City University of Hong Kong

Department of Computer Science

BSCCS/BSCS Final Year Project Report 2006-2007

06CS045

Dance Education System

(Volume 1 of 1)

Student Name : Chan Chun Pong
Student No. : 
Programme Code : BSCCS

Supervisor : Dr. LEUNG, Wing Ho Howard
1st Reader : Dr. HUANG, Chih-hao Scott
2nd Reader : Dr. KWONG, Tak Wu Sam
Student Final Year Project Declaration

I have read the project guidelines and I understand the meaning of academic dishonesty, in particular plagiarism and collusion. I hereby declare that the work I submitted for my final year project, entitled:

__________________________________________
Dance Education System

does not involve academic dishonesty. I give permission for my final year project work to be electronically scanned and if found to involve academic dishonesty, I am aware of the consequences as stated in the Project Guidelines.

Student Name: Chan Chun Pong 
Signature: ________________________________

Student ID: ______________________________ Date: __________________________
Abstract

A dance education system that works with a marker-based motion capture system was built. Students are able to learn dancing without presence of teachers. The system combines motion capture technology and OpenGL animations to provide students an intelligent and effective learning environment. The system plugs in a motion data handler and a real-time motion data handler from systems MocapViewer v1 and Simulation of Scenes with Multiple Densely-Interacting Characters (SSMDIC) respectively. Combining an evaluator for motion analysis, students can watch demonstrations of different dance motions and discover their errors in dance performance immediately without the presence of teacher. Survey results showed that most users agreed that our proposed system can help them to learn dancing efficiently. Experiment results showed that feedback given by our system can assist students to get improvement in dance learning.
Acknowledgement

I would like to start thanking my supervisor, Dr. LEUNG, Howard W H, who has been so supportive during this year. I really thank for his patience and helpful guidance. He spent lots of his valuable time to discuss the project with me. In addition, he gave me chances to acquire different experiences, such as co-authoring papers in International Conference on Ubiquitous Information Management and Communication 2007 (ICUIMC 2007) [1] and International Conference on Immersive Communications 2007 (IMMERSCOM 2007) [2] about the system and doing demonstration in different events held in City University of Hong Kong. These are really valuable experiences to me. I learnt how to work smartly and solve problem in efficient manner under his guidance. I am fortunate to be supervised by Dr. Leung.

Besides, I would like to thank Dr. Leung’s Research Assistant, Jeff Tang, for his kind support and helpful suggestions while I am doing the project.
# Table of Content

Abstract .......................................................................................................................... 3
Acknowledgement ......................................................................................................... 4
1. Introduction ................................................................................................................ 7
   1.1 Objective ............................................................................................................. 7
   1.2 Motion Capture Technology ............................................................................ 7
   1.3 Dance Training ................................................................................................. 9
2. Background Literature Review .................................................................................. 11
   2.1 Problem of Current Learning Method ............................................................. 11
   2.2 Prior work done by others ............................................................................. 12
      2.2.1 Research Projects ..................................................................................... 12
      2.2.2 Commercial Products .............................................................................. 15
   2.3 Prior work done at City University of Hong Kong ........................................ 16
      2.3.1 MocapViewer ............................................................................................ 16
      2.3.2 SSMDIC .................................................................................................... 18
3. System Design ......................................................................................................... 20
   3.1 System Description ............................................................................................ 20
   3.2 System Architecture ......................................................................................... 21
   3.3 Interface Design ............................................................................................... 23
4. System Implementation ........................................................................................... 25
   4.1 Technology ....................................................................................................... 25
      4.1.1 Languages Used ....................................................................................... 25
      4.1.2 Avatar Data .............................................................................................. 26
      4.1.3 Motion Data .............................................................................................. 26
   4.2 Platform and Tools ........................................................................................... 27
      4.2.1 OpenGL Utility Toolkit (glut) ................................................................... 27
      4.2.2 Microsoft Foundation Class Library (MFC) ............................................ 27
      4.2.3 EVa Real-Time Software (EVaRT) ............................................................ 28
   4.3 Implementation Flow ....................................................................................... 29
   4.4 Main Features ................................................................................................. 33
      4.4.1 Selection of lesson .................................................................................... 34
      4.4.2 Selection of avatar ..................................................................................... 35
      4.4.3 Demonstration of dance motion ............................................................... 40
      4.4.4 Practicing of dance motion ....................................................................... 42
      4.4.5 Evaluation of dance motion ..................................................................... 50
5. User Studies ............................................................................................................. 54
   5.1 1st Experiment ................................................................................................ 54
1. Introduction

This project is to build up a dance training system using motion capture technology. Virtual teachers in our system will demonstrate different dance motions. Users can observe and learn those moves. They can also perform what they learnt and our system will give them feedback. By studying the feedback, they will know what’s wrong with their performances and how to improve them.

1.1 Objective

The objectives of this project are based on the both advantages and disadvantages of existing dancing self-learning methods and methods from existing related research papers. A more detailed analysis on those advantages and disadvantages will be discussed in section 2. Objectives of this project are listed below:

1) Provide a system that works with motion capture system for self-learn dancing.
2) Demonstrate different dance motions by 3D virtual avatar.
3) Generate immediate feedback for users’ improvement.
4) Stimulate the interest and motivation of students.

1.2 Motion Capture Technology

The development of motion capture technology is rapid this year and it is now widely used in different aspects like making animation, sport training, etc. Fig. 1 shows a screen slot of a famous animation “Polar Express” [3].
There are several kinds of motion capture technologies. They are marker-based, electro-magnetic, optical, etc. Marker-based motion capture technology is the most popular one by its accurate measure. The marker-based motion capture system consists of some cameras shown in Fig. 2 and optical markers shown in Fig. 3. The basic idea is to use those cameras to track the movement of optical markers by detecting reflected light which emitted by cameras from markers. Markers are attached on users' suit as shown in Fig. 4. By analyzing movement track of markers, movement of the users can be obtained. The markers are relatively small, so there is negligible effect on users' movement. Users can customize positions of markers for different purposes. Usually, markers are placed at each joint and others are for post-processing when some markers are failed to be detected in the capturing process. Also, markers can be placed on face if capturing of facial expressions is required. In this project, motion capture technology is used in order to analyze students' motions.
Dancing is a popular activity in the world. It is a kind of arts and sports. Also, this activity does not require any special equipment, so people can perform it anywhere and anytime. Nowadays, people usually learn dancing by attending dance lessons or watching dance videos. In traditional lesson, teachers demonstrate dance motions and students imitate their motions. By following the feedback given by teachers, students improve their performance until they can perform same moves as teachers. Students
cannot learn without the presence of teachers in the approach of traditional lesson. However, teachers are not always available. So someone will learn by watching dance videos. They learn by observing and imitating motions demonstrated in videos. Nevertheless, this learning method is usually not effective. Detail discussion of these learning methods will be included in section 2.
2. Background Literature Review

In this section, current learning methods of dance will first be discussed. We will then discuss advantages and disadvantages of existing related systems/products.

2.1 Problem of Current Learning Method

At lessons, students learn dance motions by imitating teachers' moves. They first remember steps in dance motions and perform them. Teachers observe their motions and point their motions’ errors out. By following teachers' feedback, students can correct those errors and get improvement.

The learning method is that students have to remember those moves and then improve their performance by following teachers' feedback. However, lessons are time-limited and costly. Also, teachers are not always possible to demonstrate until all their students remember all moves. For those students that cannot remember all moves, they may waste the lessons. Moreover, teachers may not be able to take care all of their students.

Some people may choose to learn themselves by watching video. As the view point in a video is fixed, a clear demonstration may not always be available. Then some moves may be neglected. On the other hand, they can only practice by themselves and do not know how well actually they can perform. So it is difficult for them to improve as they cannot find their own mistakes out. That's why we need a system that can let people learn effectively and can work without the presence of teachers.
Table 1 shows the summary of comparison between two learning methods mentioned before and our suggested system. From the table, we can see our approach is to adopt advantages and avoid having same disadvantages of two existing learning methods.

<table>
<thead>
<tr>
<th></th>
<th>Attending lesson</th>
<th>Watching Video</th>
<th>Dance Education System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Require presence of teacher</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Require Special Equipment</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Demonstration of dance motion</td>
<td>- Time-limited</td>
<td>- No time-limited</td>
<td>- No time-limited</td>
</tr>
<tr>
<td></td>
<td>- Changeable view point</td>
<td>- Fixed view point</td>
<td>- Changeable view point</td>
</tr>
<tr>
<td>Evaluation of dance motion</td>
<td>Feedback provided</td>
<td>Feedback not provided</td>
<td>Feedback provided</td>
</tr>
</tbody>
</table>

Table 1 Comparison between different learning methods

2.2 Prior work done by others

Before the development of our system, we have studied limitations and success elements of existing systems. In this section, we will discuss those findings and what our system had adopted and improved.

2.2.1 Research Projects

Learning dancing by watching videos of professional dancers is not effective [4]. Nakamura et. al. suggested a system built with active vibro-devices and a robotic screen. The vibro-devices will vibrate at each rhythm and users will know when they have to move. Robotic screen moves according to translation in dance motions. Real-
time animations of users are shown on the screen. So, users know how badly they are performing. However, this is not enough for users to know how they can improve. It is because it is difficult for users to observing and analyzing their own motions while they are dancing.

In order to help users to improve, suitable feedback has to be given to them. To give feedback for users’ performance, data has to be collected and then we can do some analysis on their performances. EVa Real-Time (EVaRT) system is a marker-based motion capture system for obtaining 3D motion data [5]. Markers attached to suit are light (dia., 25mm) and will not disturbed users’ movement.

To analyze dance motion, some recent research suggested the use of angles at joints. Characteristic features of Japanese Dance furi are quantified by using four spatial indices [6]. Two of these spatial indices are Kamae and Koshi which are obtained by angles at joints. Four indices show significant difference between dance motions of experts and beginners. A gesture recognition engine of a dance system also made use of joint angles in the analysis [7]. The engine recognized gestures by torso orientation and joint angles. For other motions like martial arts, analysis by using angles at joints is also often. Kwon et. al. designed a Taekwondo training system built with two kinds of sensors, visual sensors and body sensors at forearm [8]. By using body sensors attached to users’ forearm, the system could obtain two angular values pitch and roll in a motion and use the values to analyze moves in Taekwondo. Some other approaches are also suggested to be used in motion analysis. Hachimura et. al suggested the use of theorem Laban Movement Analysis (LMA) for analysis of dancing movement [9]. From the theorem, they suggested that dance motion can be analyze by the kinetic energy, rate and direction of movement. However, this method
was failed to identify wrongly moved body parts in a wrong motion.

Various kinds of feedback were used in different motion training systems. Chua et al. suggested a physical task Tai Chi training system built with motion capture technology [10]. A virtual environment is setup and a virtual representation of student and a virtual instructor are placed there. He/She learns by mimicking motions of the instructor. Movement of his/her virtual representative was the only kind of feedback provided by this system. Hachimura et al. integrated motion capture technology and mixed reality technology in a dance training tool [11]. Users wear head mounted display devices and watch expert’s dance motions which will be displayed ahead of them like a “Ghost”. They learn moves by moving until “Ghost” images are always overlapped with their bodies, in other word, they moves as template motions. However, it is not effective to observe and imitate at the same time. In our design, users can dance as usual while our system will observe their performance and give them feedback.
2.2.2 Commercial Products

Fig. 5 Dancing Video Game Dance Dance Revolution (left), Para Para Paradise (right)

Some TV games also involve dance motion analysis (Fig. 5) [12][13]. One of the most popular games, Dance Dance Revolution (DDR) works with a foot panel with different step zones. Following signals shown on screen, DDR players have to step on the correct step zone in time to get higher scores. Intermediate feedback is given in term of grades like “Marvelous”, “Good”, “Almost”, “Ok”, etc and Overall feedback is given in term of ranks A, B, C or below. Para Para Paradise (PPP) is another famous game and it is similar to DDR except that PPP determines player’s performance by movement of arms. These kinds of designs are not suitable for dance training because of two reasons. First one is that dancing involved movement of the whole body, considering neither only legs nor arms is not enough to analyze dance motion. Second one is that feedback contains only grading is not sufficient for users to improve their
motions.

Despite of those limitations, there are also things that we can learn from. The animations in video games are attractive. The avatars are beautiful and make players more willing to observing their movements.

2.3 Prior work done at City University of Hong Kong

Some research projects related to motion analysis and usage of motion data are done by academic staffs at City University of Hong Kong. They are discussed below.

2.3.1 MocapViewer

Fig. 6 Screen slot of Mocap Viewer v.1

MocapViewer is a viewer of Biovision Hierarchical Data (BVH) files (a motion data
file format) developed by Mr. Edmond Ho which is a research assistant of Dr. Taku Koruma. Fig. 6 shows a screen slot of this viewer. In this viewer, motions stored in BVH files are used to animate a skeleton-liked avatar.

View point can be changed by using buttons on control panel. However, the design of this viewer is not suitable to demonstrate dance motions for dance learning because of following reasons:

1) BVH files that can be loaded are fixed and users cannot load another files. This is inconvenient and users may have their own data files instead of template files.

2) Design of view point changing function is not user-friendly. It is not easy for users to changing view by clicking buttons.

3) Users cannot move demonstration forward or backward. Demonstration cannot start from middle of dance motions. So, each time users want to view a certain part of moves, they have to view from the beginning to end until that part is shown.

4) Fixed avatar may get users boring and the skeleton-liked avatar may be too horrible for young children.
2.3.2 SSMDIC

SSMDIC is a dance education system developed by Mr. Henry Li who was research assistance of Dr. Taku Koruma. This system works with motion capture system EVaRT to capture user’s motion and calculates overall similarity between users’ motions and template motions (Fig. 7). There are some limitations that make this system not enough for education purpose and they are listed below.

1) No demonstrations of dance motions are provided. Demonstration can only be viewed during evaluation stage, a stage that users perform dance motions and system evaluates their motions. Even if users can watch demonstration at evaluation stage, view point is fixed and no function is provided to assist observation of motions.

2) Number of dance lessons provided for users is limited. Users cannot add more lessons to the system.

3) Similarity percentage is not an effective feedback. From similarity percentage,
user cannot extract useful information for improvement, for example, which moves are wrong, why a move is wrong, etc.

4) View point is fixed to that avatar will dance at the direction facing users. It is not convenient as what we see on the screen is actually at different direction. For example, when virtual teacher moves his/her left arm from his right hand side to left hand side, we will see his/her arm at our right hand side move from our right hand side to our left hand side. Although the system uses different color for left and right arm, users may not get use to distinguish two arms by using color quickly.

5) Avatar used for virtual teacher is made by cube shape body segment. It makes users difficult to observe its motion as rotation of arms or legs cannot be clearly shown.
3. System Design

As mentioned in section 2, approach of traditional dance lesson is effective if teachers are always available. So, our system design will follow the arrangement of traditional dance lesson and our system will act as a dance teacher. In this section, the system design will be discussed.

3.1 System Description

The basic principle of our system design is to provide same functions as a real dance teacher. These are demonstrating dance motions and evaluating students' performances. So in this project, the system has two main functions: demonstration and evaluation.

Fig. 8 System flow
Fig. 8 shows the system flow. Dance motions of experts are captured and stored as template lessons in motion data files. During demonstration, template lesson is loaded from file to our system and virtual teacher will demonstrate the motion. Students can watch the demonstration in different view, different speed and at different instant of a dance motion in order to clearly observe the motion. During evaluation and practicing of dance motion, data of students' dance motions are obtained by motion capture system and template motion data will be loaded from motion data files. Data is used for analysis, real-time animation of student’s virtual representative and is also stored. Feedbacks are displayed in form of similarity score and also slow-motion replay of users’ performances with body parts moved wrongly being marked out.

### 3.2 System Architecture

![Dance Education System](image)

Fig. 9 System Architecture

Fig. 9 shows our system architecture. The system is basically divided into three parts: Real-time data handler, Evaluator and Renderer. The real-time data handler is
designed to obtain and process motion data from motion capture system. The evaluator is designed to analyze students' motion and motion data for analysis is obtained from template motion data file and real-time data handler. The Renderer is designed to render animations and animations are rendered based on motion data from evaluator and avatar data from model data file.

The system in this project is developed on top of existing works. The detail information will be discussed below.

a) MocapViewer v1
The evaluation part of our dance education system is partly adopted from Mocap Viewer v1. The adopted class called “bvhobject” can extract information from motion data file BVH. The extracted information includes frame number, frame rate and joint coordinates at each frame. The types of BVH files that are allowed by “bvhobject” are restricted. The detail will be described in section 4.4.1.

b) SSMDIC
The real-time data handler and interface of our dance education system is partly adopted from SSMDIC. The part that adopted from SSMDIC is responsible for connecting motion capture system EVaRT and getting markers coordinates when request is raised. The interface of SSMDIC is built by Microsoft Foundation Class (MFC). The interface is used in our system with changes made to menu items.
3.3 Interface Design

A well interface design can shorten the learning time, stimulate users' interest and make users' operation time more efficient. Our interface is designed in these directions.

Fig. 10 Menu of our interface

Fig. 10 shows our interface design. Menu items are sorted logically by their representing functions. Normally, users first select a dance lesson. And then they can select a favorite avatar. After this, they watch demonstration performed by virtual teachers. While they are watching the demonstration, they may change viewing speed. After watching the demonstration, they practice dance motions under observation of our system. So, menu items are arranged in this order. It is for users’ easy memorization.
The menu is put at the top of the interface. At the bottom, it is the main window which shows all the animations. Fig. 11 shows an animation shown at main window. Except the animation, a slider bar will be shown at the main window when a function "Demonstration" is using by users. Fig. 12 shows a slider bar at main window. The slider bar is used to control the demonstration process which will be described at section 4.4.
4. System Implementation

After considering the system design, such as system architecture, functions and interface design, implementation of the system is started. Issues related to system implementation are going to discuss in this section.

4.1 Technology

Suitable technologies have to be selected in order to fulfill the requirements of the system. Furthermore, using correct technologies would result in rapidly development of the system.

4.1.1. Languages Used

Choosing a suitable computer language can make a system more portable. It also favors the system development when the development involves different programmers.

*Open Graphics Library (OpenGL)*

OpenGL is a popular 3D graphic application that is supported by most of the platforms. Complex 3D scenes can be rendered by simple primitives defined in OpenGL. 3D animations in our system are rendered by using OpenGL. Rendering takes place at PC and display on the screen placed in front of users.

*C++*

C++ language is a popular computing language that almost all programmers know. Most of the programming software support C++ language coding. Our system is
developed in C++ language. It will be easier for different programmers to modify and to understand the coding.

4.1.2 Avatar Data

*Virtual Reality Modeling Language (VRML)*

VRML is stored in a text file format with .wrl extension, where different 3D objects are formed by groups of polygons that are defined by specifying their coordinates, color, shininess, transparency, etc [14]. Each object contains its own set of coordinates, index of coordinates in polygons, set of normal vectors and index of normal of each polygon.

4.1.3. Motion Data

*Biovision Hierarchical Data (BVH)*

BVH is a popular motion file format. It is divided into two parts. The first part of file contains hierarchy of initial pose. In the hierarchy, a root node is first defined and then its connecting nodes and offset between them are stated. New nodes are defined with offset between them and their neighbor nodes are stated until an end site is met, e.g. hand, foot. The second part of file contains motion information. The motion is stored frame by frame. Number of frame and frame rate will be stated. In each frame, a posture is stored. The posture is described as rotation of different body segment regarding to initial pose. In other words, each frame is independent with each others.
4.2 Platform and Tools

Our system integrates several platforms and tools, OpenGL Utility Toolkit (glut), Microsoft Foundation Class Library (MFC) and Eva Real-Time Software (EVaRT). MFC is for interface control and consists of all event handlers, for example, mouse-click. EVaRT captures users’ motions and when our system gets the data, analysis is performed. The data is also used to render real-time animation. The rendering by glut is passed to MFC and is displayed at the main window of our interface.

4.2.1 OpenGL Utility Toolkit (glut)

Glut is a library of OpenGL utility [15]. Functions for drawing of geometric primitives like line, polygons are provided. Functions are used during rendering of users’ virtual representative and virtual teachers.

4.2.2 Microsoft Foundation Class Library (MFC)

MFC is a Microsoft library that includes part of Windows API in C++ class [16]. The library helps defining, controlling and managing windows object. The interface of our system is built by using MFC.
4.2.3 EVa Real-Time Software (EVaRT)

Fig. 13 EVaRT screenslot

EVaRT is an interface between users and motion capture system. Users can use EVaRT to setup and control motion capture system. Data obtained is post-processed and edited by using EVaRT. Before the use of Dance Education System, users have to setup motion capture system by using EVaRT. When the training starts, our system can retrieve real-time motion data from EVaRT. Fig. 13 shows a screen slot of EVaRT. The main window shows coordinates of detected markers. As shown in figure, the markers are linked up to form a human skeleton-liked model. However, the coordinates of markers are not exact coordinates of joints. Fig. 14 shows a marker placed at user’s elbow and red circle marks the actual position needed. As we can see, it is not exactly equal to coordinates of elbow. So, some post-processing is required.
To communicating with EVaRT, an EVaRT header file “EVaRT.h” is included. In EVaRT header files, functions for communicating with software EVaRT are provided. Two essential functions are for obtaining markers coordinates. First, connection between EVaRT and our system is required to build up. Then we use certain functions to obtain markers coordinates which are stored in an array.

### 4.3 Implementation Flow

As mentioned in **section 3.1**, our system contains two main functions, demonstration and evaluation. In a real dance lesson, demonstration is always ahead of evaluation. The order of implementation depends on this nature. The following part gives a
general idea of the implementation flow. Fig. 15, Fig. 16 and Fig. 17 shows three stages in implementation flow. Functions for demonstration were first developed and then functions for evaluation were developed. Tasks involved in the development of our system are first listed below (Table 2).

<table>
<thead>
<tr>
<th>Function Involved</th>
<th>Demonstration</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>● Write a WRL file parser</td>
<td>● Calculate actual joint position from EVaRT data</td>
</tr>
<tr>
<td></td>
<td>● Plug-in parser into Mocap viewer v.1</td>
<td>● Compare between two posture</td>
</tr>
<tr>
<td></td>
<td>● Add function “avatar changing” into Mocap Viewer v.1</td>
<td>● Display result</td>
</tr>
<tr>
<td></td>
<td>● Add new functions (selection of BVH files, selection of avatars, moving forward and backward)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● Migrate part of Mocap Viewer v.1 to SSMDIC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● Change SSMDIC to OpenGL rendering</td>
<td></td>
</tr>
</tbody>
</table>

Table 2 System task list
Fig. 15 Stage 1

Fig. 15 shows stage 1 of our system’s implementation flow. In stage 1, the implementation is concentrated on functions for demonstrating dance motions. More functions are added to Mocap Viewer v.1 like changing avatars, moving forward and backward, etc.
Fig. 16 Stage 2

Fig. 16 shows stage 2 of our system’s implementation flow. In stage 2, we concentrate on the migration of functions in Mocap Viewer v.1 to SSMDIC and also modifying functions for evaluating dance motions. At the end of stage 2, development of dance education system was almost completed and experiments were carried out.
Fig. 17 Stage 3

Fig. 17 shows stage 3 of our system’s implementation flow. After the experiments were carried out at stage 2, we analyzed the result and indicated how our system can be modified. After those modifications, we carried out the second series of experiment. At last, the project was finished.

### 4.4 Main Features

In this section, we will describe main features of our system and how actually they work. Below is a list showing all features of our system:

- Selection of lesson
- Selection of avatar
- Demonstration of dance motion
- Practicing of dance motion
- Evaluation of dance motion
4.4.1 Selection of lesson

Students can choose different dance lessons. Lessons are chosen by loading different BVH files. Fig. 18 shows how a lesson is selected. Our system provides some template lessons for users. Users can also prepare another BVH files as lessons. However, the format of those BVH files should be suitable for our system and the practice part of these lessons is slightly different with template lessons. The difference will be discussed in section 4.4.4.

![Fig. 18 Screen slot of selecting lesson](image)

When a BVH files is selected, we will first check if the BVH file is in suitable format. BVH files may have different hierarchies. Suitable hierarchy of our system contains 24 body segments and a sample is shown at Fig. 19. This hierarchy is already enough to describe all the limb movement of human. After identifying a suitable BVH files,
our system will retrieve information of the dance motion stored in the file, for example, frame number, frame rate, etc. Also, calculation of coordinates of joint at each frame will be performed by using rotation angles and offset of body segments.

![Body hierarchies with 24 segments](image)

Fig. 19 Body hierarchies with 24 segments

### 4.4.2 Selection of avatar

Our system provides different looks of virtual teachers for users. Users can choose their favorite avatars by selecting WRL files stored in the system folder. A human-like avatar favors users in the process of observing virtual teachers’ motions. Fig. 20 shows a human-like avatar and Fig. 21 shows an avatar used in SSMDIC that is
formed by cylinder-shaped body segments. It is obviously that the movement and the rotation of body segments of human-like avatars are easier to observe. Moreover, changing of avatar makes our system more interesting and this can stimulate users’ interest.

Fig. 20 Different avatars provided by our system

Fig. 21 Avatar in SSMDIC

In a WRL file, there are definitions of different objects. Each object has its own attitudes like shininess, transparency, color, etc. Objects are formed by polygons which are defined by vertexes. A body segment in a model data may be defined by several objects in the WRL file. For example, eyes, hair, mouth are always defined as different objects and head is considered as one body segment. So, the first thing done with selected WRL file is to identify which body segments each object belongs to. Fig. 22 shows 18 segments defined in the model data. The identification can be done by
searching name of objects which is defined in WRL files.

Fig. 22 Body segments in a model data

The second thing to do is to find reference point for each body segment. Reference point is needed for placing and rotating body segment at correct position during rendering. As shown in Fig. 22, some body segments are overlapped. To find a reference point, we have to consider overlapping area in order to maintain body segments keep overlapping during rotation. As observed, body segments except left and right shoulders are placed vertically. Fig. 23 shows how body segments are rendered by using reference points. Our approach is to render a segment at its reference point, translate to reference point of next segment and then render next segment and so on.
In order to find out reference point, we first find out maximum and minimum x, y and z coordinates among component objects for each segment. For body segments except left and right shoulders, following equations are used for calculating x, y and z coordinates of reference point:

\[
x_{\text{ref}} = (x_{\text{max}} + x_{\text{min}})/2
\]

\[
z_{\text{ref}} = (z_{\text{max}} + z_{\text{min}})/2
\]

\[
y_{\text{ref}} = y_{\text{max}} - (y_{\text{max}} - y_{\text{min of upper segment}})/2
\]

For left and right shoulders, following equations are used.

\[
x_{\text{ref}} = x_{\text{min}} \text{ for left shoulder}, \quad x_{\text{ref}} = x_{\text{max}} \text{ for right shoulder}
\]

\[
y_{\text{ref}} = (y_{\text{max}} + y_{\text{min}})/2
\]
The third thing is to generate OpenGL codes regarding to information given in WRL files. There are two kinds of information for an object defined in the WRL file, material information and geometry information. The material information includes color, ambient intensity, shininess, and transparency. These can be rendered easily by using OpenGL commands like `glColor()`. The geometry information is represented by an array of coordinates, a polygon-point index list, an array of normal and a normal index list. The polygon-point index states which coordinate in the array of coordinates is belonged to a vertex of a polygon. Indexes of different polygons are separated by value -1 in the array. For example, for a polygon-point index array [0, 1, 2, -1, 1, 2, 3, -1] and coordinates array [0, 0, 0, 1, 1, 1, 2, 2, 2], we can identify two polygons here as there are two -1 in the array. The first polygon is a triangle with three vertexes and their coordinates locate at position 0, 1 and 2 in the array of coordinates, these are (0, 0, 0), (1, 1, 1) and (2, 2, 2). For the array of normal and the normal index list, each normal index refers to each polygon-point index. That is, coordinates stated in 1st polygon-point index will have the normal which is pointed by 1st normal index and so on. By using this relationship, we can use OpenGL code to represent it. For the previous example, our system will generate the following code.

```c
...  
gleBegin(GL_TRIANGLES);  
gleVertex3f(0, 0, 0);  
gleVertex3f(1, 1, 1);  
gleVertex3f(2, 2, 2);  
gleEnd();
...```

\[ z_{ref} = \left( z_{max} + z_{min} \right) / 2 \]
4.4.3 Demonstration of dance motion

After selection of lessons and avatars, users can use our demonstration function to watch virtual teacher’s demonstration. In the demonstration, virtual teacher will appear in the main window and perform selected dance motion. Meanwhile, users are allowed to have following operations:

(a) Changing View point

View point can be changed by used hot keys and the mouse. Hot key list is shown below in Table 3. Below the list, there are some samples to show how view point is being changed. Users’ actions activate keyboard and mouse event handler. Decision is made according to users’ actions. Changes will be made before rendering of next frame. Fig. 24 is the normal view point. Fig. 25 and Fig. 26 show how view point is zoomed in and out. Fig. 27 shows how view point is being rotated. Fig. 28 shows how view point is being translated.

<table>
<thead>
<tr>
<th>Hot Key</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl + Left Click + Mouse moving</td>
<td>View Point Rotating</td>
</tr>
<tr>
<td>Right Click + Mouse moving (up/down)</td>
<td>Zooming in and out</td>
</tr>
<tr>
<td>Middle Click + Mouse moving</td>
<td>View Point Translating</td>
</tr>
</tbody>
</table>

Table 3 Hot key list for demonstration operation

Fig. 24 Normal view
(b) Moving demonstration to specific time instant

A slider bar shown in Fig. 29 is provided for users’ convenient. When users drag the bar, they can make the demonstration going forward and backward. The ratio of movement of bar is equal to the ratio of moving the demonstration. That is, if users move 50% rightward, the demonstration will move 50% forward. Meanwhile, the changing is immediately animated and displayed at main windows. For example, when a user wants to move demonstration backward to some instant, he/she can move the bar leftward until that instant is shown at main window.
In order to observe demonstration more easily and clearly, users may want to change speed of demonstration. Our system provides four speeds for users to change: 1x, 0.5x, 0.25x and 0.01x. Fig. 30 shows the selection of the speed 0.01x.

4.4.4 Practicing of dance motion

Sometimes moves that we learnt from watching demonstration cannot be really performed. So, after watching demonstration, users have to practice in order to really learn the moves. We provide a function called “Practice”. Details are discussed below.
Key postures are some postures that are more representative in whole dance motions.

Fig. 31 shows sequence of dance moves and first and last postures are key postures. If key postures A and D are correctly performed, when users’ left arm is at position x in posture A to position y in posture D, postures B and C are usually to be corrected. The reason is that positions of left arms in postures B and C are along track from position x to y. As key posture is so important in a performance, our system provides the “Practice” function for users to be more familiar with key postures in dance motions.

Our idea is to view key postures as the check points. Virtual teacher will demonstrate from start posture to end posture of dance motions with pause at each check point. When virtual teacher stops at a certain posture, users have to perform that posture correctly in order to make virtual teacher moves again.

While users try to perform the same posture as their virtual teachers, they can see their virtual representatives on the screen. The virtual representatives are real-time animated. The body segments of the characters are either in color of yellow or red. Yellow means that body segment moves correctly while red means it moves wrongly. Fig. 32 shows an example of a student using “practice” function and Fig. 33 shows his virtual representative and virtual teacher on the screen. As shown, the right upper arm
of virtual representative is in red color. So, that student has to correct his movement of his right upper arm. From this kind of representation, users can easily correct their postures.

Fig. 32 A student is using “practice” function

Fig. 33. Screen Slot during student in Fig. 32 using “Practice” function
As mentioned in section 4.4.1, there is difference between template motion data and motion data prepared by users during practicing. When the template motion data is loaded, key postures are already defined. So in the practice function, virtual teachers will stop at each key posture. Meanwhile, for the motion data prepared by users, key postures are not defined and our system can only generate check points disregarding key postures in actual movement. In our design, we set check point as posture at each 1 second interval as check point.

Fig. 34 Decision flow of function “Practice”

Fig. 34 shows decision flow of practice. Before the rendering of animation for each
frame, we check if the frame is a frame that stores key posture. If not, we render it. If it is a key posture, we stop animation of virtual teacher with real-animation of student’s virtual representative continues. And then our system compares student’s posture with virtual teacher’s until they are the same and the animation continues. After each time of comparison, we render student’s animation with wrongly moved body segments in red and those with right moves in yellow.

To check if two postures are the same, we calculate similarity values between movements at each joint of two postures. If all the similarity values are higher than a threshold, these two postures are considered to be the same. The similarity value is calculated by using joint angle. The processes of finding similarity value will be discussed below.

a) Normalization before comparison

Before the discussion of how normalization works, we first talk about why normalization is required. A posture is represented by coordinates of different joints. As these coordinates are absolute world coordinates, they can not be used directly. As illustrated in Fig. 35, two postures are actually the same but they are facing different directions. We now consider a situation that these two postures are compared without any processing and comparison of movement at right shoulder is taken place. From the coordinates of right shoulders and right elbows of two postures, the system will consider that one upper arm points to left and another upper arm points to right as the coordinates are toward negative x-axis ad positive x-axis respectively. The movement at the right shoulder is claimed to be wrong. Obviously, this result is incorrect. Our system has to make two postures facing the same direction before the comparison in order to get correct result. This kind of processing is required for comparison at
different joints and normalization is designed for this purpose.

![Fig. 35 Two identical postures in different direction](image)

Fig. 35 Two identical postures in different direction

![Fig. 36 T-pose](image)

Fig. 36 T-pose

We will now talk about how normalization works. In our system, a joint coordinate “Lower Torso” is chosen as central point of a posture. The comparison starts from this
central point eventually to each end site of limb, e.g. hands. We firstly define a posture called “T-pose” (Fig. 36) which is a posture used for calibration of motion capture system. This posture is now used as a template to normalize the movement of different body segments. Before each comparison at joints, one of two segments connected by the joint has to be at the same direction as that segment in T-pose. Fig. 37 illustrates change made to posture in Fig. 35 after normalization at knee and comparison at knee is ready to be performed. The process is done by calculating Euler angles between target segment and corresponding segment in T-pose and then rotates target segment by Euler angles got. Euler angles between two vectors are defined as how one vector rotates among x, y, z axis in order to be equal to another vector. After this process, it is now ready to compare the movement of two postures at normalized joint. Table 4 shows specific segments that will be normalized during comparison at different joints.

Fig. 37 posture with normalization at thigh refer to Fig. 35
Joints | Locked Segment
---|---
Lower body | Upper body
Upper body | Lower body
Neck | Upper body
Left Shoulder | Upper body
Left Elbow | Left upper arm
Left Wrist | Left lower arm
Right Shoulder | Right upper arm
Right Elbow | Right upper arm
Right Wrist | Right lower arm
Left Thigh | Lower body
Left Knee | Left thigh
Left Ankle | Left lower leg
Right Thigh | Lower body
Right Knee | Right thigh
Right Ankle | Right lower leg

Table 4 Normalization table

**b) Calculation of similarity value**

After normalization, the similarity value $S$ at joint is got by using following formula:

$$
S = \frac{(x_b - x_a) * (x_d - x_c) + (y_b - y_a) * (y_d - y_c) + (z_b - z_a) * (z_d - z_c)}{\sqrt{(x_b - x_a)^2 + (y_b - y_a)^2 + (z_b - z_a)^2} \cdot \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2 + (z_d - z_c)^2}}
$$

where $a$ is joint coordinates of one posture, $b$ is joint coordinates of another posture, $c$ is coordinates of end site of comparing segment and $d$ is coordinates of end site of another segment.

c) Using similarity value

After calculation of similarity value at each joint, we will get an array storing value of 15 joints. For a correct movement at a joint, corresponding value should be higher
than 0.9. For same postures, the movements at all joints should be correct. Otherwise, they are different postures.

### 4.4.5 Evaluation of dance motion

After practicing different moves, users can use “Evaluation” function to let our system evaluate their performance and give them feedback.

There are two kinds of feedback given by our system. The first kind of feedback is similarity rate. Movement at each joint will have a similarity score and the overall similarity score is also given. By studying similarity score, users can know movement at which joints are worse. Fig. 38 shows an example of this kind of feedback. From this feedback, we can identify movement at leg is the worst as the values of thigh and ankle are the lowest.

![Feedback showing similarity score](image)

**Fig. 38 Feedback showing similarity score**

<table>
<thead>
<tr>
<th>Joint</th>
<th>Similarity Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>neck</td>
<td>41%</td>
</tr>
<tr>
<td>lshoulder</td>
<td>6%</td>
</tr>
<tr>
<td>rshoulder</td>
<td>9%</td>
</tr>
<tr>
<td>lelbow</td>
<td>7%</td>
</tr>
<tr>
<td>relbow</td>
<td>8%</td>
</tr>
<tr>
<td>lhand</td>
<td>53%</td>
</tr>
<tr>
<td>rhand</td>
<td>27%</td>
</tr>
<tr>
<td>lthigh</td>
<td>7%</td>
</tr>
<tr>
<td>rthigh</td>
<td>1%</td>
</tr>
<tr>
<td>lknee</td>
<td>36%</td>
</tr>
<tr>
<td>rknee</td>
<td>30%</td>
</tr>
<tr>
<td>lankle</td>
<td>0%</td>
</tr>
<tr>
<td>rankle</td>
<td>0%</td>
</tr>
</tbody>
</table>

**overall: 17%**
The second kind of feedback is slow-motion reply. Motion data of virtual teacher and student is stored and also frame rate is stored. By twice the frame rate, a slow-motion replay can be animated. In order to indicate correctness of movement at joints of students in different frame, color of body segments of student’s virtual representative is selected regarding to their similarity rates at that frame. Fig. 39 shows a screen slot of slow-motion replay. There are four colors shown. Red, green, blue and yellow mean “wrong”, “nearly wrong”, “almost correct” and “correct” respectively.

Fig. 39 Slow-motion replay

*Generation of similarity score*

Just like animation can be produced by sequence of pictures, motion can be described as sequence of postures. In our design, we compare users’ postures simultaneously during performance with virtual teachers’ postures at that time instant. Method for calculating similarity value was discussed at section 4.4.4, the value will be used in a
different way in “Evaluation” function.

After two postures are evaluated, result will be stored in a one dimension array. As now sequence of postures are compared, so finally a two dimensions array will be got. The number of record is equal to frame number and in each record there are 15 entries which represent 15 body segments.

The similarity values are used to get similarity scores regarding to Table 5. 0.9 is the threshold that movement is assumed to be correct and full mark 1 is given. For similarity value that is greater than 0.7 but is smaller than 0.9, a little bit of mark will be given. For total, a totally correct posture will get 15 marks.

<table>
<thead>
<tr>
<th>Similarity value $x$</th>
<th>Similarity score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.9 \leq x \leq 1.0$</td>
<td>1</td>
</tr>
<tr>
<td>$0.8 \leq x &lt; 0.9$</td>
<td>0.4</td>
</tr>
<tr>
<td>$0.7 \leq x &lt; 0.8$</td>
<td>0.1</td>
</tr>
<tr>
<td>$0 \leq x &lt; 0.7$</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5 Score reference table

For each entry in two dimensions array, we get similarity score regarding to score reference table. At last, we get total score for each entry and divided it with number of records. As each entry is corresponded to each joint, average similarity score of all joints are got. Also, overall similarity score is got by summing up average similarity scores of joints and divide it by 15, the number of joints. These values are the value shown in Fig. 38.

The similarity values are also used for coloring of virtual representative’s body
segments following color-lookup table below.

<table>
<thead>
<tr>
<th>Similarity value $x$</th>
<th>Color of body segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.9 \leq x \leq 1.0$</td>
<td>Yellow</td>
</tr>
<tr>
<td>$0.8 \leq x &lt; 0.9$</td>
<td>Blue</td>
</tr>
<tr>
<td>$0.7 \leq x &lt; 0.8$</td>
<td>Green</td>
</tr>
<tr>
<td>$0 \leq x &lt; 0.7$</td>
<td>Red</td>
</tr>
</tbody>
</table>

Table 6 Color lookup table
5. User Studies

In order to prove the validity of our system design and to get user feedback for future development, I have invited 19 subjects to join the experiment. The experiment divides into two parts, one is for testing system usability and one is for testing positive effect of feedback given in dancing training. Details of these two parts will be discussed in this section.

5.1 1st Experiment

11 subjects have participated in this experiment. They are divided into three groups. Each group trained dancing with different systems providing format in different format. And then they were asked to fill in a questionnaire with 10 questions. The learning progresses of users were recorded for analysis. The selected dance motion was Hip-hop and varied about 30 seconds.

Background of users

All of the subjects are aged from 20 to 25. They all have no dancing experience.

Method in Experiment

Each user was asked to observe dance motion through our system within 30 minutes. They then trained by evaluation mode for five times. After each performance, they watched feedback given to them and then started the next practice. Different groups had different kinds of feedback.

Feedback for group 1 was overall similarity percentage of dance motion between theirs and templates (Fig. 40). Feedback for group 2 was similarity percentage of
movement at each body joint between theirs and templates’ which is first kind of feedback in evaluation mode of our system (Fig. 38). Feedback for group 3 was similarity percentage of movement at each body joint between theirs and templates’ and a slow-motion replay of users' motions with similarity rate of joints being shown by different colors (Fig. 39). This feedback was equal to feedbacks given in evaluation mode of our system. Their training progress was shown regarding to overall similarity rate in five sessions of training.

Fig. 40 Feedback for Group 1
Experiment Result & Discussion

Records of users from different groups are shown below.

Fig. 41 Experiment record for group 1

Fig. 42 Experiment result for group 2
Fig. 43 Experiment result for group 3

Fig. 41, Fig. 42 and Fig. 43 are experiment records for subjects respectively of group 1, 2, and 3. As expectation, similarity rate of group 3 users will grow in greatest rate, and then group 2, and group 1 the less. However, we can observe the growth is fluctuating in each group. This is a strange effect as even if there is no feedback given to users, their performance should become better and better or remain the same. After discussing with users, we came up with a conclusion that the template motion was too difficult for them. They all have no dancing experience. The dance motion is fast, complicated and long. They failed to remember all the moves and improve those moves even they know where is the mistakes. This leaded to the 2nd experiment with easier dance motion and the experiment will be discussed later.

Survey Result & Discussion

Survey with 10 questions was prepared for subjects. They could try other functions in the system if they request. Users were told to rate the aspects according to their degree
of satisfaction scoring from 1 to 5. The higher the score, the more satisfied is the user about the aspect. Scores 1 indicates that the users strongly disagree with proposition, while score 5 indicates that the users are strongly agree with the proposition. Questions asked are shown below. Question 10 asks for free opinion from users and is not included in table below.

<table>
<thead>
<tr>
<th>No.</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The dance education is easy to use.</td>
</tr>
<tr>
<td>2</td>
<td>The system can help you better understand the moves in a dance performance.</td>
</tr>
<tr>
<td>3</td>
<td>The system can help you know what and where mistakes are in your dancing.</td>
</tr>
<tr>
<td>4</td>
<td>You are satisfied with the accuracy of checking mistakes in your dancing.</td>
</tr>
<tr>
<td>5</td>
<td>The virtual teacher animations are interesting and they can stimulate your interest to use this system and to observe their motions.</td>
</tr>
<tr>
<td>6</td>
<td>You prefer learning dancing with the system to traditional “watching dance video” approach.</td>
</tr>
<tr>
<td>7</td>
<td>The system can help you learn dancing faster than watching dance video.</td>
</tr>
<tr>
<td>8</td>
<td>The interaction between teacher and student is important in dance training.</td>
</tr>
<tr>
<td>9</td>
<td>The interaction part in the system assisted you in dancing training process.</td>
</tr>
</tbody>
</table>

Table 7 Survey question

Fig. 44 shows survey result in nine bar-charts corresponding to nine questions in the survey. In figure, we can see that most users are satisfied with our proposed system after they have tried it out – in overall, about 59% respondents have given score 4 or above, while there are still a small number of users did not satisfied with the system (6%) who have given score 2 or below. We also observe that high proportion of users (about 35%) chose score 3. However, it is difficult to evaluate the effectiveness of the proposed system as score 3 is the mean score. This lead to a change of score scale in experiment 2.
Follow-up regarding to the experiment result

After getting feedback from users and professionals, system was modified in several ways.

1) To help users observing fast dance motion in demonstration, function that changing demonstration speed is added.

2) In evaluation part, instead of giving one or zero marks for a movement, marks are given regarding to similarity rate. That means a similar movement will get some score instead of zero in previous approach. In previous approach, improvement in performance is only shown in feedback when users really correct an error. Users may be depressed as it is difficult to get a higher mark. New arrangement is more
encouraging as marks will be rise with slightly improvement. Although marks
given to slightly improvement are low, users can see their improvement from
feedback.

3) In slow-motion reply section, color shown for correctness will have more level.
Instead of only yellow and red indicating right and wrong, we add blue and green
to indicate nearly correct and almost wrong respectively.

5.2 2nd Experiment

9 subjects joined this experiment. In this experiment, the dance motion to be learnt is
easier and slower than the ones used in 1st experiment. Subjects were divided into two
groups. Group 1 trained dancing with system giving overall similarity rate as
feedback. Group 2 trained dancing with system giving similarity rate on each joint
and slow-motion replay as feedback. After the experiment, both groups are free to use
our system and then fill in a questionnaire with 10 questions. Experiment and survey
result will be discussed below.

*User Background*

Eight of our subjects age 20 to 25. One of them age 26-30. Except one with 2 years
dance experience, all of them do not have dance experience.
The principle of this experiment is to test if feedback given by our system can
enhance a training process. From the result, we can see that the growth of group 2 is greater than group 1. This shows that detail feedback given can help users train in a better manner.

Survey Result & Discussion

Questionnaire with the same question (Table 7) as in experiment 1 is given to users. The rating is now changed from 1 to 5 into 1 to 6 in this survey as we thought that an even number of choices is better than a user must decide whether he/she is in favor of /against using the system. Scores ranging from 1 to 3 indicates that the users are dissatisfied with the proposition, while scores ranging from 4 to 6 indicate that the users are satisfied.
Fig. 47 shows survey result in bar-charts. In figure, we can see that all the users are satisfied with our proposed system after they have tried it out. The result of this survey is better than in 1\textsuperscript{st} experiment. There are some possible factors.
1) Some modifications are made to our systems based on users’ opinion.

2) The dance motion used this time is easier, so user can really enjoy using our system.

3) it is easier for users to see improvement from feedback given to them and they will not get depressed by seeing the same feedback all the time.

5.3 Suggested Improvement by subjects and professionals

Users gave some highly valued suggestions in the survey. Some of the most frequently suggested improvements are listed in this section.

- **beat signal**

Beat, or stepping, is important in dancing. Users suggested that our system may provide some beat signals for them, for example, “Bi-Bi” or hand clapping sound can be played at each beat.

- **training concentrated on certain part of a body**

For beginner, it may be difficult for them to move their arms and legs at the same time and in a correct manner. One of our users suggested that it will be better if the system can divide a motion of whole body into motions of different body parts. For example, a dance motion that involves moves of arms and legs are separated into two motions involving only arms movement and only legs movement. Then users can concentrate on practicing of movement of certain body part.

Dr. Taku Koruma has also given me his valued suggestions both for the system and experiment. He suggested that an easier motion can be used in the experiment. The result in experiment will be more significant. He also suggested that score can be given to some similar moves but not only giving score to totally correct moves. Then the score given to users will be more encouraging.
6. Challenges in this project

In this section, difficulties encountered while the system is built up are discussed.

1) Incompatible renderings in SSMDIC and MocapViewer v1

The rendering in SSMDIC is by DirectX 3D and MocapViewer v1 is by OpenGL. These two methods are not compatible in MFC. I have to select one of them and change the system with different rendering method into the same method. This is time-consuming to do so. After carefully studying, I found that it would be more effective to change SSMDIC to be in OpenGL. This is because rendering part of SSMDIC is clearly separated and is smaller compared to MocapViewer v1. So, changes are made to SSMDIC and are in an effective way.

2) Setting of View point during users’ performance

The view point is fixed while users are using functions “Evaluation” and “Practice”. They can change the view point by moving mouse and pressing hot key during demonstration. However, when they are performing their motion, they will not be able to using mouse or keyboard. Either using a view point from back or front of virtual teacher, the movement at other sides will be hidden. To solve this problem, we learn from the setup of a dance classroom. The screen showing from the back of virtual teacher and a mirror is placed in front of the teacher. Then movement at both sides will be viewable for users even if view point is fixed.

3) Rendering of different avatar

At section 4.4.2, we talked about methods to use avatar model data files. Finding of reference points is an important step and we have mentioned overlapping area
between body segments. However, at the beginning of this project, I mistakenly ignore the overlapping area and take the reference point wrongly. This lead to some animations that avatar’s body segments are not connected. The overlapping area is finally discovered after carefully studying.
7. Major Contribution

One of major contribution throughout this year is that I have authored and co-authored two papers in ICUMIC 2007 [1] and IMMERSCOM 2007 [2] respectively. The paper in ICUMIC 2007 was accepted and the one in IMMERSCOM was submitted. It is a valuable experience. Other contributions are listed below. I have built up a preliminary dance education system worked with motion capture system successfully. A viewer that demonstrates motion stored in motion data files in BVH format is built. It includes functions for users to change view point, demonstration speed and moving demonstration forward and backward. This viewer is used as a function to demonstrate dance motion in our system. An evaluator that compare motion data from motion capture system with motion data stored in motion data file in BVH format is made. The evaluator can compares two postures and evaluate similarity rate between two postures at different joints. A survey was conducted in order to collect users’ feedback and their expectation about the system. Numbers of valuable suggestion are received for improvement and future development. Finally, modifications of the system were made after carrying out survey to suit the requirement of users.
8. Conclusion

In this report, we have discussed the system that is for effective dance self-learning. A dance education system working with motion capture technology was made. At this moment, this system can demonstrate different dance motions with provided virtual teachers and assist users’ learning by providing feedback based on their performance.

In the process of observing virtual teacher's dance demonstration, users are able to change their viewpoint, speed of the demonstration and move the demonstration forward and backward. This system is working with a motion capture system EVaRT which is provided by CS department. The motion capture system helps our system to observe users’ motions. By analyzing their motions, our system can find out wrong postures performed by users and body parts of users that move wrongly. Moreover, users can also watch slow-motion replays of their motions in order to understand how well they have performed and how they can get improvement in the future. Survey showed users are interested in learning dance by our system and our system helps their learning. Experiment also showed training dance performance under meaningful feedback provided is important in learning process. In the coming future, the system will have further development to assist our next generation in learning dancing.
9. Future work

Besides the improvements suggested in the above section, other future developments would be introduced in this section. Some of them are technically challenge but they are very worthwhile tasks.

*Automatically divide motion sequence into small moves*

If a motion sequence can be automatically divided into small moves, the system can break a dance lesson into many sub-lessons without extra human-efforts. Users can train a motion sequence partly and concentrate on certain moves. Moreover, check point in “Practice” section can be set automatically.

*Music display*

With music, it is easier for users to follow a motion as stepping information is provided in music. Listening to music can help users more into their performance.

*Stepping error detection*

A perfect dance motion is to perform correct moves in correct stepping. At current stage of our system, users have to observe stepping error made by observing slow-motion replay themselves. If stepping error can be detected by our system, stepping error can first be get rid of at comparison between users’ motions and teachers’ motions. Then our system can tell users how well their performances are in term of stepping errors and posture errors.
10. Reference


# Appendix

## Project Monthly Progress

<table>
<thead>
<tr>
<th>Month</th>
<th>Progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sep</td>
<td>1) Finish parser for WRL files</td>
</tr>
<tr>
<td></td>
<td>2) Avatars in BVH Viewer can be changed</td>
</tr>
<tr>
<td></td>
<td>3) &quot;Forward&quot; and &quot;Backward&quot; function were added for users' convenience.</td>
</tr>
<tr>
<td>Oct</td>
<td>1) Modification of BVH Viewer was completed</td>
</tr>
<tr>
<td></td>
<td>2) Testing on BVH Viewer</td>
</tr>
<tr>
<td>Nov</td>
<td>1) Changed SSMDIC into OpenGL rendering</td>
</tr>
<tr>
<td></td>
<td>2) Migration of functions in BVH Viewer to SSMDIC started</td>
</tr>
<tr>
<td>Dec</td>
<td>1) Migration of functions to SSMDIC was completed</td>
</tr>
<tr>
<td></td>
<td>2) Modification of evaluator, real-time data handler in SSMDIC started</td>
</tr>
<tr>
<td>Jan</td>
<td>1) Modification of evaluator, real-time data handler in SSMDIC continued</td>
</tr>
<tr>
<td></td>
<td>2) Rearrangement of menu items was completed</td>
</tr>
<tr>
<td>Feb</td>
<td>1) Carried out experiment 1</td>
</tr>
<tr>
<td></td>
<td>2) Final testing</td>
</tr>
<tr>
<td>Mar</td>
<td>1) Carried out experiment 2</td>
</tr>
<tr>
<td></td>
<td>2) Final testing</td>
</tr>
<tr>
<td>April</td>
<td>1) Carried out experiment 2</td>
</tr>
<tr>
<td></td>
<td>2) Final testing</td>
</tr>
</tbody>
</table>