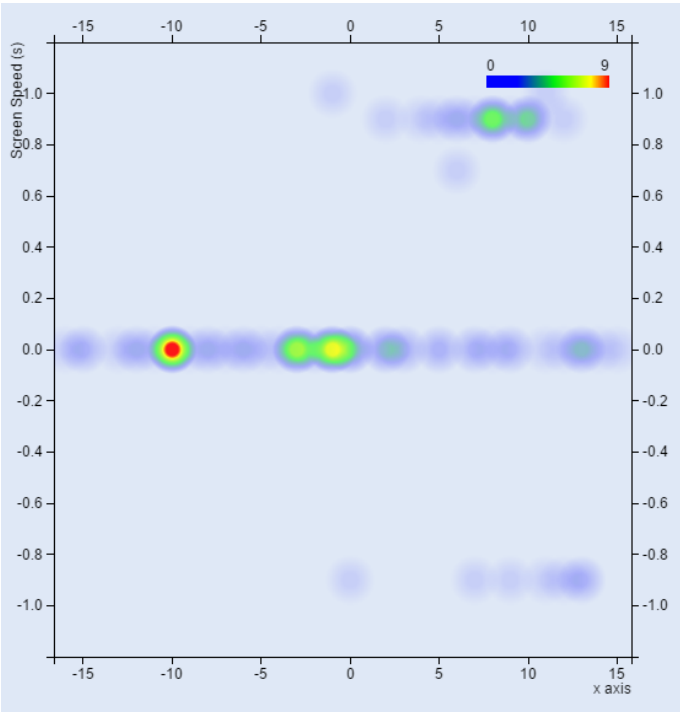


Figure 6.10: Heatmap of Patient 6 on second session

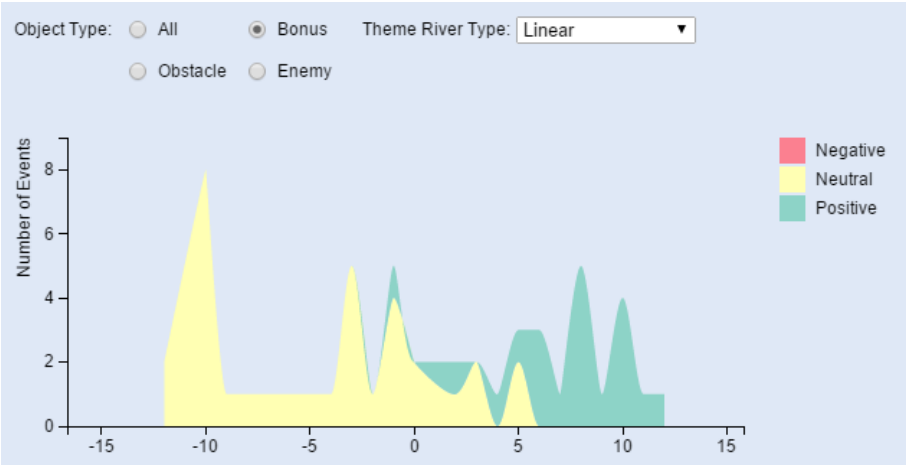


Figure 6.11: Stacked Graph of Patient 6 on second session, filtered by bonus

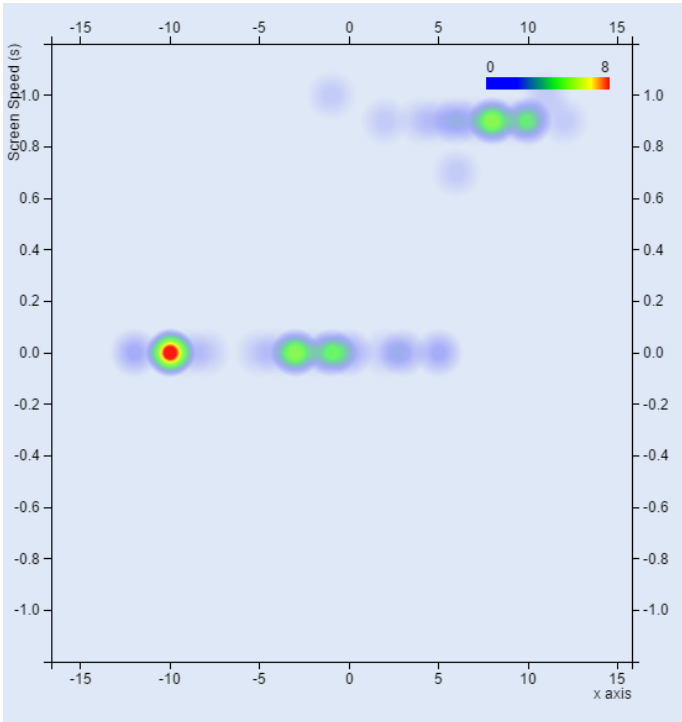


Figure 6.12: Heatmap of Patient 6 on second session, filtered by bonus

The summary visualization shown in Figure 6.13 confirmed that for 4 sessions, the player only move to the right side. If the -2.29 line is dragged to the right, the red and green area only appears from -1.96 boundary. However, from the fifth sessions, there are movements on the left side indicated by a small number of negative and positive events ($T2.1$). Figure 6.14 shows the evolution of neutral events ($T2.2$). Here, we can see that there are more neutral events on the left side compared to the right throughout all sessions. In Figure 6.15, two clustered sections are presented. In Figure 6.15a left,

we can see that both sections have similar neutral event evolution. Hence, the sections are merged together forming one sections on the right($T2.5$). Similarly, in Figure 6.15b, we can see that consecutive sections have similar neutral events evolution. Thus forming one section on the right($T2.5$). Consistent with all the previous visualization, Figure 6.16 shows that the movements are concentrated more on the right side [($T2.3$)($T2.4$)].

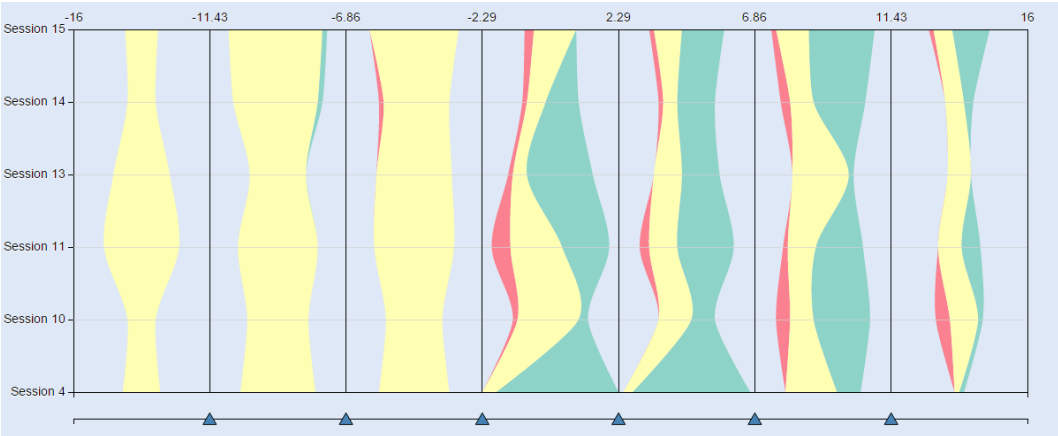


Figure 6.13: Summary visualization of Patient 6

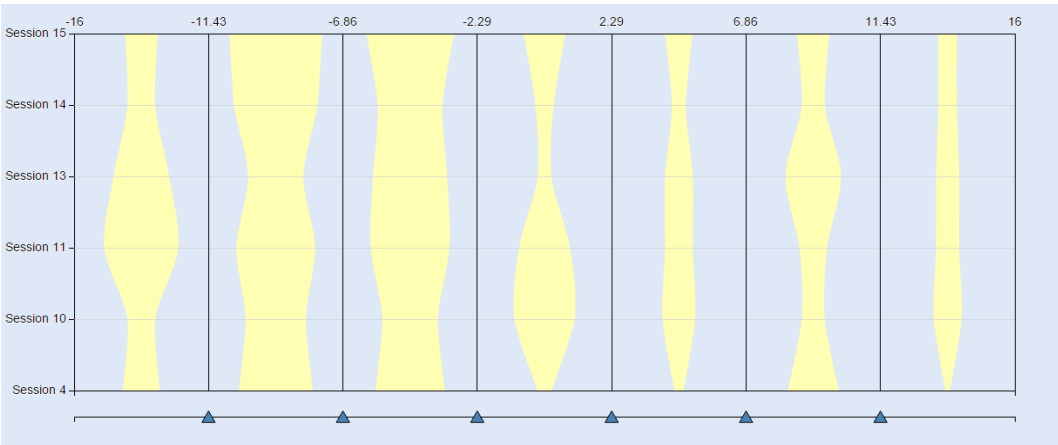
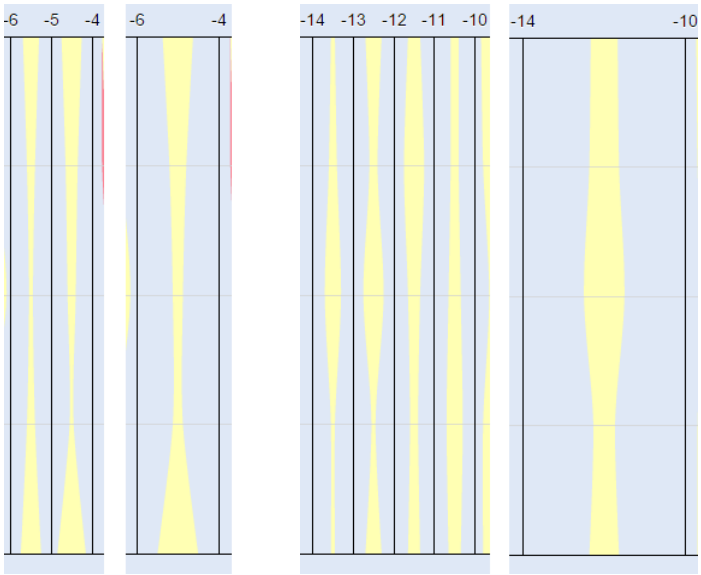


Figure 6.14: Summary visualization of Patient 6, filtered by neutral events



(a) 2 sections merged, threshold 0.05

(b) 4 sections merged, threshold 0.09

Figure 6.15: Summary visualization of Patient 6, clustered

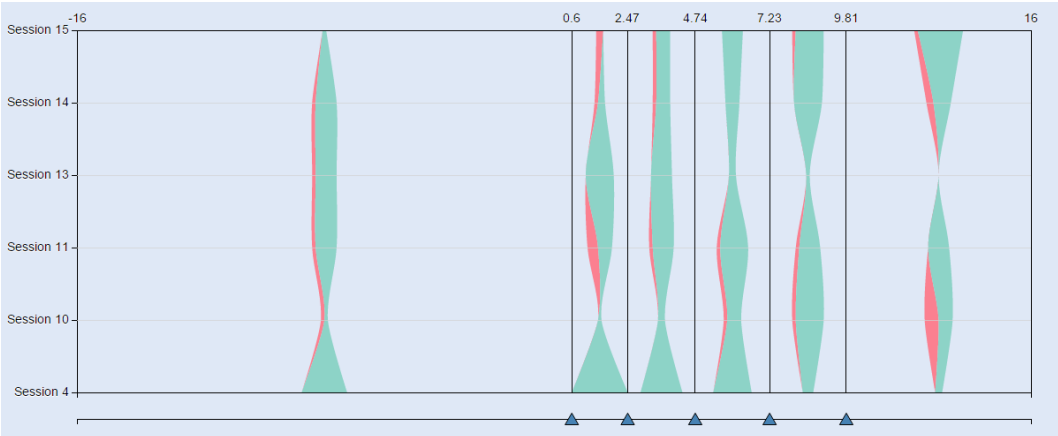


Figure 6.16: Summary visualization of Patient 6 with sections divided by number of events, filtered by positive and negative events

Movement of person with pathology are highly affected by the type of pathology. On the second case study, throughout all sessions the movements are concentrated to the right side with almost no movement on the left side. Therefore we can assume that the patient has his left side of the body affected. While for healthy person, the movement are more spread out on both left and right side with more concentration in the middle.

Chapter 7

Conclusion

In this thesis we presented a visualization interface to help healthcare professionals analyse gameplay of Hammer and Planks, a serious game which is used to rehabilitate patient with balance disorder. Player movement, represented by events happened in the game world, are visualized in two type of views: (i) *Session Visualization* which allows user to analyse movement in one session. (ii) *Summary Visualization* which allows user to analyse movement evolution over several sessions. In both visualizations, events are categorized into Positive, Neutral, and Negative to help user intuitively understand player movement throughout the session. In (i) streamgraph paradigm is used to show player movement over x-axis which in turn provides the information to which direction (left or right) the player moves more. A more detailed view is provided with heatmap showing the pace of the game when each event happened. In (ii), x-axis is divided into sections and each section represents movement evolution throughout all sessions, shown in streamgprah. Here, three types of section division are provided to help user identify interesting movement pattern: by x-range, by number of events, and by clustering. In clustering view, we proposed a clustering method based on hierarchical clustering to aggregate similar movement patterns over consecutive sections. A distance formula which consider events proportion and evolution is presented to quantify the difference of movement pattern between section.

To evaluate the interface functionality, two case studies were discussed: one from healthy person and one from patient. Both discussed cases are able to show the effectiveness of the interface in achieving the tasks defined and therefore help the healthcare professionals assessing the progress of rehabilitation.

We identified some limitations in our research that could be improved in future work. First, the distance function used in the clustering algorithm is based on euclidean distance. Although currently this distance function are able to quantify the difference in events proportion and evolution, it will be interesting to investigate other distance function and to see if it can improve the clustering. Another limitation is the data set used in the case studies isn't accompanied with pathology information. It will be interesting to study different type of pathology and it's movement pattern. In the future, these data can be used to improve the game by proposing game setting based on the type and severity of pathology. Lastly, current interface only explore log data related to events. Log data related to skeleton movement of the players throughout the game hasn't been explore. For future work, an interface visualizing the body movement and identifying different kind of movement quality and quantity (repetition, smoothness, accuracy, etc.) can help healthcare professional to improve the quality of rehabilitation process.

Bibliography

[1] Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. *Visualization of Time-Oriented Data.* Human-Computer Interaction Series. Springer, 2011. (Cited on page 10.)

[2] Gennady L. Andrienko, Natalia V. Andrienko, Peter Bak, Daniel A. Keim, and Stefan Wrobel. *Visual Analytics of Movement.* Springer, 2013. (Cited on pages 3, 12 and 13.)

[3] Natalia V. Andrienko and Gennady L. Andrienko. Spatial generalization and aggregation of massive movement data. *IEEE Trans. Vis. Comput. Graph.*, 17(2):205–219, 2011. (Cited on pages 10 and 13.)

[4] Jürgen Bernard, Nils Wilhelm, Björn Krüger, Thorsten May, Tobias Schreck, and Jörn Kohlhammer. Motionexplorer: Exploratory search in human motion capture data based on hierarchical aggregation. *IEEE Transactions on Visualization and Computer Graphics (Proc. VAST)*, December 2013. (Cited on pages 3, 12, 14 and 15.)

[5] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, December 2011. (Cited on page 16.)

[6] V. Brezinka. Treasure hunt - a serious game to support psychotherapeutic treatment of children. *Studies in Health Technology and Informatics*, 136:71–76, 2008. (Cited on page 1.)

[7] Lee Byron and Martin Wattenberg. Stacked graphs – geometry & aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1245–1252, November 2008. (Cited on page 24.)

[8] Jiri Chmelik and Jiri Sochor. Body motion visualization in virtual environment. In *Proceedings of the 30TH CONFERENCE ON GEOMETRY AND GRAPHICS*, pages 119–124, Prague, 2010. MATFYZPRESS. (Cited on page 14.)

[9] Ines Di Loreto, Benoit Lange, Antoine Seilles, William Dyce, and Sebastien Andary. Game design for all: The example of hammer and planks. *Serious Games Development and Applications*, 8101:70–75, 2013. (Cited on pages 1 and 19.)

[10] Alan Dix and Geoffrey Ellis. Starting simple: Adding value to static visualisation through simple interaction. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '98, pages 124–134, New York, NY, USA, 1998. ACM. (Cited on page 25.)

[11] Michael Friendly. The history of the cluster heat map. *The American Statistician*, 2009. (Cited on page 25.)

[12] Peter Gatalsky, Natalia Andrienko, and Gennady Andrienko. Interactive analysis of event data using space-time cube. In *Proceedings of the Information Visualisation, Eighth International Conference*, IV '04, pages 145–152, Washington, DC, USA, 2004. IEEE Computer Society. (Cited on page 10.)

[13] D.M Gavrila. The visual analysis of human movement. *Comput. Vis. Image Underst.*, 73(1):82–98, January 1999. (Cited on page 14.)

[14] D. Green and P. Wilson. Use of virtual reality in rehabilitation of movement in children with hemiplegia - a multiple case study evaluation. *Disability and Rehabilitation*, 34:593–604, 2012. (Cited on pages 1 and 8.)

[15] Susan Havre, Beth Hetzler, and Lucy Nowell. Themeriver: Visualizing theme changes over time. In *Proceedings of the IEEE Symposium on Information Vizualization 2000*, INFOVIS '00, pages 115–, Washington, DC, USA, 2000. IEEE Computer Society. (Cited on pages 10 and 24.)

[16] Yaya Heryadi, M. Ivan Fanany, and A.M. Arymurthy. Grammar of dance gesture from bali traditional dance. *International Journal of Computer Science Issues*, 9:144–149, 2012. (Cited on page 14.)

[17] S. Huron, R. Vuillemot, and J.-D Fekete. Visual sedimentation. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2446–2455, 2013. (Cited on page 27.)

[18] Oded Maimon and Lior Rokach. *Data Mining and Knowledge Discovery Handbook*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. (Cited on page 22.)

[19] Tamara Munzner. A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, November 2009. (Cited on page 3.)

[20] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, and Sung Nok Chiu. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Series in Probability and Statistics. John Wiley and Sons, Inc., 2nd ed. edition, 2000. (Cited on page 13.)

[21] O. Patsadu, C. Nukoolkit, and B. Watanapa. Human gesture recognition using kinect camera. In *Computer Science and Software Engineering (JCSSE), 2012 International Joint Conference on*, pages 28–32, May 2012. (Cited on page 14.)

[22] Abdur Rahman. Multisensor serious game-based therapy environment for hemiplegic patients. *International Journal of Distributed Sensor Networks*, 2015:12, 2015. (Cited on pages v, 1, 5, 8 and 9.)

[23] Md.Abdur Rahman. Multimedia environment toward analyzing and visualizing live kinematic data for children with hemiplegia. *Multimedia Tools and Applications*, 74(15):5463–5487, 2015. (Cited on pages v, 8 and 9.)

[24] Michalis Raptis, Darko Kirovski, and Hugues Hoppe. Real-time classification of dance gestures from skeleton animation. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '11, pages 147–156, New York, NY, USA, 2011. ACM. (Cited on page 14.)

[25] Yuko Tashiro and Tsuyoshi Saitoh. A study on motion visualization system using motion capture data. In *ICAT*, pages 314–315. IEEE Computer Society, 2007. (Cited on page 14.)

[26] Christian Tominski, Petra Schulze-Wollgast, and Heidrun Schumann. 3d information visualization for time dependent data on maps. In *Proceedings of the Ninth International Conference on Information Visualisation*, IV '05, pages 175–181, Washington, DC, USA, 2005. IEEE Computer Society. (Cited on page 10.)

[27] Ulanbek D. Turdukulov, Menno-Jan Kraak, and Connie A. Blok. Visual analytics: Designing a visual environment for exploration of time series of remote sensing data: In search for convective clouds. *Comput. Graph.*, 31(3):370–379, June 2007. (Cited on page 10.)