

# A sketch-based interface to modify and reproduce motion sequences

Hyunho Jang · Jaewoong Jeon · Eisung Sohn ·  
Soon-Bum Lim · Yoon-Chul Choy

Published online: 19 February 2013  
© Springer Science+Business Media New York 2013

**Abstract** We present a sketch-based user interface, which was designed to help novices to create 3D character animations by multi-pass sketching, avoiding the ambiguities usually present in sketch input. Our system also contains sketch-based editing and reproducing tools, which allow paths and motions to be partially updated rather than wholly redrawn; and graphical block interface permits motion sequences to be organized and reconfigured easily. A user evaluation with participants of different skill levels suggest that novices using this sketch interface can produce animations almost as quickly as users who are experienced in 3D animation.

**Keywords** Sketch-based interface · Multiple-pass sketching · Computer animation · Human computer interaction

## 1 Introduction

The current demand for 3D animations is difficult to meet because it takes time to learn how to use current animation authoring tools. Among the varied user interfaces that have been developed for 3D animation, sketch-based techniques come nearest to addressing the problem of accessibility by novices. This type of interface allows a user to draw the path to be taken by a character [2, 10, 12] or the trajectory of a rigid body [17, 19]. These approaches offer relatively accessible interfaces, but do not permit detailed 3D motions to be created without the use of a 3D input device such as a tracker or 3D mouse. Exceptions are Motion Doodles [20] and the system developed by Jeon et al. [8], which do allow a user to create the motion of a character using cursive gestures. Motion Doodles has a one-pass sketching interface, which is simple, but the subset of possible motions it can offer is limited

---

H. Jang · J. Jeon (✉) · E. Sohn · Y.-C. Choy  
Multimedia Graphics Lab—Department of Computer Science, Yonsei University, 134 Shinchon-Dong  
Sudaemoon-ku, Seoul 120-749, Republic of Korea  
e-mail: demiblu@gmail.com

S.-B. Lim  
Department of Multimedia Science, Sookmyung Women's University, Seoul, Republic of Korea

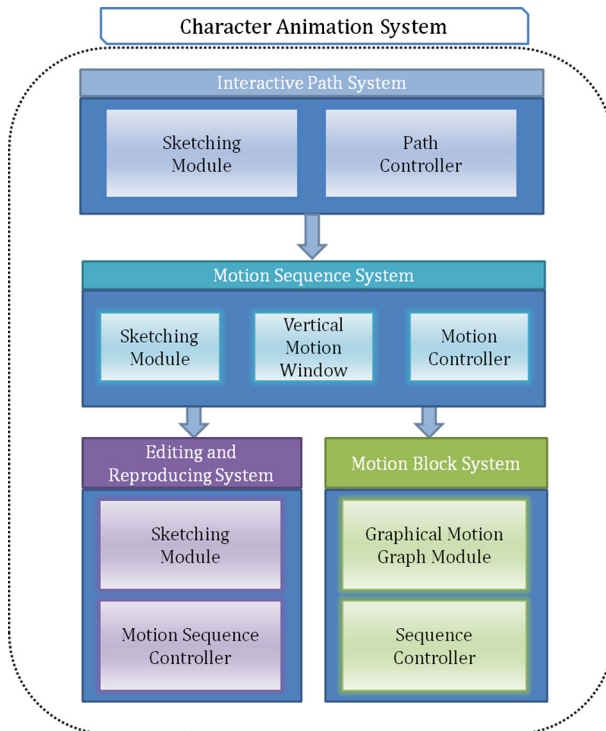
by 3D ambiguities. Jeon et al. [8] avoid these ambiguities through the use of a multiple-pass sketching interface to specify the motion of a character or object. Both systems only offer controls for a single subject and lack the tools to edit motion sequences after they have been created.

Although users can input motion information easily with a sketch-based interface, this naturally commits them to perform the sketching necessary to specify the motions. This can be tedious if the interface required the repeated creation of identical motions, which is likely to occur in creating similar motions sequences for multiple characters.

In previous research, an entire motion sequence has been constructed from a single sketched stroke. But this means that a whole stroke has to be redrawn to make a change, even if the user only wants to change part of it. This suggests the need for a partial editing function, not only for changing the type and properties of the motion, but also for editing its location and length.

We address this need with a new sketch system for authoring motion sequences. Our system use multiple-pass sketching to avoid spatial ambiguities problems; but it offers new editing and reproducing techniques, and a motion block interface, for modifying motion sequences. The system contains an authoring tool for motion sequences, an interactive path editor, an editing and reproducing module for motion sequences, and a motion block interface, as shown in Fig. 1.

The user of this new system firstly selects a character to be animated, and then draws the path for that character to follow. Next, the system presents purpose-specific windows and camera modules which allow the user to specify the character’s detailed motions. A user can



**Fig. 1** Overview of our new sketch-based 3D character animation system

create several motion sequences for different characters. Our system shows the motion sequence for each character as a colored curve on a ground plane, and it can also arrange a number of motion sequences neatly in a synchronization window. Additional sketching interfaces are available to modify the properties of motions and to reproduce motion sequences. The system is composed of authoring system of motion sequences, interactive path system, editing and reproducing system of motion sequences, and motion block interface as shown in Fig. 1.

## 2 Related work

Research on sketch-based interfaces has been going on [18] since the early days of computer graphics, and it has recently gained impetus from advances in tablet computers. SKETCH [22] and Teddy [6] are user interfaces for creating 3D models from users' sketches, and demonstrate that a sketch-based interface can be suitable for novice users. There are the target users for our own system.

Several sketch-based interfaces have already been developed for creating 3D animations. Typically, they enable a user to draw paths for subsequent character animation [1]. Van de Panne [21] indicates character motions as footprints along user-drawn paths. Igarashi et al. [5] maintains that users prefer sketching to other interfaces such as 'driving' or 'flying' for path creation. Our system is different from these because sketching is used to create character motions as well as paths, and both motion and paths can be reused and modified.

There has also been a lot of research on combining techniques for creating 3D animation with sketch-based interfaces [10, 12]. Animations can be created by combining paths created by sketching with motion capture. Popović et al. [17] combine a sketch-based interface with data acquired from a 3D tracker to create a rigid-body animation. Balaguer et al. [2] use a data reduction algorithm to create 3D paths, and Dontcheva et al. [4] provide an acting-based system for creating and modifying character animation. Traditional key-frame techniques have also been adapted for creating animations [7, 19]. McCann and Pollard [15] also proposed an animation controller that synthesizes motion sequences from motion fragments. Our system is not like these systems, either, because ours is based on gestures and supports the editing and reuse of motions.

However, our work is closely related to Motion Doodles [20] and Jeon et al.'s system [8], both of which allow 3D character animations to be created by combining cursive motion gestures with pre-defined character motions. Motion Doodles has a single-pass sketching method, which is simple and intuitive, but it also brings ambiguity in mapping 2D sketch input to 3D motions. This problem is circumvented in Motion Doodles by the introduction of assumptions which limit users to a subset of possible 3D motions. But this makes it difficult for a user to create an animation in which a character moves towards or away from the camera. The multi-pass sketching technique which Jeon et al. proposed in their previous paper [8] overcomes the fundamental drawback of single-pass sketching. But this system, like Motion Doodles, requires a user to draw each motion sequence in a single continuous stroke, and does not allow partial modification of a motion sequence or the reuse of previously created motions. Furthermore, both of these earlier systems still have the long-standing problem of sketch-based interfaces in creating multiple motion sequences. Our new system addresses these problems.

In interactive motion editing techniques, Lee and Shin [13] proposed motion editing system in which a hierarchical curve fitting technique is combined with an inverse kinematics solver to animate human-like figures. Kwon et al. [11] addressed group motion editing by

means of an interactive shape-manipulation technique. Kim et al. [9] also presented interactive motion editing techniques for synchronized multi-character motions. All these systems are largely focused on the creation and editing of natural-looking motions (and then connecting them) or on the interpolation of motion algorithmically. Our system certainly supports interactive motion editing, but it is more concerned with motion sequences and the types and properties of the motions which compose them.

Our system allows users to create and edit character motions by sketching, and Min et al. [16] and Lo and Zwicker [14] have presented similar interfaces for generating motions interactively. However, unlike our system, Min et al.'s system creates character motions by connecting the trajectory of a 2D sketch input by the user with a certain part on a character. Lo and Zwicker focus on a searching algorithm, but also present a sketch-based interface that allows users to modify a motion by sketching the desired trajectory. Our system is different in that it allows users to create character motions by inputting gestures on a plane in 3D. Moreover, other two systems we have just mentioned have the disadvantage of not letting users partially edit a motion because motions are created in a single stroke and the users cannot reuse the motion-specifying strokes and motion sequences entered previously.

There has also been research on techniques for sketching on 3D surfaces. Jeon et al. proposed [8] a 3D sketching interface for 3D animations which combines a vertical motion window and a cross-motion window with a path-dependent vertical plane. Our system provides a similar auxiliary surface for users to modify detailed information about a motion.

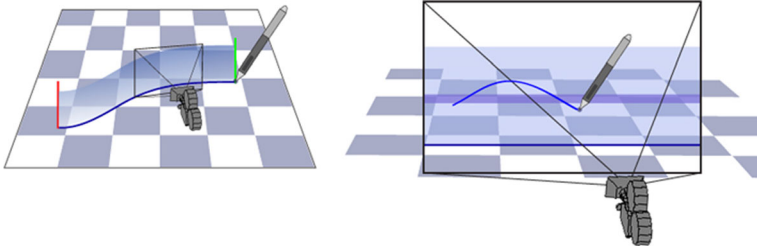
A virtual camera, which provides the user's view, plays an important role in sketching on a 3D surface and creating 3D character animations. Jeon et al. [8] described a camera system which can automatically scroll and zoom in response to the user's input, allowing the user to concentrate on motion sketching. The camera module in our new system is a development of this earlier approach.

### 3 A sketch-based interface for designing a 3D character motion sequence

#### 3.1 Creating a motion sequence

To create an animation, a user of our system first draws a path. The strokes made on the screen are projected on to a ground plane in 3D. They are then sampled and approximated by a uniform cubic B-spline. Then the system erects a window on the path, which provides the user with an appropriate view to sketch the detailed movements of a character, as shown in Fig. 2. Window-specific camera modules provide basic functions such as scrolling and zooming, as well as algorithms to handle defects in the camera motion path such as sudden tangent reversals. This multi-pass approach allows users to create motions, including those directly towards or away from the camera, without encountering spatial ambiguities. Our system assumes that the character moves forwards. To specify detailed movements of a character, the user inputs a polyline, from which individual gestures are extracted. These gestures are parsed into several categories of information, including the type of motion and its parameters, which are assembled into a character motion sequence.

We also provide tools that make it easier to edit and reproduce existing motion sequences. We provide this functionality in our system by analyzing motion sequence sketched by the user, and then drawing each motion on the ground plane. A sequence is segmented into strips, which are colored to correspond to different types of motion, as shown in Fig. 3.



**Fig. 2** The camera module allows users to sketch on surfaces

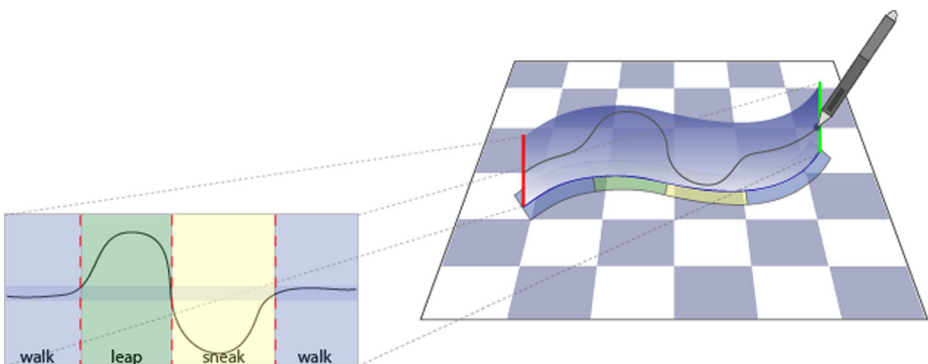
By looking at the differently colored strips, the user can check an animation sequence easily, and then modify it if necessary. Users can modify either the path created previously, or motions on the currently drawn path.

### 3.2 Editing a motion within a motion sequence

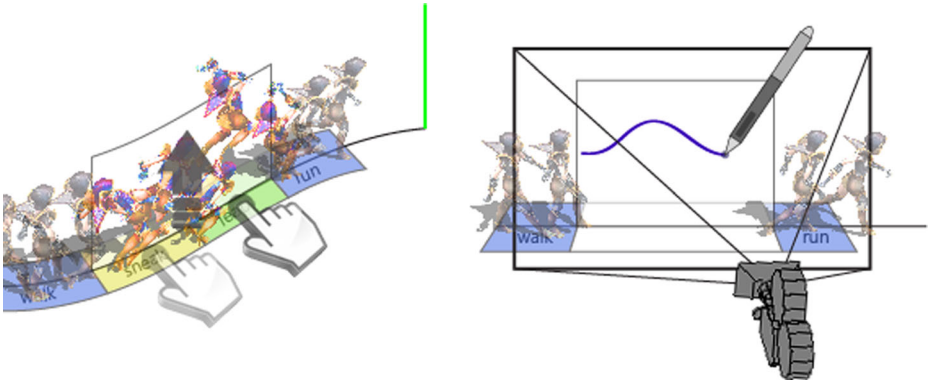
Our system allows a user to modify the motions in a sequence in various ways by sketching; the values of motion attributes (for example, jump height), and the motion type can be changed; and the user can also modify one or more contiguous segments of a motion sequence. When the user chooses a specific motion or group of motions, the system creates a selected editing window (SEW), vertically above the motion strip, so that the user can modify the motion by sketching. To allow the user to input the gestures needed to create a new motion in the SEW, an appropriate camera module and camera path for this task are automatically selected, as shown in Fig. 4.

When a motion is to be modified, the user draws the new motion on the SEW and this new motion replaces the original motion if the distance between the start-point and the end-point of the newly drawn motion exceeds the threshold value.

If this requirement is met, the system can automatically complete the motion gesture using a value derived from the inputted sketch, even if the user's sketch does not finish at the edge of the SEW. A corner-detection algorithm [3] is used to segment the sketched motion. Our framework for mapping the user's strokes to motion gestures is related to the work of Thorne et al. [20] and Jeon et al. [8].



**Fig. 3** A user's sketch is interpreted and segmented into strips corresponding to different types of motion, which are then colored



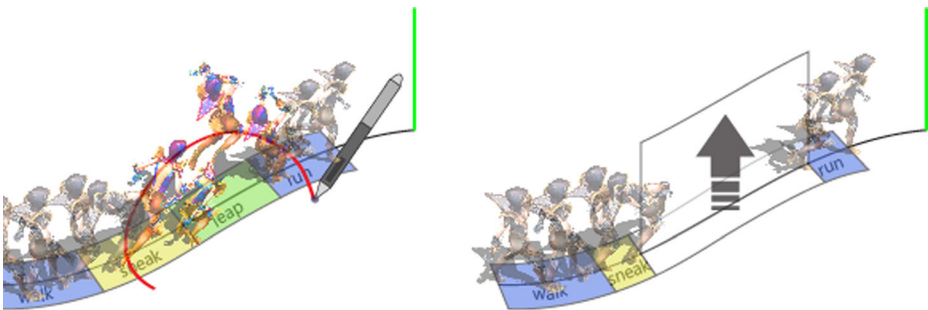
**Fig. 4** (left) The user selects motion strips to edit, and the system erects the selected editing window (SEW); and (right) the user starts to edit the motion by sketching

A modification to a motion sequence does not have to be restricted to a single motion: it is possible to sketch over any part of a motion sequence. Then the motions before and after the range determined by the sketch provide the start- and end-points of the motion, as shown in Fig. 5.

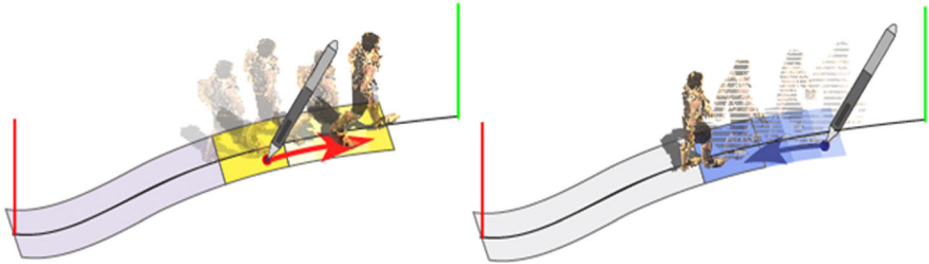
The user is also able to sketch more than one motion in an SEW. The motions before and after the new input are checked by the system. If one of these adjoining motions is truncated and its new length is less than a minimum, it is automatically merged with the motion at the end of the new input. Finally, the motion sequence is updated using interpolation to smooth the joints between the old and new segments of the motion sequence.

If the user wants to modify the length of a motion with minimum effect on its internal details, then the motion length editing gesture can be used. The user clicks on the start or end of a target motion, depending on the direction in which they want to extend or diminish that motion. If the user drags this boundary into another section, the target section and its motion expand; and if the user drags it inwards, the target section and its motion shrink, as shown in Fig. 6. The length of the motion changes in real time as the user sketches.

Many existing sketch-based animation systems oblige users to use long strokes to input the whole path needed for an entire motion sequence. Unlike these systems, ours allows the reuse and modification of motion sequences, making it more suitable for the creation of complicated scenes involving a lot of characters. Motions, paths, and paths with their



**Fig. 5** (left) A user can specify the range of a modification by sketching over any part of a motion sequence; then (right) the system erects an SEW on the selected part of that motion sequence

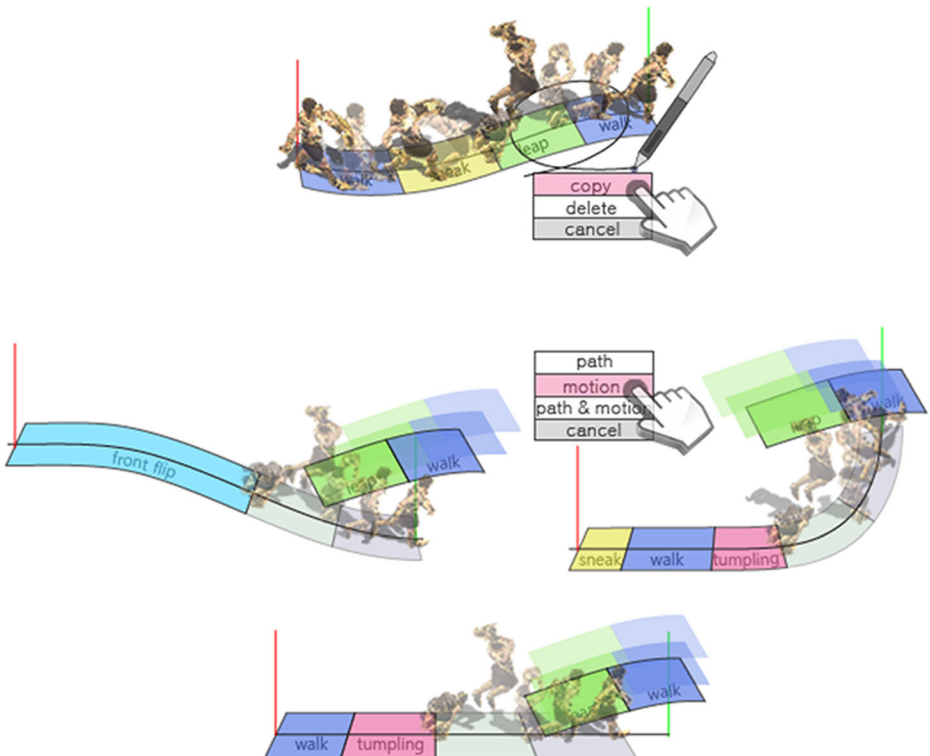


**Fig. 6** A user can adjust the length of a motion using the sketching interface

associated motions can all be saved, wholly or in part, and recalled later. These sequences subsequently can be combined with other motion sequences, as shown in Fig. 7.

### 3.3 Editing the path underlying a motion sequence

Each sketched path is stored as a uniform cubic B-spline, which provides affine invariance and supports local modifications. A user can modify all or part of the path that underlies a motion sequence. To modify the whole path, the user clicks on an icon located at the start-point of each motion sequence, and then draws a new path. Then the system automatically places the original motions on the new path, with new lengths determined by scaling. If the



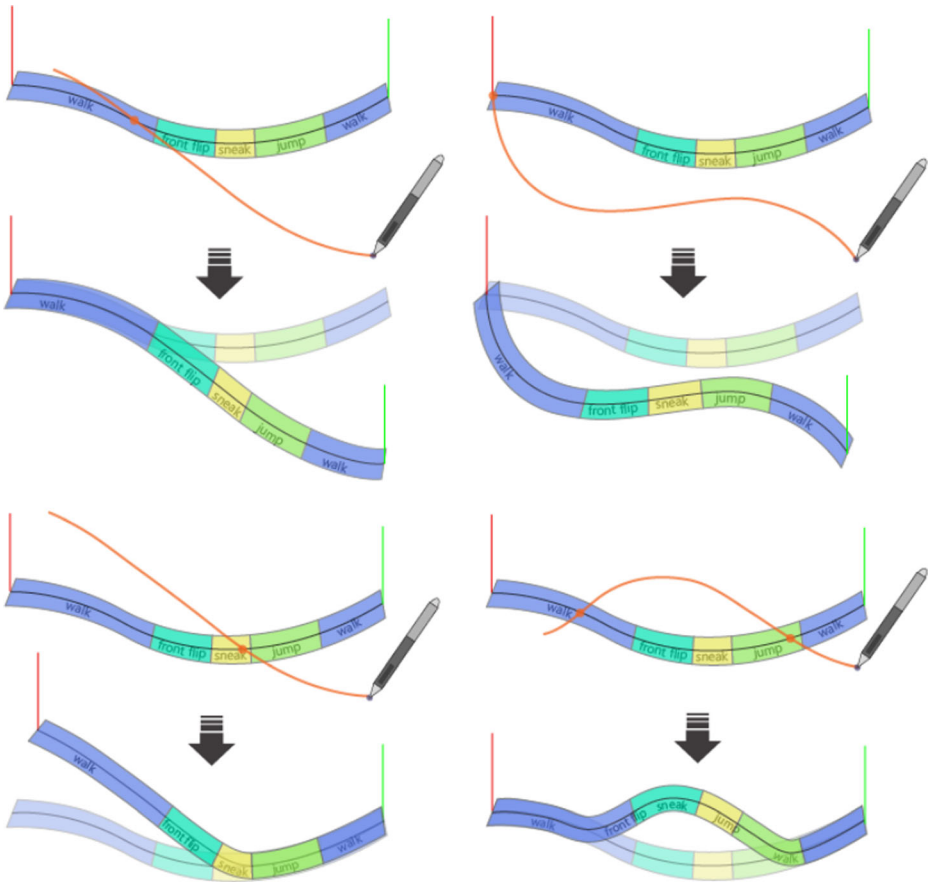
**Fig. 7** (top) A user can copy part of a motion sequence; and (bottom) reuse it repeatedly



user is unsatisfied with the outcome, the length of each motion can be changed, as already described.

The user can also edit part of a path by drawing a new curve. If this crosses the original curve once, the cross-point of the new curve and the original curve divides each curve into two segments which include the start-point and the end-point. Here, the segment of the new curve with the shorter distance is disposed. If the segment of the new curve including the start-point is disposed, the resulting edited path will be composed of the segment of the original curve including the start-point and the segment of the new curve including the end-point. On the other hand, if the segment with the end-point is disposed, the resulting path will be composed of the segment of the new curve including the start-point and the segment of the original curve including the end-point. If the new curve crosses the old curve twice, then its central section replaces the old curve, as shown in Fig. 8. The curve representing the modified path is smoothed by sampling and conversion back into a B-spline.

After the path has been modified, the system fits the original motion to the new path. The length of the new motions are scaled pro rata, but they can subsequently be modified if required.



**Fig. 8** (top left, bottom left, and bottom right) A user can edit part of a path; or (top right) redraw a path while preserving the original motions



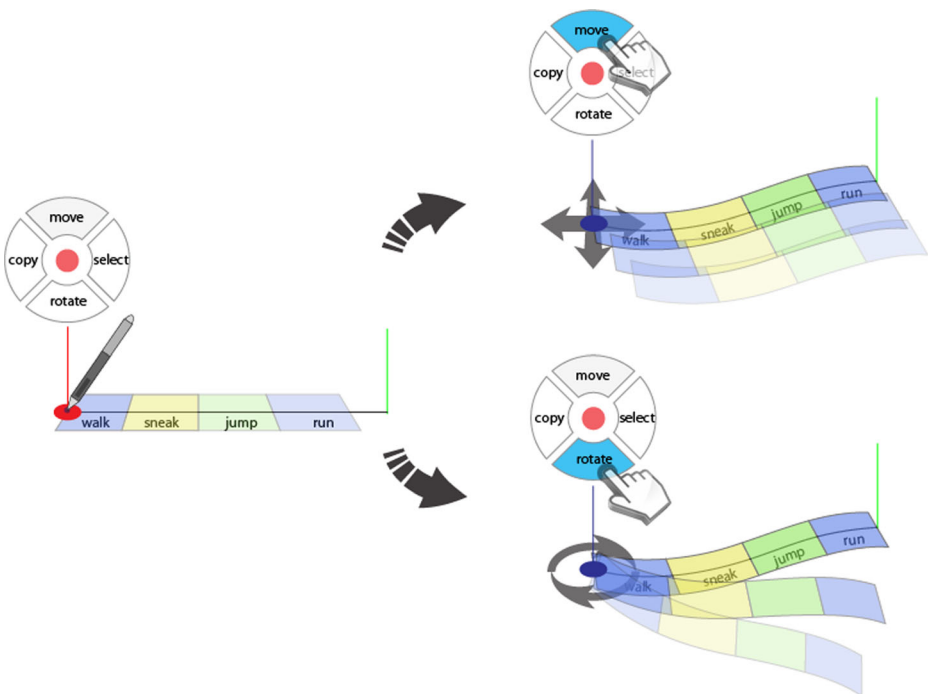
In addition to enabling users to edit the original path, the system also allows them to extend it. If the user clicks on an icon located on the ground plane at the end-point of a path, the system creates an SEW, allowing motions to be added to the new path, in the same way that the original motions were created. Motion and property values can be added to the new path using the camera module.

Figure 9 shows how a path and its motion sequence can be moved to another position on the ground plane or rotated in that plane using either its start- or end-point as the center of rotation.

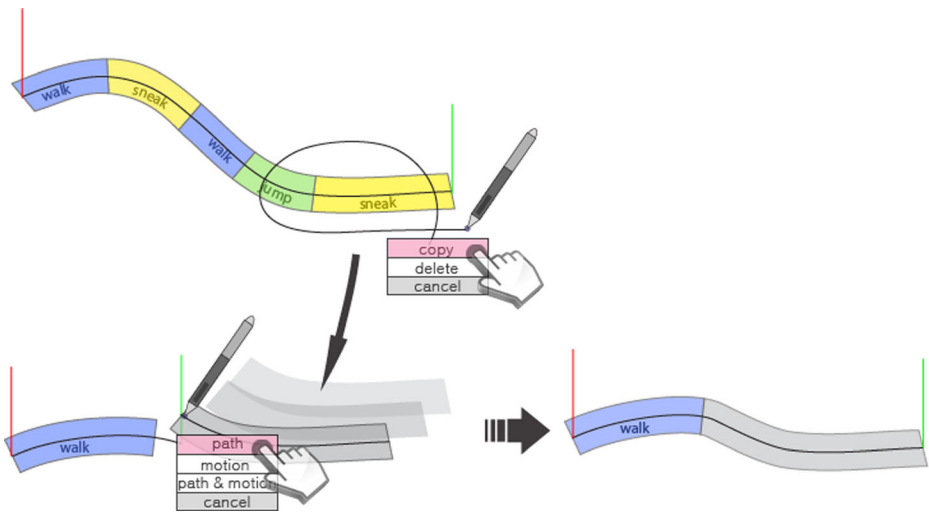
In previous systems, user-created paths could not be reused. Our system provides gesture-based commands to save and reuse paths. A whole path, or part of a path, must first be selected: the path alone, just its associated motion, or both may be chosen. A partial or whole path copied in this way can be inserted at the start, in the middle, or at the end of another path. The system then smooths the modified curve and represents it as a single uniform cubic B-spline, as shown in Fig. 10.

### 3.4 Motion block interface

Creating character motions by sketching has the advantage of allowing users to enter parameter values such as the type, height, and the speed of the motion simultaneously using appropriate gestures. But a single sketched curve cannot provide the amount of information necessary for operations such as deleting part of a motion or altering the sequence in which a series of motions are performed. Even a simple combination of



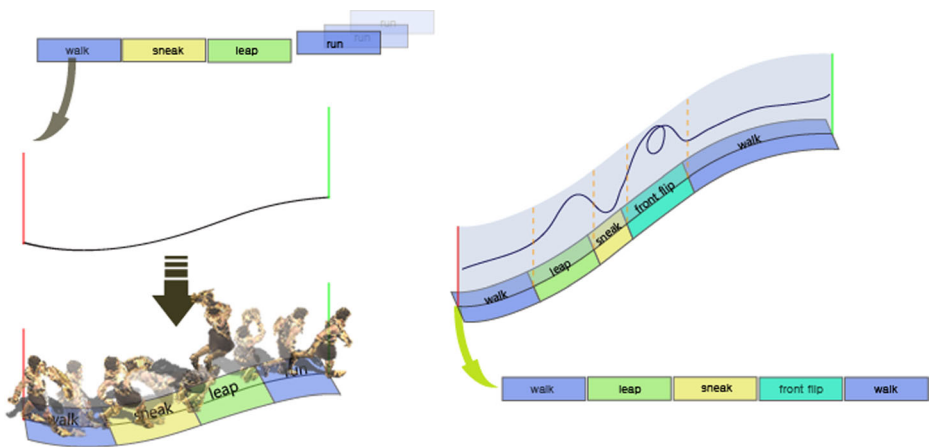
**Fig. 9** A user can move a path and its motion sequence to any position on the ground plane; or (*right*) rotate a path and its motions



**Fig. 10** A user can copy part of a path and paste it into another path

basic motions, without adjusting any specific attributes, will be tiresome if it is necessary to draw each motion individually.

This motivates the introduction of the motion block interface, which supplements sketching in our system. A sequence of motion blocks is displayed as a row of rectangular boxes, each of which is colored and labeled to identify the type of motion that it contains. The user can configure new motion sequences by combining the block in different ways. If motion attributes, such as the height of a jump, are needed during the configuration process, default values will be used. The user can also reconfigure a motion sequence by moving the blocks. The system automatically creates the motion blocks for motions entered by sketching. In this case the attributes of the motions are obtained from the user’s gestures, as shown in Fig. 11.



**Fig. 11** (left) Creating a motion sequence using the motion block interface; and (right) extracting a motion block from a motion sequence

A sequence of motion blocks can be saved as files and reloaded when required. When the user places a series of motion blocks on a different path, the system evenly scales all the motions to the length of the new path. A motion sequence can also be placed on part of a path by selecting that part with a gesture and indicating the block that is to be placed there, as shown in Fig. 12.

The system provides various additional features for configuring motion sequences. The user can make gestures to relocate motions inside a sequence of blocks. They can also move a specific motion within a sequence, swap the locations of two motions, or delete unwanted motions. These tools make it much quicker to create the large number of motions required to fill a long path, and the reuse of the same motion sequence on different paths.

#### 4 Evaluation and discussion

We implemented our system on a notebook PC with an Intel Core i7 processor running at 2.3 GHz, and 8Gb of RAM. Interaction is handled by an auxiliary tablet PC with a Fujitsu T4220 Core 2 Duo 2 GHz processor. Sketching on this tablet PC is performed with a stylus.

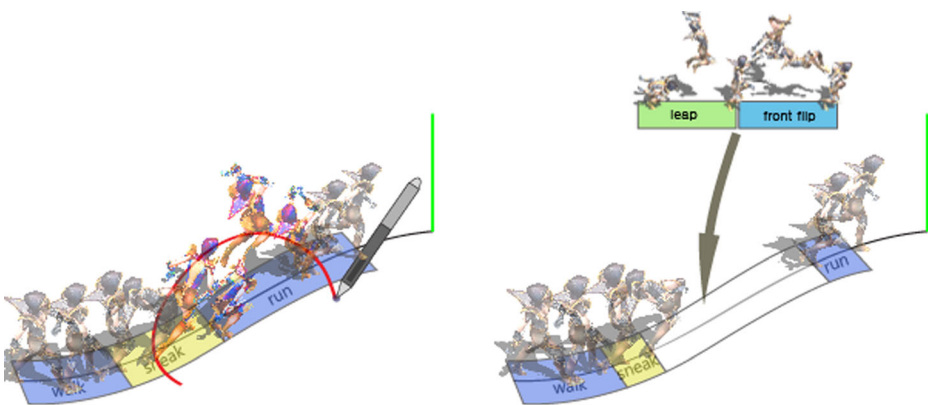
We evaluated the usability of our system on several tasks: creating character animations, and editing and reusing motion sequences and paths with different types of movement. Figure 13 shows an animation created and edited by a novice user.

Some examples of sketching and the resulting animations are also shown in Fig. 14.

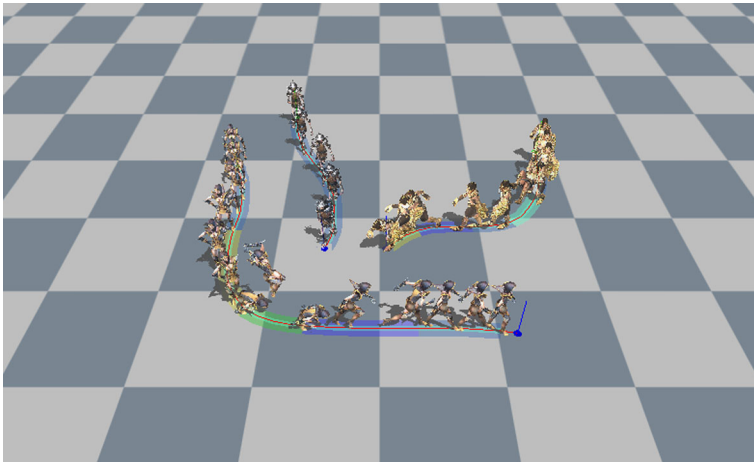
##### 4.1 User study: editing a motion within a motion sequence

In order to assess the usability of our system we recruited some potential users and asked them to edit motion sequences from various viewpoints using our system and existing sketch-based 3D animation systems. To survey user opinions from novice to expert in equal ration, test participants were chosen as follows.

Eight novice users (four female), eight computer science researchers (one female), and eight expert 3D animators (two female) participated in this study. Although the



**Fig. 12** (left) A user can select part of a path by means of a gesture; and then (right) assign motion blocks to the selected location

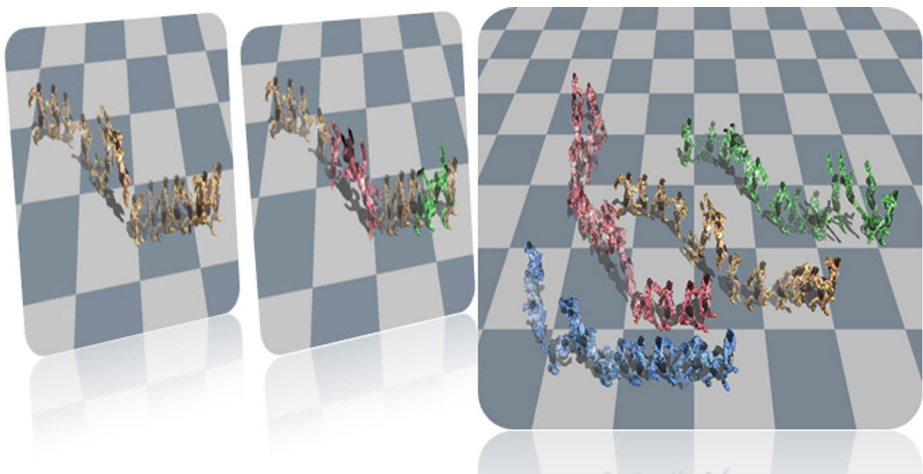


**Fig. 13** Animation created by a novice user of our system

novice users did not have any previous experience in creating and editing 3D character animations, most of them were familiar with a mouse and stylus. Four of the computer science researchers had previous experience with commercial 3D animation tools, while the other four did not.

The eight experts were all accustomed to commercial 3D animation tools and had more than 5 years' experience in related fields. The average age of the novices was  $23\frac{1}{4}$ , ranging from 10 (two school students) to 33. The average age of the computer science researchers was  $28\frac{3}{4}$ , ranging from 26 to 34. The average age of the experts was  $34\frac{1}{2}$  ranging from 32 to 37.

Before the test, we showed short instructional videos explaining Motion Doodles [20] (“System A”), Jeon et al.’s system [8] (“System B”), and our system (“System C”) to the



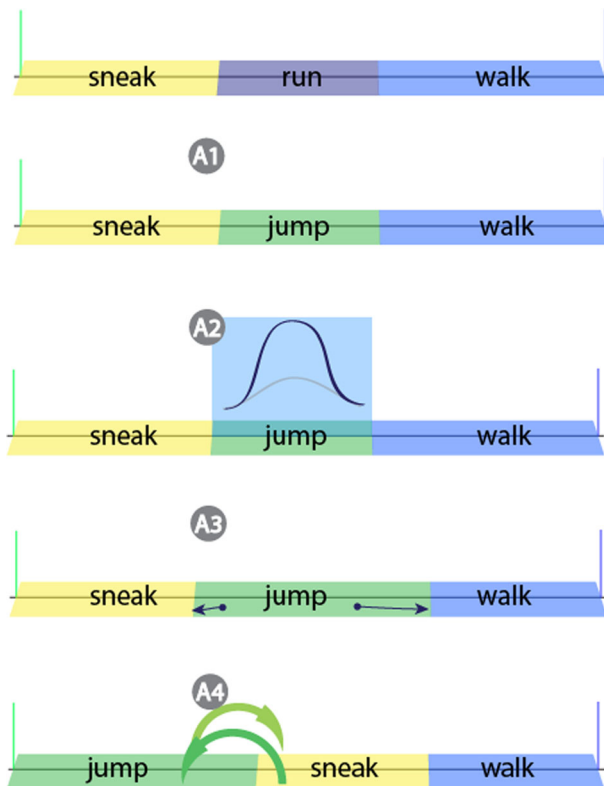
**Fig. 14** (left) sketch of a 3D motion sequence for a character animation which combines walking, jumping, and tiptoeing; (center) an edited 3D motion sequence (consisting of a jump to a front flip and tiptoeing to a front handspring); and (right) various motion sequences created by motion reuse

participants for about 5 min each and then we briefly explained each system. In order to discourage prejudice, the participants were told that all three systems were different versions of our current system.

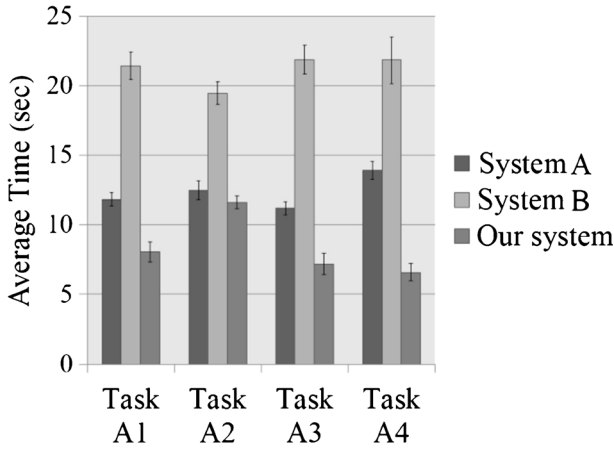
We first showed a motion sequence containing sneaking, running, and walking; then the participants in the first usability test were asked to make four changes to a motion within this sequence shown in Fig. 15 as rapidly as possible.

We recorded the time it took the participants to complete each task and the aggregate results are shown in Fig. 16. Error bars show the 95 % confidence interval. System B was the slowest for all the tasks (statistically significant with  $p < 0.05$ ), and it was also reported that System B tended to produce unwanted changes in motions and paths. Although System A was faster than System B, due to the difference between multi-pass and single pass system, it was still slower than our system for Tasks A1, A3, and A4 (statistically significant with  $p < 0.05$ ). Moreover, System A was not allowed to produce paths with smooth curve and exhibited problems such as unwanted changes in motions and paths. Our system performed best on all the tasks and also preserved those motions that were not meant to be modified.

We also asked the participants the following three questions. Q1: Were you satisfied with the ease of completing the given tasks using each system? Q2: Were



**Fig. 15** User tasks: (A1) change the running motion in a motion sequence to a jump; (A2) edit the height property of a jump motion; (A3) change the length of a jump motion in a motion sequence; and (A4) swap the position of a jump and a sneaking motion



Tasks	System A		System B		Our system	
	Avg. time	SD	Avg. time	SD	Avg. time	SD
Task A1	11.8	1.20	21.4	2.43	8.0	1.83
Task A2	12.5	1.67	19.5	2.10	11.6	1.11
Task A3	11.2	1.15	21.8	2.58	7.2	1.90
Task A4	13.9	1.64	21.9	4.19	6.6	1.51

**Fig. 16** Average time required to edit a motion within a sequence, using each system

you satisfied with the speed of completing the given tasks using each system? Q3: Do you agree that a partial editing feature is necessary in creating motion sequences with sketch-based interfaces? Then we recorded responses on a seven-level Likert scale (“strongly disagree” (0) to “strongly agree” (7)) (Table 1).

These responses suggest that our sketch-based motion sequence editing interface meets expectations in terms of both speed and convenience. Subsequent interviews also indicated that the absence of partial editing feature may lead to inconvenience in motion sequence construction using sketch-based interface.

The participants’ remarks suggested that:

- A sketch-based interface facilitates the creation of 3D animations. But proper editing functions are necessary to avoid having to re-sketch entire motion sequences.
- If a system cannot edit just part of a motion sequence, editing produces unwanted changes.
- Sketching is a comfortable way of modifying the height or length of a motion.

**Table 1** The mean response for each question

Systems questions	System A		System B		Our system	
	Mean response	SD	Mean response	SD	Mean response	SD
Q1	2.25	0.73	3.12	0.74	5.54	0.93
Q2	3.62	0.71	2.25	0.79	5.70	0.95
Q3	5.91(SD=0.71)					

#### 4.2 User study: editing the path underlying a motion sequence

We showed the participants a motion sequence composed of sneaking, running and walking. We asked the participants to perform the five different tasks shown in Fig. 17. The time it took the participants to complete each task and the aggregate results are shown in Fig. 18. Error bars show the 95 % confidence interval.

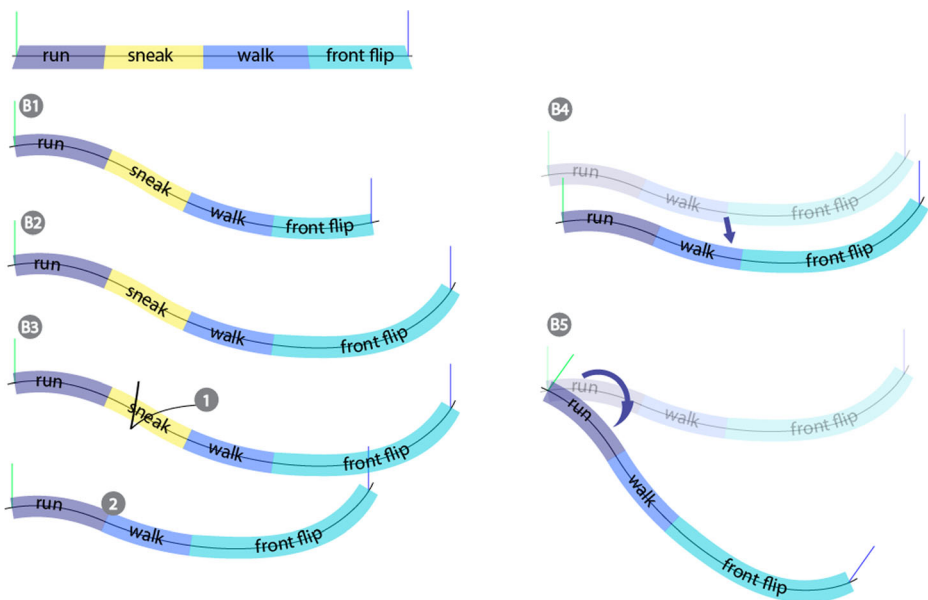
It took longer to complete the task using Systems A or B than it did with our system (statistically significant with  $p < 0.05$ ). We attribute this to the need to re-sketch entire motion sequences when using the other systems. We also asked the users to answer the following three questions. Q1: Could you complete the tasks easily using each system? Q2: Were you satisfied with the speed of working with each system? Q3: Do you see the need for a path editing feature when creating motion sequence with a sketch-based interface? Then we invited responses on a seven-level Likert scale (Table 2).

These responses suggest that our path editing meets expectations in terms of both speed and convenience. Subsequent interviews indicated that a path editing feature is necessary for sketch-based animation system.

The participants' remarks suggested that:

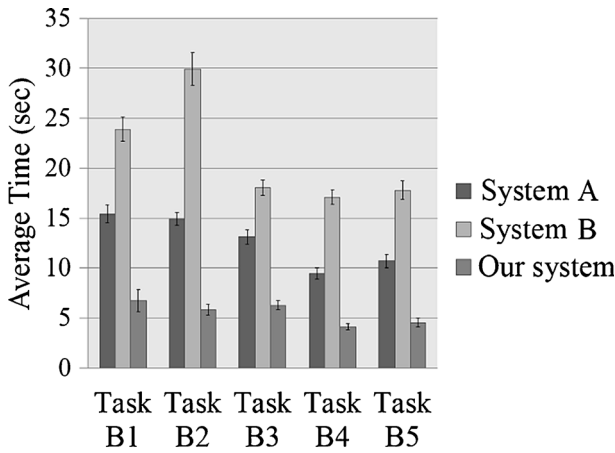
- A sketch-based interface is well suited for creating and modifying a character's path.
- It did not take users long to get used to sketch-based path editing.

Participants, including novice users, quickly adapted to using our system for creating a character's path by sketching, which is an intuitive activity. We believe that these users rated our system highly because they were not only able to input and edit a character's motion by sketching, but also to reallocate motions to new or modified paths. They also noted the value



**Fig. 17** User tasks: (B1) editing a whole path while maintaining its associated motions and their sequence; (B2) increasing the length of a path; (B3) deleting a motion and its path together; (B4) moving a motion sequence and its path to a different location; and (B5) rotating a motion sequence and its path





Tasks	System A		System B		Our system	
	Avg. time	SD	Avg. time	SD	Avg. time	SD
Task B1	15.4	2.15	23.9	2.97	6.75	2.80
Task B2	14.9	1.55	29.9	4.07	5.8	1.37
Task B3	13.1	1.84	18.0	1.97	6.3	1.17
Task B4	9.4	1.36	17.1	1.84	4.1	0.72
Task B5	10.7	1.65	17.8	2.35	4.5	1.12

**Fig. 18** Average time required to perform each path editing task with three systems

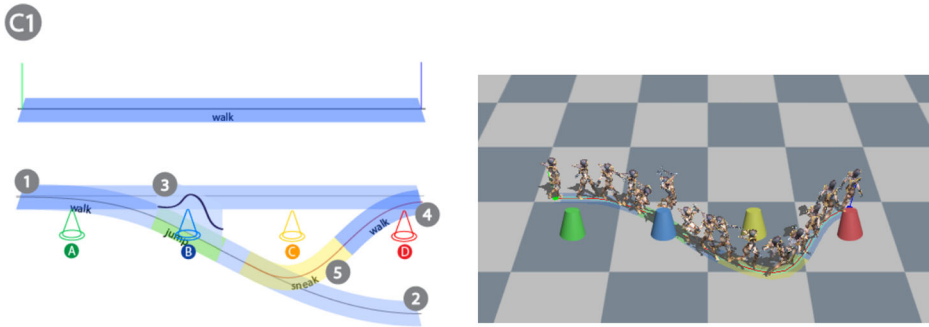
of reusing previous paths by partial editing, rotating and moving, because of the difficulty of redrawing a path accurately.

### 4.3 User study: novices vs experts

In order to see whether our system is easy to use regardless of expertise, we performed another test. We first recruited 32 new participants who weren't involved in the previous tests. They were 16 novice users (nine female) and 16 expert 3D animators (eight female). The novice users did not have any previous experience in creating or editing 3D character animations. Three of them were not even familiar with a mouse and stylus. The 16 experts were all accustomed to commercial 3D animation tools and each had more than 3 years' experience in related fields. The average age of the novices was 26 1/2, ranging from 10 (three school students) to 39. The average age of the experts was 32, ranging from 26 to 36.

**Table 2** The mean response for each question

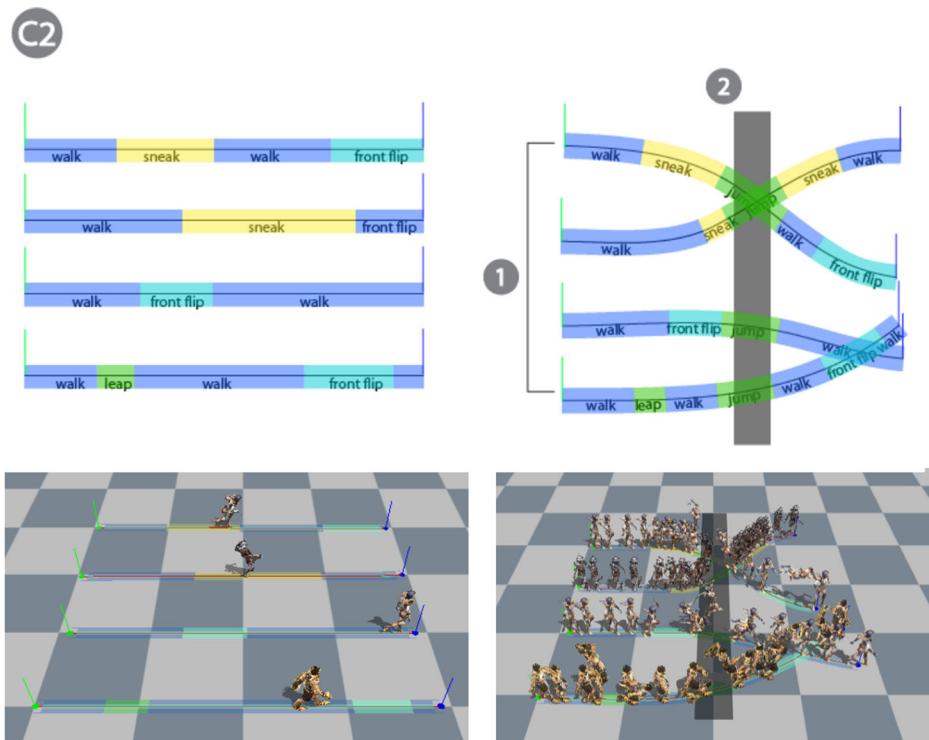
Systems questions	System A		System B		Our system	
	Mean response	SD	Mean response	SD	Mean response	SD
Q1	2.25	0.53	3.16	0.56	5.79	0.72
Q2	2.91	0.71	2.00	0.65	6.00	0.83
Q3	5.83(SD=0.76)					



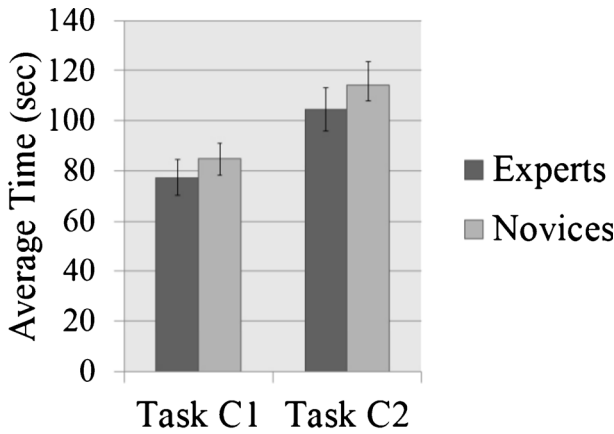
**Fig. 19** (left) Task C1 includes motion sequence translation (1), path editing (2,4), editing of part of a motion sequence (3,5); and (right) an animation created by a participant

Before the evaluation test, we showed instructional videos of our system to the participants for about 5 min each and also explained it briefly.

These new participants were asked to edit a motion sequence and its path as rapidly as possible. We specified two scenarios: For Task C1 we provided a motion sequence consisting of a walking motion, and asked the user to move it to another place (1), to edit its path (2), to edit part of the motion sequence to allow the character to jump over the obstacle B (3), to edit the path to let the character move between obstacle C and D (4), and finally to edit the



**Fig. 20** (above) Task C2 includes editing various motion sequences and their paths; and (below) an animation created by a participant



Tasks \ Systems	Experts		Novices	
	Avg. time	SD	Avg. time	SD
Task C1	77.18	14.54	84.64	13.21
Task C2	104.49	17.93	114.35	19.02

Fig. 21 Average time required by two different user groups to perform two motion editing tasks

motion so that the character performed a sneaking motion while passing obstacle C, as shown in Fig. 19. For Task C2 we provided four different motion sequences. They are composed of four different motions: walking, sneaking, a front flip and a leap.

Users were asked to edit the path of each motion sequence to match an example provided; and then to edit part of each motion so that the four characters each jump over obstacle 2, as shown in Fig. 20. We recorded the time it took the participants to complete each task and the aggregate results are shown in Fig. 21. Error bars show the 95 % confidence intervals.

The results from both tasks showed statistically insignificant ( $p < 0.05$ ) differences between the performance of the novices and the experts, even though children and some users unfamiliar with PCs were very slow.

These results suggest that:

- The time required to perform our tasks was determined by the user’s ability to sketch well, and not their experience or knowledge of 3D animation.
- Novice users can easily use a sketch-based interface and a motion block interface to create and edit character motion sequences.
- 13 of the participants agreed that the proposed system supports easy modification and reuse of 3D character animation.

### 5 Conclusions

We have presented a sketch-based user interface which allows users with various levels of skill to create, edit and reuse 3D motion sequences easily and quickly. Our system combines a sketch-based interface with a motion block interface to increase usability.

We performed a test to see whether our system is efficient. The results suggested that users could quickly learn our interface sufficiently well to edit actual 3D character animations. Our system's support for the editing of partial motions was useful for changing properties of a motion and for editing paths. We also found out that it was easy to reallocate motions using motion blocks. We also found that even novice users could perform tasks at a similar speed to animation experts.

In the near future, we intend to add facilities that will allow users to use sketching interfaces to edit individual frames of a character motion. We also plan to develop an interface which will allow users to create and edit the movements of a stationary character.

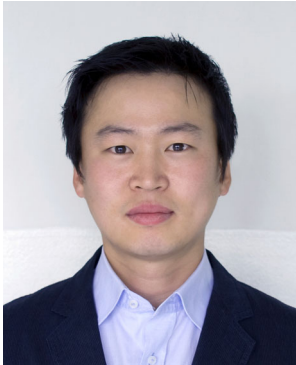
We hope that our system be used in education of animations, and for purposes such as storyboarding.

**Acknowledgments** We would like to thank Minwoo Kim and Hyowoo Kim for valuable suggestions. We appreciate NCORE games Inc. for their 3D models and discussions. This research was supported by Seoul R&BD Program(PA110072) and the National Research Foundation of Korea Grant funded by the Korean Government(Ministry of Education, Science and Technology). [NRF-2011-355-D00059].

## References

1. Arikan O, Forsyth DA (2002) Interactive motion generation from examples. In *Proceedings of ACM SIGGRAPH 2002*, pp. 483–490. doi: [10.1145/566570.566606](https://doi.org/10.1145/566570.566606)
2. Balaguer JF, Gobbetti E (1995) Sketching 3D animations. *Comput Graph Forum* 14(3):241–258. doi:[10.1111/j.1467-8659.1995.cgf143\\_0241.x](https://doi.org/10.1111/j.1467-8659.1995.cgf143_0241.x)
3. Chetverikov D, Szabo Z (2003) A simple and efficient algorithm for detection of high curvature points in planar curves. *Lect Notes Comput Sci* 746–753. doi:[10.1007/978-3-540-45179-2\\_91](https://doi.org/10.1007/978-3-540-45179-2_91)
4. Dontcheva M, Yngve G, Popović Z (2003) Layered acting for character animation. *ACM Trans Graph* 22(3):409–416. doi:[10.1145/882262.882285](https://doi.org/10.1145/882262.882285)
5. Igarashi T, Kadobayashi R, Mase K, Tanaka H (1998) Path drawing for 3D walkthrough. In *Proceedings of ACM UIST 1998*, pp. 173–174. doi: [10.1145/288392.288599](https://doi.org/10.1145/288392.288599)
6. Igarashi T, Matsuoka S, Tanaka H (1999) Teddy: a sketching interface for 3D freeform design. In *Proceedings of ACM SIGGRAPH 1999*, pp. 409–416. doi: [10.1145/311535.311602](https://doi.org/10.1145/311535.311602)
7. Igarashi T, Moscovich T, Hughes JF (2005) Spatial keyframing for performance-driven animation. In *Proceedings of SCA 2005*, pp. 107–115. doi: [10.1145/1073368.1073383](https://doi.org/10.1145/1073368.1073383)
8. Jeon J, Jang H, Lim SB, Choy YC (2010) A sketch interface to empower novices to create 3D animations. *Comput Animat Virtual Worlds* 21:423–432. doi:[10.1002/cav.353](https://doi.org/10.1002/cav.353)
9. Kim M, Hyun K, Kim J, Lee J (2009) Synchronized multi-character motion editing. *ACM Trans Graph* 28(3):1–9. doi:[10.1145/1576246.1531385](https://doi.org/10.1145/1576246.1531385)
10. Kovar L, Gleicher M, Pighin F (2002) Motion graphs. In *Proceedings of ACM SIGGRAPH 2002*, pp. 473–482. doi: [10.1145/566570.566605](https://doi.org/10.1145/566570.566605)
11. Kwon T, Lee KH, Lee J, Takahashi S (2008) Group motion editing. *ACM Trans Graph* 27(3):1–8. doi:[10.1145/1360612.1360679](https://doi.org/10.1145/1360612.1360679)
12. Lee J, Chai J, Reitsma PSA, Hodgins JK, Pollard NS (2002) Interactive control of avatars animated with human motion data. In *Proceedings of ACM SIGGRAPH 2002*, pp. 491–500. doi: [10.1145/566570.566607](https://doi.org/10.1145/566570.566607)
13. Lee J, Shin SY (1999) A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of ACM SIGGRAPH 1999*, pp. 39–48. doi:[10.1145/311535.311539](https://doi.org/10.1145/311535.311539)
14. Lo WY, Zwicker M (2010) Bidirectional search for interactive motion synthesis. *Comput Graph Forum* 29(2):563–573. doi:[10.1111/j.1467-8659.2009.01626.x](https://doi.org/10.1111/j.1467-8659.2009.01626.x)
15. McCann J, Pollard N (2007) Responsive characters from motion fragments. *ACM Trans Graph* 26(3):6. doi:[10.1145/1276377.1276385](https://doi.org/10.1145/1276377.1276385)
16. Min J, Chen YL, Chai J (2009) Interactive generation of human animation with deformable motion models. *ACM Trans Graph* 29(1):1–12. doi:[10.1145/1640443.1640452](https://doi.org/10.1145/1640443.1640452)
17. Popović J, Seitz SM, Erdmann M (2003) Motion sketching for control of rigid-body simulations. *ACM Trans Graph* 22(4):1034–1054. doi:[10.1145/944020.944025](https://doi.org/10.1145/944020.944025)
18. Sutherland IE (1964) Sketchpad: a man-machine graphical communication system. In *Proceedings of the SHARE Design Automation Workshop*, pp. 6.329–6.346. doi: [10.1145/800265.810742](https://doi.org/10.1145/800265.810742)

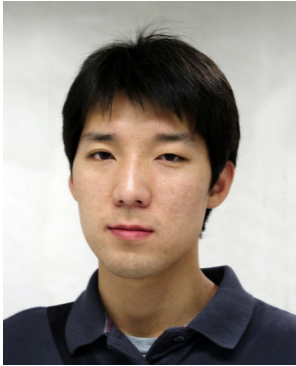
19. Terra SCL, Metoyer RA (2004) Performance timing for keyframe animation. In *Proceedings of SCA 2004*, pp. 253–258. doi: [10.1145/1028523.1028556](https://doi.org/10.1145/1028523.1028556)
20. Thorne M, Burke D, van de Panne M. (2004) Motion Doodles: an interface for sketching character motion. In *Proceedings of ACM SIGGRAPH 2004*, pp. 424–431. doi: [10.1145/1186562.1015740](https://doi.org/10.1145/1186562.1015740)
21. van de Panne M (1997) From footprints to animation. *Comput Graph Forum* 16(4):211–223. doi:[10.1111/1467-8659.00181](https://doi.org/10.1111/1467-8659.00181)
22. Zeleznik RC, Herndon KP, Hughes JF (1996) Sketch: an interface for sketching 3D scenes. In *Proceedings of ACM SIGGRAPH 1996*, pp. 163–170. doi: [10.1145/237170.237238](https://doi.org/10.1145/237170.237238)



**Hyunho Jang** received his MS in Computer Science from Yonsei University, Korea, in 2007. He is currently working towards his PhD in the Department of Computer Science at Yonsei University, Korea. His research interests include sketch-based interface, 3D game, 3D animation and mobile interface.



**Jaewoong Jeon** received his MS and PhD in Computer Science from Yonsei University, Korea, in 2006 and 2010. He is currently working as a researcher at the Software Application Research Center, Yonsei University, Korea. His research interests include sketch-based interface, 3D animation, dynamic storyboard, non-photo realistic rendering, and mobile interface.



**Eisung Sohn** received his PhD in Computer Science at Yonsei University, Korea, in 2012. He is currently working as a researcher at the Software Application Research Center, Yonsei University, Korea. His research interests include interactive animation tools and interactive shape modeling.



**Soon-Bum Lim** received his BS degree from Seoul National University, Korea, in 1982 and his MS and PhD degrees in computer science from KAIST(Korea Advanced Institute of Science and Technology), Korea, in 1983 and 1992, respectively. From 1989 to 1997, he was the engineering director for the font technology and printer division at Human Computer, Inc, and Trigem Computer, Inc. From 1997, he was an assistant professor in the Dept. of Computer Science at Konkuk University in Korea. From 2001, he is currently a professor in the Dept. of Multimedia Science at Sookmyung Women's University in Korea. His main research interests are computer graphics, web and mobile multimedia contents, user interface, and electronic publishing such as font, eBook, and XML documents.



**Yoon-Chul Choy** received his BS degree from Seoul National University in 1973, and MS and PhD degrees from the University of California, Berkeley in 1976 and 1979 respectively. He worked as a senior researcher at Lockheed and Rockwell International from 1979 to 1982. He has been with the Department of Computer Science at Yonsei University, Seoul, Korea, where he is currently a Professor since 1984. He was the president of Korea Multimedia Society and the director of Software Application Research Institute. He was also a Visiting Professor at the University of Massachusetts and Keio University, Japan. His research interests include computer graphics, user interface (such as sketch-based interface) and mobile multimedia.