Motion Control with Strokes

Masaki Oshita
Kyushu Institute of Technology
oshita@ces.kyutech.ac.jp

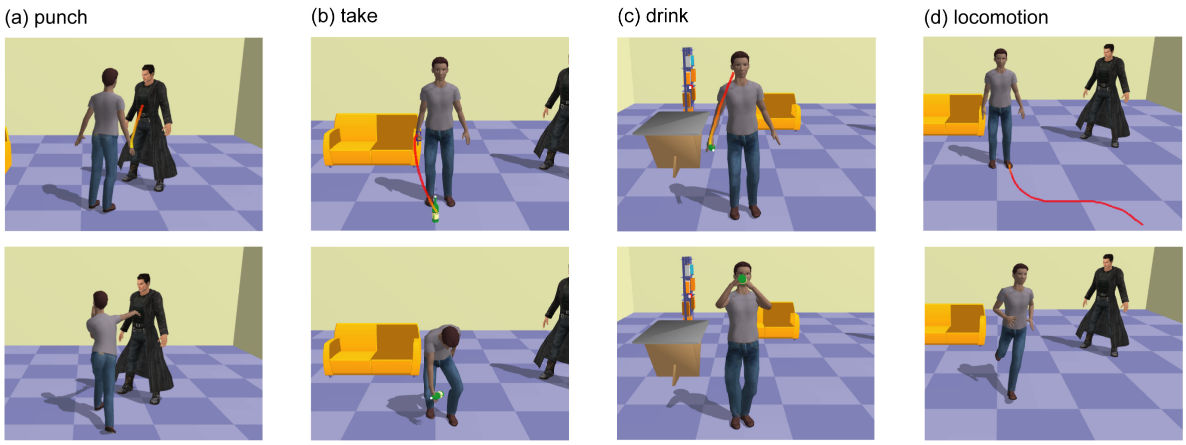

Figure 1: Examples of stroke-based motion control. Input strokes (above) and generated motions (below). (a) A stroke from the hand to the other character. (b) A stroke from the prop to the hand. (c) A stroke from the held prop to the head. (d) A stroke from the character to a target point on the ground.

Abstract

We present a novel interface for interactive and intuitive motion control of human figures. By draw a stroke on the screen using a pen or a mouse, a user of our system can make virtual human figures perform various types of motions. An important feature of “stroke” is that it has initial and terminal points. With a single stroke, users can specify the source and target of an action at the same time. An appropriate action is determined based on single strokes and the context. For example, by drawing a stroke from the foot of a character to a ball, the character kicks the ball. By drawing a stroke from a prop to one of the character’s hands, the character takes the prop with the specified hand. By drawing a stroke from a character to any point on the ground, the character walks to the specified position. First, we categorize selectable objects into some types. The motions are then described using combinations of the defined types. We also detail the implementation of our system and the methods for motion generation, and finally we discuss some issues based on experiments with the implemented system.

Keywords: user interface, motion control, character animation.

1. Introduction

Interactive control of human figures has been a major issue in interactive applications such as computer games. Currently gamepad, mouse or keyboard based interfaces are commonly used. These interfaces are suitable for simply moving a character or making the character perform fixed kinds of actions. However, a user cannot control actions that interact with other objects in many ways. To execute such actions, the users have to choose the desired actions from a hierarchical menu or have to learn combinations of buttons or keys.

In this paper we present a novel interface for interactive and intuitive motion control of human figures utilizing strokes. By drawing a stroke on the screen using a pen or a mouse, a user of our system can make virtual human figures perform various types of actions. An important feature of a “stroke” is that it has an initial and a terminal point. Human actions also have the source and target. Based on this insight, by drawing a single stroke, users can specify the source and target of an action at the same time. An appropriate action is determined based on single strokes and the context (Figure 1). For example, by drawing a stroke from the foot of a character to a ball, the character kicks the ball. By drawing a stroke from a prop to one of the character's hands, the character takes the prop with the specified hand. By drawing a stroke from a character to any point on the ground, the character walks to the specified position. Using this approach, many types of actions can be specified by single strokes. Moreover, the types or styles of actions are optionally controlled based on the intermediate parameters of the strokes such as the drawing speed or trajectory.

In this paper, we describe the principles of our interface design. We first categorize selectable objects into some types. The actions are then described using combinations of the defined types. We also explain the implementation of our system and the methods for motion generation. Finally, we discuss some issues based on experiments. To evaluate the effectiveness of the stroke-based interface, we also have implemented a transitional mouse-and-menu-based interface and compared both the interfaces.

2. Related Work

Many trajectory-based interfaces have been used for locomotion control of virtual figures [1][2][3][4]. In such systems, users can draw a desired locomotion path using a pen or mouse. However, in most of these systems, the user can only control the locomotion paths. Thorne et. al. [5] make a human figure perform various types of walking and jumping by recognizing "gestures" that are drawn along with the trajectory. Oshita [6] uses pen pressure and tilt to control many degrees of freedom (DOF) of a human figure at the same time. However, these interfaces are still limited to locomotion such as walking, running or jump. They cannot handle other kinds of actions that especially involve interactions with other objects or characters in the environments. The existing systems use "trajectory" to specify just a path. In such sense, the initial and terminal points of trajectory simply represent the beginning and ending point of locomotion. On the other hand, our interface use "strokes" to specify the source and target of the action and to generate various types of motions based on two specified subjects.

Igarashi et. al. [7] recently proposed a keyframe animation technique that blends key postures associated with positions in a 3D space, based on the distance between each position and the mouse pointer. By placing appropriate keyframes in a3D space and by drawing a stroke on the 3D space, a smooth animation is synthesized. This method is suitable for controlling the details of an action through the user's performance rather than for making a human figure perform various types of actions.

Some interfaces using input devices that have higher DOF such as a foot pressure sensor pad [8] or silhouette images of a human figure acquired from camera vision [4] have been proposed to control complex motions. The input data from such devices are used to search for an appropriate motion from a database. More commonly, a keyboard-based interface is used in many applications in which the user presses keys associated with the desired actions. These interfaces are good for selecting fixed motions from a database; however, it is difficult to control interactions with other objects, because the target objects or positions cannot be specified through these interfaces. A combination of a keyboard-based and a mouse-based interface is used in some applications, it is difficult for the users to use two devices at the same time and to lean such the interface. Using our stroke-based interface, the user can control both the target positions and the type of motion at the same time.

3. Interface Design

This section describes the principles of our stroke-based interface design. The basic idea of the interface is that we use strokes to specify both the source and target of an action, and that the resulting action is then determined from them. By specifying the source and target of an action at the same time with a single stroke, various types of motions can be specified intuitively.

In the rest of this section, we categorize selectable objects into some types and then describe each type of motion using a combination of the defined types.

3.1. Strokes

A "stroke" is represented by a series of points that are drawn by the user with a pen tablet or a mouse device. In our system, a stroke has both initial and terminal points that are used to specify the source and target of the action. Strokes also have intermediate properties such as drawing speed or drawing trajectory that are

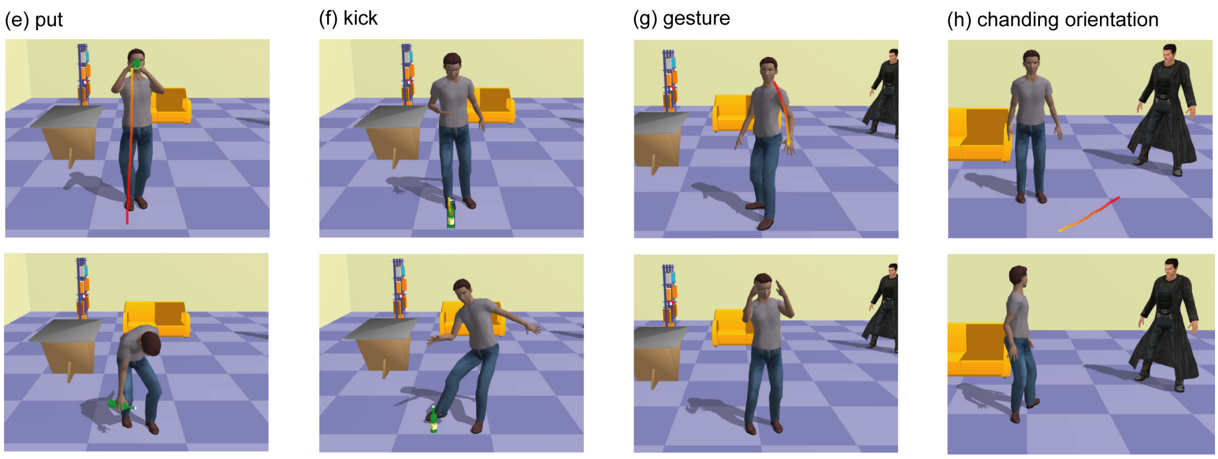

Figure 2: More examples of stroke-based motion control.. Input stroke (above) and generated motions (below). (e) A stroke from the held prop to the ground. (f) A stroke from the foot to the prop. (g) A stroke from the hand to the head. (h) A stroke on the ground for orientation change.

optionally used to change the details of the type or style of the action.

3.2. Target Selection

Selectable objects are categorized into six types.

- A body part of the character that the user wants to control. (BODY)
- A body part of other characters that the controlled character is going to interact with. (SOMEONE)
- A prop that the character is going to handle. (PROP)
- A prop that the character is holding. (HELD PROP)
- A point on the ground. (GROUND)
- A place where the character can move to or put something onto, e.g. a table or chair. (PLACE)

Note that a stroke consists of 2D points in the screen coordinates. We need to compute the selected objects or 3D points in world coordinates. Picking techniques are a common way to do this. By computing the half line that corresponds to the 2D screen point, and by checking the intersection of the half line with all objects, characters, and the ground in the scene, the selected target and the specific 3D positions on the target object are computed. In our stroke-based interface, basically only the targets of the initial and terminal points need to be computed. In the case where a stroke specifies a locomotion path, the intermediate 3D points should also be projected onto the ground.

3.3. Action Determination

The types of actions are determined by single strokes (Figures 1 and 2). We describe some basic interpretations of strokes using terminology as defined above, although other interpretations are also possible depending on the application.

Locomotion. A stroke from BODY to GROUND or PLACE is considered to specify the position to which the user wants to move the character. Therefore, this stroke is interpreted as locomotion. The stroke trajectory is also used as the locomotion path. This may also involve a sitting down or standing up motion in case that a PLACE is specified or the character is on a PLACE now.

Contact motion. A stroke from BODY to PROP or SOMEONE is considered to be an interaction with the target using the specified body part. The type of motion depends on the selected body part and the target. For example, a stroke may mean hitting, manipulating the prop, or simply touching it. The target position of the motion is also controlled based on the selected 3D point.

Taking prop. A stroke from PROP to BODY (hand) is considered to be a taking action.

Using prop. A stroke from HELD PROP to SOMEONE, other PROP or BODY is considered to be using the prop with a purpose; for example, pouring water from a hand held pitcher to a target cup, or using a weapon against other characters. This may also include

source \ target	BODY	SOMEONE	PROP	HELD PROP	GROUND	PLACE
BODY	gesture	contact motion (hitting, manipulate, touch, etc)			locomotion	
SOMEONE	n/a	n/a	n/a	n/a	n/a	n/a
PROP	taking	n/a	n/a	using the prop	moving	
HELD PROP	using the prop (hitting with the prop, applying the prop, etc)				putting the prop	
GROUND	n/a	n/a	n/a	n/a	changing orientation	
PLACE	n/a	n/a	n/a	n/a		

Table 1: Motion determination from the source and target (initial and terminal points) of a stroke.

throwing a prop to a target in the case where the target is far away from the character.

Putting prop. A stroke from HELD PROP to GROUND or PLACE is considered to be that the character is putting it down on the selected target.

Moving prop. A stroke from PROP to GROUND or PLACE is considered to be moving the prop directly.

Gestures. A stroke from BODY to BODY is used to specify some gestures. For example, a stroke from the hand to the other hand indicates handclap as well as a stroke from the hand to the head indicates worrying, scratching head or something.

Changing Orientation. A stroke on the ground (from GROUND to GROUND) is also used to move a character. Compared to the strokes from BODY to GROUND which specify the absolute target positions, this is considered to be the specification of relative movements or orientation. Therefore, we use these for changing orientations of characters without moving their positions.

These interpretations are summarized in Table 1. As can be recognized from the table, BODY, PROP, or HELD PROP become the source (the initial point of the stroke) in those combinations that make sense. This table also shows that the types of motions are changed based on both initial and terminal points and the applicability of the stroke-based approach. Note that the direction of stroke is also important. For example, a stroke from BODY to PROP is interpreted as a contact motion, while a stroke from PROP to BODY is interpreted as a taking motion. Of course, as mentioned before, the interpretations can be modified or new interpretations can be added for unused combinations in Table 1, depending on the applications.

The intermediate properties of a stroke are also used to change the type of motion. For example, a slow stroke from the character's hand to other character's hand may mean a handshake motion, while a quick stroke may mean a punch motion. In this case the drawing speed is used to determine the type of action. The trajectory of the stroke also can be used. For example, if a punch stroke starts moving down and then moving up to the target, it may mean an uppercut instead of a straight punch.

4. Implementation

This work's main contribution is the idea of the stroke-based interface. Here, we introduce some aspects of our implementation and discuss those considered to be useful for anyone attempting to implement a stroke-based interface.

Our stroke-based interface can be used with any motion generation technique. The user can control target positions for the motions with this interface; therefore, motion generation techniques that have the ability to generate a desired motion dynamically based on the given parameters are required. For example, motion graphs [9][3][4] , motion blending [10][11][12][13] or inverse kinematics [14] are suitable for this purpose. Motion trees that are commonly used in computer games are also applicable.

We used motion blending [11] [6] to generate locomotion such as walking, running, turning and stepping since these require dynamic variation based on given parameters. We also used importance based inverse kinematics

[14][15] for other motions that interact with other objects, since there are many types of motions and it was difficult to prepare sufficient example motions for motion blending. In addition to the inverse kinematics, we also implemented an automatic step generation mechanism that executes an approaching step motion before a contact motion in the case that the motion target is out of a reachable range. This kind of technique is very important for users to make the character perform the desired motions without having to worry whether the character reaches the target from a standing position or not. For motion transition, we used a simple posture blending method that blends the joint rotations of each angle, although more sophisticated methods for smooth transitions have been proposed [16][11].

Template based techniques are suitable to determine what action should be executed based on an input stroke. By preparing templates that describe the execution condition for each motion and then comparing them with the condition at runtime, an appropriate motion can be selected. This approach has extensibility since a designer can easily add new templates.

"Smart object" approaches are suitable for object manipulating motions [17][18]. By setting appropriate ways of holding or placing for each object, they are able to be animated with the characters. Object depending motions are also associated with the objects.

5. Experiments

We have implemented some types of actions in our system. Example actions are shown in Figures 1 and 2, and in the accompanying movie. As expected, a variety of kinds of motions can be generated through simply drawing single strokes. We have tried both pen and mouse devices for stroke drawing. As only the initial and terminal points of strokes are important for many of the motions, precise control of the intermediate trajectories are not necessary and so a mouse-based input device works as well as a pen tablet device.

To evaluate the effectiveness of the stroke-based interface, we experimented by comparing a mouse-and-menu-based interface (Figure 3) in which the user first chooses the type of action from a hierarchical menu shown on the lower part of the screen and then chooses the target of the action by clicking the object on the screen. As we designed actions in the menu so that they involve the source or target of an action, e.g. "kick with the right foot", the user only has to choose one target object at most.

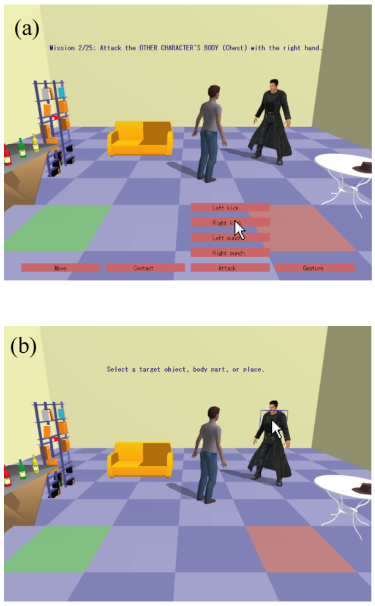

Figure 3: Menu-based interface for comparison. (a) A type of action is chosen from a hierarchical menu, and then (b) a target is chose for the action by clicking the object.

Ten male graduate and undergraduate students participated as the subjects in the experiment. The subjects are requested to control a human figure according to action commands shown randomly one by one on the screen. The input time and the number of input errors for each command are measured and recorded. The experiments are conducted as follows: First, the basic idea of the stroke-based interface is explained to the subjects with a demonstration. So that they can learn the interfaces, they are then instructed to use the system and to perform 20 actions for each interface according to carefully designed and programmed guidance. The subjects are then requested to execute 25 actions for each interface and the results are recorded. Since the order of trying each interface may affect the results, half of the subjects are instructed to use the stroke-based interface first, and then the menu-based interface. The other half is instructed to use them in the opposite order.

The results of the experiments are shown in Figures 4 and 5. As input time is largely affected by the type of the target object, we

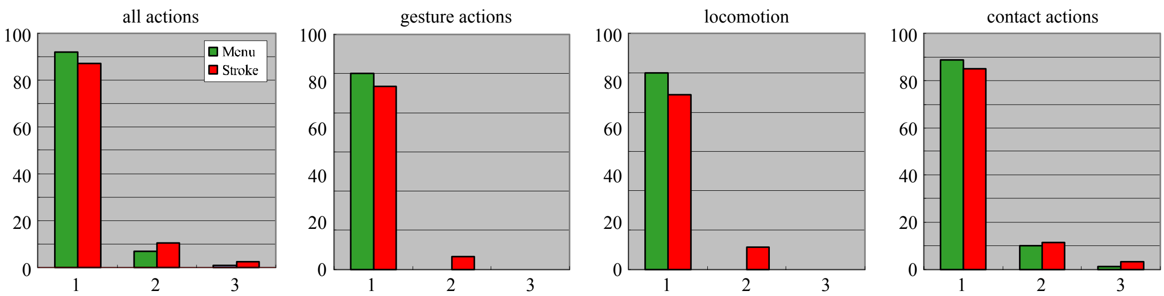

Figure 4: Number of input times in the experiments. The bars show the probability of each number of input times for each interface and each type of actions.

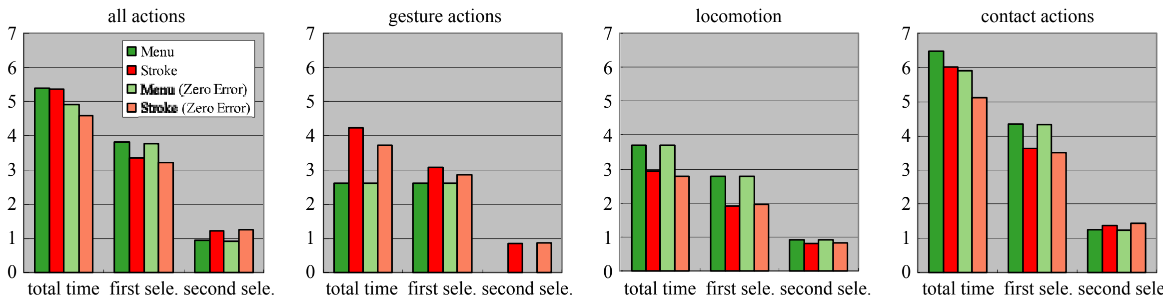

Figure 5: Input time in the experiments. The bars show the input time in seconds for each interface and each type of actions. Partial input times for starting a stroke or selection an action and for drawing the stroke or selecting a target object of the action are also shown.

categorized all actions into 3 types based on the presence of the target selection; gesture actions, contact actions and locomotion. The results are summarized for each type of actions in addition to that for all actions. Gesture actions involve no specific target object specification such as handclap, troubled gesture or drinking from a holding prop. In the menu-based interface they are simply chosen from the menu and then immediately executed, while in the stroke-based interface they are executed by drawing a stroke between two body parts. Contact actions involve specific target selections such as taking, putting or hitting something. In our experiments, locomotion involves loose target selection such as “walk to the red floor tile” which makes the target selection easier compared to contact actions. Locomotion paths are not considered in our experiments, since it is difficult to specify them with the menu-based interface.

Figure 4 shows the probability of the number of input times for each interface. Two or three input times mean that the subjects have missed object selection, drew an inappropriate stroke or chose the wrong action from the menu. More than three inputs are not observed. The results show that the subjects experienced slightly more input errors with the stroke-based compared to the menu-based interface. For contact actions, the users have to select a small object on the screen in both interfaces. As a result, some selection error is caused. For gesture actions, users do not have to choose a target in the menu-based interface; thus, it shows zero input error. On the other hand, users have to draw a stroke from a body part to another, which sometimes causes input errors. The target selection becomes difficult especially when the target object is hidden by other objects or the source and target objects are close to each other. We discuss this matter in the discussion section.

Figure 5 shows the input time. The total time shows the duration from the time when the action command is displayed on the screen and the time when the subject completed the input for the action command. The time for playing the action is not counted. Since the stroke-based interface takes a lot of time when the target is difficult to select, this badly affects the average time; the input time only for action commands with zero error input are also summarized to evaluate the interfaces in cases where the target is easy to select. The first selection time in the graph shows how long the subjects take to start drawing a stroke, or to choose an appropriate action from the menu. The second selection time shows how long the subjects take to finish a stroke, or to select a target object after the action selection in the menu-based interface.

The average input time for all actions are almost the same for both interfaces. However, they have different trends for each type of actions. Since gesture actions do not involve target selections in the menu-base interface as explained above, they take a much shorter time than the stroke-based interface. On the other hand, the stroke-based interface shows faster input time for contact actions and locomotion. There is not much difference between both interfaces for the second selection time The time to star drawing a stroke with the stroke-based interface is much faster compared to the time to action a selection with the menu-based interface, in which the users have to select a combination of a type of action and a body part from the menu, while they are specified with a single stroke in the stroke-based interface.

The results show that the stroke-based interface is effective, especially when an action involves interaction with objects in the environment. From interviews with the subjects after the experiments, they said that they preferred the stroke-based interface even for gesture actions regardless of input time. In the stroke-based interface, the users do not have to switch from the menu to the 3D scene and they can always focus on the 3D scene, which is less stressful. The subjects also pointed the difficulty of target selection, which we expect to improve.

6. Discussion

As shown in the experiments, one issue with our interface is that it is view dependent; that is, it is difficult to select a target that is covered by other objects. Moreover, in cases where the initial and terminal points are very close to each other, it is difficult to control intermediate parameters. To overcome these problems, an adaptive view control scheme would be effective. Such an adaptive view control depends on the targets that the character is supposed to interact with, which is also application dependent; however, this paper does not cover such a view control technique. Automatic camera control in which the camera moves so as to track the controlled character without changing its orientation is used in our implementation. A manual view angle control interface is also provided. It should also be noted that automatic view control during the user drawing a stroke should be avoided as it is very difficult for the users to select a target while the view is changing.

For the same reason, we found that it is difficult to input a stroke from or to a character that is moving. Therefore, the user usually has to wait until the character stops to input the next motion. As well, the stroke-based interface takes more time to initiate an action compared to the gamepad-based approach in which a user can simply press a button to execute an action. For actions that require quick responses, the combination of our interface with a click-based interface is convenient. By clicking only the target of the action and allowing the system to automatically decide the appropriate source of action, the user can specify action very quickly. We have experimentally implemented this click-based interface for attacking motions. By clicking another character, the controlled character performs punches or kicks using an appropriate limb without the user specifying which limb should be used to attack the other. This click-based interface works well with the stroke-based interface.

As our interface provides many kinds of motions, users have to remember them to control many motions: however, since our stroke-based interface is very intuitive, it is not much trouble compared to other interfaces with keyboard or button combinations. Although it may be possible to extend our interface to specify multiple targets with a stroke and to generate more complex motions, we would not encourage this since it would make stroke drawing much difficult. Moreover, most human motions can be specified through just two subjects.

7. Conclusion

In this paper we presented a novel stroke-based motion control interface. The concept is very simple, and it is effective and useful in many applications. Using our interface, many kinds of motions become possible in interactive applications.

Gesture recognition of strokes [5] will make more kinds of motions possible by interpreting the intermediate trajectories of strokes; however, we have not implemented this much yet. By combining our stroke-based approach with gesture recognition, the interface would become more intuitive compared to just a gesture recognition interface, since by specifying the source and target of action with a stroke the type of motion is roughly constrained.

We are planning to develop further applications, and to test our interface using them. We think the interface is useful for those types of actions that involve various props such as sword or gun fighting. User testing especially with non-experienced users or children is also important for future work.

Acknowledgement

This research was partially supported by a Grant-in-Aid for Scientific Research (16650022) from the Ministry of Education, Science, Sports and Culture.

References

[1] H. C. Sun and D. N. Metaxas. Automating Gait Generation. In Proc. of SIGGRAPH 2001, 261-270, 2001.

[2] M. Gleicher. Motion Path Editing. In Proc. of ACM SIGGRAPH Symposium on Interactive 3D Graphics 2001, 195-202, 2001.

[3] L. Kovar, M. Gleicher, and F. Pighin. Motion Graphs. ACM Transactions of Graphics (Proc. of SIGGRAPH 2002), 31, 3, 473-482, 2002.

[4] J. Lee, J. Chai, P. Reistma, J. Hodgins, and N. Pollard. Interactive Control of Avatars Animated with Human Motion Data. ACM Transactions of Graphics (Proc. of SIGGRAPH 2002), 22, 3, 491-500, 2002.

[5] M. Thorne, D. Burke. M. van De Panne. Motion Doodles: An Interface for Sketching Character Motion, ACM Transactions of Graphics (Proc. of SIGGRAPH 2004), 23, 3, 2004.

[6] M. Oshita. Pen-to-mime: A Pen-Based Interface for Interactive Control of a Human Figure. Eurographics Workshop on Sketch-Based Interfaces and Modeling 2004, pp. 43-52, 2004.

[7] Takeo Igarashi, Tomer Moscovich, John F. Hughes. Spatial Keyframing for Performance-driven Animation. ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2005.

[8] K. K. Yin and D. K. Pai. FootSee: an Interactive Animation System. In Proc. of ACM SIGGRAPH / Eurapraphics Symposium on Computer Animation 2003, 329-338, 2003.

[9] J. Lee and K. H. Lee. Precomputing Avatar Behavior from Human Motion Data, Proc. of Eurographics / ACM SIGGRAPH Symposium on Computer Animation 2004.

[10] L. Kovar and M. Gleicher, M. Automated Extraction and Parameterization of Motions in Large Data Sets. ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004), pp. 559-568, 2004.

[11] S. I. Park, H. J. Shin, and S. Y. Shin. On-line Locomotion Generation Based on Motion Blending. Computer Animation and Virtual Worlds, 15, pp. 125-138, 2004.

[12] C. Rose, M. F. Cohen, and B. Bodenheimer. Verbs and Adverbs: Multidimensional Motion Interpolation. IEEE Computer Graphics and Applications, 18, 5, 32-40, 1998.

[13] D. J. Wiley and J. k. Hahn. Interpolation Synthesis of Articulated Figure Motion. IEEE Computer Graphics and Applications, 17, 6, 39-45, 1999.

[14] H. J. Shin, J. Lee, M. Cleicher, and S. Y. Shidn. Computer Puppetry: An Importance-Based Approach, ACM Transactions on Graphics, 20, 2, 67-94, 2001.

[15] R. Bindiganavale and N. Badler. Motion abstraction and mapping with spatial constraints. Workshop on Motion Capture Technology, Geneva, Switzerland, November 1998.

[16] C. Rose, B. Guenter, B. Bodenheimer and M. F. Cohen. Efficient Generation of Motion Transitions using Spacetime Constraints. SIGGRAPH '95 Proceedings, pp.147-154, 1995.

[17] M. Kallmann and D. Thalmann. Smart Objects. Proc. Computer Animation 99, pp. 138-146, 1999.

[18] K. Perlin and A. Goldberg. Improv: A System for Scripting Interactive Actors in Virtual Worlds. In Proc. of SIGGRAPH 1996, 205-216, 1996.