

Effect lines for specifying animation effects

Yoshikazu Kato, Etsuya Shibayama
Department of Mathematical and Computing Sciences
Tokyo Institute of Technology
2-12-1, O-okayama, Meguro, Tokyo 152-8552, Japan.
{yoshika0, etsuya}@is.titech.ac.jp

Shin Takahashi
Department of Computer Science / TARA Center
University of Tsukuba
1-1-1 Tennoudai, Tsukuba, Ibaraki 305-8577, Japan.
shin.takahashi@acm.org

Abstract

When we create 2D animations on a desktop computer using programs such as Microsoft PowerPoint or Macromedia Flash, we assign effects to objects and define their parameters, such as their path, speed, and time of movement. To do this, we use conventional interfaces like menus or dialog boxes. However, the motion effects associated with each object are not displayed on the canvas explicitly, so the user can investigate the properties of each object only by opening a dialog box to see the animations that are associated with the object. Moreover, setting various parameters using menus and dialog boxes is time-consuming because the user must set each parameter individually.

We have therefore developed a method that uses effect lines to set each effect and its parameters. Effect lines are a popular technique that is used in comics and cartoons. They depict information on the effects of an object, such as its speed, length of path, and degree of rotation. They enable us to set effects using simple gestures. In addition, the effect lines can be combined. That is, the effects that each set of effect lines have are merged. This feature makes effect lines flexible for users to create various animations. Effect lines enable the user to recognize animation by displaying the effect lines on each key frame during on-screen editing.

We have developed a prototype animation authoring system. Using this system, the user can create various animations with effect lines. We evaluated the effectiveness of our method as a visual input and pictorial output method, and discuss its use. The user can set most of the "start effects" found in PowerPoint by using effect lines.

Keywords

effect lines, speed lines, key-frame animation, animation effects, animation authoring, visual input method, presentation design

1 Introduction

Much of the authoring software used for making web pages and presentations, such as Microsoft PowerPoint and SWiSH2 [1], has an interface that enables the user to create 2D animations by selecting and applying motion effects that are built into the system. Various parameters for each effect are defined using a dialog box. For example, to animate a box that rotates 360 degrees, the user creates a box on the canvas, selects the "rotation" effect from the menu, and sets its parameters, such as the number of degrees and direction, with a dialog box.

However, there are two problems with such interfaces. First, the motion effects associated with each object are often not displayed on the canvas explicitly, so to investigate the properties of an object the user has to open a dialog box to see the animations that are associated with the object. Second, setting various parameters using menus and dialog boxes is time-consuming because the user must set each parameter individually.

To cope with these problems, we propose a new method that specifies animation effects by drawing effect lines. Effect lines are a popular technique that is used in comics and cartoons. They are auxiliary lines drawn adjacent to objects and characters to depict non-physical things, such as the emotions of characters and the movement of objects.

In addition, we can understand some parameters from their depiction. For example, speed lines are effect lines that represent the motions of objects in still pictures and they show the parameters of that motion, such as its speed, distance, and direction.

We designed several effect lines for specifying animations. They can be used to achieve the following simultaneously:

- Choose target objects: the user can choose target objects and where to draw them.
- Select motion effects: the user can select motion effects based on the type of effect line. It is also possible to combine effects by drawing different effect lines together.
- Set parameters: the user can set multiple parameters for a motion effect using the shapes of the lines.
- Help comprehension of animations: displaying effect lines on screen during editing can help the user to comprehend animation in a still picture. Furthermore, effect lines are as natural as a picture.

Figure 1 shows two examples of setting motion using our effect lines. The upper figure shows an effect that indicates the movement of a car to the left. In this example, the group of parallel lines represents movement to the left. The length and direction of the effect lines indicate the distance and direction of movement, respectively. The lower figure shows an effect that causes a question mark to tumble to the lower left. In this example, the spiral mark represents rolling. The length of the spiral line indicates the distance that the object rolls.

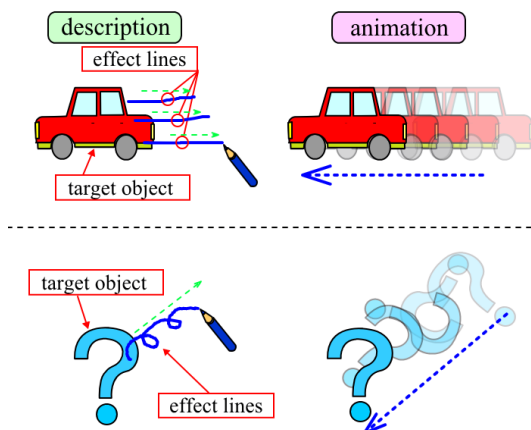


Figure 1. Defining animation with effect lines: top “move parallel”, bottom “roll”.

We have designed and implemented a prototype animation authoring system that uses effect lines, and used our system to create several animations. We also evaluated the advantages of our effect lines when setting up animation effects.

In the next section, we classify and explain the original effect lines that are used in comics and cartoons. We describe the design of the interface for our animation authoring system using effect lines in Section 3. Section 4 introduces the implementation of our prototype system and how to use it. Section 5 discusses how well effect lines work as input and output. Related work is mentioned in Section 6. Finally, we draw our conclusions in Section 7.

2 Effect lines

Several types of effect lines are commonly used in pictures such as those in comics and cartoons. These include speed, contour, and concentrated lines.

Speed lines Speed lines are drawn to depict the motion of objects. There are two types of speed line: many parallel (or nearly parallel) lines drawn as the background of a picture (Fig. 2(a)) or lines drawn behind an object (Fig. 2(b)).

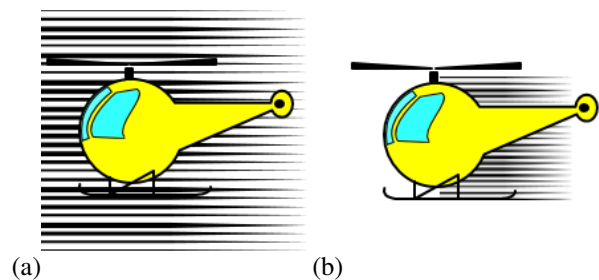


Figure 2. (a) Speed lines drawn as background to an entire picture. (b) Speed lines drawn only from an object.

Contour lines To depict the motion of objects, contour lines are also used. Full or partial contour repetitions are often used to depict the motion of an object (Fig. 3(a) and 3(b)). Their appearance is very dependent on the shape of the object.

Concentrated lines Concentrated lines converge on a certain object or point to an entire picture (Fig. 4). They are drawn to express the speed of an object or character, a feeling of tension, a feeling of oppression, and so on.

There are many other types of effect lines, which can be drawn to depict various effects. Figure 5 shows effect

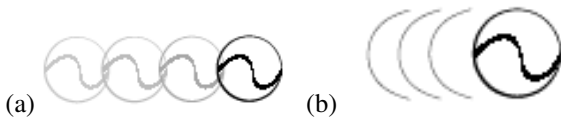


Figure 3. Contour lines[12]: (a) full contour repetition, (b) partial contour repetition.



Figure 4. An example of concentrated lines.

lines that depict the motion of a conductor's wand, singers singing, and joy.



Figure 5. An example of other types of effect lines

3 Design of the user interface

We designed interfaces to set animation effects by using effect lines. With these interfaces, we can edit key frames in an animation.

3.1 Basic effects

We designed seven kinds of effect lines as basic effects. Figure 6 names each effect and gives an example.

Move A pair of lines is extended from the target object in the opposite direction of movement. The distance

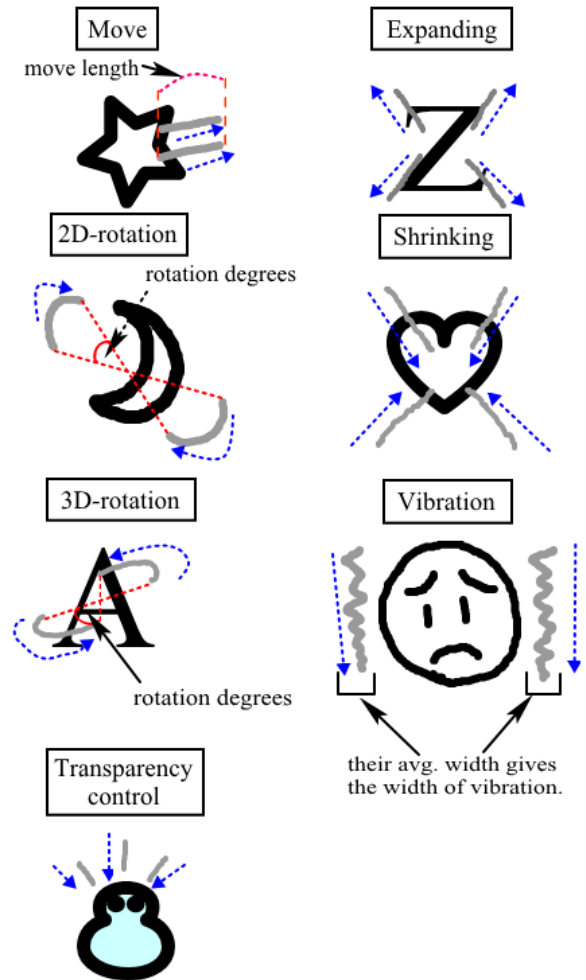


Figure 6. How to input the basic effects.

moved is represented by the length of the lines, and the speed by the number of lines.

2D/3D-rotation A pair of arcs surrounds the target object. If they do not overlap the target object, they represent 2D rotation. If they overlap the target object, they represent 3D rotation. The degree of rotation is determined by the length of the arcs, and the direction of rotation is determined by the direction of the stroke.

Transparency control A set of short lines is directed towards the target object. This produces the "appeal" or "fade-in" effect in PowerPoint. The speed with which the transparency decreases depends on the number of strokes.

Expanding/Shrinking Four lines overlapping the target object. These represent animation in which the target object expands or shrinks. The direction of the

strokes determines whether they represent expanding or shrinking: if they are drawn from the inside of the object outward, they represent expansion, and the opposite represents shrinkage. The initial size of the object depends on the starting points of the strokes. The final size of the object is the size of the target object.

Vibration A pair of wavy lines drawn on either side of the target object. The vibration direction is perpendicular to the direction of the wavy lines.

3.2 Combining animation effects

Effect lines contain information on the operation performed by the object. If several sets of effect lines are associated with an object in a given key frame, the effects defined by the effect lines must occur simultaneously. Therefore, when a user draws effect lines to an object that are associated with other effect lines, their effects are combined and applied to the object. For example, if we draw effect lines for “rotate +45 degrees” on an object that is already associated with the “move” effect, the object moves while rotating 45 degrees in the clockwise direction. We’ll show this example in Section 4.4.

3.3 Creating animations with effect lines

In general, when we author an animation using conventional authoring software, the initial state of the animation is considered. After setting the initial state, we define how objects change their state. By contrast, effect lines are usually drawn to an object that has completed an effect. As described above, effect lines contain information on the completed operation that has been enacted by the object. Therefore, we cannot set a possible future motion with effect lines. We propose a method of editing animation that uses effect lines. In it, the final state of the animation is prepared first, and then we define how the objects moved before reaching that state.

When a user draws effect lines on the editing screen, our method creates a new key frame that represents the state immediately before the effect occurred. For example, if the user draws the effect lines shown in the upper part of Figure 1, the system creates a key frame in which the car is in the initial location shown before it moves.

4 Prototype System

We have implemented a prototype animation authoring system.

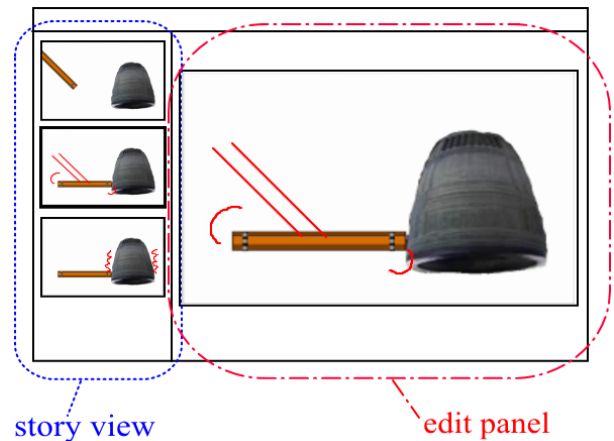


Figure 7. A screen image of our system and the "KANETUKI" (striking a bell) animation.

4.1 Overview of the user interface

Figure 7 shows an overview of the user interface in our authoring system. The system has an editing panel and a story view panel.

The editing panel is for editing and displaying key frames. It is also possible to preview the animation. The user positions objects and draws the effect lines to create an animation. The effect lines are visible only while editing key frames. When previewing animations, the effect lines are not displayed in this panel.

The story view panel shows the series of key frames created in the edit panel. This helps the user to understand the animations. For example, the story view in Figure 7 illustrates the following: (1) there is a bar and a bell; (2) the bar moves towards the bell, while rotating, and (3) strikes the bell, causing the bell to vibrate.

4.2 Drawing effect lines

To distinguish effect lines from other target objects, the user must change the input mode explicitly. To create objects, the user draws them by dragging while pressing the left mouse button (if a pen interface is used, then the pen button is not pressed). Conversely, to draw effect lines, the user drags the mouse while pressing the right button (or while pressing the button when using a pen interface). Whenever the user inputs effect lines, the system shows a preview of the effect.

It is possible to combine effects, as described in Section 3.2. For example, if we draw effect lines for “rotate +45 degrees” on an object that is already associated with the “move” effect, the object will move while rotating 45

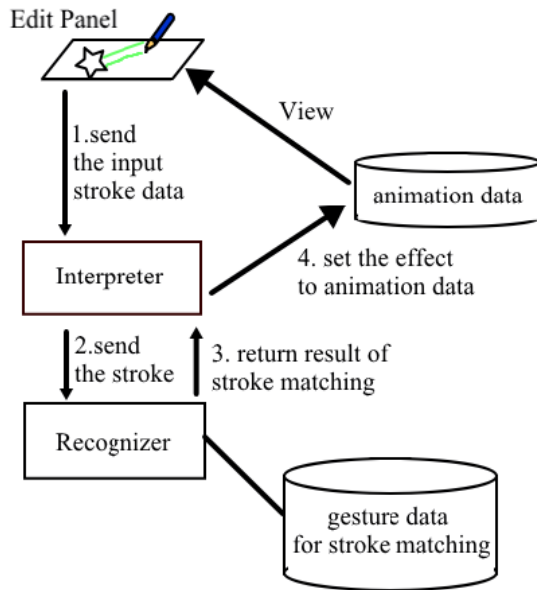


Figure 8. The components of this system and the process used to manipulate the user input.

degrees in the clockwise direction. We will show this example in Section 4.4.

4.3 System architecture

Figure 8 shows a schematic of the components in our system. The process of manipulating input effect lines is as follows:

1. The edit panel sends the input stroke to the interpreter as stroke data.
2. The interpreter sends the stroke to the recognizer in order to manipulate it.
3. The recognizer compares it with the data for stroke matching and returns the result to the interpreter.
4. The interpreter manipulates the stroke with the result and sets the effect as animation data.

To implement the interpreter and recognizer for the input strokes, we used the SATIN toolkit [7]. With SATIN and Quill [11], we can train the system to recognize types of strokes by providing examples of the strokes.

4.4 Example

Figure 9 shows an animation in which an airplane climbs and descends gradually, while moving to the right and rotating 45 degrees.

The user makes the animation in the following way:

- (A) The user selects an image of a plane from a file and puts it at the final position, where the plane will stop.
- (B) The user draws “move” effect lines behind the plane. This operation generates a new key frame that precedes the original key frame; the two key frames make an animation in which the plane moves from the end of the line to its current position.
- (C) Then, the user adds “rotate 45 degrees” effect lines to the last key frame. This added effect is merged with the already specified “move” effect, changing the angle of the airplane in the starting key frame.
- (D) The user switches from the current key frame to the previous one in the edit panel by using the slide bar or the “back” button. Then, the user draws “move” effect lines. This operation generates a new key frame that precedes the current key frame, and creates an animation in which the plane climbs to the upper-right.

5 Discussion

5.1 Evaluation

One of the merits of our method is that the user can combine basic effect lines to depict various animation effects, as shown in the example (“move” and “rotate +45 degrees”) in Figure 9.

We compared our method with the effects in the Microsoft PowerPoint menu in Office XP, in order to evaluate the animation effects generated by our technique. Since our method defines the “starting effect” only with effect lines, as stated in Section 3.3, we made our comparison with the starting effects in PowerPoint.

We designed seven basic animation effects (See Section 3.1 and Figure 6) and evaluated how many of the “starting effects” in PowerPoint we can create, when these basic effects and their parameters are defined using effect lines. For example, we can create “stretch” in the PowerPoint effect menu, which is a composite of “fade-in (decreasing transparency)” and “expansion”. Fourteen effects in PowerPoint involve using a mask, which is used to make an object partially transparent, such as the “dissolve-in”, “ease-in” and “wheel” effects. Since we cannot set the motion of these masks, we excluded these 14 effects from our comparison. Table 1 shows the results of the comparison.

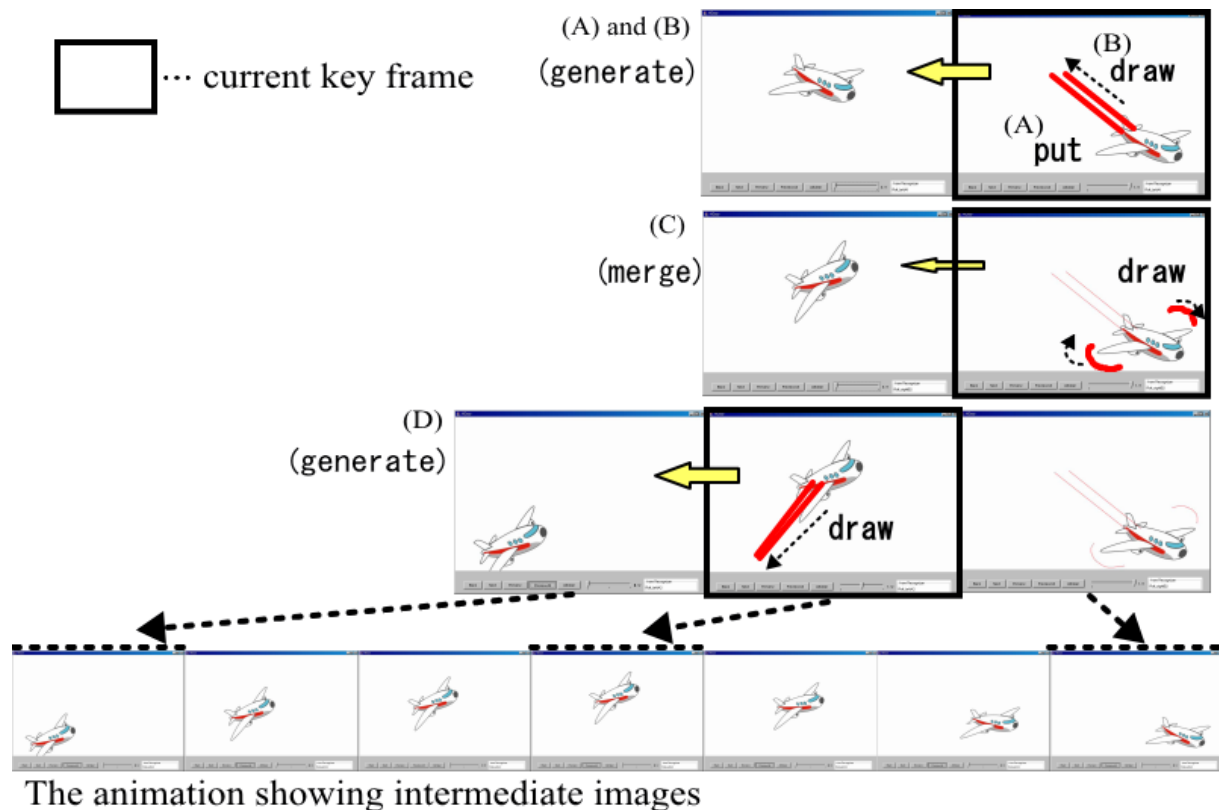


Figure 9. An example of animation authoring using our system.

There are 14 effects using a mask, which is used for making an object partially transparent, such as “dissolve-in”, “ease-in” and “wheel” effects. Since we cannot set the motion of these masks, we excepted those 14 effects from comparison targets.

We were able to create 32 of the 36 PowerPoint effects using our basic effects set. We could not change the color as in “type color” and “bound”, “light-speed”, and “swish” all required an acceleration parameter for their motion.

5.2 Expressiveness of effect lines

By drawing effect lines, the user can specify the target object, select the type of animation effect, and specify multiple parameters, simultaneously. Moreover, effect lines can be combined to make new effects. However, our experience in creating animations using this method has shown that some effects are difficult to depict with effect lines.

First, it is difficult to specify animation effects that cause the target object to vanish, such as whiteout, because effect lines are drawn to the final state of the object. We are considering two ways to cope with this problem. One is to incorporate “ghost objects” into our system so that we can include vanishing effects. The other method is to draw

such effects in the starting key frame. We plan to determine which is the better method.

Second, some animation effects are difficult to represent using “lines”. For example, our system currently does not provide a way to specify an effect that changes the color of objects or an effect that accelerates a motion. It is possible to use special lines to depict such effects, but this may not be an effective method. We plan to add more types of effect “lines” after studying various drawing techniques.

5.3 Effect lines as a static representation of animation

Effect lines displayed on each editing key frame help the user to understand the animation. Moreover, they naturally fit in still pictures. However, it is difficult to specify and understand the exact value of each parameter only from the shape of effect lines. It is also necessary to consider the method used to depict the effect lines. For example, we can’t recognize which the direction an object with “rotate +45 degrees” or “rotate -45 degrees” rotates to. We plan to consider methods of drawing lines with a pen that display the values of various parameters.

Able to create	appeal, crawl-in, slide-in, flash, expand, fade&zoom, fade, fade&turn (text only), up, compress, stretch, rise-up, unfold (text only), grow&turn, zoom, spinner, down, revolve, curve, credit-title, spiral-in, thread, boomerang, flip (text only), whip (text only), glider, sling, turn, pinwheel, fold, float, magnify.
Unable to create	type color (text only), bound, light-speed, swish (text only)

Table 1. The effects in PowerPoint that we were able (or unable) to create from the basic effects described in Section 3.1. “text only” means that the effect is applicable only to a text object.

6 Related work

Some developers have helped non-programmers, such as children, to create animated simulations or animations using pattern-matching methods. The user specifies animations using some before and after pictures that describe how the objects in the picture move. Such software includes Stagecast Creator [4], which is an expansion product of KidSim [5], Agent Sheets [14], and Viscuit [6]. However, it is difficult to combine rules in the same way as effect lines. For example, to combine “OBJECT X moves to the right” and “OBJECT X rolls”, a new rule must be defined that describes the combination of the two motions. Using effect lines, we can achieve this simply by drawing two effect lines into the picture.

Alice [3, 2] can create 3D character animations. To create an animation, users specify animation scripts that call animation commands in Alice’s animation library. Scripts can be created by writing text or by using the GUI, but it is still difficult for non-programmers, such as animation creators, to create animations in a language-based system such as Alice.

Some researchers have proposed methods for creating character animation that use a motion path with a 2D [1, 16, 15, 17] or 3D GUI [13]. Recently, this method has been adopted in Microsoft PowerPoint and Macromedia Flash. The user can specify the locus of objects directly, making it easy to specify an object’s motion in animation programs or 3D space. However, this method inputs only the motion path of an object.

Speed lines[8, 12], motion lines[10], cartoon blur[9], and

other methods can depict animation effects in still pictures. The flow visualization technique used to depict fluid movement has a similar purpose. However, they are not used to specify animations, which is the differentiating feature of our method.

7 Conclusion

We propose a method that uses effect lines to add animation effects in 2D animation authoring. The user can easily select a target object, choose the animation effects, and set their parameters simultaneously, in an intelligible way. Animation effects can easily be combined by drawing multiple sets of effect lines in the picture. In addition, displaying the effect lines on each key frame during on-screen editing allows the user to recognize animation. As we consider this method suitable for pen input, it can be used to create simple animations, design presentations, etc., on Tablet PCs or PDAs. Although we have not yet tested it fully, our method will help users to make animations more easily.

References

- [1] R. M. Baecker. Picture-driven Animation. In *Proceedings of AFIPS Spring Joint Comp. Conf.*, volume 34, pages 273–288, 1969.
- [2] M. Conway, S. Audia, T. Burnette, D. Cosgrove, and K. Christiansen. Alice: lessons learned from building a 3D system for novices. In *Proceedings of ACM SIGCHI2000*, pages 486–493. ACM SIGCHI, 2000.
- [3] M. Conway, R. Pausch, R. Gossweiler, and T. Burnette. Alice: A rapid prototyping system for building virtual environments. In *Proceedings of ACM CHI’94 Conference on Human Factors in Computing Systems*, volume 2, pages 295–296. ACM SIGCHI, 1994.
- [4] A. Cypher. Stagecast Creator. <http://www.acypher.com/creator/>.
- [5] A. Cypher and D. C. Smith. KidSim: end user programming of simulations. In *Proceedings of the SIGCHI*, pages 27–34. ACM SIGCHI, 1995.
- [6] Y. Harada and R. Potter. Fuzzy rewriting - soft program semantics for children. In *IEEE Symposium on Human-Centric Computing Languages and Environments*, pages 39 – 46, Oct. 28–31 2003.
- [7] J. I. Hong and J. A. Landay. SATIN: A toolkit for informal ink-based applications. In *Proceedings of ACM UIST2000*, pages 63–72.
- [8] S. C. HSU and I. H. H. LEE. Drawing and animation using skeletal strokes. In *Proceedings of ACM SIGGRAPH*, pages 109–118. ACM SIGGRAPH, 1994.
- [9] Y. Kawagishi, K. Hatuyama, and K. Kondo. Cartoon blur: Non-photorealistic motion blur. In *Proceedings of Computer Graphics International 2003*, pages 276–281, June 2003.
- [10] A. Lake, C. Marshall, M. Hariis, and M. Blackstein. Stylized rendering techniques for scalable real-time 3D animation. In

Proceedings of ACM SIGGRAPH 2000, pages 13–22. ACM SIGGRAPH, 2000.

- [11] A. C. Long, Jr. *Quill: a Gesture Design Tool for Pen-based User Interfaces*. PhD thesis, University of California at Berkeley, Berkeley, CA, Dec. 2001.
- [12] M. Masuch, S. Schlechtweg, and R. Schulz. Speedlines: Depicting Motion in Motionless Pictures. In *SIGGRAPH'99 Conference Abstracts and Applications*, page 277. ACM SIGGRAPH, 1999.
- [13] N. Osawa and K. Asai. An immersive path editor for keyframe animation using hand direct manipulation. In *Proceedings of the Seventh International Conference on Information Visualization*, pages 524–529. IEEE, 2003.
- [14] A. Repenning and T. Samner. Agent Sheets: A medium for creating domain-oriented visual languages. *Computer*, 28:17–25, 1995.
- [15] J. T. Stasko. The path-transition paradigm: a practical methodology for adding animation to program interfaces. *Journal of Visual Languages and Computing*, pages 213–236, Sept. 1990.
- [16] J. T. Stasko. Tango: A framework and system for algorithm animation. *IEEE Computer*, 23(9):27–39, Sept. 1990.
- [17] J. T. Stasko. Using direct manipulation to build algorithm animations by demonstration. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 307–314. ACM SIGCHI, 1991.