



HAL
open science

Visual Analysis of Body Movement in Serious Games for Healthcare

Oky Purwantiningsih, Arnaud Sallaberry, Sebastien Andary, Antoine Seilles,
Jérôme Azé

► **To cite this version:**

Oky Purwantiningsih, Arnaud Sallaberry, Sebastien Andary, Antoine Seilles, Jérôme Azé. Visual Analysis of Body Movement in Serious Games for Healthcare. PacificVis: Pacific Visualization Symposium, Apr 2016, Taipei, Taiwan. pp.229-233, 10.1109/PACIFICVIS.2016.7465276 . lirmm-01304601

HAL Id: lirmm-01304601

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-01304601>

Submitted on 20 Apr 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Visual Analysis of Body Movement in Serious Games for Healthcare

Oky Purwantiningsih*
LIRMM - Université Paul Valéry
Montpellier, France

Arnaud Sallaberry*
LIRMM - Université Paul Valéry
Montpellier, France

Sébastien Andary†
NaturalPad
Montpellier, France

Antoine Seilles†
NaturalPad
Montpellier, France

Jérôme Azé*
LIRMM - Université de Montpellier
France

ABSTRACT

The advancement of motion sensing input devices has enabled the collection of multivariate time-series body movement data. Analyzing such type of data is challenging due to the large amount of data and the task of mining for interesting temporal movement patterns. To address this problem, we propose an interface to visualize and analyze body movement data. This visualization enables users to navigate and explore the evolution of movement over time for different movement areas. We also propose a clustering method based on hierarchical clustering to group similar movement patterns. The proposed visualization is illustrated with a case study which demonstrates the ability of the interface to analyze body movements.

Index Terms: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—Clustering; H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical User Interfaces

1 INTRODUCTION

Visualizing movement data has been proved to be challenging due to the large volume of data [2]. This is also the case with visualizing body movement data which is usually multivariate and time oriented. The challenge in visualizing time-series data lies in finding the temporal trends or pattern in the data as has been discussed extensively in [1]. A few visualization techniques focus on time-series body movement data (e.g. [3, 4, 6, 9, 12, 13]). In healthcare, such data are generated from serious games which are used for patients' rehabilitation and training. These games use motion sensing input device technologies such as Kinect and Wii Balance Board to capture players' movements. Data generated from the game are then analyzed by healthcare professionals to make a diagnosis of patients' progress. However, analyzing large amounts of data manually is long, difficult and tedious. Therefore, a visualization interface is needed. The challenge lies in determining how to visualize such data, so that healthcare professionals can evaluate progress and make decisions for the next rehabilitation step.

Currently, serious games provide some charts which visualize the player's body movements with respect to the horizontal axis and vertical axis [10, 11]. However, the information that can be gathered from the visualization is not enough for the healthcare professional to establish an informed diagnosis. It is not immediately clear how often the player moves to the right or to the left. It is also not clear which type of events (i.e., avoiding an enemy, catching the bonuses) are associated with the movements, which is crucial since the therapist needs to know if the player is able to move his body in the necessary direction. Most importantly, they need to know the

evolution of the players' body movements over time to see if they have developed strategies to achieve the required movement for the rehabilitation.

To address these problems, we propose an interactive visualization allowing healthcare professionals to analyze the evolution of players' body movements throughout several sessions of a serious game. We also propose an approach to easily navigate and highlight movement patterns. It is based on a clustering algorithm that identifies similar movement patterns. New distance functions are defined to quantify similarity between movement patterns, taking into account both movement evolution and proportion of movement frequency.

2 DOMAIN PROBLEM CHARACTERIZATION

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

For the research, we explore data from a game called Hammer and Planks which is developed by NaturalPad¹. This game is designed so that the player has to move his body right, left, forward and backward, in order to recover full mobility. It is a vertical shooter game. The game world is in a 2D environment vertically scrolling from top to bottom in which a player navigates a ship from left to right through moving his body. While navigating the ship to collect driftwood/planks (bonuses), the player also needs to kill enemies and to avoid reefs scattered all over the ocean (obstacles).

It is possible to play Hammer and Planks in two different ways: (i) BodyTilt: The player puts both hands on his/her hips and moves the upper body (from the waist up) to the right, to the left, forward or backward to navigate the boat. (ii) HandPoint: the player lifts one of his/her forearms in front of the body with the palm facing forward. Navigating the boat can be done by moving the forearm to the right, to the left, forward, or backward. Basically, there are two types of movement directions: a horizontal movement moves the boat horizontally and a vertical movement changes the boat speed. Body movements are recorded with a Kinect. Before each session, the healthcare professionals 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 sessions.

A traditional therapy routine usually involves the therapist ordering a patient to perform several movements repetitively [11]. By the end of the session, the therapist will analyze 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

*e-mail: firstname.lastname@lirmm.fr

†e-mail: firstname@naturalpad.fr

¹www.naturalpad.fr

this analysis, the therapist will then configure a new routine to further the patient’s progress, if needed. To make his analysis, he/she needs to answer questions such as “has the patient improved in the game overtime?”, “has he/she improved in a certain area overtime?” or “is there an area where the patient acts the same?”. Here, we propose a tool to help him perform his analysis. In collaboration with experts, we have identified five tasks that need to be performed with the tool.

Before listing these tasks, we need to further describe the data generated from a game session. The gameplay contains information about the player, the game setting, the player’s movements throughout the game and every event that occurred 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 away. The movement of the boat is considered as the movement of the player’s body. We will analyze this body movement data. The following are the tasks we identified in collaboration with experts: **(T1)** Visualize, navigate and be able to compare the evolution of the number of events throughout all sessions within a certain x -area, **(T2)** Select and visualize the number of events of a certain event type in a certain x -area throughout all sessions, **(T3)** Visualize, navigate and be able to compare the distribution of a certain number of events over x -area among all sessions, **(T4)** Select and visualize the distribution of certain number of events over x -area for a certain event type throughout all sessions, **(T5)** Extract and visualize similar pattern of number of events evolution throughout all sessions over a certain x -area.

3 DATA ABSTRACTION

In this section, we discuss the design of the 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 clustering algorithm proposed to be used in the visualization will be discussed.

3.1 Game Event Structure

The goal of Hammer and Planks game is to kill all the enemies while avoiding any attack from the enemies and obstacles [5]. Along the way, players can also catch bonuses to increase their score. Based on these, we identified three different objects within the game: Bonus, Obstacle and Enemy. For each of these objects, there are certain events associated. Each event has an event type. In total, there are 7 event types: (1) Catch: when a bonus is caught (2) Miss: when a bonus is missed or a player’s attack on an enemy is missed (3) Dodge: when an obstacle is avoided (4) Collision: when the player’s boat collides with an enemy or an obstacle (5) Kill: when an enemy is destroyed by the player’s boat (6) Hit: when the player’s attack hits an enemy (7) Hurt: when the enemy’s attack hits the 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 1.

Event Type	Bonus	Obstacle	Enemy
Positive	(1)		(5), (6)
Neutral	(2)	(3)	(2)
Negative		(4)	(4), (7)

Table 1: Event Type grouping

3.2 Clustering

In understanding the evolution of movements among different sessions over x -area, it is interesting to see the common evolution of different sections of the x -area **(T5)**. The idea is to cluster similar distribution of movements (which is represented by events in

the game) so that consecutive sections which have similar evolutions are 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 sections will be presented. Then, a clustering algorithm based on hierarchical clustering which incorporates this distance function will be explained.

3.2.1 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 that 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 a 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 $n_{ses} \times 3$ dimensions. For instance, the following matrices represent 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. In 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 second session he got 40 negative, 10 neutral, and 30 negative events, etc.

Since the idea was to aggregate consecutive sections with similar movement patterns, a function to quantify the similarity between two sections is needed. Here, we focus on patterns described by triplets of values as described in the example. In this case, we quantify the similarity by defining a distance function. Thus, a distance equal to 0 indicates that both sections are similar, while a distance equal to 1 indicates that both sections are different. We identified that two types of distances need to be considered: (i) how different are the sections in terms of the proportion of the event types in each section (ii) how different are the sections in terms of the evolution of each event type throughout the sessions. For instance, with the (i) approach, s_2 and s_3 are similar because the triplets of each session 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 sections (s_1, s_2) of S , distance $d(s_1, s_2)$ 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 (NED) 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 by 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 the normalized value of s_1 and s_2 . Since the first distance is to see the difference of event type proportion in each section, the values are normalized by the maximum value of each session in each section. In this case, two consecutive sections with different number of events but similar proportions of event types will have a distance equal to 0. For example, the matrices presented previously can be normalized into the following matrices:

$$s'_1 = \begin{bmatrix} \frac{1}{2} & 1 & \frac{3}{10} \\ 1 & \frac{1}{4} & \frac{9}{10} \end{bmatrix}, s'_2 = \begin{bmatrix} \frac{1}{2} & 1 & \frac{1}{4} \\ 1 & \frac{1}{4} & \frac{3}{4} \end{bmatrix}, s'_3 = \begin{bmatrix} \frac{1}{2} & 1 & \frac{1}{4} \\ 1 & \frac{1}{4} & \frac{3}{4} \end{bmatrix}$$

Thus, distance $f(s_1, s_2) = 0.06$, and distance $f(s_2, s_3) = 0$.

The second distance function $g(s_1, s_2)$ is based on a Normalized Euclidean Distance of the same event type j between different sections. 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 values defined: (i) if there is no event in both event type pairs, the distance is 0 (ii) if there is at least one event in one section and no events in the other section, the distance is 1 (iii) if both event type pairs have 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. Besides, when only one section is empty while the other has some events, it will be considered as different sessions 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 by maximum distance $\sqrt{|s_1|}$.

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

and for $j \in \{0, 1, 2\}$,

$$g_j(s_1, s_2) = \begin{cases} 0 & \text{if } s_k[i][j] = 0 \text{ for each } 0 \leq j < |s_k| \text{ and } k \in \{1, 2\} \\ 1 & \text{if } s_k[i][j] = 0 \text{ for each } 0 \leq j < |s_k| \\ & \text{and } \exists s_p[i][j] \neq 0, k, p \in \{1, 2\}, k \neq p \\ \frac{\sqrt{\sum_{i=0}^{|s_1|} (s''_1[i][j] - s''_2[i][j])^2}}{\sqrt{|s_1|}} & \text{otherwise} \end{cases}$$

Here, s''_1 and s''_2 represent the 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 by 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 $g(s_1, s_2) = 0$ and distance $g(s_2, s_3) = 0.06$.

3.2.2 Clustering Algorithm

The clustering algorithm follows Hierarchical Clustering algorithm [8], which is a clustering analysis method used to build a 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, the distances of consecutive sections are calculated. Two consecutive sections with a distance below a specified threshold are then merged. Thus resulting in a new set of sections. Then, the distance of each pair of 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 falling below the threshold.

4 VISUAL MAPPINGS AND INTERACTIVE FUNCTIONALITIES

This section presents the visualization and interaction methods developed based on the visualization requirements and data abstraction discussed in Sections 2 and 3. Since users are interested in the evolution of movement over a certain x -area, the visualization is 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 detail.

4.1 Visualization by range of x -area

To fulfill (T1), a streamgraph metaphor is chosen to show the evolution of movement (represented by events) over time. Time is often displayed on x axis, but here, x axis represents 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 [7]. Here, sessions represent time with the earliest one shown at the bottom and the latest one shown at 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 the session number and the event category. The number of events within this group is then presented in the vertical streamgraph layout. Each event category is represented as an area with different colors: red (Negative), Yellow (Neutral), and Green (Positive). For each section, it is possible to see which session has the most or least number of events by comparing the total length of all event categories in one session (see Figure 1.a).

4.2 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 (T3). Therefore, based on the distribution of positive and negative events, one section in the chart may have bigger x range than the other section. Only by comparing the size of the sections, it's possible to know in which area most of the events are concentrated. Figure 1.c shows that the events are more concentrated in the right area of the screen.

4.3 Visualization by clustering

This third type of visualization fulfills (T4). Initially the chart is divided into sections with range equals to an x axis unit. Depending on the threshold value inputted by the user, consecutive sections with a distance below the threshold will be merged. This process is repeated until there is no pair of sections with a distance below the threshold. The input threshold ranges from 0 to 1. Thus, when user inputted threshold = 0, none of the sections will be merged. Besides, when user inputted threshold = 1, all of the sections will be merged. Figure 1.b shows when user input threshold = 0, while Figure 1.d shows threshold = 0.35. As we can see, there are some sections which are merged, indicated by a bigger section size.

4.4 Interaction Technique

On top of the chart, an interaction bar is provided where the user can interact with and change some variables in the chart. In the first panel of the interaction bar (Figure 1.e), three sliders are provided: (i) a slider to input the number of slices so that each section will have the same x range (ii) a slider to input the number of slices so that each section will have the same total number of positive and negative events (iii) a slider to input a threshold so that each section will have a cluster of similar movement evolution. When a user uses slider (i) and (ii), the value on the slider defines the number of sections on the chart. For (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 the total

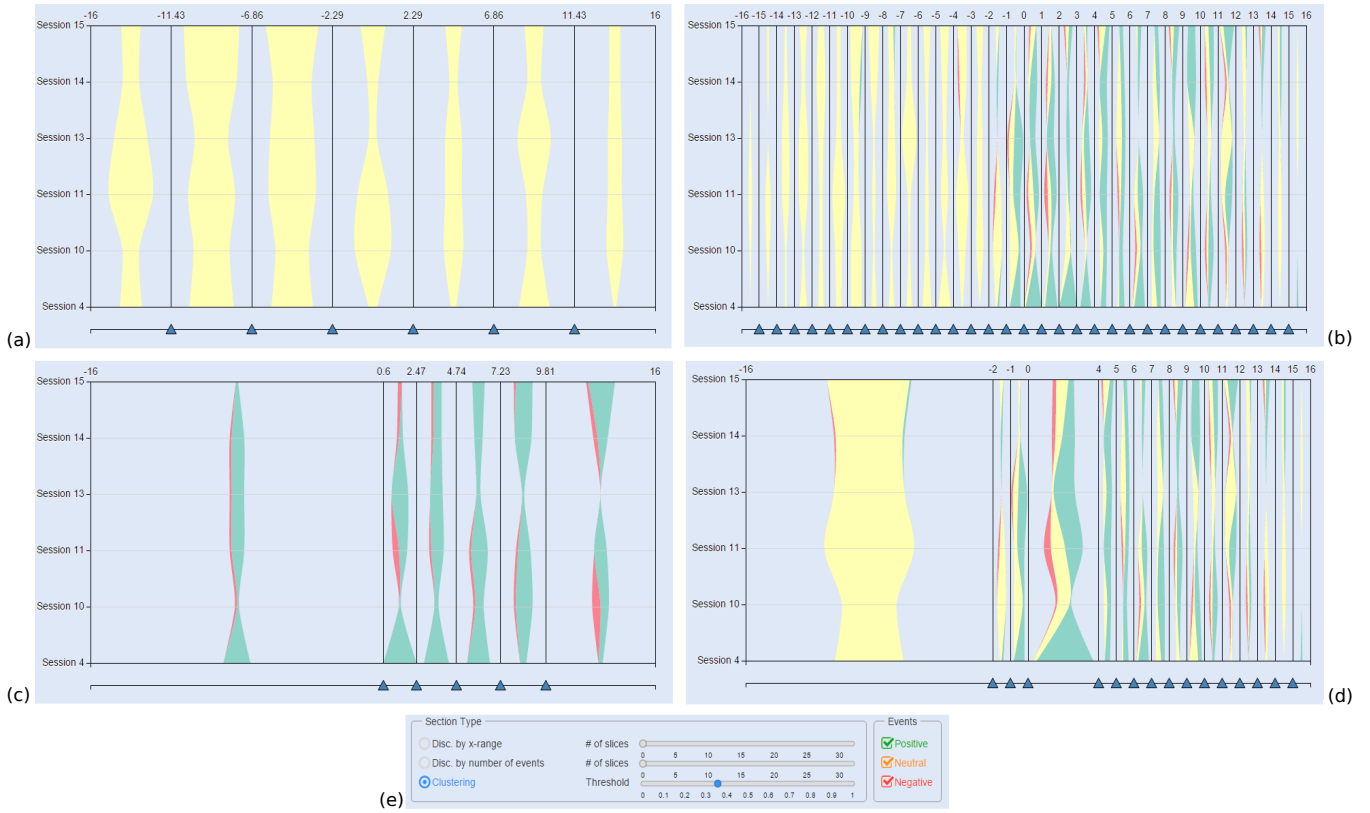


Figure 1: Visualization divided by x range filtered with Neutral events (a), clustered with threshold=0 (b), divided by total number of Positive and Negative events (c), clustered with threshold=0.35 (d), and navigation bar (e)

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, the user has the options to choose which event type to show on the chart. This fulfills requirements (T2) and (T4). By default all event types will be shown.

Once the chart is generated, the 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 to change the range of its neighboring sections (T1, T3). 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 a different x range. Therefore, the user may be able to determine areas in which a certain type of event appears. 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 the user to know the distribution of event type within a section.

5 CASE STUDY

To evaluate the visualization functionality developed in this article, a case study will be presented. In this case study we use data set from games played by patients. These data sets are collected by NaturalPad. Unfortunately, it is not possible to get information concerning the patients' pathology due to confidentiality reasons. This case study is based on a patient dataset containing 15 sessions which are played over the course of 3 weeks. These sessions are of two game types: HandPoint (6 sessions) and BodyTilt (9 sessions). Here, only the 6 sessions of HandPoint exercise will be discussed.

The visualizations in Figure 1 show that the patient mostly moved to the right side. Figure 1.a shows the evolution of neutral

events T2. Here, we can see that there are more neutral events on the left side compared to the right throughout all sessions. In Figure 1.b, we can see that four sections between 0 and 4 have similar neutral event evolutions. Therefore, the sections are merged together forming one section T5 (see Figure 1.d). Consistent with all the previous visualizations, Figure 1.c shows that the movements are concentrated more on the right side [T3, T5]. In particular, for sessions 4, 10, 11 and 13, the patient did not move to the left side (only neutral events). From session 14 to 15, movements on the left side appear indicated by a small number of negative and positive events T1. We can assume that the patient has his left side of the body affected (hemi-neglect), but he/she starts recovering. The therapist is able to detect this evolution and proposes new sessions or exercises adapted to the patient's state.

6 CONCLUSION

In this paper we presented a visualization interface to help health-care professionals analyze gameplay of Hammer and Planks, a serious game which is used to rehabilitate patients with balance disorder. In order to identify movement patterns of interest, we propose new metrics for the evaluation of distances between sequences of patient movements. The purpose is to aggregate similar movement patterns over consecutive sections. We also propose an interface which allows the therapist to interactively explore the patterns by dynamically changing aggregation parameters. Therapists can then discover the movement patterns.

For future work, we plan to perform evaluations to validate the metrics and the visualization. We also want to develop new visualization features and to study how these techniques could be extended to other kinds of data.

REFERENCES

- [1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Human-Computer Interaction Series. Springer, 2011.
- [2] G. L. Andrienko, N. V. Andrienko, P. Bak, D. A. Keim, and S. Wrobel. *Visual Analytics of Movement*. Springer, 2013.
- [3] J. Bernard, N. Wilhelm, B. Krüger, T. May, T. Schreck, and J. Kohlhammer. MotionExplorer: Exploratory search in human motion capture data based on hierarchical aggregation. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2257–2266, 2013.
- [4] J. Chmelik and J. Sochor. Body motion visualization in virtual environment. In *Proceedings of the Conference on Geometry and Graphics*, pages 119–124, 2010.
- [5] I. Di Loreto, B. Lange, A. Seilles, W. Dyce, and S. Andary. Game design for all: The example of hammer and planks. *Serious Games Development and Applications*, 8101:70–75, 2013.
- [6] Y. Heryadi, M. I. Fanany, and A. Arymurthy. Grammar of dance gesture from bali traditional dance. *International Journal of Computer Science Issues*, 9:144–149, 2012.
- [7] S. Huron, R. Vuillemot, and J.-D. Fekete. Visual sedimentation. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2446–2455, 2013.
- [8] O. Maimon and L. Rokach. *Data Mining and Knowledge Discovery Handbook*. Springer-Verlag New York, Inc., 2005.
- [9] O. Patsadu, C. Nukoolkit, and B. Watanapa. Human gesture recognition using kinect camera. In *Proceedings of the International Joint Conference on Computer Science and Software Engineering*, pages 28–32, 2012.
- [10] A. Rahman. Multimedia environment toward analyzing and visualizing live kinematic data for children with hemiplegia. *Multimedia Tools and Applications*, 74(15):5463–5487, 2015.
- [11] A. Rahman. Multisensor serious game-based therapy environment for hemiplegic patients. *International Journal of Distributed Sensor Networks*, 2015:12, 2015.
- [12] M. Raptis, D. Kirovski, and H. Hoppe. Real-time classification of dance gestures from skeleton animation. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 147–156, 2011.
- [13] Y. Tashiro and T. Saitoh. A study on motion visualization system using motion capture data. In *Proceedings of the International Conference on Artificial Reality and Telexistence*, pages 314–315, 2007.