Gesture Recognition For Physical Therapy Using Golden Section Search

BARANDINO, RIZALINO III,Ateneo de Davao University
MAGULTA, EMMANUEL HECTOR, Ateneo de Davao University

Virtual reality is widely used for gaming and entertainment purposes but it is also currently used in the medical field. The use of Microsoft's Kinect is applied in physical therapy. Kinect's ability to track joint positions, body gestures and physical movements makes it a very useful tool in physical therapy. Golden Section Search is an efficient method in developing pattern or gesture recognition systems. The challenge therefore is how to utilize Kinect's gesture recognition features with the use of Golden Section Search to create a virtual rehabilitation system for physical therapy patients. This virtual rehabilitation system can be used in a clinical setting and at home which can reduce the frequency of hospital visits of the patient, resulting in the reduction of therapy cost. There are different levels of rehabilitation needed by the patient. However, this study is focused on the different physical therapy exercises of the patient. This research will provide valuable information regarding the use of Kinect's gesture recognition with Golden Section Search in developing a virtual rehabilitation system for physical therapy patients to improve their physical well-being as well as reducing their costs.

General Terms: Golden Section Search, Physical Therapy

Additional Key Words and Phrases: Kinect, Gesture Recognition, Virtual Rehabilitation

1. INTRODUCTION

1.1 Background of the Study

Virtual reality is an artificial environment that is created with special software and presented to the user in such a way that the user is able to recognize himself and operate in that environment. Virtual reality has been widely used for gaming and general entertainment purposes, but the technology has also found its way into the military and medical fields [1].

One way of using this artificial environment is with the use of Kinect. Kinect is an add-on device for the Microsoft Xbox 360 gaming system that enables users to control games, movies and music with physical motion or voice commands and without the need for a separate input controller like a joystick or keyboard. The controller-free gaming environment provided by Kinect makes it possible for sensors to process basic gestures, facial characteristics, sounds and even full body motion activities such as jumping and kicking [2]. By being able to recognize basic gestures and body motion, Kinect can be used in the medical field especially in physical therapy.

Physical Therapy is relevant in the field of medicine which focuses on restoring the health and functional abilities of people after an injury or any trauma to the body. Many people that require physical therapy sometimes are unable to go to therapy clinics because of factors like time and money. With the current advancements in technology it is significant to develop a way to perform physical therapy sessions at the comfort of your own home using the available technology we have today. By using the components of Microsoft's Kinect and the optimization of the Golden Section Search algorithm, it would make it easier for a physical therapy patient to have their therapy sessions progress monitored and analyzed.

1.2 Problem Statement

The study sought to investigate gesture recognition using Golden Section Search to create virtual rehabilitation for physical therapy patients. Moreover, the study sought to answer the following questions:

1. Is Kinect a viable device with the use of its sensor in detecting gestures for physical therapy exercises?

2. Is Golden Section Search algorithm a good algorithm for gesture recognition?

3. Would this research make the physical therapy session cost-efficient?

1.3 Objectives

The Study sought to identify the gesture recognition for physical therapy using Golden Section Searchalgorithm. Specifically, the study intended to accomplish the following objective:

1. To be able to record and play physical therapy exercises as gestures.
2. To be able to apply Golden Section Search to detect gestures.
3. To be able to recognize gestures using Golden Section Search algorithm.

1.4 Significance of the Study

There are multiple ways of having rehabilitation depending on what level of rehabilitation is needed by the patient. Effective rehabilitation programs aid the patients in optimizing their level of physical, psychological and social function, while also reducing their length of stay, re-admission rates and use of primary health care resources. One best way is having the rehabilitation program right inside the patient's home, where it would be very cost-efficient for the patient.

Virtual reality offers the possibility to be precisely adapted to the patient's therapy and to be specific. Virtual reality environments can provide realistic training for the patient in different scenarios and phases of the rehabilitation. It is now conceivable that a study on gesture recognition using Golden Section Search algorithm will be of aid to physical therapy patients. This study could be developed using current, widely available, affordable virtual reality platforms, such as the Kinect. This study will contribute in the field of medical rehabilitation and will also contribute in the field of technology.

1.5 Scope and Limitations

The study covers the recording and detection of physical therapy exercises gestures with the use of Microsoft's Kinect. A device that can be used in capturing 3-dimensional gestures. Record and playback gesture exercises. The study will also be optimizing the usage of Kinect for Windows Depth Image Camera. The study is limited to the joint gestures of a single person only. Multiple person detections, detection and recognition of facial properties, and voice command recognitions are beyond the scope of this study. The study is also limited to the use of Golden Section Search algorithm and Kinect SDK 1.7.

2. REVIEW OF RELATED LITERATURE

2.1 Golden Section Search

The Golden Section Search is an elegant and robust method of locating a minimum in such a bracket. It is an efficient algorithm that finds the minimum value in a range using the golden ratio. This method involves evaluating the function at some point **x** in the larger of two intervals **(a,b) or (b,c)**. [4] If *f(x) < f (b),* then **x** becomes the midpoint, replacing **b**, and **b** moves to become the end point. If *f(x) > f(b),* then **b** remains the midpoint **x** replacing one of the end points. The width of the interval will reduce and the position of the minima will be better defined (Figure 2). The procedure will repeat until desired width is achieved.

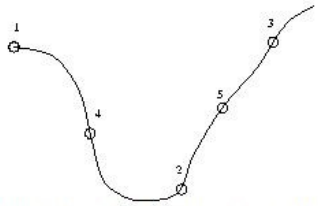
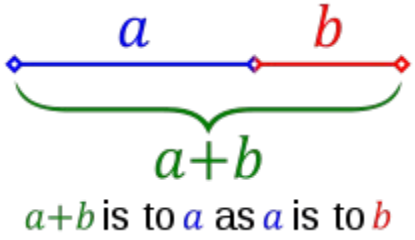


Figure 2: Golden Section Search. Initial bracket (1,2,3) becomes (4,2,3), (4,2,5)...

The Golden Section Search requires no information about the derivative of the function. If such knowledge is known, then it can be used to predict where best to choose the new point **x** in the algorithm, leading to faster convergence [4].

2.1.1 Golden Ratio

The golden ratio, also known as the divine proportion, golden mean or simply as golden section, is a number when taking the ratios of distances in a simple geometric figure [5]. Two numbers are in the golden ratio if their ratio is the same as the ratio of their sum to the larger of the two quantities. An example of the golden ratio in line segments is seen on the figure below.



2.1.2 Golden Section Search Algorithm

A working algorithm

```
GoldenSearch(f, a, b, tol) :=
    grat ← (√5 − 1) / 2
    d ← grat · (b − a)
    x2 ← a + d
    x1 ← b − d
    err ← 100
    numit ← 0
    while err > tol
        numit ← numit + 1
        if f(x1) > f(x2)
            xopt ← x1
            err ← (1 − grat) · |(b − a) / xopt|
            if err > tol
                b ← x2
                x2 ← x1
                d ← grat · (b − a)
                x1 ← b − d
        otherwise
            xopt ← x2
            err ← (1 − grat) · |(b − a) / xopt|
            if err > tol
                a ← x2
                x1 ← x2
                d ← grat · (b − a)
                x2 ← a + d
    ( xopt )
    ( f(xopt) )
```

2.2 Physical Medicine and Rehabilitation

Physical medicine and rehabilitation is the health care specialty focused on restoring the health and functional abilities of people after acute illness or injury such as stroke, spinal cord injuries, heart surgery, amputation, joint replacement, sports injuries or spinal disorders. Physical medicine and rehabilitation (PM&R), also referred to as physiatry, is a medical specialty concerned with diagnosis, evaluation, and management of persons of all ages with physical and/or cognitive impairment and disability. This specialty involves diagnosis and treatment of patients with painful or functionally limiting conditions, the management of comorbidities and co-impairments, diagnostic and therapeutic injection procedures, electrodiagnostic medicine, and emphasis on prevention of complications of disability from secondary conditions.

Physiatrists are trained in the rehabilitation of neurologic disorders, and in the diagnosis and management of impairments of the musculoskeletal (including sports and occupational aspects) and other organ systems, and the long-term management of patients with disabling conditions. Physiatrists provide leadership to multidisciplinary teams concerned with maximal restoration or development of physical, psychological, social, occupational and vocational functions in persons whose abilities have been limited by disease, trauma, congenital disorders or pain to enable people to achieve their maximum functional abilities.

2.3 Gesture Recognition

Interface with computers using gestures of the human body, typically hand movements. In gesture recognition technology, a camera reads the movements of the human body and communicates the data to a computer that uses the gestures as input to control devices or applications. For example, a person clapping his hands together in front of a camera can produce the sound of cymbals being crashed together when the gesture is fed through a computer.

One way gesture recognition is being used is to help the physically impaired to interact with computers, such as interpreting sign language. The technology also has the potential to change the way users interact with computers by eliminating input devices such as joysticks, mice and keyboards and allowing the unencumbered body to give signals to the computer through gestures such as finger pointing.

Unlike haptic interfaces, gesture recognition does not require the user to wear any special equipment or attach any devices to the body. The gestures of the body are read by a camera instead of sensors attached to a device such as a data glove. In addition to hand and body movement, gesture recognition technology also can be used to read facial and speech expressions (i.e., lip reading), and eye movements.

Recognizing gestures as input allows computers to be more accessible for the physically-impaired and makes interaction more natural in a gaming or 3-D virtual world environment. Hand and body gestures can be amplified by a controller that contains accelerometers and gyroscopes to sense tilting, rotation and acceleration of movement -- or the computing device can be outfitted with a

camera so that software in the device can recognize and interpret specific gestures. A wave of the hand, for instance, might terminate the program.

In addition to the technical challenges of implementing gesture recognition, there are also social challenges. Gestures must be simple, intuitive and universally acceptable. The study of gestures and other nonverbal types of communication is known as kinesics.

2.4 Kinect and Kinect used in Rehabilitation

Kinect is Microsoft's motion gaming system for the Xbox 360. The system provides a natural user interface (NUI) that allows users to interact intuitively and without any intermediary device, such as a controller. Kinect is Microsoft's motion sensor add-on for the Xbox 360 gaming console. The device provides a natural user interface (NUI) that allows users to interact intuitively and without any intermediary device, such as a controller.

The Kinect system identifies individual players through face recognition and voice recognition. A depth camera, which "sees" in 3-D, creates a skeleton image of a player and a motion sensor detects their movements. Speech recognition software allows the system to understand spoken commands and gesture enables the tracking of player movements.

Although Kinect was developed for playing games, the technology has been applied to real-world applications as diverse as digital signage, virtual shopping, education, telehealth service delivery and other areas of health IT.

In March 2011, doctors at Sunnybrook Hospital in Toronto began using Kinect to manipulate medical images through gestures during some operations. Doing so allows doctors to interact with the images without leaving the sterile operating area, which means that they don't have to scrub up repeatedly. Keeping doctors within the sterile area lowers the risk of contamination and can prevent delays of up to an hour over the course of an operation. Cumulatively, that time could enable more operations to be performed. The hospital also plans to use Kinect for other purposes, such as physiotherapy.

Released November 4, 2010, Kinect had sold 80 million units by January 3, 2011, achieving the Guinness World Record for the fastest-selling consumer electronics device.

Kinect's development codename was Project Natal. Microsoft chose the name Kinect as a portmanteau of the words *kinetic* (meaning *related to or producing movement*) and *connect*, which the company considers the two key purposes of the system.

2.4.1 Kinect's Specifications

Microsoft Kinect was originally built for the Xbox 360 video game console as an add-on device but later the Windows PC version of it was also released. Kinect provides users with natural user interface (NUI); users can control the whole system/game with either gestures or with voice commands. The software and camera technology of the Kinect were separately developed by Rare and Primesense, respectively.

Kinect consists of an RGB (Red, Green, Blue) camera, a depth sensor and a multi array microphone. The added value of Kinect compared with other cameras is it has the depth sensor that

offers capturing 3D. To accomplish this, the depth sensor has an infrared projector that is found right beside the camera at the front of the Kinect and a monochrome CMOS sensor. Kinect's depth sensor can be adjusted to either near (seated) range mode or far (default) range mode, depending on the prescribed setting of the user. In seated mode, the person within the range of 0.4-3m (1.3-9.8ft) can be seen, though the recommended practical range of this mode is 0.8-2.5m (2.6-8.2ft). In default mode, standing person within 0.8-4m (2.6-13.1ft) are detectable. The recommended practical range of this mode is 1.2-3.5m (3.9-11.5ft). The depth sensor's angular view is 57 degrees horizontal and 43 degrees vertical with a pivot able to tilt to 27 degrees up or down. Kinect's microphone can process 4 channels of 16-bit audio at a sampling rate of 16kHz.

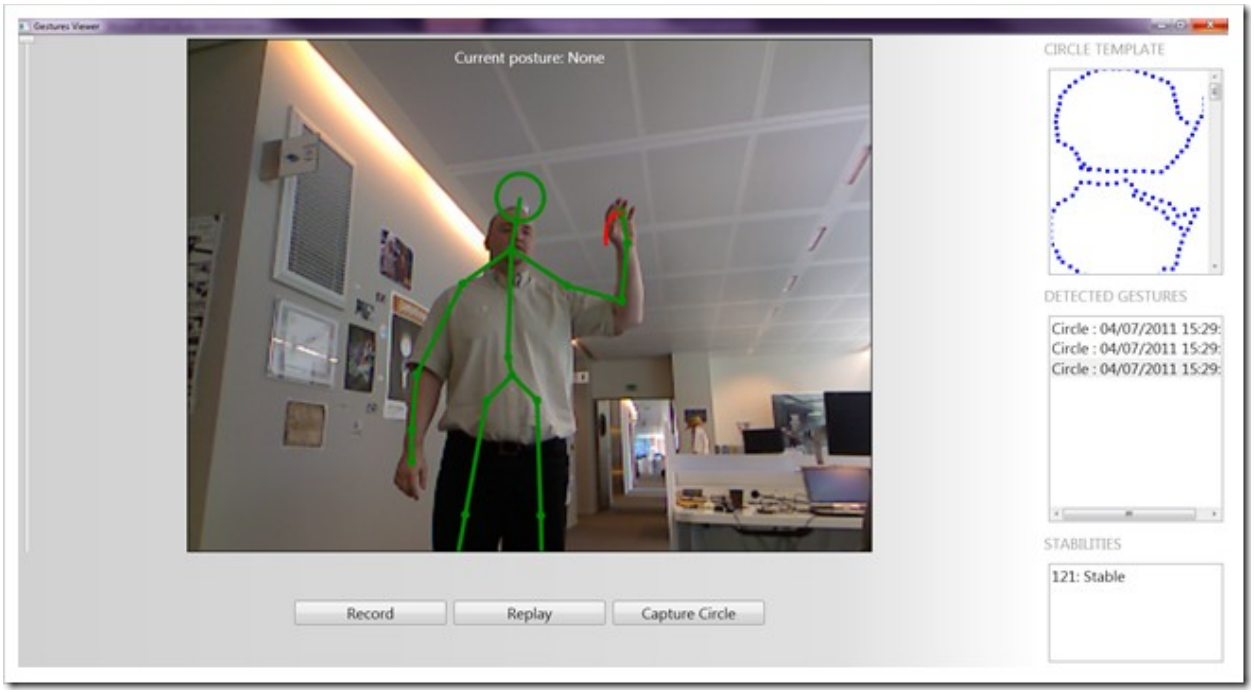
The Kinect sensor can detect up to 6 people but can only track only 2 of them. Developers can access the raw data from any depth, RGB streams or any microphone array. Developers can benefit from skeletal information that includes 20 joints per active individual detected on range, and up to 2 to 6 recognized individuals. Individuals who face the sensor are the ones who will be recognized. For development purposes, there is a SDK for Windows that can be programmed in different languages of C++, C#, and Visual Basic.Net [6].

2.4.2 Kinect Sensor: Technical Considerations for RE Development

Microsoft Kinect provides a software development kit (SDK) which gives developers access to body joint positions and orientations. These joint positions and orientations can be classified as part of the 3D-skeleton projected through the Kinect. The depth image of Kinect contains information relating to the distance of the 3D object that surfaces from the Kinect's camera. Depth image reveals extra information about 3D positions of pixels. This helps with the thorough gesture recognition. Segmentation and background subtraction becomes relatively easier and more accurate; this encourages RE developers to prefer depth sensors over RGB cameras in motion capture applications [6].

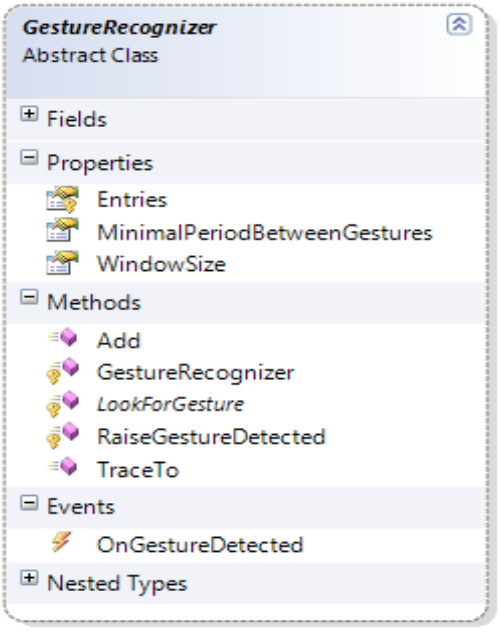
2.5 Gesture Detection in Kinect

Gesture detection in Kinect can be in two ways which are the algorithmic search and the template search.



2.5.1 Gesture Detector Class

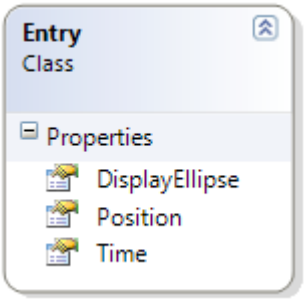
To **standardize** the use of our gestures system, we will therefore propose an abstract class *GestureDetector* inherited by all gesture classes:



This class provides the *Add* method used to record the different positions of the skeleton's joints. It also provides the abstract method *LookForGesture* implemented by the children. It stores a list of *Entry* in the property Entries whose role is to save the properties and timing of each recorded position.

2.5.2 Drawing Store Positions

The Entry class also stores a WPF ellipse that will be used to **draw the stored position**:



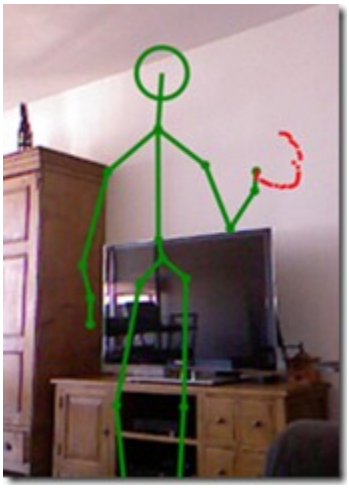
Via the *TraceTo* method of the *GestureDetector* class, we will indicate which canvas will be used to draw the stored positions.

In the end, all the work is done in the *Add* method:

```
1  public virtual void Add(Vector position, SkeletonEngine engine){
2     Entry newEntry = new Entry {Position = position.ToVector3(), Time = DateTime.Now};
3     Entries.Add(newEntry);
4     if (displayCanvas != null) {
5        newEntry.DisplayEllipse = new Ellipse{
6           Width = 4,
7           Height = 4,
8           HorizontalAlignment = HorizontalAlignment.Left,
9           VerticalAlignment = VerticalAlignment.Top,
10          StrokeThickness = 2.0,
11          Stroke = new SolidColorBrush(displayColor),
12          StrokeLineJoin = PenLineJoin.Round  };
13       float x, y;
14       engine.SkeletonToDepthImage(position, out x, out y);
15       x = (float)(x * displayCanvas.ActualWidth);
16       y = (float)(y * displayCanvas.ActualHeight);
```

```
17
18        Canvas.SetLeft(newEntry.DisplayEllipse, x - newEntry.DisplayEllipse.Width / 2);
19        Canvas.SetTop(newEntry.DisplayEllipse, y - newEntry.DisplayEllipse.Height / 2);
20        displayCanvas.Children.Add(newEntry.DisplayEllipse);}
21     if (Entries.Count>WindowSize) {
22        Entry entryToRemove = Entries[0];
23        if (displayCanvas != null) {
24           displayCanvas.Children.Remove(entryToRemove.DisplayEllipse); }
25        Entries.Remove(entryToRemove); }
26     LookForGesture(); }
```

The use of the *SkeletonToDepthImage* method which converts a 3D coordinate to a 2D coordinate between 0 and 1 on each axis. So in addition to saving the position of the joints, the *GestureDetector* class can draw them to give **visual feedback** that greatly simplifies the development and debug phases:



As we can see above, the positions being analyzed are shown in red above the Kinect image. To activate this service, the developer just needs to put a canvas over the image that shows the stream of the Kinect camera and pass this canvas to the *GestureDetector.TraceTo* method:

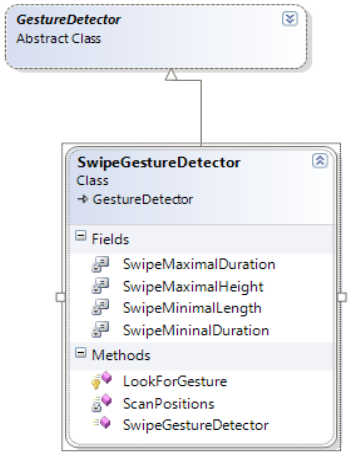
```
1  <Viewbox Margin="5" Grid.RowSpan="5">
2     <Grid Width="640" Height="480" ClipToBounds="True">
```

```
3        <Image x:Name="kinectDisplay"></Image>
4        <Canvas x:Name="kinectCanvas"></Canvas>
5        <Canvas x:Name="gesturesCanvas"></Canvas>
6        <Rectangle Stroke="Black" StrokeThickness="1"/>
7     </Grid>
8  </Viewbox>
```

The Viewbox is used to keep the image and the canvas at the same size. The second canvas (kinectCanvas) is used to display the green skeleton (using a class available in the sample: (*SkeletonDisplayManager*).

2.5.3 Algorithmic Search

The **algorithmic** search browses the list of positions and checks that predefined constraints are always valid. The *SwipeGestureDetector* class is responsible for this search:



For the *SwipeToRight* gesture, ints:

- Each new position should b
- Each position must not exceed in height the first by more than a given distance (20 cm)
- The time between the first and last position must be between 250ms and 1500ms
- The gesture must be at least 40 cm long

The *SwipeToLeft* gesture is based on the same constraints except for the direction of the movement of course.To effectively manage these two gestures, we use a generic algorithm that checks the four constraints mentioned above:

```
1  bool ScanPositions(Func<Vector3, Vector3, bool>heightFunction, Func<Vector3, Vector3, bool>directio
   nFunction,Func<Vector3, Vector3, bool>lengthFunction, int minTime, int maxTime){
```

```
2       int start = 0;
3       for (int index = 1; index <Entries.Count - 1; index++) {
4           if (!heightFunction(Entries[0].Position, Entries[index].Position) | | !
    directionFunction(Entries[index].Position, Entries[index + 1].Position)){
5               start = index; }
6           if (lengthFunction(Entries[index].Position, Entries[start].Position)){
7               double totalMilliseconds = (Entries[index].Time Entries[start].Time).TotalMilliseconds;
8               if (totalMilliseconds>= minTime&&totalMilliseconds<= maxTime) {
9                   return true;} } }
10      return false;}
```

To use this method, we must provide three functions and a time-delay to check. So to manage the two gestures, simply call the following code:

```
1.          protected override void LookForGesture(){
2.              // Swipe to right
3.              if (ScanPositions((p1, p2) => Math.Abs(p2.Y - p1.Y) <SwipeMaximalHeight, // Height
4.                  (p1, p2) => p2.X - p1.X > -0.01f, // Progression to right
5.                  (p1, p2) => Math.Abs(p2.X - p1.X) >SwipeMinimalLength, // Length
6.                  SwipeMininalDuration, SwipeMaximalDuration)) // Duration {
7.                  RaiseGestureDetected(SupportedGesture.SwipeToRight);
8.                  return; }
9.              // Swipe to left
10.             if (ScanPositions((p1, p2) => Math.Abs(p2.Y - p1.Y) <SwipeMaximalHeight,  // Height
11.                 (p1, p2) => p2.X - p1.X < 0.01f, // Progression to right
12.                 (p1, p2) => Math.Abs(p2.X - p1.X) >SwipeMinimalLength, // Length
13.                 SwipeMininalDuration, SwipeMaximalDuration))// Duration{
```

```
14.         RaiseGestureDetected(SupportedGesture.SwipeToLeft);
15.         return; }}
```

With this class it is really **simple** to add gestures that are detectable with constraints.

2.5.4 Conclusion

So we have at our disposal a set of tools for working with Kinect. In addition we have two systems to detect a large number of gestures. Using these tools we can now aspire to make an application that can be applied to certain fields in the industry.

2.5 Computational Intelligence and Game Design for Effective At-Home Rehabilitation

The aim of this article is to describe a game engine that has all the characteristics needed to support rehabilitation at home. The low-cost tracking devices recently introduced in the entertainment market allow measuring reliably at home, in real time, players' motion with a hands-free approach. Such systems have also become a source of inspiration for researchers working in rehabilitation. Computer games appear suited to guide rehabilitation because of their ability to engage the users. However, commercial videogames and game engines lack the peculiar functionalities required in rehabilitation: Games should be adapted to each patient's functional status, and monitoring the patient's motion is mandatory to avoid maladaptation. Feedback on performance and progression of the exercises should be provided. Lastly, several tracking devices should be considered, according to the patient's pathology and rehabilitation aims.

2.6.1 Results

The result of this analysis has led us to develop the Intelligent Game Engine for Rehabilitation (IGER) system, which combines the principles upon which commercial games are designed with the needs of rehabilitation. IGER is heavily based on computational intelligence: Adaptation of the difficulty level of the exercise is carried out through a Bayesian framework from the observation of the patient's success rate. Monitoring is implemented in fuzzy systems and based on rules defined for the exercises by clinicians. Several devices can be attached to IGER through an input abstraction layer, like the Nintendo® (Kyoto, Japan) Wii™ Balance Board™, the Microsoft® (Redmond, WA) Kinect, the Falcon from Novint Technologies (Albuquerque, NM), or the Tyromotion (Graz, Austria) Timo® plate balance board. IGER is complemented with videogames embedded in a specific taxonomy developed to support rehabilitation progression through time.

2.6.2 Materials and Methods

The development of an effective game engine for rehabilitation has to be based on inputs from clinicians and patients. Several meetings have been organized to elicit such specifications. The key issue is that rehabilitation games cannot work as stand-alone applications but must be included into a broader structure involving patients, therapists, clinicians, hospitals, and institutions at the regional/national level. This is specifically the approach pursued inside the REWIRE project, recently

funded by the European Commission. The REWIRE platform implements such a broad perspective by integrating three main hierarchical components: A hospital station (HS), a networking station, and a patient station (PS), under the assumption that such a structure enables effective support of at-home rehabilitation.

The HS is used by clinicians to define and schedule the rehabilitation at home. It also monitors the patient's progression remotely and supports a virtual community of patients and clinicians that helps, educates, and motivates the patients. The network station is installed at the health provider site, at a regional level. It provides advanced data mining functionalities to discover patterns in rehabilitation treatments among hospitals and regions. The PS is installed at patients' homes and has at its core the IGER, which guides patients through their actual rehabilitation schedule using engaging and targeted videogames and monitors patients' movements and their correct execution of the exercises.

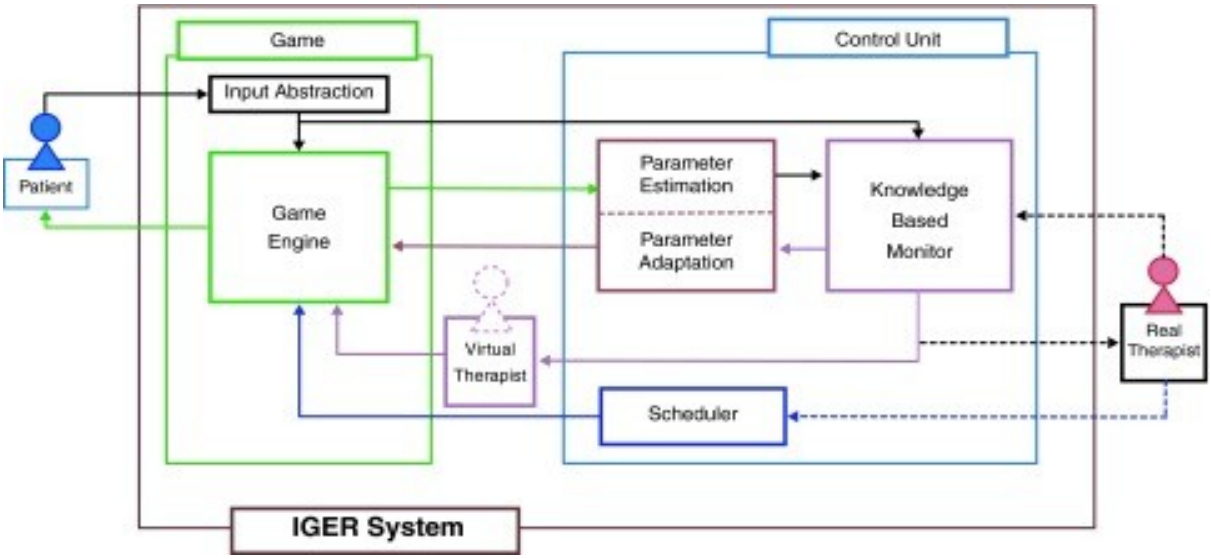


Fig. 1 The Intelligent Game Engine for Rehabilitation (IGER) and its game and control modules. The connections with the patient and the therapist at the hospital and the virtual therapist, which provides real-time feedback to the patient, are highlighted.

2.6.3 The IGER game engine

The IGER is the core component of the PS. It comprehends a game engine and a game control unit. The former provides all the basic gaming functionalities (input data, animation, collision detection, rendering, and game logic); the latter controls the game, and it has been developed to match the needs of games for rehabilitation:

1. It schedules the games, chosen and configured according to the framework.

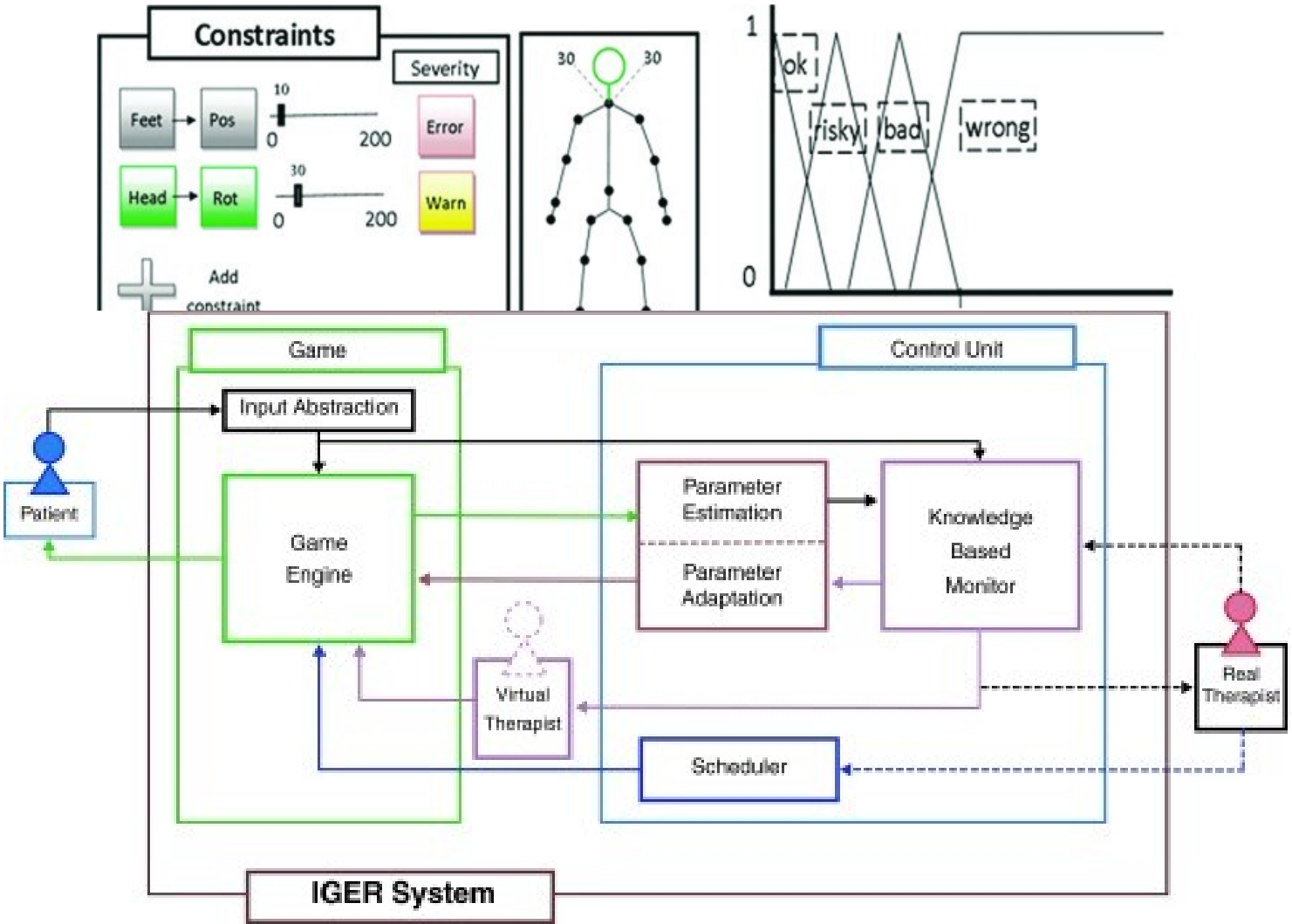
2. It adapts the game difficulty level to the actual patient's performance capacity, so that an adequate challenge level is maintained.

3. It supervises the gaming sessions and monitors whether or if the patient's movements comply with the specifications set by the therapist.

4. It displays a virtual therapist (VT) to advise and provide feedback to the patient on the therapy.

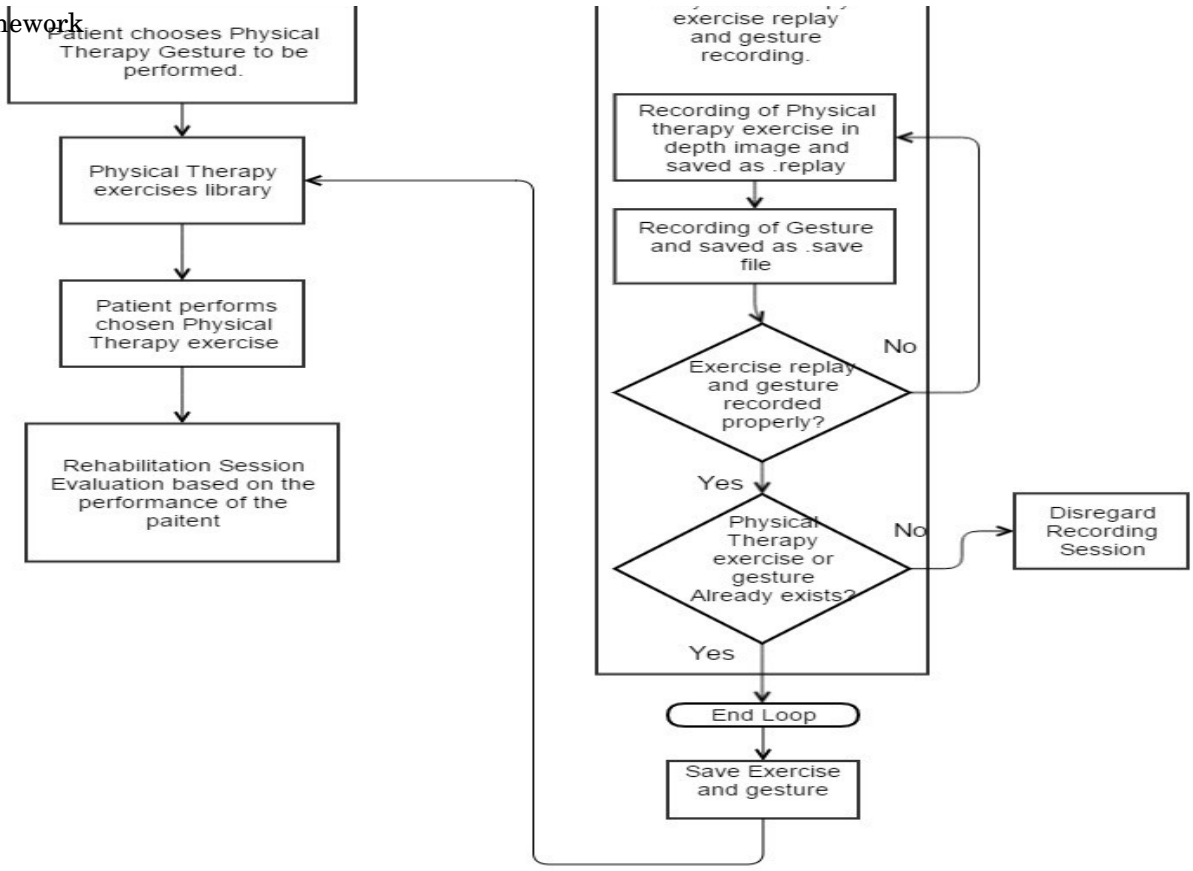
Games defined for rehabilitation have to be parametric. Specifically, games should be defined by parameters that can be regulated and adapted, depending on the level of game difficulty. Such parameters are initialized inside the hospital, where the clinician prescribes the therapy and are continuously adapted to each patient's progression. Such parameters can be, for instance, either the frequency of the obstacles and of the targets inside the game or the range of movements required to complete a game successfully. The control unit implements a real-time adaptation of the parameters through a Bayesian model: The patient's performance is monitored, and the success rate is computed online. The parameters are then increased or decreased such that a certain success rate is maintained, where the amount of change is provided by the model.

The patient's movement is also used for monitoring. To this aim, a set of rules of correct execution is defined inside the HS. Such rules represent a knowledge base from the therapist's experience and are coded into logical propositions like, for instance, “do not bend the trunk” or “keep your feet slightly apart.” Such rules are then fuzzyfied, defining a maximum range allowed for correct execution (e.g., maximum trunk bending allowed in 20°) and associating automatically intermediate ranges with the values in between. A visual interface, showing an avatar inside the game environment, helps the therapists in defining the constraints on the patient's motion.



3. METHODOLOGY

3.1 Conceptual Framework



3.2 Methodology

3.2.1 Data Gathering of Physical Therapy Exercises

The gathering of data of Physical Therapy Exercises must be from a validated source. We used the book Therapeutic Exercise: Foundation and Techniques by Carolyn Kisner and Lynn Colby. The exercises we extracted from the book are classified as "Self-help" exercises which do not require the aid of a physical therapist.

3.2.2 Recording of Physical Therapy Exercise replays.

The recording of the Physical Therapy Exercise replay is done by first taking the skeletal frame and serializing its contents. The exercise replays is then saved with the file extension ".replay" and stored in a chosen directory. This replay is now ready for replay when needed.

3.2.3 Recording of Physical Therapy Exercise Gestures.

The recording of the Physical Therapy Exercise Gestures is done by recording the data points done in the performed gesture. It is then stored with a file extension ".save" which is can now be used in gesture detection as it is used as a template gesture.

3.2.4 Golden Section Search Algorithm

Data points gathered by the exercise from the gesture recorded by the Kinect for Windows device must undergo processes for it to be ready to be fed to the algorithm.

3.2.4.1Data standardization

Data points gathered must be standardized for it to be fed to the algorithm and compared. Data points have different orientations when first recorded and must be standardized to a certain template which can be processed by the algorithm.

3.2.4.2Rotate gesture

Gestures will be rotated so that the first point is at 0 degrees.

3.2.4.3Rescale gestures to 1x1 ratio

Gestures will be rescaled to a 1x1 reference graduation.

3.2.4.4Center the gesture to origin axis

- Compare Gesture Data
- Results to Local Extremum
- Range of Gesture data values

3.2.5 Compare gestures

Data which is now standardized are ready to be fed to the algorithm in order for it to be compared. The data of the gestures is fed to the algorithm and the algorithm gets the local extremum and then gets the range of comparison for the gestures.

3.2.6 Recognize Exercise Gestures

Gestures being performed will now be compared to the template gesture and if the local extremum is within range it will be displayed as a recognized gesture in the system.

3.2.7 Display feedback of detected exercise gestures

The system feedback contains the gesture name that has been recognized and time stamp when the gesture was performed.

4. RESULTS AND DISCUSSIONS

4.1 Results and Evaluation

4.1.1 Recorded Exercise Gesture Testing

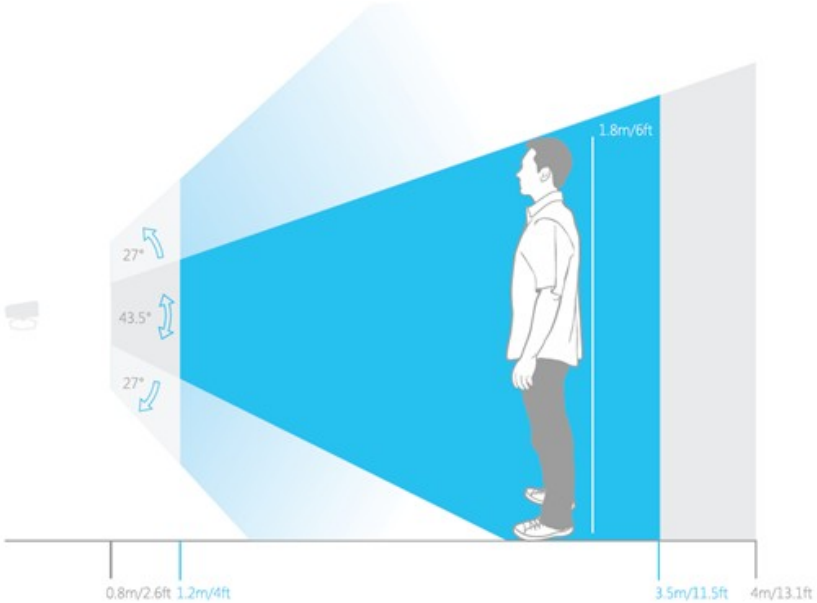
Crossover Armstretch		
n=40	Detected: **NO**	Detected: **YES**
Actual: **NO**	18	2
Actual: **YES**	1	19
	Accuracy: 0.9	
	Precision: 0.95	

Pendulum		
n=40	Detected: **NO**	Detected: **YES**
Actual: **NO**	16	4
Actual: **YES**	5	15
	Accuracy:0.8	
	Precision: 0.75	

Internal Rotation		
n=40	Detected: **NO**	Detected: **YES**
Actual: **NO**	17	3
Actual: **YES**	2	18
	Accuracy: 0.85	
	Precision: 0.9	

Each of the Physical Therapy Gestures has 20 training samples, using the confusion matrix to get the accuracy and precision of the algorithm averaging of Accuracy: 0.85(85%), precision: 0.8666(87%).

4.2 Discus
4.

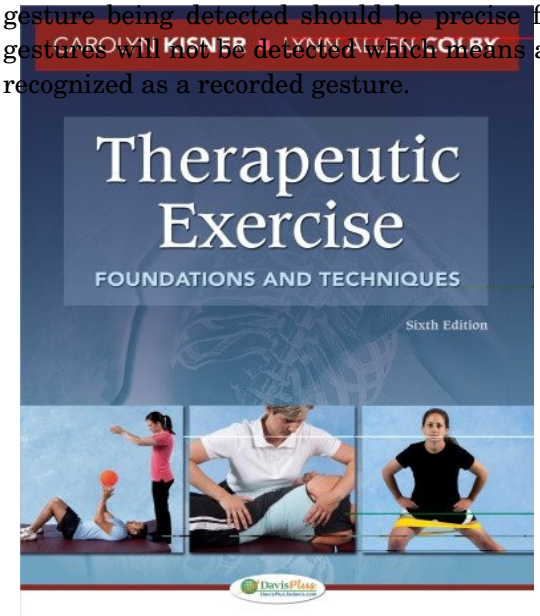


Depth Sensor Range	1.2 to 3.5 meters
Depth Image Stream	320 x 240 16-bit, 30 fps
Color Image Stream	640 x 480 32-bit, 30 fps
Audio Stream	16-bit, 16 kHz
Field of View	Horizontal: 27 degrees Vertical: 43 degrees
Motor Tilt Range	Vertical: ±27 degrees

Kinect skeletal tracking system area range
Kinect sensor specifications

This study is limited to the use of Kinect for Windows device but has proved to be enough in performing necessary data collection for gesture recognition required in this study. The Kinect SDK 1.7 or 1.8 is used in this study for it contains the required drivers to be used in the development of the system using Kinect for Windows device.

Golden Section Search is the algorithm used to compare the gesture being detected and the gesture template that has been recorded. With this algorithm the proponents can get a score (between 0 to 1). The Golden Section Search gets the local extremum of the two gestures and then compares them. This means that the gesture being detected should be precise for it to be recognized by the algorithm. Thus, not precise gestures will not be detected which means a lot of gestures which are not performed well would not be recognized as a recorded gesture.



Physical Therapy Exercises: Foundation and Techniques by Carolyn Kisner and Lynn Allen Cosby

This study uses Physical Therapy Exercises from the book published by Carolyn Kisner and Lynn Allen Cosby. The exercises gathered by the proponents from this book are exercises that are "Self-help" exercises which means that these exercises can be done by the patient alone without the aid of a physical therapist.

5.CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The system used in this study performed well in recording and replaying the Physical Therapy exercises and also in recording and detecting Physical Therapy Gestures. The Golden Section Search algorithm proved to be a good algorithm to be used in gesture recognition. The device used in this study, Kinect for Windows, served its purpose well in achieving the goals of this study. Also, this system can make physical therapy sessions more cost efficient assuming that the devices needed for this system is already available for the users.

5.2 Recommendations

First and foremost, the proponents recommend to future Computer Science student researchers that this study is open for further improvements.

The study has performed well in recording and playing recorded Physical Therapy Exercises and in recording and detecting Physical Therapy gestures. But the overall performance and devices and algorithms used for this study is open for further improvements.

Second, usage of more or other gesture detection algorithms is also recommended. Algorithms which utilizes the data that can be gathered using the Kinect for Windows device can be applied in this study for further improvements.

Third, the study proved that gesture recognition using Kinect for Windows device is enough for this study but adding other devices will aid in further improvements of this study. We recommend utilizing other features of the Kinect for Windows device like audio recognition and near mode of the Kinect for windows device for further improvements of this study.

References

Cameirão MS. Badia SB. Oller ED. Verschure P. Neurorehabilitation using the virtual reality based Rehabilitation Gaming System: Methodology, design, psychometrics, usability and validation. J NeuroengRehabil. 2010;3:7–48.
Koster R. A Theory of Fun for Game Design. Scottsdale, AZ: Paraglyph Press; 2004.
Rizzo A. Kim GJ. A SWOT analysis of the field of virtual reality rehabilitation and therapy. Presence.2005;14:119.
CGN 3421, Computer Methods. www.ce.ufl.edu/~kgurl

[1] http://www.virtual-reality-rehabilitation.com/a/virtual-reality/what-is-virtual-reality

[2] http://www.webopedia.com/TERM/K/kinect.html

[3] ErdenetsogtDavaasambuu, Chia-Chi Chiang, John Y.Chiang, Yung-Fu Chen, SukhbaatarBilgee. A Microsoft Kinect Based Virtual Rehabilitation System (2012)

[4] http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/BMVA96Tut/node17.html

[5] http://mathworld.wolfram.com/GoldenRatio.html

[6] Hossein Mousavi Hondori1 andMaryam Khademi2, A Review on Technical and Clinical Impact of Microsoft Kinect on Physical Therapy and Rehabilitation 11 December 2014