Fall 2013

Repetitive Component Based Motion Learning with Kinect

Govind Kalyankar
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_projects

Part of the [Computer Sciences Commons](#)

Recommended Citation

Kalyankar, Govind, "Repetitive Component Based Motion Learning with Kinect" (2013). *Master's Projects*. 339.

DOI: <https://doi.org/10.31979/etd.hwx8-55d8>

https://scholarworks.sjsu.edu/etd_projects/339

This Master's Project is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Projects by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

Repetitive Component Based Motion Learning with Kinect

A Project

Presented to

The Faculty of the Department of Computer Science

San Jose State University

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

By

Govind Kalyankar

Dec 2013

©2013

Govind Kalyankar

ALL RIGHTS RESERVED

The Designated Project Committee Approves the Project Titled

Repetitive Component Based Motion Learning with Kinect

By

Govind Kalyankar

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSE STATE UNIVERSITY

Dec 2013

Dr. Chris Tseng   Department of Computer Science

Dr. Jon Pearce   Department of Computer Science

Dr. Steven Macramalla   Department of Psychology

ACKNOWLEDGEMENTS

I take it as a great honor to thank Dr. Chris Tseng for his guidance and faith in me throughout the project. Thanks to my committee members, Dr. Jon Pearce, for his great idea, advice and sponsoring the equipment and Dr. Steven Macaramalla for his constant feedback and suggestions about bio-mechanical principles.

ABSTRACT

Repetitive Component Based Motion Learning with Kinect

By Govind Kalyankar

Today's world wants quick, smart and cost effective solutions to their problems. People want to learn everything online. They are interested in learning new techniques and every kind of art in a limited amount of time because they are busy with their own work and have very short time to take in class instructor led training. This is an attempt to fulfill the same so that the people can easily learn and master a new kind of art by themselves by using Kinect. The focus of this project is to master Kung-Fu, an ancient form of Chinese Martial Arts. Kung-Fu requires a good amount of practice and therefore needs a Kung-Fu expert mentoring all the time, which is quite expensive. Therefore, the idea is to develop an application which will help the user in imitating the actions performed by the experts and then comparing the performed actions with the captured recordings of experts. While doing the same motions like a master, users can judge themselves by using Kinect. The User Interface is developed in such a way that it will give the user the summary of his performance by comparing his motions with the recorded motions along with the instructions for the next step and with the suggestions to improve, if user fails to do that . The UI provides User Progress Information on the completion of each task. It also provides a record and replay option which helps the user in reviewing his/her actions. Application focuses on simple Kung-Fu punch movement patterns to do quality analysis and data quantity. This project is a combination of Bio-Mechanical principles and Neural Networking technology. It implements real time motion tracking, coaching and evaluation by providing real time feedback using artificial neural network while capturing motion.

Table of Contents

Table of Contents	vi
List of Figures	viii
1. Introduction	1
1.1 Kinect Sensor	3
1.2 Getting Inside a Kinect Sensor	3
1.3 The Depth Sensor	4
1.4 The Kinect Microphones	6
1.5 Skeleton Tracking	7
2. Biomechanics Principles.....	9
2.1 Eight Biomechanics Principles	9
3. Application Environment Setup	12
3.1 Development Environment.....	12
3.2 Reference JavaScript Library.....	12
3.3 PHP Tremani Neural Network.....	12
4. Motion Data Processing.....	13
4.1 DTW Algorithm	13
4.2 Redesigned DTW for Project	14
4.3 Output of DTW Algorithm.....	18
5. Artificial Neural Network.....	19
5.1 PHP Neural Network.....	23
5.2 Neural Network Object Oriented Design in PHP	26
6. Zigfu JavaScript API	28
6.1 ZDK for JavaScript.....	29

6.2 ZDK for Unity3D	29
6.3 Record Script.....	29
6.4 Replay Script	31
6.5 Color Bone Based on Principle Score	32
7. Internal Working.....	34
7.1 Login Systems	34
7.2 Real time online Virtual Sifu feedback system	36
7.3 Database Schema	41
8. User Interface	45
8.1 Website Features	46
8.2 Score Points Tree.....	47
8.3 UI for recording user's motion.....	51
8.4 Virtual Sifu Workflow and Screenshots	53
9. Conclusion and Future Work.....	61
10. List of References	62

List of Figures

Figure 1. Kinect Sensor.....	3
Figure 2. Kinect Sensor Hardware.....	4
Figure 3. Kinect Depth Sensor.....	4
Figure 4. Joint Information	7
Figure 5. 3x3 Rotation Matrix	15
Figure 6. Rotation Matrix Element Position	15
Figure 7. X axis rotation	16
Figure 8. Y axis rotation	16
Figure 9. Z axis rotation.....	16
Figure 10. DTW values grouped per principle	18
Figure 11. Performance graph of Biomechanical Principle Hip Initiation	20
Figure 12. Performance graph of Biomechanical Principle shoulder down	21
Figure 13. Performance graph of Biomechanical Principle Spinal Posture.....	22
Figure 14. Training Neural Network for Principle 1, 2, 3.....	23
Figure 15. Using already Trained Network	24
Figure 16. Neural Network Object Oriented Design	26
Figure 17. Class Diagram of Login System.....	34
Figure 18. Uml class diagram of online real time Virtual Sifu feedback.....	36
Figure 19. Structure of record_replay_skeleton.js JavaScript file	37
Figure 20. Structure of record_replay_datasaver.js JavaScript file	38
Figure 21. Structure of DTW.php file.....	39
Figure 22. Structure of Principle.php file	40
Figure 23. Database Schema.....	41
Figure 24. High Level View of Application	46

Figure 25. Score Tree.....	49
Figure 26. User performing action like Expert	51
Figure 27. Indication of wrong movement by red orange color	51
Figure 28. List of Violated principles with link address.....	52
Figure 29. Virtual Sifu Suggestion	52
Figure 30. Virtual Sifu Action Flow	53
Figure 31. Hip Initiation Violated.....	54
Figure 32. Spinal Transfer Violated.....	55
Figure 33. Home Page.....	56
Figure 34. Cosmic Dragon Tree.....	56
Figure 35. Sash/Badge on a completion of the form.....	57
Figure 36. Virtual Sifu Explanation, Feedback, Recommendation tabs	57
Figure 37. Biomechanics principle demonstration and explanation videos.....	58
Figure 38. Expert Demonstrating tiger kicking and tiger wrist	59
Figure 39. User Performance Page	60
Figure 40. FAQ page.....	60

1. Introduction

The Virtual Sifu is an entirely new and unique approach to learn Kung-Fu. It gives all the Kung-Fu lovers an opportunity to learn Kung-Fu online with experts instantly tracking their own performance. The application allows a user to watch the videos online and navigates the user through the stepwise instruction led learning of each principle of Kung-Fu.

Each move is divided into multiple steps with unlimited attempts to make the move so that any naive user can start with the move and master every principle till perfection.

The application mirrors live actions of users and judges the principles against them. For example, if the user's stance is incorrect, the body parts in incorrect stance become red in mirrored image and user can adjust the stance accordingly by aligning the limbs or body in order to make it perfect.

The user can then record his actions by simply clicking on the record button. The application thus in response will calculate the performance statistics of the recorded move and also provide feedback and intuitive suggestions for the move made. The application also provides the replay option which will tell the user, where exactly the move is missed. The application keeps a track of your past performance records and allows the user to start with the move he last left with.

On the successful completion of every move the user is bestowed with the badge which is required for a user to move up to the next level.

Commercially available real-time motion tracking systems are marker-based, which require the users to wear obtrusive devices [26] resulting in comparatively complex, expensive and difficulty in maintaining systems.

The application is built with the user's perspective in mind where there is absolutely no effort in setting up the Virtual Sifu. With easy interactive user interface, the application also provides an intuitive step by step guide in navigating the user throughout the process of learning Kung-Fu. As the application is online, there is no overhead of purchasing and installing a whole bunch of supporting software to get started, thus making it a really cost effective solution.

When it comes to comparing the Virtual Sifu with other existing software, the application exists online whereas the earlier implementations of this project were offline. There was a significant overhead of installing and managing the tons of software to setup Virtual Sifu. However, this application has integrated everything and the user doesn't have any overhead of managing anything.

The early implementations didn't have interactive user interface and the feedback was not real time whereas this implementation provides the instant and real time feedback system which instructs the user of his error while performing the actions.

The application makes use of the PHP neural network which processes lot of data on server side and calculates the action's results effectively. The application takes a minimum amount of memory for loading and running, making it really fast and providing a real life learning experience.

Microsoft Kinect has gained substantial reputation for its smooth tracking capability. It is affordable and is widely available and it's easy to use as compared with other motion-capturing devices such as optiTrack. Many developers are now increasingly using Kinect for developing various kinds of Motion related projects. In our case, we are interested in a project that involves real time Motion-Comparison by implementing various motion comparison techniques. There are several approaches that use motion-comparison using Microsoft Kinect, for example, in the

modern game development, the performer motion is placed in sync with the pre-recorded object or avatar appeared on the screen.

Kinect is a motion sensing input device developed for Xbox 360 video game console and Windows PCs. Users can control and interact with the Xbox 360 without requiring touching the game controller through the Natural user interface (NUI) provided by Microsoft using the voice commands and user gestures. Microsoft released the Kinect in November of 2010, as the first commercial release of a 3D camera also known as a Depth Camera Device. This camera made possible to see in 3D.

1.1 Kinect Sensor

The Kinect sensor bar has two cameras, a dedicated infrared light source and microphones (it has 4 microphones). It has signal processing hardware's which is capable to make sense of all the data the camera, microphone and infrared light able to capture. By taking a combination of these sensors output, a program able to recognize and track objects in play space. It is also capable of determining the sound signal direction and separate it from background noise.

1.2 Getting Inside a Kinect Sensor

Figure 1. Kinect Sensor

The Figure 1 shows that Kinect sensor has two cameras one is in the middle and a special light source on the left side. It has 4 microphones arranged along the sensor bar. Combining all these devices provides the view of the world in front of it [19].

Figure 2. Kinect Sensor Hardware

Figure 2 shows all the hardware of the Kinect that makes sense of the information being supplied from various devices. It also has a tiny fan to reduce the heat produced by some components by sucking the air along the circuit to keep cool. The gear assembly and electric motor are responsible for adjusting Kinect angle of view vertically.

1.3 The Depth Sensor

Figure 3. Kinect Depth Sensor

Figure 3 shows the projector is the left-hand-item looks like a camera but in fact it is a tiny infrared projector [2]. On the right side it has the infrared camera. The LED that displays the Kinect device status is in between the camera and projector and a camera used to capture the standard 2D view of a scene.

Kinect uses an RGB camera to capture the color image frames generated after players' gestures. With this color image library included in Kinect SDK, we are allowed to choose a different frame rate and format as per user need. Various available format are listed below:

- **Resolution_640×480_Fps30**

By default color image library is set to 640x480 resolution RGB mode with 30 frames per second frame rate.

- **Resolution_1280×960_Fps12**

If a user would like to change resolution higher than 640x480 then color image library also allows resolution of 1280x960 in RGB mode with same frame rate of 30 frames per second as RGB 640x 480.

- **Raw_Yuv_Resolution_640×480_Fps15**

This format of resolution is generally used for raw uncompressed format also known as YUI format. This format is available in 640x480 resolution format with the frame rate of 15 frames per second.

- **Yuv_Resolution_640×480_Fps15**

This format of resolution is generally used for uncompressed format also known as YUI format. This format is available in 640x480 resolution format with the frame rate of 15 frames per second.

1.4 The Kinect Microphones

As we know that the Kinect sensor contains 4 microphones arranged at the bottom of the bar.

Two microphones are on the right and left ends and remaining two were placed on the right side of the unit. These microphones are required to determine the direction of coming a particular voice from the room.

It works because sound takes some time to travel through air. The travelling speed of sound is much slower than light.

When you speak to the Kinect sensor, each microphone will receive your voice at different times because there is some distance between the sound source and the microphone. Then software will extract your voice waveform from the each microphone produced sound signal & making use of timing information by calculating the source of sound in a room.

It can even work with several people in the same room and find which person is talking by using calculation to find the direction of the particular person voice is coming and can then direct the microphone array to listen to that area of room [19]. Then to understand the speech content it removes the unwanted noises or sounds.

1.5 Skeleton Tracking

Figure 4. Joint Information

As we saw earlier, the Kinect SDK uses Depth Stream to detect the user standing in front of the sensor. The sensor can capture or detect at max six people but cannot track all of them at the same time. It can only track maximum two users at a time. The Zigfu JavaScript SDK also follows same Joint anatomy like Kinect SDK.

Kinect SDK API has a skeleton class which is used to hold a set of motion data. A skeleton provides twenty position sets, one for each body joint, and provide twenty rotations sets which correspond to twenty bones made up of 20 joints.

Each joint has been given the absolute position of X, Y, Z for each joint in a unit of meter in the play space. It provides the set of 20 joints beginning from Joint Hip Center, growing up to the spine then reaches to the head and then moves left from shoulder center till Joint Hand Left , then grows to write in the same way , then goes down to the joint left till Foot Left, then grows to right and ends with a joint foot right.

The absolute X, Y, Z position refers to the distance from joint point to Kinect sensor in X, Y, Z axis respectively [23]. A bone made up of two joints is wrapped with Absolute or hierarchical rotation, in the form of a matrix [23].

Bones are made up of two joints and are distinguished by their ending joint, so that the sequence of bones set are in a sequence of their ending joints order.

Absolute rotation here defined as the rotation relative to sensor, which is absolute in Kinect play space [23]. Horizontal rotation defined as the rotation relative to the bones ends with parent joints of that of current bone [23].

2. Biomechanics Principles

Biomechanics is the science referring to the external and internal forces acting on the human body and the produced effects as a result of these forces. Specifically, it is the study related to human movement and explains the forces causing this movement.

It is very well known that athletes perform sports skills to varying degrees of success. Biomechanics principles are related to both increased performance and injury prevention. For a player, it is very important to conceptualize and visualize the kind of changes they want to make and the importance of these changes for both their long term and short term goals. For example, a Kung-Fu player might want to modify his/her leg stance in order to reduce his injury risk. There are certain physical laws of mechanics which are involved and can be applied to the human body. Two types of analysis are being used and are named as Qualitative and Quantitative analysis. Qualitative analysis describes each move for example, keep back straight, take a horse stance while performing a punch motion whereas Quantitative analysis describes various factors for example, Distance, Speed and Time.

In order to introduce Biomechanics, Karate expert Dr. Steven Macramalla, SJSU faculty divided the simple punch motion into eight biomechanics principle. These principles are used to determine the user's expert level on overall punch motion.

2.1 Eight Biomechanics Principles

- Hip Initiation
- Spiral Transfer
- Opposition Recoil
- Wrist-Elbow-Shoulder Alignment

- Wrist Alignment with Chest
- Shoulders Down
- Spinal Posture
- Stance

Each principle is based on the movement of a group of body joints. For example, Principle 1 Hip Initiation requires the movement of 8 joints relative to each other. As Zigfu API has a list of 20 Joints varying from 1 to 24, so below we have grouped some set of joints of particular principles.

HipInitiation	HipLeft	KneeLeft	AnkleLeft	FootLeft	HipRight	KneeRight	AnkleRight	FootRight			
SpiralTransfer	Spine	ShoulderCenter	Head	ShoulderLeft	ShoulderRight						
OppositionRecoil	Spine	ShoulderCenter	ShoulderRight	ElbowLeft	WristLeft	HandLeft	HipLeft	KneeLeft	AnkleLeft	FootLeft	HipRight
Wrist-Elbow-ShoulderAlignment	ShoulderRight	ElbowRight	WristRight	HandRight							
Wristalignmentwithchest	Spine	ShoulderCenter	ShoulderRight	ElbowRight							
ShouldersDown	ShoulderLeft	ShoulderRight									
Spinalposture	HipCenter	Spine	ShoulderCenter	Head	ShoulderLeft	ShoulderRight					
Stance	KneeLeft	AnkleLeft	FootLeft	KneeRight	AnkleRight	FootRight					

Table 1. Joints matching to Bio-mechanical principle

The left column shows the list of principles and in the same row of the principles, the associated joints are written.

I have prepared the data of each principle in the same way with the same sequence of joints in a text file as an output format of DTW (dynamic time wrapping) algorithm to make it easy to feed the same format data to Neural Network, whereas in the table given below we have shown list of principles common to each body joint.

Hip	Hip Initiation	Opposition Recoil			
UpperBody	Spinal posture	Spiral Transfer	Opposition Recoil		
Head	Spinal Posture	Spiral Transfer			
Neck	Spinal Posture	Spiral Transfer			
Left Shoulder	Spinal Posture	Spiral Transfer	Shoulders Down		
Right Shoulder	Spinal Posture	Spiral Transfer	Opposition Recoil	Wrist-Elbow-Shoulder Alignment	Shoulders Down
Left arm	Opposition recoil				
Right arm	Wrist-Elbow-Shoulder Alignment				
Left elbow	Opposition Recoil				
Right elbow	Wrist-Elbow-Shoulder Alignment				
Left Wrist	Opposition Recoil				
Right Wrist	Wrist-Elbow-Shoulder Alignment	Wrist alignment with chest			
Left Leg	Stance	Hip Initiation	Opposition Recoil		
Right Leg	Stance	Hip Initiation	Opposition Recoil		
Left Knee	Stance	Hip Initiation	Opposition Recoil		
Right Knee	Stance	Hip Initiation	Opposition Recoil		
Left Ankle	Stance	Hip Initiation	Opposition Recoil		
Right Ankle	Stance	Hip Initiation	Opposition Recoil.		

Table 2 Bio-mechanical principle matching to joints

The purpose of introducing these principles in this project is to create 8 sets, one per principle with the data of the joints associated with that principle. These sets will be used to judge or analyze the player and give the user a feedback indicating which principle is violated.

3. Application Environment Setup

3.1 Development Environment

- Zigfu JavaScript SDK.
- PHP implementation using WAMP server.
- Apache
- MySql
- PHP Tremani Neural Network library.
- Matlab for testing and assistance.
- Twitter Bootstrap v2.3.2

3.2 Reference JavaScript Library

Zigfu JavaScript SDK

- The ZDK JavaScript bindings provide an access to the depth, image, and skeleton data from the Kinect and user engagement control and some higher-level abstractions for user interface components [25].

3.3 PHP Tremani Neural Network

- The Tremani Neural Network used to build, train and use neural networks in PHP. Easy to use and set up, and does not depend on any external software to be installed on the web server. The software is open source under the BSD license.

4. Motion Data Processing

This Chapter covers the process of preparation of data used for finding scores of biomechanical principles and overall motion. As we saw earlier, every bio-mechanical principle is associated with a certain number of joints. In Zigfu, each joint has 12 degrees of freedom in which position has 3 degrees of freedom and remaining is a 3x3 matrix of rotation vector.

For example a basic punch is made up of 8 biomechanical principles. So the data preprocessing job can match composite motion data set with 8 bio-mechanical principles. From table 1, we know which joints are associated with the given biomechanical principle. Therefore we had a challenge to minimize the complexity of joint data by reducing to one value per joint, so that every biomechanical principle will hold the one dimensional data containing the value of each joint defined for that principle.

So to tackle this challenge we decided to use DTW (Dynamic Time Warping) algorithm. This algorithm is used for measuring the similarity of two sequences which may vary in speed or time. After using this algorithm, we can compare user's skeleton file with a master skeleton file stored on the server and obtain one single value per joint representing the overall difference between the user and master of that particular joint.

4.1 DTW Algorithm

This algorithm measures the similarities between any two given sequences that vary in time and space. This algorithm is capable of finding similarities even if there is a delay in user's motion. For example, in one video, a person is walking slowly and in another video the person is walking fast but we can still identify the similarity in pattern in these two videos. The algorithm can be employed in any type of Linear Form Data. This algorithm can be applied in videos, graphics and audio too.

Consider we have two signals having sequential values which are evenly spaced, so to compare those signals, we keep adding differences in the frequency. If the two signals are having correct alignment, then there should not be any problem but the problem occurs if we observe a variation in the signal alignment. If one of the two signals is stretched or compressed when it is compared with the other signal, then we must know which points need to be compared. The DTW algorithm overcomes the above problem by using “Dynamic Programming Approach.”

In general, the DTW algorithm finds an optimal match between two input sequences with certain restrictions. The sequences are "warped" non-linearly in time dimension to determine and to measure the differences of the two sequences without certain non-linear variations in the time dimension. DTW uses the non-linearly wrapped sequences in the time dimension.

4.2 Redesigned DTW for Project

In our situation, we cannot use 1-D array because Zigfu has 12 degrees of freedom as mentioned in the previous chapter. We studied several punch motions and decided that it's always hard to keep the user position relative to Kinect Sensor, even with the picture on screen making possible for a player to overlay it before start recording. So, we came to the conclusion that the positional data relative to Kinect Sensor does not make much sense for consideration of different motion sets.

Thus, we have got rid of three degrees of freedom, which are positioned X, Y, Z, from the composite motion data structure. For rotation, we have two choices- Hierarchical Rotation and Absolute Rotation. As we calculate each and every joint value as a separate value based on their matching to respective bio-mechanical principle, the influence of rotation data of parent bone should be reduced to a minimum. Therefore, we have decided not to use hierarchical rotations

and chose absolute rotation. To decrease nine dimensions of each joint data, we have decided to use Euclidean value.

We have retrieved user's motion profile data stored on the server in JSON format and retrieved data of each frame into a PHP array format from JSON file and stored in Master Array and Student Array respectively. We had 9 rotational values for each joint in Zigfu JS api which we have converted into yaw, pitch & roll rotations in order to get X, Y, Z rotation of a particular joint.

$$R(\alpha, \beta, \gamma) = R_z(\alpha) R_y(\beta) R_x(\gamma) = \begin{pmatrix} \cos \alpha \cos \beta & \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma \\ \sin \alpha \cos \beta & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma \\ -\sin \beta & \cos \beta \sin \gamma & \cos \beta \cos \gamma \end{pmatrix}.$$

Figure 5. 3x3 Rotation Matrix

We obtained following Zigfu rotation matrix per frame for each joint

In order to convert this into 3 rotation angles we used following equations.

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

Figure 6. Rotation Matrix Element Position

$$\alpha = \tan^{-1}(r_{21}/r_{11}),$$

Figure 7. X axis rotation

$$\beta = \tan^{-1} \left(-r_{31} / \sqrt{r_{32}^2 + r_{33}^2} \right),$$

Figure 8. Y axis rotation

$$\gamma = \tan^{-1}(r_{32}/r_{33}).$$

Figure 9. Z axis rotation

After converting 3x3 rotation matrix into 3 degree rotation, we have modified the DTW algorithm to handle the composite motion data. In DTW, we calculated the Euclidean Distance of every pair of 3D numbers and then applied Dynamic Programming Approach of constructing a matrix from topmost left corner to the bottom right corner so that the bottom right corner will eventually hold the minimum sum referred as “Similarity Value”. DTW code is given below.

```
function DTWDistance($masterArray, $studentArray){
    $M=count($masterArray);
    $N=count($studentArray);

    $DTWMatrix=array();
    $DTWMatrix[0][0]=0;
    for ($j=1; $j<=$M; $j++) {
        $DTWMatrix[$j][0]=INF;
```

```

    }

    $jointReader=new JointDataReader();

    for($i=1;$i<=$N;$i++){

        $DTWMatrix[0][$i]=INF;

    }
    for($i=1;$i<=$M;$i++){

        for($j=1;$j<=$N;$j++){

            $cost=$jointReader->EuclideanDistance($masterArray[$i-1],
            $studentArray[$j-1]);
            $DTWMatrix[$i][$j] = $cost + min($DTWMatrix[$i - 1][$j],
            min($DTWMatrix[$i][$j - 1], $DTWMatrix[$i - 1][$j - 1]));

        }

    }

    return $DTWMatrix[$M][$N];
}

function EuclideanDistance($MJData , $SJData){
    return sqrt(pow(($MJData->r1 - $SJData->r1), 2) + pow(($MJData->r2 -
$SJData->r2), 2)
        + pow(($MJData->r3 - $SJData->r3), 2));
}

```

Once we fed Master Data and Student Data to DTW, we got similarity values indicating a difference of student and master motion profile. After this, we assigned each joint the DTW value to the respected joints belongs to bio-mechanical principle.

4.3 Output of DTW Algorithm

```
1 Principle 1 - Hip Initiation :
2 12.00408436425 23.003443372759 11.569912719434 2.5844931960146 23.003443372759 10.71
3
4 Principle 2 - Opposition Recoil :
5 4.090085661293 14.970572648107 18.392777574389 140.0301187203 140.0301187203 12.0040
6
7 Principle 3 -Spinal Transfer & Spinal Posture :
8 2.501793704298 4.090085661293 4.7466225625539 11.259750861142 18.392777574389
9
10 Principle 4 - Shoulders Down :
11 18.392777574389 11.259750861142
12
13 Principle 5 - Wrist-Elbow-Shoulder-Alignment :
14 17.98844571123 22.515698224423 18.392777574389 20.696414638741
15
16 Principle 6 - Wrist Alignment With Chest :
17 22.515698224423
18
19 Principle 7 - Stance :
20 2.5844931960146 23.003443372759 23.003443372759 10.719115768347 20.696414638741 20.6
```

Figure 10. DTW values grouped per principle

Above images show the sample output of the DTW algorithm obtained after processing parsed JSON data of Master and Student, we recorded a total of 91 files of students in JSON format and then we sent this whole data to DTW to compare them with recorded Master files stored on server & obtain the similarity values per record files. Now we are ready to use this DTW data of each file to train Neural Network.

5. Artificial Neural Network

A neural network helps to find complex relationships between data. Initially, we used a large set of data that hold unknown relationships between output and input. A neural network is used to detect that unknown relationship. Once we got success in finding a relationship, the neural network was used to compute the output by feeding a similar input. So essentially, the neural networks quickly learn about the complex relationships between the output and input.

After obtaining similarity score of all Student motion profiles after comparing with Master motion profile using DTW, we were ready to train a neural net by feeding the obtained motion data to get their bio-mechanical score.

We sampled three groups of motion profile data obtained from students performing a simple punch. We had a recording session in an AI class at SJSU and we recorded 20 students' motion data performing simple punch. Every student has given two attempts to perform simple punch and depending on each student performance Dr. Steven (Karate Expert) judged each student based on three biomechanical principles Hip Initiation, Spiral Transfer and Shoulder Down and gave an overall score out of 10 for each student motion profile. We have gathered all scores data, motion data and assumed the three biomechanical principles Hip initiation, Spiral transfer and Shoulder down of each Student motion profile data which holds the same score as their individual motion profile overall score. We used this motion profile data of every student to get DTW score of each Student motion profile as compared to Master motion profile.

Neural Network is also implemented by the Matlab net toolbox. We have separated the DTW values by joints and grouped these values to put into a matrix, then the matrix is transposed to satisfy the requirement of nnet input and fed into a corresponding matrix of bio-mechanical score as a target, and we got following performance graphs:

Figure 11. Performance graph of Biomechanical Principle Hip Initiation

Figure 12. Performance graph of Biomechanical Principle shoulder down

Figure 13. Performance graph of Biomechanical Principle Spinal Posture

The Blue lines indicate the “Training Result”, the Green lines indicate the test set “Validation Result” and the Red lines indicate the “Test Result.” We can see from the diagram that the three lines have similar trend but varies largely in some points.

Training results are approximate because of the smaller data set, though it is still satisfactory.

Bigger training set can produce the best performance. After training, neural network object can be used directly to estimate the biomechanical principles score by providing DTW values as an input.

5.1 PHP Neural Network

I have used Tremani Neural Network library in PHP. The Tremani Neural Network allows building, training and employing trained neural networks in PHP. You can easily use it and set up, and it does not require installation of other external software on the web server. This software is open source so it can be modified easily.

```
2  require_once ("class_neuralnetwork.php");
3  // Create a new neural network with 3 input neurons,
4  // 4 hidden neurons, and 1 output neuron
5  $n = new NeuralNetwork(8, 5, 1);
6
7  // Add test-data to the network. In this case,
8  // we want the network to learn the Principle validation Principle
9  $n->addTestData(array (1.1, 5, 0.8,2.5,5, 2.4, 6.5 ,6.5), array (0.4));
10 $n->addTestData(array (1.8, 8.3, 1.1, 0.8, 8.3, 5.5,10.5,10.5), array (0.2));
11 - - -
12 - - -
13 -$n->addTestData(array (1.86 ,2.52 ,1.39 ,0.90 ,2.52 ,1.82 ,5.92 ,5.92), array (0.1));
14
15 // train the network in max 1000 epochs, with a max squared error of 0.01
16 while (!($success = $n->train(1000, 0.2)) && ++$i<$max) {
17     echo "Round $i: No success...<br />";
18 }
19 // print a message if the network was successfully trained
20 if ($success) {
21     $epochs = $n->getEpoch();
22     echo "Success in $epochs training rounds!<br />";
23 }
24 // Save Weights of trained samples so that we dont need to train every time to get score
25 $n->save('my_network_pl.ini');
26
27 $lr = $learningRates[array_rand($learningRates)];
28 $m = $momentum[array_rand($momentum)];
29 $r = $rounds[array_rand($rounds)];
30 $e = $errors[array_rand($errors)];
```

Figure 14. Training Neural Network for Principle 1, 2, 3

Initially we trained the neural network so that we could employ it for later use. I have created three training PHP files, one for each principle. We collected few values per record file and grouped these values into individual principle joint groups. For example, the principle “shoulder down” has two joints associated; one is the right shoulder joint and another one is the left shoulder joint. So for this principle we will have two values as a DTW result; one for the right shoulder and another for the left shoulder. In Training Dataset of a particular principle, there were certain steps required to train neural network.

```
$n = new NeuralNetwork(i/p neurons, hidden neurons, o/p neurons);
```

This object called would create a neural network with given input array, hidden neurons and output value of the given input. Then we added test-data to the network. In this case, we wanted the network to learn the principle violation function ().

```
$n->addTestData (array (11, 50, 8,25,5, 24, 65 ,65), array (0.4));
```

Then we trained the network in max 1000 epochs, with a max squared error of 0.01.

```
$n->train (1000, 0.01))
```

Once we got success in training, we called the save () function which would save weight and which was used to load the saved weights into the initialized neural network. This way you won't need to train the network each time the application has been executed.

```
$n->save (file);
```

This saved the results into .ini file which would be loaded by the test program to pass input to it and get a NN output without requiring to train a NN every time because on the server if you train then it would cause a delay in getting score values which was not a good idea.

```
<?php
error_reporting(0);
$n = new NeuralNetwork(8, 5, 1); //initialize the neural network
$n->setVerbose(false);

// load the saved weights into the initialized neural network. This way you
won't need to train the network each time the application has been executed

$n->load('my_network_p1.ini');

$resultP1 = $n->calculate($p1);

?>
```

Figure 15. Using already Trained Network

Once we trained Neural Network for a particular principle, we created NNp1test.php, NNp2test.php & NNp3test.php three files which were used to send DTW data of particular joints as an input array to a particular principle test file to get the score as an output quickly.

The function used

Load(Trained NN file of Hip Init)

//Loads trained NN file for particular principle

Calculate (Input Array)

//Return NN score of a particular principle

This application can serve multiple users at the same time because I have used various caching techniques to handle multiple requests efficiently without requiring user to wait.

5.2 Neural Network Object Oriented Design in PHP

Figure 16. Neural Network Object Oriented Design

This figure shows overall object oriented design of Neural Network to get a real time judgment score without requiring the user to wait for getting an output. Here, as I said earlier in this chapter, we had created three Neural Network training files, after successful training three “ini” files generated, which held saved weights in the initialized network. For example, “my_network_p2.ini” file held saved weights of initialized principle 2 neural network.

Similarly, we used these .ini files to load into a particular principle test program so that just DTW value array would be passed to test file as an input and receive scores as an output. For

example, here “NNp2test.php” loaded my_network_p2.ini and used “Calculate ()” method to get the score by feeding given DTW value input array. Then finally this score was sent to the JavaScript file which took care of checking if principles are violated or not. Call “Visualize color ()” method to show the bone color of violating the principle in red.

6. Zigfu JavaScript API

Zigfu is the easiest way to play and make apps based on Kinect. You just have to Download the plugin, connect your sensor to get started. The ZDK JavaScript bindings provide an access to the depth, image, and skeleton data from the Kinect and user engagement control and some higher-level abstractions for user interface components [25]. Written in JavaScript, it supports most popular JavaScript frameworks, such as JQuery, prototype.js, and MooTools.

We have decided to use Zigfu because it has certain advantages and Zigfu is a perfect match for our project requirements.

Why we chose Zigfu because,

- Easiest to install and get going
- Provides free up-to-date professional version of ZDK (Zigfu development kit).
- Willingness to provide support
- Built off the latest versions of the official Microsoft Kinect SDK, OpenNI/NITE [25].
- Useful for building motion-enabled websites.
- Apparently developed by some smart guys.

The Zigfu Development Kit (ZDK) is developed by Zigfu to make a motion-controlled, cross-platform apps with Kinect in JavaScript/HTML5, Unity3D and Flash [17]. The main advantage of Zigfu is that Applications developed with ZDK are portable across all web browsers, 3D sensors, operating systems, computer middleware [25].

6.1 ZDK for JavaScript

The ZDK made possible for developers to develop websites which can be motion-controlled by a simple code of JavaScript's and HTML5. The ZDK JS (JavaScript) binding gives developer the access to image, depth, and skeleton data coming from Kinect. The ZDK also provides the control over the type of some higher-level abstractions and user engagement for user interface components written in JavaScript [25]. The ZDK is compatible with almost all popular JavaScript frameworks such as JQuery, MooTools and Prototype.js.

6.2 ZDK for Unity3D

The ZDK also supports both Web Player and Unity built for making motion-controlled apps. The ZDK integrated with Unity package has already included new features and fully functional unity sample scenes so the developer does not need to start from scratch in developing new games. The main advantage of this package is that it works with both Windows and Mac, with the Microsoft Kinect SDK, with 3D sensors and with Open NI/NITE prime packages [25].

6.3 Record Script

To start with Zigfu, we built the script to control the User Interface with JavaScript in a web browser and integrated this script with the Virtual Sifu website. In the script, we wanted the motion data generated to be stored in a variable and sent to the server by using AJAX technology. AJAX is also known as “Asynchronous JavaScript” which is used to update the given part of a web page without requiring loading of web page every time. In order to start with, we embedded Zig.js file, which had written all functional classes used to communicate with the Kinect sensor in the browser.

```

<script src=zig.min.js></script>
..
..
zig.embed();

/*
  Initializes the UI with the start and replay buttons
*/
function initUI() {
  // Detach Stop and Stop Replay
  $stopButton.detach();
  $stopReplayButton.detach();
  // Append start and start replay
  $buttonContainer.append($startButton);
  $buttonContainer.append($startReplayButton);
}
/*
  Initializes user data objects
*/
function initMotionCaptures() {
  resetUserdata();
  // Last recording should be set to {motiondata:[]}
  lastRecording = {};
  lastRecording.motiondata = [];
} // initMotionCaptures
/*
  resets the userdata variable to {motiondata:[]}
*/
function resetUserdata() {
  userdata = {};
  userdata.motiondata = [];
} // resetUserdata

/*
  Stashes the recorded data to the variable used by record_replay_skeleton.js
*/
function saveRecording(userdata) {
  // Clone the userdata
  lastRecording = JSON.parse(JSON.stringify(userdata));
}

/*
  Start the recording
  Parameters:
  evt - Not used
*/
function startRecording(evt) {
  //console.log("Started recording");

  // Resets the userdata to store new Kinect data captures from scratch
  resetUserdata();

  // Detach the start/replay buttons, append the stop button, keep
  listeners

  $startButton.detach();

```



```

    $startReplayButton.detach();
    $buttonContainer.append($stopButton);

    // Reset frame counter
    frameCounter = 0;
    // Increment session counter
    sessionCounter += 1;
    // Allow recording
    isRecording = true;
} //startRecording

/*
  Stops recording
*/
function stopRecording() {
    // Save the userdata to be able to be replayed
    saveRecording(userdata);

    // Detach Stop button, bring back initial button setup
    initUI();

    // Reset values that should reset
    frameCounter = 0; // count the frames since recording session
    isRecording = false; // true if recording a session
} //stopRecording

```

6.4 Replay Script

After completion of recording script, we wanted a script which would regenerate user/player motion by using recorded data. So we built a replay script which would replay user's recorded motion on a server in a browser.

```

function startReplay() {
    // Detach Start and StartReplay buttons from UI, append stop replay
    button
    $startButton.detach();
    $startReplayButton.detach();
    $buttonContainer.append($stopReplayButton);

    // Begin replay
    isReplaying = true;
    doReplay(0, lastRecording);
}
/*
  Does the replay
  Recursively calls itself.
  Note: Depends on the radar variable from record_replay_skeleton.js
  @param frameCounter count of frames elapsed in the replay
*/

```

```

function doReplay(frameCounter, lastRecording) {
    // Play back the recorded frames until finished
    if (isReplaying && frameCounter < lastRecording.motiondata.length) {
        radar.onFrameUpdate(lastRecording.motiondata[frameCounter]);
        // update again after timeout
        setTimeout( function () {
            doReplay(frameCounter + 1, lastRecording);
        }, FRAME_TIMEOUT);
    }
    else {
        // End of replay
        console.log("Replay ended");
        stopReplay();
    }
}
/*
    Ends the replay and brings back UI to initial setup
*/
function stopReplay() {
    // Detach Stop replay button, bring back initial button setup
    initUI();
    // Reset values that should reset
    frameCounter = 0; // count the frames since recording session
    isReplaying = false;
}

```

6.5 Color Bone Based on Principle Score

```

// Color bones based on kungfu principle scores
function colorBonesByScores (score1, score2, score3, boneEles, boneMap) {
    var jid1, jid2; // temporary joint ids
    var j1, j2; // temporary joint indices within their array container
    var b; // temporary bone object

    for (var i = 0; i < boneEles.length; i++) {
        b = boneMap[i];
        jid1 = b.jointId1; // id of first joint of bone
        jid2 = b.jointId2; // id of second joint of bone
        j1 = ZigfuIdToJointIndex(jid1);
        j2 = ZigfuIdToJointIndex(jid2);
        if (jointIsViolated(j1, score1, score2, score3)
            || jointIsViolated(j2, score1, score2, score3)) {
            // Violated, color red
            boneEles[i].attr("stroke", "red");
        }
        else {
            // Not violated, color black
            boneEles[i].attr("stroke", "blue");
        }
    }
} // colorBonesByScores

```

A bone is colored red if any of its related principles are violated. Bones with no principle violation were colored blue after receiving values from Neural Network.

Parameters provided were score1 Hip Initiation score, score2 Shoulders Down score, score3 Spinal Transfer & Spinal Posture, boneless array of bones as query elements and bone Map mapping of bones to joints.

7. Internal Working

7.1 Login Systems

Figure 17. Class Diagram of Login System

The system has two types of users: Web Administrator and Members. Web Administrator has all privileges of deleting user account, ban user, change database schema, add new tables etc. whereas Members are like registered users will use all website features like recording movement, watching explanatory videos, learning tips from the master, judging their motion online with immediate feedback from Virtual Sifu.

The user has to log into his/her account by entering his/her username and password, the system will validate the user login details, and if details match with the stored details on the database, then user login is considered successful and the system connects the user in his/her account. If not successful, the system produces warning of bad email or password and prevents user entry but the user will get multiple repeated attempts. The anonymous user can register by entering chosen username, email id, password and dynamically generated captcha to ensure that the user is not a bot. Then the registration class will validate his/her details such as email has valid format or not, any random integer, string etc. Then on successful registration, the user details will be entered in tables and all Kung-Fu form tables will get a new entry for this new user and initialize all fields to zero which indicates no progress. We have also provided options such as Forget password. When the user selects forget password option then the system will ask the user to enter an email address to send an activation link to the user, so the user can reset his/her password.

7.2 Real time online Virtual Sifu feedback system

Figure 18. Uml class diagram of online real time Virtual Sifu feedback

As we can see from the above figure that when a user logs in and selects a particular Kung-Fu form to learn and click “Action” button to record his/her motion on a window appearing on the browser, then the user will see two options there, “Record” and “Replay”. When the user clicks the Record button, then a series of events occurs from client side to server side.

As user logs in and goes to Action page of a particular form to learn and record his/her motion then the user will be given four choices “Record”, ”Stop Recording”, ”Replay” and “Stop Replay” whereas button “Stop Recording”& “Stop Relay” are hidden and will appear, once the user start recording or replay.

If user clicks on the record option then the server will call record_replay_skeleton.js JavaScript file which renders moving skeleton on browser imitating user motions, storing user motion data to use it for producing users recorded motion replay. This file triggers events to stop recording, start recording, start replay and stop the replay.

Figure 19. Structure of record_replay_skeleton.js JavaScript file

As the user start recording, his/her motion record_replay_skeleton. js file sends generated JSON data to another JavaScript file record_replay_datasaver.js which will copy all the data sent to it. This file has a function “copyLastUserProfile (user data, amt)” which will make AJAX call, once it gathers 8/16 frames data to send it to “DTW.php” file stored on a server by POST request. The record_replay_datasaver.js file is also responsible for receiving principle scores from “DTW.php” and also responsible for deciding which joints and bones should be shown in red, if the received principle score is below or equal to threshold (in our case threshold is 3). To show the bones in red, this file has a defined function visualizeScoring (score1, score2, score3).

record_replay_datasaver.JS
+ Function : sendPrincipleCheck(userdata, amt) + Function : copyLastUserProfile(userdata, amt) + Function : visualizeScoring(score1, score2, score3)
This Javascript file sends data of every 8 fram to DTW.php, take principle scores form DTW.php Shows skeleton joints in red.

Figure 20. Structure of record_replay_datasaver.js JavaScript file

Implementation details of these JavaScript files are given in Zigfu chapter. Once the user has completed recording by pressing “Stop Recording” button, then the user can view replays and replay will clearly show the user where he was doing wrong. That’s why we provided replay option which would help the user to observe carefully about his/her recorded motion problems, if any.

As I said earlier, while recording, the user data of every 8/16 frames will be sent to “DTW.php” to calculate similarity of current motion of the user with the Master file stored on the server. Then “DTW.php” will send DTW scores to the already trained Neural Network Files to receive scores of three principles- Hip Initiation, Shoulder Down, Spinal posture and transfer the “DTW.php” files which use DTW algorithm, used to find similarity between any two motion sequences. This file quickly takes the received JSON data, parses it and prepares it to be fed to DTW algorithm and also finds the similarity value of all joints to send it to the trained neural network.

Figure 21. Structure of DTW.php file

You will find the implementation details of this file in the chapter “Data Processing”. “DTW.php” file will call principle class object inside program and pass the similarity data of the joints belonging to a particular principle function declared in “Principle. class. PHP” file, so that the “principle.php” file use this data, feed this data to the particular principle trained Neural Network file and obtain scores from it. The Neural Network scores obtained will be sent back to “DTW.php” and “DTW.php” will send received scores back to the JavaScript file. Then the JavaScript file will decide whether the principles are violated or not. If violated, then all the joints and bones belonging to a particular principle will be displayed red on the browser. This whole sequence of processing takes around 100-200ms of delay but it is not perceivable by human eyes so we can call it “Real Time Processing”.

Principle.class.php
- input1 : String (DTW values Array)
+ Principle1 (Dtw values of joints related to Hip Init.) : ArrayList Organized principle 1 joints DTW data and feed it to NN1test.php
+ Principle2 (Dtw values of joints related to Shou.down) : ArrayList Organized principle 2 joints DTW data and feed it to NN2test.php
+ Principle3 (Dtw values of joints related to Spir. Tran) : ArrayList Organized principle 3 joints DTW data and feed it to NN3test.php
+class_neuralnetwork.php This is a NN class being called inside this file.

Figure 22. Structure of Principle.php file

This file is responsible for communicating with Neural Network files and getting principle score back from Neural Network and subsequently send received scores to DTW.php. The Neural Network files Implementation details are provided in the Neural Network chapter.

7.3 Database Schema

In order to manage with user profiles, user progress, user registration, user updates, user earned points and badges, we have created a Database Design which will handle all this. We have created a table for each of the dragon form, moves, and principles.

Figure 23. Database Schema

Each Dragon form or each move of dragon form depends on certain principles which are common to all forms or applicable to all forms. So we have one table for principle names as Table Principle and have the following fields.

PK	FIELD	TYPE	LENGTH
PK	Id	[int] IDENTITY(1,1)	
	Uid	Int	
	Form_id	Int	
	P1	Int	
	P2	Int	
	P3	Int	
	P4	Int	
	P5	Int	

Table 3. DB table for storing principle information of all forms

This is an important table that will store information of each user activity such as for user id 10, it will store all information about this user's activity, keep track of completion of forms , current status of forms, completion status principle of a particular form with its form id.

Another shared table is “Total_points _attempts” common for all dragon form tables. This table is used to keep track of each user total points perform and attempt for the particular form. As in our application, we want to restrict users to earn more points just by mastering only one form. To restrict user up to certain points, in a particular form, to be considered as master, we need a way to store total successful attempts so far so that after defined number of successful attempts, the user won’t earn points after that for that particular form. Once user master one form, he/she has to master its parent or neighboring form to keep earning points and badges. This table has following fields:

PK	FIELD	TYPE	LENGTH
PK	Id	[int] IDENTITY(1,1)	
	Uid	Int	
	Form_id	Int	
	Total_points	Int	
	Total_attempts	Int	

Table 4. DB table Total_points _attempts

Apart from these two tables, all tables are now related to a particular form. Each Dragon form table holds information about user who attempted this form, user status such as completed or not, user performance details about this table Child Dragon Form. Below you can see table structure of the forms:

PK	FIELD	TYPE	LENGTH
PK	Id	[int] IDENTITY(1,1)	11
	Uid	Int	
	Boa	Int	
	Tiger	Int	
	Completed	Int	
	Medal	Nvarchar	

Table 5. Earth Dragon Table

PK	FIELD	TYPE	LENGTH
PK	Id	[int] IDENTITY(1,1)	
	Uid	Int	
	TgKick	Int	
	TgWrist	Int	
	Completed	Int	
	Medal	Nvarchar	

Table 6. Tiger Form Table

As we can see, the Earth Dragon has columns “Tiger” and “Boa” because these are the Child forms of the Earth Dragon. So Earth Dragon table holds total earned points for Tiger and Boa form of the user and if the user has earned all points of these children then Earth dragon will unlock action button or in short, we can say that the user can record complete earth dragon form at the same time. Similarly Tiger form table also has columns like Tgkick and TgWrist because these are the child form so before allowing user to learn or record this form, this table checks column Tgkick and TgWrist points. If user had earned sufficient points, then this table will

update completed status of this form. Similarly all Dragon Form Tables have a similar table defined like above table.

8. User Interface

In today's high speed internet world, almost everything is online. All systems support the use of the high speed Internet and so it is always possible to make the learning systems available online. People nowadays, are interested to learn various kinds of arts, but have very limited time to find a tutor/teacher who can be available at their ease. Here, the focus is on mastering the user in Cosmic Dragon Form, a type of Kung-Fu, an ancient form of Chinese Martial Arts. It takes a lot of efforts and practices to master Cosmic Dragon Form and therefore demands physical presence of a professional Kung-Fu expert monitoring all the time, which is quite expensive & time consuming. Therefore, I have proposed an idea to develop a new way of providing learning experience by creating a website called "Virtual Sifu", where a Virtual Kung-Fu master normally called as Sifu, will give the user, the capability to record, learn, get immediate corrective feedback, online monitoring of user's move and detailed attention at the user's move in real time. In this application, players can learn and perfect different Kung-Fu techniques and forms with the help of Virtual Sifu online monitoring system. All these Kung-Fu forms have been pre-recorded by Kung-Fu experts with the use of Kinect, a motion sensing input device and stored on the servers. This online Virtual learning is designed to help the users to master and improve their martial arts with the real time feedback by comparing the user's movements against those of a pre-recorded motion profile of master and using Artificial Neural Network to predict the user's performance to give an immediate feedback to the user.

Figure 24. High Level View of Application

This is the High level view of this application. Here you can see users standing in play space are performing actions in front of Kinect and with use of Kinect my application is rendering user's moving skeleton in browser and at the same time generated joint data is being processed in real time by comparing users dynamically generating motion data with the predefined data of Master to find the similarity. As per user similarity with master the score is given from 1-10, 1 being lowest and 10 being accurate and if the score is too low in range of 1-3 then users wrongly moving body parts are shown in red so user can quickly change movement to see his body parts in blue (not red).

8.1 Website Features

- **Step By Step Instruction**

Virtual Sifu breaks down every move.

- **Get Immediate Feedback**

Virtual Sifu lets you know your accuracy and where you need to pay attention.

- **Deepen Your Practice**

Virtual Sifu links the martial techniques to lessons on biomechanical basis and to the relevant philosophical principles.

- **Easy navigation**

Virtual Sifu makes it to switch the moves user want to practice.

- **Earn Badges & Rewards**

Virtual Sifu will give users a reward on mastering each form.

- **Online replays**

User can watch replays of their own recorded motion to observe wrong movement easily because the bone color will be changed to red in the replay as well as recording.

- **Point System**

To earn a sash/medal a user must accumulate given number of points . User earns points by doing steps and the whole form of high level of perfection multiple times i.e. not just one and not one step multiple times beyond the allowed number of attempts.

8.2 Score Points Tree

The aim of this website is to make the user a master of Cosmic Dragon Form by providing step by step instructions for each part (also called child) of this form, monitor each and every joint, bone movement during recording and quickly indicate wrong movement by showing that bones or joints in red, so that the user can quickly change movement while performing without requiring to wait for getting feedback. We have divided Cosmic Dragon Form into various small forms to make it easy to learn step by step for beginners. The user has to complete all child forms of parent form to be eligible for recording of whole parent form at one go. In short, you have to follow the bottom up hierarchy as shown below.

The tree below shows that the cosmic dragon is divided into three main forms Sub-Earth Dragon, Celestial Dragon, and Earth Dragon. Sub-Earth Dragon is further divided into two sub forms- Panther and Python, Celestial Dragon is further divided into Cobra and Leopard and Earth Dragon is further divided into two sub forms- Boa and Tiger. Tiger form again has two children forms- Kicking Form and Wrist Form. Tiger Kicking Form is further categorized into 5 moves named as a First move, Second move, Third move, Fourth move and Fifth move. The user will have to start from one of the 5 moves present at leaf node, in order to learn Tiger Kicking Form and similarly to learn Tiger form, the user has to complete Tiger Kicking and Tiger Wrist form to start recording or learning of whole tiger form. The same way the user has to master Tiger Kicking and Tiger Wrist form before recording or learning whole Earth Dragon form, so this way the user has to traverse tree from the bottom level to root. After completion of each form, the user will be rewarded with Sashes, Badges etc. Earning sash and badge will unlock recording feature of its parent form so the user can record whole dragon form at one go.

Figure 25. Score Tree

The user is allowed to start with any form first but whichever form he/she starts with, it should be the leaf node of that form because all the forms at leaf node are unlocked by default for recording, so that the user can start recording with any one of the leaf node forms. For example, the user can start with any one of their 5 moves present at the bottom level of Tiger Kicking form. The user will earn points after every successful attempt up to a certain number of attempts, after which the user has to move to another form to keep earning points.

The above tree shows a square bracket with two values for each form, the first value in square brackets is the points awarded for every successful attempt and second value is the maximum successful attempt rewarded. For example, Earth Dragon has values [10000, 4] indicates 10000 points rewarded after a successful attempt of it and second value tells that the user will be

rewarded for first 4 successful attempts. For example, all moves present at the bottom level of Tiger form have been given 120 points for each successful attempt with a given limit of 5 successful attempts. Once a user reaches to 5 successful attempts, then the user won't get rewarded after any successful attempt on the same form, so the user has to switch to another form to earn points to go upwards in a tree. If the user wants to earn badges in Earth Dragon form then he/she has to complete following conditions.

Earth Dragon Badge= 4 successful attempts of Earth dragon form

Earth Dragon=Tiger Badge + Boa Badge

$$= (5000 * 4) + (5000 * 4) = 40000 \text{ points}$$

Tiger Badge = Wrist form sash + Tiger Kicking sash

$$= (2500 * 4) + (2500 * 4) = 20000 \text{ points}$$

Tiger Kicking sash=Move 1 sash + Move 2 sash + Move 3 sash + Move 4 sash + Move 5 sash

$$= 600 + 600 + 600 + 600 + 600 = 3000 \text{ points}$$

Move 1 sash= 5 successful attempts of Move 1 = Total $120 * 5 = 600$ points

Move 2 sash= 5 successful attempts of Move 2 = Total $120 * 5 = 600$ points

Move 3 sash= 5 successful attempts of Move 3 = Total $120 * 5 = 600$ points

Move 4 sash= 5 successful attempts of Move 4 = Total $120 * 5 = 600$ points

Move 5 sash= 5 successful attempts of Move 5 = Total $120 * 5 = 600$ points

So this way user has to traverse each selected form from its bottom level to root level.

8.3 UI for recording user's motion

Figure 26. User performing action like Expert

In the above figure, the user is trying to perform the same action, as expert videos showed on the left side, using comparison of motion capture formats. In this case, the user stance is perfect matching with Masters Motion, so the system did not indicate any warning of wrong movement because it was perfect movement.

Figure 27. Indication of wrong movement by red orange color

As you can see in above figure, the user skeleton is shown in red/orange color because of his/her wrong movement and violation of principle defined by master for a particular move. It shows bone in red/orange color which is not moving or rotating like a master. So here you can see that the user will quickly get an idea and he/she can change their motion to appear bone in blue as an

indication of right movement. So here, the user is getting this feedback coaching in fraction of seconds or in real time.

Figure 28. List of Violated principles with link address

Here you can see that the tab “Virtual Sifu Feedback on Principles” shows the list of violating principles with the given link to their explanation and demonstration page. So the user just needs to click on principle link, so he/she will be navigated to the explanation page of the particular form.

Figure 29. Virtual Sifu Suggestion

As you can see from the above figure, the user did not complete form correctly so the Virtual Sifu has suggested a user to focus on the principle “Hip Initiation” and “Spiral Transfer”

principle and also provided links of web pages having an explanation and demonstration video for learning these principles in details.

8.4 Virtual Sifu Workflow and Screenshots

Figure 30. Virtual Sifu Action Flow

The Flow Chart is the steps involved when a user comes to action page or hits “Action” button to initiate motion capturing of the selected form. As User hits “Record” button, the user can record

his motion of selected forms and as the data is generated for every 8/16 frames, it will be sent to DTW algorithm, then to the Neural Network to get score and joints are shown in red if violated, so this is called Real Time Comparison. Once done, the user hits “Stop Recording” button, the Virtual Sifu quickly gives feedback of user’s performance indicating percentage of each principle involved and overall percentage. If user performed so well then it is considered as a successful attempt and Virtual Sifu quickly tells the user remaining successful attempts to get a badge otherwise Sifu will quickly suggest to focus on weak areas and asks the user to record again.

Figure 31. Hip Initiation Violated

Figure 32. Spinal Transfer Violated

The above images are indicating users about the violated principle in real time by showing all bones in red belonging to violating principle.

Figure 33. Home Page

This is the home page of our website displaying features of the website such as Step by Step instruction, Get Immediate Feedback; deepen your Practice and Easy Navigation. The home page also displays the action gallery of Dr. Macramalla (Kung fu expert).

Figure 34. Cosmic Dragon Tree

This image depicts the tree structure of all forms i.e. navigating from Parent Form to Child Form.

Figure 35. Sash/Badge on a completion of the form

The above image shows that user earned a sash/badge on mastering particular form.

Figure 36. Virtual Sifu Explanation, Feedback, Recommendation tabs

The above images show the explanation of a particular form, benefits and cautions along with explanatory videos on right side, with step by step instructions.

Martial Art Principles Common to all forms

[Home](#) / Principles

Principle 1 - leg Stance

The stance is called horse stance because it mimics someone sitting in the saddle of a large horse, feet in the stirrups, bracing and squeezing the horses body with your inner thighs. The feet are parallel with the toes point straight forward. The knees are bent as comfortably close to ninety degrees as you can, with the knee cap coming over the point where the big toe and second toe meet, while you rotate the thighs outward.

Principle 2 - Opposition Recoil

The hip then travels forward. When it reaches as far forward as in comfortable, it then returns to its neutral position.

Kung Fu Principle
Demonstration &
Explanation

Figure 37. Biomechanics principle demonstration and explanation videos

The above screen shows the videos of expert demonstrating and explaining principles which are common to all cosmic dragon forms.

Figure 38. Expert Demonstrating tiger kicking and tiger wrist

The above screenshot displays karate expert video demonstrating the form given on right side. User can see his/her progress related to individual form on every page listed with the form name and video. Here you can see, the progress for Tiger Kicking Form is 75% and it also says, 1 more successful attempt remaining, so the user will quickly get the idea about his/her own status about particular form. It also shows that Tiger wrist form progress is 0% and 4 more successful attempts are required in order to master this form. If you observe carefully then you can see “Action” button on the Tiger Wrist form is hidden because the user has not completed child forms of Tiger Wrist form. So the user has to master all of its child forms in order to unlock the “Action” button so that users can record this whole form at once.

Figure 39. User Performance Page

The above image shows that users can see his/her own performance on attempted forms so far.

The performance page will also display lists of badges or sashes received so far.

FAQ Frequently Asked Questions

[Home](#) / [FAQ](#)

What is the final output of learning this art?

How can we protect ourself by injuries while performing this form?

At each level of form we have given step by step directions and tips to safe practice.

Can I send you my progress ?

What preacautions to be takens to avoid injuries?

How can I share my learing problems with you?

How can I suggest my feedback?

Do you have any other websites like this?

How can I observe changes in my body after this?

Can I get certificate If I complete this successfully?

Figure 40. FAQ page

The FAQ page is provided for the user to give answers to questions that users may have.

9. Conclusion and Future Work

This paper has implemented a website with a User Interface developed in such a way that it will give the user the summary of users' performance by comparing his motion with the pre-recorded motion on a web server in “real time” along with instructions for the next step and on successful completion or suggestion to improve if user fails to do that. The UI provides the user progress in the form of the “Performance Graph”. It also provides a different mode like record and replay which is used by the user to assess his/her performance. It focuses on sample movement patterns to do the “data quantity and quality analysis”. This paper has implemented a real time motion tracking, evaluation using artificial neural network technology and Zigfu JavaScript api while motion capturing. We have calculated some basic but important parameters from captured motion data, such as the rotation and translation of body segments and then analyzed motion flaws that hid behind it in real time.

The current application is generating User Skeleton in browser and is allowing the user to record and replay motion online and receive immediate feedback so that user need not install Kinect library, Visual studio on the computer. They just need to install browser plugin for Zigfu and that's it, they are ready to go just by connecting Kinect to laptop or pc. We still need to make an effort to provide more controls inside the Kinect capture screen such as Save, Load, Forward etc. to improve the user experience. We can provide a gaming environment by using Unity3d and make website “Motion Enabled” so that the users will wave a hand towards the left or right to go back and forth. We can also add voice commands feature to make this website more exciting.

10. List of References

- [1] *Dynamic time warping*. In Wikipedia. Retrieved May 15 2012 from
- [2] http://en.wikipedia.org/wiki/Dynamic_time_warping
- [3] Debra Howcroft and John Carroll, *A proposed Methodology for Web Development*, IS Research Centre, University of Salford.
- [4] Monk, S.R. And Sommerville, 'Schema evolution in OODBs using class versioning'. SIGMOD, Rec. 22(3):16-22.
- [5] Rath, T.M., & Manmatha, R. (20 June 2003). *Word image matching using dynamic time*
- [6] *Warping*. 2003 IEEE Computer Society Conference on Computer Vision and Pattern
- [7] Recognition (CVPR'03). Vol. 2, pp. (II-521- II-527). Do: 10.1109/CVPR.2003.1211511
- [8] Kinect. In Wikipedia. Retrieved May 15 2012 from
- [9] <http://en.wikipedia.org/wiki/Kinect>
- [10] Meredith, M. & Maddock, S. (N.D.). *Motion Capture File Formats Explained*. Retrieved
- [11] <http://www.dcs.shef.ac.uk/intranet/research/resmes/CS0111.pdf>
- [12] *Vocaloid Promotion Video Project (N.D.)*. Retrieved May 15 2012 from
- [13] http://www.geocities.jp/higuchuu4/index_e.htm
- [14] *KINECT for Windows (N.D.)*. Retrieved May 15 2012 from
- [15] <http://www.microsoft.com/en-us/Kinectforwindows/develop/overview.aspx>
- [16] *NI-Mate*. Retrieved May 15, 2012 from <http://www.ni-mate.com/>
- [17] Naveen Kumar Keerthy, *Virtual Kung fu Sifu with Kinect*, May 2012
http://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1250&context=etd_projects

- [18] B. PRABHAKARAN and CHUANJUN LI, S. ZHENG, *Segmentation and Recognition of Motion Streams by Similarity Search*, *ACM Transactions on Multimedia Computing, Communications and Applications*, Vol. 03.
- [19] David Catuhe, *programming with the Kinect for windows software development kit*, Microsoft Press, 2012, ISBN: 978-0-7356-6681-8.
- [20] Dumas & Wicke, J., *Influence of the volume and density functions within geometric models for estimating trunk inertial parameters*, *Journal of applied biomechanics*.
- [21] E.A. Hendriks, G.A. Ten Holt, M.J.T. Reinders, *Multi-Dimensional Dynamic Time Warping for Gesture Recognition*, 13th annual conference of the Advanced School for Computing and Imaging, June 2007.
- [22] Hemed Ali, ID: 22185631 , *Motion Comparison using Microsoft Kinect*.
- [23] Jing Sun, *Motion Learning with Biomechanics Principles*, San Jose State University, May 2013.
- [24] <http://channel9.msdn.com/coding4fun/Kinect/ZDK--Zigfu-Development-Kit--Commercial-Kinect-Development-library-for-Unity3D-and-JavaScript>
- [25] <http://www.Zigfu.com>
- [26] Licong Zhang¹, Jürgen Sturm², Daniel Cremers², Dongheui Lee. Real-time Human Motion Tracking using Multiple Depth Cameras.