Proceedings of the IIEEJ Image Electronics
and Visual Computing Workshop 2012
Kuching, Malaysia, November 21-24, 2012

A DATA-GLOVE-BASED INTERFACE FOR MOTION SELECTION AND STYLE CONTROL

Nik Isrozaidi Nik Ismail[1, 2] and Masaki Oshita[1]

[1]Kyushu Institute of Technology, Iizuka, Fukuoka, Japan
[2]Universiti Teknologi Malaysia, Johor, Malaysia

ABSTRACT

In this paper, we present a data-glove-based interface that controls many types of motion and their styles. We use the combination of the positions of both hands to select a motion style and finger angles to control the motion parameters for motion blending. We define mappings between the hand positions and motions so that the motion directions are related to the hand positions. As an advantage of our interface, by using hand positions, the user can easily guess and remember the mapping even for motions with vertical directions such as jumping. In this paper, we present experimental results of our interface for motion type selection. We compare our data-glove interface with a gamepad interface. The results show our interface is easy to guess the mappings between hand positions and motions especially for the motions with vertical movements when they did not know the mappings. On the other hand, the results show that our interface requires a little extra time for users to handle compared to the gamepad interface once the subjects remember the mappings for both interfaces.

1. INTRODUCTION

Building a motion control interface for virtual human animation is one of the most challenging parts of computer graphics field. There is a demand for users to be able to select many types of motion and also change the style of motion in these applications. Currently, most interactive applications such as three-dimensional computer games employ traditional input devices such as a keyboard, mouse and gamepad.However, traditional input devices can select only a limited number of motion types because they have a small number of degrees of freedom.Moreover, in many current interactive applications only predefined motion can be selected and performed. In some applications, such as fighting games, users wish to change the motion style according to their preferences.

A solution to realize the control of motion style is motion blending. Motion blending generates a resulting motion by blending a set of example motions with given motion parameters [1].Because the motion blending can be only applicable to the same types of example motions, to utilize this technique, set of example motions must be prepared for each type of action. In addition, the system needs an interface with that the user can select a type of motion and motion parameters. For the motion parameters, any numbers of parameters to represent the style of the motion such as speed, direction, target height, etc. can be used. These parameters must be given each example motion either manually or automatically [1].

We proposed a data-glove-based interface that can select many types of motion and control the motion style [2]. The key idea is that we use the combination of the positions of both hands to select a motion type. We define mappings so that the motion directions are related to the hand positions. For example, the crouching motion can be mapped to a lower position of the hands. For style control, we use finger angles to control the motion parameters for motion blending. As an advantage of our interface, by using hand positions, the user can easily guess and remember the mapping even for motions with vertical directions such as jumping. This is an important advantage compared to gamepad or mouse, which only has two dimensional positions or directions and cannot specify vertical directions. In addition, with our interface, it is easy to select a motion type and control the motion parameters separately using different inputs.

In this paper, we present experimental results of our interface for motion type selection. We compare our data-glove interface with gamepad interface. We asked subjects to use both interfaces and measured the number of errors and response time. The results show our interface is easy to guess the mapping between hand positions and motions especially for the motions with vertical movements when they did not know the mappings. On the other hand, the results show that our interface requires a little extra time for user to handle compared to the gamepad interface once the subjects remember the mappings for both interfaces.

The rest of this paper is organized as follows. Section 2 review related work. Section 3 briefly explains our system and interface design [2]. Section 4 presents our experimental results and discussion. Section 5 concludes this paper.

2. RELATED WORK

In this section, we discuss related works from the motion control points of view.

2.1. Gamepad interface

Traditional input devices such as mouse, keyboard, gamepad and joysticks have been used as motion control interface for ages.Many researchers and commercial games developers explored these input devices to find good control of character motions. Many of these interfaces are simply for selecting one of predefined motions. However in general, the more the number of motions increase, the more it becomes difficult for users to remember the combinations of mapping between inputs and motions. Moreover, if they want to control motion styles or full body motions of characters, the motion control interface problem is not easy, becausethe animated character has many more of degrees of freedom compared to the traditional input devices.

Laszio and colleagues [3] have developed an interactive system to control physically simulated planar character using the mouse as input device. They map the mouse movements to join torque or to desire values for a PD servo. Using their interface, the users were able to control the jumping robots. A spatial key framingtechnique has been proposed by Igarashi and colleagues [4]. Using this technique, users set up acorrespondence between a 3D position and the character's posture. The users dragged a 3D markerto create a performance by interpolating among the corresponding character poses.However, these systems did not allow the user to select a motion type. If the users want to perform different type of motions, they would need to change the motion data.

2.2. Gesture-based interface

Recently some game console have a device that captures the user' body movements such as Nintendo Wii (controllers with acceleration sensors) and Microsoft Kinect (an optical motion capture sensor). With such devices, the users can select motion types by using gestures. In addition, it is even possible to control the full body movements of the character directly by moving their body.

There are various researchers that employ such motion capture (gesture-based) interface. Freeman and colleagues [5] used the full body motion of a user acquired with a camera input device to controlcharacters. They used image moments and optical flow to detect the user'smovement and allowing the user's motion to be directly mapped to the character's motion. However,they didn't control each part of the character's body. They just made the character perform predefinedmotions according to the user's gestures, such as leaning or walking in different directions.A sparse marker setworn by the user was built by Chai and Hodgins [6]. They construct a motion graph [7] from prerecordedmotion data, and search for motion segments that are close to the current marker positionsto reconstruct the full-body motion of the user. However, a large space is required to operate thesesystems.

If the players' body movements are directly mapped to the characters' body movements, some space is required to control the character and it is difficult to walk through a virtual world with such interface. Moreover, this way is difficult to make the character perform dynamic motions such as high kick and back flip jump because of the physical constraints. On the other hand, if a gesture-based approach is employed with such motion capture device, it is possible to realize more intuitive mapping compared to our interface, which use only hand positions. However, it makes the gesture recognition more complex and there might be more chances of recognition errors.

2.3. Data-glove-based interface

There are other researchers that employ data glove for controlling characters.

Okada [8] proposed a puppet mechanism for motion generationusing data glove. The user can create the motion of an articulated figure using finger actions as in the control of a puppet. He also introduced the physical constraint of gravity and ground contact to generate natural motion. However, using his mechanism, virtual human hands, feet and neck can only be moved in one direction.

Komura and Lam [9] developed a data glove interface to control a walking motion using joints of the index and middle fingers and the position of a data glove. They proposed a new method to map the user's finger motions to the locomotion of a three-dimensional virtual human using preprocessing data. With this method, they can only control locomotion.

Nik, Ishiguro and Oshita [10] introduced position-based control and a motion-based control interface using data gloves. Using position-based control, the users can control the waist, leg or arm using the corresponding hand positions with different control modes. While using the motion-based control mode, a motion is selected by the hand position. However, they only considered six motions and did not employ motion blending.

3. USER INTERFACE

In this section, we describe our data-glove-based interface including the system structure and discussion on how we determine our data-glove-based interface.

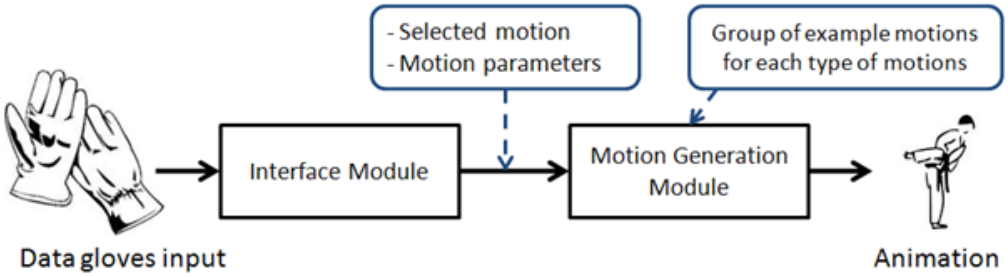

Figure 1: System overview

3.1. System overview

The structure of our system is shown in Figure 1. Our system consists of an interface module and a motiongeneration module. This system can be combined with any interactive applications that require motion style control such as fighting games and virtual communication using avatars.

The interface module interprets input data from a position-tracking device and data gloves worn by the users and sends a selected motion type and style parameters to the motion generation module.

The implementation of our motiongeneration module is based on a global motion blending method [1] with a set of example motions for each type of motion and motion parameters. The necessary numbers of example motions for each type of motion depend on the number of motion parameters. In order to blend them, the motion parameters for each example motion are given manually in advance. The resulting motion is generated by blending example motion according to the blending weight calculated from given motion parameters from the interface module.

3.2. Interface design

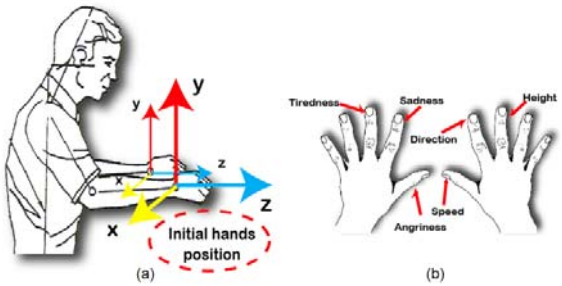

Figure 2: Interface design: (a) motion selection and (b) motion parameters

Figure 2 shows our interface design for motion selection (Fig. 2(a)) and motion-parameter control (Fig. 2(b)). Our interface employs a pair of data-gloves that acquire the angle of 10 fingers and tracking sensors that acquire the position and orientation of both hands. Although our interface should work with any products, in our experiment, we used 5DT as data-glove and Polhemus Fastrak as tracking sensor.

3.2.1. Motion selection

To select a motion type, the user needs to move one or both hands from their initial position in one of 26 directions in three-dimensional space at once as shown in Figure 2(a). Although the hand position can be any point in three-dimensional continuous space, we divide the space into 26 discrete segments based on the directions from the initial positions (any combinations of up, down, forward, backward, left, right and 20 diagonal directions of both hands). A motion type will be selected based the displacement of user's hand position from their initial position.

Table 1 gives an example of the mapping between the hand positions and motion types that we used in our experiment.

Table 1: Examples mappings for data gloves

Motion Types	Hand Positions	
	Right Hand	Left Hand
1. Left punch		forward
2. Right kick	down, forward	
3. Jumping	Up	up
4. Left block		right, up, forward
5. Crouching	down, forward	down, forward
6. Walking	forward	forward

3.2.2. Parameter control

In addition to motion selection, the user can control the motion style by controlling its parameters by bending his/her fingers. Figure 2(b) presents the example of mapping between motion parameters and the user's fingers. Using finger angles, in theory at most 10 motion parameters can be controlled independently. However, practically, it would be difficult to control such many motion parameters. Although we have not experimented with our motion parameter control interface yet, we consider that the appropriate number of motion parameters would be between 2 and 4.

The motion parameters must be set before selecting the motion type. For example, the user can perform a powerful left punch motion by bending their right thumb and left thumb to set the motion speed and angriness parameters and moving their left hand forward to select a punching motion. If the users just move one or both hands from their initial position without setting any motion parameters, a default motion type without style will be generated.

3.3. Discussion

In this subsection, we discuss how we determine our data-glove-based interface. There are other possibleways of using hand positions and finger angles tocontrol several motion parameters and select one motion type at the same time.

The issue that we have to deal with that the data types of the data gloves and the selected motion are different. The handpositions, orientations and finger angles data from the data gloves are continuous data, while the data of the motion type are one-dimensional and discrete. Thus, the continuous data of data gloves should be converted to discrete data of the motion type.

Selecting motion types using hand orientations is less accurate when selecting a large number of motion types compared with using hand positions because hand positions have large control spaces compared to hand orientations.

On the other hand, when using combinations of finger angles to select a motion type, it is difficult to remember the mapping. Therefore, we associated the directions of hand positions with the directions of the motions. We choose to use finger anglesto control the motion parameters because they do not depend on the hand positions.

4. USER EXPERIMENTS AND DISCUSSION

We have developed a simple windows based applicationand conducted user experimentsto evaluate the performance of data-glove-based interface comparing with a gamepad interface.In these experiments, we have used the equivalent mapping for a gamepad and data gloves. Ten undergraduate and postgraduate students participated in our experiments.

In this research, we focused on evaluation of our motion selection interface. Evaluation of our motion parameter control interface is a future work.

4.1. Gamepad interface

We implemented a gamepad interface for comparison. We used common type of gamepad, which have 8 buttons. There are 4 left side buttons, which usually are pushed by the left thumb, and 4 right side buttons, which usually are pushed by the right thumb.

Table 2 gives an example of the mapping between combinations of gamepad buttons and motion types that we used in our experiment. In this experiment, we used a similar way of mapping with our data-glove interface. We assigned motion based on their moving directions except that gamepad buttons only have two-dimensional positions compared to hand positions that three-dimensional positions. For example, to distinguish between walking and jumping, the combination of “up” and “up” buttons are assigned to walking, while the combination of “left” and “right” buttons are assigned to jumping, which may not so intuitive. Although we did not experiment it this time, the analog sticks that a common gamepad has can be used for motion parameter control. If there are two analog sticks, up to 4 motion parameters can be controlled.

Table 2: Example mappings for gamepad

Motion Types	Hand Positions	
	Right Button	Left Button
7. Left punch		up
8. Right kick	up	down
9. Jumping	left	right
10. Left block	right	left
11. Crouching	down	down
12. Walking	up	up

This is kind of unconventional way of mapping between buttons and actions. We chose to use this mapping to compare the data-glove and gamepad interface that are defined with the same approach. In many practical applications, the left side buttons are used to control walking direction, while the right side buttons are used to perform basic motions. The combinations of buttons are used to perform special motions. In addition, a specific button can be defined to represent vertical movements (e.g. making a “jump” button). If we had such a conventional mapping, the result of the experiment may be different. However, we did not experiment such an interface this time. Comparing our interface with other way of mappings of gamepad interface is a future work.

4.2.Experiment procedure

We have designed experiments with two-stages to evaluate the performance of both interfaces. We also divided the participants into two groups and asked them to use the interfaces in different orders to avoid the bias caused by the order. In these experiments, we used six motions. At each stage, the subjects were requested to select a motion typethat randomly shown on the screen six times.

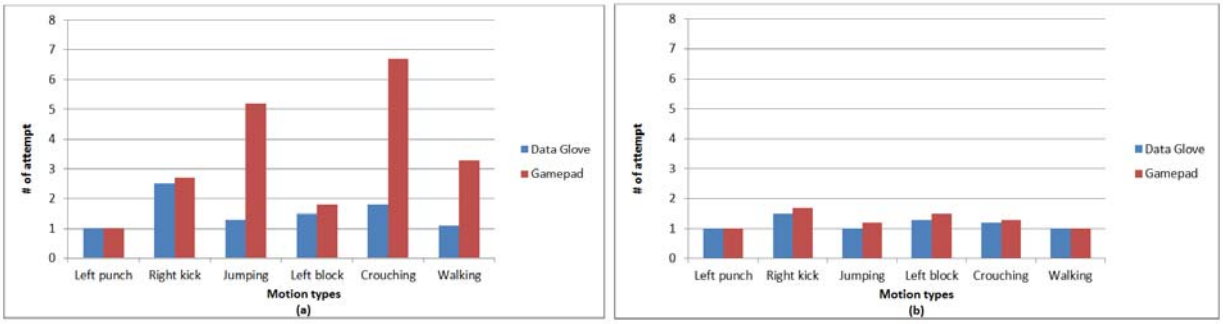

Figure 3: Comparisons of number of attempt: (a) stage one and (b) stage two

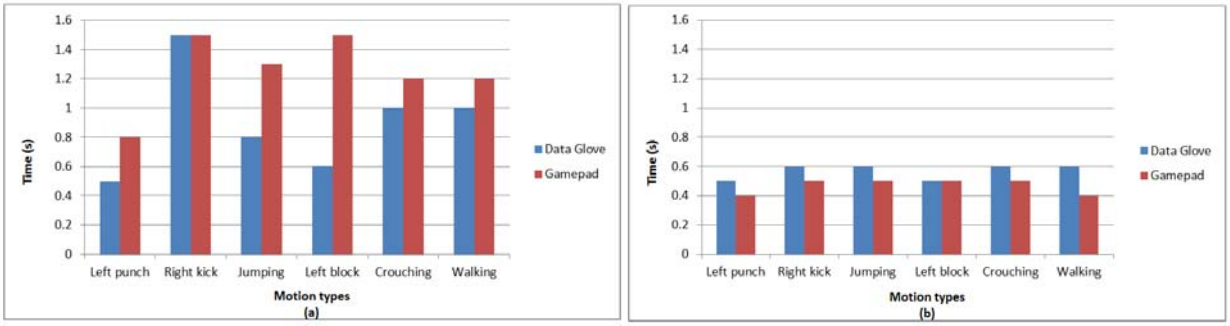

Figure 4: Comparison of time for final attempt: (a) stage one and (b) stage two

In the first stage, the mapping is not shown to them and they need to guess it.This stage is for evaluating each interface whether it is easy for the subject to guess the mapping between inputs and motions for the first time.In the second stage, all the mappingsare shown before the experiment and the subjects are given time to remember it as much as possible. It took about 5 minutes. This stage is for evaluating each interface whether it is easy to handle once they remember the mapping. We record the number of attemptsuntil the subjects select the correct motion. We also record the response time required to select the motion after the instruction is shown on the screen. When the subjects made different motions, we recorded the response time for successful attempt. These scores are recorder for each task of each trial or each subject.

When finished with the first interface, the subjects switched to other interface and did the same things. Finally we summarized the average scores for each stages and each interface.

4.2. Results and discussion

The result of experiments is shown in Table 3 and Table 4.These results lead us to the following discussion.

Figure 3(a) and 3(b)show the comparison of the numbers of attempt between data-glove-based interfaces and gamepad interface in stage one and stage two. In the first stage, the number of attempts is smaller with data-glove-basedfor the motion with vertical directions compared to the gamepad interface.This supports our expectation that our data-glove-interface can provide intuitive mappings for the motions with vertical directions. In the second stage, there is not much difference on the number of attempts between two interfaces, because the subjects remember all mappings and there are only small numbers of errors that they accidentally made.

Table 3: Result for stage one

Motion Types	Interface			
	Data-glove		Gamepad	
	#attempt	Time (s)	#attempt	Time (s)
1. Left punch	1.0	0.70	1.0	0.60
2. Right kick	1.5	1.05	1.7	1.02
3. Jumping	1.0	0.70	1.2	0.60
4. Left block	1.3	0.91	1.5	0.90
5. Crouching	1.2	0.84	1.3	0.80
6. Walking	1.0	0.70	1.0	0.60

Table 4: Result for stage two

Motion Types	Interface			
	Data-glove		Gamepad	
	#attempt	Time (s)	#attempt	Time (s)
1. Left punch	1.0	0.70	1.0	0.80
2. Right kick	2.5	1.75	2.7	1.62
3. Jumping	1.3	0.91	5.2	4.16
4. Left block	1.5	1.75	1.8	1.08
5. Crouching	1.8	1.26	6.7	5.36
6. Walking	1.1	0.77	3.3	2.64

Figure 4(a)and 4(b) show the comparison on the time taken to choose a correct motion types between data-glove-based interface and gamepad interface in stage one. The result shows that theresponse times are faster with data-glove-based.We believe that this reason is same with the results of the number of attempts. However, the gamepad interface is slightly faster than data-glove-based interface after the subjects remember the mappings for both interfaces in the second stage.

In summary, the results show our interface is easy to guess the mappings between inputs and motions especially for the motions with vertical movements when they did not know the mappings. On the other hand, the results show that our interface requires a little extra time for users to handle compared to the gamepad interface once the subjects remember the mappings for both interfaces.

When the number of motions is small, it is not difficult for the users to remember all mappings. In such case, a gamepad interface is better than our data-glove-based interface in term of efficiency. In case that there are many types of motions including one with vertical directions and the users do not have to remember the mappings well, our data-glove-based interface is more preferable than a gamepad interface. We experimented with six types of motions this time. Conducting experiment with a large number of motions to prove the advantage of our interface is a future work.

6. CONCLUSION

In this paper we presented experimental results of our data-glove-based interface for motion type selection. We compared our interface with a gamepad interface in term of number of errors and response time. The result proved the advantage of our interface.

Our future work is conducting experiments to evaluate the combination of motion selection and parameter control. It also includes conducting experiments with different mappings, many motions and various subjects as well as improving our interface.

7. REFERENCES

[1] S.I. Park, H.J. Shin and S.Y. Shin, "On-line Locomotion Generation Based on Motion Blending,"*ACM SIGGRAPH Symposium on Computer Animation*, pp. 113–120, 2002.

[2] N.I. Nik Ismail, and M.Oshita, "Motion Selection and Motion Parameter Control Using Data Gloves," *2011 IEEE International Games Innovation Conference*, California, U.S.A., pp.113-114, 2011.

[3] J. Laszlo, M. Van De Panne, and E. Fiume, "Interactive Control for Physically-based Animation," In *Proc. ACM SIGGRAPH 2000*, 201–208, 2000.

[4]T. Igrashi,T. Moscovich, and J.F. Huges, "Spatial Keyframing for Performance-driven Animation,"*In Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, 107–115, 2005.

[5] W.T. Freeman, D.B. Anderson, P.A. Beardsley, C.N. Dodge, M. Roth, C.D. Weissman, W.S. Yerazunis, H. Kage, K. Kyuma, Y. Miyake, and K. Ichitanaka," Computer Vision for Interactive Computer Graphics," *IEEE Computer Graphics and Applications 18*,3, 42–53, 1998.

[6] J. Chai,and J.K. Hodgins, "Performance Animation from Low-dimensional Control Signals,"*ACM Trans. on Graphics 24*, 3, 686–696, 2005.

[7] L. Kovar, M. Gleicher, and F. Pighin, "Motion Graphs,"*ACM Trans. on Graphics 21*, 3, 473–482, 2002.

[8] Y. Okada, "Real-time Motion Generation of Articulated Figures Using Puppet/Marionette Metaphor for Interactive Animation Systems,"*3rd IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP03)*, pp.13-18, 2003.

[9] T. Komura and C.L. Lam, "Real-time Locomotion Control by Sensing Gloves,"*Computer Animation and Virtual Worlds*, Volume 17, Number 5, pp. 513-525, 2006.

[10] N.I. Nik Ismail, K. Ishiguro and M. Oshita, "Real-time Character Motion Control Using Data Gloves,"*2nd International Conference on Computer Games, Multimedia and Allied Technology (CGAT)*, Singapore, pp. 307-314, 2009.