Title	Sketch2Motion: Sketch-Based Interface for Human Motions Retrieval and Character Animation
Author(s)	PENG, Yichen
Citation	
Issue Date	2021-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/17209
Rights	
Description	Supervisor:　　　　　　,　　　　　　　　,

Master's Thesis

Sketch2Motion: Sketch-Based Interface for Human Motions Retrieval and Character Animation

PENG, Yichen

Supervisor Kazunori Miyata

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
(Knowledge Science)

February. 2021

Abstract

Character animation is a challenging task in traditional animation design. To create the expected motions with a high level of reality, human motion data is used to rig up the character model. In recent years, Motion Capture (Mocap) is being widely used to record subjects' joints movement and turn physical motion into digital data for Computer Graphics (CG) animation. However, the Mocap process requires special equipment such as the Mocap suit with a set of markers or motion sensors which are not easily accessible to common animators. Although there are thousands of accessible Mocap data on the Internet, it is laborious and time-consuming to retrieve an expected one because most of the motion libraries require keywords to category the motion which is unintuitive.

More recently, the 2D pose estimators such as Openpose and Alphapose which are trained by using deep learning (DL) algorithm were proposed to directly track motions from a video. The animator can record their motion trajectories using an RGB camera as a motion reference input. Although applying the pose estimation makes it easier to reproduce physical motion, it takes effort on computation. Besides, it requires further DL process such as Generative Adversarial Networks (GANs) to covert 2D motion data to 3D which increase the noise and invalid pose in a motion sequence.

In this study, instead of querying by keywords or generating desired motion sequences using the DL process, we propose a sketch-based user-interface for users to retrieve the anticipate motions. Because the human motion data is spatially and temporally dynamic, it is more intuitive to describe the motion by using trajectories curves than text keywords. We extract the trajectories of 5 joints (head, left hand, right hand, left feet, right feet) movement from the Mocap database to draw the motion map for each joint. Each motion sequence is related to its motion map.

After comparing the performance of two different image feature descriptors, we then match the features between the users' sketch and motion map by applying an image patch descriptor called Scale-invariant feature transform (SIFT). SIFT extracts the key points in both images and measuring the similarity of the features based on the Euclidean distance of feature vectors. The most similar motion map can be selected accordingly.

In our interface, users can browse overall motion by viewing the skeleton motions in the database by simply drawing the expected motion trajectories as the input query. Inspired by ShadowDraw UI, we suggest user draw the valid strokes by displaying the similar trajectories' maps behind their sketches. Since browsing the overall motion sequence from the first to the last frame is time-consuming, we researched the visualization of motion data and designed two motion maps (one in the absolute

coordinate system and the other in a relative coordinate system) for users to check the motion once the sketching completed.

We verify the utility of the proposed sketch-based user interface with a user study, in which the participants retrieved the expected motion sequence from the database with 98 Mocap data proposed. A comparative study is designed to confirm the effectiveness of our interface. The participants are asked to retrieve a target category of motion from a small-size motion library we constructed with and without using the sketch-based interface. Besides, a questionnaire is conducted for an evaluation study with 5-point Likert-scales. The participants were asked to score our interface up after experience the operations.

To have a better understanding of how the users prefer to sketch the motion, we observe all of the input query sketches. It is considered that the users have the same habit of sketching gestures to the specific category of motions. Therefore, it is verified that the proposed sketch-based interface is also applicable for other types of motions which can be the straightforward future work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we introduce the research background of sketch2motion, which includes character animation and motion capture. We then describe our research purposes, the contribution and the thesis's structure.

1.1 Research Background

1.1.1 Keyframe Animation

In the late 19th century, an English photographer, Edward James Muggeridge, examined to use the multiple-exposure cinematography technique to shot a set of images illustrating a locomotion of human movement as shown in Figure 1.1.

The similar idea has been used in designing the character animation. It was started from the frame-by-frame animation at the very beginning. The animator has to draw the character and its pose in each frame. In terms of the early 2D animation, Disney started with creating animation in 24 frames per second (fps) which means the animators should draw 24 individual drawings for a one second animation play. Since the motion of the character cannot be viewed until a whole sequence of motion is completed drawing, the keyframe poses would be designed on a motion storyboard at first as shown in Figure 1.2.

Figure 1.1: An example of locomotion presenting a male jumping.
(Copyright of the National Museum of American History, Smithsonian Institution. http://www.si.edu/.)
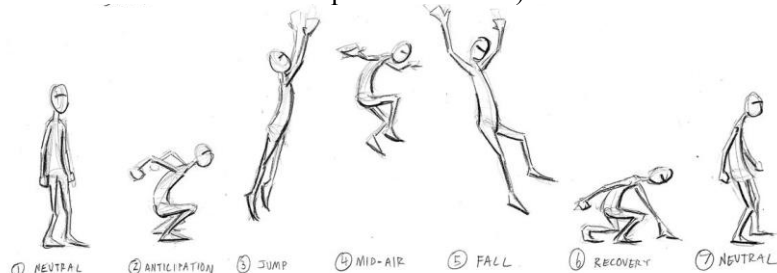
Figure 1.2: The keyframe poses of jumping motion on a motion storyboard
(Copyright of the Department of Computer Science and Engineering, University of Washington. https://www.cs.washington.edu/)

1.1.2 Character Animation

In recent years, there are several of softwares proposed by such as Adobe *Animate* [1] (for 2D animation) showing in Figure 1.3, AutoDesk *Maya* [2] (for 3D animation) showing in Figure 1.4 to increase the efficiency of the character animation production. By applying these kinds of softwares, the animators are allowed to set up a skeleton of the character and transform each joint position to design the keyframe poses. The animation between two adjacent keyframe poses can be automatically generated. Finally, after animators taking care of the timing of each keyframe and directly adjusting them on the timeline, a skeleton animation can be finished.

However, it is still a challenging task to produce a high-quality character animation, because the animators are required to observe thousands of the human motions in real life that they are able to reproduce them in the virtual world by using such softwares. Moreover, the human motion is spatially and temporally dynamic, looking into the motion sequence to understand the patterns of them requires some efforts from the animator watching the motion from the first to the last frames and changing the viewpoints.
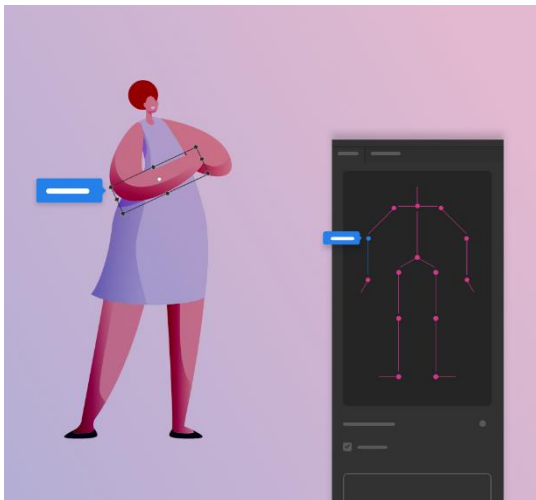
Figure 1.3: Building 2D skeleton by using Adobe Animate (copyright of Adobe Animate https://www.adobe.com/products/animate.html)
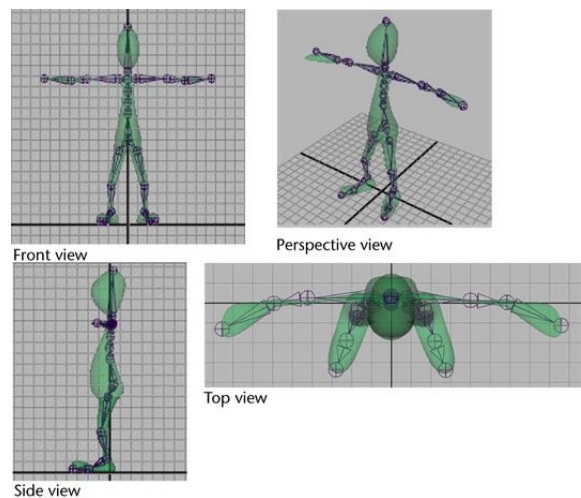
Figure 1.4: Setting up 3D skeleton by using Maya (copyright of AutoDesk Maya https://www.autodesk.co.jp/products/maya/over view?plc=MAYA&term=1-YEAR&support=ADVANCED&quantity=1)

1.1.3 Motion Capture (MoCap)

More recently, to reproduce the expected motions with a high-level of quality in a short time, Motion Capture (MoCap) technology is being widely used to capture the 3D human motion as the 3D skeleton animation data. (MoCap is not only able to capture joints

movement of human skeleton but also available to tracking the facial expression, finger movement and so on, but we refer to the joints tracking mainly in this thesis.)

In terms of the MoCap applied in CG animation such as video games and filmmaking, optical Mocap system is mostly adopted mentioned. Applying the optical MoCap requires special equipments such as the Mocap suit with a set of markers and two or more cameras which are calibrated to provide overlapping projected space. The 3D position of each marker is triangulated by projecting the 2D position from each image taken by the cameras.

In addition to the optical system, there are other types of motion capture systems such as inertial sensor, magnetical, mechanical, and image recognition systems. Figure 1.5 shows examples of these motion capture systems, such as optical MoCap [3], Inertial sensor MoCap [4], magnetical MoCap[5], mechanical MoCap [6].

Opitcal MoCap [vicon]
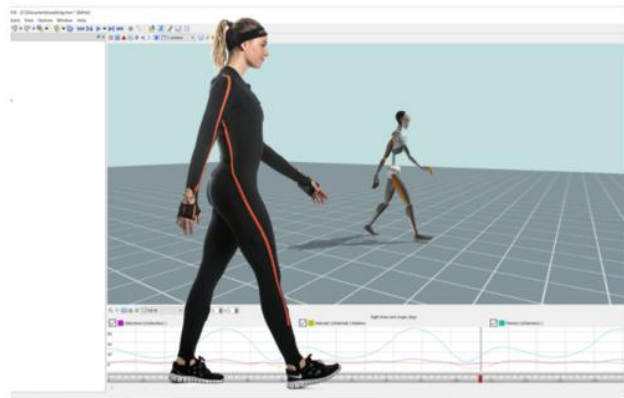
Inertial sensor MoCap [Xsens]

Magnetical MoCap [K-Vest]

Machanical MoCap [Gypsy 7]

Figure 1.5: Four kinds of MoCap systems

Note that that the image recognition MoCap approach is markerless. Only an RGB camera or depth sensors are required, so that it can be used for many different situations such as outdoor shooting and underwater shooting. Moreover, it is easily accessible for the common animator to record their own expected motions. For example, the *Microsoft Kinect* [7] and the *Intel Real-Sense* [8] are the well-known depth sensors which are able to capture users' motion in real-time. Furthermore, the DL model such as Openpose [9] and Alphapose [10] were proposed to allow user to convert their video to skeleton animation data even with a web camera, although it can only project the joints' position in 2D. Each coin has two sides, comparing to the approach requiring markers, the image recognition MoCap has a low accuracy and high proportion of noise poses comparatively.

1.2 Research Objective

Although the previous approaches to reproduce a desired motion take many efforts, there are thousands of motion data libraries on the Internet that the animators can easily access. Unfortunately, most of the libraries are labeled the motion by text keywords
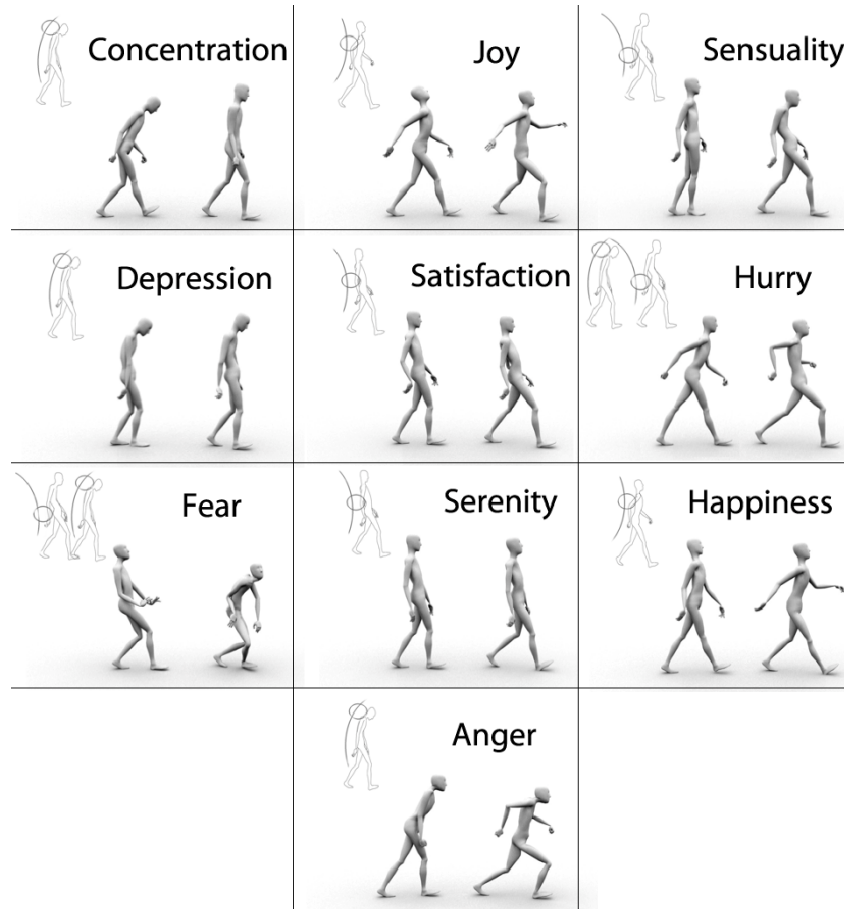
Figure 1.6: An example of grouping walking animation in different emotions. [11]

4

(label). As mentioned before, it is considered that the human motion data is spatio-temporal which is difficult to understand their patterns with only several of the keywords. Besides, most of the keywords are subjectively decided. For example, as shown in Figure 1.6 [11], walking with concentration and walking depressively have similar patterns visually but have great different description in keywords.

Therefore, either applying MoCap approach or retrieving the motion data from the existed motion data libraries is not a proper way to reproduce the expected character motion. To solve these issues, we propose a sketch-based user-interface for users to retrieving the desired motion. We extract the trajectories of 5 key joints (head, left hand, right hand, left feet, right feet) of each motion from our data library to draw their motion maps. In our interface, users are allowed to retrieve their desired motion by drawing the key joints' trajectories as a query input. We compare the feature between plural motion maps and the query sketch by applying SIFT [12] algorithm. In additional, inspired by ShadowDraw UI [13], we suggest user to draw the valid strokes by displaying the similar motion maps behind their sketches.

1.3 Contribution

The proposed UI applies a similar approach of ShadowDraw UI to support the animators to animate the skeleton while allows them to browse the retried motion sequence interactively. Applying our approach to retrieve the motion is simplified as much as an unskilled user (animator) can draw the sketch without difficulties. Besides, due to the popularity of SNS tools such as *Instagram* and *Tik Tok*, users take pleasure in sharing their creations. For this applicational scenario, the proposed UI allows common user to design simple character animation on a tablet or even a smart phone and share it though the SNS tools.

1.4 Thesis's Structure

This thesis consists of five chapters. In Chapter 2, we introduce related work on sketch-based UI for character animation and clarify the contribution of this research. In Chapter 3, we introduce the details of our UI, the motion dataset and similarity match algorithm used in our method. In Chapter 4, we design a user study to certificate the validity and discuss its results. Finally, in Chapter 5, we make a conclusion of this research and introduce the possible future tasks.

Chapter 2

Related Works

In this chapter, we firstly introduce the related works of this thesis including sketch-based interactive interface, motion visualization and retrieval. Then we briefly introduce motion retargeting of character animation. Finally, we describe the research position of this study.

2.1 Sketch-based interface

In the field of the visual concept design always relies on sketching. Sketching is utilized in all phases of the creative process, from designing shapes and defining motion with several key poses on the storyboards.

Sketch-based interactive design system includes designing the static 3D shapes, retrieving image and creating dynamic visual effects from simple 2D strokes. We introduce them separately in this section.

2.1.1. Sketch-based Shape design

Since the sketch is one of the most comprehensible visual information for depicting a shape, Teddy [14] was proposed for designing 3D freeform shape with 2D strokes input. As shown in Figure 2.1, users are allowed to define an external edges with the 2D strokes to create a new 2D polygon and defined the structure of the 3D mesh by adding internal edges again with the 2D strokes.

a) snake   b) snail   c) cherry   d) muscular arm
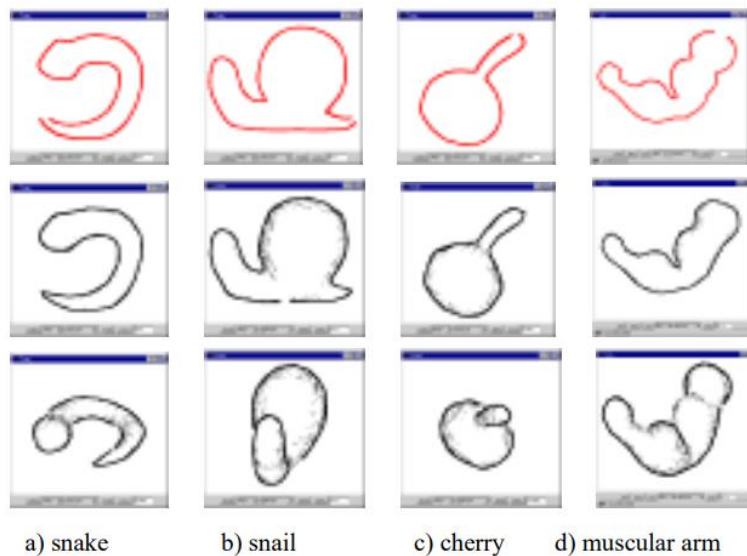
Figure 2.1: Creating the 3D mesh from 2D strokes in *Teddy*.

A similar system based on the data-driven method was proposed by E.Mathias et al. in 2012 [15]. A 2D sketch-like drawing dataset was set up by rendering from viewing the 3D models from several viewpoints. They then matched the feature between users' sketch and the plural drawing dataset. The eligible 3D model in the library would be selected related to the most similar image in the dataset.

More recently, for the further expansion of this idea, there is a research proposed by C.Zou [16]. The shape descriptor, Pyramid-of-Parts, was applied to segment the 3D model before rendering 3D shapes into 2D contours, so that each segmented part of the 2D contours can be labeled allowing more effective matching of the input query strokes.

2.1.2 Sketch-based Dynamic Effect Design

The sketch can be not only used to design static 3D shapes, but also can be considered as a hint input for generating animation. Energy-Brushes [17] was proposed for generating the dynamic effects of stylized animations such as waves, splashes (the effect of water drops), fire and smoke. As shown in Figure 2.2, artists are allowed to roughly sketch the internal forces of the special effects such as air expansion, wind blowing or water flowing to define their own stylized effects interactively.
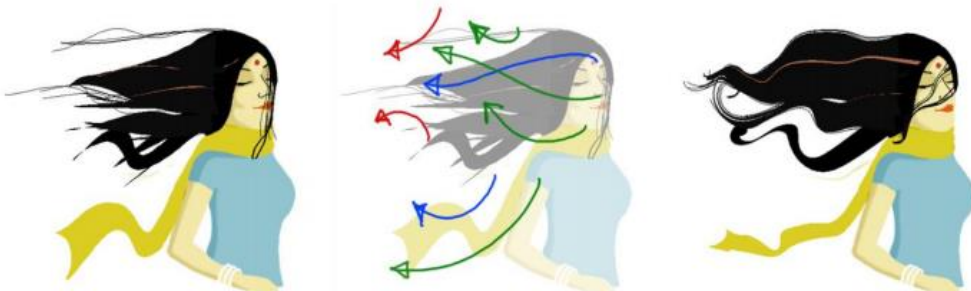
Figure 1.2: An Example of using energy brush to design dynamic effect such as hair blown.
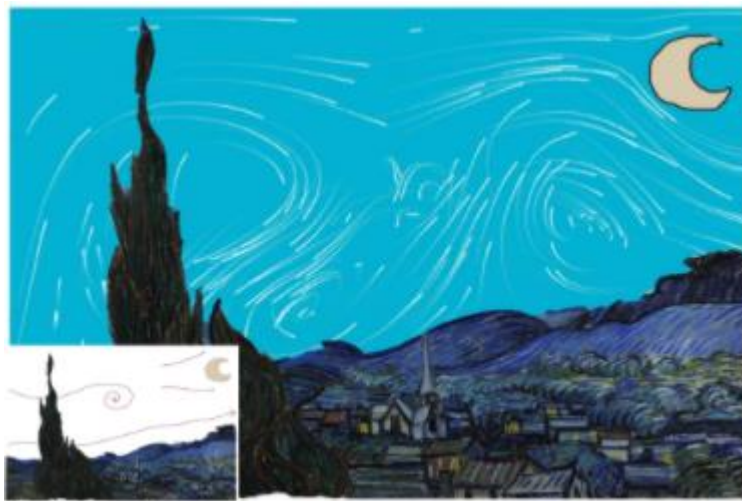
Figure 2.3 An example of reproducing the dynamic version of Starry Night by utilizing Sketch2VF

There is a similar interactive interface, called Sketch2VF [18]. They focused on generating fluid effect from sketches. They utilized a GAN model to generate static velocity fields from a sketch input. As shown in Figure 2.3, a swirl shape of moving stars was designed from several of simple strokes by applying the proposed interface.

2.1.3 Sketch-based Character Animation

In addition to considering the underly forces to generate dynamic effect, the sketches can be also considered as the trajectories for leading the skeleton animations. SketchiMo [19] was proposed to simplify the process of editing character motion. In their interface as shown in Figure 2.4, the artists are allowed to design the 3D character motion by sketching the motion trajectories from a specific viewpoint. Besides, the sketches are editable that the designed motions are easily adjustable and the result motion is calculated by utilized Inverse Kinematic (IK) algorithm.

Figure 2.4: Designing character motion by utilizing SketchiMo

On the other hand, M. G. Choi proposed an interface [20] to support users without artistic expertise to retrieve the expected motion from a motion library by drawing simple stick figure as shown in Figure 2.5. The motion sequence is generated according to the sketched stick figures as the input poses of the keyframes.
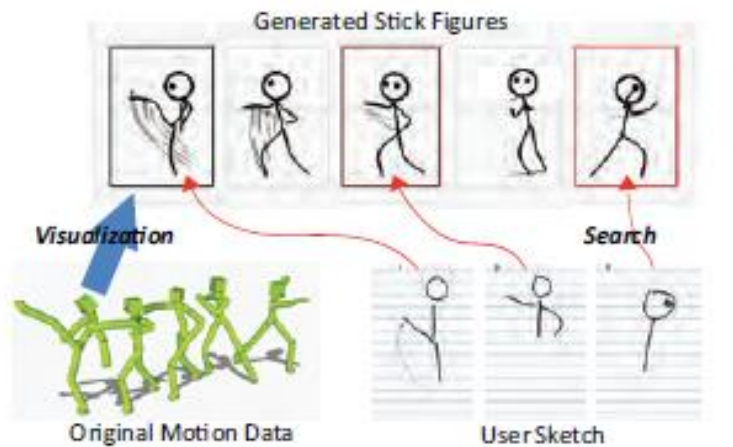
Figure 2.5: Stick figure.
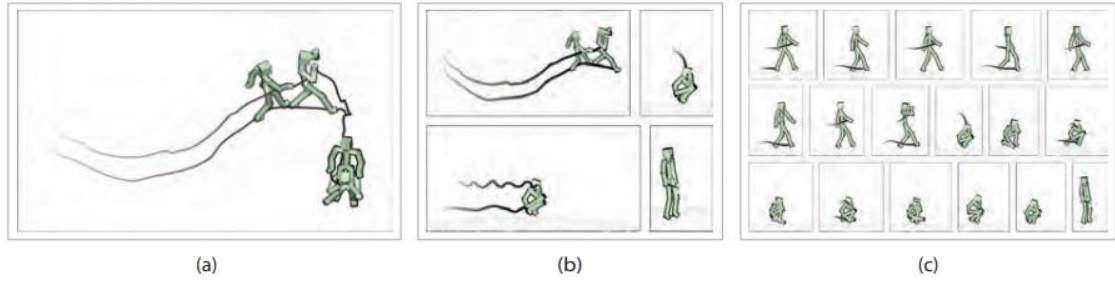
Figure 2.6: A motion storyboard of a long sequence of skeleton animation

A similar work [21] was proposed by the same researcher for artists to quickly browse through a long sequence of the animation data and semi-automatically synthesizing a motion storyboard as a locomotion. As shown in Figure 2.6, although it has less relationship with sketching, it provided a new type of interface to arrange a motion sequence with a long and complex motion contents by transition of the locomotion.

2.1.4 Sketch-driven Drawing Assistant

Sketch-driven interactive drawing assistant interfaces have been widely utilized for improving users' drawing skill. The major idea was firstly proposed by ShadowDraw [13], a drawing assistant interfaces for guiding the freehand drawing of objects. They extracted edges from a photo data library to construct the dataset. As shown in Figure 2.7, in their interface, the features of users' sketches are extracted in real-time and drawing suggestions are provided accordingly by weighting the data image by the features' matching scores.
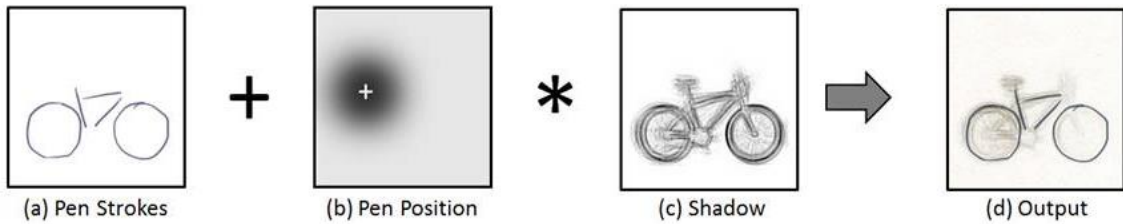
Figure 2.7: ShadowDraw UI

In addition to drawing assistant, Shadow-like UI is also utilized in image synthesis. An interactive sketch-based image synthesis interface was proposed [22] to generate realistic images of animal from a rough sketches (masks in ellipse shape) input. They trained an end-to-end DL model to recognize the relationship between the body structure of each kind of animal and synthesis their image accordingly. Figure 2.8 illustrates the interface of their system.

9
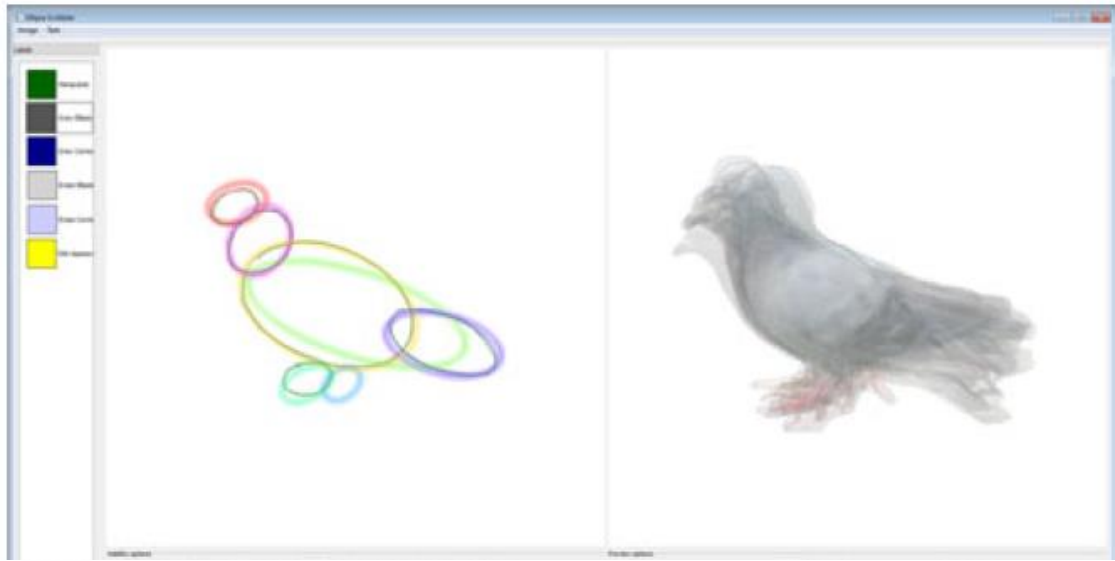
Figure 2.8: Interactive Sketch-driven image synthesis. (Left: Masks in ellipse shape, right:
Generated result with shadow.)

Besides images synthesis or drawing assistant, shadow-like interface is also able to
be applied in supporting calligraphy learning. A calligraphy training system using
projection mapping technique [23] was developed to improve the calligraphy skill.
Users are allowed to customize the expected calligraphy style by browsing the shadow-
like recommendation while writing in real-time. Figure 2.9 shows their interface.

Figure 2.9: The interface of calligraphy training system using projection mapping

2.2 Motion Visualization and Retrieval

With the development of the MoCap technique, in order to have a better understanding of human motion and a better management of the motion data library, researchers are trying to find a solution of illustrating the motion using different kinds of approaches. In this section, we introduce several related works on this topic.

2.2.1 Convey Motion

Convey motion is a type of motion visualization with showing the static poses of keyframe following timeline as shown in Figure 2.10. For example, Motion Belts [24] was proposed to automatically generate the outline of a motion sequence for browsing. However, the keyframes of a motion sequence was selected only according to a specific time unit, so that the poses of some of the keyframe are possibly meaningless. Actually in 2005, Action Synopsis [25] was proposed to solve this issue, they introduced an algorithm for carefully selecting the key poses in based on analyzing the joints movements (transitions and rotations) of the skeleton.
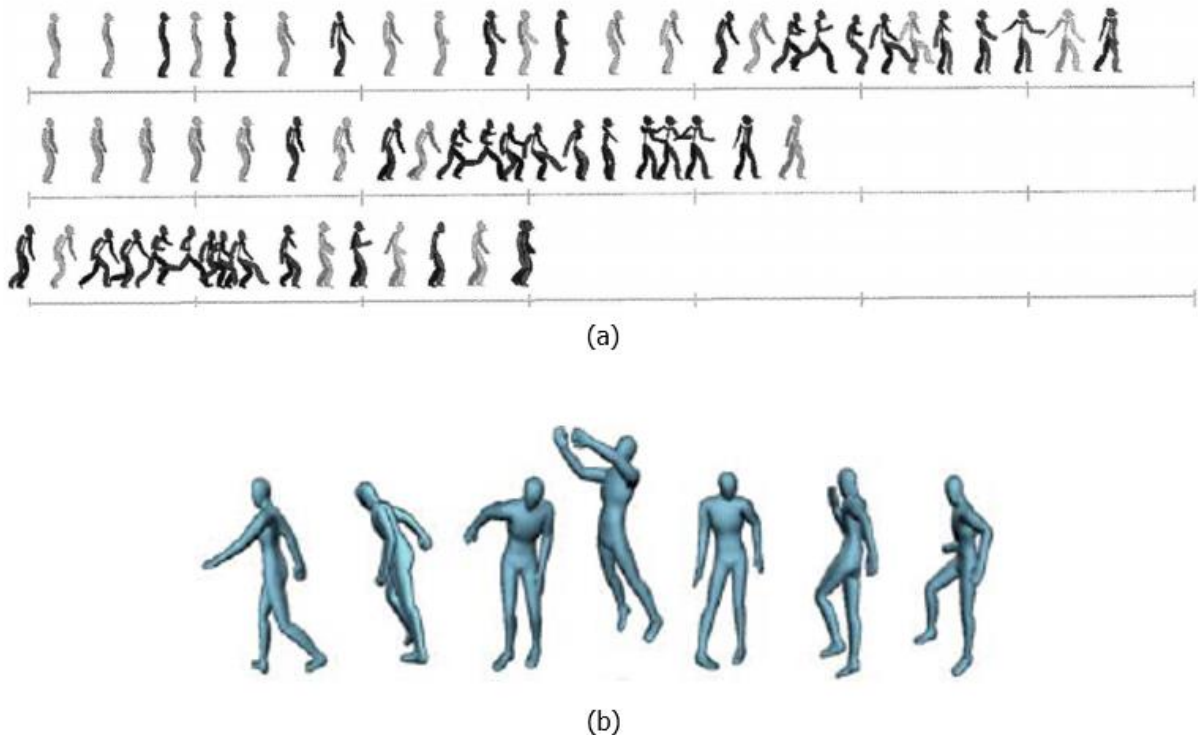
Figure 2.10: (a) Motion Belts. (b) Keyframe selecting by applying Action Synopsis.

2.2.2 Motion Cues

Although convey motion is a solution for understanding how a motion happening by illustrating the locomotion following the timeline, the spatial changing information of the motion is not intuitive enough. Motion Cues [26] was proposed to synthesize a 2D image of an animated character by generating intuitive motion cues derived from 3D skeletal motion sequence. For example, arrows (red arrows in Figure 2.11), noise waves (around the hands in Figure 2.11(c)), and trajectories (around the legs in Figure 2.11(a) of the motions shown in Figure 2.11.

(a)   (b)   (c)   (d)
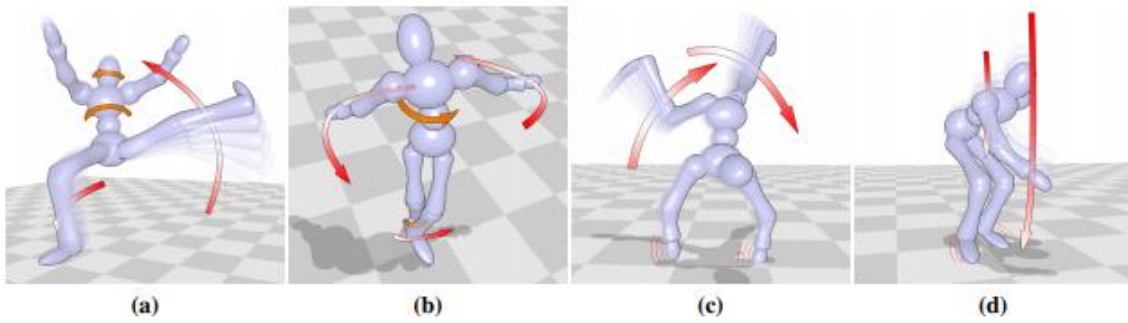
Figure 2.11: Motion Cues

2.2.3 Motion retrieval

S. Zamora proposed a system for detecting general articulated skeleton figures in sketch-based applications [27]. As shown in Figure 2.12, their interface is able to recognize each construction of an articulated skeleton model from the rough sketches. It gave the hint to Stick Figures [20] to propose an interface that allows users to retrieve from motion data library by drawing rough sketches as mentioned before.
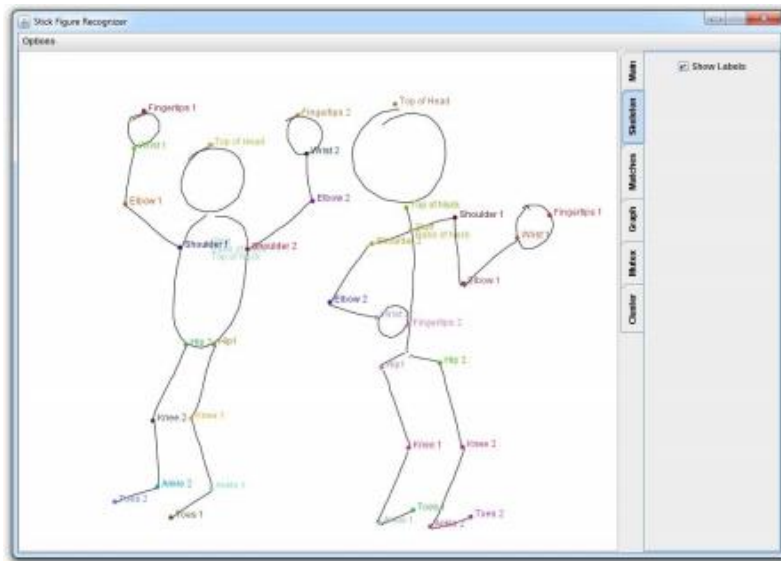
Figure 2.12: Sketch-based recognition system for general articulated skeleton figures.

12

2.3 Motion Retargeting

In terms of utilizing MoCap technique to reproduce motion on a virtual character, there is a challenging task that rigging up character with different bone proportions of skeleton. Date back to 1998, Gleicher et al. [28] examined a skeleton motion retargeting by formulating a spacetime optimization problem with kinematic constraints. They utilized the formulation to treat the entrie motion sequence. Figure 2.13 shows their method retargeting the motion even while scaling the character. However, as shown in Figure 2.14, it fails to complete the retargeting task led by the lack of physical constraints.

Figure 2.13： Retargeting the motion even while scaling the character.
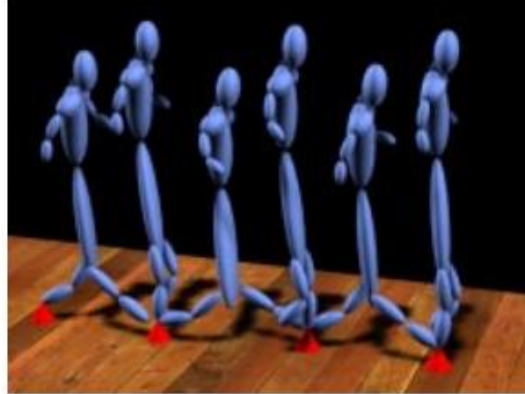
Figure 2.14: An example of unnatural motions led by the lack of physical constraints.

Recently, with the development of DL framework, K. Aberman [29] proposed a DL model for motion retargeting on skeleton with even different structures. Its result is much
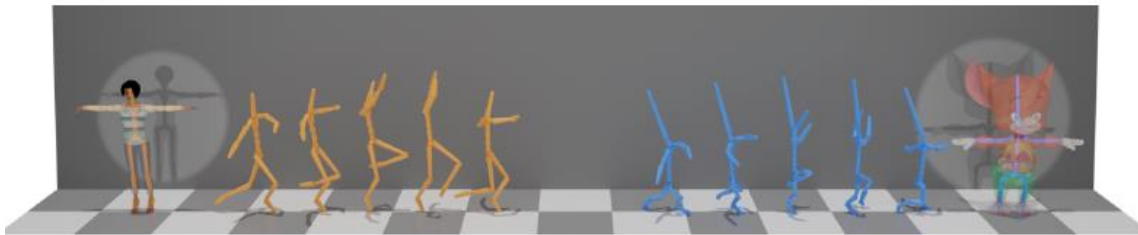
Figure 2.15: Skeleton-aware networks for deep motion retargeting

robust showing as Figure 2.15.

2.4 Research Position

In this thesis, we propose Sketch2Motion, a sketch-based interactive interface for users to retrieve expected motion from the motion data library. Our interface can utilize shadow-like guidance to support users in designing skeleton motion.

On one hand, for the issue that neither capturing a motion nor retrieving a motion from an opensource motion data library is a proper method for the common animator to search the desired motion as the reference to design their character animation. Sketch2Motion allows users to retrieve desired motion by intuitively drawing the trajectory curves of the key joints.

On the other hand, with the increasing number of active users of the SNS tools such as Instagram, Tik Tok, and Facebook, the users are taking pleasure in sharing their creations. For this reason, Sketch2Motion also allows the common users to design a simple character animations by simply drawing several strokes on a tablet or even a smartphone without the requirement of any professional skill. With this benefit, they can share their character via SNS tools.

Besides, we make a hypothesis that the users are more inclined to define an entire motion by depicting the trajectory curves of 5 key joints (head, hands, and feet). We verify this hypothesis by conducting a questionnaire in our user study.

Chapter 3

Sketch2Motion

In this chapter, we describe the framework of our study, the datasets and the similarity measuring algorithm we used to build up our prototype and the contents of our UI.

3.1 Framework of Proposed System

The overall workflow of the proposed system is shown in Figure 3.1. To construct the motion data library, we firstly drew the trajectory map (we will describe the concept later in this section) of each motion sequence. For each 3D skeleton motion, we defaultly set virtual camera at the side view and confirmed that the entire motion is performing inside of our scene. Besides the trajectory map, we drew other two motion maps for each skeleton motion, the one in the absolute coordinate and another in relative coordinate. They are accordingly showed to the users as the information of the motion. For each of the trajectory map, we then utilized the SIFT descriptor to extract the features. The feature of user's query sketch is extracted in real-time so that the feature matching between the sketch and trajectory map is able to run after each stroke of the sketch being drawn. Meanwhile, the user is allowed to browse the motion maps in two difficult coordinates and the matched animation playback in a sub-window to decide whether the skeleton animation is desired or not.
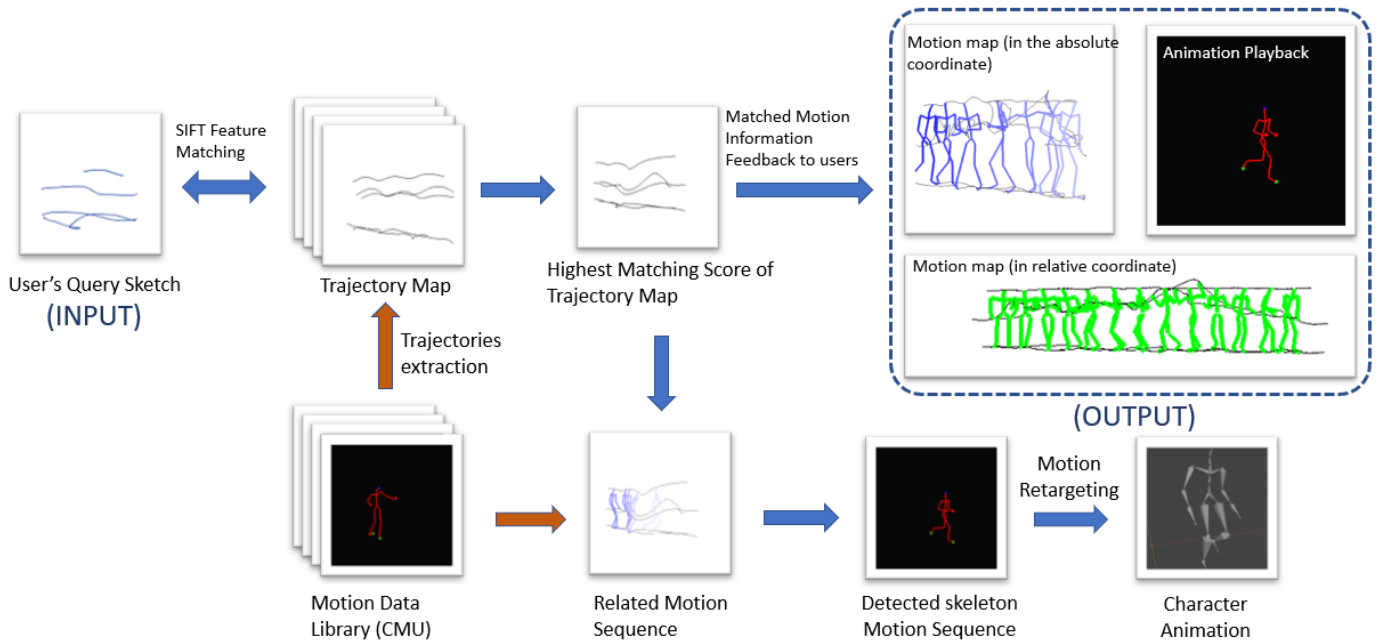
Figure 3.1: Framework of proposed system

3.1.1 Construction of Motion Dataset

In this thesis, we utilized the CMU (Carnegie Mellon University) MoCap data [30] to construct our data library. CMU MoCap data used ASF/AMC file to store subjects' skeleton and motion data. An example of a set of the ASF/AMC file is showing in Figure
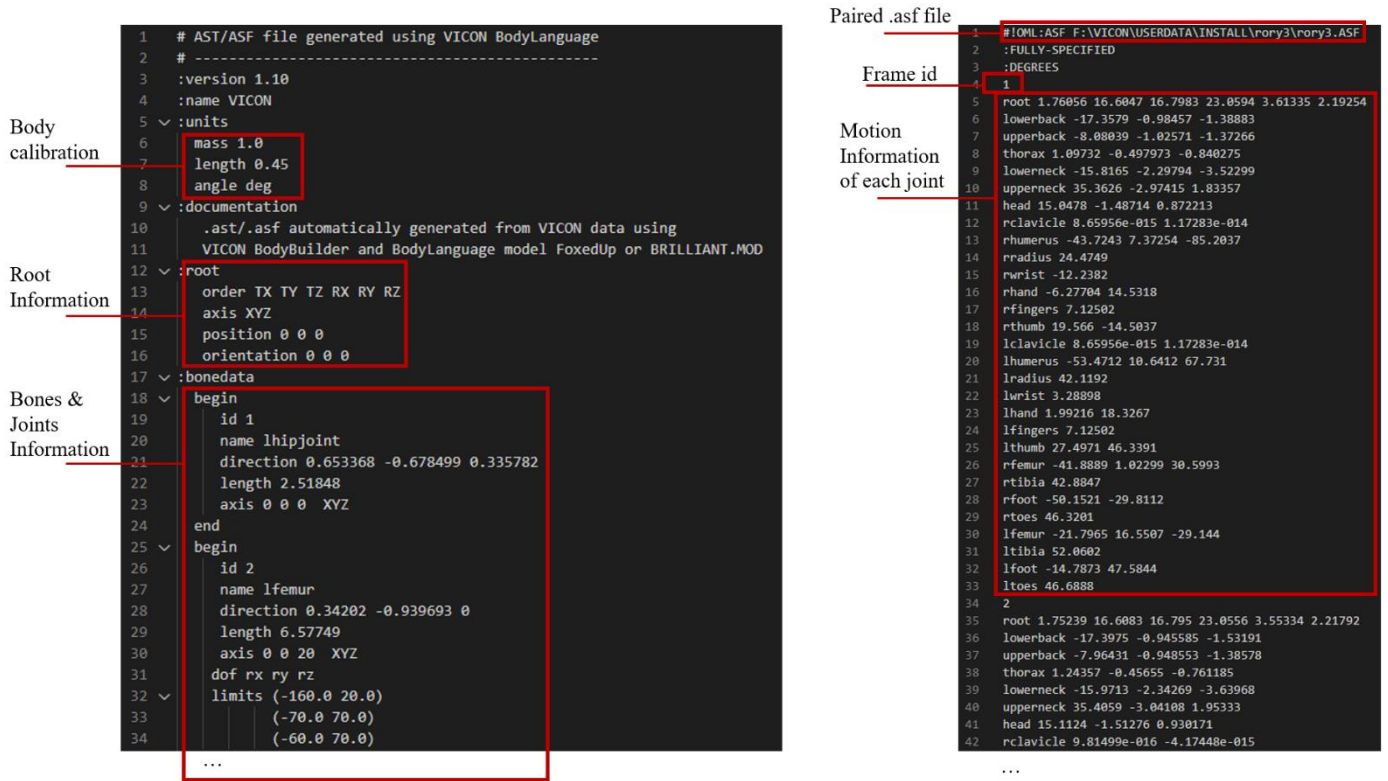
Figure 3.2: An example of a set of amc/asf file.

3.2.

- ASF (Acclaim Skeleton File) file: In the ASF file a base pose is defined for the skeleton that is the starting point for the motion data. The body calibration data of the subjects is stored in this file. Each subject should have a .asf file.
- AMC (Acclaim Motion Capture) file: The AMC file stores the orientation data of 29 joints of the skeleton defined by ASF file as shown in Figure 3.3.
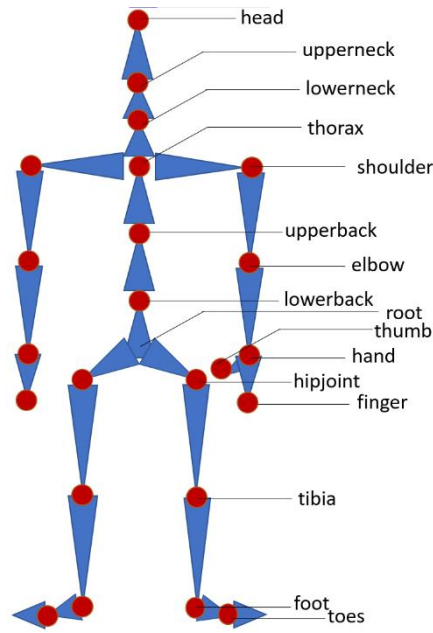
Figure 3.3: The skeleton model of AMC/ASF file.

CMU Mocap dataset stores 2605 skeleton motions including 6 motion categories (human interaction, interaction with environment, locomotion, physical activities & sports, situations & scenarios and test motions) and 23 subcategories. Starting with the comparatively simple motion, we mainly use locomotion and sport motions to build the prototype in our study.

To sort the bones and joints information in AMC/ASF file, we adopted a parser in Python using matplotlib to draw the joints and bones statically. Figure 3.4 illustrates an example that the AMC file was parsed by our AMC parser.
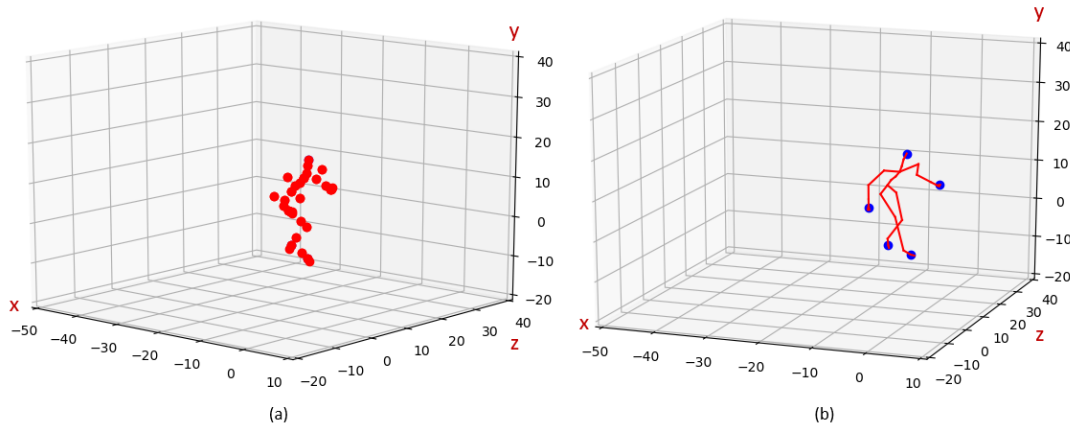
Figure 3.4: (a) Joints and (b) Bones drawn by using matplotlib.

After parsing the information in the file, we apply OpenGL to play the skeleton motion. The reason we utilized OpenGL is that we are possible to customize viewpoint of our

camera in the virtual space, which convenient for us to depict the trajectory's map of each motion. Figure 3.5 shows an example of playing the motion clips in the virtual space using OpenGL.

Figure 3.5: The motion sequence played in virtual space.
(with the head, hands and feet show in blue, red and green respectively.)

To depict the trajectory map and motion maps, we utilized OpenCV to draw the strokes and save them as an image. In our prototype, we draw only the trajectory curve of 5 joints (head, left hand, right hand, left feet, right feet). We make a hypothesis that these 5 joints are the most important joints to depict an entire human body movement, and we verify the hypothesis by a questionnaire in our user study. Figure 3.6 shows a set of results of output images. The resolution of the trajectory map and the absolute coordinated motion map are set to 512x512. We use the side view of the animation, and fix the virtual camera after confirming the entire motion sequence is inside of our scene to draw the absolute coordinated motion map. The blue skeleton figures in Figure 3.6 (b) shows an example while the bright blue shows the pre-movement, and the pale blue shows the post-movement. On the other hand, to illustrate the key pose sequence horizontally, we define a relative coordinate motion map and set it to 1461x130 as shown in Figure 3.6 (c).
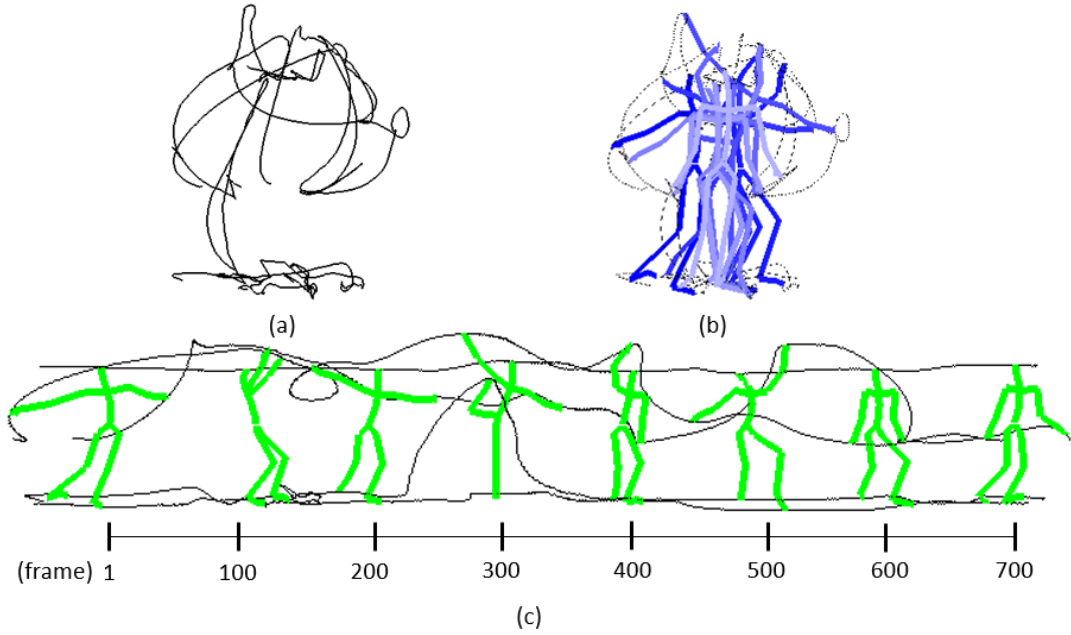
Figure 3.6: A set of result of image output. (a) Trajectory map. (b) Motion map (absolute coordinate). (c) Motion map (relative coordinate).

3.1.2 Image Descriptor (Feature Matching Algorithm)

To match the feature between the query sketch and the trajectory map, we considered two different image descriptors in this study.

- Fréchet distance: It is an algorithm for measuring of similarity between curves as shown in Figure 3.7. The high levels of accuracy and effectiveness has been proved by previous research [31]. However, to measure similarity between each input stroke and joints' trajectory, it is necessary to label each trajectory with its joints and define the meaning of each stroke before users inputting, which is complicating operation for users.
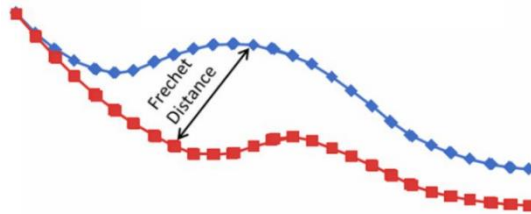
Figure 3.7: Frechet distance.

- Scale-invariant feature transform (SIFT): It is a traditional image descriptor used in the field of image processing. This describing process is scale invariant and can

detect key points in an image, and is a local feature descriptor. We made a preliminary experiment on it by using a small number of simple sketch dataset (50 sketches of ladder and 50 sketches of tomato). Figure 3.8 shows a set of top five results of the experiment which has only one noise image (ladder) in totally 5 results with sketching a tomato as an input.
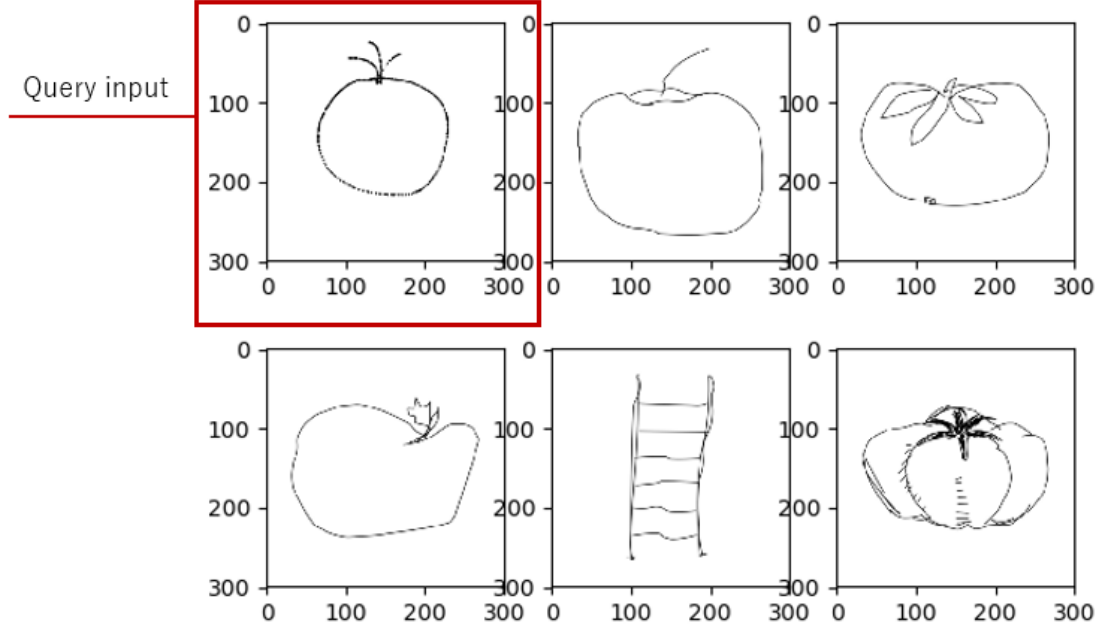
Figure 3.8: One of the results of SIFT descriptor preliminary experiment. (A set of top 5 results with one matching failed.)

Therefore, we decided to apply SIFT descriptor to solve the similarity measuring task. We compute the descriptors locally over 102x102 patches for the 512x512 image. The patches used to compute adjacent result of descriptors overlap by approximately 50%.

3.2 User-Interface

We developed our UI by utilizing PyQT5 in Python. As shown in Figure 3.9, we constructed our UI with 3 windows.

The main window is the interactive canvas for the users to draw the query strokes shown as the blue curves in Figure 3.9. Once a stroke has been completed, the entire painting draw by the user would be sorted as a PNG image file which we can extract its feature by using the SIFT descriptor. We then measure the similarity between the input image and each of the trajectories' map in the dataset. Top 3 of the best matched trajectory map would be displayed as the shadow-like guidance behind the user's strokes to lead them draw their stroke along the valid traces of 5 joints.

20

The first sub-window on the top right displays the motion map of the best matched skeleton animation. As shown as the skeleton in gradient blue, the motion map is depicted by the keyframes of the skeleton animation, which we used a fixed virtual camera taking multiple-exposures to skeleton moving in the absolute coordinate. The saturation of the blue illustrates chronological order of keyframes. The bright blue shows the pre-movement while the pale blue shows the post-movement as shown in Figure 3.9 (top right). To make it easier for users to understand, the thin black curves behind the skeleton illustrate the trace of the movements of key joints.

Since the skeleton pose in the keyframes are possibly seriously overlapping that cannot be clearly shown to the users. We set up the second sub-window in the bottom display another motion map that the skeleton animation is playing in the relative coordinate. The green skeleton in Figure 3.9 shows the pose of the keyframes in chronological order from left to right. Repeatedly, black curves behind the skeleton illustrate the trace of the movements between each of the keyframe.
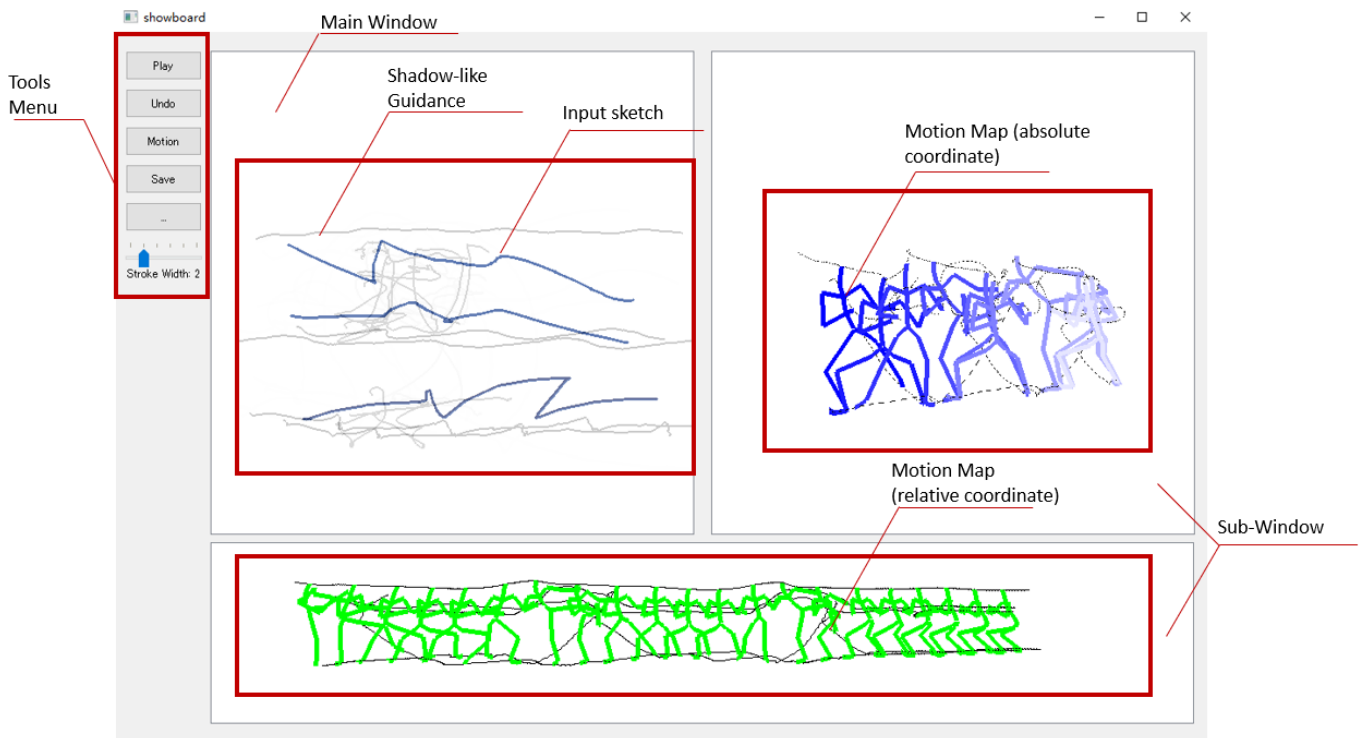
Figure 3.9: The UI of our prototype.

It is still not clear enough by browsing only the static motion maps on both sub-windows to check the matched motion. By clicking the 'Play' button on the top left after sketching, the third sub-window would display for playing the overall skeleton animation as shown in Figure 3.10. This window is built up by using the PyGame library in Python to project the OpenGL's view on to PyGame scene. Besides, we set up the KeypressEvent that the users can operate the OpenGL's view interactively, for example the rotation or

transition of the virtual camera, the skeleton animation play back or play forward and so on.

The tool menu is on the top left of our interface:

- The 'Play' button is used to control the animation playback window as mentioned.
- The 'Undo' button is used to return to the pervious strokes history when the new stroke is unexpected.
- The 'Motion' button is clicked to show the two motion maps.
- The 'Save' button is clicked to save the matched AMC/ASF file and the inputted query sketch.
- The '…' button is used to load the saved sketch or other depicted motion trajectories map.
- The 'Stroke Width:' slider is used to adjust the width of the input strokes.
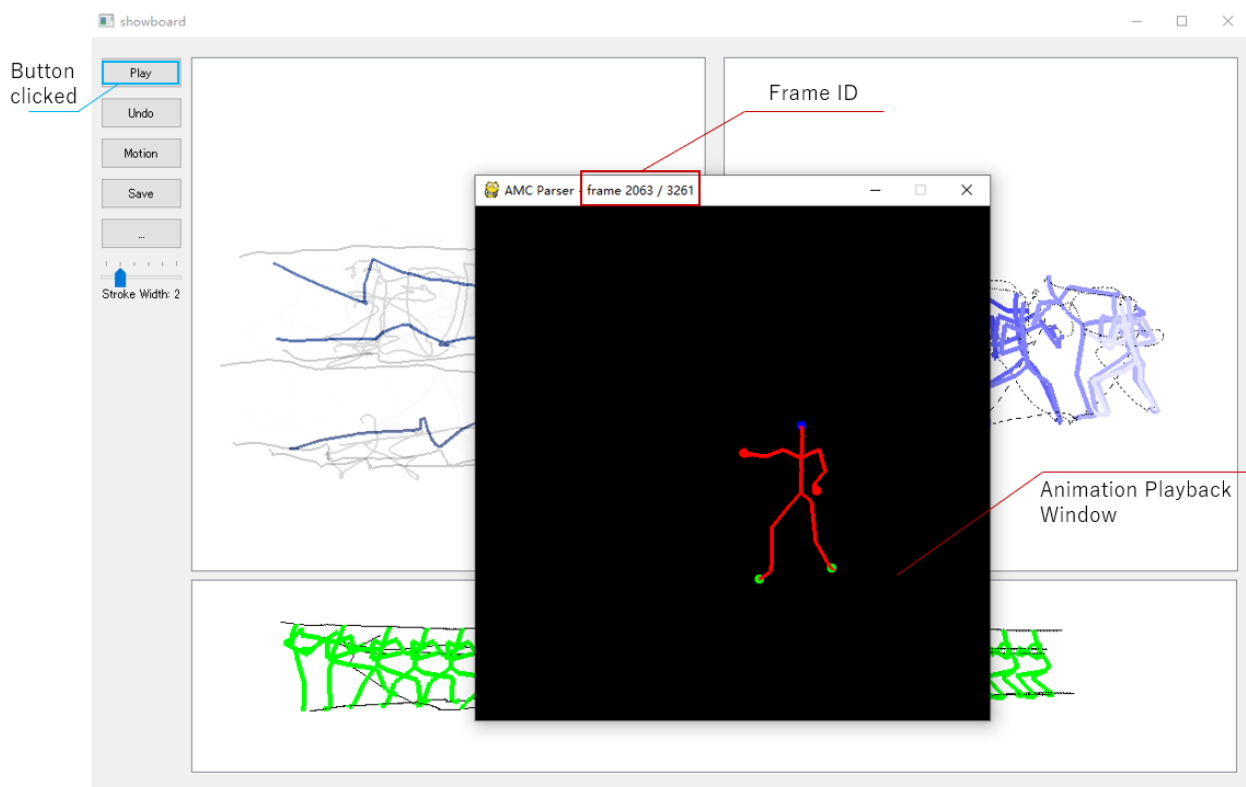
Figure 3.10: The third sub-window for animation play back.

Chapter 4

User Study & Evaluation

In this chapter, we firstly introduce the user study that is designed to verify the utility of proposed interface. We then describe the results that we observed from the user study and a result of questionnaire answered by the participants.

4.1 User Study

In this section, we introduce the environments of our experiment, the comparative & evaluation experiments of our user study.

4.1.1 Experiment Environment

The prototype was running in the follow system environment:

OS : Windows 10 Home x64

CPU : Intel Ⓡ Core(TM) i7-8700 CPU @ 3.20GHz 3.19GHz

GPU : Geforce GTX 2070 8GB

RAM : 16GB

Tools : Anaconda, PyOpenGL, PyGame, Python-OpenCV, PyQT5

We wrote the prototype totally in Python environment and set up the virtual system environment with the help of Anaconda env management.

4.1.2 Experiment Design

To construct our prototype, we totally chose 93 skeleton motions data from CMU motion data library which includes 20 motions of dance, 10 motions of walking in different speed, 10 motions of walking in different style, 14 motions of running, 3 motions of making cartwheel on the ground 11 motions of punching/kicking and 15 motions of skateboarding and 10 motions of jumping as shown in Figure 4.1.

For the comparative study, the participants were asked to imagine a scene that they plan to design a character animation, and they had to pick an expected motion from a specific category (e.g. found an expected walking motion). We asked the participants to retrieve desired motion from totally 4 categories of motion, walking, running, cartwheeling and punching/kicking separately. In order to ask the participants to find the target motion from our data library, we required some additional motion data as noise data. Note that dancing, jumping, and skateboarding are considered as the noise motion

in this user study. We recorded the time cost for each task.
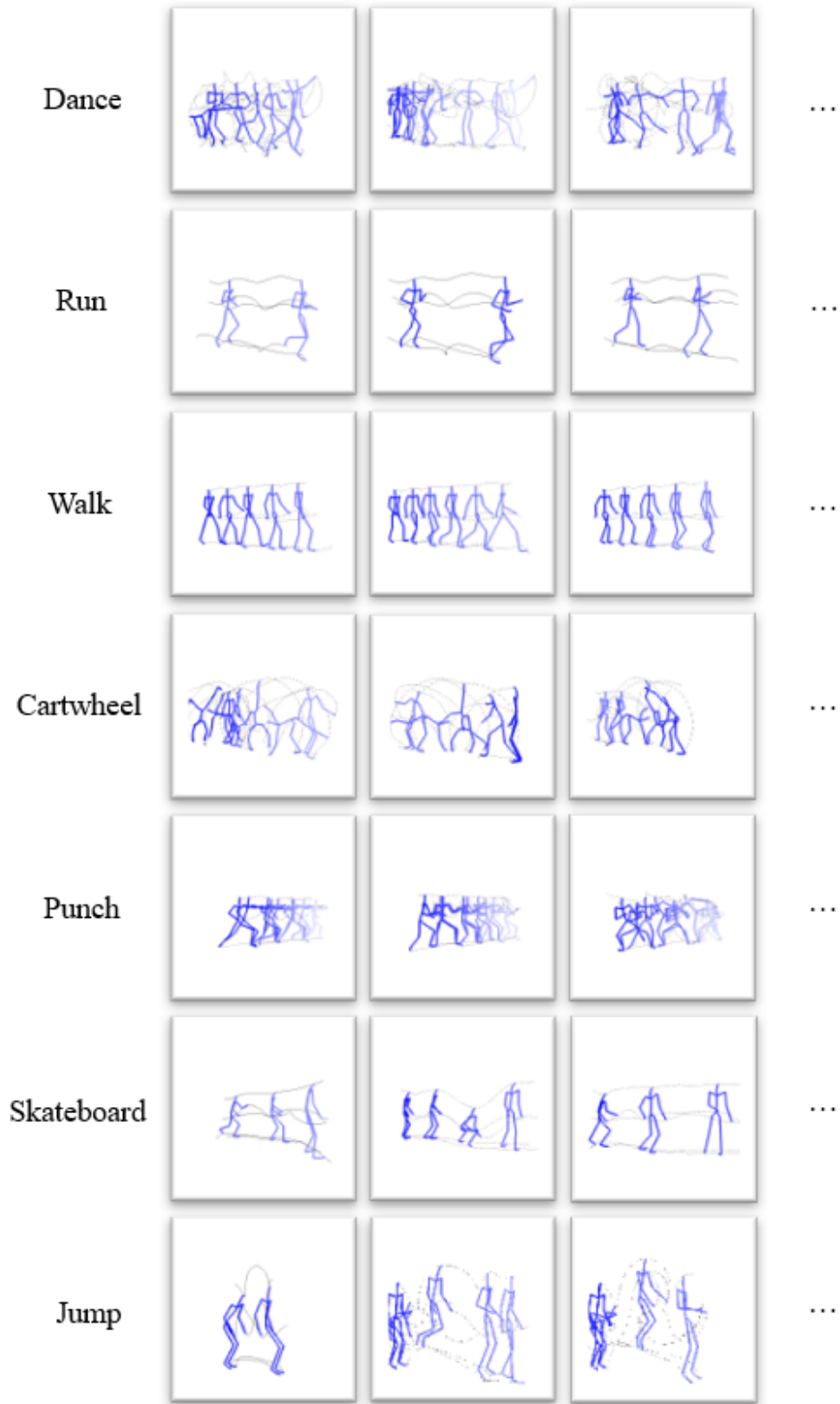
Figure 4.1: The motion dataset for constructing the prototype.

The participants were asked to examine two different types of interface to retrieve motion data (the details of the UI is described in section 4.2.1). For the first approach, similar to the public motion data library on the Internet, we only provided the motion map (absolute coordinate) for browsing. They were allowed look over the sequences of motion maps file-by-file on the provided program. The displaying order in the motion data list was randomly shuffled for each task. (Actually providing the motion map is better than some of motion libraries which has only keywords of motion contents.)

For the second approach, the participants were asked to apply the proposed interface to retrieve the desired motion from our motion library. They had approximately 5-minutes trial to familiar with our sketch-based interface before the experiment.

4.1.3 Evaluation Study Design

For the evaluation study of our interface, we conduct a questionnaire following the questions. Note that the first 5 questions adopted Likert-scaled with 5 points (1 for useless, 5 for very useful):

- Q1) Did the shadow-like guidance displayed in the main window provide hint to draw the motion trace?
- Q2) Did the motion map (absolute coordinated skeleton) displayed at the sub-window on the top right make you understand the overall skeleton motion easier?
- Q3) Did the motion map (relative coordinated skeleton) displayed at the sub-window in the bottom make you understand the overall skeleton motion easier?
- Q4) Did you have clearer understanding of the overall skeleton motion by browsing the animation in the playback window?
- Q5) Do you consider that the proposed interface simplified the process of retrieval motion data from a motion data library?
- Q6) Which interface do you prefer to apply when you are about to retrieve motion data from a motion data library? (browsing the motion map only or sketching the similar motion trace)
- Q7) With using the proposed interface, which category of motion is the easiest to define with strokes? (walking, running, cartwheeling or punching/kicking)
- Q8) Which joint movement do you think that you need to draw its trace to define a skeleton motion sequences? (head, hands, feet or others)

The result of the questionnaire is shown in table 4.1 and 4.2.

Participant	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
P1	5	4	4	5	5	With interface	walk	hands
P2	4	4	3	4	4	Without interface	walk	head
P3	5	5	5	4	5	With interface	walk	hands
P4	5	4	5	5	5	With interface	walk	hands
P5	5	5	5	5	4	With interface	cartwheel	head
P6	4	3	4	5	4	With interface	walk	hands
P7	4	3	3	4	4	With interface	Punch/kick	body(others)
P8	4	4	4	4	3	With interface	walk	hands
P9	4	5	5	5	2	Without interface	Punch/kick	hands

Table 4.1: The results of questionnaire.

4.2 Observation

In this section, we discuss the details of the experiments, analyzing the result of our objective data we collected in the user study and the subjective evaluation result of the questionnaire.

Totally 9 graduate students studying in JAIST have been taken part in our user study. (4 females and 5 males, the average age is around 25-years-old)

4.2.1 Objective Evaluation

Although none of the participant is a professional animator, we firstly set up scenario that each of them should retrieve a target motion as a reference to design character animation. We introduced the CMU motion data library to the participant, and showed them only searching with keywords is available on the CMU library website before asking them to do any task. Before asking them to do any task, we explained the proposed sketch-based interface and the functionality of each window and buttons. The environment of
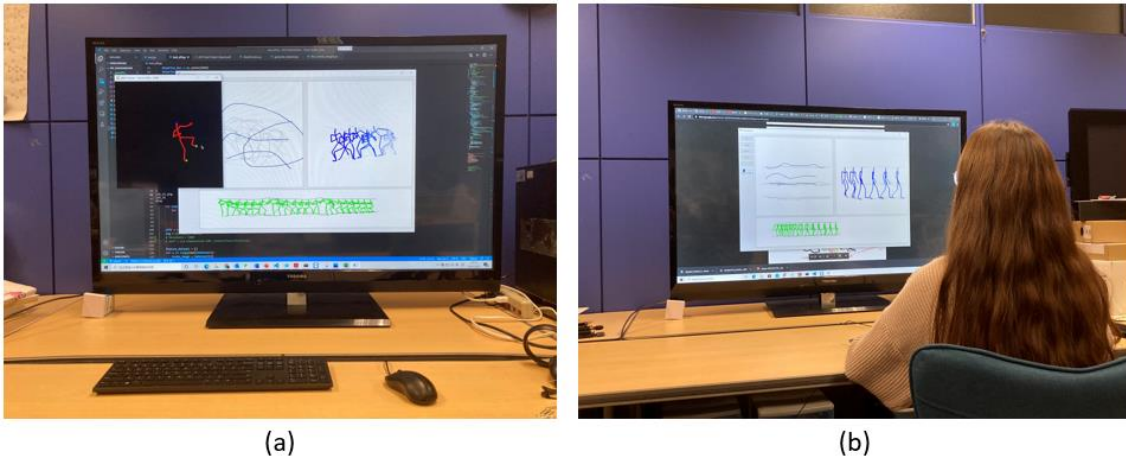
Figure 4.2: (a) the environment of experiment. (b) Scene of the operating system.

26
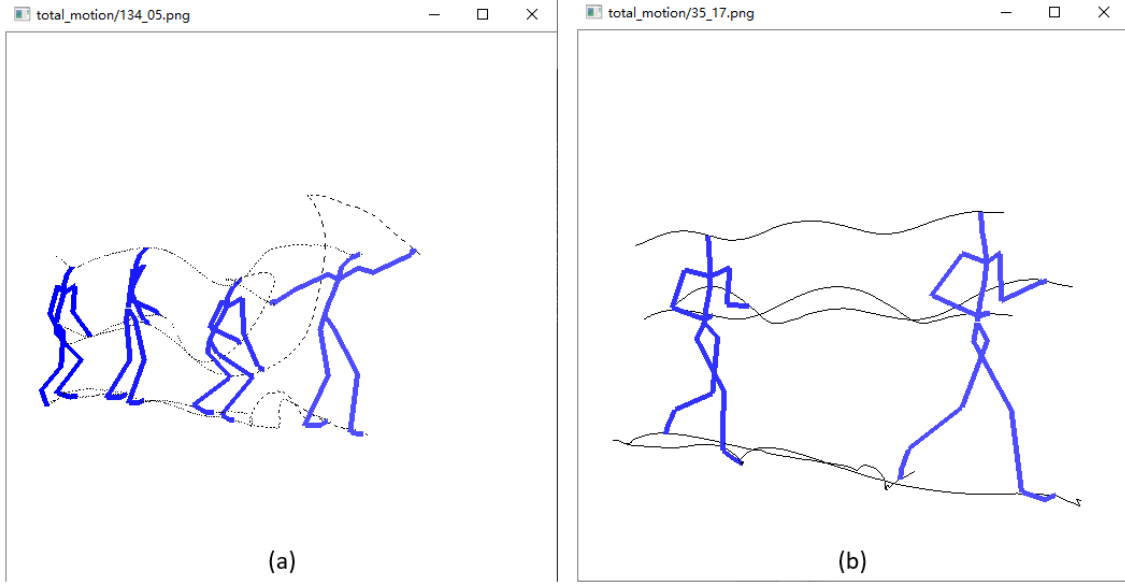
Figure 4.3: The window of interface utilized in the comparative group study. (a) A motion (skateboarding) from the random list; (b) A motion (running) from the random list.

the experiment is shown in figure 4.2.

For the comparative group of the objective evaluation study we set up a simple interface that the participant can only browse the motion map file-by-file and switch the map by pushing the 'left & right arrow' keys as shown in Figure 4.3. The order list of the motion file was randomly rolled. For another group which operating with the proposed sketch-based interface, each of the participant should be trained and experienced to familiar to the operation for approximately 5 minutes. Besides, the order to complete the tasks with using 2 interfaces is randomly switched for each participant.

In the study, the participanst were asked to retrieve the 4 target motion sequences in different categories. We measured the time cost by finishing each task of each participant. The result of the measurement is shown in Figure 4.4. In the box-and-whisker diagram, the gradient blue color boxes illustrates that the comparative group was cost a same level of less time to finish tasks (except for the cartwheeling because of the small proportion in the entire library), which were avengely 13.49s cost for retriving the walking motions, 12.18s for the running motions, 23.08s for the cartwheeling and 14.12s for the punching respectively. On the contrary, time cost level had a different by retrieving the target motions with proposed sketch-based interface, which were avengely 12.37s for the walking motions, 15.70s for running motions, 23.65s for cartwheeling motions and 14.12s for punching motions respectively.

According to our observation of the comparative study, it is considered that a

Time cost of searching target motion w/o proposed system

Figure 4.4: Time cost of searching target motion with and without proposed comparatively less time cost without using the sketch-based interface is because of small numbers of motion data (98 in total). The proportion of each target motion category was approximately 1/9 of the entire library (expected of the cartwheeling). Besides, the simple interface of the comparative group did not allow the participant to comparing the motion in the same category because the file list was randomly shifted. Therefore, once the target category of the motion shows up with switching different files by the participants, the participants would pick it up without further browsing.

On the other hand, it is mentioned that retrieving the cartwheeling motion spent the

Figure 4.5: (a) The trajectories' map of cartwheeling. (b) The trajectories' map of punching/kicking.

longest time. For the first reason, the numbers of motion in this category was comparatively less (3 only). Secondly, the trajectory maps of the cartwheeling are intricate and complex with multiple strokes' intersection as shown in Figure 4.5. Besides, it goes similarly in the punch motions, as shown in Figure 4.5, the numbers of frame in the punching motion sequences were more than about 2300 frames for each sequence. Therefore, the traces of the joints were twists and turns which would should take effort on defining them with the typical strokes.
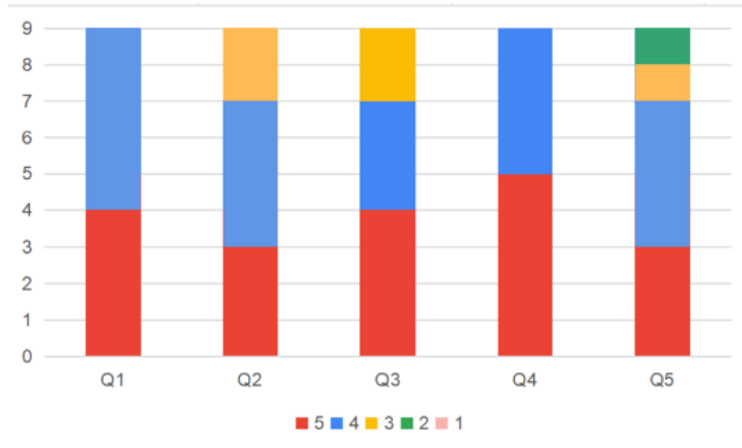
Two examples of the experiment data is shown in Figure 4.6.
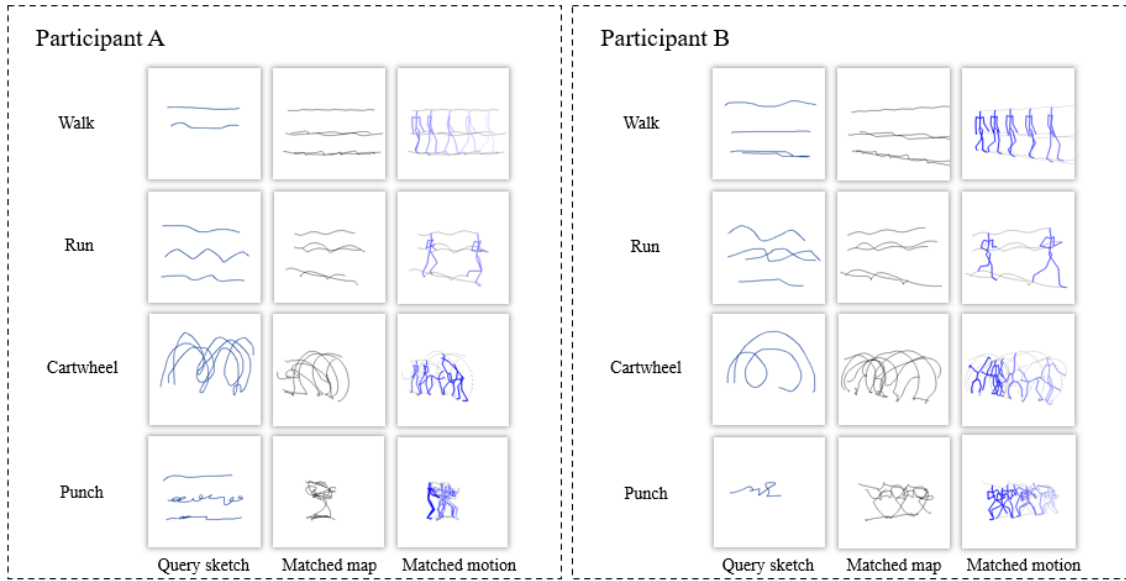
Figure 4.6: Two examples of experiment data

4.2.2 Subjective Evaluation

As mentioned before, we conducted a questionnaire to evaluate the validation of the proposed sketch-base interface. Figure 4.7 shows the result of the 5-levels Likert-scales

Figure 4.7: The scores of the 5-levels Likert-scales questions.

questions. Although the overall scores are in a mid-high level of subjective, it is considered that the validation of both of the motion maps (absolute coordinate one and relative coordinate one) are in a comparatively low level. On the other hand, the scored of the validation of the shadow-like guidance in the main window and the animation playback sub-window were highly evaluated. Therefore, it is certificated that browsing the static motion maps to manage a skeleton motion is not clear enough while the proposed shadow-like trace hints and playback interface is effective.

The results of question 6,7,8 is shown in Figure 4.8.
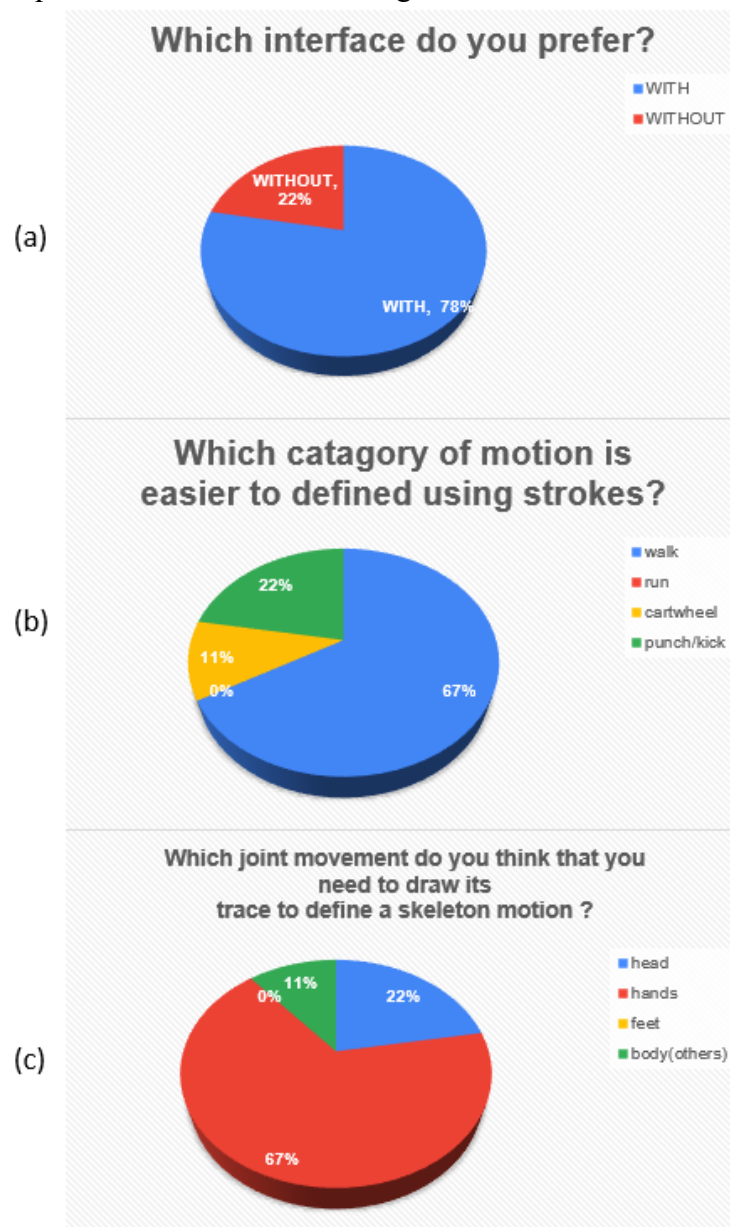
Figure 4.8: (a) Q6 result in Pie chart. (b) Q7 result in Pie chart. (c) Q8 result in Pie chart.

The results of question 7 shows that 7/9 participants in our user study preferred proposed interface for retrieving motions. Among the rest of 2 participants, one said that he/she found that browsing directly in motion map (using the interface of comparative group) to search the motion when the small numbers of the motion data in the library is more convenient.

The results of question 8 illustrates that applying proposed interface to retrieve the simple motion such as gaits is more effective than the complex one. It is considered that the reason of none of the participant had answered running is because the trajectories' map of the running motion has less difference from the gait motion. In other words, because of the static strokes fails to carry the speed information of the movement that running map are sometime misunderstood by the system to the gait map instead.

The results of question 9 illustrates that in most of cases, the participants prefer to define an entire skeleton motion by drawing the head and hands movement. To optimize the system and improve the effectiveness of the algorithm, it is not necessary to draw the feet movement from the skeleton movement on the trajectories' map.

4.3 Discussion & Limitation

In the early stage in designing the user study, the motion sequence of dancing was about to be included as the target motions. However, we found that the movement in a dancing map of an entire sequence is too complex and redundant so that the similarity measuring between query sketches and trace map fails to have an acceptable performance from the preliminary experiment. As shown in Figure 4.9, although we constructed the library using only 20 dance motions, the same sequence matched with different query sketches multiple times. As mentioned in the observation section, the proposed sketch-based interface had a better performance in retrieving a simple motion than a complex one. In terms of dealing the complex and redundant motion such as dance, it is necessary to slice the sequence into short sequences or reduce the trace's curve for example the joints of two feet.

Figure 4.10 shows the query sketches of all the participants in the user study. Although there were only 9 participants in the study, it is considered that the habit of drawing the strokes to define a motion has its rules. For example, the participants preferred to depict a gait motion by drawing the smooth curves while some wavy curves for running, the spring shape lines for the cartwheel, and the folded line for the punching/kicking. It was certificated that our interface can recognize users' expected motion only though even a single shape of curves, and drawing entirely 5 joints' traces is unnecessary.

Besides, all the query strokes are in the same width 2 pixels by default, so that

functionality of the 'strokes width' slider is seemed to be surplus.
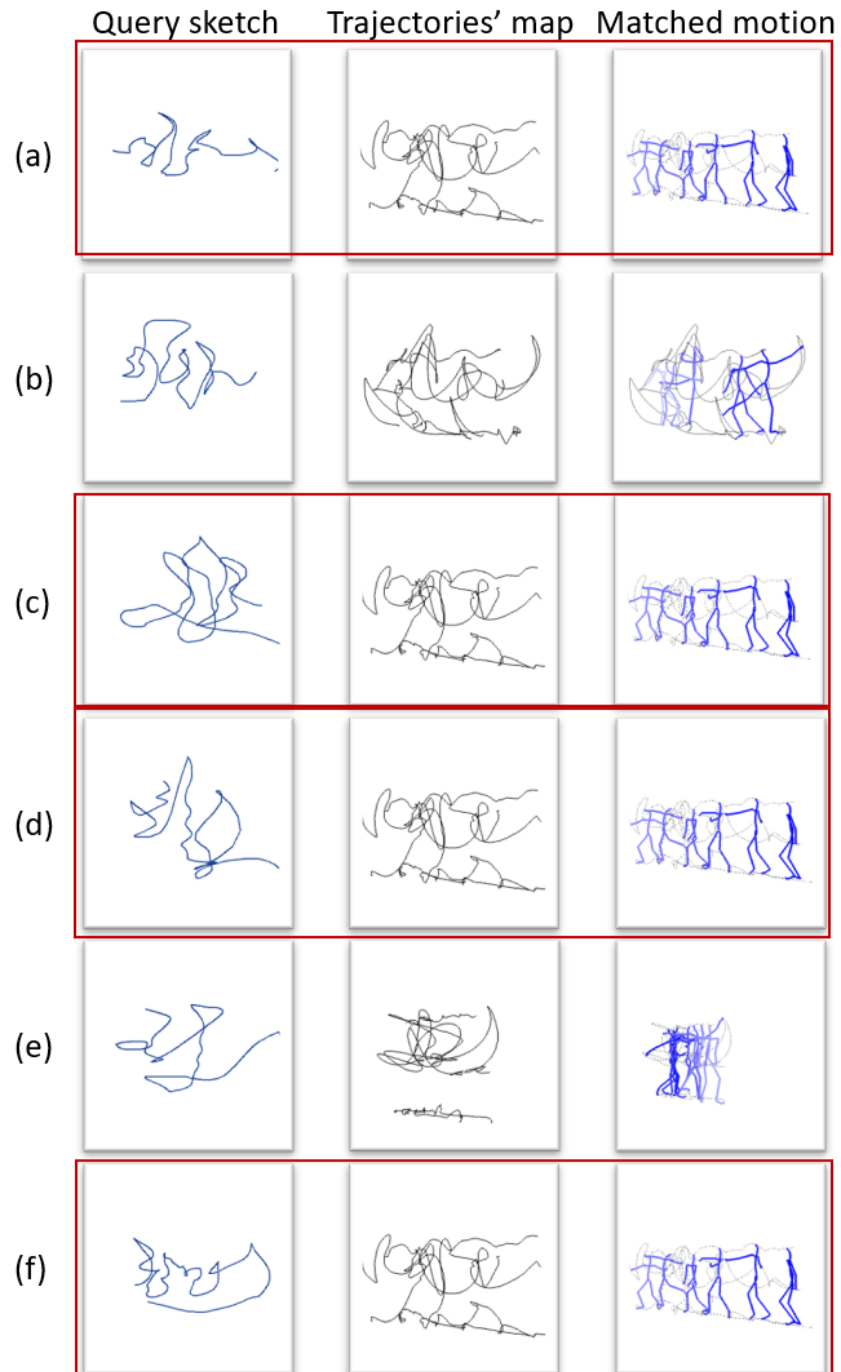
Figure 4.9: Failure similarity matching happen in (a),(c),(d) and(f).

Figure 4.10: The query sketches for each category of target motions.

Chapter 5
Conclusion & Future Works

In this thesis, we proposed Sketch2Motion, a sketch-based interface to help the users to retrieve their expected motion in a motion data library rather than picking up the motion by only browsing its motion map or searching only by keywords. In our interface, the users were allowed to browse not only the 2 motion maps both in the absolution coordinate and the relative coordinate, but also the skeleton animation playback in a sub-window interactively by sketching the traces of the 5 key joints. We utilized a SIFT descriptor to measure the similarity between the strokes input and the trajectories' map of motion in the data library. Besides, shadow-like guidance was provided behind the query sketch to support the users to draw the validation traces of the movement.

A user study is designed to certificate the effectiveness of the proposed sketch-based interface. In the comparative study, although the time cost of using our interface was in the same level of time cost of using the file-by-file browsing interface, we considered that it is because the sizes of the data library we constructed were too small that our interface was not able to perform better than another interface in terms of reducing time cost. It is convinced that our interface is more convenient in a motion library in a typical size with hundreds of motion categories.

In the evaluation study, a questionnaire was conducted to verify each functionality of the proposed UI using 5-points Likert-scales. According to the scores of the results, it is considered that the proposed shadow-like guidance and the animation playback in real-time are necessary when sketching to search for an expected motion. Both two of the motion maps (absolute coordinate and relative coordinate) were comparatively low scored, which proved that only browsing the static motion maps no matter in which of the coordinate system is not clear enough to understand an overall motion sequence. Browsing the skeleton animation in real-time is necessary.

To understand the habit of the users when they defined motion with strokes, we observed all the query sketch of the participants. It is considered that the user has the same habit of animating a gait motion with drawing smooth curve, a running motion with a wavy line, and so on. we believed that each category of motion related to its shape of curves. Therefore, the proposed sketch-based interface is not only applicable to the motion categories mentioned in our user study but also to some other types of motion. Our interface is simplified as much as the common user can draw it without difficulty.

The further goals of this research are introduced as follow:

- Optimizing the interface to apply for the complex motion. As mentioned in the discussion section, the proposed interface failed to well perform in retrieving the complex and redundant motion sequences such as dance. It is necessary to redraw the traces' map by sliding along motion sequence into several short sequences or reducing the numbers of joints that were projected out on the trajectories' map.

- For the limitation that our interface failed to meet the demands of depicting the speed of movement. The reason for the shortcut is that we used the SIFT descriptor to measure the similarity between the input query sketch and the image in the dataset, which is ignoring the temporal information on both sides. It is necessary to try a new motion trace arrangement method. For instance, measuring the length of a section of the trace and counting the numbers of keyframe sorted on the specific section to calculate the speed of joint movement on the trace.

- The proposed interface is only supporting the user for retrieving skeleton motion from the motion library, which is not enough for creating character animation. Figure 5.1 shows an example of covert AMC/ASF file to BVH file and playing in the Blender with the fixed structure of skeleton without retargeting to any character model. There are many pieces of research that are discussing the method to automatically achieve the retargeting tasks. One of the examples is K. Aberman's [29] work. However, it takes several hours to train a new character model even using their pre-trained DL framework. In our future work, we are about to look for a more suitable one to utilize as a plug-in of our sketch-based interface.
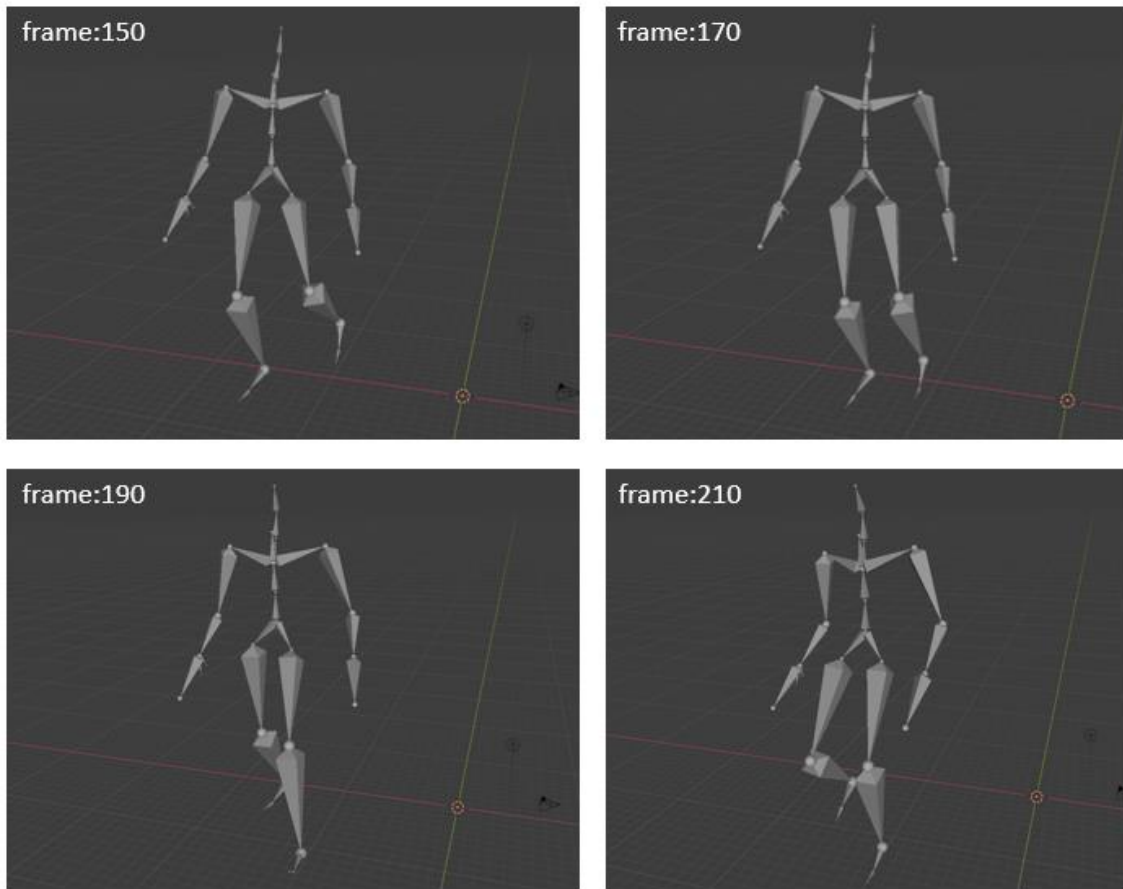
Figure 5.1: An example of covert AMC/ASF file to BVH file and playing in the Blender.

Acknowledgements

During the preparation of the master's thesis, I have received a lot of invaluable bits of help from many people. Their comments and advice contribute to the accomplishment of the thesis.

First of all, the profound gratitude should go to my supervisor Professor Kazunori Miyata and Senior Lecturer Haoran Xie for their teaching, supporting, and giving advice on my research. Professor Kazunori Miyata accepted me as a research student and gave me the chance to study and research with seniors in the Miyata Lab. He has inspired me when I was lost the direction in my research. His profound insight and accurateness about my paper taught me so much that they are engraved. He provided me with beneficial help and offered me precious comments during the whole process of my writing, without which the paper would not be what it is now.

Secondly, I would like to express my sincere gratitude to Senior Lecturer Haoran Xie for giving me the chance to participate in several cooperative-research programs which gives me great benefits. During the 2-and-half-years-long life in JAIST, Miyata Lab, I have had several chances to join the group research in the Lab with the other students in the laboratory. Fortunately, our works have been accepted by several journals and we proudly presented our work in front of other researchers in the same research field which is my precious memory.

Besides, by the honor of recommended, I gained an opportunity to take part in the research program which is cooperative with Igarashi Lab, The University of Tokyo, and the project is still advancing in progress. Meanwhile, I am sincerely grateful to Assistant Professor Tsukasa Fukusato, the Ph.D. students Chunqi Zhao and Toby Chong Long Hin in Igarashi Lab. The idea at the very beginning of this thesis would not be born without their inspiration in each brainstorming discussion.

Moreover, I would like to express my sincere gratitude to all the participants in the user study of this thesis. With their valuable suggestion and feedback, we finally learned lessons and gain experience in practice. This thesis would not be processed without their patient cooperation.

Last by not less, I would like to extend my deep gratefulness to my family who supported me and gave me the chance to study abroad in Japan which made my accomplishments possible.

Bibliography

[1] 2D Animation Software, Flash Animation | Adobe Animate. Adobe Animate, www.adobe.com/products/animate.html.

[2] Maya Software | Computer Animation & Modeling Software | Autodesk. Autodesk Maya, www.autodesk.com/products/maya/overview?support=ADVANCED&plc=MAYA&term=1-YEAR&quantity=1.

[3] Vicon | Award Winning Motion Capture Systems. Vicon, 13 Jan. 2021, www.vicon.com.

[4] Xsens. Home - Xsens 3D Motion Tracking. Xsen, www.xsens.com.

[5] Walker, Cordie. "K-VEST 6Dof Electro Magnetic System. Golf Science Lab, 21 Apr. 2017, golfsciencelab.com/3-types-motion-capture-systems-use/kvest-6d.

[6] Gypsy 7 Electro-Mechanical Motion Capture System. Gypsy 7, metamotion.com/gypsy/gypsy-motion-capture-system.html.

[7] Kinect - Windows App Development. Microsoft Kinect, developer.microsoft.com/en-us/windows/kinect.

[8] Intel® RealSenseTM Technology. Intel, www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html.

[9] Cao, Zhe, et al. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1 (2019)

[10] Fang, Hao-Shu et al. RMPE: Reginal Multi-person Pose Estimation. *IEEE International Conference on Computer Vision.* pp. 2353-2362, (2017)

[11] Marco, Romeo et al. A Reusabel Model for Emtional Biped Walk-Cycle Animation with Implicit Retargeting. *Articulated Motion and Deformable Objects,* vol 6169, (2010)

[12] Lowe, David G. Distinctive Image Features from Scale-Invariant Keypoints. *International Joranl of Computer Vision.* Vol.60 pp.91-110. (2004)

[13] Lee, Y.J., Zitnick, L. and Cohen, M. ShadowDraw: Real-Time User Guidance for Freehand Drawing, *ACM Transactions on Graphics*, vol.30, no.4 (2011)

[14] Igarashi, T., Matsuoka, S., and Tanaka, H. Teddy: A Sketching Interface for 3D Freeform Design, *SIGGRAPH '99: Proceedings of the 26th annual conference on*

Computer graphics and interactive techniques, pp.409-416 (1999)

[15] Eitz, M. et al. Sketch-Based Shape Retrieval, *ACM Transactions on Graphics*, vol.31, no.4 (2012)

[16] Zou, C. et al. Sketch-based Shape Retrieval using Pyramid-of-Parts, arXiv preprint arXiv:1502.04232. (2015)

[17] Xing, J. et al. Energy-Brushes: Interactive Tools for Illustrating Stylized Elemental Dynamics, *UIST '16: Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pp.755-766 (2016)

[18] Hu, Z.Y. et al. Sketch2VF: Sketch-Based Flow Design with Conditional Generative Adversarial Network, *Computer Animation and Virtual Worlds*, vol.30, no.3-4 (2019)

[19] Choi, B. et al. SketchiMo: Sketch-based Motion Editing for Articulated Characters, *ACM Transactions on Graphics*, vol.35, no,4 (2016)

[20] Choi, M. G., et al. Retrieval and Visualization of Human Motion Data via Stick Figures. *Computer Graphics Forum*, vol. 31, no. 7, 2012, pp. 2057–65. (2012)

[21] Choi, M.G. et al. Dynamic Comics for Hierarchical Abstraction of 3D Animation Data, *Computer Graphics Forum*, vol.32, no.7 (2013)

[22] Turmukhambetov, D. et al. Interactive Sketch-Driven Image Synthesis, *COMPUTER GRAPHICS forum*, vol.34, no.8, pp.130-142 (2015)

[23] Z. He, H. Xie and K. Miyata, Interactive Projection System for Calligraphy Practice," 2020 Nicograph International (NicoInt), pp. 55-61 (2020)

[24] Yasuda, H. et al. Motion Belts: Visualization of Human Motion Data on a Timeline, *IEICE Transactions on Information and Systems*, vol.E91-D, no.4, pp.1159-1167 (2008)

[25] Assa, J., Caspi, Y. and Cohen-or, D. Action Synopsis: Pose Selection and Illustration, *ACM Transactions on Graphics*, vol.24, no.3, pp.667-676 (2005)

[26] Bouvier-Zappa, S., Ostromoukhow, V. and Poulin, P. Motion Cues for Illustration of Skeletal Motion Capture Data, *NPAR '07: Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, pp.133-140 (2007)

[27] Zamora, S. and Sherwood, T. Sketch-Based Recognition System for General Articulated Skeletal Figures, *SBIM '10: Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, pp.119-126 (2010)

[28] Gleicher, M.. Retargetting motion to new characters. *Proceedings of the 25th annual conference on Computer graphics and interactive techniques.* pp. 33-42. (1998)

[29] Kfir, Aberman et al. Skeleton-Aware Networks for Deep Motion Retargeting, *ACM*

Transactions on Graphics, vol.39, no.4 (2020)

[30] Carnegie Mellon University - CMU Graphics Lab - Motion Capture Library. *CMU MoCap Dataset,* mocap.cs.cmu.edu.

[31] Eiter, Thomas, Mannila, Heikki. Computing Discrete Frechet Distance, *Computing Discrete Frechet Distance* (1994)