

Research Article

Structural Motion Grammar for Universal Use of Leap Motion: Amusement and Functional Contents Focused

Byungseok Lee, Donghwe Lee, and Seongah Chin 

Division of Media Software, Sungkyul University, Anyang City, Republic of Korea

Correspondence should be addressed to Seongah Chin; solideochin@gmail.com

Received 19 May 2017; Revised 17 November 2017; Accepted 27 November 2017; Published 9 January 2018

Academic Editor: Stefano Stassi

Copyright © 2018 Byungseok Lee et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Motions using Leap Motion controller are not standardized while the use of it is spreading in media contents. Each content defines its own motions, thereby creating confusion for users. Therefore, to alleviate user inconvenience, this study categorized the commonly used motion by Amusement and Functional Contents and defined the Structural Motion Grammar that can be universally used based on the classification. To this end, the Motion Lexicon was defined, which is a fundamental motion vocabulary, and an algorithm that enables real-time recognition of Structural Motion Grammar was developed. Moreover, the proposed method was verified by user evaluation and quantitative comparison tests.

1. Introduction

Interface technology, which supports the interaction between content and users, is continuously being developed. Recently, the technology is transforming into a natural user interface (NUI) method that provides users with a bigger sense of reality compared with the conventional method, which focusses on the use of mouse and keyboard. NUI is an up-to-date means of interacting with computers that has gradually drawn more interests in human-computer interaction (HCI). NUI comprises voice interface, sensory interface, touch interface, and gesture interface. Leap Motion is a finger gesture interface-supported device [1, 2]. The infrared ray cameras attached to the Leap Motion controller capture and analyse the hand gesture, and the content recognizes the motion. The Leap Motion controller introduces a new novel gesture and position tracking system with submillimeter accuracy. Its motion sensing precision is unmatched by any depth camera currently available. It can track all 10 of the human fingers simultaneously. As stated by the manufacturer, the accuracy in the detection of each fingertip position is approximately 0.01 mm, with a frame rate of up to 300 fps.

For these benefits, the Leap Motion controller is widely used in various applications such as games [3], sign languages

[4], musical instruments [5], mixed reality [6], and rehabilitation and medical applications [7].

In particular, Leap Motion gesture recognition in Amusement (game) Contents plays a crucial role in keeping the player engrossed in the game. It also increases the immersive sense of the Amusement Content because Leap Motion uses the player's gestures without any controllers in real time as the player interacts with the content. Games that use gesture recognition can capture the player's attention easily through the progress of the game [8].

Research on the recognition of Leap Motion has been carried out in technical studies. Some studies on the use of SVM were reported in [9], and studies using HMM were investigated in [10–12]. However, these studies use machine learning, which requires feature extraction, normalization, and time-consuming training procedures.

As described above, we realise the use of Leap Motion in contents is expanding and the technology of recognition has a cumbersome preprocessing task. Although many studies investigated movement recognition through Leap Motion and content application, authors have not found any literature reported on standardized motion grammar. This study actually is designed to target leap motion gestures that have been used in games since game users are inconvenienced by having to learn different motions for content because they

all have their own motions. A preliminary conference paper is shown in [13].

To this end, this study defined the Motion Lexicon (ML) that can be universally used in Amusement and Functional Contents and designed the Structural Motion Grammar (SMG) composed of the combination of ML. Then, the tree of SMG was recognized in real time thorough coupling a motion API without using complex procedures such as feature extraction and training process like a machine learning algorithm. Then, the defined motions were then tested for verification.

2. Related Work

Researchers have studied the accuracy and robustness of Leap Motion [14, 15]. Weichert et al. [14] analysed the accuracy and robustness of Leap Motion and applied the research on industrial robots. Guna et al. [15] conducted research on the accuracy and reliability of the static and dynamic movements of Leap Motion.

The Leap Motion's movement recognition has also been investigated [16–21]. Marin et al. [16, 17] conducted research on the multiclass classifier by coupling Leap Motion with a Kinect and depth camera, while Vikram et al. [18] studied the recognition of handwritten characters. Lu et al. [19] proposed the Hidden Conditional Neural Field (HCNF) classifier to recognize the moving gestures. Boyali et al. [20] researched the robotic wheelchair control, which applied block sparse, sparse representative, and classification. Seixas et al. [21] compared the screen tab and selected gesture of both hands.

The use of Leap Motion on sign language is also being investigated [22–24]. Chuan et al. [22] investigated the recognition of English sign language using the 3D motion sensor, while Khan et al. [23] researched the prototype that can convert sign language to text. Mohandes et al. [24] investigated the Arabic sign language recognition.

Researchers also investigated content using Leap Motion [25–27]. The research evaluated 3D pointing tasks using Leap Motion sensor to support 3D object manipulation [25] through some controlled experiments including exposing test subjects to pointing task and object deformation, measuring the time taken to perform mesh extrusion and object translation. Sutton [26] presented the air painting method using Leap Motion that could be an input data to the Corel Painter Freestyle application. The painting process through gestures was implemented. A study about sound synthesis and interactive live performance using Leap Motion was also reported [27]. The study implemented a 5-grain granular synthesizer making users trigger individual grains.

And there were also studies about various contents and techniques using Leap Motion, aforementioned in Introduction [3, 6, 7]. Lee et al. [3] studied a game model using Leap Motion that combined gesture-dependent impact levels with the physical characteristics of players. A game was realised in which the gesture was recognized to be associated with a player's gesture with a gesture-specific weight. Davis et al. [6] proposed the work to establish new modalities

in interactions for architecture students in mixed reality environments. The menu interface design supported the real-time design of large interior architectural spaces experienced in mixed reality environments. Iosa et al. [7] conducted a study to test the feasibility, the compliance, and the potential efficacy of using Leap Motion controller-based system for progressing the recovery of elderly patients with ischemic stroke.

3. Methods

To accomplish the proposed method, the common motions used in Amusement and Functional Contents were first classified. Based on the classifications, the ML that can be used universally was defined. Then, SMG was defined through the combination of ML. Also, the recognition step is provided.

Figure 1 shows the overall flow of the proposed method as an example. Leap Motion, which is a form of NUI, enables the free movement of hands and its recognition. To define ML, the contents were divided into Amusement and Functional Contents. The representative motions were then selected. Then, we defined the Structural Motion Grammar (SMG) composed of the combination of ML. Every motion can be represented in the SMG that is visualized in a tree structure. Prior to defining the selected motions, the features of Leap Motion API were analysed to define the ML. For instance, Leap Motion API translated the orders by specifying them from the top to the bottom classes based on the top-down method. When the first condition of identifying static or dynamic movement was applied, the identification of the motion whether it is static or dynamic will be possible. When the second condition of hand API was applied, the information on hands can be classified. When the last condition of finger API was applied, the information on fingers can be classified. Based on this information, the differentiated motions can be defined and can be laid out in diverse forms of SMG. More comprehensive gestures will be defined in the following sections.

3.1. Content Classification. To define the universal motions that use Leap Motion, the representative motions are needed to be extracted by each content classification. The digital contents where leap motion is applicable can be classified into Amusement Content and Functional Content based on their purposes. Both types of contents have subgenres, and commonly used motions were extracted through the classification and analysis of the genres.

3.1.1. Amusement Content. Amusement Content is also known as the game content. This content can be classified into the following subgenres based on the motions: Action, FPS (First Person Shooting), Simulation-Racing/Flight, Arcade, Sports, and Role-playing. Of the six genres, Sports and Role-playing were excluded because they did not fit in the current study. Sports games were not fit for Leap Motion usage because multiple players need to be controlled simultaneously.

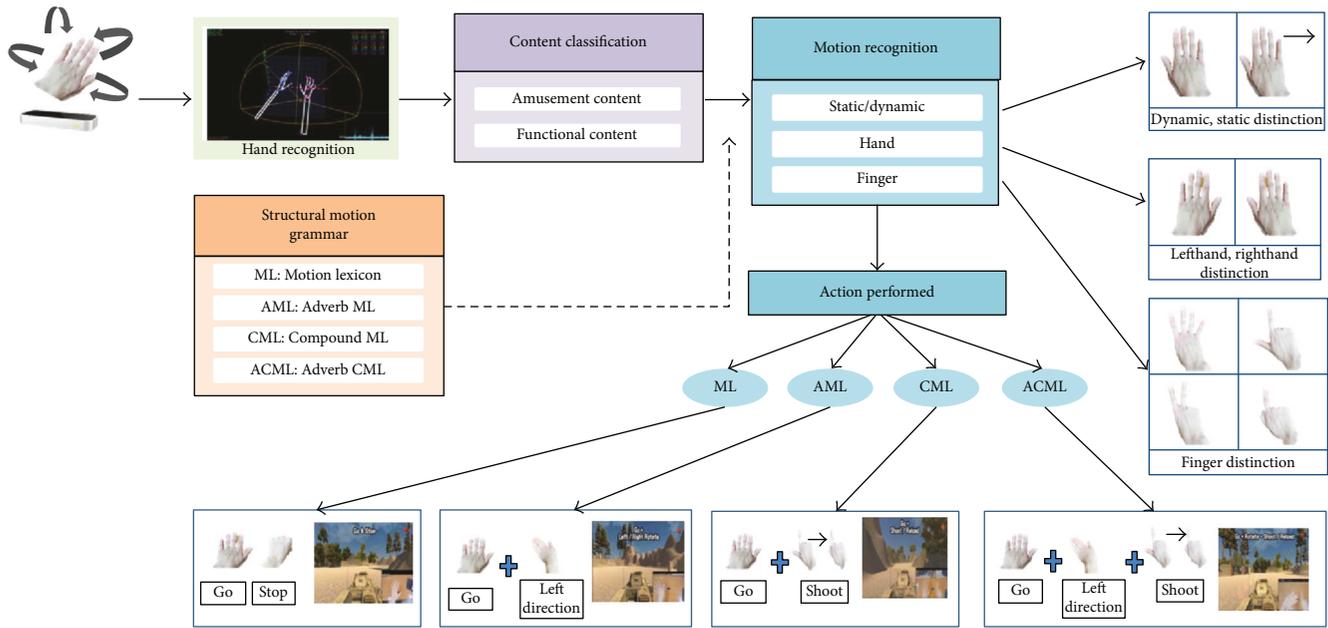


FIGURE 1: Schematic of the proposed method.

For Role-playing games, which have a high level of freedom, defining the motion has limitations because its interface and the number of possibilities are very complex and diverse.

To this end, the four genres, namely, FPS, Action, Simulation-Racing/Flight and Arcade, were analysed and common motions were extracted. Table 1 shows the representative motions by game genres. Within the FPS genre of Amusement Content, the “Sudden Attack (Nexon Co.)” is a representative game, and its motions are: “Move,” “Jump,” “Run,” “Sit,” “Shot,” “Reload,” and “Weapon Change.” In Table 1, three games (Sudden Attack, King of Fighters, and Cookie Run) are just representative examples of games from lots of games that we looked into to define the common motions.

The motions can be comprehensively categorized into movement and action. In this study, ML is defined based on the framework that the left hand is the movement while the right hand is the action.

3.1.2. Functional Content. Functional Content was classified into Experience and Creation Content and Teaching and Learning Content. With the recent expansion of the virtual reality market, numerous Experience contents or disaster reaction training contents use NUL. A representative example of lecture content is e-Learning, which is a form of Teaching and Learning Content that provides lecture videos online to overcome the drawbacks of offline education, such as being closed and collective. Table 2 shows the representative motions used by each Functional Content. The VR Museum of Fine Art (Steam VR Co.) is a representative example of Experience and Creation Content, and its motions are “Zoom In,” “Zoom Out,” “Using Tool,” and “Rotation.”

3.2. Motion Lexicon. Motion Lexicon (ML) consists of the motions that have been analysed within the Amusement

Content and Functional Content using the hand and finger API. To define ML, the hand and finger API reflecting the features of the genres have been analysed. Tables 3 and 4 show the defined ML by contents. More specifically, Table 3 defines the motions for both left and right hands to be used for FPS, Action, Simulation-Racing/Flight, Arcade, Sports, and Role-playing games. For Action, Simulation-Racing/Flight and Arcade games, the left hand was defined for movement and the right hand was defined for action because both motions occur simultaneously. Table 3 shows the details of ML, image, and motion principle. “Go” for the left hand was denoted by having all fingers straight, whereas “Stop” was denoted by having all fingers folded. When defining ML, “Jump” or “Sit” motions were linked to the up and down directions. “Shot,” “Reload,” and “Weapon Change” motions for the right hand were also defined by linking with the actual motion.

For Functional Content, “Zoom In,” “Rotation,” “Play,” “Pause,” and “Rewind” were representative motions. Given a very wide range of motions, not all of them can be defined. Therefore, motions that were commonly used have been defined.

Table 4 shows the Experience and Creation Contents of ML and explains their ML, image and motion principle. Here, ML comprises “Zoom In,” “Zoom Out,” and “Rotation.” For “Zoom In” and “Zoom Out,” the Vector3 coordinate was applied to both hands, and the movement of the x -axis was recognized as the distance. “Zoom Out” was defined as having both hands together, and “Zoom In” as having both hands apart. For “Rotation,” the horizontal and vertical condition was applied to the hand to identify whether the hand was horizontally positioned. The rotating counterclockwise motion was defined when the left hand moved towards the right side of the x -axis, and the clockwise rotation motion was defined when the right hand moved to the left side of the x -axis.

TABLE 1: Motion category by Amusement Content genres.

Game genre	Motion	Game example
FPS	Move, Jump, Run, Sit, Shot, Reload, Weapon Change	 <p>Sudden Attack (Nexon Co.)</p>
Action	Move, Jump, Run, Attack (Skill)	 <p>King of Fighters (SNK)</p>
Arcade	Move, Jump, Function	 <p>Cookie Run (Devsisters)</p>

TABLE 2: Motion category by Functional Content genres.

Functional Content	Motion	Example
Experience and Creation	Zoom In, Zoom Out, Using Tool (Drawing, Attaching, Cutting, and so forth), Rotation, and so forth	 <p>VR Museum of Fine Art (Steam VR Co.)</p>
Teaching and Learning	Play, Fast Play, Stop, Rewind	 <p>ICT in Education (KRCS Group)</p>

TABLE 3: Amusement Content Motion Lexicon.

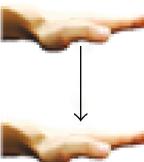
ML	Image	Motion principle
Go (G)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Number of finger condition
Stop (ST)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Number of finger condition
Left Direction (LD)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Degree condition
Left Hand		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Degree condition
Right Direction (RD)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Vector3 coordinate condition
Jump (J)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Vector3 coordinate condition
Sit (S)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Vector3 coordinate condition
Roll (R)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Vector3 coordinate condition

TABLE 3: Continued.

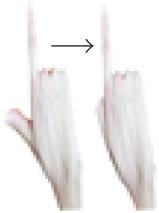
ML	Image	Motion principle
Shot (sh)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Number of finger condition (iv) Degree condition
Reload (r)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Number of finger condition
Weapon Change (ch)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Horizontal and vertical condition
Right Hand		
Kick (k)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Vector3 coordinate condition
Punch (p)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Vector3 coordinate condition
Function1 (F1)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Number of finger condition
Function2 (F2)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Number of finger condition

TABLE 3: Continued.

ML	Image	Motion principle
Drift (D)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Horizontal and vertical condition (iv) Number of finger condition
Booster (B)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Number of finger condition

Table 4 shows the ML defined in relation to the Teaching and Learning Content motions. Here, ML comprises “Play,” “Fast Play,” “Rewind,” and “Pause.” The horizontal and vertical conditions and Vector3 coordinate condition were applied to both hands, and “Play” was defined when the distance of the two hands on the x -axis was 0. Specifically, this is the same motion as clapping.

For “Fast Play” and “Rewind,” the movement was the same, but with different hands and directions. For “Fast Play,” the left hand was moved to the right side of the x -axis; while for “Rewind,” the right hand was moved to the left side of the x -axis. For “Pause,” the number of finger condition was applied to identify whether there are two left fingers.

3.3. Structural Motion Grammar. Structural Motion Grammar is a combination and grammaticalization of the aforementioned ML that has been defined. It consists of ML (Motion Lexicon), AML (Adverb and ML), CML (Compound ML), and ACML (Adverb and Compound ML). Figure 2 is a schematic tree of the classification and coupling of the ML that has been defined.

ML can be SMG by itself, such as the “Rotation” motion of the Experience and Creation Content. SMG is connected to ML. The process of “Rotation” motion has been identified with arrows within the schematic tree.

AML is a combination of ML and Adverb and Adverb was used as a part of speech that supports ML. For instance, for the left hand motion that was responsible for movement, the ML of “Go” was recognized and, at the same time, the SMG of the “Right Direction + Go” was expressed with the coupling of the Adverb of “Right Direction.” Within the schematic tree, the SMG leads to AML, which then leads to the ML/Adverb. The process of “Right Direction + Go” motion has been identified in arrows on the schematic tree.

CML was used when two types of motions were executed using ML and ML. For example, the left hand that was responsible for movement recognizes the ML of “Go,” and at the same time, the right hand can express the “Shot” motion with the integration of ML. On the schematic tree, SMG leads to CML, which then leads to ML/ML. The process of “Go + Shot” motion has been identified with arrows on the schematic tree.

ACML is a combination of ML and ML and Adverb vocabularies and was used when three motions were executed. For instance, the left hand responsible for movement recognizes the ML of “Go” and also recognizes the Adverb of “Left Direction” simultaneously. The right hand can express “Shot” with the integration of ML. On the schematic tree, the SMG leads to ACML, which then leads to the ML/ML/Adverb. The process of “Left Direction + Go + Shot” was identified with arrows on the schematic tree. In this study, the vocabulary combinations based on the aforementioned schematic tree have been used to define the SMG. The red dotted arrows indicate the recognition procedures that satisfy SMG. For example in Figure 2, Go and Shot literally means that a game player wants to make a tank go forward and shot at the same time. Thus, SMG can be classified into CML broken down into ML (Go) and ML (Shot).

A formal representation of SMG is the form of context-free grammar (CFG) since SMG can be broken down into a set of production rules. SMG illustrates all possible motions in given formal motions. We also define SMG as a theoretical form as below.

$$\begin{aligned}
 \text{SMG} &= \text{AML} \parallel \text{CML} \parallel \text{ACML} \parallel \text{ML}, \\
 \text{AML} &= \text{ML} + \text{Adverb}, \\
 \text{CML} &= \text{ML} + \text{ML}, \\
 \text{ACML} &= \text{ML} + \text{ML} + \text{Adverb}, \\
 \text{ML} &= \text{G} \parallel \text{ST} \parallel \text{S} \parallel \text{LD} \parallel \text{RD} \parallel \text{J} \parallel \text{S} \parallel \text{R} \parallel \text{sh} \parallel \text{r} \parallel \text{ch} \parallel \text{k} \parallel \text{p} \parallel \\
 &\text{F1} \parallel \text{F2} \parallel \text{D} \parallel \text{B} \parallel \text{ZI} \parallel \text{ZO} \parallel \text{RO} \parallel \text{p} \parallel \text{fp} \parallel \text{rw} \parallel \text{PA}, \\
 \text{Adverb} &= \text{LD} \parallel \text{RD}.
 \end{aligned}$$

3.4. Motion Recognition. Given that SMG has a combination of ML that represents a motion either using one hand or two hands, the SMG is decomposed into four children ML, AML, CML, or ACML; then, the recognition steps of ML are carried out. Recognition refers to the conditions that can explain the recognizable API on the Leap Motion device and define the motions. Leap Motion, which is a form of NUI, provides various APIs [2]. Among the numerous APIs, most of the contents in the market use the hand and finger API. These contents receive their data from the upper-most frame, where the hand is recognized to track and collect information. The hand API that has received the data can recognize the

TABLE 4: Functional Content Motion Lexicon.

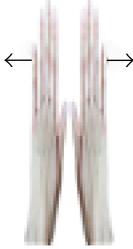
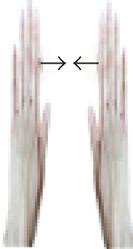
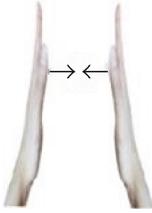
Functional Content	ML	Image	Motion principle
Experience and Creation	Zoom In (ZI)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Vector3 coordinate condition
	Zoom Out (ZO)		
	Rotation (RO)		
Teaching and Learning	Play (p)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Horizontal and vertical condition (iv) Vector3 coordinate condition
	Fast Play (fp)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Vector3 coordinate condition
	Rewind (rw)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Vector3 coordinate condition

TABLE 4: Continued.

Functional Content	ML	Image	Motion principle
	Pause (PA)		(i) Dynamic and static classification condition (ii) Left and right hand classification condition (iii) Number of finger condition

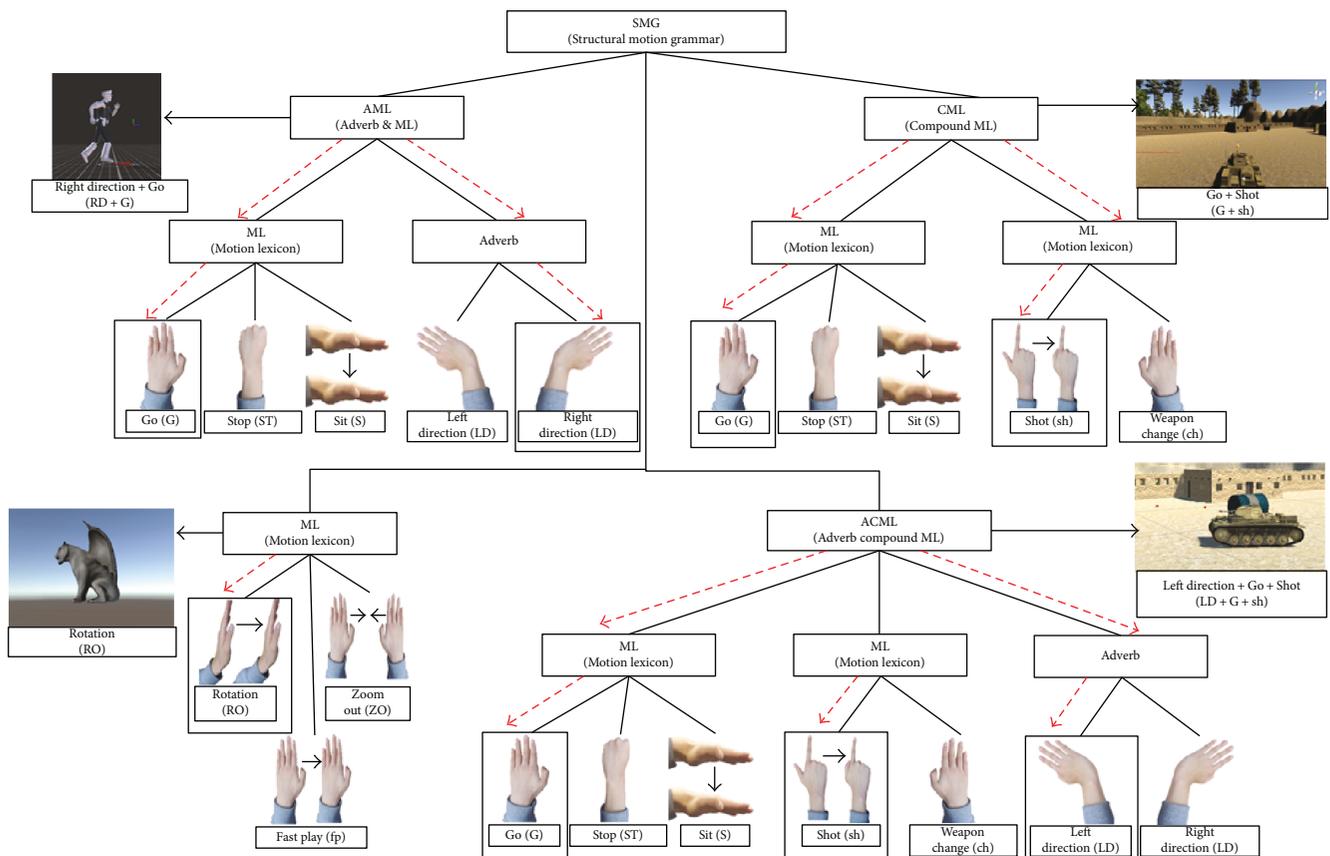


FIGURE 2: Structure of the motion grammar tree.

existence of the left or right hand and distinguish the left from the right hand. In addition, the API can identify the speed, location, and degree of the hand. Finger API can distinguish each finger and identify the speed, location, and degree of the fingers. While the data on speed and location were continuously updated, the former data was compared with the current data by tracking the hands and fingers. These comparison results can help distinguish whether ML is dynamic or static.

The algorithms SMG (mr_SMG), ML or ML_Adverb (mr_ML and mr_ML_Adverb), Hand Count (HC), Hand Feature (HF), Finger Count (FC), and Finger Feature (FF) are defined as shown in Figure 3. Supposing that the shooting motion has been defined within the FPS content, the first step

was to apply the dynamic and static classification conditions to the shooting motion. Then, using the data on hand API, the classification conditions for the left and right hands were applied. Finally, using the data on finger API, the conditions on the number of fingers on the right hand, as well as the degrees of the fingers, were applied. When two fingers of the right hand were used, the API identifies whether the fingers are thumb and forefinger and distinguishes the x - z axis degree of the thumb. The shooting motion was recognized only when all of the aforementioned conditions were set. Given that the shooting motion has been defined only for the right hand, the direction and movement motions will be defined to the left hand, enabling the use of both hands for manipulation.

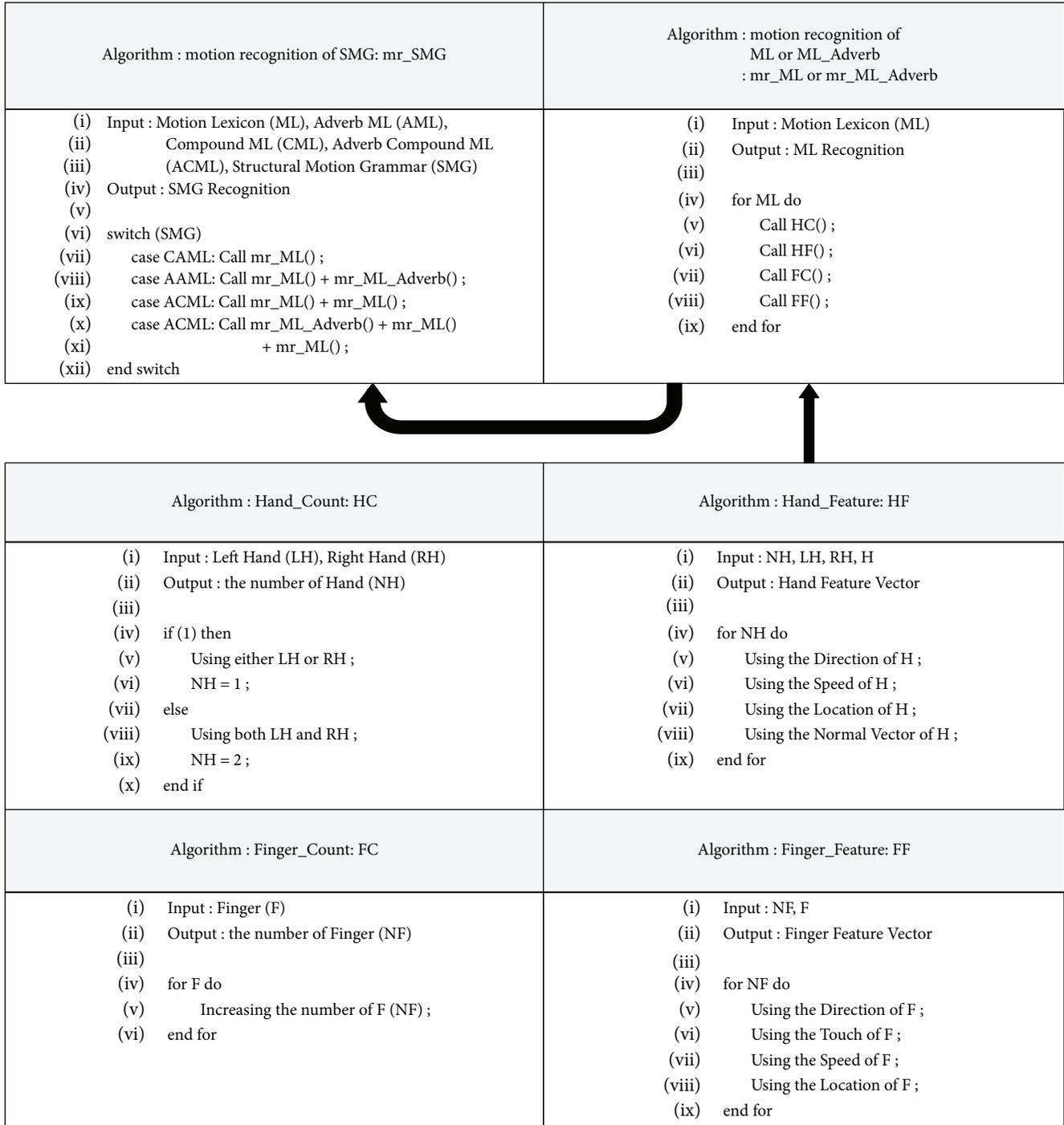


FIGURE 3: Motion recognition algorithm.

4. Experiments

4.1. SMG Recognition Rate Comparison Test. The following experimental environment was set up to evaluate the SMG suggested in this study. The desktop used for simulation was installed with Window 7 64bit OS, with Geforce GTX 770 as the graphics card. For software, Unity 5.3.1f1 version was installed, and Leap Motion was established

for the hardware. The motion recognition module was developed using C#.

For the test method, the Amusement and Functional Contents motions defined in this study and established into grammar (ours) were compared with the Leap Motion SVM [28] method through a quantitative evaluation. Each motion was tested 20 times as the number of input. The inputs are composed of the features of each gesture that

are resampled to the number of points. The training count-down value is set to three in which the training begins in three seconds.

The correlation output value above 0.7 is thought of as recognized as well. And the recognition rates were illustrated on a graph. Table 5 shows the quantitative evaluation recognition rate of the ML grammar.

Compared to SVM, the recognition rate of ours for dynamic motions moving towards x -axis and y -axis was higher in the following motions: “Jump” (J), “Sit” (S), “Roll” (R), “Kick” (k), “Punch” (p), “Zoom In” (ZI), “Zoom Out” (ZO), “Rotation” (RO), “Play” (p), “Fast Play” (fp), and “Rewind” (rw). For motions that require having the hands horizontal to the x -axis, ours show a higher recognition rate than SVM in the following motions: “Drift” (D), “Weapon Change” (ch), “Rotation” (RO), and “Play” (p). However, SVM had a higher recognition rate for overall static motions compared to ours in the following motions: “Go” (G), “Left Direction” (LD), and “Right Direction” (RD). For distinguishing the number of fingers in static motions, SVM shows a higher recognition rate when the number of fingers ranged between 1 and 3, while ours had a higher recognition rate when the number of fingers was 0 or 4 to 5. These motions are “Booster” (B), “Function1” (F1), “Function2” (F2), “Stop” (ST), and “Pause” (PA).

Figure 4(b) shows the grammar recognition rate of AML, which is a combination of ML and Adverb. For AML that consists only of static motions, SVM had an overall higher recognition rate than ours in the following motions: “Left Direction + Go” (LD + G) and “Right Direction + Go” (RD + G). In contrast, for AML that consists of a combination of static and dynamic motions, ours had a higher recognition rate than SVM in the following motions: “Left Direction + Jump” (LD + J), “Right Direction + Jump” (RD + J), “Left Direction + Roll” (LD + R), “Right Direction + Roll” (RD + R), “Left Direction + Sit” (LD + S), and “Right Direction + Sit” (RD + S).

Figure 4(c) shows the grammar recognition rate of CML, which is a combination of ML and ML. For CML that consists of a static motion of the left hand and a dynamic motion of the right hand, ours had a higher recognition rate than SVM in the following motions: “Go + Weapon Change” (G + ch), “Go + Kick” (G + k), and “Go + Punch” (G + p).

Figure 4(d) shows the grammar recognition rate of ACML, which is a combination of ML and ML and Adverb. For ACML, which consists of three motions, the recognition rates of ours and SVM were similar.

The results of ours and SVM show that the recognition rate changes depending on various factors that include the following: static motion that distinguishes the number of fingers and dynamic movement that moves towards a specific direction and a combination of motions. The last factor comprises the combination of two motions, namely, static motion + static motion, static motion + dynamic motion, and dynamic motion + dynamic motion. When additional static or dynamic motions were added to these combinations, a combination of three motions was made. Overall, the results show that ours

TABLE 5: Comparison of motion recognition rate.

ML	Ours (%)	SVM (%)
Go (G)	100	100
Right Direction (RD)	80	100
Roll (R)	80	70
Weapon Change (ch)	90	85
Function1 (F1)	85	95
Booster (B)	90	80
Rotation (RO)	80	70
Rewind (rw)	85	80
Stop (ST)	90	80
Jump (J)	90	70
Shot (sh)	80	85
Kick (k)	90	70
Function2 (F2)	95	100
Zoom In (ZI)	85	80
Play (p)	80	70
Pause (PA)	90	80
Left Direction (LD)	80	100
Sit (S)	90	70
Reload (r)	100	100
Punch (p)	80	70
Drift (D)	85	75
Zoom Out (ZO)	85	80
Fast Play (fp)	85	80
—	—	—

had a higher recognition rate for diverse factors compared to SVM.

4.2. Content Application Test. The defined grammar was applied to the Amusement Content to carry out the test. Table 6 explains the applicable grammar comprising ML, AML, CML, and ACML. Here, ML comprises “Go” and “Stop”; AML has “Go” + “Left Direction” and “Go” + “Right Direction,” CML has “Go” + “Shot” and “Go” + “Reload,” and ACML has “Go” + “Right Direction” + “Shot,” “Go” + “Right Direction” + “Reload,” “Go” + “Left Direction” + “Shot,” and “Go” + “Left Direction” + “Reload.” The average content frame environment was 16 ms (60 fps). This was the average of the frame changes when executing the contents, where frames have been expressed with GUI. Using the Unity Profiler, the content execution was optimized to apply Leap Motion to the Amusement Content. The results showed no significant difficulties in using the Leap Motion as a substitute to keyboards, and interactive execution was possible.

4.3. User Study. To verify the research results qualitatively, the research carried out a survey on 104 people. The subjects of the survey were given comprehensive explanations of the needs of the SMG and its defined concept and were shown a simulation video of the research results. The participants

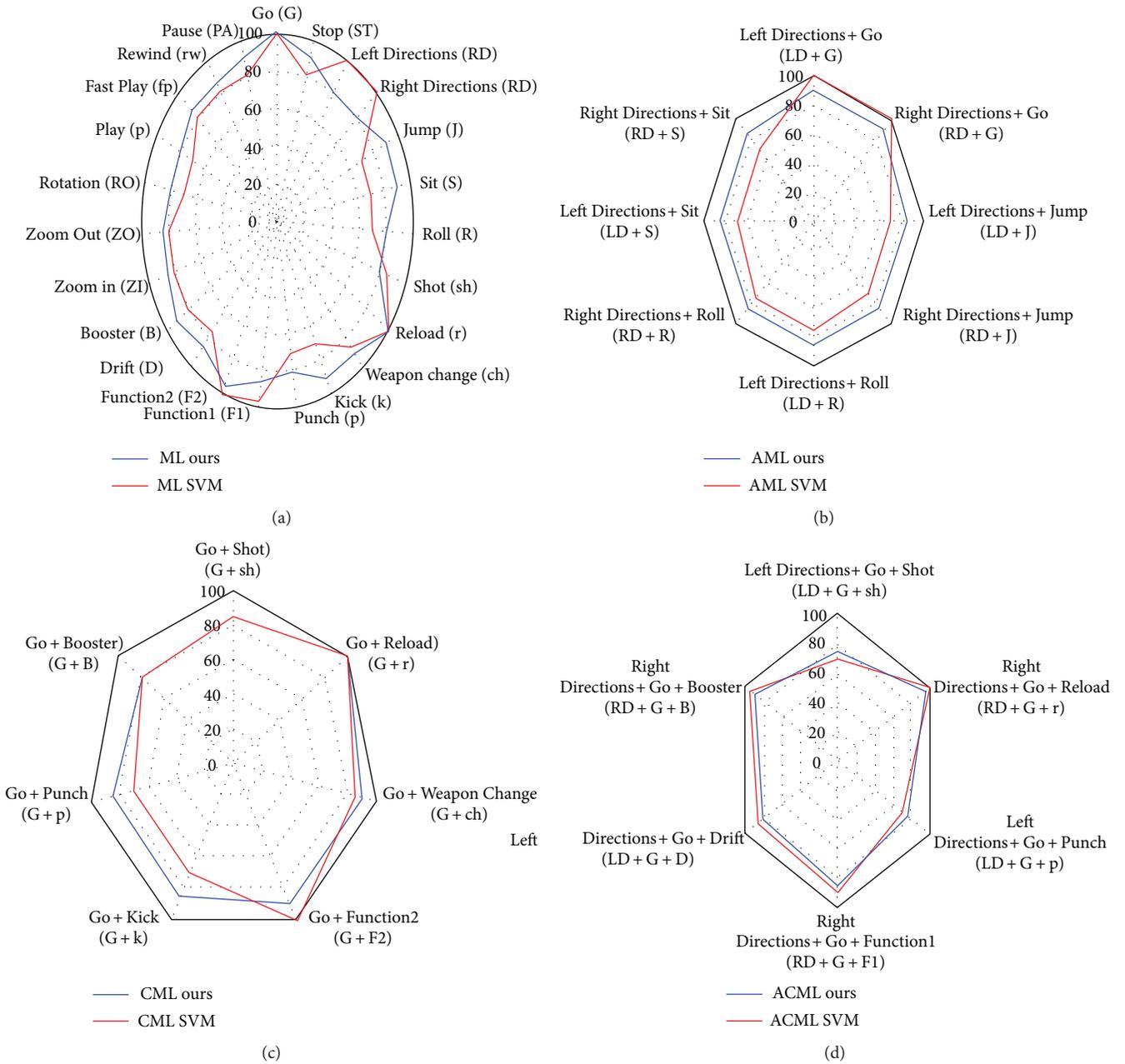


FIGURE 4: Comparison of motion recognition rate between ours and SVM [28].

of the test were between the age groups of 20 and 30 and had prior knowledge and experience on games and Leap Motion.

Google Survey was used to receive more objective responses for the survey by granting subjects with access convenience and sufficient amount of time. The questionnaires and simulation videos were uploaded on the Google program. The questionnaire comprises four questions, and the detailed contents are shown in Table 7. Each part was based on the Likert scale, ranging from one to five points. The left image of Figure 5 shows the captured images of the simulation video that include the game contents and contents of the research development. The right image of Figure 5 shows the results of the user evaluation with the following

responses. “(Q1) Were the contents appropriately classified according to genre?” received 4.62 points. “(Q2) Are the class structures of the defined language appropriate in terms of linguistic view?” received 4.21 points. “(Q3) Can the defined motion language be used for the contents?” received 4.51. “(Q4) Are the motions defined in the Clay Art content useful?” received 4.6 points. We interpreted these scores considering that the users had an appropriate evaluation of the research results.

5. Conclusion

This study defined the SMG that can be applied to the universal content environment of NUI Leap Motion, moving

TABLE 6: Application of defined actions to content.

Grammar	Image		Content environment
ML			
AML			
CML			
ACML			

TABLE 7: User assessment questions.

Questions	Question contents
Q1	Were the contents appropriately classified according to genre?
Q2	Are the class structures of the defined language appropriate in terms of linguistics?
Q3	Can the defined motion language be used for the contents?
Q4	Are the motions defined in Clay Art useful?

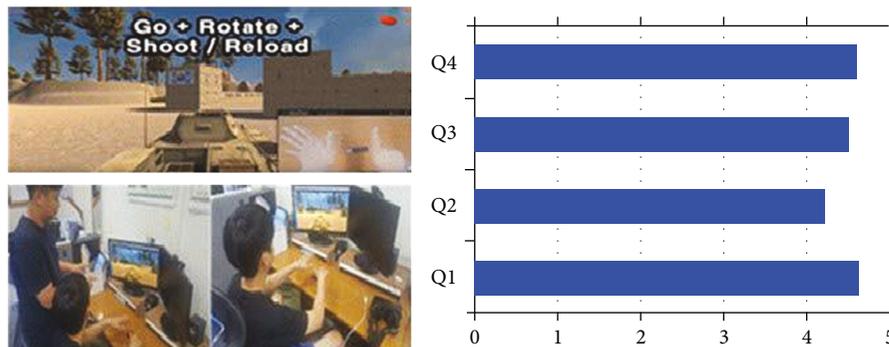


FIGURE 5: Demonstration of the Amusement Content.

beyond the conventional content interface environment. Owing to the variation of the defined motions among contents in the content market environment, the contents were classified and the SMG that can be applied universally has been defined. The contents were classified into Amusement and Functional contents.

These two types of contents were classified into the subcategories: Action, FPS, Adventure, and Racing/Aviation for Amusement Content and Experience and Creation, as well as Teaching and Learning, for Function Content. The

representative motions that were commonly used in the classified contents were investigated, and ML was defined using Leap Motion API. For Action, FPS, Adventure, and Racing/Aviation, the motions were distinguished into right and left hands and were defined. For Experience and Creation and Teaching and Learning, the motions that users can comfortably use have been defined. The motions that have been distinguished into right and left hands have been combined into three types of grammar, while a single ML was also allowed to be a grammar item by itself. The SMG

was completed by applying the four types of grammar to all content motions.

Comparisons with a conventional mouse, a keyboard, and other traditional interaction methods are considered to be of sufficient value. It is also necessary to analyse the time required to learn how to interact. This series of experiments should be added as a future study. Further studies that build a database of more comprehensive gestures will be considered for future works as well.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

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

References

- [1] L. Sousa, J. Monteiro, P. J. S. Cardoso, and J. M. F. Rodrigues, "Natural user interface based on gestures recognition using leap motion sensor," *Dos Algarves: A Multidisciplinary e-Journal*, vol. 26, no. 1, pp. 107–121, 2015.
- [2] January 2016, <https://www.leapmotion.com/>.
- [3] C. Lee, W. Lee, S. Kim, and S. Chin, "Enhanced visual appearance, punch-style weight and physical characteristics based leap motion game," *International Journal of Computational Vision and Robotics*, vol. 6, no. 3, pp. 195–204, 2016.
- [4] L. E. Potter, J. Araullo, and J. Carter, "The leap motion controller: a view on sign language," in *Proceedings of the 25th Australian Computer-Human Interaction Conference on Augmentation, Application, Innovation, Collaboration - OzCHI '13*, pp. 175–178, Adelaide, Australia, 2013.
- [5] E. S. Silva, J. A. O. de Abreu, J. H. P. de Almeida, V. Teichrieb, and G. L. Ramalho, "A preliminary evaluation of the leap motion sensor as controller of new digital musical instruments," in *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration OzCHI'13*, pp. 295–304, Adelaide, Australia, 2013.
- [6] M. M. Davis, J. L. Gabbard, D. A. Bowman, and D. Gracanic, "Depth-based 3D gesture multi-level radial menu for virtual object manipulation," in *2016 IEEE Virtual Reality (VR)*, pp. 169–170, Greenville, SC, USA, 2016.
- [7] M. Iosa, G. Morone, A. Fusco et al., "Leap motion controlled videogame-based therapy for rehabilitation of elderly patients with subacute stroke: a feasibility pilot study," *Topics in Stroke Rehabilitation*, vol. 22, no. 4, pp. 306–316, 2015.
- [8] N. Bianchi-Berthouze, W. W. Kim, and D. Patel, "Does body movement engage you more in digital game play? And why?," in *ACII 2007: Affective Computing and Intelligent Interaction*, pp. 102–113, Springer, Berlin, Heidelberg, 2007.
- [9] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with jointly calibrated leap motion and depth sensor," *Multimedia Tools and Applications*, vol. 75, no. 22, pp. 14991–15015, 2016.
- [10] P. Suryanarayan, A. Subramanian, and D. Mandalapu, "Dynamic hand pose recognition using depth data," in *2010 20th International Conference on Pattern Recognition*, pp. 3105–3108, Istanbul, Turkey, 2010.
- [11] K. Y. Fok, C. T. Cheng, and N. Ganganath, "Live demonstration: a HMM-based real-time sign language recognition system with multiple depth sensors," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, p. 1904, Lisbon, Portugal, 2015.
- [12] P. Kumar, H. Gauba, P. P. Roy, and D. P. Dogra, "Coupled HMM-based multi-sensor data fusion for sign language recognition," *Pattern Recognition Letters*, vol. 86, pp. 1–8, 2017.
- [13] B. Lee, K. Park, S. Ghan, and S. Chin, "Designing canonical form of finger motion grammar in leapmotion contents," in *Proceedings of the 2016 International Conference on Mechatronics, Control and Automation Engineering*, pp. 51–53, Bangkok, Thailand, 2016.
- [14] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller," *Sensors*, vol. 13, no. 5, pp. 6380–6393, 2013.
- [15] J. Guna, G. Jakus, M. Pogačnik, S. Tomažič, and J. Sodnik, "An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking," *Sensors*, vol. 14, no. 2, pp. 3702–3720, 2014.
- [16] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with leap motion and kinect devices," in *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 1565–1569, Paris, France, 2014.
- [17] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with jointly calibrated leap motion and depth sensor," *Multimedia Tools and Applications*, vol. 72, pp. 14991–15015, 2015.
- [18] S. Vikram, L. Li, and S. Russell, "Handwriting and gestures in the air, recognizing on the fly," *Proceedings of the CHI*, vol. 13, pp. 1179–1184, 2013.
- [19] W. Lu, Z. Tong, and J. Chu, "Dynamic hand gesture recognition with leap motion controller," *IEEE Signal Processing Letters*, vol. 23, no. 9, pp. 1188–1192, 2016.
- [20] A. Boyali, N. Hashimoto, and O. Matsumoto, "Hand posture control of a robotic wheelchair using a leap motion sensor and block sparse representation based classification," in *The Third International Conference on Smart Systems, Devices and Technologies, SMART 2014*, pp. 20–25, Paris, France, 2014.
- [21] M. Seixas, J. Cardoso, and M. T. G. Dias, "One hand or two hands? 2D selection tasks with the leap motion device," in *The Eighth International Conference on Advances in Computer-Human Interactions, ACHI 2015*, pp. 33–38, Lisbon, Portugal, 2015.
- [22] C. H. Chuan, E. Regina, and C. Guardino, "American sign language recognition using leap motion sensor," in *2014 13th International Conference on Machine Learning and Applications*, pp. 541–544, Detroit, MI, USA, 2014.
- [23] F. R. Khan, H. F. Ong, and N. Bahar, "A sign language to text converter using leap motion," *International Journal on Advanced Science Engineering and Information Technology*, vol. 6, no. 6, pp. 1089–1095, 2016.
- [24] M. Mohandes, S. Aliyu, and M. Deriche, "Arabic sign language recognition using the leap motion controller," in *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, pp. 960–965, Istanbul, Turkey, 2014.
- [25] J. C. Coelho and F. J. Verbeek, "Pointing task evaluation of leap motion controller in 3D virtual environment," in *Creating the*

Difference: Proceedings of the ChiSparks 2014 Conference, pp. 78–85, The Hague, Netherlands, 2014.

- [26] J. Sutton, “Air painting with Corel Painter Freestyle and the leap motion controller: a revolutionary new way to paint,” in *ACM SIGGRAPH 2013 Studio Talks*, p. 21, Anaheim, CA, USA, 2013.
- [27] L. Hantrakul and K. Kaczmarek, “Implementations of the leap motion device in sound synthesis and interactive live performance,” in *Proceedings of the 2014 International Workshop on Movement and Computing - MOCO '14*, pp. 142–145, Paris, France, 2014.
- [28] “Ver. 0.31, Copyright by Robert O’leary,” November 2016, <https://github.com/roboleary/LeapTrainer.js>.



Hindawi

Submit your manuscripts at
www.hindawi.com

