Motion Amplifiers: Sketching Dynamic Illustrations Using the Principles of 2D Animation

Rubaiat Habib Kazi, Tovi Grossman, Nobuyuki Umetani, George Fitzmaurice

Autodesk Research

{rubaiat.habib, tovi.grossman, nobuyuki.umetani, george.fitzmaurice}@autodesk.com

Figure 1. Motion amplifiers produce the stylized and exaggerated dynamics of classical 2D animations.

ABSTRACT

We present a sketching tool for crafting animated illustrations that contain the exaggerated dynamics of stylized 2D animations. The system provides a set of motion amplifiers which implement a set of established principles of 2D animation. These amplifiers break down a complex animation effect into independent, understandable chunks. Each amplifier imposes deformations to an underlying grid, which in turn updates the corresponding strokes. Users can combine these amplifiers at will when applying them to an existing animation, promoting rapid experimentation. By leveraging the freeform nature of sketching, our system allows users to rapidly sketch, record motion, explore exaggerated dynamics using the amplifiers, and fine-tune their animations. Practical results confirm that users with no prior experience in animation can produce expressive animated illustrations quickly and easily.

Author Keywords

Sketching; amplifiers; principles of animation; stylized.

ACM Classification Keywords

H.5.2. [Information interfaces and presentation]: User Interfaces - Graphical user interfaces.

"*Animation can explain whatever the mind can conceive. This facility makes it the most versatile and explicit means of communication yet devised for quick mass appreciation.*"
– Walt Disney [27]

CHI'16, May 07-12, 2016, San Jose, CA, USA
 ISBN 978-1-4503-3362-7/16/05…$15.00
DOI: http://dx.doi.org/10.1145/2858036.2858386

INTRODUCTION

In the early days of 2D animation, crude technical limitations compelled master animators to push the limits of the medium by developing a number of expressive techniques [27]. By exaggerating the dynamics of the physical world, these techniques, known as the *principles of animation* [27], turned hand-drawn 2D animation into a communicative, sophisticated art form. As described by Thomas and Johnston [27], twelve principles of animation guide the motion design, visual design, as well as the clarity and communicative aspects of 2D animation. A master animator can compose the performance (animation) of a character by applying these principles to various extents.

The importance of 2D animation stylization is well recognized, and it has been widely used in 3D animation [20]. However, producing stylized 2D animations is tedious, even today, requiring specialized skill and manual key-framing. In the computer graphics community, researchers have explored a number of approaches to simulate cartoon-style animation [1, 2, 4, 10, 30]. However, much of the focus of these works is to simulate a particular effect or a small subset of effects, and they provide little artistic freedom [30].

To make animations more accessible, a number of performance-based casual animation authoring tools using sketching [8, 12, 18] and direct manipulation [22] have been developed. While these tools empower end users, the resulting animations typically lack the expressiveness of classical 2D animations due to the absence of the hand-drawn animation principles used by master animators. Most notably, animated objects in such applications are typically rigid [8, 18], and the continuous deformations required to realize the above-described principles on animations can only be coarsely approximated [15].

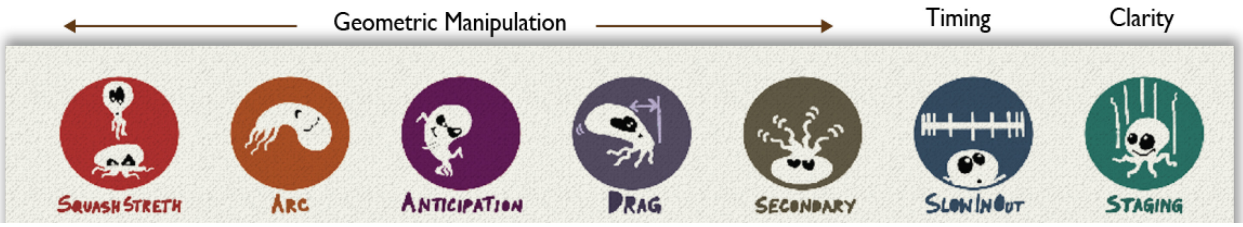

Figure 2: We devised the principles of animation as a set of seven motion amplifiers. The amplifiers are independent from each other and can be arbitrarily mixed to design motion effects of an animated object. These tools correspond to the principles of animation that address the geometric manipulation, clarity, and temporal aspects of animation.

In this paper, we adopt and expose the animation principles as a set of concrete, high-level motion design tools (Figure 2) displayed as icons (or visual metaphors). We refer to these motion design tools as *amplifiers*, since they enhance the communicative aspects of 2D animation. The design of these amplifiers conforms to the natural language of traditional animation and facilitates rapid exploration of the animation principles in a way not possible before. The result is a sketching tool that enables amateurs, as well as experts, to sketch expressive 2D illustrations with looping animations using these amplifiers.

With our system, users can sketch drawings and record motions through direct manipulation, and they can apply the animation principles by simply tapping on the corresponding *amplifier* icons. Each amplifier imposes deformations to an underlying grid, which in turn updates the corresponding strokes. Users can adjust the strength of the amplifiers and manipulate the motion profiles of the animation for finer artistic control. Our design enables users to define and control the dynamic simulation of an object through sketch-based interaction techniques. Our implementation is purely geometric and does not require skeletal information, connectivity, or pre-existing data, thus providing artistic freedom and expressiveness. Animated illustrations created with our tool exhibit the exaggerated dynamics of classical 2D animations, which are tedious for experts and almost impossible for amateurs to create with existing tools. Our user evaluation confirmed the benefits of the system. Even first time users with no prior animation experience are able to create expressive, dynamic illustrations in minutes.

RELATED WORK

This section reviews prior work in stylized computer animation, existing animation tools for novices, and the emergence of dynamic illustrations as a new medium.

Simulating Stylized Animation Techniques

Incorporating stylized animation principles into traditional animation techniques (i.e., key-framing or procedural animation) is a challenging problem. The problem of characterization is foremost, since these principles do not strictly follow physical rules. Researchers in the computer graphics community have explored a variety of techniques to simulate stylized animations.

Physically-based models have been previously explored to simulate stylized animation effects (e.g., *squash and stretch*, *follow through*) based on elastic deformations and collision handling. Simulating such deformation accurately is possible [10], but it often lacks artistic freedom and expressiveness. Moreover, Stylized animations often diverge from physical realism.

The use of implicit surface deformations [1] and time-warping methods [4] has also been explored to exhibit cartoon-style animation effects. Other approaches have leveraged example-based [19, 21] or data-driven techniques [2] to guide the deformation and animation of an object. In contrast, we provide flexible tools to design exaggerated and imaginary motions based on empirical animation principles, precluding the use of existing motion capture data and examples.

Techniques from image and signal processing methods have also been applied to modify and adapt animated motion. Bruderlin and Williams [3] introduced multiresolution motion filtering techniques to edit, modify, blend, and align motion parameters for articulated figures, thus enabling higher-level motion editing with many degrees of freedom. In a similar vein, the *Cartoon Animation Filter* [30] modulates an arbitrary input motion signal to exhibit stylized animation effects in the output motion, taking a single parameter as input. We drew inspiration from these works in terms of providing high-level motion controls to simplify the tasks of animators; however, the lack of modularity in the interfaces of the animation effects limits expressiveness.

Process: Performance Driven Animation Tools in HCI

Crafting stylized and expressive 2D animation remains tedious and difficult. The HCI and graphics communities have explored the use of performance-driven techniques for easy-to-use animation authoring tools using sketching, direct manipulation, and motion capturing [2, 9]. These are complementary to traditional key-framing and procedural animation techniques. In motion sketching systems, the user sketches the motion path to animate the object [8], textures [18], and even the articulated figures [28]. Other works have looked into sketching dynamic line of actions (poses) for animating 3D characters [12, 13], which often exhibit stylized animation effects. For deformable objects, direct manipulation interfaces enable users to construct and manipulate the handles to animate the shape by relying on their natural sense of timing [22]. While these tools have proven to be effective in disseminating the power of animation to amateurs, in general, the resulting artifacts lack the expressiveness of classical 2D animations created by

master animators. In contrast, we leverage performance-based animation techniques and enable users to compound animations by applying the amplifiers in the post-processing stage.

Dynamic Illustrations: An Emerging New Media

Dynamic and interactive pictures are becoming increasingly popular among artists, scientists, and designers to enrich the contents found in e-books, websites, and teaching materials. Researchers have explored a variety of techniques and user interfaces to create video textures [26], cinemagraphs [16], and animated illustrations [5, 18], all of which exhibit continuous animation effects while preserving the unique, timeless nature of still pictures. Interactive illustrations (driven by data and parameters) [17, 29] provide a more playful and informative experience by responding to the user's input. Researchers have examined sketch-based interfaces for the authoring of dynamic illustrations for exploring and visualizing dynamic systems [17, 31]. These tools allow users to leverage the communicative aspects of animation and interactive storytelling in a spectrum of application domains. Inspired by this line of work, our design explores the avenue that falls between traditional illustration and animation tools for crafting stylized and expressive animated illustrations.

THE PRINCIPLES OF 2D ANIMATION

The foundation of our design is based on twelve principles of 2D animation illustrated in the seminal book "*The Illusion of Life*" [27]. We also interviewed two professional animators (*A1*, *A2*) with more than 25 years of industry experience to guide our design and better understand the principles in practice with state-of-the-art tools.

In our design, we classify the twelve principles of animation into five broad categories. These categories include principles related to the **geometric manipulation** of objects over time (*squash and stretch*, *arcs*, *secondary motion*, *follow through*, *anticipation*), **temporal controls** (*slow-in and slow-out, timing*), **visual design** (*solid drawing, appeal*), **clarity and communication** (*staging, exaggeration*), and the **process of animation** (*straight ahead action, pose to pose*). Curious readers are encouraged to consult with the book, "The Illusion of Life" [27], for more details.

However, characterizing some of these principles can be extremely challenging. For instance, the principle of *staging* communicates the attitude, mood, or idea on the animation. *Staging* directs the audience's attention to the most important idea or object. Another example is *exaggeration,* which accentuates the essence of an idea via the design and the action. Characterizing these principles of **clarity and communication** is challenging, since they deal with higher level aspects and emotions. The application of these principles is subject to an artist's preference, and style. We also do not address the principles related to **visual design**, since they are not directly associated with motion design.

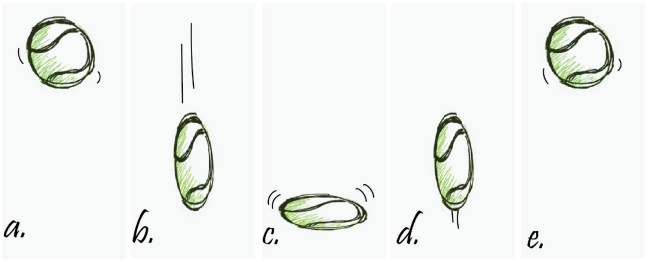

Figure 3: A classic example of *squash and stretch*. As the ball falls, it stretches. When it impacts the ground, it goes from stretch directly to squash and then goes back to stretch before returning to its initial position, creating a *bouncing* effect.

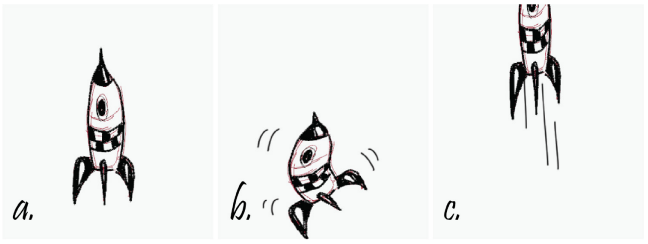

Figure 4: The principle of *anticipation* is demonstrated by the rocket moving downwards before going up. It prepares the audience for a major action.

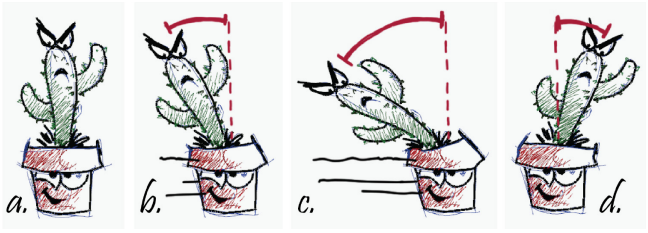

Figure 5: *Follow through.* (a) The main body (pot) and plant in rest position. (b)-(c) The movement of the flower plant is delayed (*follow through*) in relation to the main body (pot). (d) When the main body (pot) stops, the flower plant continues to move forward before settling back (*overlapping action*).

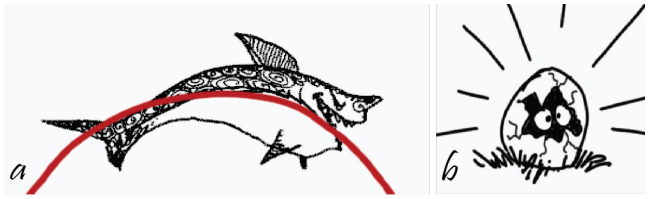

Figure 6: (a) The principle of *arcs*. (b) The principle of *staging* (surrounding strokes) highlights the egg (right).

In this paper, we address the principles related to the **geometric manipulation** of objects, **temporal controls,** and partially **clarity and communication**.

While these 12 principles of animation have long been understood in the artistic community, today's user interfaces for animation provide a plethora of low level controls, which are tedious and cumbersome to use and foreign to animators. Gilland summarizes this disconnect:

"*In the world of computer generated imagery, much of the academic language of physics has carried through into the animation software that has been developed in the last two decades. This phenomenon occurred for the simple reason that the initial designers and creators of this software were indeed scientists, not animators or artists. Unfortunately… there was a disconnect between the two disciplines, as the majority of software developers seemed to be unware of the fact that animation had already been evolving a language of its own for the past 60/70 years.*" – Joseph Gilland [11]

This disconnect hinders the rapid experimentation of strong motion design and exaggeration dynamics, which is an essential aspect for learning and mastering of classical 2D animations. As such, for individuals doing traditional 2D animations, these principles of animation are often forgotten (*A1*). We seek to remedy this by introducing amplifiers to the user interface that map directly to a chosen subset of the described animation principles.

DESIGN GOALS

Our observations and discussion with professionals elicited several interesting insights that were important to account for in our own designs.

Language of Animation in User Interfaces

Animators often develop their own terminologies to characterize certain animation procedures or effects (e.g., "Get more stretch on him," "Look at the anticipation in the drawing!") [11, 27]. The professionals (*A1*, *A2*) we consulted stressed the need for UI metaphors that resonate with how animators understand, learn, and execute animations.

Sketch-based UI for Design and Rapid Exploration

Animation is the art of exaggeration and dynamics, pushing the limits of motion design with continuous experimentation [11, 27]. Our goal is to facilitate the intuitive experimental aspects of motion design by leveraging the free-form nature of sketching and utilizing user's intuitive sense of time and space. This is complimentary to a majority of the traditional animation tools that divide spatial and temporal controls.

Exposure to the Principles of Animation

We intended to facilitate an experimental approach to learning and mastering animation [24]. In our design, amateurs not familiar with the principles of 2D animation should be exposed to them in such a way that they can easily apply the principles in the spirit of playful learning.

Looping Illustrations

A casual search in popular community art sites (e.g., *Deviantart*, *Tumblr*) reveals a growing interest in animated illustrations (mostly in .GIF format). Our goal was to explore the rich design space of dynamic illustrations without burdening users with the complexity of timeline-based interaction techniques.

Modularity and Independence

The modularity of the animation principles enables an artist to deconstruct a complex animation into a number of understandable components that can be mixed in a variety of ways. Preserving modularity and independence is key to the successful execution of the animation principles. It also preserves artistic expressiveness.

SYSTEM UI AND INTERACTION

We now present our new sketch-based interface for crafting animated illustrations using the principles of 2D animation. The system employs performance-driven animation techniques by capitalizing on the free-form nature of sketching and direct manipulation. The absence of a timeline orients the overall user experience to a 2D illustration tool, rather than a traditional 2D animation tool. The resulting artifacts are illustrations with looping animations.

The interface contains a main authoring canvas and a toolbar on the right consisting of two tabs (Figure 7). The top of the main canvas displays the 7 amplifiers for adding motion effects. The *Tools* tab consists of several standard drawing tools. The *Motion Controls* tab provides control parameters for the amplifiers. Figure 10 illustrates the full workflow of animating with the system, which we now describe.

Figure 7: The system interface comprising the main canvas and the global toolbar for drawing tools (left) and motion amplifier control tools (right). Three amplifiers are in effect for the selected object.

Drawing and Motion Recording

The user begins by sketching the picture to be animated using a pen tool. The user then selects the group of strokes to create an *object*. The canvas can have many objects at any one time. When an object is selected, a contextual radial menu displays the translation and rotation icons and pivot point (Figure 10b). The user can record a translation or rotation motion by directly manipulating the object through the corresponding icons (Figure 10c, Figure 12d).

Amplifiers: Tools for Motion Effects Design

When an object is assigned a motion (translation or rotation), the user interface displays the seven motion amplifiers (Figure 7). These amplifiers correspond to seven of the twelve principles of animation. The first five amplifiers are associated to geometric manipulation (*squash and stretch*, *arcs*, *anticipation*, *follow through*, and *secondary motion*), the sixth *amplifier* addresses temporal aspects (*slow-in and slow-out*), and the final *amplifier* is an embodiment of clarity and communication (*staging*).

We not address the principles related to visual design (*solid drawing, appeal*), since they are not directly associated with animation. We also omit *Exaggeration* as it is a broad concept that is challenging to characterize. For process of animation (*straight ahead action* and *pose to pose*), we employed performance driven animation techniques, which are complimentary to key-framing techniques. Finally, while *timing* does not have a corresponding amplifier, it is supported through a motion profile tool which is described later.

The user can activate an *amplifier* by tapping the corresponding icon, and the resulting animation will exhibit the corresponding effects immediately. This encourages rapid experimentation and allows users to explore various combinations of amplifiers. The amplifiers are independent from each other, thus preserving expressiveness and modularity. Activating a particular *amplifier* may reveal contextual UI controls in the canvas (Figure 10g). Below, we discuss each *amplifier* in detail.

Slow-in and slow-out

This *amplifier* slows down the object at the beginning and at the end of the trajectory. When activated, the system displays a corresponding motion profile widget at the bottom of the canvas (Figure 7). A motion profile defines the value of a parameter (in this case, velocity) along the trajectory of the motion path [18]. For rotation, the motion profile defines the value of the parameter for an oscillating cycle. By default, the *slow-in and slow-out* profile has a bell-shaped curve, implying that the object starts at a slow pace, reaches its peak velocity in the mid-point of its motion and then slows down again.

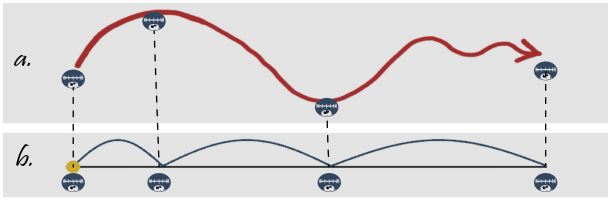

Figure 8: (a) Injecting s*low-in and slow-out break-points* by clicking directly on the motion path. (b) The corresponding motion (velocity) profile.

Slow-in and slow-out is a default effect in many animation tools (e.g., PowerPoint). However, in our system, the user can inject additional s*low-in and slow-out* effects along a trajectory by directly tapping on the motion path (Figure 8a). The user can override the profile by sketching a new profile inside the widget. The ability to quickly refine the velocity by sketching, along with immediate feedback, facilitates rapid exploration. A yellow control point, displayed on both the in-canvas path, and the motion profile curve, allows the user to fine-tune the effect and coordinate it with an object's position (Figure 9) in an interactive and iterative way. Together, the motion profiles, immediate feedback, and control point address the *timing* principle, as described in the principles of animation [27].

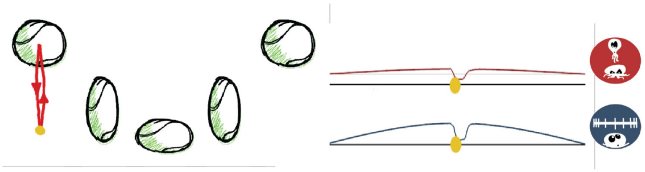

Figure 9: The bouncing ball displaying the motion path and control point (left). The appropriate *squash and stretch* (up-right) and *slow-in and slow-out* profile (bottom-right) achieves the desired effect.

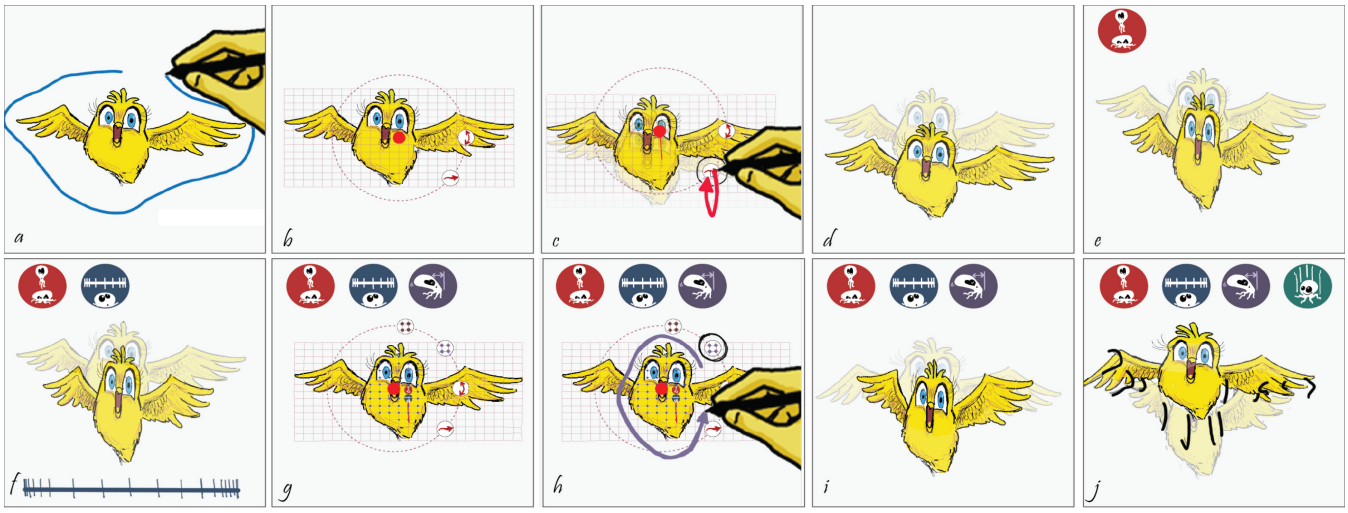

Figure 10: Workflow for animating an object. (a) The user sketches a bird and creates an object by lassoing the strokes (b) A radial menu surrounds the selected object. (c-d) The user records a motion (moving up and down) by direct manipulation. (e) Applying *squash and stretch* amplifier stretches the object along the motion direction. (f) The user activates the *slow-in and slow-out* amplifier. (g) Activating the *follow through* amplifier displays two additional icons in the radial menu. (h) The user specifies the *rigid region* of the bird using the *Rigid* tool. (i) The wings of the bird (*passive region*) now exhibit *follow through* behavior. (j) Adding the *staging* amplifier displays motion trails dynamically.

Squash and Stretch

This effect stretches the object in the direction of the motion path trajectory. Similar to *slow-in and slow-out*, a motion profile widget controls the amount of stretch along the trajectory of the motion path. By default, the object starts from its normal position, stretches out to its peak value in the mid-point of the trajectory and reverts back to its normal position at the end. The user can override the profile by sketching and inserting additional *squash and stretch break points* on the motion path, similar to *slow-in slow-out breakpoints* in Figure 8. A grey line in the motion profile indicates the neutral value. Any value above the grey line stretches the object and below the grey line squashes the object (Figure 9(right)). A slider controls the overall strength of *squash and stretch* of the selected object.

In general, *squash* is a consequence of a collision or abrupt change in direction. Our system does not explicitly handle collisions between objects. However, a collision can be easily simulated by setting the motion profiles appropriately. For example, Figure 9 shows how a user can create the classic example of the bouncing ball by configuring the motion profiles of the *squash and stretch* and *slow-in and slow-out* amplifiers.

Follow through

Follow through enables users to define the parts of the object that are loose (e.g., clothing, antenna of a car), squishy, or prone to independent movement (e.g., large stomach, loose skin) (Figure 12a-e). When activated, the radial menu displays two additional icons: *rigid* and *stationary*. The user can specify the *rigid* and *stationary* regions of the object through sketching. The *rigid* and *stationary* regions cannot deform, and the *stationary* part is also fixed in its location. The remaining part of the object is considered *passive.* The *passive* region exhibits *follow through* by catching up with the *rigid* region. A *follow through strength* slider controls the amount of stiffness of the object to simulate diversified material properties.

Arcs

In cartoon drawings and animations, artists often use a line of action to describe the character's principal shape and pose [13]. When *arcs* is activated, the system dynamically deforms the object's computed line of action along the trajectory of the motion path (Figure 11), while preserving the volume of the object.

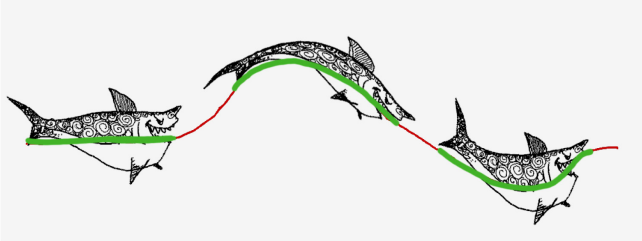

Figure 11: The *arcs amplifier* dynamically deforms a computed line of action (green) along the motion path (red).

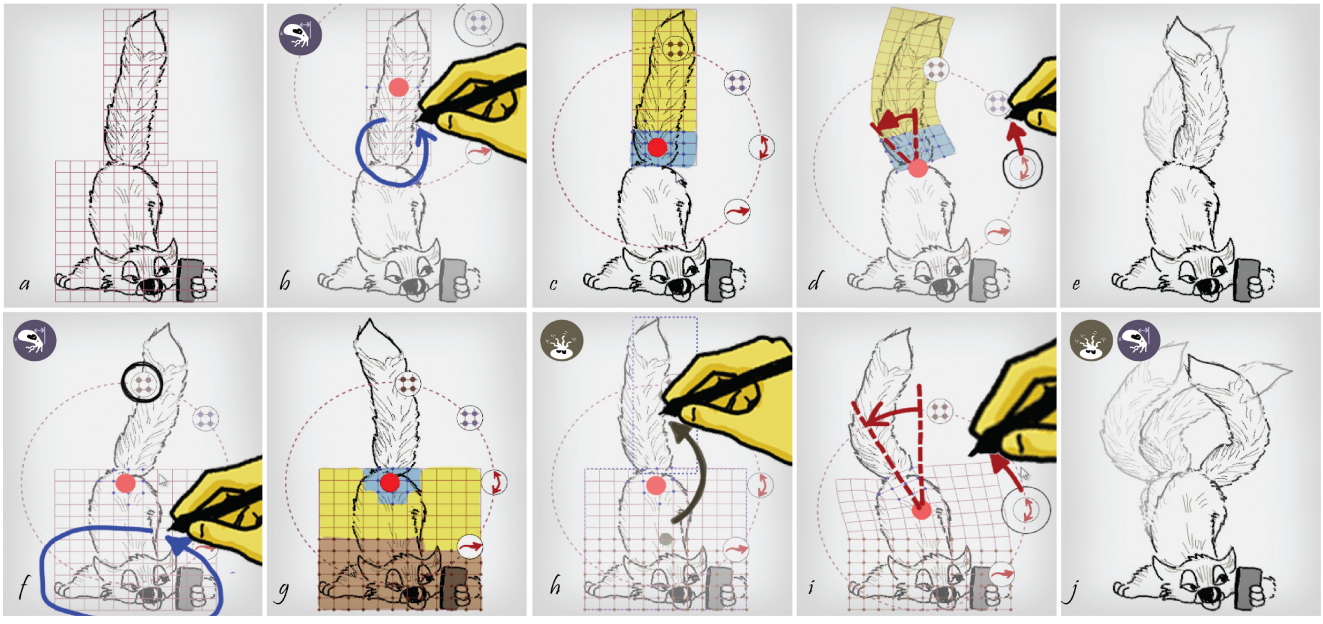

Figure 12: (a) *Follow through* and *secondary motion* using multiple objects. (b-c) The user activates the *follow through* amplifier. The user then sketches the *rigid region* (blue) of the object. The remaining part is *passive* (yellow). (d) The user then records a *rotation* motion. (e) The *rigid region* is now oscillating while the *passive region* catches up with the *rigid region*. (f) The user then selects the other object (main body) and specifies the *stationary region* of the object. (g) The *stationary*, *rigid*, and *passive* regions of the object. (h) The user then activates the *secondary motion* and defines the tail as a secondary object to the fox (primary object). (i) Recording a *rotation* motion for the primary object. (j) In the resulting animation, the hierarchical *follow through* (or deformation) of the tail (secondary object) adds more dimension to the animation.

Secondary motion

This amplifier supplements a main action by adding a secondary motion (Figure 12f-j). When activated, the user can sketch a stroke from the selected object to another object to define a parent-child relationship (Figure 12h). The location of the child object remains fixed to the relative position of its parent. As such, the child object exhibits hierarchical *follow through*, as determined by the motion of its parent object (Figure 12j).

Anticipation

The anticipation amplifier pulls the object backwards before the primary action to build momentum (Figure 13). To better articulate the desired behavior, the user can sketch a stroke that defines the target deformation of the object during *anticipation*. We refer to this strokes as *anticipation pose*. The implementation of *anticipation* is volume preserving. Thus, when the length of the *anticipation pose* is greater than the length of the line of action, it exhibits stretch behavior (Figure 13f). Otherwise, it exhibits squash behavior (Figure 13a-b). The *anticipation strength* slider controls the amount of adherence of the deformation to the *anticipation pose* (Figure 13d-f).

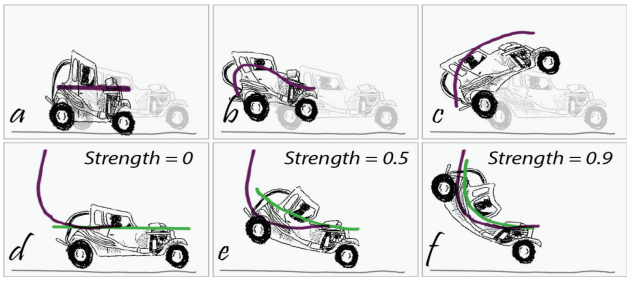

Figure 13: The anticipation amplifier pulls the object backwards before the primary action to build momentum (a) The default *anticipation pose* (violet). (b-c) Other *anticipation poses* sketched by the user. (d-f) The *anticipation strength* drives a computed line of action (green) of the object to the target *anticipation pose*.

Staging

Staging is a very broad concept and can be achieved in many ways. In our system, we employ *staging* by exhibiting motion trails along an object's trajectory, which have been often used to stylize animated cartoons and visually enhance the perception of motion [14, 25]. A *staging threshold* slider controls the velocity threshold for displaying the motion trails. The *staging density* and *staging length* sliders control the density and length of a motion path respectively (Figure 14).

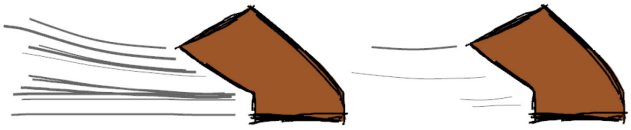

Figure 14: *Staging* adds dynamic trails. The *density* and *length* slider controls the number and length of the trails.

IMPLEMENTATION

Our implementation uses OpenGL in C++. The interface is developed with Qt (a C++ cross platform UI framework).

The strokes are stored as vector graphics. Once an *Object* (consisting of a group of strokes) is created, the bounding box of the *Object* is discretized into a square uniform grid that is orthogonal to the screen (Figure 16a). Other than *staging*, each amplifier controls the movement and deformation of the background grid. In turn, the strokes inside the grid are updated via bilinear interpolation of the enclosing grid. In our current implementation, there is no overlapping between the objects.

We use a shape matching technique [23] for the deformation of the background grid. The shape matching technique is physically less accurate compared to nonlinear finite element methods. However, it is desirable in our application due to its speed, controllability, and stability. Our implementation prevents inversion by making the grid stiffer when it is close to being inverted. The specific implementation of each amplifier is described below.

Slow-in and slow-out applies a default motion profile to the objects velocity, but does not apply any deformations. In *slow-in and slow-out* motion profile, the velocity (v) of the object is computed as a function of the corresponding motion path position (p) (Figure 15). If the p_t is the position and v_t is the velocity of the object at time $\boldsymbol{t}$, the new position of the object at step $\boldsymbol{t}$+1 is computed as follows.

$$v_t = \text{motion\_profile}(p_t)$$
$$p_{t+1} = p_t + v_t$$

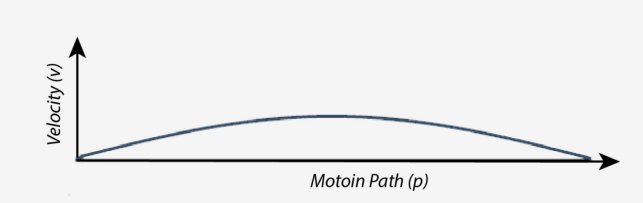

Figure 15: *Slow-in and slow-out* motion profile curve. The velocity of the object at any position in the path is computed from the corresponding motion profile curve.

Squash and stretch is implemented by adjusting the horizontal (h_s) and vertical (v_s) scaling factor of the grid, as determined by the effects motion profile and motion direction. To calculate the scale factors, we project the tangent vector of the trajectory into the x axis ($\mathbf{p}_x$) and y axis ($\mathbf{p}_y$). If the object is at position $\boldsymbol{t}$ ($0 \leq \boldsymbol{t} \leq 1$) in the motion path, the value of squash-and-stretch ($\boldsymbol{v}_{ss}$) is computed from the corresponding motion profile curve. Thus, the scale of the x and y axes are computed as follows.

$$\boldsymbol{v}_{ss} = \text{s\&s\_profile}(\boldsymbol{t}) * (\text{length}(\mathbf{p}_x) - \text{length}(\mathbf{p}_y))$$
$$h_s = 1 + \boldsymbol{v}_{ss}$$
$$v_s = 1 / h_s$$

where, $-0.5 \leq \boldsymbol{v}_{ss} \leq 0.5$. If an object traverses horizontally, then $\text{length}(\mathbf{p}_x) > \text{length}(\mathbf{p}_y)$, and the horizontal scale (h_s) would be

greater than 1, implying that the object stretches along the x axis. This technique works well for horizontal and vertical motions. However, the limitation of this method is that when the velocity direction is roughly diagonal, the x and y projections are almost equal. In that case, the object doesn't exhibit *squash-and-stretch*. To alleviate this problem, implemented a method that stretches the individual grid points in the passive region towards the motion direction. But, the directional stretch implementation exhibits unpleasant and undesirable results. This is due to the fact that, by design, we do not deform the rigid grid points, which creates undesirable effects in the boundary of rigid and passive regions. Exhibiting sophisticated *squash and stretch* for arbitrary motion directions remains as a future work. For rotation, we linearly map the motion profile curve to a single oscillation cycle, where v_{ss} is a function of current angular position of the object.

$$\boldsymbol{v}_{ss} = \text{s\&s\_profile } (\boldsymbol{angle})$$

For *follow through*, our implementation uniformly animates the grid points inside the *rigid region*. The deviation from the equilibrium (initial position in Figure 16a) causes forces to accelerate the grid points in the *passive region* back to the equilibrium configuration (Figure 16b). The grid points are represented as a set of particles. In every step, each particle (in the *passive region*) is pulled towards its goal position using the shape matching technique [23]. The grid stiffness is controlled with the *follow-through strength* slider, and the other parameters were manually tuned. The purely geometric nature of this technique simplifies the pre-processing step and requires no skeletal or connectivity information.

Secondary motion is implemented by locking the grid of the child object to the orientation and position of the parent object. In this implementation, in the parent grid, we compute the nearest grid point to the center of the child grid. The global transformation (position and orientation) of the child object is determined by the position and orientation of that grid point in parent grid. The *secondary motion* then adds hierarchical deformation to the child grid with the *follow through* effect.

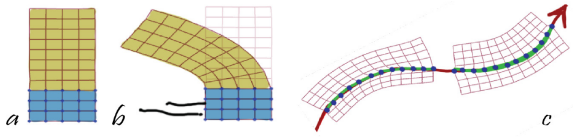

Figure 16: a) Objects are discretized into a square uniform grid. The object in initial position. (b) When the *rigid region* (blue) starts moving, the *passive region* (yellow) lags behind for inertia. (c) The imaginary line of action (green) is moved along the motion path by moving the *rigid* points (blue) of the grid.

For *arcs*, we define the centered horizontal axis of an object's grid as an imaginary line of action for that object in its rest (initial) position. When animating with *arcs*, the rigid points in the line of action are moved along the motion path trajectory (Figure 16c), while the remaining grid points follow passively and deform the object.

For *anticipation*, we linearly interpolate the line of action (i.e., the grid points in the horizontal axis) between the default position (horizontal axis at rest position) and the target *anticipation pose*. If there are n-1 horizontal grid cells, let the initial positions of the n rigid grid points in the horizontal axis is $\mathbf{i}(s)$ $[0 \leq s \leq n]$. We then uniformly segment the resulting *anticipation pose* stroke into n-1 segments, and compute the target positions $\mathbf{t}(s)$ $[0 \leq s \leq n]$ for each of the rigid grid points. During *anticipation* phase, the position of the rigid points $\mathbf{p}(s)$ are computed by interpolation between $\mathbf{i}(s)$ and $\mathbf{t}(s)$ (Figure 17).

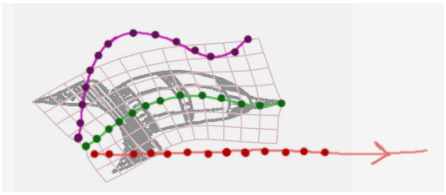

Figure 17: The line of action (green) is computed by interpolating between the initial horizontal axis (red) and *anticipation pose* (violet).

The *anticipation strength* slider controls the strength of attraction to the anticipation pose. We did not implement *anticipation* for rotation in our current prototype.

For *staging,* we first compute the solid (with overlapping strokes) and void grid cells (Figure 18a). We then sample a set of solid grid points from the outer surface of the opposite motion direction (Figure 18b). The dynamic motion trails emanate from these grid points. The motion trails are computed by storing the positions of the corresponding grid points for a given number of time-frames (Figure 18c). We randomize the thickness, position, and length of the trails for stylization.

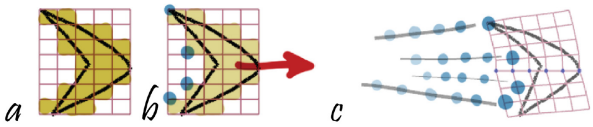

Figure 18: (a) The solid grid cells (yellow). (b) Selected solid grid points (blue) as sources of trails. (c) Trails are computed by sampling the grid points in time.

Finally, when *arcs, slow-in and slow-out*, and *squash and stretch* are all used in conjunction, we use a time-warping [4] method to lag the back end of the object in time behind the front. This creates a non-linear *squash and stretch* effect.

USER EVALUATION

We conducted a user evaluation with both professional animators and illustrators, to gain insights about our tool. This study was also used to gather feedback on how our system compares to existing approaches.

Participants

Six participants (3 males), aged 22 to 44 years old (average 32), took part in the study. Half the participants had no prior experience with animation tools (*P3*, *P4, P6*). Two

participants were professional animators (*P1*, *P5*), two were industrial designers (*P3*, *P4*), and the remaining two were illustrators (*P2*, *P6*). All the participants had moderate to good drawing skills.

Study Protocol

All the experiments were conducted using the Wacom CINTIQ 21ux tablet display. The evaluation period lasted for approximate 1.5 hours for each participant and consisted of the following steps:

Overview and training (20~25 minutes). After filling out a background questionnaire, each participant was given a brief overview of the principles of animation. The instructor walked the participant through a step-by-step tutorial to familiarize the participant with the system.

Exercise Task (15~20 minutes). In this step, participants were given an exercise task consisting of three animated drawings and asked to reproduce the given animations (Figure 1c, 8, and 11). The animated objects covered all the seven motion amplifiers and motion profiles. Participants were prompted with the video of the target effects on a separate display, which they could use for reference. No time limit was imposed and the facilitator did not intervene unless the participant had trouble completing the task. Our goal was to observe whether the participants could independently reproduce a target effect using the system.

Freeform Exploration and Feedback (30~40 minutes): The participants were free to explore the tool to create dynamic illustrations of their own. Finally, the participants filled out a questionnaire to provide feedback about the system.

Results and Discussion

Overall, the participants found the interface to be easy, playful, novel, and useful. Participants' average rating of the system's overall ease of usage was 4.16 out of 5 (min 4). Regarding the design of the amplifiers:

> *P4: "It's about how do you re-interpret these technical (animation) terms into functions in a way that is easy and accessible to an average user."*

The idea of using the language of animation was refreshing to some participants (*P1*, *P4*, *P5*), since the animators didn't have to translate their ideas into technical terms.

> *P5: "I work with a lot of people who came from being a classical animator for 20 years or so, and, as soon as someone from the Effects department is talking about the technical terms, they just phase out. They don't want to hear it, unless you put it in terms they understand. It's great to be able to put it in terms that artists understand."*

Feedback indicated that the modularity and independence of the amplifiers provides more artistic freedom (*P1*, *P4*, *P5*) and helps users to see what each amplifier adds to the animation (*P1*, *P3*, *P5*). These aspects potentially make it suitable for a learning environment.

> *P1: "The UI is going to teach them (animators) in the language they are reading [in] the book. It helps them to associate better and learn faster".*

Participants were pleased to see that they were able to quickly craft animated illustrations with exaggerated dynamics, even if they were not previously familiar with the principles of animation. Participants also reported that unlike traditional animation tools (e.g., Maya), the interface was streamlined so that they did not have to make guesses (*P1*, *P2*, *P4*). *P3* stated that sketching as a way of animating combines the best of drawing (Photoshop) and animation (After Effects) tools.

> *P5: "The 2D productions these days seem to have lost a bit of the organic look that came from the hand drawn 2D animations. Because, they are using software that just move the objects on a surface or plane. Whereas, this is bringing back a lot of that classical animation."*

Our participants had mixed reaction about the expressiveness of the system. *P1*, *P2*, *P3*, and *P5* felt that the tool opened up many possibilities for both amateurs and professionals and provided an industry standard baseline for 2D stylized animated illustrations. While *P4* and *P6* believed that adding a timeline and more fine-tuning capabilities would open up this tool for high-end professionals. The participants expressed their desire to use the tool for animation practice (*P1*, *P5*), presentations (*P1*, *P3*, *P4*), GIF animations for the web (*P2*, *P6*), icons (*P2*, *P3*, *P4*), and spot illustrations (*P2*).

Exercise Task Performance

On average, the three exercise tasks were completed in a total of 11:45 minutes (min 7:10 minutes, max 17:00 minutes). The first two animations were completed by all participants without any assistance. For the third animation (Figure 1c-d), the three participants (*P3*, *P4*, *P6*) with no prior animation experience required some assistance to complete the task. This was due to the inability to modularize the animation conceptually. The three participants completed the cart animation and correctly created a secondary motion relation to the rat, but they had trouble with the relation order and the deformation behavior of the rat (which is secondary to the cart). Once explained, they were able to complete the task. Overall, the outcomes of the exercise task confirmed the ease of usage of our system.

Freeform Usage Observations

Our participants authored a range of animated illustrations with motion amplifiers in the freeform usage stage. In Figure 19a, *P6* used *squash and stretch*, *slow-in and slow-out*, and *follow through* to animate a jumping rabbit. The motion profiles helped the participant synchronize the effects. Another participant (*P1*) recreated one of his frame-by-frame hand drawn 2D animation that he had done 15 years ago (Figure 19b). *P5* sketched an animated character waving his hands (Figure 20). The *follow through* and *secondary motion* in the hand added more dimension to the animation. We were pleased to see that the participants used all seven of the amplifiers across their animated drawings.

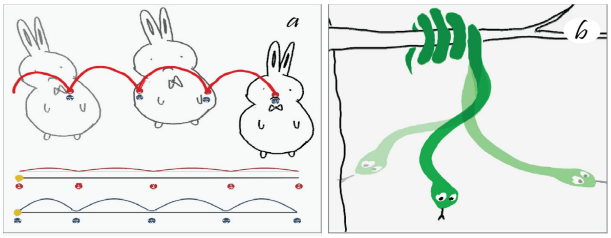

Figure 19: (a) A jumping rabbit. (b) An animated snake.

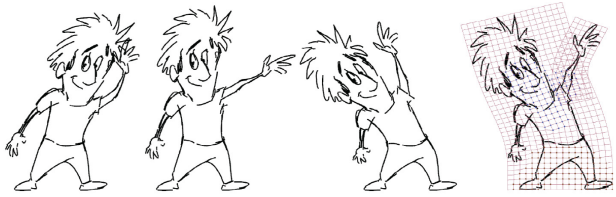

Figure 20: An animated character waving his hand.

Limitations and Future Work

While our participants were mostly positive, they also pointed out some limitations that could guide future enhancements. Several of our participants (*P2*, *P5*) recommended having collision controls between objects.

While our tool enables a wide range of animation effects, it has its limitations compared to purely hand-drawn 2D animations. For instance, animation effects consisting of topological changes (Figure 21a) cannot be achieved. Addressing such effects would require us to employ vector graphics with time-varying topology [7]. While our exploration was conducted in the context of looping dynamic illustrations, it would be interesting to explore the use of amplifiers in traditional key-framing techniques (*P5*).

The algorithms which we used worked very well for the majority of animations. However, our solver sometimes exhibits glitches and unexpected behavior in sharp curvatures (Figure 21b) and hierarchical deformations. We hope to explore refinements to our algorithms to address these conditions.

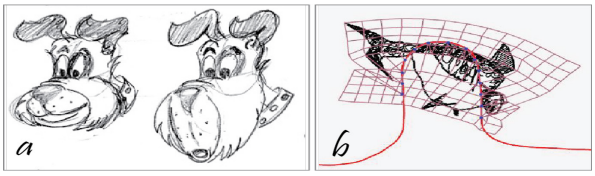

Figure 21: (a) As the character changes from squash to stretch position, the mouth changes from a line to an ellipse. (b) Our implementation can create unexpected artifacts in sharp turns.

In the future, we also plan to facilitate animation coordination between multiple objects, as suggested by *P1*, *P2*, and *P4*. In the absence of a global timeline, one potential approach is to sketch functional relationships between objects' attributes, as demonstrated in Kitty [17]. It would also be interesting to analyze the input motion path through multi-stage segmentation [28] and automatically compute the motion profiles and default amplifiers. We averted these automated suggestions to preserve expressiveness.

Discussion

"[*In animation*] *there are so many areas to be explored … effects to be created, new wonders to be seen… someone or some group of artists will surely discover new dimensions to delight and entertain the world.*" [27]

The motion amplifiers are attributed to the principles of animation that had been devised and nurtured by Disney animators in the 1940s. Beyond Disney, master animators and studios across different cultures and eras have devised their own animation styles and techniques. For instance, the 2D animations by Ryan Woodward depict a distinct fluid-like motion transitions from frame-to-frame. Is this phenomena considered as an animation principle, or, is it a style? What is the boundary between animation styles and principles? We conjecture that the principles of animation present a particular artistic style. The characterization of artistic styles (visual and motion design) into concrete motion design tools is an interesting avenue for future exploration. However, such characterization is often very challenging.

The principles of animation exemplify how a complex, stylized art form can be decomposed into understandable chunks. Seymour Papert refers to this as breaking down a complex problem into "mind-size bites," which makes knowledge more communicable, more assimilable, more simply constructible [24]. User interfaces for animation that leverage these "mind-size bites" as a natural vocabulary to understand and compose complex animations might have powerful influence on how animators think, learn, and explore.

To elevate this art form to an even higher level, clearly this characterization is inadequate. Can we design tools that not only explain existing principles but also inspire animators to innovate new ones? This remains as an open question for us and our readers to consider.

CONCLUSION

We have presented a novel sketch-based interface for crafting stylized animated illustrations by leveraging the exaggerated dynamics of the principles of 2D animation. Our design formulates the principles of animation as a set of motion amplifiers that can be combined at will. The key idea in our design was to leverage the language of animation as a natural way to think and communicate with computers through freeform sketching. The goal of our design is to increase one's ability to modularize the fundamental concepts of 2D animation and combine them together in new and powerful ways. By representing the principles of animation in a simplified manner, our system offers users, particularly those with no prior experience in animation, the opportunity to rapidly explore animation effects and produce expressive animated illustrations.

REFERENCES

1. Bajaj, C. (1997). *Introduction to implicit surfaces*. Morgan Kaufmann.
2. Bregler, C., Loeb, L., Chuang, E., & Deshpande, H. (2002, July). Turning to the masters: motion capturing cartoons. In *ACM Transactions on Graphics (TOG)*(Vol. 21, No. 3, pp. 399-407). ACM.
3. Bruderlin, A., & Williams, L. (1995, September). Motion signal processing. In Proceedings of the 22nd annual conference on Computer graphics and interactive techniques (pp. 97-104). ACM.
4. Chenney, S., Pingel, M., Iverson, R., & Szymanski, M. (2002, June). Simulating cartoon style animation. In Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering (pp. 133-138). ACM.
5. Chuang, Y. Y., Goldman, D. B., Zheng, K. C., Curless, B., Salesin, D. H., & Szeliski, R. (2005, July). Animating pictures with stochastic motion textures. In*ACM Transactions on Graphics (TOG)* (Vol. 24, No. 3, pp. 853-860). ACM.
6. Coros, S., Martin, S., Thomaszewski, B., Schumacher, C., Sumner, R., & Gross, M. (2012). Deformable objects alive!. *ACM Transactions on Graphics (TOG)*, *31*(4), 69.
7. Dalstein, B., Ronfard, R., & Van de Panne, M. (2015). Vector Graphics Animation with Time-Varying Topology. *ACM Transactions on Graphics*, 12.
8. Davis, R. C., Colwell, B., & Landay, J. A. (2008, April). K-sketch: A kinetic sketch pad for novice animators. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 413-422). ACM.
9. Dontcheva, M., Yngve, G. and Popović, Z., 2003. Layered acting for character animation. *ACM Transactions on Graphics (TOG)*, *22*(3), pp.409-416.
10. Faloutsos, P., Van De Panne, M., & Terzopoulos, D. (1997). Dynamic free-form deformations for animation synthesis. *Visualization and Computer Graphics, IEEE Transactions on*, *3*(3), 201-214.
11. Gilland, J. (2012). *Elemental Magic: The Art of Special Effects Animation*. CRC Press.
12. Guay, M., Ronfard, R., Gleicher, M., & Cani, M. P. (2015). Space-time sketching of character animation. *ACM Transactions on Graphics (TOG)*, *34*(4), 1.
13. Guay, M., Ronfard, R., Gleicher, M., & Cani, M. P. (2015, May). Adding dynamics to sketch-based character animations. In *Sketch-Based Interfaces and Modeling (SBIM) 2015*.
14. Haller, M., Hanl, C., & Diephuis, J. (2004). Non-photorealistic rendering techniques for motion in computer games. *Computers in Entertainment (CIE)*,*2*(4), 11-11.
15. Igarashi, T., Moscovich, T., & Hughes, J. F. (2005). As-rigid-as-possible shape manipulation. *ACM transactions on Graphics (TOG)*, *24*(3), 1134-1141.
16. Joshi, N., Mehta, S., Drucker, S., Stollnitz, E., Hoppe, H., Uyttendaele, M., & Cohen, M. (2012, October). Cliplets: juxtaposing still and dynamic imagery. In *Proceedings of the 25th annual ACM UIST* (pp. 251-260). ACM.
17. Kazi, R. H., Chevalier, F., Grossman, T., & Fitzmaurice, G. (2014, October). Kitty: Sketching dynamic and interactive illustrations. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (pp. 395-405). ACM.
18. Kazi, R. H., Chevalier, F., Grossman, T., Zhao, S., & Fitzmaurice, G. (2014, April). Draco: Bringing life to illustrations with kinetic textures. In Proceedings of the 32nd annual ACM conference on Human factors in computing systems (pp. 351-360). ACM.
19. Koyama, Y., Takayama, K., Umetani, N., & Igarashi, T. (2012, July). Real-time example-based elastic deformation. In *Proceedings of the 11th ACM SIGGRAPH/Eurographics conference on Computer Animation* (pp. 19-24). Eurographics Association.
20. Lasseter, J. (1987, August). Principles of traditional animation applied to 3D computer animation. In ACM Siggraph Computer Graphics (Vol. 21, No. 4, pp. 35-44). ACM.
21. Martin, S., Thomaszewski, B., Grinspun, E., & Gross, M. (2011). Example-based elastic materials. *ACM Transactions on Graphics (TOG)*, *30*(4), 72.
22. Moscovich, T., Igarashi, T., Rekimoto, J., Fukuchi, K., & Hughes, J. F. (2005). A multi-finger interface for performance animation of deformable drawings.*Proc. of User Interface Software and Technology (UIST'05), Seattle, WA, ACM Press*.
23. Müller, M., Heidelberger, B., Teschner, M., & Gross, M. (2005, July). Meshless deformations based on shape matching. In ACM Transactions on Graphics (TOG) (Vol. 24, No. 3, pp. 471-478). ACM.
24. Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
25. Schmid, J., Sumner, R. W., Bowles, H., & Gross, M. (2010). Programmable motion effects. *SIGGRAPH'10: ACM SIGGRAPH 2010 papers*, 1-9.
26. Schödl, A., Szeliski, R., Salesin, D. H., & Essa, I. (2000, July). Video textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (pp. 489-498). ACM Press/Addison-Wesley Publishing Co..
27. Thomas, F., Johnston, O., & Frank. Thomas. (1995). The illusion of life: Disney animation (pp. 306-312). New York: Hyperion.
28. Thorne, M., Burke, D., & van de Panne, M. (2007, August). Motion doodles: an interface for sketching character motion. In ACM SIGGRAPH 2007 courses (p. 24). ACM.
29. Victor, B. (2009). Drawing Dynamic Visualizations. *CUSE*.
30. Wang, J., Drucker, S. M., Agrawala, M., & Cohen, M. F. (2006, July). The cartoon animation filter. In ACM Transactions on Graphics (TOG) (Vol. 25, No. 3, pp. 1169-1173). ACM.
31. Zhu, B., Iwata, M., Haraguchi, R., Ashihara, T., Umetani, N., Igarashi, T., & Nakazawa, K. (2011, December). Sketch-based dynamic illustration of fluid systems. In *ACM Transactions on Graphics (TOG)* (Vol. 30, No. 6, p. 134). ACM.