

Designing Canonical Form of Finger Motion Grammar in Leapmotion Contents

Byungseok Lee, Kangrae Park, Seokhyun Ghan and Seongah Chin*

Division of Media Software, Sungkyul University, South Korea

*Corresponding author

Abstract—This paper proposes methods that can standardize the different content motions that use Leapmotion in order to define and recognize them. A current Gesture interface is built and used to fit the individual contents. This creates confusion among the users because each applied field uses a different motion. Therefore, we defined the motions that were used in the contents in diverse fields into Motion grammar, proposed a method to recognize the defined language through a Motion Tree, and verified its accuracy through experiments.

Keywords—leapmotion; finger motion; motion grammar

I. INTRODUCTION

Interface technology has progressed from a command-line interface (CLI) to a graphical user interface (GUI). Current research is toward a natural user interface (NUI) in order to have more realistic mutual exchanges between contents and users. NUI is a new type of interface that does not use a traditional mouse or keyboard. The Voice Interface, Sensory Interface, Touch Interface, and Gesture Interface are some examples of NUI. Leapmotion, an interface that uses the hands, has a limited scope of recognition compared with the Kinect interface, but it has a higher rate of recognition. Leapmotion is being applied to make new types of contents[1-4]. However, for an interface to be user-friendly, it needs to be learned for a long period of time under a set of defined rules [5]. As for Leapmotion, there is no defined motion language in its canonical form, but different motions have been designed for individual content. For instance, there is the inconvenience of having to learn different motion languages for each type of content because they are all different.

Therefore, in this research, we define a motion language that can be used for universal contents, and we develop motion vocabularies that can be recognized. In order to do so, we extract the motions that are commonly used among the different types of contents. We then define a motion language that can be used universally based on the categorized motions. The defined motions are made into vocabularies and listed in a table, and the vocabularies are combined to form a motion grammar tree. We conduct a quantitative evaluation in order to verify the recognition rate of the defined motions.

II. FINGER MOTIONS IN CONTENT

The contents that use Leapmotion can be categorized by their objectives as amusement contents (game contents) or functional contents. The two categories have specific genres,

which are categorized and analyzed to extract commonly used motions in the main body of this paper.

TABLE I. MOTION CATEGORY BY AMUSEMENT CONTENT GENRES

Game Genre	Game Motion	Game Example
Action-FPS	Move, Jump, Run, Sit, Shoot, Reload, Change weapon	Sudden Attack, Special Force, AVA, etc.
Action-Combat	Move, Jump, Run, Attack (Skill)	King of Fighter, Street Fighter, etc.
Simulation-Racing/Aviation	Move, Break	Re-Volt, Asphalt, etc.
Adventure	Move, Jump, Functions	Cookie Run, Nom, etc.

A. Game and Motion

Games are categorized largely by platform and genre. The standard of categorizing games by genres is ambiguous since new intergenre games are constantly being developed. In this study, we categorized games into the following genres based on motion: Action-FPS, Action-Combat, Simulation-Aviation/Racing, Adventure, Sports, and Role-Playing. We excluded Sports and Role-Playing from the six genres. Since multiple players need to be manipulated in a Sports game, it is not fit to apply motion language to this genre of game. Role-Playing games have more freedom than other genres, but there are limitations in defining the motion language because there are a diverse number of cases, and thus the interface is also that much more complex.



FIGURE I. VIRTUAL DISASTER EXPERIENCE CONTENT

TABLE II. MOTION CATEGORY BY FUNCTIONAL CONTENT GENRES

Functional Contents	Motion	Examples
Experience and Creation	Zoom in, Zoom out, Use of tools (Draw, Paste, Cut, etc.), Rotate	Virtual Experience
Lecture and Learning	Play Video, Fast Motion Play, Stop, Reverse	Video Lecture

Therefore, in this paper, we categorize and analyze the motions for four genres: Action-FPS, Action-Combat, Simulation-Aviation/Racing, and Adventure. In addition, we extract common motions. We noticed that the motions can be categorized into those involving movement and action. We would like to define the motion language within a framework where the left hand is designated for motion, while the right hand is designated for action. Table I lists the representative motions for the game genres that we have categorized. These motions are necessary in developing Leapmotion-related contents.

B. Human-Computer Interaction

Recent advances in virtual reality technology have led to the creation of many experience-oriented learning and disaster drill contents. Figure 1 shows disaster experience content developed by CNBOX to enhance reaction capability in case a fire breaks out in a subway. E-Learning is a representative example of lecture content. It provides lecture videos online in order to overcome the limitations of online education including closeness and collectivity. Table II lists the representative motions used in each of the functional contents.

III. LEAPMOTION AND FINGER MOTION GRAMMAR

A. Leapmotion

Leapmotion, as shown in Figure 2, captures a hand motion in 3D, analyzes it, and carries out motion control on the contents [6]. The infrared camera within the radius captures the video information and determines the x , y , and z axes. By extracting certain points on the monitor, Leapmotion traces their trajectories and analyzes the traced points to recognize motion. Since Leapmotion recognizes motions within a space of approximately 8 ft 3 in within a 150° radius at 290 frames per second (fps), the subject on the screen also moves precisely with the motions of the user. In this research, we define motions by content based on Leapmotion principles and an application program interface (API).

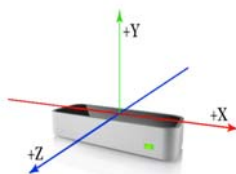


FIGURE II. LEAPMOTION

B. Leapmotion API

Leapmotion provides a diverse API. In addition to including hands and fingers, as shown in Figure 3, there are images, tools, and arms. Of the many types of API that exist in the marketplace, hands and fingers are most widely used. Hands-and-fingers APIs receive data from the uppermost frame. The API of the hand that received the data can distinguish between the left and right hands. The finger API can distinguish between the fingers and collects data including speed, location, and degree.

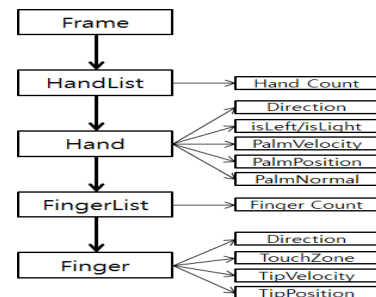


FIGURE III. HAND AND FINGER APIS

In this research, we define motion language using a hand-and-finger API. For example, we define shooting in Action-FPS contents as motion. The action of shooting involves the use of the right thumb and second finger. By applying the hand-and-finger API in the hand API stage, we determined whether there is a right hand. Then, in the finger API stage, we concluded that there are two fingers on the right hand, and that these two fingers are the thumb and second finger. When the degree of the x - z axis of the thumb is satisfied, motion is recognized. Once all these conditions are met, the action of shooting is recognized. Since the action of shooting has been recognized only for the right hand, we can combine the right and left hands by defining the direction and motion of the left hand.

C. Finger Motion and Structure

Motions are to be used in Action-FPS, Action-Combat, and Simulation-Racing/Aviation games. Since Action, Adventure and Simulation games need a combination of motions because they require the simultaneous use of both hands, we used the left hand for moving and the right hand for attack/function to define the motion. The defined finger-motion language categorizations and combinations are expressed through a tree structure. Finger motion grammar (MG) can be divided into Motion Lexicon (ML), Adverb + ML (AML), and Compound + ML (CML) through structural parsing. Figure 4 shows the Motion Tree (MT) that has parsed motion language.

In conclusion, in games and contents that use Leapmotion, all motions can be expressed in the grammar of a motion language Motion Tree (MT).

IV. RESULTS AND EVALUATION

A. Experiment Environment

The following conditions were set up to evaluate the motion language defined in this paper. The desktop computers for the simulation had Windows 7 64-bit operating systems, and the graphic card was a Geforce GTX 770. Unity 5.3.1f1 was installed as software, and Leapmotion was the hardware used.

B. Experiment Method and Results

The defined motions for each piece of content were extracted to conduct the experiment. We conducted a quantitative evaluation on the recognition rate by testing 20 times per motion. Content class A refers to Action, Adventure, and Racing/Aviation contents.

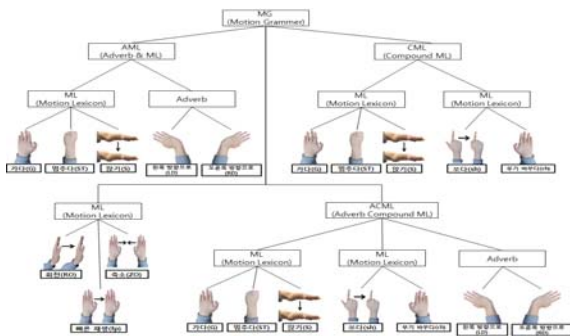


FIGURE IV. STRUCTURE OF MT (MOTION TREE)

Going left (LD + G), going right (RD + G), and going while shooting (G + sh) are vocabularies that were combined to form a grammar motion. The (LD + G) and (RD + G) motions were made only with the left hand, and their recognition rate was found to be 90%. The (G + sh) motion is a grammar combination of left-hand vocabulary and right-hand vocabulary, and its recognition rate was 80%. The (ST) motion is a grammar with a single left-hand vocabulary with a recognition rate of 90%. The motion recognition rate for amusement contents was relatively high.

An experiment on content class measured the recognition rate during two experiments, and the creative motions during two lectures and learning motions. The recognition rates for rotation (RO) and zoom in (ZI) showed an accuracy rate of 90% and 100%, respectively. The recognition rates for play (p) and rewind (rw) were also 90% and 100%, respectively (Figure 5).

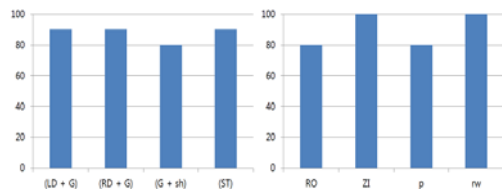


FIGURE V. SUCCESS RATE OF MOTION RECOGNITION

C. Conclusion

In this paper, we examined finger motions that are defined in contents that use Leapmotion, and defined a Motion Grammar that can be used universally. Using this information, we conducted an experiment. We selected Leapmotion as the appropriate interface for NUI, and categorized and defined universal motion languages by contents because motion languages vary among the contents that exist in the market. We looked into the representative motions that the categorized contents commonly use, and defined a motion language using a Leapmotion API. For Action, FPS, Adventure, and Racing/Aviation contents, we distinguished between the right and left hands. For experience and creation, and lecture and learning, contents, we defined motions that can be easily and readily accessed by users. By combining the vocabularies that were defined by distinguishing between the right and left hands, we created three types of grammar. A single motion vocabulary can be a type of grammar. A motion language tree was built for all content motions by applying the four types of grammar. We also conducted a quantitative evaluation for the defined motion language. The recognition rates for each piece of content were shown in a graph, and all contents showed a high recognition rate.

ACKNOWLEDGMENTS

This research was partially funded by National Research Foundation, NRF (No. 2015R1D1A1A01057725).

REFERENCES

- [1] LE Potter, J Araullo, L Carter, "The Leap Motion controller: A view on sign language," Proceedings of Computer-Human Interaction Conference, 2013, pp. 175-178
- [2] ES Silva, JAO de Abreu, JHP de Almeida, "A Preliminary Evaluation of the Leap Motion Sensor as Controller of New Digital Musical Instruments," Centro de Informática - Universidade Federal de Pernambuco (UFPE) Caixa Postal 50. 740-560- Recife - PE - Brasil
- [3] S Bak, C Lee, H Shin, "Edutainment content production utilizing the leap motion device," Proceedings of The Korea Contents Associations, Vol. 2015, No.5, 2015, pp. 345-346
- [4] K Kang, K. Kang, H Kim, G Jeon, S Hong, "Unity Game Development Using Leap-Motion," Proceedings of Korean Institute of Information and Scientists and Engineers, Vol. 2014, No.6, 2014, pp. 1974-1976
- [5] K Lee, "Designing physical game interfaces using gestures," The Korean Society for Computer Game, Vol. 24, No.1, 2011, pp. 135-144.
- [6] Leapmotion, <https://www.leapmotion.com/>, 2016, 06.