

A GIS-based Serious Game Recommender for Online Physical Therapy

Imad Afyouni
GIS Technology Innovation Center
Umm Al-Qura University, KSA
iafyouni@gistic.org

Akhlaq Ahmad
International Islamic University Malaysia
Engineering College, Umm Al-Qura University, KSA
aajee@uqu.edu.sa

Faizan Ur Rehman
GIS Technology Innovation Center
Umm Al-Qura University, KSA
fsrehman@uqu.edu.sa

Mohamed Abdur Rahman
Department of Computer Science
Umm Al-Qura University, KSA
marahman@uqu.edu.sa

Ahmad Qamar
College of Computer and Information Systems
Umm Al-Qura University, KSA
amqamar@uqu.edu.sa

Saleh Basalamah
GIS Technology Innovation Center
Umm Al-Qura University, KSA
smbasalamah@uqu.edu.sa

ABSTRACT

As human-centered interactive technologies, serious games are getting popularity in a variety of fields such as training simulations, health, national defense, and education. To build the best learning experience when designing a serious game, a system requires the integration of accurate spatio-temporal information. Also, there is an increasing need for intelligent medical technologies, which enable patients to live independently at home. This paper introduces a novel e-Health framework that leverages GIS-based serious games for people with disabilities. This framework consists of a spatial map-browsing environment augmented with our newly introduced multi-sensory Natural User Interface. We propose a comprehensive architecture that includes a sensory data manager, a storage layer, an information processing and computational intelligence layer, and a user interface layer. Detailed mathematical modeling as well as mapping methodology to convert different therapy-based hand-gestures into navigational movements within the serious game environment are also presented. Moreover, an Intelligent Game Recommender has been developed for generating optimized navigational routes based on therapeutic gestures. Motion data is stored in a repository throughout the different sessions for offline replaying and advanced analysis; and different indicators are displayed in a live manner. This framework has been tested with Nokia, Google maps, ESRI map, and other maps whereby a subject can visualize and browse the 2D and 3D map of the world through therapy-based gestures. To the best of our knowledge, this is the first GIS-based game recommender framework for online physical therapy. The prototype has been deployed to a disability center. The obtained results and feedback from therapists and patients are very encouraging.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org. Copyright is held by the owner/author(s). Publication rights licensed to ACM.

HealthGIS'14, November 4-7, 2014 - Dallas/Fort Worth, TX, USA.

ACM 978-1-4503-3136-4/14/11...\$15.00

<http://dx.doi.org/10.1145/2676629.2676634>

Categories and Subject Descriptors

H.2.8 [Database Applications]: Spatial databases and GIS;
H.5.1 [Multimedia Information Systems]

General Terms

Design, Human Factors

Keywords

Serious Games, GIS, e-Therapy, Recommendation, Kinect, Leap

1. INTRODUCTION

Research has shown that proper therapies for disabilities that might affect human body such as Hemiplegia¹ can help affected patients to quickly return to normal life [12]. Traditionally, an expert attaches a goniometer to the patient's body to measure the angle of motion for a given joint. However, this method is time consuming and requires supervision of experienced therapists. The lack of trained professionals with respect to the growing number of patients has prompted the development of automated home-based e-health systems for physical rehabilitation [11]. Previous studies suggest gathering kinematic therapy data and generating post-therapy analytics reports help the therapist and caregiver track the condition of the affected areas of the patient [15]. Some of the kinematic metrics that reflect the underlying gathered data include: positions, velocities, angular velocities, acceleration and angular acceleration. Although therapist-assisted hand rehabilitation is commonplace, home-based therapy is gaining popularity these days because it is flexible, relaxing and easier to repeat alone or with the presence of a caregiver.

To help in home-based hand-Therapy, various computing platform such as Microsoft Kinect [2], Microsoft Digits, and Leap motion, to name a few, have recently emerged that help in identifying physiological and gait parameters from a hand therapy session in real-time. Although Microsoft Kinect is a popular platform for full body gesture recognition [6], the current version of Kinect cannot track subtle finger and forearm rotational and angular movement such as pronation and supination or forearm, circumduction of thumb or fingers,

¹Disability that affects the body and joint movements in one half of the body

to name a few. Recently, a new 3D motion-based gesture control device has been introduced, called LEAP. The sensing range and very high resolution makes it the best candidate among the state of the art for an in-home hand-Therapy measurement device [13].

To add to the physical rehabilitation, serious games are getting popularity in a variety of fields such as training simulations, health, national defense, and education [8, 9]. Research has shown the effectiveness of such games in improving the learning skills of subjects [10]. Particularly, serious games have become quite popular in recent years for use in mental and physical rehabilitation. In [7], the authors have reported a significant potential for building serious games for treating patients with physical disability. To build the best learning experience when designing a serious game, an authoring system requires the integration of accurate spatio-temporal information. For instance, games such as driving a car near some personalized points of interest, or controlling a flying kite over your favorite city always sound motivating and exciting, especially, for children [1, 4, 5]. However, little work has been done in GIS domain with the aim of developing serious games [3].

This paper presents a GIS- based serious game environment integrated within an e-health framework that offers several advantages over state of the art systems for therapy design and monitoring. First of all, this framework consists of a spatio-temporal map-browsing environment augmented with our newly introduced multi-sensory Natural User Interface. The GIS-based game provides a deeply immersive map navigation experience to the users. The map navigation activity is performed using the Leap Motion controller, which is used to detect 18 different primitive hand therapies [14]. The map browsing activity can be used for training simulations, for educational purposes (e.g., city simulation, virtual earth) or for pure entertainment. Another sensor called Microsoft Kinect is used, which tracks movements of 20 major joints in the body at 30 frames per second. This gives our proposed system the ability to record the motion of each joint as well as its ROM multiple times per second. Secondly, the gestures produced by the users are converted into game movements using an intelligent gesture mapping engine. The gestures are natural and intuitive enough for the users to learn quickly and start using the immersive environment without a steep learning curve. A Game Recommender engine reads joint-motion data and devises map browsing paths to visit geo-spatial locations and perform different activities. Thirdly, the user also has the option of using our OpenStreetMap (OSM) based local map server or internet based map servers such as Nokia, Apple, Google, and ESRI, etc.

Fourthly, the system is web-based; hence, users do not need complex and expensive setup on their client machines. The patient, at home or outside, would just need a web browser, a Kinect, and a LEAP device to start performing her/his prescribed therapy sessions. The system provides a 3D live view of the body movements and hand joints with angular and rotational data. Fifthly, a novel authoring interface has been developed for therapy design. This authoring interface allows a therapist to define the joints of interest based on a human body anatomy model. The system allows a user to save the therapy session in his/her own repository and share only with a community of interest through a dedicated cloud environment. The system also allows replaying any session in 3D environment where a therapist can see the complete motions of a patient, and the patient can see the model therapy suggested by the therapist. Finally a therapist, a patient or a caregiver can see the live plots consisting of quality of improvement metrics by analyzing current or past therapy session data.

The remainder of the paper is organized as follows. Section 2 describes the mathematical modeling as well as our methodology

for mapping therapies to GIS-based serious games. Section 3 details the architecture of the developed framework along with algorithms for therapy-based route recommendation. Section 4 highlights the implementation and experimental setup. Section 5 shows the experimental results and some real demonstration scenarios, while Section 6 draws some conclusions and highlights future work.

2. MODELING APPROACH

This section presents a modeling approach to represent therapies in terms of joints and motions associated with them as well as other dynamic parameters to help generate an appropriate GIS game for a given patient. This modeling approach has been developed based on principles defined with the help of medical experts. In medical literature, movements are expressed in terms of joints and primitive gestures. For example, the bending of the arm that results in the finger tips touching the shoulder is called flexion of the elbow, since the act of flexion happens around the elbow joint. The range of motion (ROM) of the elbow joint for flexion hence depicts the range of angles traversed by the arm during the flexion movement. For a normal human being, the range is 0 degrees (fully extended arm) to 140 degree (fully flexed arm). To model these movements, we have therefore associated each movement with its related joint as well as the normal range of motion. However, a physically impaired person lacks the ability to perform a movement from beginning to end like a healthy person. To enable the system to detect these partial movements and to separate them from other non-related movements, a movement indicator is needed. We have included the direction of change in angle as an indicator to show the type of movement taking place. For example, if the angle around the elbow starts decreasing, it means that the elbow flexion movement has started. Similarly, an increase in the angle shows the start of the elbow extension movement.

Therapy data model. For the purpose of modeling, we consider a therapy as

$$\mathcal{T} = \langle g1, g2, g3, \dots, gn \rangle \quad (1)$$

where \mathcal{T} is the therapy devised by a therapist for a given disability, and $g1, g2, g3, \dots, gn$ are the primitive gestures required to accomplish that therapy. Also , let \mathcal{J} be the set of joints of the human body being tracked.

$$\mathcal{J} = \langle j1, j2, j3, \dots, jn \rangle \quad (2)$$

As examples of trackable joints, we consider:

$j1$ = left knee, $j2$ = right elbow, $j3$ = right shoulder.

Each joint in the human body has a number of motions associated with it. For instance, a wrist joint has flexion, extension, radial deviation, and ulnar deviation motions. The set of all possible motions related to the joints in the human body is represented by \mathcal{M} . As a result, a primitive gesture g can be expressed as follows:

$$g = \langle j, m, nROM, di, p, du, n, r \rangle \quad (3)$$

where $j \in \mathcal{J}$ is a joint; $m \in \mathcal{M}$ is a motion associated to j ; $nROM$ describes the normal range of motion for a given joint; $di \in \{\text{clockwise}, \text{anti-clockwise}\}$ is the direction of movement; $p \in \{\text{frontal}(xy\text{-plane}), \text{transverse}(xz\text{-plane}), \text{sagittal}(yz\text{-plane})\}$ represents the plane of movement as explained in Figure 1; du is the gesture duration as suggested by the therapist; n is the number of repetitions of that gesture during the therapy; and finally r depicts the potential resistance that is the weight carried by some parts of the body for developing all of the muscles that control the patient's hands.

³<https://developer.leapmotion.com/leapjs/api-guide>

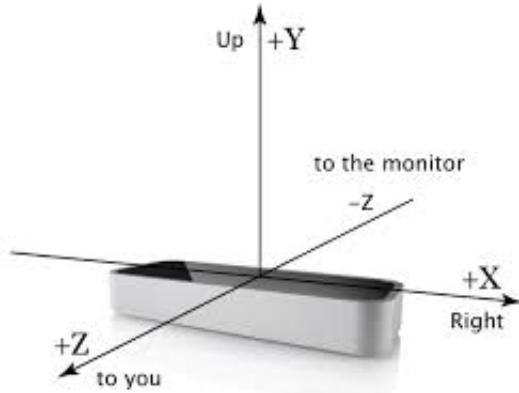


Figure 1: Explanation of the three leap planes as frontal (xy-plane), transverse (xz-plane), and sagittal (yz-plane) ³

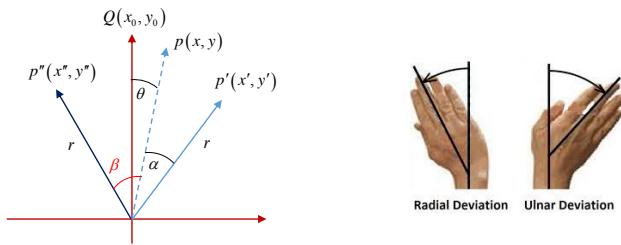


Figure 2: Therapy Model for Right Hand

As an example, the therapy “elbow bend” can be expressed as a combination of primitive gestures “elbow flexion” and “elbow extension”. This can be represented as follows.

$$\mathcal{T}_{\text{ElbowBend}} = \langle g_{\text{ElbowFlexion}}, g_{\text{ElbowExtension}} \rangle$$

$$g_{\text{ElbowFlexion}} = \langle \text{elbow}, \text{flexion}, 140, \text{clockwise}, \text{sagittal}, 2\text{sec}, 5, 0 \rangle$$

$$g_{\text{ElbowExtension}} = \langle \text{elbow}, \text{extension}, 140, \text{anti-clockwise}, \text{sagittal}, 2\text{sec}, 10, 0 \rangle$$

where the last value 0 indicates that there is no resistance applied since nothing is carried in hand.

From a gesture duration and the number of repetition as prescribed by the therapist, the system can then compute the total therapy duration as follows:

$$\text{TherapyDuration} = \sum \text{gesture's duration} * \text{number of repetition}$$

At a given temporal instant, we consider a joint to be in a particular state; and at that state the joint produces one or more motions related to that state. A joint state can be expressed as a combination of these motions and their respective angles for that particular joint. We define the set of states \mathcal{S} as follows. Particularly, we have developed a hand therapy data model, which is directly linked to the serious game recommender engine. A wrist joint can exist in a state of flexion and ulnar deviation at the same time (Figure 2). We can write the state S_j for a wrist joint j as:

$$S_j = \langle (m_1, \alpha_1, \beta_1), (m_2, \alpha_2, \beta_2), \dots, (m_n, \alpha_n, \beta_n) \rangle \quad (4)$$

As illustrated in Figure 2, θ is the angle measure of the initial deviation from the standard position for the hand; θ angle is usually equal to 0 for a person without disability to have a straight initial hand position. α_t is the angular movement suggested by the therapist for the ulnar deviation, provided that $0^\circ \leq \alpha \leq \phi_r^o$; $0 \leq t \leq n$ is the time instant when the snapshot is taken; ϕ_r^o is the extreme possible value (r for right). This value can be defined by the therapist for

each difficulty level. β_t is the angular movement suggested by the therapist for the radial deviation, provided that $0^\circ \leq \beta \leq \phi_l^o$; ϕ_l^o is the extreme possible value (l for left).

Upon examining the patient for the first time, the therapist will provide the parameters α, β, θ and the number of repetitions for each gesture. The system defines a set of initial coordinates $P(x, y)$, $P'(x', y')$, and $P''(x'', y'')$ that actually represent the points or milestones on the GIS game panel. As shown in Figure 2, different coordinates are computed and recorded by the system for therapy-based route generation and further pattern analysis. $Q(x_0, y_0)$ depicts the standard hand position for a normal user; $P(x, y)$ is the initial position of the hand that needs the therapy; $P'(x', y')$ is the right extreme position; and $P''(x'', y'')$ is the left extreme position. When the patient starts playing, he/she has to reach those milestones; this means that he/she has to move his/her hand to achieve certain angles α, β defined by the therapist. From these measured angles, we also define the ϵ_t error value as the difference between the actual and attained angles α_t and β_t . Depending upon the value of ϵ_t , whether it is within the threshold defined by the therapist, the system can decide to design the next game with a higher difficulty level or to repeat the current therapy. Based on this average error for each session, the therapist can extract a report about the performance of the patient.

We also define: $\mathcal{P}_i = S_1, S_2, \dots, S_j, \dots, S_n$ (5)

Where \mathcal{P}_i is a primitive therapeutic context or snapshot of the skeleton at a particular moment in time and each element of the set represents the state S for the joint j in the human body.

A typical therapy consists of a number of movements that take different joints from one state to another in a given sequence. Hence we can represent a therapy as a set of states the whole body goes through. Since each state is represented by P as shown above, we can represent a therapy state \mathcal{TS} as an ordered n-tuple of primitive therapeutic contexts P .

$$\mathcal{TS} = \langle \mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n \rangle \quad (6)$$

Where \mathcal{P}_0 is the initial state of the therapy.

It is worth noting that this hand therapy model is presented in 2D planes for simplicity, but this can be extended and applied to 3D hand motions. In addition, this hand therapy model is only applied to the wrist joint of the hand; other joints should have their own state modeling. The state of the full human skeleton at any moment in time can hence be expressed as a set of states of all joints. We call this static snapshot of the skeleton a primitive therapeutic context.

Sensor data. Leap and Kinect are the cheapest sensor devices that are available to store data of 3D motions. Using these sensors, there is no need to connect a part of the body with a wire, which makes it more convenient in serious games. These wireless devices store 3D motion data accurately. In addition, these sensors provides data for each and every joint like PIP, DIP and MCP joints of the fingers and thumb. Sensor data is represented in our framework as follows.

$$\text{SensorData} = \{\text{LeapData}, \text{KinectData}\}$$

$$\text{KinectData} = \langle \text{UID}, \text{joints}, \tau \rangle \quad (7)$$

where joints is the set of joints $\subseteq J$; UID is the user identifier; and τ is the timestamp. Each joint is represented by the following parameters:

$j \in \text{joints} = \langle \text{angle}, \text{name}, x, y, z \rangle$ providing the angle, joint name, and the current position in a local reference for that joint. The leap motion controller provides us with a diversity of information about hands and pointables involved in the serious game. Those parameters are highlighted as follows.

$$\text{LeapData} = \langle \text{frameRate}, \text{hands}, \text{pointables}, \text{timestamp} \rangle \quad (8)$$

where $frameRate$ is the current number of frames per second; $hands$ and $pointables$ represent the set of hands and pointables detected by the leap, respectively; A hand data mainly contains the hand identifier, its direction, and different values about the palm position and status: $hand = \langle id, direction, palminfo \rangle$. Whereas pointable data includes its position and direction as follows: $pointable = \langle id, direction, handid, positioninfo \rangle$.

User constraints. Based on the user profile and therapist recommendations, the system extracts an initial set of constraints for each patient so that the intelligent game recommender can take these constraints into consideration. The set of constraints is represented as follows: $C = \langle c_1, c_2, c_3, \dots, c_n \rangle$, where c_i includes the following parameters:

$$c_i = \langle CID, gestureID, UID, constraintType \rangle \quad (9)$$

In (9), CID is the constraint identifier; $gestureID$ is the gesture (i.e., a couple of joint,motion) to which that constraint is associated; UID is the user identifier; and finally the constraint type is represented by one of the three possible disabilities:

$ConstraintType = \langle speed, ROM, number of repetition \rangle$; this means that a constraint is related either to the speed in achieving some gesture, in the maximum range of motion (ROM) or in the number of repetitions for that particular gesture. One gesture can also have different constraints at the same time. Those constraints are updated by the system upon finishing each session. The system analyses session data recorded by the patient, and based on the current performance the parameters regarding speed, ROM, and number of repetitions are updated. Some constraints can also be removed.

2.1 Mapping Physical Therapy to GIS-Based Serious Games

We have developed a number of natural map browsing gestures to make the browsing experience simple and intuitive. Figure 3 shows a gesture mapping table, which outlines different gestures that have been implemented such as wrist ulnar and radial deviation for moving right and left and wrist flexion and extension for zooming in and out. We have taken these gestures from therapies suggested by therapists to patients suffering from Hemiplegia and other motor disability problems. Each gesture is detected by a gesture specific algorithm that parses the incoming sensory data stream and interprets hand position data to find the particular gesture. As shown in Figure 3, we have designed both 2D and 3D map browsing gestures such that each primitive gesture can be converted into a primitive therapy. As an example of moving around a 2D map, a user has to grab the map by making a clenched fist gesture. Once the map has been grabbed, moving around requires the motion of the clenched fist towards right or left. To zoom into or out of the map, clenched fists of both the hands are used. A different set of gestures have been designed in the case of 3D map browsing, which is shown in Figure 3.

Figure 3 shows the mapping between hand gestures performed by a given user and the primitive user interface that results in the map browsing environment. The normal ROM is given for each joint movement as well the device needed to detect the movement. The last column shows the map environment for which we have implemented the given gesture. For instance, the radial deviation of the wrist joint means moving the hand in a way that the wrist remains fixed while the tips of the fingers sway towards the body. In case of ulnar deviation, the tips of the fingers sway away from the body. We have implemented the primitive action of going towards right through the radial deviation of the left hand or the ulnar deviation of the right hand. Similarly, going left is implemented by the radial deviation of the right hand or the ulnar deviation of the left hand.

| Hand Gesture | Therapy | Map Movement | Range of Motion (Normal Person) | Device | Body Part | Map |
|--------------|-------------------------------|---|--|--------------|--------------------------|--------------|
| | Radial Deviation | Go Left | 0 → 20 | Leap | Wrist | Nokia 3-D |
| | Ulnar Deviation | Go Right | 0 → 30 | Leap | Wrist | Nokia 3-D |
| | Extension / Hyper Extension | Zoom Out | 0 → 60 | Leap | Wrist, Fingers | Nokia 3-D |
| | Flexion | Zoom In | 0 → 90 | Leap | Wrist, Fingers | Nokia 3-D |
| | Abduction/Adduction | Move Up | 0->20 | Leap | Fingers | Nokia 3-D |
| | Fist Clench | Hold and move the map up, down, left, right | Based on movements of Palm, elbow and shoulder | Leap, Kinect | Palm, Elbow and shoulder | All 2-D Maps |
| | Two hands with clenched fists | Zoom In and Zoom out | Based on distance between two hands | Leap, Kinect | Palm, Elbow and shoulder | All 2-D Maps |

Figure 3: Modeling and mapping physical therapy to GIS games

The maximum range of ulnar deviation is 0 to 30 degrees while that of radial deviation is 0 to 20 degrees.

Algorithm 1: Mapping hand gestures to 2D Maps for left, right, top and down movements

Data: $currentLeapFrame; previousLeapFrame;$
 $\langle frameRate, hands, pointables, timestamp \rangle$

Result: Map navigational movement:
 $\langle id, name, angle, duration, spatialdistance \rangle$

```

1 begin
2   if previousLeapFrame is null then
3     lastFrame = currentLeapFrame;
4     return null;
5   deltaXY = extractDifferenceOnXYAxis(currentLeapFrame,
6   previousLeapFrame);
7   currentCenterCoordinates = extractMapCenter();
8   currentZoomLevel = extractMapZoomLevel();
9   scalingFactor = computeScalingFactor(CurrentZoomLevel);
10  newCenterCoordinates =
11    updateMapCenter(currentCenter, deltaXY, scalingFactor);
12  regenerateMap(newCenterCoordinates);
13  mapMovement =
14    extractMapMovement(currentCenterCoordinates,
15    newCenterCoordinates,
16    currentLeapFrame, previousLeapFrame);
17  return mapMovement;
18 end

```

Flexion and extension of the wrist is used to fly low or high. This is equivalent to zooming in and zooming out respectively. Flexion of the wrist means bending of the wrist in such a manner that the tip of the fingers sway towards the inside. Extension is the opposite

movement. When fingers are swayed towards the outside from the anatomical position, this action is called hyperextension. Flexion and extension have a range of 0 to 90 degrees while hyperextension has a range of 0 to 60 degrees. Abduction is the movement of a part of the body away from the body while adduction is the opposite. Finger abduction means opening the fingers so that they move away from the middle finger. Adduction of fingers means moving the fingers towards the middle finger. The normal range of motion for abduction and adduction of fingers is 0 to 20 degrees. The clenching of the fist is used to lock in on 2-D maps. Once the map is locked in, moving the clenched fist moves the map in the direction of motion of the fist. Opening the fist releases the map. Once the map is released, moving the hand moves the pointer on the screen. Similarly, the clenching of the two fists allows a user to zoom in or out of the map. Increasing the distance between the two fists zooms into the map while bringing the fists closer results in a zoom out operation.

Mapping hand gestures to GIS-based serious games requires specific algorithms for each kind of movement. For instance, Algorithm 1 has been developed for mapping hand gestures to navigate in 2D maps through the serious game environment. This algorithm takes Leap data frames as input, and especially works on the current and previous frames to compute the difference in angles, duration, and spatial distance in order to generate the next navigational movement.

As illustrated in Algorithm 1, every Leap frame has to go through the following actions:

- Extracts relevant leap data of all the hand-joints for the current frame
- Compute the difference of joints-data with compared to the previous frame: deltaXY (line 5).
- Retrieve the current center coordinates and zoom level of the map (lines 6-7).
- Based on current map zoom level, generate the scaling factor and render the map with the new center point (lines 8).
- Update the map center based on the scaling factor, deltaXY , and the current center point (lines 9-10).
- Extract the next map movement by taking into account the current and previous center coordinates, as well as the current and previous leap frames (line 11).

This algorithm is applied to 2D Maps for left, right, top and down movements; other algorithms that take Leap data as input for all map movements such as zoom in, zoom out and combinations of these movements have been also designed, but not shown for lack of space.

3. SYSTEM ARCHITECTURE

Our system has three types of users: (i) a patient; (ii) a therapist; and (iii) a caregiver. A brief description of our framework's life cycle is as follows. A therapist uses an authoring interface to define a specific therapy. The interface displays a human body anatomy model, where each of the joints that are capable of being traced is displayed as a hotspot on the body. When a user clicks on a joint, a list of movements related to that joint is displayed with check boxes on the side. A therapist can associate a subset of body joints with a subset of primitive or high-level actions and other trackable metrics. The therapist associates the exercise to a specific therapy stored earlier in database and uploads data to the cloud storage. This exercise is then assigned to any patient suffering from a related disability. When the patient starts a therapy session

using the GIS game, the game engine requests geo-spatial map data from the map server. The gestures performed by the user for playing the game are detected by the sensors and sent to the game engine. A therapy-mapping module then translates these gestures into appropriate movements in the game environment. While the user enjoys an immersive game playing experience, the data captured by the sensors is stored in a predefined format for later playback, processing, and mining. The next parts of this section includes the full description of our framework architecture, along with the algorithm developed for GIS-based game recommendation.

3.1 Framework Design

Our proposed system consists of a three-layered architecture, namely, the storage layer, the core information processing and computational intelligence layer and the user interface layer. The storage layer stores data coming from sensors for the purpose of analysis, but also includes a knowledge base about therapy data and user profiles as well as data on navigational movements. The core information processing and computational intelligence layer includes different components such as the inverse kinematic and spatio-temporal analyser as well as the intelligent game recommender. This game recommender interprets body gestures from sensor data streams and the knowledge base and generates therapy-based navigational routes adapted to the user constraints. The user interface layer provides an authoring environment for therapy design. The output part is made up of different windows showing the Leap and Kinect screens as well as the Quality of Improvement metrics. An interactive graph generation screen is also part of the output interface. Three types of users exist for the system, i.e., the patient, the therapist and the caregiver. The therapist performs the job of designing and storing therapies in database. The recorded therapy sessions are uploaded to the cloud for later use by the patient. The patient can then download these sessions and practice similar therapy sessions. These sessions can then be uploaded for comparison and analysis.

Figure 4 shows the high level framework architecture. We have two types of visualization interfaces. Firstly, the framework shows live physical world activity into a 3D environment, which acts as a serious game environment. Secondly, we have developed 2D interfaces to show analytical output where live plotting of different quality of performance metrics is shown. Details of the components of the system are discussed below. The Game Recommender Engine is discussed separately in Section 3.2.

Live Data Manager. It processes raw data streams from Leap and Kinect sensors. Leap data contains the locations of hand joints observed 60 to 140 times per second whereas Kinect data set contains the location of 20 joints of the body that can be observed 30 times per second. Data is stored in JSON or industry standard BVH format.

Session Recorder. It records the exercise session which can then be saved to a session repository or forwarded to the media extractor for live-view in 3D world or for live plotting. The session recorder provides options such as which joints need to be tracked and also displays live graphs for those joints on the screen.

Session Data. Session data is stored so that it can be played back later in 3D world. Session repository can also be accessed securely by authenticated community of interest (COI) for online health record sharing purposes. COI, also referred to as caregiver social network, is a subset of social ties for a patient who gives medical services to him/her. An example of caregiver social network might be a patient's parents, family members, relatives, medical staff, etc.

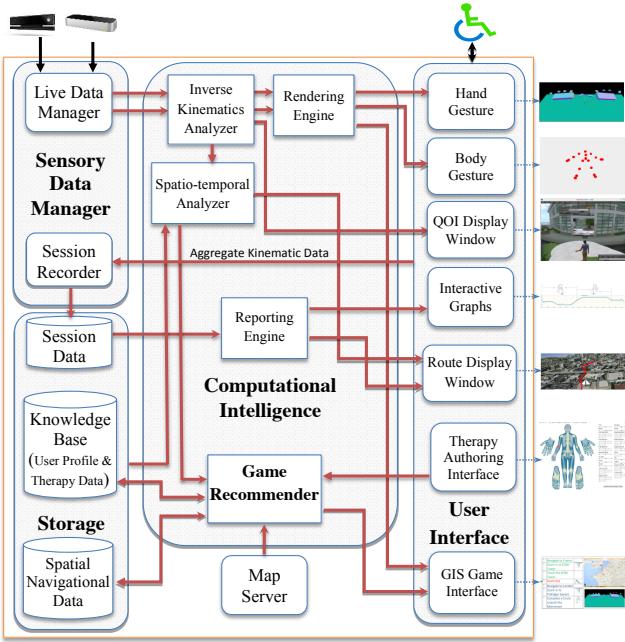


Figure 4: System architecture

Knowledge Base. It includes therapy data as well as the user profiles. The User Profile database acts as an electronic medical record (EMR) which is used to store detailed information about disabled patients. An example of a patient record is family information, type of hand-disability, name of the therapist who is overseeing, types of hand-therapy the patient has to conduct, past therapy history, improvement parameters, etc.

The Therapy Database stores details about disability type, therapy type, types of motions involved in each therapy, joints and muscles to be tracked for each motion type, metrics related to those joints and muscles, normal ranges for each of the factors and metrics, and improvement metrics for each disability type, to name a few.

Spatial Navigational Data. This database contains previous navigational routes computed for a given patient. Those routes are generated by the Game recommender and based on users' feedback and experience. Data from that storage is requested whenever a new difficulty level is being devised for a new therapy session.

Inverse Kinematic Analyser. This component employs analytics and algorithms to retrieve particular therapy information from the raw session data and provide live visualization of different quality of improvement metrics of different hand joints and motions. This also includes a motion extractor component that retrieves raw motion data from Session Repository, combines it with user preferences from the User Profiles database and Therapy database and forwards it to the Session Player.

Rendering Engine. It manages the movement of the hand and other joints positions in the 3D visualization interface over the web. This component leverages the WebGL capability of the web browser to render 3D hand skeletal movement.

Spatio-temporal Analyzer. It takes care of the users' movements with respect to the recommended path. If user somehow change the path then it will recompute the path. It will also take care

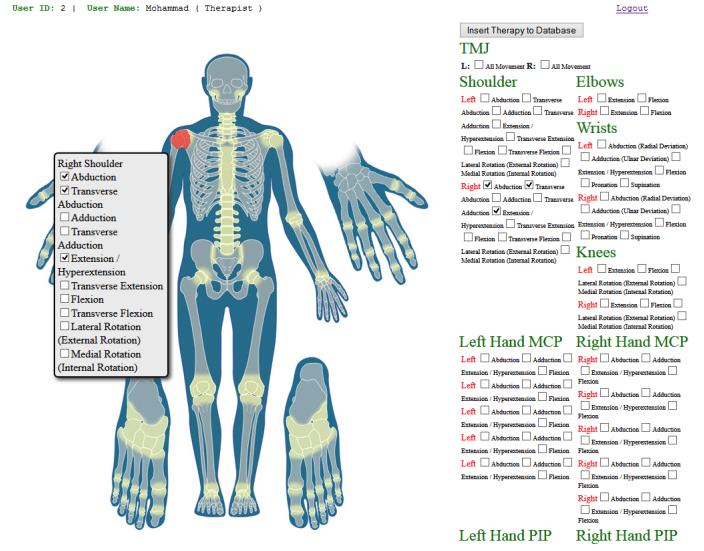


Figure 5: Authoring interface for therapy design

the quality of the path. Another important aspect for this component is the pattern analysis, since paths generated from previous sessions are compared with actual and suggested routes by the system for performance analysis and pattern mining.

Map Server. The Map server is responsible for the map visualization. It generates the tiles based on the input of users like Google, Nokia, ESRI, Staemen. It renders a map tiles using raw geo-spatial data that can be used for efficient caching and dynamic rendering.

Reporting Engine. The Reporting Engine takes the list of joint movements from database for a particular therapy and filters the session files based on these movements. Extracted data is then processed to detect rehabilitation patterns. Data is then sent to the Reporting Engine to be plotted in a visually attractive manner where interactive graphs are plotted by the system from session data files. A person can thus easily understand or analyze the graphical feedback and see the range of motion or quality of improvement for various part of the body.

Live 2D Kinematic Data Visualization. This component takes output from Kinematic Analyser to produce web-based graphs containing quality of improvement metrics.

Live 3D Gesture Visualization. The therapy environment is set up such that a therapist can record an exercise session in 3D environment and assign a model therapy to a patient. A patient can replay the assigned hand therapy being performed by an avatar-hand on the screen. The system can record the patient's session and send it back to the therapist. The therapist can replay any session recorded by the patients or view the graphs containing significant joint data. Temporal data available from a number of sessions over a long period can be used to monitor the effectiveness and progress of the rehabilitation process.

Authoring Interface. The Authoring Interface provides the therapist with an easy to use environment for designing therapies (Figure 5). The interface displays a human body anatomy model, where each of the trackable joints is displayed as a hotspot on the body.

When a user clicks on a joint, a list of possible movements related to that joint is displayed. A therapist can associate a subset of body joints with a subset of primitive or high-level actions.

3.2 Intelligent Game Recommender

This component is the heart of the whole framework, which take feeds from the therapy and user profile databases. Therapy Data stores knowledge of the joints and motions that are included in a given therapy. The user profile as well as user constraints are also extracted for adaptive route generation. Navigational Data includes generated paths for a given patient for previous levels so that knowledge and pattern mining can be performed afterwards. The intelligent game recommender generate an adaptive path that includes all motions required to complete the therapy. Algorithm 2 describes the main steps for creating therapy-based dynamic routes within the GIS game environment. This algorithm takes the therapy gestures $\langle g_1, g_2, \dots, g_n \rangle$ as well as the user constraints $\langle c_1, c_2, \dots, c_n \rangle$ as input. It will also make use of the previously introduced algorithm in Section 2.1 for mapping hand gestures related to the corresponding action taken while playing the game. Newly generated game will be stored in cloud-based geo-spatial Game for Therapy.

Algorithm 2: Intelligent route recommender based on therapy data and user constraints

Data: $\mathcal{T} = \langle g_1, g_2, \dots, g_n \rangle$ therapy data; $C = \langle c_1, c_2, \dots, c_n \rangle$; user constraints; $initialPosition$, $endPosition$, $zoomLevel \in \{lowZoomLevel, \dots, highZoomLevel\}$

Result: An adapted route based on a sequence of navigational movements

```

// M: an ordered set of navigational movements:⟨id, name, angle, duration, spatialdistance⟩
1 begin
2   foreach gesture  $g_i \in \mathcal{T}$  do
3     // check constraints related to gesture  $g_i$ ;
4     foreach constraint  $c_j \in C$  with  $gestureID = g_i$  do
5       Adapt gesture  $g_i$  to  $c_j$ ;
6       Adapt ROM;
7       Adapt speed;
8       Adapt number of repetition;
9     end
10    // Call Algorithm 1;
11    mapMovement  $\leftarrow$  translate gesture  $g_i$  to navigational movement;
12    Add mapMovement to  $M$ ;
13  end
14  return generateAdaptedRoute( $M$ ,  $initialPosition$ ,  $endPosition$ ,  $zoomLevel$ );
15 end

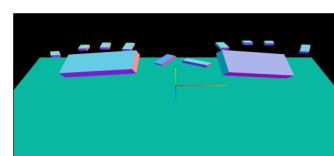
```

As illustrated in Algorithm 2, every standard therapy proposed by the therapist for a given patient has to be adapted according to the constraints defined for that particular patient. As mentioned earlier, the initial set of constraints is defined by the therapist and based on the user profile. However, those constraints are updated upon finishing each session based on the performance of the patient. This route recommender also takes into account the source and destination requested by the patient to make the game more personalized. The zoom level is also considered but within a certain restriction; it is up to the system to recommend certain valid zoom levels depending on the source and destination, as well as the therapy duration derived from the gestures duration as explained in Section 2. The following actions are applied in this intelligent route recommender:

- For each gesture in the prescribed therapy, retrieve the set of related constraints.



(a) Experimental setup. (A) Kinect device; (B) LEAP device; (C) a user that is tracked by both LEAP and Kinect to deduce rotational and angular motions from hand joints



(b) Live rendering of LEAP stream showing hand skeleton



(c) Live 3D rendering of Kinect provided skeleton

Figure 6: Experimental setup and user interface

- For each constraint, adapt that gesture according to the constraint. Adapting the gesture means tuning the values of one the following parameters: speed, ROM, and number of repetitions (lines 2-8).
- Call Algorithm 1 or an equivalent algorithm for different gestures (i.e., other than left, right, up, down). This algorithm is capable of translating that particular gesture to a navigational movement. Then, add this map movement to the ordered set of navigational movement for path generation (lines 9-10).
- Generate an adapted route based on this set of movement, the initial and end positions, as well as the zoom level.

The output of this algorithm is an adapted route based on a sequence of navigational movements. A navigational movement is defined as: $\langle id, name, angle, duration, spatialdistance \rangle$; where id and $name$ is the movement identifier and name respectively; and $angle$ is the movement angle (e.g., 60° to the left); $duration$, and $spatialdistance$ are self explained.

4. IMPLEMENTATION

The therapy environment consists of a PC running Windows 8 with 12GB of RAM with a LeapMotion for Windows device attached to it to capture joint data. We have used the open source 3D JavaScript framework three.js to create 3D games related to hand-Therapy. We have developed algorithms to capture raw joint data from different joints of a subject's hand. The 3D JavaScript framework is configured to relay the live joint data captured from the 3D motion sensor to the web where the hand movements of the avatar's hand representing the subject in the physical world is mapped and synchronized. The overall graphics engine uses WebGL along with HTML5. Our framework saves hand motion



Figure 7: User interface of a GIS serious game

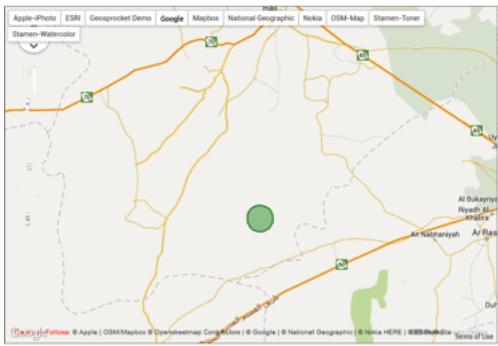


Figure 8: A 2D view of Google Maps with the green ball pointer showing that the map is locked for movement

data in both JSON format and industry standard BVH format. The JSON format has more resolution than the BVH format. We use JSON format for data plotting since we cannot compromise the raw data while for online animation of historical motion we use BVH motion file that is lightweight and requires less bandwidth over Internet. Motion data analyzing and plotting has been implemented using PHP in a server side computer. We use Amazon Elastic Compute Cloud (Amazon EC2) for horizontal and vertical storage facility. Our developed platform is available on Amazon EC2 for public demonstration at <http://54.187.127.36/midml/>. Currently, our framework has been tested by two patients with mild Hemiplegia and their physiotherapists.

Figure 7 illustrates the framework interface where a user is navigating through the map browsing game environment which is tracked by both LEAP and Kinect to deduce rotational and angular motions from hand joints. The 2D map interface illustrated in Figure 8 shows a grey ball as a pointer, which changes color to green once the map is locked. The 3D interface presented in Figure 9 shows a kite like object flying in the 3D terrain following the user's hand movements. This trail of user navigation in 3D map is mapped to the therapy devised to that patient. Navigational movements are proposed are angles of deviations and other therapeutic data are recorded for pattern analysis and for producing quality of improvement feedback about the patient's situation.

5. EXPERIMENTAL RESULTS

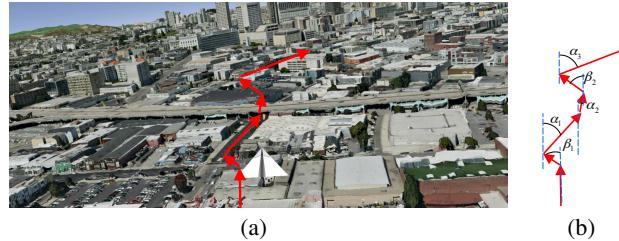


Figure 9: Therapy-based route recommendation. (a) 3D view of Nokia Here Maps with a moving kite. (b) Ulnar and radial deviations performed by the patient

After initial collaboration with 3 local disability centers that provide hand-therapy services to diversified types of patients having different levels of Hemiplegia, we have collected data about different hand therapy modules, the kinds of activities and actions performed during each therapy module, the types of data or metric therapists want to keep an eye about, and the types of joints that are tracked. A therapist stores a model therapy session in the session repository or cloud and then allocates the right therapy module with relevant complexity level to a patient. The therapist can also view his/her recordings in 3D environment. The therapist can view the sessions recorded by a patient at home or at the hospital in both 3D environment or can visualize the plots of different metrics to see the improvement of the patient. A patient can visualize the model therapy session, record his/her own session and share with the therapist. A caregiver family member can see the improvement metrics from the past session analytics. We have collected session data conducted by 1 therapist, 8 normal subjects and two Hemiplegic patients from June 2013 till July 1st, 2014. The following section summarizes our detected motion types and the analysis of the captured test data.

5.1 Evaluation of LEAP-based Primitive and Complex Motion Tracking

As a sample exercise we have implemented a therapy consisting of six movements for the forearm and two joints. The game considered in this test scenario is a 3D map browsing session (see Fig. 10) where a subject browses a map by going left (radial deviation), right (ulnar deviation), zoom in (wrist flexion), zoom out (wrist extension/hyperextension), and circling around the airplane (pronation/supination). LEAP is used to monitor pronation and supination of the forearm as well as flexion, extension, radial deviation and ulnar deviation for the wrist.

The subject starts with a flexed forearm and an open palm. A number of movements are performed in which the palm faces up or down (supination and pronation) and the tip of the fingers face straight up or down (flexion and extension of the wrist). The hand and forearm are moved forward and backward to change the angle formed at the elbow (elbow flexion and extension), so that the patient can follow the recommended route. The system provides live feedback through hand and skeleton animations as well as informing the user about the state of the respective joints (Figure 10).

Figure 10-a shows a certain session in which hand horizontal movement is plotted. X axis shows the number of frame while the Y axis shows normalized range of motion. As seen in figure 3-b, during the therapy session, the user only moved his/hand once in the right direction and once in the left direction. In other words, the positive Y value shows ulnar deviation while the negative Y value indicates a radial deviation. Hence, by real-time intercepting the

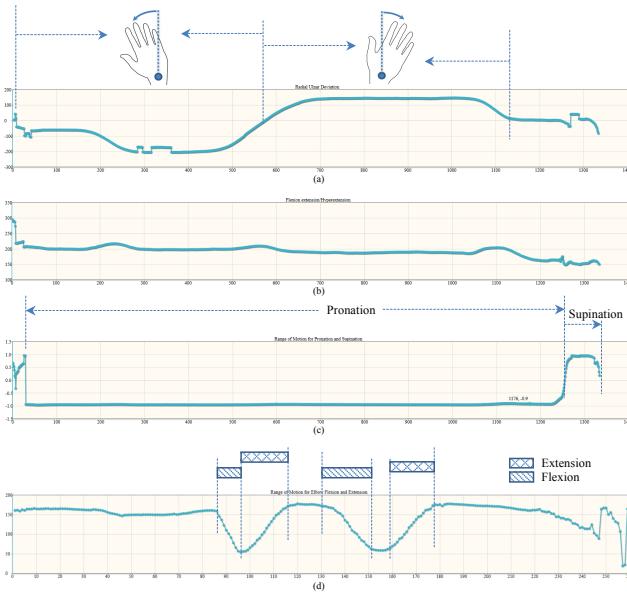


Figure 10: Live plots showing kinematic analysis of: (a) wrist radial-ulnar deviation; (b) flexion extension/hyperextension of wrist; (c) pronation-supination of palm surface; and (d) flexion-extension of elbow joint

Leap and Kinect frames and looking at the Y value gives us a great insight about the radial/ulnar deviation movement of the wrist.

Similarly, Figure 10-b shows that the wrist was initially hyperextended for a short duration and then it was in the extension state for the rest of the therapy session. Figure 10-c shows the subject's palm was in pronation state (palm surface facing downward) for most of the time. Meanwhile at the end, it was in supination state (palm facing up). Finally, Figure 10-d shows the movement of the elbow while elbow was flexing and extending to adjust the map browsing directions.

5.2 Quality of Improvement Evaluation

Figure 11 shows the quality of improvement of a given patient for a particular therapy over different sessions. In the wrist pronation graph, the x-axis shows the frame number while the y-axis shows the value of the palm normal. Initially, the palm normal is -1 which means that the palm is facing downwards. This state is called pronation (Figure 11(a)). In the third session (khaki color), the hand is supinated to its fullest, so we can see the graph going all the way up to +1. The other sessions show partial ability to move the hand from the pronated state to the supinated state. In the wrist adduction graph (Figure 11(b)), the zero lines represents the point where the hand is not making any angle with respect to the forearm in the horizontal position. Thus, the middle finger is pointing straight in this position. Radial and ulnar movements of the hand produce positive and negative values.

6. CONCLUSIONS AND FUTURE WORK

The research presented in this paper introduces a GIS-based serious game recommender integrated within a therapy modeling framework which facilitates a therapist to design a complex and high-level therapy by combining a set of primitive therapies. Using our modeling approach, each therapy can be mapped to a set of body joints and motions. We have also developed a comprehensive three-layered architecture where a multidimensional storage layer, an information

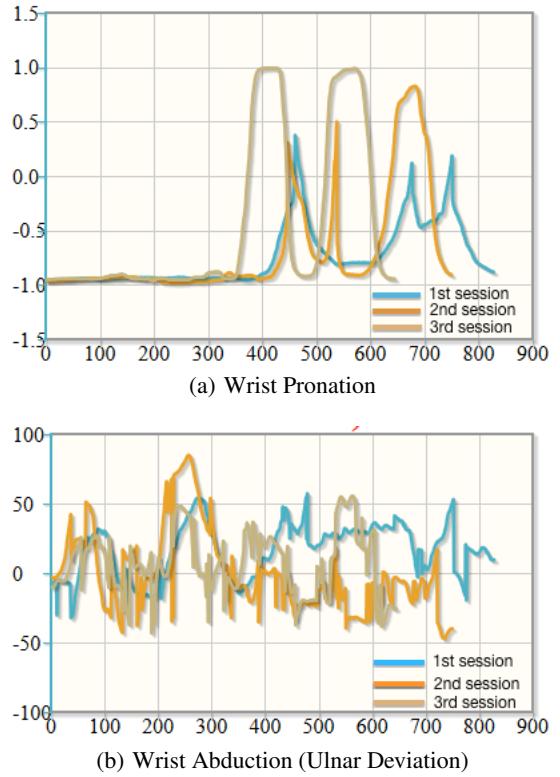


Figure 11: Comparing different sessions of a patient therapy

processing and computational intelligence layer, as well as different user interface components have been developed. The developed open source framework is web-based and hence, a patient can perform the therapies at any-time and anywhere. We have developed an intelligent game recommender within a map-browsing environment where each game is represented by navigational movements generated for a particular therapy actions; and generates metrics and live plot after each game-play. Analytics to parse the Kinect and LEAP generated joint and motion data to infer high-level therapy actions such as navigating and moving within a 3D environment have been also presented. A therapist or an authorized caregiver family member can visualize live or statistical plots of the captured joint and motion data to observe the improvement of affected joints of the patient over time.

Our initial test results show that this framework has potential to explore for more advanced serious games and detailed clinical data mining and analysis. We will be looking at the current shortcomings and feedback from the therapists to make the framework robust and complete. We are at the stage of integrating more primitive actions so that a therapist can produce and make a wide range of high-level therapies. Also, eliminating environmental as well as hardware seen noise is another step that we are focusing since there can be many distractions in the real-life scenario when we deploy to the real patients.

Acknowledgements

This project is supported by the NSTIP strategic technologies program (11-INF1700-10 and 11-INF1703-10) in the Kingdom of Saudi Arabia. We kindly acknowledge the useful suggestions from Dr. Mohamed F. Mokbel of University of Minnesota, Dr. Walid G. Aref of Purdue University and Dr. Ahmed Lbath of University of Grenoble

Alpes, France. We also thank ESRI USA and ESRI Middle East for giving us constructive feedback while we developed the system. We would like to thank GISTIC and Advance Media Laboratory of Umm Al-Qura University, Saudi Arabia for providing the necessary resources.

References

- [1] D. Ayala, O. Wolfson, B. Xu, B. Dasgupta, and J. Lin. Parking slot assignment games. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 299–308. ACM, 2011.
- [2] Y.-J. Chang, S.-F. Chen, and J.-D. Huang. A kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. *Research in developmental disabilities*, 32(6):2566–2570, 2011.
- [3] Z. Cheng, F. Hao, Z. JianYou, and S. Yun. Research on design of serious game based on GIS. In *Proceedings of the IEEE 11th International Conference on Computer-Aided Industrial Design & Conceptual Design (CAIDCD)*, volume 1, pages 231–233. IEEE, 2010.
- [4] R. Jacob, P. Mooney, P. Corcoran, and A. C. Winstanley. Haptic-GIS: Exploring the possibilities. *SIGSPATIAL Special*, 2(3):13–18, 2010.
- [5] K. O. Jordan, I. Sheptykin, B. Grüter, and H.-R. Vatterrott. Identification of structural landmarks in a park using movement data collected in a location-based game. In *Proceedings of The First ACM SIGSPATIAL International Workshop on Computational Models of Place*, pages 1–8. ACM, 2013.
- [6] K. Khoshelham and S. O. Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012.
- [7] L. Omelina, B. Jansen, B. Bonnechere, S. Van Sint Jan, and J. Cornelis. Serious games for physical rehabilitation: designing highly configurable and adaptable games. In *Proceedings of the 9th International Conference on Disability, Virtual Reality & Associated Technologies*, pages 195–201, 2012.
- [8] R. M. A. Qamar, A. and S. Basalamah. Adding inverse kinematics for providing live feedback in a serious game-based rehabilitation system. In *Proceedings of the 5th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, 2014.
- [9] Q. A. H. D. Rahman, M. A. and S. Basalamah. Modeling therapy rehabilitation sessions using non-invasive serious games. In *Proceedings of the IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, 2014.
- [10] P. Rego, P. M. Moreira, and L. P. Reis. Serious games for rehabilitation: A survey and a classification towards a taxonomy. In *Proceedings of the 2010 5th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6. IEEE, 2010.
- [11] E. E. Stone and M. Skubic. Passive in-home measurement of stride-to-stride gait variability comparing vision and kinect sensing. In *Proceedings of Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6491–6494. IEEE, 2011.
- [12] C. Wang, K. S. Phua, K. K. Ang, C. Guan, H. Zhang, R. Lin, S. G. C., B. T. Ang, and C. W. K. Kuah. A feasibility study of non-invasive motor-imagery bci-based robotic rehabilitation for stroke patients. In *Proceedings of the 4th International IEEE/EMBS Conference on Neural Engineering*, pages 271–274. IEEE, 2009.
- [13] F. Weichert. et al. Analysis of the accuracy and robustness of the leap motion controller. *Sensors*, 13(5):6380, 2013.
- [14] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler. Analysis of the accuracy and robustness of the leap motion controller. *Sensors*, 13(5):6380–6393, 2013.
- [15] W. Zhao, J. Chai, and Y.-Q. Xu. Combining marker-based mocap and rgb-d camera for acquiring high-fidelity hand motion data. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 33–42. Eurographics Association, 2012.