Corrective Self Defense Training Unit using sensing of Kinect maps

Mayank Vij
Jaypee Institute of Information Technology

Suma Dawn
Department of Computer Science
Jaypee Institute of Information Technology

ABSTRACT
In the sports fields and technology, the accuracy of an athlete can never be monitored with 100% accuracy with naked eyes. Moves that are done by an athlete or a sports person in the game such as his stride length and the stride frequency cannot be easily seen by a person and hence, detection and correction of the minute mistakes that are being made are overlooked. For instance, while doing push-ups a person's head, neck, waist and ankle should be in a single straight line. In this paper, we discuss a sports training system which uses Microsoft Kinect for motion sensing. The infrared depth stream translates the presence of a user in real world, into digital space by tracking his/her joint coordinates. As per our empirical studies, Kinect maps the x,y and z coordinates of 20 primary joints of the body. This data forms the basis of our calculations as elaborated in the literature. This work gives a basis of how movements and postures may be corrected by the use of sensors like Kinect in an effective and efficient manner.

General Terms
Kinect, Motion Sensing, Sports, Training, Sports Training, Gestures, HMM.

Keywords
Kinect, Motion Sensing, Sports, Training, Sports Training, Gestures, HMM.

1. INTRODUCTION
This literature aims to propose a module for training an ordinary person on how to defend himself in a daily scenario with some basic martial arts moves. The computer would be trained by a trained person in a particular art of martial arts with the help of the Kinect sensor. At first the person is monitored and all the movements are captured by the Kinect, after which an average of each movement over a large number of repeated moves is taken as the perfect move. Putting this as a benchmark an ordinary user which uses the module would be able to learn it in the beginning and then while practicing, the moves done would be compared to the perfect ones and the error and places where the user lacks would be told to him.

The user can also go for a fight in the virtual world using his avatar in there to compete with various others. Over here Kinect would monitor the level of accuracy of the person and the opponent would fight back in the same way.

In the work, the Kinect collects motion data from different moving body segments during training sessions, with the intention of processing the data and reporting derived measures to coaches in either true or near real-time [1]. As in many of the existing motion sensing systems, this architecture of is based on an assumption of in-network data processing. In-network processing means raw data collected by a set of individual's on-body SUs are initially processed (i.e. filtered, compressed, and more) by an on-board Processing Unit (PU) or an on-body server (such as a PDA). The processed data is then delivered to a remote repository where more computational intensive operations on the data will be carried out. In-network processing is generally considered as beneficial in many on body wireless sensor network scenarios because bandwidth is limited. However, on-body wireless motion sensing systems for sports training face a different set of design challenges.

2. DESIGN OVERVIEW
A platform where sensor based tracking permits information to be gathered about the body of the user is envisioned. The detection and recording of instrument movement would permit the system to not only measure a trainee's progress in acquiring psychomotor skills and compare these data to normative databases, but also to evaluate instrument effectiveness so as to reduce errors.

From a training perspective, the sensor based system shall track and return information on various performance metrics such as position and velocity of the parts of the body ,total path length of motion, erratic movements, time taken, number of attempts, dexterity, etc [2]. It shall be capable of providing users with guidance and feedback during a move.

The aim is to bridge the gap between VR systems and trainers and combine the advantages of both approaches to design a system that is simple, yet effective and efficient. It would be a knowledge-based sensor system, which can provide training to any person for basic self-defense moves [6]. The design employs sensor technology and knowledge-based engines [3].

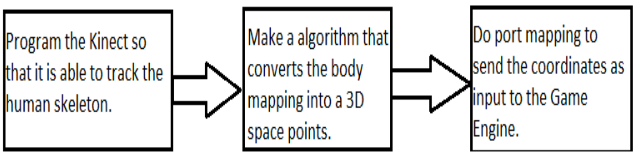

Fig. 1. Generalized Flowchart of the complete work

3. FRAMEWORK
In Figure 2, the system on the left is a model of the traditional system, where the trainer teaches the user and receives visual and force feedback. The system on the right is a model of the proposed design where the trainer teaches the simulator as it does to a normal user in the first case. The system learns the moves from the trainer and learns them [4]. The computer then acts as a supervisor when the user is being trained. The

supervisor represents the sensing interface and the knowledge based computer system [5].

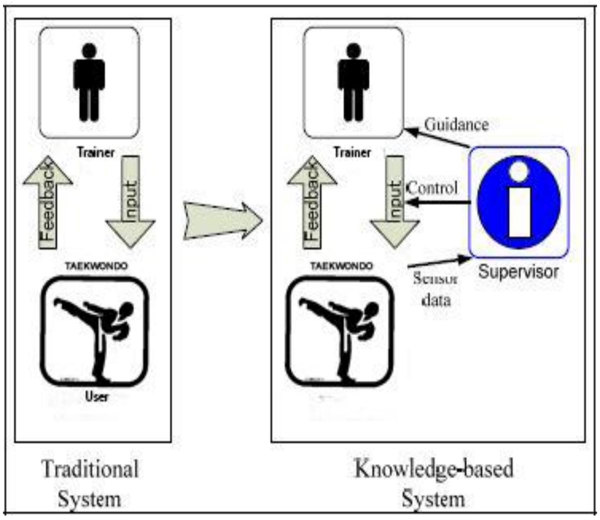

Fig. 2. Traditional System Vs. Knowledge-based system

Kinect Framework

The Kinect uses the motion done by the user as its input hence using it for the program's learning as well as the training of the user. Also, it uses the depth mapping and RGB image to segregate the body from the background and hence develop a skeleton that it uses to map the movement of the user [11], [12].

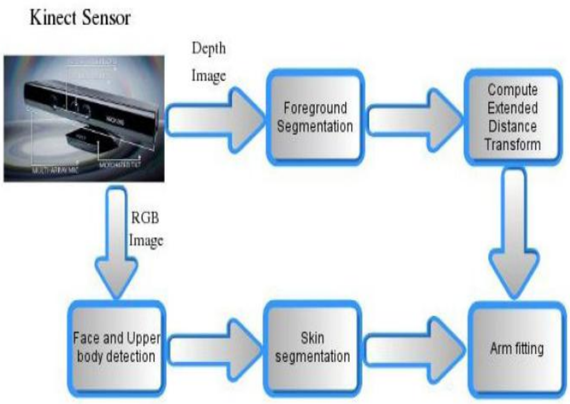

Fig. 3. Overview of the complete Algorithm

4. METHODOLOGY

The Kinect sensor and the program initially would know nothing about any art of self-defence. It would have to learn all the techniques from the scratch just like any person would do from a trainer.

For each move would be made a move space which would tell us what areas in the 3D plane in front of the sensor would be the ones that can be used during any particular move and what all areas are there which can be classified as no fly zones. This would determine the degree of correctness of any move keeping in mind a small bit of error space as no person can do a move accurately in many of the initial tries [7]-[10].

As the difficulty increases, the area would be made smaller in such a manner that the error space still remains.

The work flow is given below:

i) Benchmarking: A martial artist's body coordinates are stored as reference.

ii) 3D avatar mapping: A three dimensional human model in Unity gets the feed from Kinect and therefore emulates the user's movements.

iii) Gesture Recognition: Using Hidden Markov Model to classify user actions into previously learned gestures.

iv) Error Estimation: This is done by comparing user's 3D positions with benchmark and calculating the deviance of the primary joints from correct value.

4.1 Benchmarking

A user's joints positions can be corrected by comparing it with a martial artist's positions, considering the latter to be an accurate benchmark. Hence with the help of Kinect sensor we record the body coordinates of a martial artist as he executes a punch. This action is repeated twenty times as a result of which we obtain 20 instances of the joints. On taking the mean of each column (20 x 20 matrixes) the benchmark coordinates against which all comparisons will be made for a 'punch' are obtained. Such comparison references will be obtained for all the desired moves.

4.1.1 Data Collection

The Kinect sensor tracks 20 joints of the human body by infrared depth sensing. This data is then passed to a 3D human avatar tracker which is developed in Unity Game Engine. This tracker converts the raw Kinect sensor data into the 3D positions of twenty joints, such as elbows, hips, and wrists, for each frame in the recording.

4.2 Gesture Recognition

It is essential to include gesture recognition within the model in order to reduce irrelevant coordinate comparison. The need for this arises in scenarios where although the audio command is for a kick, but the user ends up doing a punch. In such cases, gesture recognition will detect the anomaly in the beginning itself and avoid the program from doing heavy computations of joint comparison for two completely different actions.[4]The 3-D joint locations of individuals are provided while performing simple gestures. From this data, a hidden Markov model (HMM) for each of the gestures is learnt and these models are used to classify unseen gestures.

4.2.1 HMM Model

The intuition behind the model for gestures is that a gesture consists of distinct phases where the joint velocities and joint angular velocities are similar.

1) The states of the hidden variable correspond to the different phases of the gesture. For example, a jump gesture can be thought ofhaving up and down phases, where the former is characterized by positive velocities in the y direction andthe latter is characterized by negative velocities in the y direction.

2) The transition probabilities are such that, from any given state, the same state(With high probability), or the next state (with low probability) can be transitioned, but a previous state would not be transitioned nor skip a state.

4.2.2 Learning
To learn the parameters for the HMM for a particular gesture, all the files containing 3D joint positions of the joints for this gesture have to be collected. For each file, the sequence of joint positions into joint displacements from the previous time step is converted, which can be treated as a sequence of emissions from the HMM.

Using these sequences of emissions as training data and specifying a number of hidden states, expectation maximization can be used to learn the best-fit parameters for the HMM.

4.3 Error Estimation
Once a gesture is classified into one of the known categories, the error estimation can begin.

4.3.1 Classification
To estimate the error, the gesture is classified according to the correctness range. For the previously calculated benchmark of an action, for example that of a punch, an equation of the arm's axis can be formed as the 3D coordinates of the shoulder, elbow and wrist are known.

From this equation, a conic region is calculated by varying the solid angle formed at the shoulder joint (assuming that the elbow and wrist lie on the same axis for this particular punch).

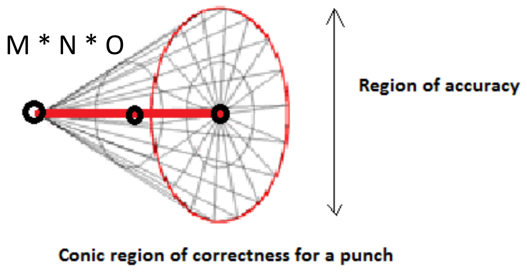

Fig.4. Conic Region of Correctness for a Punch, depicting region of Accuracy.

If the user's punch coordinates lie in the defined conic region, as shown in Fig. 4, then it is a correct move. If it is an outlier, than the displacement from the original position will be calculated.

4.3.2 Joint to Joint Mapping
The displacement of user's joint is calculated with reference to the martial artist's set benchmark. A negative sign will denote a required shift in the position from left to right and vice versa.

This comparison of the joints will be visually depicted with the help of a 3D avatar resembling the user, and the trace of the exact coordinates. This way, the user can readjust his position so as to align the body with the correct position depicted by the outline.

5. RESULT & FINDINGS
5.1 Initial Results
Unity framework and the Kinect SDK were successfully integrated hence the user movements were 95% correctly emulated by the avatar. Under the recognition phase, the HMM worked with good accuracy when 3 layers were considered.

5.2 Data Processing
The volume of data involved was very heavy as 3 dimensional matrix – M x N x O was being dealt with, where 'M' is the number of joints, 'N' is 3 referring to xyz coordinates of joints and 'O' refers to the frames per second, considered as 30 for 30fps.

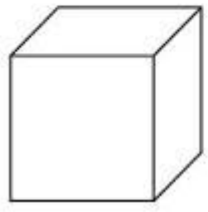

Fig 5. Data Representation using cubic formation.

Noticing that many of the data samples had anomalies such as a mixture of several gestures or long periods of walking orstanding before or after the gesture, the data samples were cleaned to distill them to just the gestures that were wanted for learning, as these anomalies in the data samples might reduce the effectiveness of the implementation.

Fig 6. Gesture mapping using Dept sensor attached to Unity Game Engine.

5.3 Findings
1. It is important that the actions in an NUI based project feel as natural to the user as possible and need not be limited to any specific type.

2. Toolkits and open source code have limited power. Further development have been done to enhance its existing power or to design a purposeful entity using existing capabilities as shown in Fig 5, wherein we have used a model to emulate the tasks as depicted by the depth map.

6. CONCLUSIONS

There were certain problem areas that were identified while preparing the self defense module. The Kinect user mapping works correctly only when the user is in complete view of the camera such that the skeleton is detected. Hence there is a space constraint. Since only xyz values are known the module can be corrective in only that respect. Moreover, Natural interaction supporting functions are still in its growth phase. A better version can be designed if the information of angles and joint rotation is procured. Given the significant development within the two individual areas, their combination has vast scope for exploration and expansion.

7. REFERENCES

[1]Abhishek Kar.” Skeletal Tracking using Microsoft Kinect”, Department of Computer Science and Engineering, IIT Kanpur.

[2] “Real time Human pose recognition in parts from Single depth image”, Microsoft Research Cambridge & Xbox Incubation.

[3] T.Roberts. “Natural Full Body Interaction for Navigation in Dismounted Soldier Training”,in Interservice/Industry Training, Simulation, and Education Conference 2011.

[4] Takayuki Nakamura,” Real-time 3-D Object Tracking Using Kinect Sensor”,in 2011 IEEE International Conference on Robotics and Biomimetics December 7-11, 2011, Phuket, Thailand.

[5] A.Poolton.” Conducting a Comparative Study of Traditional and Hybrid Xbox Control Systems within a Developed Game”, NCCA, Bournemouth University,2011.

[6] “Army Trends Towards Blended Training”,I/ITSEC Showdaily,November 2011

[7] Navid Zolghadr zolghadr and Csaba Szepesv . ”An adaptive algorithm for Finite stochastic particle monitoring”,in Department of Computing Science, University of Alberta, AB, Canada .

[8] Antonio Ricciardi and Patrick Thill , ”Adaptive AI for Fighting Games”,Stanford University 2008.

[9] Niemeyer,G., PING: Poetic Charge and Technical Implementation, The MIT Press,2005 from http://www.jstor.org/stable/20206073

[10] Allen,R.,The Emergence Project: “The British Soul”, The MIT Press,2005 from http://www.jstor.org/stable/20206074

[11] Wikipedia, Natural User Interface, from http://en.wikipedia.org/wiki/Natural_user_interface

[12] Wikipedia, Control Flow Graph, from http://en.wikipedia.org/wiki/Control_flow_graph.